Preface

About ZoneRanger

The ZoneRanger appliance provides enterprises with a mechanism for extending the reach of management applications into firewall-protected networks. ZoneRanger, working together with the Ranger Gateway software component, serves as a proxy firewall for management traffic, simultaneously enabling the flow of management traffic between applications and devices, and inspecting/filtering the traffic in order to mitigate security risks.

ZoneRanger has been designed for demilitarized zones (DMZs), which are the most common form of firewall-protected network, but can also be used in any situation where firewalls prevent the normal flow of management protocol traffic.

About this Guide

This guide provides detailed information intended to help users successfully deploy, configure and operate ZoneRangers within their networks. Topics covered include:

- The ZoneRanger architecture
- Deployment arrangements and options for ZoneRanger and Ranger Gateway
- Foundational concepts and mechanisms required to understand, install, configure, and operate ZoneRangers and Ranger Gateways
- Proxy and management services provided by ZoneRanger
- ZoneRanger and Ranger Gateway user interfaces

Audience

This guide is intended for persons with a good understanding of network protocols and the management of networks in general.

Technical Support

Technical assistance is available when you purchase a support contract.

Support covers the ZoneRanger device, the Ranger Gateway software, and the technical documentation. Please have the ZoneRanger serial number located on the back of the device ready before calling Tavve technical support.

You can contact Tavve technical support at:

- E-mail: support@tavve.com
- Telephone: +1 919-654-1235
- Fax: +1 919-380-7147
Document Organization

This guide is divided into six sections:

- Part I, *ZoneRanger and Ranger Gateway Overview*, describes the architecture of the ZoneRanger and Ranger Gateway within a network environment. This section provides the reader with a framework for how the ZoneRanger and Ranger Gateway are deployed and operate within the enterprise.

- Part II, *ZoneRanger Concepts*, describes the high level concepts employed by the ZoneRanger and Ranger Gateway. Understanding of ZoneRanger and Ranger Gateway concepts is a necessary prerequisite for the rest of this guide.

- Part III, *ZoneRanger Services*, describes the proxy and management services provided by ZoneRanger and Ranger Gateway. The functionality of each service is described in detail, along with associated deployment and configuration options.

- Part IV, *ZoneRanger User Interfaces*, provides reference documentation for the various user interfaces used to configure and operate the ZoneRanger and the Ranger Gateway.

- Part V, *ZoneRanger Applications*, describes separately licensed ZoneRanger feature sets for specific management applications.

- *Appendices*, provides additional information on miscellaneous topics associated with ZoneRanger and Ranger Gateway.
# Table of Contents

Preface ........................................................................................................................................................................... 4  

Part I: ZoneRanger and Ranger Gateway Overview ........................................................................................................ 8  
Chapter 1: Zone Ranger and Ranger Gateway Architecture ......................................................................................... 8  

Part II: ZoneRanger Concepts ......................................................................................................................................... 13  
Chapter 2: Address Patterns ......................................................................................................................................... 15  
Chapter 3: Address Transforms ..................................................................................................................................... 16  
Chapter 4: Audit ............................................................................................................................................................... 18  
Chapter 5: Backups/Profiles ........................................................................................................................................... 20  
Chapter 6: Destination Groups ......................................................................................................................................... 22  
Chapter 7: Device Groups ............................................................................................................................................... 23  
Chapter 8: GV1/RGVI ....................................................................................................................................................... 25  
Chapter 9: IPv6 ................................................................................................................................................................. 32  
Chapter 10: Joining ......................................................................................................................................................... 33  
Chapter 11: Licensing ...................................................................................................................................................... 35  
Chapter 12: Managed Nodes ........................................................................................................................................... 36  
Chapter 13: Node Groups ............................................................................................................................................... 37  
Chapter 14: Pooling/Redundancy/VIP/Grouping ............................................................................................................... 38  
Chapter 15: Proxy Access Control .................................................................................................................................. 42  
Chapter 16: Proxy Caching .............................................................................................................................................. 47  
Chapter 17: Proxy Map .................................................................................................................................................... 49  
Chapter 18: Server Groups ............................................................................................................................................ 56  
Chapter 19: Whitelisting .................................................................................................................................................. 59  

Part III: ZoneRanger Services ........................................................................................................................................... 60  
Chapter 20: Discovery ..................................................................................................................................................... 60  
Chapter 21: Forwarding ................................................................................................................................................... 63  
Chapter 22: FTP Proxy ..................................................................................................................................................... 67  
Chapter 23: HTTP/HTTPS Proxy .................................................................................................................................... 69  
Chapter 24: ICMP Proxy .................................................................................................................................................. 75  
Chapter 25: NTP Proxy .................................................................................................................................................... 76  
Chapter 26: Polling ............................................................................................................................................................ 79  
Chapter 27: Root Cause .................................................................................................................................................. 81  
Chapter 28: SNMP Proxy ................................................................................................................................................. 82  
Chapter 29: TCP Proxy .................................................................................................................................................... 93  
Chapter 30: Telnet/SSH Proxy ......................................................................................................................................... 95  
Chapter 31: TACACS+/RADIUS Proxy .......................................................................................................................... 104  
Chapter 32: TFTP Proxy .................................................................................................................................................. 110  
Chapter 33: Traffic Monitoring .................................................................................................................................... 112  
Chapter 34: Whitelisting ............................................................................................................................................... 113  

Part IV: ZoneRanger and Ranger Gateway Interfaces .................................................................................................. 114  
Chapter 35: ZoneRanger Web Interface ........................................................................................................................ 115  
Chapter 36: Ranger Gateway Viewer ........................................................................................................................... 214  
Chapter 37: ZoneRanger Text Interface ........................................................................................................................ 248  
Chapter 38: Ranger Gateway Command Interface .................................................................................................... 300  

Part V: ZoneRanger Applications ................................................................................................................................... 354  
Chapter 39: HP OM ......................................................................................................................................................... 355  
Chapter 40: Web File ...................................................................................................................................................... 361  
Chapter 41: Generic Inbound TCP Proxy ........................................................................................................................ 367  

Appendices ........................................................................................................................................................................ 374  
A. SNMP Agent ................................................................................................................................................................. 374  
B. ZoneRanger and Ranger Gateway Traps .................................................................................................................... 381
Part I. ZoneRanger and Ranger Gateway Overview

Chapter 1: Zone Ranger and Ranger Gateway Architecture

Introduction and Deployment Architecture

ZoneRanger is a hardware appliance or a virtual appliance (VM) that provides enterprises with a mechanism for extending the reach of management applications into firewall-protected networks. ZoneRanger, working together with the Ranger Gateway software component, serves as a proxy firewall for management traffic, simultaneously enabling the flow of management traffic between applications and devices, and inspecting/filtering the traffic in order to mitigate security risks.

ZoneRangers are typically installed in a network zones, such as a DMZ, where there are devices to be managed that management applications are unable to reach due to firewall-based network partitioning. The Ranger Gateway software component is typically installed on the same server as the management application, and acts as the interface between the management application and one or more ZoneRangers. Ranger Gateway functions as a transparent proxy, intercepting and relaying management protocol traffic addressed for managed devices, so that the management application can remain unaware that the Ranger Gateway and ZoneRanger are being used. As a result, ZoneRanger and Ranger Gateway can be used with a wide variety of management applications.

A simple ZoneRanger configuration is illustrated in the following figure.

![Figure 1-1. Simple ZoneRanger configuration](image)

Note that the Ranger Gateway software is installed on the same server as the management application (e.g. CiscoWorks), and that the ZoneRanger is installed in the remote network alongside the managed devices. The Ranger Gateway and ZoneRanger communicate using a single SSL-encrypted TCP connection. All management protocol traffic being proxied through the ZoneRanger is multiplexed over this single connection, resulting in a dramatic reduction in firewall rules and associated configuration effort.
The following figure shows a more advanced ZoneRanger deployment scenario.

In this scenario there are multiple remote networks to be managed (i.e. multiple DMZ's), and multiple management applications. A redundant pair of ZoneRangers is installed in each DMZ, and instances of the Ranger Gateway (RG) software have been installed on the majority of the management application servers. An SSL-encrypted TCP connection is maintained between each Ranger Gateway instance and each ZoneRanger, so that each management application is able to reach all of the DMZ devices that need to be managed. As a result, there is a many-to-many relationship between Ranger Gateways and ZoneRangers: each Ranger Gateway instance can be joined to multiple ZoneRangers and each ZoneRanger can be joined to multiple Ranger Gateways.

The figure also shows two management application servers, one with CiscoSecure ACS and one with a Trap/Syslog Receiver, that do not have the Ranger Gateway software installed, but instead interact with the ZoneRangers using Ranger Gateway software installed on another server. Depending on the nature of the management application, the management protocols being used, and the server hardware involved, this simplified approach may be advantageous in some situations. In most cases, however, installing the Ranger Gateway on the same server as the management application is the preferred approach.

Ranger Gateway software can be installed on any of the various hardware platforms that support the following operating systems:

- Centos 5.2 or later
- Red Hat Enterprise Linux version 4.0 or higher
- Solaris 2.8 or higher
- SuSE Linux version 11.1 or higher
- Windows 2008 Server R2, Windows 2012 Server

**ZoneRanger Services**

The primary function of the ZoneRanger is to act as an application-layer proxy firewall for the protocols most typically used by management applications. ZoneRanger provides proxy services covering a variety of protocol scenarios:
• Request/response protocols, where the requests are originated by the management application:
  – ICMP
  – SNMP
• Request/response protocols, where the requests are originated by the managed devices:
  – TACACS+
  – RADIUS
  – NTP
  – Generic TCP
• Session-oriented protocols, where the sessions are initiated by the management applications:
  – SSH
  – HTTPS
• Event/notification protocols, where events are generated by managed devices and are filtered and forwarded to management applications:
  – SNMP
  – Syslog
  – NetFlow
  – sFlow
  – Generic UDP
• File transfer protocols, where the transfers are initiated by the management applications:
  – FTP
  – TFTP

ZoneRanger’s proxy services are transparent, in that management applications are not specifically aware that the Ranger Gateway and ZoneRanger are being used, and do not need to be configured in a special way in order to incorporate the use of the proxy. This approach simplifies management application configuration, and enables ZoneRanger and Ranger Gateway to be used with a wide variety of management applications.

In addition to its role as a management proxy firewall, ZoneRanger can also be configured to act as a remote management station, performing network discovery, IP, SNMP, and TCP polling, and root cause analysis. As a result, users have the option of having their management applications poll their managed devices, with the ZoneRanger acting as a proxy firewall for the polling traffic, or having the ZoneRanger poll the managed devices, and forward status traps to their management applications.

**User Interfaces**

Each ZoneRanger and Ranger Gateway instance can be configured and administered using a text-based, command-line style user interface, or a graphical user interface. The user interfaces for ZoneRanger and Ranger Gateway are described further in the following sections.

---

1 ZoneRanger also provides limited support for TFTP transfers initiated by managed devices.
**ZoneRanger User Interfaces**

ZoneRanger provides two primary user interfaces:

- A web based interface that can be accessed directly via HTTP on TCP port 80 or HTTPS on TCP port 443, or via proxy through a joined Ranger Gateway.
- A text-based command-line interface that can be accessed directly via Telnet on TCP port 23, or SSH on TCP port 22, or via proxy through a joined Ranger Gateway.

When accessing the ZoneRanger through either the web interface or text interface, you will need to authenticate with a login ID and password. ZoneRangers support two authorization levels (Administrator and Operator) and are shipped with the following initial user configuration:

<table>
<thead>
<tr>
<th>User Name</th>
<th>Password</th>
<th>Access Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>admin</td>
<td>Administration</td>
</tr>
<tr>
<td>operator</td>
<td>operator</td>
<td>Operator</td>
</tr>
</tbody>
</table>

When using the web interface, the administrator authorization level enables access to all ZoneRanger configuration and operation menus. Users having operator access are able to view the ZoneRanger status, and information regarding managed devices and associated protocol traffic, but are not allowed to configure the ZoneRanger or initiate administrative actions. For security, users are automatically logged out of the ZoneRanger web interface after 30 minutes of inactivity. Reference documentation for the ZoneRanger web interface is provided in Chapter 30.

The text interface requires administrator authorization privileges. Users with operator authorization are not permitted to access the ZoneRanger text interface. Reference documentation for the ZoneRanger text interface is provided in Chapter 32.

**Ranger Gateway User Interfaces**

Ranger Gateway provides two user interfaces:

1. A GUI-based client application, called the Ranger Gateway Viewer.
2. A text-based command-line interface, consisting of a set of commands that can be executed from an operating system shell on the server where the Ranger Gateway has been installed.

The Ranger Gateway command interface is implemented as a set of executable commands, installed in a bin directory. In the case of Solaris and Linux, this directory also includes a command to start the Ranger Gateway Viewer application. The location of this bin directory is dependent on the operating system:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>install_dir/gateway/bin</td>
</tr>
<tr>
<td>Solaris</td>
<td>install_dir/gateway/bin</td>
</tr>
<tr>
<td>Windows</td>
<td>install_dir\bin</td>
</tr>
</tbody>
</table>

where <install_dir> is the directory where the Ranger Gateway software was installed.

To start the Ranger Gateway Viewer on a Solaris or Linux system, run the RangerGateway command from an operating system shell. To start the Ranger Gateway Viewer on a Windows system, select **Programs > Tavve > Ranger Gateway Viewer** from the Start menu. Reference documentation for the Ranger Gateway Viewer is provided in Chapter 33.
All of the commands which comprise the Ranger Gateway command interface are executed from an operating system shell. You can execute any command with a -? argument to display the usage information for that command, as illustrated in the following figure.

![Ranger Gateway Command example](image)

**Figure 1-3. Ranger Gateway Command example**

Reference documentation for the Ranger Gateway command interface is provided in Chapter 36.
Part II. ZoneRanger Concepts

This section introduces and describes foundational concepts and mechanisms that are important for ZoneRanger users to understand, so that they can properly configure and administer their ZoneRangers, and obtain maximum value from their ZoneRanger deployments. Later chapters will build on these concepts and mechanisms and will assume that the reader is reasonably familiar with the content of this section.

The following concepts and mechanisms are discussed:

- **Address Patterns** – IP address or hostname values that contain wild card characters or range descriptions and thus can describe a range of IP addresses or hostnames.
- **Address Transforms** – Rules that specify how to transform IP addresses or hostnames to new values (e.g. to accommodate NAT).
- **Audit** – The automated process whereby Ranger Gateway and ZoneRanger perform periodic self-checks, and, where necessary, perform corrective actions.
- **Backups/Profiles** – Mechanisms for saving and restoring all or part of a ZoneRanger and Ranger Gateway configuration.
- **Destination Groups** – ZoneRanger mechanism for defined named groups of Ranger Gateways and UDP packet destinations to be applied to forwarding rules as a means to create and manage fewer rules.
- **Device Groups** – Ranger Gateway mechanism for defining named groups of managed devices, and using these named groups in a variety of configuration rules.
- **Gateway Virtual Interface (GVI) and Remote Gateway Virtual Interface (RGVI)** – Mechanisms whereby the Ranger Gateway intercepts requests generated by management applications that are destined for managed devices, so that these requests can be relayed through a ZoneRanger to the target devices.
- **IPv6** – If the ZoneRanger has a ZR-SPX license, it can be configured to be installed in an IPv6 network. When the ZoneRanger is configured in an IPv6 network, IPv6 addresses may be used in proxy addresses.
- **Joining** – The process by which a working association between Ranger Gateway and a ZoneRanger is established.
- **Managed Nodes** – Mechanism whereby a ZoneRanger user identifies the set of nodes (e.g. network devices, servers) to which ZoneRanger proxy and management services will be applied.
- **Node Groups** – ZoneRanger mechanism for defining named groups of IP address patterns, and using these node groups in a variety of configuration rules particularly forwarding and proxy rules.
- **Pooling/Redundancy/VIP/Grouping** – Mechanisms for providing high availability and/or load-balancing ZoneRanger deployments.
- **Proxy Access Control** – Mechanism whereby the Ranger Gateway identifies the management protocol to be used for a given destination address, transport protocol, and destination port. Also can be used to restrict access to proxy services based on source address, destination address, transport protocol and destination port.
- **Proxy Caching** – Mechanism for reducing management traffic by saving the results of recent ICMP and SNMP proxy requests, and, where appropriate, returning saved values instead of passing the request along to the managed device.
• **Proxy Map** – Mechanism whereby the Ranger Gateway selects an appropriate ZoneRanger to proxy a management protocol transaction, based on configured rules.

• **Server Groups** – ZoneRanger mechanism for defining named groups of TACACS+/RADIUS servers and associated settings.

• **Whitelist** – ZoneRanger mechanism to define a specific list of devices from which only those devices will the ZoneRanger either receive data or send data.

Each of these concepts and mechanisms are described in further detail in the following chapters.
Chapter 2: Address Patterns

An address pattern is an IP address or hostname value that contains wild card characters or range descriptions and thus can be used to describe a range of IP addresses or hostnames. Address patterns are commonly used in Ranger Gateway and ZoneRanger commands and configuration tables, in order to provide a concise mechanism for specifying a set of related addresses.

For example, all of the IP addresses in the 64.1.25.0/24 subnet, can be described using the following address pattern:

62.1.25.*

The * wildcard character can appear in any part of the address. More specific ranges can also be specified, such as:

64.[1-2].25.[1-10]

Wildcard characters can also be used with host names. For example:

* .company.com

or

*.*.company.com

Wildcard and non-wildcard characters cannot be combined within a part of the address. For example, the following address patterns would not be allowed:

abc*.company.com

64.1.25.1[6-8]

In order for a hostname pattern to match a given address, the number of address parts must match. For example, the hostname server1.company.com matches the pattern *.company.com, but the hostname server1.payroll.company.com does not.

IPv6 address patterns are also supported. FC00:1:2::*, FC00:1:[2-9]:AB::[1-F]
Chapter 3: Address Transforms

An address transform is a simple rule that specifies how to take a given IP address or hostname value, and transform it to a new value. Address transforms are used by the Proxy Map service in the Ranger Gateway in order to accommodate static NAT configurations, where the address for a managed device as specified by a management application is different from the actual address of the managed device. Address transforms, in this case, are used to convert the device address as specified by the management application into the real address that the ZoneRanger will use to communicate with the managed device.

For example, the following address transform indicates that the first three parts of the resulting address should be 192.168.1, and the wild card character in the last part of the transform indicates that that the last part of the resulting address should be copied from the original address.

\[192.168.1.*\]

The following table illustrates the addresses that would result from applying a variety of simple address transforms to a number of input addresses:

<table>
<thead>
<tr>
<th>Original Address</th>
<th>Address Transform</th>
<th>Resulting Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.3</td>
<td>192.168.1.*</td>
<td>192.168.1.3</td>
</tr>
<tr>
<td>64.2.37.4</td>
<td>192.168.1.*</td>
<td>192.168.1.4</td>
</tr>
<tr>
<td>64.2.37.55</td>
<td>10.1.<em>.</em></td>
<td>10.1.37.55</td>
</tr>
<tr>
<td>192.168.2.10</td>
<td>10.1.<em>.</em></td>
<td>10.1.2.10</td>
</tr>
</tbody>
</table>

Address transforms can also perform simple computations. For example, the following address transform indicates that the first three parts of the resulting address should be 192.168.1, and the \(*+2\) expression in the last part of the transform indicates that that the last part of the resulting address should be calculated by taking the last part of the original address and adding 2.

\[192.168.1.*+2\]

The following table illustrates the addresses that would result from applying a variety of address transforms to a number of input addresses:

<table>
<thead>
<tr>
<th>Original Address</th>
<th>Address Transform</th>
<th>Resulting Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.3</td>
<td>192.168.1.*+2</td>
<td>192.168.1.5</td>
</tr>
<tr>
<td>64.2.37.4</td>
<td>192.168.1.*+2</td>
<td>192.168.1.6</td>
</tr>
<tr>
<td>64.2.37.25</td>
<td>10.1.<em>-2.</em>+2</td>
<td>10.1.35.27</td>
</tr>
<tr>
<td>192.168.3.200</td>
<td>10.1.<em>-2.</em>+2</td>
<td>10.1.1.202</td>
</tr>
<tr>
<td>192.168.3.254</td>
<td>10.1.<em>-2.</em>+10</td>
<td>10.1.1.8</td>
</tr>
</tbody>
</table>

Note that + and – are the only operations allowed, and all arithmetic is modulo 256.

Address transforms can also be used with host names. For example, the following address transform indicates that the last two parts of the resulting hostname should be dmz1.com, and that the first part of the hostname should be copied from the original hostname.

\[*.dmz1.com\]

The number of parts in the original hostname does not need to exactly match the number of parts in the address transform. For example, the following table illustrates the addresses that would result from applying the address transform \[*.dmz1.company.com\] to a variety of input hostnames:
<table>
<thead>
<tr>
<th>Original Address</th>
<th>Address Transform</th>
<th>Resulting Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>host1.company.com</td>
<td>*.dmz1.company.com</td>
<td>host1.dmz1.company.com</td>
</tr>
<tr>
<td>host2.company.com</td>
<td></td>
<td>host2.dmz1.company.com</td>
</tr>
<tr>
<td>host3.net1.company.com</td>
<td></td>
<td>host3.dmz1.company.com</td>
</tr>
<tr>
<td>host4.net2.branch1.company.com</td>
<td></td>
<td>host4.dmz1.company.com</td>
</tr>
</tbody>
</table>

Note that wildcard characters in address transforms always refer to the corresponding part in the input address or hostname, as counted from left to right.

Wildcard and non-wildcard characters cannot be combined within a part of an address transform. For example, the following address transform is invalid:

```
  dmz*.company.com
```

Examples showing how address transforms can be used when dealing with NAT scenarios, or when managing network zones with overlapping address spaces, are provided in Chapter 3.
Chapter 4: Audit

In order to provide a high level of reliability, the ZoneRanger appliance and the Ranger Gateway software both incorporate a periodic self-audit process. The audit process in each case runs on a five-minute interval, performing a series of pre-configured tests, verifying that services are configured correctly, that components are able to communicate as required, and that the software is operating properly. In the normal case, where no problems are detected, the audit will simply terminate, and reschedule itself to run again five minutes later. If any problems are detected, the audit process will inform the user in a variety of ways:

- The ZoneRanger web interface can be used to view the results of the most recent audit for that ZoneRanger (on the Information section of the ZoneRanger dashboard, and on the View > System Audit page).
- The Ranger Gateway Viewer can be used to view the results of the most recent Ranger Gateway audit, and the results of the most recent audits for any joined ZoneRangers.
- The auditResult command on the Ranger Gateway can be used to view the results of the most recent Ranger Gateway audit, and the results of the most recent audits for any joined ZoneRangers.
- The audit process can be configured to send an SNMP trap whenever a problem is detected.
- The audit process can be configured to send a Syslog message whenever a problem is detected.

When audit results are presented to the user, a description of the problem being reported is provided, along with an assessment of the severity of the problem. Two severity categories are supported:

- **Major**: The problem that has been detected is significant and may affect the ability of the ZoneRanger or Ranger Gateway to provide necessary services. Corrective action should be taken immediately.
- **Minor**: The problem that has been detected may impair the operation of one or more services, but does not necessarily require immediate attention.

In addition to informing the user, the audit process may, where appropriate, automatically perform one or more corrective actions in an attempt to resolve the problem that has been detected. The ability to automatically correct a problem fundamentally depends on the nature of the problem. For example, if a ZoneRanger is unable to communicate with a Ranger Gateway, or if a service has been configured incorrectly, the audit process will notify the user, as described above, but no automatic corrective action will be taken. If an internal software error has been detected, on the other hand, the audit process will perform an escalating series of corrective actions, typically beginning with restarting the affected internal service, and potentially restarting the entire software application, or, in the case of ZoneRanger, rebooting the appliance, if a problem persists across multiple audit cycles.

It should be noted that the Ranger Gateway Viewer, the ZoneRanger web interface, and the listAuditResults command all display the results of the most recent audit. As a result, when a problem detected by the audit process is corrected, that problem will remain in the list of detected problems until the next audit cycle. When the next audit cycle completes, the problem should no longer be listed, provided that the corrective action was successful.
In addition to the primary audit process, as described above, the ZoneRanger appliance also includes a secondary audit, which runs every thirty minutes. The goal of the secondary audit is to verify that the primary audit is doing its job. If the secondary audit determines that the primary audit is no longer functioning properly, the ZoneRanger appliance will be automatically rebooted. This two-layer audit approach helps to ensure that ZoneRanger will continue to operate reliably, even when unexpected software problems occur. Note that the Ranger Gateway does not include a comparable secondary audit, because the Ranger Gateway software is typically installed on the same server as one or more management applications, and users will typically prefer to keep these applications running, even if Ranger Gateway problems are detected.
Chapter 5: Backups/Profiles

ZoneRanger provides two mechanisms for managing sets of configuration information: backups and profiles. Backups and profiles are similar in that in each case, a set of configuration information for a ZoneRanger is gathered and saved in a file that can be restored to the same ZoneRanger, or applied to a different ZoneRanger. The primary difference between backups and profiles is that backups contain the content of the ZoneRanger’s database of discovered devices, as well as any polling/management configuration associated with those devices that is also stored in the database, while profiles do not.

ZoneRanger Profiles

Profiles are intended to be used as a form of configuration template, so that a ZoneRanger configuration can be developed, captured, and easily replicated across a set of ZoneRangers that may be installed in different network regions, and be managing different sets of devices. Backups, on the other hand, can be used to roll back a given ZoneRanger’s configuration to a previous state, to auto-configure a new ZoneRanger that is being added to an existing pool, or to replace an existing ZoneRanger that may have failed or is being redeployed to another location.

Profiles are created (a.k.a. saved) using the ZoneRanger web interface, from the Administration > Profiles page or executing the profile command on the Ranger Gateway. The user has two options with regards to the location(s) where the profile file will be stored:

- The profile can be saved on the ZoneRanger only.
- The profile can be saved on the ZoneRanger, and also copied to a joined Ranger Gateway.

When loading (i.e. applying) a profile, the user can choose to load any profile stored on that ZoneRanger, or stored on any joined Ranger Gateway. Note that a profile can only be loaded on a ZoneRanger that is at the same software version as the ZoneRanger that created the profile. The ZoneRanger web interface can also be used to delete profiles from the ZoneRanger or any joined Ranger Gateway. The profile command on the Ranger Gateway can also be used to restore or delete profiles.

ZoneRanger Backups

Backups can be created using the ZoneRanger web interface, from the Administration > Backup/Restore page, or by executing the Ranger Gateway backup command. When creating a backup from the web interface, the user has two options with regards to the location(s) where the backup file will be stored:

- The backup can be saved on the ZoneRanger only.
- The backup can be saved on the ZoneRanger, and also copied to a joined Ranger Gateway.

When creating a backup using the Ranger Gateway backup command, the backup will always be saved on the ZoneRanger, and copied to the Ranger Gateway where the command was executed. Note that the backup command is primarily intended to allow users to perform automated periodic backups. For example, if the Ranger Gateway is installed on a Solaris or Linux operating system, a cron job can be configured to run this command at scheduled times. In a similar manner, the Scheduled Tasks mechanism can be used to automate backups on a Windows operating systems. Backups are stored in the Ranger Gateway store/zr/backup directory.
Backups can be restored from the ZoneRanger web interface or by executing the Ranger Gateway backup command. When restoring from a backup, the user can choose any backup file stored on that ZoneRanger, or stored on any joined Ranger Gateway. Note that a backup file can only be loaded on a ZoneRanger that is at the same software version, model number (e.g. ZR-50, ZR-200, ZR-500, ZR-SPX, or ZR-MSP) and hardware version as the ZoneRanger that created the backup file. The ZoneRanger web interface can also be used to delete backup files from the ZoneRanger or any joined Ranger Gateway. The `backup` command on the Ranger Gateway can also be used to restore or delete backups.

It should be noted that there is a small amount of ZoneRanger configuration information that is currently not included in a backup and must always be configured manually:

- IP interface configuration (i.e. which interfaces are enabled, and their associated IP addresses and network masks).
- Joining passcode.
- SNMP agent configuration (e.g. sysContact and sysLocation)

**Ranger Gateway Backups**

Backups of the Ranger Gateway configuration can be created and restored using the Ranger Gateway `rgBackup` command. Only backups of the same Ranger Gateway version may be restored. By default, Ranger Gateway backups are located in the `<install_dir>/backups` directory. However, the `[-d directory]` option may be used to create or restore Ranger Gateway backups from an alternate directory. Note, when a Ranger Gateway backup is restored, the Ranger Gateway software will be restarted.
Chapter 6: Destination Groups

A destination group is a named set of rules which can be used in forwarding rules to define where a UDP datagram will be forwarded. Each individual rule in a destination group is comprised of a Ranger Gateway or Data Diode and the ultimate destination of the UDP datagram. Destination groups provide a mechanism to improve the organization of forwarding rules by grouping all rules that are configured with the same Ranger Gateways and final destinations together. This allows for the creation and management of fewer Forwarding Rules.

For example, if a ZoneRanger was joined to three Ranger Gateways (RG1, RG2, RG3) that were used to forward syslog messages to three management applications (appl, app2, app3). If there were also five specific syslog filters configured on the ZoneRanger to process syslog messages and forward those messages to each of the management applications, that would require the creation of 15 forwarding rules.

With destination groups, a single destination group (DG1) could be created with rules for each Ranger Gateway and management application as the destination. Then only 5 forwarding rules would need to be created for the syslog messages.
Chapter 7: Device Groups

A device group is a named set of IP addresses, or address patterns, defined using the `deviceGroup` command on the Ranger Gateway, or using the Ranger Gateway Viewer. Device groups are similar to address patterns in that they provide a concise mechanism for referring to a collection of devices in configuration rules. The advantage of device groups is that they can refer to an arbitrary collection of devices with disjoint addresses, as opposed to address ranges and wild cards which can only refer to contiguous IP address spaces. This is useful when you need to apply configuration settings to devices of a particular type (e.g. routers), where it is unlikely that the addresses of these devices will happen to fall within a contiguous range.

For example, consider the following network:

The managed network in the figure contains two routers (10.1.1.1, and 10.2.1.50), three servers (10.1.1.22, 10.1.1.40, and 10.2.1.18), and one ZoneRanger (10.1.1.100). In order to facilitate different configuration settings for different device types, we could define two device groups:

- MyRouters: 10.1.1.1, 10.2.1.50
- MyServers: 10.1.1.22, 10.1.1.40, 10.2.1.18

Device groups can be used to define rules associated with Proxy Access Control (see Chapter 15) and Proxy Map (see Chapter 17) services on the Ranger Gateway. For example, one of the tables used in Proxy Access Control consists of rules of the following form:

`<src-address> <dest-address> <port-configuration>`

The `<src-address>` and `<dest-address>` values in these rules can be specified in the form of specific addresses, address patterns, or device groups. Using the MyRouters and MyServers device groups defined above, and assuming we have three defined port configurations, unimaginatively named `portConfig-1`, `portConfig-2`, and `portConfig-3`, we could configure Proxy Access Control using the following rules:

- `*.*.*.* @MyRouters portConfig-1`
- `*.*.*.* @MyServers portConfig-2`
- `*.*.*.* 10.1.1.100 portConfig-3`
In addition to user-defined device groups, there are two special device groups that are defined automatically:

- The *Local* device group consists of all addresses that are local to the Ranger Gateway server.
- The *ZoneRanger* device group consists of the IP addresses of all joined ZoneRangers.

Using these device groups, in the network shown above, if we wanted to only allow proxy requests originating from the Ranger Gateway server, the Proxy Access Control rules above could be rewritten as:

- @Local @MyRouters portConfig-1
- @Local @MyServers portConfig-2
- @Local @ZoneRanger portConfig-3

Device groups can also contain device groups. For example, we could define a new group called *MyRoutersAndServers* as follows:

```
MyRoutersAndServers: @MyRouters, @MyServers
```

Once a device group is defined, and configuration settings have been associated with it, those settings can be applied to a new device by simply adding that device to the group. Similarly, a device that is included in a configured device group can be reverted back to default settings by simply removing that device from the group.

---

2 As a convenience, the ZoneRanger device group also includes any IP addresses that map to a joined ZoneRanger based on the Proxy Map configuration.
Chapter 8: GVI/RGVI

In order for the ZoneRanger to proxy management traffic to managed devices, the management traffic generated by a management application must first be routed to the Ranger Gateway. This can be accomplished in a variety of ways:

1. Configure the management application to direct management traffic associated with devices located in firewall-partitioned networks to the Ranger Gateway.
2. Provide a mechanism whereby management traffic destined for devices located in firewall-partitioned networks is intercepted by the Ranger Gateway.

The primary advantage of the latter approach is that the management application can send requests to devices located in firewall-partitioned networks as if it was communicating with those devices directly. The fact that the requests are intercepted and processed by the Ranger Gateway is effectively hidden from the management application. This ability to hide the existence of the proxy from the management application is referred to as transparent proxy. The preferred mechanisms for supporting transparent proxy on the Ranger Gateway are the Gateway Virtual Interface (GVI) and the Remote Gateway Virtual Interface (RGVI). These mechanisms are described in the following sections.

GVI

The GVI mechanism is used when the management application and Ranger Gateway software are installed on the same server. The internal architecture and operation of the GVI mechanism is illustrated in the following figure.

![Figure 8-1. GVI Architecture](image-url)
When the GVI service is enabled, the Ranger Gateway creates a virtual point-to-point interface on the management application server, and adds one or more static routes to the management application server so that traffic destined for devices located in firewall-partitioned networks is routed to this virtual interface (1). The GVI service in the Ranger Gateway, as the creator/owner of the virtual interface, receives all traffic that is routed to the virtual interface (2). The GVI service consults with the Proxy Access Control service in the Ranger Gateway to determine if the traffic should be allowed, and to identify the protocol-specific proxy service (e.g. SNMP proxy, TCP proxy) that should handle the traffic. If the request is allowed, the GVI service forwards the traffic to the selected proxy service (3). The proxy service consults the Proxy Map service in the Ranger Gateway in order to identify identify a ZoneRanger that is able to relay the traffic to the target device, and to translate the target address, if necessary, and then forwards the traffic to the selected ZoneRanger (4)(5), which in turn, forwards the traffic to the target DMZ device (6). Where applicable, proxy services may also perform validation and filtering of the management traffic, as appropriate for the service being used.

The GVI service also includes a route manager that simplifies creation and management of the static routes that are needed so that management traffic is routed to the virtual interface. The route manager can be configured with a set of subnets or individual IP addresses that should be routed to the virtual interface, and will automatically create the associated static routes when the GVI service is enabled, and will delete these routes when the GVI service is disabled. If the GVI service is enabled and the Ranger Gateway software is stopped, the route manager will automatically remove any static routes associated with the virtual interface, and will reconfigure these routes when the Ranger Gateway software is restarted. As a result, there should be no need to redefine static routes if the management application server is rebooted, because the virtual interface static routes will be reconfigured when the Ranger Gateway software is started.

The virtual interface created by the GVI service emulates a point-to-point interface. As such, a local IP address and a remote IP address must be associated with this interface. By default, the GVI service configures the virtual interface with the following addresses:

- Local: 192.168.48.1
- Remote: 192.168.48.2

Alternative addresses can be configured if these addresses create a conflict. Please contact Tavve Support for more information if you need to change these addresses.

In order to route a subnet corresponding to a set of managed devices to the virtual interface, the route manager creates a static gateway route to the virtual interface’s remote address. For example, in order to route the 10.1.10.0/255.255.255.0 subnet to the virtual interface, the following route would be defined:

```
10.1.10.0 255.255.255.0 192.168.48.2
```

Before creating a static route for a given subnet, the route manager checks to see if any of the IP addresses being used to communicate with joined ZoneRangers lie within the subnet being added. In order to ensure that communication with joined ZoneRangers can continue, the route manager automatically creates host routes for any such addresses. These host routes override the subnet routes in the system routing table for the given IP addresses, and effectively ensure that traffic destined for joined ZoneRangers is routed to the gateway that would have been used in the absence of any virtual interface routes.
For example, consider the network in the following figure:

![Network Diagram]

Figure 8-2. GVI Example

In the example, a Ranger Gateway is being used to manage two firewall-partitioned networks. Two ZoneRangers have been deployed into each of these networks. In order to intercept management traffic destined for these networks, two GVI subnet routes have been configured:

- 10.1.1.0/24
- 10.1.2.0/24

When the GVI service is enabled, the GVI route manager will automatically create static routes indicating that traffic destined for these subnets should be routed to the GVI interface's remote address (192.168.48.2). Given that the IP addresses of each of the ZoneRangers lie within these subnets, the GVI route manager will automatically create host routes for each of the joined ZoneRangers indicating that traffic destined for the ZoneRanger addresses should be routed via the original default gateway that was configured for the Ranger Gateway server (64.1.2.1). The figure also shows the original default routing rule (0.0.0.0/0 → 64.1.2.1), and a simple Proxy Map configuration indicating which ZoneRangers should be used to proxy traffic for which device addresses.

If the GVI service is enabled, and a request to join to a given ZoneRanger is received by the Ranger Gateway, the GVI route manager will automatically create a host route for that IP address, where necessary to ensure that traffic destined for the ZoneRanger will bypass the virtual interface. Host routes for joined ZoneRangers will automatically be removed from the system routing table if the ZoneRanger is unjoined, any overlapping virtual interface routes are removed, or the GVI service is disabled.

The GVI service can be controlled and configured using the Ranger Gateway Viewer or the gvi Ranger Gateway command.
On Windows servers, once the GVI is enabled, run the Windows command `ncpa.cpl` to display the current list of network interfaces. Using the **Advanced > Advanced Settings...** menu item, verify that the GVI virtual interface is last in the access order for network services.

**RGVI**

The RGVI mechanism is used when the management application and Ranger Gateway software are installed on different servers, as illustrated in the following figure.

![RGVI Architecture](image)

*Figure 8-3. RGVI Architecture*

The RGVI mechanism makes use of a thin *RGVI client*, installed on the management application server. The RGVI client intercepts traffic destined for managed devices, and relays this traffic to the Ranger Gateway software using a UDP-based communication protocol.

The primary advantages of RGVI are as follows:

- The processor/memory footprint of the RGVI client is considerably less than the Ranger Gateway software.
- The RGVI client can be installed on some operating systems for which the Ranger Gateway software is not currently supported.

The internal architecture and operation of the RGVI mechanism is illustrated in the following figure.
The RGVI Client is configured with the addresses of one or more Ranger Gateway servers to which it can connect in order to provide proxy services. When the RGVI client is initialized on the Management Application Server, it creates a virtual point-to-point interface which is used to intercept traffic destined for managed devices, similar to the approach used for GVI, then attempts to connect to one of the configured Ranger Gateway servers. When the connection to a Ranger Gateway server is established, the Ranger Gateway will send the RGVI client a list of individual IP addresses and subnets that the RGVI client should intercept. On receipt of this list, the RGVI client will configure corresponding static routes on the management application server so that traffic destined for devices located in firewall-partitioned networks is routed to its associated virtual interface (1). The RGVI client, as the creator/owner of the virtual interface, receives all traffic that is routed to the virtual interface (2), then relays this traffic to the Ranger Gateway server to which it has connected (3)(4). Within the Ranger Gateway, this traffic is received by the RGVI service which consults with the Proxy Access Control service in the Ranger Gateway to determine if the traffic should be allowed, and to identify the protocol-specific proxy service (e.g. SNMP proxy, TCP proxy) that should handle the traffic. If the request is allowed, the RGVI service forwards the traffic to the selected proxy service (5). The proxy service consults the Proxy Map service in the Ranger Gateway in order to identify a ZoneRanger that is able to relay the traffic to the target device, and to translate the target address, if necessary, and then forwards the traffic to the selected ZoneRanger (6), which in turn, forwards the traffic to the target DMZ device (7). As in the case of GVI, where applicable, proxy services may also perform validation and filtering of the management traffic, as appropriate for the service being used.

The RGVI mechanism allows a single Ranger Gateway server to support multiple management application servers, as illustrated in the following figure.
The RGVI service in the Ranger Gateway is configured with a list of permitted clients, and each client can be configured with its own list of host and/or subnet addresses to be intercepted, or can inherit the list of host and/or subnet addresses configured for the GVI service. When a client connects to the RGVI service within the Ranger Gateway, the service checks its configuration to verify that the client is permitted to use the RGVI service, and if so, pushes the corresponding set of host and/or subnet addresses to the client, which then configures the corresponding routes on the management application server. Permitted clients can also be specified using address patterns or device groups, allowing multiple client addresses to share the same list of host and/or subnets to be intercepted. Note that if the list of hosts and/or subnets to be intercepted for RGVI client entry is modified while one or more of the set of the matching clients is running and connected to the Ranger Gateway, these clients will need to be restarted in order for the modifications to take effect.

In addition to checking the client's IP address to ensure that a client is permitted to access the RGVI service, the Ranger Gateway also authenticates each client using SSL certificates. The list of trusted RGVI subjects and corresponding certificate authorities can be configured using the `trustSSL` Ranger Gateway command. Note that Ranger Gateway to ZoneRanger messaging and RGVI share a common list of trusted certificate authorities, but have distinct trusted subject lists.

Although SSL-based authentication is always provided for RGVI client-to-server connections, the RGVI service can optionally be configured to provide encryption and integrity checking for any data being transferred between the client and the Ranger Gateway. The option to provide encryption and integrity checking for data traffic can be enabled or disabled using the Ranger Gateway Viewer or the `rgvi` Ranger Gateway command. Note that if the Ranger Gateway is configured to provide encryption and integrity checking, all RGVI clients that can connect to that Ranger Gateway must also be configured to provide encryption and integrity checking. Similarly, if encryption and integrity checking is disabled on the Ranger Gateway, it must also be disabled on all RGVI clients.
By default, the RGVI client communicates with the Ranger Gateway using a custom UDP-based protocol. The port on which the Ranger Gateway listens for incoming connections from RGVI clients is configurable (default is 1194). Note that if the RGVI port on the Ranger Gateway is modified, it will also be necessary to modify the configuration for each RGVI client to use the same port.

In order to provide high availability, each RGVI client can be configured with the IP addresses of multiple Ranger Gateways. When the client is started, it will attempt to connect to one of the Ranger Gateways in the list. If the connection attempt succeeds, the RGVI client will direct all proxy traffic to the connected Ranger Gateway. If the connection attempt fails, the RGVI client will attempt to connect to a different Ranger Gateway. If the RGVI client has successfully connected to a given Ranger Gateway, but subsequently loses connectivity, the client will time out and attempt to connect to a different Ranger Gateway.

The RGVI service can be controlled and configured using the Ranger Gateway Viewer or the `rgvi` Ranger Gateway command.

RGVI client installation and configuration instructions for various operating systems are provided in Appendix J.
Chapter 9: IPv6

When a ZoneRanger has a ZR-SPX license, it may be configured in an IPv6 network on the Configuration → System page IP tab. IPv6 routes can be added on the Administration > IP Route page. Generally, IPv6 addresses must be specified in CIDR format if a prefix length is needed.

The following concepts support IPv6 addresses.

- Address Patterns
- Address Transforms
- GVI/RGVI
- Proxy Caching
- Proxy Map

The following services support IPv6 addresses.

- ICMP Proxy
- SNMP Proxy

Diagnostic tools relating to ICMP ans SNMP are generally supported for IPv6 addresses.

To enable the Gateway Gateway to process IPv6 addresses, use the `portMap` command:

```
portMap add ::/0 ::/0 Default
```

To enable IPv6 for the GVI interface, use the `gvi` command.

```
gvi config ipv6_enable true
```

To enable IPv6 for the RGVI interface, add the line `tun-ipv6` to the openVPN client configuration file.
Chapter 10: Joining

In the ZoneRanger architecture, the ability to proxy traffic between a management application and the devices in a firewall-partitioned network requires two components:

- One or more ZoneRanger appliances, located in the same network partition as the managed devices.
- An installed instance of the Ranger Gateway software component, which is typically installed on the same server as one or more management applications.

The role of the Ranger Gateway is to act as an interface between the ZoneRangers and management applications, relaying proxy traffic to/from ZoneRangers that are able to communicate directly with the managed devices.

For security purposes, before a Ranger Gateway can relay management traffic to/from a given ZoneRanger, it must first be joined to that ZoneRanger. Joining is a simple process, initiated by the user, and can be performed using any of the following user interface mechanisms:

- Ranger Gateway Viewer
- Ranger Gateway command
- ZoneRanger web interface

Joining essentially establishes a persistent relationship between a Ranger Gateway and a ZoneRanger, so that the joined entities can cooperate in the provision of proxy services. Security for the joining process is implemented in two layers:

1. The Ranger Gateway and ZoneRanger must authenticate each other using SSL certificates.
2. The Ranger Gateway and ZoneRanger must be configured with matching passcodes.

SSL authentication, when properly configured, can provide a high level of security. In order for a Ranger Gateway and a ZoneRanger to establish an SSL connection, the following conditions must be satisfied:

1. The Ranger Gateway must present an SSL certificate that has been signed by a certificate authority recognized by the ZoneRanger, and must have a distinguished name that matches one of the entries in the ZoneRanger’s configured list of trusted subjects.
2. The ZoneRanger must present an SSL certificate that has been signed by a certificate authority recognized by the Ranger Gateway, and must have a distinguished name that matches one of the entries in the Ranger Gateway’s configured list of trusted subjects.

By comparison, passcode authentication offers a more casual level of security, and is intended more to prevent unintentional joining between a Ranger Gateway and a ZoneRanger.

---

3 The Ranger Gateway software can also be installed on a separate server, located in the same subnet as the management application server.

4 When joining to a ZoneRanger from the Ranger Gateway, it is also possible to specify the passcode of the ZoneRanger as part of the request, even if that passcode does not match the Ranger Gateway’s configured default value.
A single Ranger Gateway can be joined with up to 250 ZoneRangers. Given that each Ranger Gateway instance typically serves a single management application instance, this allows a given management application to extend its reach into many firewall-partitioned networks, providing that one or more ZoneRangers have been installed in each such network. Similarly, each ZoneRanger can be joined with up to 250 Ranger Gateway instances, allowing managed devices in the firewall-partitioned network where the ZoneRanger has been installed to interact by proxy with a large number of management applications. The many-to-many joining relationship between Ranger Gateways and ZoneRangers is illustrated in the following figure.

Figure 10-1. Many to Many ZoneRanger configuration

The figure shows four Ranger Gateway instances, each joined with six ZoneRangers. Each of the ZoneRangers is joined with all four Ranger Gateways. The number of joining relationships that need to be established, then, depends on the number of management applications being used, and the number of firewall-partitioned networks to be managed.

If there is a firewall between a Ranger Gateway and a ZoneRanger that need to be joined, as is typically the case, a firewall rule must be configured, to allow the Ranger Gateway and ZoneRanger to communicate. All management protocol traffic being proxied between a Ranger Gateway and a ZoneRanger is multiplexed over a single TCP connection, so a single TCP firewall rule, specifying the configured messaging port\(^5\) as the destination port is all that is required.

Note that the figure also shows two management applications: CiscoSecure ACS and a Trap/Syslog Receiver that make use of a Ranger Gateway installed on a different server. This approach can be used to reduce the number of Ranger Gateway instances, and associated firewall rules and joining relationships, in cases where the nature and volume of the protocol traffic allows.

\(^5\) The default messaging port is TCP 4854.
Chapter 11: Licensing

ZoneRanger may be a physical appliance or a virtual appliance (VM). When in the form of a physical appliance, the ZoneRanger is manufactured with a particular license specifying the number of devices it is allowed to manage. This license is a permanent license with no expiration date. When in the form of a virtual appliance (VM), the ZoneRanger must obtain its license via a Ranger Gateway License Server or via a provided Activation Key. This license will also specify the number of devices the ZoneRanger is allowed to manage but it will have an expiration date and will need to be renewed periodically.

Ranger Gateway License Server

A Ranger Gateway, when configured with licenses provided by Tavve, may be a license server for any ZoneRanger VMs joined to it. Once a ZoneRanger VM has joined to a Ranger Gateway License Server, the Administration > License Activation page on the ZoneRanger VM web interface may be used to specify which of the available licenses on the Ranger Gateway License Server should be allocated to this ZoneRanger VM. Once the license is allocated to the ZoneRanger VM, the ZoneRanger VM is now activated and must maintain communication with the Ranger Gateway License Server in order to maintain its activation. If communications between the ZoneRanger VM and the Ranger Gateway License Server is lost, the ZoneRanger VM license will eventually be deactivated.

The Administration > License Activation page may also be used to choose a different license to be activated on the ZoneRanger VM from any available licenses on the Ranger Gateway License Server. Whenever any license is activated on the ZoneRanger VM, the ZoneRanger software may restart.

If a new ZoneRanger VM license is activated that has fewer nodes than are currently managed by the presently activated license, all managed nodes will be automatically unmanaged. For example, if a ZoneRanger VM has an activated ZR-200 license with 150 managed nodes and the ZoneRanger VM is then activated with a ZR-50 license, all of the 150 managed nodes will be automatically unmanaged once the new license activation is complete.

If the expiration time is reached on a currently activated ZoneRanger VM, the ZoneRanger VM will automatically be deactivated and restarted.

Activation Key

A ZoneRanger VM may be activated by using an Activation Key provided by Tavve. On the Administration > License Activation page on the ZoneRanger VM web interface, an Activation Key may be entered in order to activate the license on the ZoneRanger VM. The Pending Token as displayed on the Administration > License Activation page must be provided to Tavve Software in order for an Activation Key to be generated. Once the Activation Key is entered, the ZoneRanger VM will be activated after a software restart.
Chapter 12: Managed Nodes

For all node-based ZoneRanger models (e.g. ZR-50, ZR-200, and ZR-500), the majority of the proxy and autonomous management services are available only to the set of nodes (a.k.a. devices) that have been designated as managed. In each case, the model number indicates the maximum number of nodes that can be managed by a given ZoneRanger.

In order for a node to be managed, it must first be discovered. Discovery, in the context of ZoneRanger, is the process whereby a ZoneRanger analyzes its surrounding network and populates its database with the nodes, interfaces, subnets, and TCP ports that are encountered. Depending on the discovery options settings the discovery process can range from minimal (i.e. simply analyze a list of seed node addresses in an attempt to identify devices that support multiple IP addresses) to fairly aggressive (e.g. ping ranges, broadcast pings, ARP cache queries, and routing table queries). The ability to identify devices (a.k.a. nodes) with multiple IP interfaces is particularly important, because it is the number of managed nodes that is limited based on the license, not the number of interfaces. For example, a router may have several interfaces, each with its own IP address. If the ZoneRanger discovery process is able to determine that these IP addresses are associated with a single device, that device will count as a single node towards the license limit.

Nodes can automatically be designated as managed during the discovery process, or can be managed or unmanaged manually using the ZoneRanger web interface. If the Auto manage newly discovered nodes discovery option is enabled, newly discovered nodes are automatically designated as managed when discovery runs, until the license limit is reached. After the license limit is reached, any additional newly discovered nodes are designated as unmanaged. After discovery runs, you can unmanage one or more nodes, and manage others, using the ZoneRanger web interface.

If the Auto manage newly discovered nodes discovery option is not enabled, newly discovered nodes are designated as unmanaged, regardless of the license limit. After discovery runs, you can use the ZoneRanger web interface to manage one or more of the discovered nodes.

The ZR-SPX model has no license limit with respect to proxy and forwarding services. That is, a ZR-SPX will proxy traffic to/from any device, regardless of whether or not that device has been designated as managed or even exists in the database. Note that autonomous management features (e.g. ZoneRanger polling, root cause) remain limited to managed nodes, even for the ZR-SPX model, and that the ZR-SPX has a limit of 50 managed nodes.

The ZR-SPX model is engineered based on traffic capacity, rather than node limits. As such, customers are encouraged to consult with Tavve Sales in order to determine the number of ZR-SPX units that would be required for any given application/deployment.
Chapter 13: Node Groups

There are many configuration areas in ZoneRanger which accept a list of IP addresses and/or address patterns. The same list of devices may need to be applied to multiple configuration areas. **Node Groups** represent a collection of address patterns that can be applied to the following IP address based configurations:

- Forwarding rules
- TACACS+/RADIUS proxy rules
- Inbound TCP proxy rules
- Outbound ICMP proxy caching
- IP Interface polling
- SNMP Manager rules
- SNMP Disallowed lists.

**Node Groups** are maintained on the **Configuration > Node Management** page **Node Groups** tab of the ZoneRanger Web GUI. **Node Groups** may contain any number of valid address patterns as well as other **Node Groups**. When specifying a **Node Group** in a configuration rule or within another **Node Group**, the name of the **Node Group** must be prefixed with '@' to indicate that the value should be interpreted as a **Node Group**. For example, the **Node Group** “webservers” would be represented as @webservers. It may represent the following:

```
10.254.1.[1-5]
10.10.23.25
@otherwebservers
```

Note that **Node Groups** do not contain hostnames.
Chapter 14: Pooling/Redundancy/VIP/Grouping

ZoneRanger deployments typically place two or more ZoneRangers in each firewall-partitioned network, for one or both of the following reasons:

- High availability (i.e. if one ZoneRanger fails, the other ZoneRanger(s) can handle the required management protocol proxy traffic).
- High capacity (i.e. by deploying multiple ZoneRangers in a load balancing configuration, the total volume of management protocol traffic that can be proxied to/from the firewall-partitioned network is increased).

High availability and high capacity are supported by a variety of Ranger Gateway and ZoneRanger mechanisms as described in the following sections.

Pooling

The simplest configuration for multiple ZoneRangers in the same network partition is referred to as pooling. A pool of ZoneRangers is essentially a set of multiple ZoneRangers, each unaware of the others, that are deployed in the same network partition and are joined to the same Ranger Gateway(s). Each Ranger Gateway that uses the pool is configured with an understanding that each of the ZoneRangers in the pool is equally capable of relaying management protocol traffic to a given set of devices (i.e. the devices in the network partition where the ZoneRanger pool is deployed). The Ranger Gateway can be configured to distribute management protocol proxy transactions across the pool in a load-balancing fashion, in order to achieve high capacity. In addition, if the Ranger Gateway detects that it is unable to communicate with one of the members of the pool, traffic will be distributed to the remaining members, in order to achieve high availability.

Pooling is implemented within the Ranger Gateway’s Proxy Map service, and affects management protocol traffic originated by the management application, such as ICMP proxy or SNMP Get/Set proxy. The following figure illustrates a simple pool of four ZoneRangers. Note that the Proxy Map configuration indicates that each of the ZoneRangers is equally capable of relaying management protocol traffic to any of the managed devices in the DMZ where they have been deployed.
Redundancy

Redundancy is a more sophisticated form of multiple ZoneRanger configuration. When using redundancy, a persistent relationship is configured between the redundant ZoneRangers, so that they become aware of each other. When two or more ZoneRangers are configured to be redundant, configuration changes to one of the ZoneRangers, including changes to configuration options associated with managed devices, will automatically be propagated to the other ZoneRangers, making it easier to keep the configurations of the redundant ZoneRangers in sync. Whenever a redundant ZoneRanger that has been down or was otherwise unreachable is restored, and communication with its peers is reestablished, the ZoneRanger will resynchronize its configuration with its peers in order to pick up any configuration changes that may have occurred.

Redundant ZoneRangers perform discovery and polling independently. As such, discovery results and device status information are not propagated between redundant ZoneRangers. If polling configuration settings associated with discovered devices are modified on one ZoneRanger, the changes are propagated to all redundant ZoneRangers, each of which will attempt to locate the corresponding device in their own database of discovered devices, and make identical changes.

ZoneRangers can be configured to be redundant during initial configuration, or at a later time, using the ZoneRanger web interface. Note that redundancy and pooling can be used simultaneously. That is, the Proxy Map service on the Ranger Gateway can be configured to distribute management protocol traffic across the set of redundant ZoneRangers, in the same way as across a simple pool of ZoneRangers. In that sense, Redundancy provides all of the features of pooling, plus the added value of automatically synchronizing the configurations of the redundant ZoneRangers. A redundant pool of ZoneRangers is illustrated in the following figure.

![Figure 14-2. Redundant ZoneRanger Pooling Example](image-url)
Virtual IP

Redundant ZoneRangers may be configured with a Virtual IP, forming a VIP Cluster. Within a VIP Cluster, a virtual IP address is defined and shared across a set of redundant ZoneRangers. At any given time, one of the ZoneRangers will be active with respect to the virtual IP address (i.e. it is configured to receive traffic destined for the virtual address), while the others will be passive. If the passive ZoneRangers detect that the active ZoneRanger has failed, or is no longer communicating, one of the passive ZoneRangers will become active, so that there should always be a healthy ZoneRanger able to receive and process traffic directed towards the virtual address.

The advantage of creating a VIP Cluster is that managed devices can be configured to forward protocol traffic such as SNMP traps, Syslog messages, NetFlow messages or sFlow messages to a single address (i.e. the virtual IP address), and whatever ZoneRanger happens to be active on that address at that time will relay the traffic to the intended management application(s) via one or more Ranger Gateways.

Note that in addition to the virtual IP address, each ZoneRanger also will have its own unique real address. As such, an alternative to using virtual IP is to configure managed devices to forward management protocol traffic to multiple ZoneRangers, specifying the real IP address in each case. In the case of SNMP traps, and Syslog messages, each Ranger Gateway will remove the duplicate traffic, so that each actual trap or message should only be forwarded once to the listening management applications. This approach minimizes the possibility of lost traps or messages, at the cost of increased management traffic (e.g. the managed device must forward each SNMP trap or Syslog message multiple times). Creating a VIP Cluster will eliminate this additional traffic, at the cost of the possibility of lost traffic during the brief period of time that it will take for a passive ZoneRanger to detect that the active ZoneRanger has failed, and to become active.

![Figure 14-3. Redundant ZoneRanger within a VIP Cluster](image)

Note that in this figure, ZR-2 is currently active, with respect to the VIP address (10.1.1.60), while the other three ZoneRangers are currently passive. Managed devices, in this case would typically be configured to forward traffic such as SNMP traps and Syslog messages to the 10.1.1.60 address.
**Grouping**

*Grouping* is another concept that applies when multiple ZoneRangers are deployed into the same network partition. A set of ZoneRangers is considered to be grouped when they are all configured with the same group name\(^6\). Grouping impacts the behavior of the Ranger Gateway with respect to two proxy features:

- The Ranger Gateway’s algorithm for de-duplicating forwarded SNMP traps and Syslog messages depends on *grouping*, in that if a Ranger Gateway receives duplicate SNMP traps or Syslog messages forwarded from multiple ZoneRangers, the duplicates will be removed only if the configured group names of the forwarding ZoneRangers are identical.

- When using one of the three-part community string formats (i.e. `community@ZoneRanger@device` or `device@ZoneRanger@community`) in conjunction with SNMP Get/Set proxy, if the group name is used in place of the ZoneRanger in the community string, the Ranger Gateway will select one of the joined ZoneRangers in the group to relay the request to the managed device.

Note that it is possible to configure a set of ZoneRangers to have the same group name with or without configuring redundancy. For example, if a pool of ZoneRangers is configured without configuring redundancy, the group names of each of the ZoneRangers in the pool still need to match in order for SNMP trap and Syslog message de-duplication to work properly. If redundancy is configured, the redundant ZoneRangers will automatically have the same group name (i.e. the group name from the source ZoneRanger will be propagated to any target ZoneRangers).

---

\(^6\) The group name of a ZoneRanger can be configured during initial configuration, or at any subsequent time via the ZoneRanger web interface.
Chapter 15: Proxy Access Control

Proxy Access Control on the Ranger Gateway governs the handling of management traffic originated by management applications and destined for managed devices (e.g. ICMP request, SNMP Get/Set request, HTTPS, SSH, FTP), enabling users to configure what clients are allowed to use what protocols for given managed devices. Traffic originated by managed devices is typically governed by configuration rules within the ZoneRanger (e.g. forwarding rules, TACACS+/RADIUS server groups), and is outside of the scope of Proxy Access Control.

Whenever a proxy request is received from a management application, the Ranger Gateway uses Proxy Access Control configuration rules to determine:

- Whether the proxy request should be allowed or discarded.
- If the request is allowed, the protocol being used (e.g. for validation, or special processing).
- If the request is allowed, the port translation rule, if any, that should be applied before presenting the request to a managed device.

Proxy Access Control is organized into two stages, based on two configuration tables, the portMap table and the portConfig table:

- The portMap table consists of an ordered set of rules of the following form:
  
  \[(src-address, dest-address, port-config-name)\]

  where \(src-address\) is the IP address associated with the requesting client, \(dest-address\) is the IP address associated with the target managed device, and \(port-config-name\) is the name of the port configuration to be used in the second stage.

  In the first stage, the Ranger Gateway takes the \(src-address\) and \(dest-address\) for a given request, and searches the portMap table for the first matching rule. If no matching rule is found, the request is discarded.

- The portConfig table consists of a set of rules of the following form:
  
  \[(port-config-name, transport, rg-port, protocol, zr-port)\]

  where \(port-config-name\) is the name of a port configuration (as identified in the previous stage), \(transport\) indicates whether the request is using ICMP, UDP, or TCP, \(rg-port\) is the destination port associated with the request as received by the Ranger Gateway, \(protocol\) identifies the management protocol to be used for the request, and \(zr-port\) either specifies the destination port that the ZoneRanger should use when forwarding the request to the target device, or a translation rule that can be used to calculate the port that should be used based on the \(rg-port\).

  In the second stage, the Ranger Gateway takes the \(port-config-name\) that was identified in the first stage, the \(transport\) associated with the request, and, where applicable, the destination port associated with the request (a.k.a. \(rg-port\)) and searches the portConfig table for the first matching rule. If no matching rule is found, the request is discarded. Note that if the transport is ICMP, the \(rg-port\), \(protocol\), and \(zr-port\) fields are not used.

To illustrate this process, consider the network and configuration tables in the following figure:
This figure shows two management application servers (10.10.1.2 and 10.10.1.3), only one of which (10.10.1.3) contains a Ranger Gateway. The 10.10.1.2 server is assumed to be using the 10.10.1.3 server to relay management traffic. According to the portMap table, if an application running on server 10.10.1.2 initiates a request destined for device 10.1.1.22, the port configuration named portConfig-1 will be selected. If the same application initiates a request for any other managed device, no matching port configuration will be found, and the request will be discarded. According to the portConfig table, the portConfig-1 port configuration allows ICMP proxy and SNMP proxy on UDP port 161. Requests from 10.10.1.2 to 10.1.1.22 involving other transport protocols and/or ports will be discarded.

Referring back to the portMap table, if an application running on server 10.10.1.3 initiates a request to any of the managed devices, the port configuration named portConfig-2 will be used, because the *.*.*.* destination address pattern will match all destination addresses. The portConfig table shows that portConfig-2 allows ICMP proxy, SNMP proxy on UDP port 161, SSH proxy on TCP port 22, and HTTPS proxy on port 443 (which will be translated to port 8443 before presenting the request to the target device). Requests from 10.10.1.3 involving other transport protocols and/or ports will be discarded.

Although standard well-known port values have been used in this example for each management protocol, it is also possible to allow supported protocols to be used on non-standard ports (e.g. to confuse, or hide from port scanners, for improved security).

---

A management application server with no Ranger Gateway installed can proxy traffic through a Ranger Gateway installed within another server in a variety of ways, including SOCKS, joined ZoneRanger proxy ports (i.e. 200xx), or by enabling IP forwarding on the Ranger Gateway server, and configuring the other server to route management traffic to the Ranger Gateway server.
The intent of the two-stage approach is to allow a small number of port configurations to be defined, and re-used across multiple devices. As an example, it may be useful to define custom port configurations for specific device types (e.g. Windows server, Cisco router), and configure the portMap table so that the appropriate port configuration is used in each case. When this approach is used, the port configuration details only need to be specified once per device type.

The portMap table also supports the use of address patterns (see Chapter 2) and device groups (see Chapter 7) in the src-address and dest-address fields, in order to allow a single rule to be applied to multiple devices.

The default portMap configuration, for a new ZoneRanger, is as follows:

```
*.*.*.* @ZoneRanger ZoneRangerDefault
*.*.*.* *.*.*.* Default
```

This configuration indicates that requests from any source directed towards a joined ZoneRanger will be governed by the ZoneRangerDefault configuration, and that all other requests will be governed by the Default configuration. The ZoneRangerDefault rule is configured first, so that requests directed towards ZoneRangers will match that rule, as opposed to the Default rule. In order to restrict the Ranger Gateway so that only traffic originated by applications on the Ranger Gateway server itself will be processed, the portMap table would need to be configured as follows:

```
@Local @ZoneRanger ZoneRangerDefault
@Local *.*.*.* Default
```

**Note:** When restricting the Ranger Gateway to only accept local traffic it is highly recommended that the @Local device group be specified, as opposed to specifying individual local IP addresses. The source address for traffic received via GVI or RGVI will typically be a special address associated with the underlying virtual point-to-point interface, so configuring rules based on specific IP addresses may not produce the expected results.

The Default port configuration for a new ZoneRanger contains the following rules:

```
Default TCP 22 SSH
Default TCP 443 HTTPS
Default UDP 161 SNMP
Default ICMP
```

This configuration allows the use of SSH on TCP port 22, HTTPS on TCP port 443, SNMP on UDP port 161, and ICMP. The ZoneRangerDefault port configuration contains the following rules:

```
ZoneRangerDefault TCP 22 SSH
ZoneRangerDefault TCP 23 TELNET
ZoneRangerDefault TCP 80 HTTP
ZoneRangerDefault TCP 443 HTTPS
ZoneRangerDefault TCP 5432 SQL
ZoneRangerDefault UDP 161 SNMP
ZoneRangerDefault ICMP
```
This configuration includes all of the rules from the Default configuration, plus the ability to use Telnet on TCP port 23, HTTP on TCP port 80, and SQL on port 5432. Note that these rules only govern access to the ZoneRanger via proxy through the Ranger Gateway. Direct access to ZoneRanger ports can be enabled and/or disabled from the Configure > System page Ports tab on the ZoneRanger web interface or using the Ranger Gateway portControl command.

The rg-port value in a port configuration rule can indicate a specific port, as described earlier, but can also indicate a contiguous range of ports. For example, adding the following rule would enable SSH to be used on TCP ports in the range 300-310:

**Default TCP 300-310 SSH**

Port transformations can also be used with port ranges. For example, the following rule would enable SSH to be used on TCP destination ports in the range 300-310, but would transform these ports to the 8300-8310 range before forwarding protocol messages to the target devices:

**Default TCP 300-310 SSH *+8000**

Similarly, the following rule would enable SSH to be used on TCP destination ports in the range 300-310, but would transform these ports to the 250-260 range before forwarding protocol messages to the target devices:

**Default TCP 300-310 SSH *-50**

There may be times when it is useful to proxy a TCP-based management protocol that is not explicitly supported by ZoneRanger (e.g. a proprietary or vendor-specific protocol). This form of proxy can be enabled by adding a port configuration rule, specifying TCP as the protocol. For example:

**Default TCP 300-310 TCP**

Note that where TCP is specified as the protocol, the ZoneRanger does not provide any application protocol inspection/filtering. However, given that ZoneRanger provides a break in the TCP protocol (i.e. there are two TCP connections for each proxy session: one between the management application and the Ranger gateway, and one between the ZoneRanger and the managed device), the management application is essentially protected from TCP/IP layer attacks.

For each joined ZoneRanger, the Ranger Gateway allocates a set of special ports that can be used to proxy management protocol traffic to the ZoneRanger itself. The ports assigned for each joined ZoneRanger can be listed using the listTcpPorts command on the Ranger Gateway. These special ports are typically allocated in the 20000's range. For example, for a single joined ZoneRanger, the listTcpPorts command might display the following:

```
ZR-Name http=20005 https=20006 sql=20007 ssh=20008 telnet=20009
```

Access to joined ZoneRangers via these special ports is also governed by port configuration rules. When the Ranger Gateway receives a request on one of these special ports, the Ranger Gateway identifies the ZoneRanger associated with the port, then maps the protocol associated with the port to the corresponding port on the ZoneRanger⁸, according to the following table:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td>80</td>
</tr>
<tr>
<td>https</td>
<td>443</td>
</tr>
<tr>
<td>sql</td>
<td>5432</td>
</tr>
<tr>
<td>ssh</td>
<td>22</td>
</tr>
<tr>
<td>telnet</td>
<td>23</td>
</tr>
</tbody>
</table>

---

⁸ Note that in some cases the port on the ZoneRanger is an internal port that can only be accessed by proxy through the Ranger Gateway.
Once the ZoneRanger address and the port on the ZoneRanger have been identified, the Ranger Gateway looks up the applicable port configuration in the portMap table, based on the source address and ZoneRanger address, then looks for the first matching rule in the portConfig table, based on the transport (TCP) and the ZoneRanger port. For example, assuming the listTcpPorts command output is as follows:

```
ZR-1 http=20005 https=20007 sql=20008 ssh=20009 telnet=20009
```

and if the ZoneRanger named “ZR-1” has an IP address of 10.10.4.5, if a request comes in on port 20008, the ZoneRanger address and port will be:

```
address=10.10.4.5, port=22
```

The Ranger Gateway will look first for a matching rule in the portMap table, using 10.10.4.5 as the destination address, then will look for a matching rule in the portConfig table using 22 as the rg-port. Assuming the default portMap and portConfig configuration, the following rules will be selected and the request will be allowed to proceed:

- **portMap:**
  
  `*.*.*..* @ZoneRanger ZoneRangerDefault`

- **portConfig:**
  
  `ZoneRangerDefault TCP 22 SSH`

Although at first glance this approach for handling the special Ranger Gateway ports may seem a little complicated, it has a significant advantage in that it allows access to a specific ZoneRanger/service to be governed by a single rule, regardless of whether the service is being accessed using special Ranger Gateway ports, or more directly via the GVI. In essence, access to a ZoneRanger service via a special Ranger Gateway port is made to look like the equivalent GVI request, then is processed accordingly.
Chapter 16: Proxy Caching

One of the primary advantages of ZoneRanger is that it is able to act as a proxy for management traffic on behalf of a wide variety of management applications. As a result, it is possible to have multiple management applications simultaneously proxying ICMP and SNMP traffic through a common pool of ZoneRangers to a common set of managed devices. As the number of management applications increases, and as polling rates increase, managed devices and the networks in which they reside may become inundated with repeated requests for the same information from different applications.

The ZoneRanger proxy caching feature addresses this issue by saving results of recent ICMP and SNMP proxy requests in an internal database, and using this cached information to respond to subsequent requests, as opposed to passing the proxy requests on to the managed devices.

ICMP and SNMP proxy caching are enabled and configured separately, as described in the following sections.

ICMP Proxy Caching

ICMP proxy caching is configured based on rules associated with specified IP addresses or IP address ranges. Options associated with each rule include the following:

- Whether or not a successful ping request to a managed device will be remembered in the cache, and how long a cached response will be considered to be valid.
- Whether or not an unsuccessful ping request to a managed device will be remembered in the cache, and how long a cached response will be considered to be valid.

When an ICMP ping request is received by the ZoneRanger, if ICMP proxy caching is enabled, the ZoneRanger will attempt to locate a matching caching configuration rule, and a cached result:

- If a valid cached result is found, based on the applicable configuration rule, the ZoneRanger will immediately send a positive response back to the Ranger Gateway, if the cached result was positive, or will discard the request if the cached result was negative.
- If there is no cached result, or if the cached result is determined to have expired, the request will be forwarded on to the managed device and will be processed normally.

Whenever an ICMP proxy request is forwarded to a managed device, the cache is refreshed based on the result. If the managed device responds to the ping, a positive result will be stored in the cache with the timestamp updated to the current time. If the request times out, a negative result will be stored in the cache.

SNMP Proxy Caching

SNMP proxy caching is configured based on rules associated with specific SNMP Object Identifiers (OIDs). Each rule has associated options specifying whether or not caching is enabled for that OID value, and if so, how long a cached response will be considered to be valid. When an SNMP Get/Set request is received by the ZoneRanger, if SNMP proxy caching is enabled, the ZoneRanger will inspect the OIDs specified in the request and attempt to locate matching caching configuration rules, and associated cached results:

- If valid cached results are found for all of the specified OIDs, the ZoneRanger will immediately send a response back to the Ranger Gateway with the cached values.
• If valid cached values are found for some but not all of the specified OIDs, a pared down request will be forwarded to the managed device, with the OIDs associated with valid cached results removed. When the response comes back from the managed device, the returned values will be stored in the cache and combined with the previously located cached values and the resulting response will be returned to the management application via the Ranger Gateway.

• If no valid cached results are found, the entire request will be forwarded to the managed device, the returned values will be stored in the cache, and the result will be returned to the management application via the Ranger Gateway.

Whenever an SNMP proxy request is forwarded to a managed device, the cache is refreshed based on the results with the timestamp updated to the current time.

SNMP proxy caching is based on OIDs so that users can configure different expiry times for different types of data. For example, information that is subject to frequent change, such as the status of an interface (i.e. ifOperStatus) should only be cached for a short period of time, if at all, while information that is relatively static, such as the contact name for a device (i.e. sysContact) can reasonably be cached for a relatively long period of time.

When the ZoneRanger searches for a rule to match a requested OID value, the OID value associated with each configured rule is treated as a prefix. As such, each rule is considered to match the specified OID value, and also to match the tree of OID values that begin with the specified value. For example, if the ZoneRanger were to receive a request for OID 1.3.6.1.2.1.1.4, a configured rule with OID value 1.3.6.1.2.1.1 would be treated as a match. This approach allows a single configured rule to be applied to a large number of OIDs, effectively reducing the number of rules that need to be configured. It is important to note that the configured rules are searched in order, and the first matching rule for a given OID is used. For example, if the following rules were configured:

```
1.3.6.1.2.1.1.4, 10 minutes
1.3.6.1.2.1.1, 5 seconds
```

If a request for OID 1.3.6.1.2.1.1.4 is received, the first rule will be applied, but if a request for OID 1.3.6.1.2.1.1.5 is received, the second rule will be applied.
Chapter 17: Proxy Map

The Proxy Map service in the Ranger Gateway supports the handling of management traffic originated by management applications and destined for managed devices (e.g. ICMP request, SNMP Get/Set request, HTTPS, SSH, FTP), enabling users to configure which ZoneRangers should be used to proxy traffic for given managed devices. Traffic originated by managed devices is typically governed by configuration rules within the ZoneRanger (e.g. forwarding rules, TACACS+/RADIUS server groups), and is outside of the scope of the Proxy Map service.

The Proxy Map service provides the following functions:

- For each proxy transaction (for example, ICMP echo request, SNMP request, SSH session request), the Proxy Map service identifies one or more joined ZoneRangers that are able to relay the proxy traffic to the target device, and selects one of these ZoneRangers to handle the given transaction.

- If network address translation (NAT) is in effect, the Proxy Map service translates the target device address associated with the request at the Ranger Gateway to the corresponding device address that the selected ZoneRanger must to communicate with the target device.

In simple ZoneRanger installations, the default Proxy Map configuration settings might be sufficient for the Proxy Map service to operate. For example, if a Ranger Gateway is joined to a single ZoneRanger, and no NAT is in effect, the Proxy Map service does not need additional configuration information to select a ZoneRanger and identify the target address for a proxy transaction. In general, the Proxy Map service configuration must be modified if any of the following conditions are true:

- The Ranger Gateway is joined to multiple ZoneRangers that are managing different devices (that is, the proxy map service cannot assume that any ZoneRanger can proxy traffic to any managed device).

- The firewall(s) associated with one or more of the network zones where joined ZoneRangers are installed are configured for NAT.

- IP Address Aliasing is being used (see Appendix D) and the alias IP addresses defined for managed devices do not match the actual IP addresses.
In order to describe the Proxy Map service in detail, it is useful to consider the network example shown in the following figure:

![Figure 17-1. Proxy Map Example](image)

Note the following from this figure:

- A single Ranger Gateway supports multiple management applications. In general, management applications can be co-resident with the Ranger Gateway software, or may execute on other servers.

- The Ranger Gateway is joined to three ZoneRangers (ZR-1, ZR-2, and ZR-3). ZR-1 manages devices in DMZ 1, while ZR-2 and ZR-3 manage devices in DMZ 3.

- Firewall 1 is not configured for NAT. Firewall 2 is configured to translate 64.2.37.* addresses to 192.168.1.* addresses.

When any of the management applications in this example initiate a proxy transaction, the initial request is relayed to the Ranger Gateway, along with some form of information that indicates the target DMZ device, as described in the following examples:

- Management Application 1 could initiate a proxy transaction, such an ICMP echo request, an SNMP Get request, or an SSH session request, directly to IP address 62.1.25.15. The Ranger Gateway can intercept the request via GVI, and must select a ZoneRanger (ZR-1) to relay the transaction. In this case, because no NAT is required, the Ranger Gateway will indicate to the selected ZoneRanger that the target DMZ device address is 62.1.25.15.
• **Management Application 1** could initiate a proxy transaction, such as an ICMP echo request, an SNMP Get request, or an SSH session request, directly to IP address 64.2.37.1. The Ranger Gateway can intercept the request via GVI, and must select a ZoneRanger (ZR-2 or ZR-3) to relay the transaction. In this case, because NAT is required, the Ranger Gateway will indicate to the selected ZoneRanger that the target DMZ device address is 192.168.1.1.

• **Management Application 2** could send an SNMP Get request to the Ranger Gateway, indicating 62.1.25.30 as the target DMZ device using a community string convention. The Ranger Gateway must select a ZoneRanger (ZR-1) to relay the transaction, and in this case, because no NAT is required, will indicate to the selected ZoneRanger that the target DMZ device address is 62.1.25.30.

• **Management Application 3**, a SOCKS-enabled SSH client, could initiate an SSH session to address 64.2.37.3, using the Ranger Gateway SOCKS server. In this case, the target device address is passed to the Ranger Gateway along the SOCKS protocol. The Ranger Gateway must select a ZoneRanger (ZR-2 or ZR-3) to relay the session, and in this case, given that NAT is in effect, must translate the target address to its corresponding address (192.168.1.3) before passing the request to the selected ZoneRanger.

The Proxy Map service in the Ranger Gateway would be responsible for selecting the ZoneRanger for each transaction and performing any necessary address translation before relaying the transaction to the selected ZoneRanger. The Proxy Map service makes these decisions based on configuration settings, ZoneRanger status information, and the content of an internal configuration table referred to as the *active proxy map*. Each entry in the active proxy map consists of the following fields:

- **rg-address**
  The host name or IP address of the target device for a proxy transaction, as indicated to the Ranger Gateway by the management application.

- **zoneranger**
  The host name or IP address of a ZoneRanger that might be selected to relay a proxy transaction.

- **zr-address**
  The actual host name or IP address that the ZoneRanger should use to access the target device. Note that if NAT is not in effect, this field can be omitted.

The active proxy map configuration table corresponding to the example above is as follows:

<table>
<thead>
<tr>
<th>rg-address</th>
<th>zoneranger</th>
<th>zr-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>62.1.25.15</td>
<td>ZR-1</td>
<td></td>
</tr>
<tr>
<td>62.1.25.30</td>
<td>ZR-1</td>
<td></td>
</tr>
<tr>
<td>64.2.37.1</td>
<td>ZR-2</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>64.2.37.1</td>
<td>ZR-3</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>64.2.37.2</td>
<td>ZR-2</td>
<td>192.168.1.2</td>
</tr>
<tr>
<td>64.2.37.2</td>
<td>ZR-3</td>
<td>192.168.1.2</td>
</tr>
<tr>
<td>64.2.37.3</td>
<td>ZR-2</td>
<td>192.168.1.3</td>
</tr>
<tr>
<td>64.2.37.3</td>
<td>ZR-3</td>
<td>192.168.1.3</td>
</tr>
</tbody>
</table>

If the target address for a proxy transaction is 62.1.25.15, the Proxy Map service would look up all entries with 62.1.25.15 in the **rg-address** column, and given that there is only one matching entry, would select the corresponding ZoneRanger (ZR-1). Given that the **zr-address** column for this entry is blank, the original target address would be passed on to the selected ZoneRanger.
If the target address for a proxy transaction is 64.2.37.3, the Proxy Map service would look up all entries with 64.2.37.3 in the rg-address column. In this case, two entries are found, one for ZR-2 and one for ZR-3. The Proxy Map service selects one of the entries based on configured criteria and status information, then passes the zr-address value from the selected entry (192.168.1.3) to the selected ZoneRanger.

Note that although these examples have used IP addresses, the Proxy Map service can work with IP addresses, host names, or combinations of both. In addition, the Proxy Map service can be configured with rg-address values specified as address patterns, or device groups. For example:

- 62.2.37.*
- 62.2.37.[3-7]
- *.company.com
- @MyGroup

When NAT is in effect, each entry in the active proxy map must include a zr-address value. In order for a single address pattern entry to be used for multiple devices, the zr-address must be specified as an address transform. Using address patterns and address transforms, the active proxy map configuration for the example network could be reduced to the following:

<table>
<thead>
<tr>
<th>rg-address</th>
<th>zoneranger</th>
<th>zr-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>64.1.25.*</td>
<td>ZR-1</td>
<td></td>
</tr>
<tr>
<td>64.2.37.[1-3]</td>
<td>ZR-2</td>
<td>192.168.1.*</td>
</tr>
<tr>
<td>64.2.37.[1-3]</td>
<td>ZR-2</td>
<td>192.168.1.*</td>
</tr>
</tbody>
</table>
The following flow chart illustrates the operation of the algorithm used by the Proxy Map service.

![Flow Chart](image)

**Figure 17-2. Proxy Map algorithm**

This algorithm makes use of a number of configuration settings:

- If the `resolve_host_names` setting is enabled, the address associated with the target device at the Ranger Gateway (that is, the `rg-address`) is resolved to an IP address before the active proxy map lookup is performed.

- When multiple ZoneRangers are available to relay a transaction for a given device, the selection is based on the `balance_zoneranger_selection` setting. If this setting is enabled, the Proxy Map service attempts to spread transaction load evenly across the available ZoneRangers, by tracking recent history of selection decisions, and preferring ZoneRangers that have been selected less frequently.

  If this setting is disabled, the proxy map service tends to prefer the ZoneRangers from which the Ranger Gateway has received the most recent communication. Disabling this setting provides the highest reliability, because the best ZoneRanger is selected for each transaction, and ZoneRangers that may be out of service tend to be bypassed. The disadvantage of this approach is that proxy traffic might be concentrated on a single ZoneRanger, while other available ZoneRangers may be bypassed.

- If no entries in the active proxy map match the given `rg-address`, and the `allow_unconfigured_routes` setting is enabled, the Proxy Map service simply selects the best available ZoneRanger, using the same criteria as described above, making the assumption that any joined ZoneRanger should be able to reach the target device, and that no NAT is in effect.
If this setting is disabled, and there are no matching entries in the active proxy map, the Proxy Map service indicates that there is no route to the target device. The intent of the `allow_unconfigured_routes` setting is to allow simple configurations (for example, one Ranger Gateway joined to two ZoneRangers that both manage the same DMZ, with no NAT) to operate without requiring the active proxy map to be configured. Where NAT is in effect, or where a Ranger Gateway is joined to ZoneRangers that manage disjoint networks, this setting must be disabled.

Proxy Map configuration settings can be modified using the Ranger Gateway Viewer or the `proxyMap` command.

In some cases, the target of a proxy transaction might be the ZoneRanger itself (e.g. querying ZoneRanger MIB values via SNMP proxy, or accessing the ZoneRanger text interface using SSH proxy). To handle these cases, the Proxy Map service algorithm performs an initial check to see if the `rg-address` matches the host name or IP address of a joined ZoneRanger. If a match is found, active proxy map lookup is bypassed, and the indicated ZoneRanger is selected as the best route to itself. The Ranger Gateway indicates 127.0.0.1 as the target address to the ZoneRanger, so that the ZoneRanger will know that the intended target of the transaction is the ZoneRanger itself.

The Proxy Map service automatically collects entries with identical `rg-address` values into groups and arranges these groups in an ordered list, according to the following rules:

- Groups of entries where the `rg-address` is specified as a specific value, as opposed to an address pattern, are placed at the head of the list, so that they can be given preference.
- Groups of entries where the `rg-address` value is specified as an address pattern are placed at the end of the list, in the order in which the groups were originally added to the active proxy map.

When looking up entries for a given `rg-address` value, the Proxy Map service searches through this ordered list, locates the first matching group, and selects a ZoneRanger from the entries in the first matching group, based on ZoneRanger status information, the `balance_zoneranger_selection` setting, and, where applicable, recent selection history, as described above.

The Proxy Map service can be especially helpful in situations where an organization needs to manage a network where IP address ranges are reused across multiple network zones. For example, this situation can arise whenever companies that have been using private Internet addresses are merged. In the absence of NAT, the recommended solution is to define unique virtual addresses that are mapped to real device addresses by the Proxy Map service.

The `proxyMap` command on the Ranger Gateway can be used to generate, modify, and list the contents of the active proxy map, and can also be used to view and modify configuration settings for the Proxy Map service.
For example, consider the network shown in the following figure:

![Network Diagram]

**Figure 17-3. Duplicate IP Proxy Map algorithm**

Note that addresses 192.168.1.2 and 192.168.1.3 are defined in both DMZ 1 and DMZ 2, and that virtual address spaces 10.1.1.* and 10.2.1.* have been configured to map to the devices in the two DMZ’s.

Even though this approach requires some amount of effort to manage and configure virtual addresses, it should be noted that in most cases, the scope of these addresses is confined to the management application server. As such, routers, firewalls, and other applications would have no awareness or visibility of these addresses, resulting in simpler configuration and maintenance than alternatives such as static NAT.
Chapter 18: Server Groups

ZoneRanger TACACS+ proxy and RADIUS proxy services can be used to proxy TACACS+ and/or RADIUS traffic from managed devices to configured authentication, authorization, and accounting (a.k.a. AAA) servers. Configuration of these proxy services is organized around the concept of server groups, where each server group contains the following information:

- The name of the server group
- A set of entries of the following form:
  (Ranger Gateway, TACACS+/RADIUS Server)
- Protocol options related to TACACS+
- Protocol options related to RADIUS

The underlying assumption behind server groups is that there may be multiple TACACS+/RADIUS servers in a group, primarily for reasons of high availability, and that any of the TACACS+/RADIUS servers in a group are equally able to handle authentication and authorization requests from a given set of devices.

ZoneRanger also supports the ability to define multiple server groups, and to associate different server groups with different device addresses, so that TACACS+/RADIUS traffic for different devices can be handled by different groups of servers. Each server group has its own set of (Ranger Gateway, TACACS+/RADIUS Server) entries and protocol-specific options. Once a set of server groups has been defined, proxy rules must be configured for each protocol, associating managed devices, or groups of managed devices with the server group that should be used for those devices. Each proxy rule associates an IP address, or range of IP addresses, with a server group name. Separate proxy rule tables are provided for TACACS+ and RADIUS.

For example consider the network shown in the following figure:

![Figure 18-1. Server Groups Example](image-url)

In this example, there are two redundant pairs of TACACS+/RADIUS servers:
• acs1 and acs2
• acs3 and acs4

Note that acs1 and acs2 have the Ranger Gateway software installed on the same server, while acs3 and acs4 are served by Ranger Gateway instances installed on servers rg3 and rg4. Assume that managed devices 10.1.1.22, 10.1.1.40, and 10.1.1.64 are to be served by acs1 and acs2, and that the router (10.1.1.1) and ZoneRangers (10.1.1.100 and 10.1.1.101) are to be served by acs3 and acs4. In order to support this configuration, the following server groups would be defined:

• server-group-1
  – acs1, 127.0.0.1
  – acs2, 127.0.0.1

• server-group-1
  – rg3, acs3
  – rg3, acs4
  – rg4, acs3
  – rg4, acs4

The pairs configured for each server group have two parts:

• A Ranger Gateway that can be used to relay a request to a TACACS+/RADIUS server.
• The address that the Ranger Gateway should use to communicate with the TACACS+/RADIUS server.

In server-group-1, where the Ranger Gateway instances are installed on the same physical servers as acs1 and acs2, the Ranger Gateway address is the same as the server address, and the Ranger Gateway can use the localhost address (127.0.0.1) to communicate with the TACACS+/RADIUS server software installed on the same physical server. In server-group-2, where the Ranger Gateway instances to be used are installed on separate servers (rg3 and rg4), either Ranger Gateway instance can be used to relay traffic to either TACACS+/RADIUS server, so additional pairs are configured, essentially listing all possible ways to reach all possible servers.

For any given request, the ZoneRanger will perform the following steps:

1. Identify the server group associated with the requesting device, based on configured rule tables associated with the TACACS+ and RADIUS services.
2. Go through the set of (Ranger Gateway, TACACS+/RADIUS server) entries associated with that server group, to identify a set of Ranger Gateway candidates. For example, if server-group-1 was selected in the previous step, the Ranger Gateway candidates would be rg3 and rg4.
3. Select a Ranger Gateway from the set of Ranger Gateway candidates, based on recent transaction history.
4. Relay the request to the selected Ranger Gateway, listing all TACACS+/RADIUS server candidates associated with the selected Ranger Gateway. For example, if rg3 was selected, the TACACS+/RADIUS server candidates would be: acs3 and acs4.

When the Ranger Gateway receives the relayed request, a TACACS+/RADIUS server will be selected from the list of server candidates, based on recent transaction history, and the request will be relayed to the selected server.
Server Groups are configured on the Configure > Access Control page Server Groups tab of the ZoneRanger web interface. Proxy rules for TACACS+ and RADIUS are configured on the TACACS+ and RADIUS tabs.

The simplest possible server group configuration is to define a single group. The following steps would be required:

- Define a single server group named MyServerGroup
- Add the following proxy rule to the TACACS+ table:
  
  *.*.*.* MyServerGroup

- Add the following rule to the RADIUS table:
  
  *.*.*.* MyServerGroup

Using this configuration, the ZoneRanger will select a server from MyServerGroup to handle TACACS+ and RADIUS requests from all managed devices. If there was a need to configure a second server group to handle requests originated by specific devices, the following steps would be required:

- Define a new server group (e.g. MyOtherServerGroup)
- Insert proxy rules for the specific IP addresses or IP address ranges to the top of the TACACS+ table:
  
  10.254.1.1 MyOtherServerGroup
  10.254.2.[10-20] MyOtherServerGroup
  *.*.*.* MyServerGroup

- Insert proxy rules for the specific IP addresses or IP address ranges to the top of the RADIUS table:
  
  10.254.1.1 MyOtherServerGroup
  10.254.2.[10-20] MyOtherServerGroup
  *.*.*.* MyServerGroup

When handling a TACACS+ or RADIUS request from a given device, the ZoneRanger will search through the proxy rules table associated with the protocol being used for the first rule that matches the requesting device’s address. As such, it is important to ensure that specific address rules are placed ahead of overlapping range or wild-card rules.
Chapter 19: Whitelist

Inbound

ZoneRanger can receive many different types of inbound data such as SNMP traps, Syslog messages, TACACS+ requests, etc. In the case of a node-licensed ZoneRanger, the source of the information will be verified as a managed node before the data will be processed based on the ZoneRanger configuration. In the case of a ZR-SPX licensed ZoneRanger, no management check occurs and the data will be processed based on the ZoneRanger configuration.

ZoneRanger may be configured with a specific set of devices (“whitelist”) from which it will receive information. Thus, when the Whitelist feature is enabled, only inbound data with a source address configured in the whitelist, will be further processed based on the ZoneRanger configuration. If the source address of the inbound data is not specified in the whitelist, it will be dropped with no further processing.

For security purposes, the whitelist provides a mechanism for the ZR-SPX licensed ZoneRanger to restrict the set of IP addresses from which it will accept information. In the case of the node-licensed ZoneRanger, the use of the whitelist provides an additional security check as well as a performance improvement in that the ZoneRanger will no longer need to verify whether or not the incoming source address is from a managed node.

Joined Ranger Gateways and redundant ZoneRangers are automatically whitelisted, but will not appear in the whitelist configuration. However, new Join requests from another Ranger Gateway and Redundancy requests from another ZoneRanger will be subject to the whitelist. Thus, the IP address of the new Ranger Gateway or new redundant ZoneRanger must be specified in the whitelist for the request to be successful.

Outbound

ZoneRanger can proxy many different types of outbound data such as SNMP proxy, ICMP proxy, TCP Proxy, etc. In the case of a node-licensed ZoneRanger, the destination of the request will be verified as a managed node before the request will be processed based on the ZoneRanger configuration. In the case of a ZR-SPX licensed ZoneRanger, no management check occurs and the request will be processed based on the ZoneRanger configuration.

ZoneRanger may be configured with a specific set of devices (“whitelist”) to which it will send information. Thus, when the Whitelist feature is enabled, only outbound data with a source address configured in the whitelist, will be further processed based on the ZoneRanger configuration. If the source address of the outbound data is not specified in the whitelist, it will be dropped with no further processing.

For security purposes, the whitelist provides a mechanism for the ZR-SPX licensed ZoneRanger to restrict the set of IP addresses from which it will send information. In the case of the node-licensed ZoneRanger, the use of the whitelist provides an additional security check as well as a performance improvement in that the ZoneRanger will no longer need to verify whether or not the outgoing source address is from a managed node.

Enforcing the whitelist for outbound requests will include any traffic sent from the ZoneRanger. Thus is will effect Discovery, Root Cause, and Diagnostics requests initiated by the ZoneRanger as well as requests proxied from a joined Ranger Gateway.

Joined Ranger Gateways and redundant ZoneRangers are automatically whitelisted, but will not appear in the whitelist configuration. However, new Join and Redundancy requests initiated from the ZoneRanger will be subject to the whitelist. Thus, the IP address of the new Ranger Gateway or new redundant ZoneRanger must be specified in the whitelist for the request to be successful.
Part III. ZoneRanger Services

Chapter 20: Discovery

Populating the ZoneRanger Database

Most ZoneRanger management services as well as proxy services for non-SPX ZoneRanger models require the ZoneRanger database to be populated with information about the entities that the ZoneRanger is intended to manage, such as nodes, interfaces, and TCP ports.

The process of analyzing the network and populating the ZoneRanger database is called discovery. The process can range from a simple analysis to identify interfaces and subnets associated with a predefined set of configured seed nodes, to an aggressive search for network device information, combining techniques such as seed node analysis, ping range sweeps, ARP cache and route table searches, broadcast pings, and root cause path analysis.

The extent and aggressiveness of the network analysis that is performed is based on a variety of configuration options. This chapter describes how discovery is configured and executed on a ZoneRanger.

The ZoneRanger database is organized into tables based on the types of entities that can be discovered, such as nodes, interfaces, subnets, and TCP ports. The database also includes relationships between entities, for example, the interfaces and TCP ports associated with a node, the interfaces on a given subnet, and so on.

ZoneRanger discovery is an automated process that can be invoked manually as needed, or configured to run periodically. The first time discovery runs on a ZoneRanger, the database is populated based purely on the results of analyzing configured seed nodes and ping ranges. On subsequent discovery runs, existing database content is merged with new discovery results so that existing entities are updated with any changes, and new entities can be added to the database.

The Discovery Algorithm

To understand the various discovery configuration options, a simplified overview of the discovery algorithm is helpful. The ZoneRanger discovery algorithm comprises two main types of activity:

- Accumulating addresses (that is, building a list of entity addresses to analyze)
- Analyzing addresses (that is, analyzing the entity addresses in the list).

The simplest way to accumulate addresses is to configure seed nodes and ping ranges. Configured seed nodes are automatically added to the list of addresses to be analyzed. Any addresses in a configured ping range that pass the include network/exclude network filter criteria and respond to a ping are also added to the list of addresses to be analyzed.

The primary purpose of analyzing addresses is to identify nodes, interfaces, subnets, and TCP ports, and the relationships between these entities, to populate the database.

In addition, depending on configuration options, additional addresses may be discovered while analyzing addresses:

- In the interface table of a device
- In the ARP cache of a device
- In the route table of a device
- As the result of entities responding to broadcast ping requests
During the routing path analysis required for the ZoneRanger root cause algorithm, most of these mechanisms can be disabled using configuration options. To limit discovery to the portions of the network that are of interest, addresses discovered during analysis activities are added to the list of addresses to be analyzed only if they pass the include network/exclude network filter criteria.

**Default discovery behavior**

The following default discovery settings (Configuration > Discovery) are generated during initial ZoneRanger configuration:

- **Search for additional nodes** is enabled, and the Read IP route table, Read ARP cache, and Broadcast ping enabled suboptions are enabled.

- **Auto configure polling for newly discovered nodes** is enabled.

- **Auto Manage newly discovered nodes** is enabled.

- The **Seed Node List** is populated with a single entry containing the default gateway IP address as specified during initial ZoneRanger configuration.

- The **Include Networks** list is populated with a single entry, generated by taking the IP address of the ZoneRanger specified during initial ZoneRanger configuration, and masking off the last three octets. For example, if the ZoneRanger's IP address is 10.254.1.190 the default include network is 10.0.0.0 which limits discovery to IP addresses of the form 10.n.n.n. The **Exclude Networks** list is initially empty.

- The **Ping Ranges** list is initially empty.
• Several common TCP services and their associated ports are configured for discovery.

• A list of common sysObjectID device type mappings is provided for commonly used routers and switches.

During initial configuration, you are given the option to automatically run discovery after initial configuration is complete. If this option is accepted, discovery runs, using these default settings, immediately after the ZoneRanger completes its initial startup. In addition, the periodic discovery option will be enabled with a default interval of 7 days. If this option is declined, you can wait until the ZoneRanger starts up, modify the discovery settings as desired, and then either invoke discovery manually or configure periodic discovery.

**Incremental Discovery**

As an alternative to running the full discovery algorithm, ZoneRanger also provides a mechanism whereby the user can request that a short list of IP addresses or hostnames be scanned and incrementally added to the database. When this mechanism is invoked, the ZoneRanger will scan each provided IP address or hostname using SNMP and TCP and will update the ZoneRanger database with the resulting information. If a new IP address or hostname is scanned, information for that device will be added to the database. If an existing IP address or hostname is scanned, the information in the ZoneRanger database for that device will be updated.

Note that incremental discovery is intended as a shortcut, to be used in cases where executing the full discovery algorithm would consume too much time. Accordingly, the incremental discovery algorithm does not gather subnet or path information for added devices. As a result, root cause information will not be available for these newly added devices until a full discovery is executed.
Chapter 21: Forwarding

ZoneRanger can be configured to listen for UDP traffic on specified ports, and forward that traffic through a Ranger Gateway to a specified destination host and port that is reachable from the Ranger Gateway. ZoneRanger can forward the following types of traffic:

- Trap
- Syslog
- sFlow
- NetFlow
- Generic UDP

Unless the ZoneRanger model is SPX, the traps, datagrams, and messages that ZoneRanger forwards must originate from managed nodes. In the case of the SPX model, all information is forwarded regardless of originating address. Forwarding rules are used to specify the type, source, and destination of UDP information on the ZoneRanger to which to forward to a Ranger Gateway and ultimately a management application.

![Figure 21-1. ZoneRanger Forwarding](image)

When UDP information is forwarded from the ZoneRanger to a Ranger Gateway to its ultimate destination, the source address of the UDP information when it is received by the management application is the address of the Ranger Gateway since it sent the UDP data. However, The Ranger Gateway can be configured for particular protocols to “spoo” the address of the information to be that of the original sending device instead of the address of the Ranger Gateway. Depending on the requirements of the management application, spoofing may need to be configured for particular protocols.

Forwarding Rules

Forwarding rules define the source and destination information of the UDP data to be forwarded. Forwarding rules are configured on the ZoneRanger Configuration > Forwarding page. Each forwarding rule is associated with a Ranger Gateway. Any number of forwarding rules may be associated with a Ranger Gateway.

Forwarding rules are defined by the following criteria:

- Type of UDP traffic
- Source address of the UDP traffic
- Local port ZoneRanger listens for UDP traffic
- Destination host to which Ranger Gateway will forward UDP traffic
- Destination port on destination host to which Ranger Gateway will forward UDP traffic
- Any filtering associated with this type of UDP traffic.

**Forwarding Logging** may be enabled to determine if UDP data is received by the ZoneRanger and whether or not that UDP data passes a configured Forwarding Rule. The /log/udpFwd.log file can be downloaded using a joined Ranger Gateway.

**Trap Forwarding**

ZoneRanger has the capability to receive SNMP traps from managed devices and forward those traps through a Ranger Gateway to another application. When an SNMP trap is received by ZoneRanger, the trap is verified to be syntactically correct. Thus, if the SNMP trap does not meet the RFC definition of a correctly formatted SNMP trap, it will be discarded. If the SNMP trap is inspected to determine whether or not it is syntactically correct, it will be processed by the ZoneRanger forwarding service.

Trap filters are a named set of conditions which may be created in ZoneRanger to refine the forwarding of SNMP traps. Through the ZoneRanger **Configuration > Forwarding** page **Trap Filters** tab, a specific set of conditions may be created to apply to Trap Forwarding Rules. The following conditions may be used in the creation of Trap Filters:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trap</td>
<td>Previously defined trap known by its name.</td>
</tr>
<tr>
<td>Enterprise ID</td>
<td>SNMPv1 Enterprise OID of a trap or an OID prefix. If the trap is not SNMPv1, its Enterprise OID is described in RFC 3584.</td>
</tr>
<tr>
<td>Generic Type</td>
<td>SNMPv1 Generic Type of a trap. If the trap is not SNMPv1, its Generic Type is described in RFC 3584.</td>
</tr>
<tr>
<td>Specific Type</td>
<td>SNMPv1 Specific Type of a trap. If the trap is not SNMPv1, its Specific Type is described in RFC 3584.</td>
</tr>
<tr>
<td>Trap OID</td>
<td>SNMPv2c Trap OID of a trap or an OID prefix. If the trap is SNMPv1, its Enterprise OID is described in RFC 3584.</td>
</tr>
<tr>
<td>Variable Binding</td>
<td>Variable Binding value of a trap, defined by its index starting at 1. An '<em>' may be used at the beginning and/or end of the value to denote a wildcard match. Any '</em>' inside the value is not treated as a wildcard.</td>
</tr>
<tr>
<td>Agent</td>
<td>SNMPv1 Agent of a trap. If the trap is not SNMPv1, the Agent is described in RFC 3584. Multiple agents can be listed using commas between IP addresses and may be an address pattern.</td>
</tr>
<tr>
<td>Version</td>
<td>SNMP version of the trap</td>
</tr>
<tr>
<td>Community</td>
<td>SNMP community string. In the case of an SNMPv3 trap, the user name will be compared rather than the community string.</td>
</tr>
<tr>
<td>Filter</td>
<td>Defines another previously defined trap filter as a condition for this filter</td>
</tr>
</tbody>
</table>

Each condition may be defined as positive (default) or negative (“Not”). Also, each trap filter may be configured so that the trap passes the filter if it meets all of the specified conditions or if it meets at least one of the specified conditions.
Each verified SNMP trap will be compared to the set of all configured Forwarding Rules. If the SNMP trap meets the conditions of a Forwarding Rule, the SNMP trap is securely sent to the corresponding Ranger Gateway to be ultimately sent to the Destination Host and Port as configured in that specific Forwarding Rule. An SNMP Trap may match multiple Forwarding Rules, even to the same Ranger Gateway. Unless the group name of the ZoneRangers which are forwarding the traps is the same in which case the traps will be deduplicated, the traps will be forwarded multiple times.

After a trap has passed a Forwarding Rule, the trap may also be configured to be converted to an SNMPv1 or SNMPv2c. If an SNMPv3 or SNMPv2c inform is received and it is to be forwarded as SNMPv1, it will be converted to a trap. In this case, the ZoneRanger will respond to the originating device with an appropriate response after forwarding the trap.

The ZoneRanger is only able to process an incoming SNMPv3 Inform if there is a configured SNMPv3 user or the Inform is using noAuthNoPriv Security Level. When the ZoneRanger is able to process an incoming SNMPv3 Inform, the ZoneRanger will convert the Inform to an SNMPv3 Trap, forward the trap based on any configured forwarding rules, and respond to the client that the Inform was received. ZoneRanger can forward SNMPv3 traps which use any Security Level regardless of whether or not there is a configured SNMPv3 user.

There are some limitations when SNMPv3 users are not configured for SNMPv3 traps and informs:

1. Encrypted notifications will not match any trap filters using properties of the PDU with the exception of version.
2. The ZoneRanger will not return responses to the client when it receives an SNMPv3 Inform.
3. Duplicate encrypted notifications will not be discarded on the Ranger Gateway.

**Syslog Forwarding**

ZoneRanger has the capability to receive Syslog messages from managed devices and forward those messages through a Ranger Gateway to another application. When a Syslog message is received by ZoneRanger, the message is inspected to determine whether or not to be syntactically correct. Thus, if the message does not meet the RFC definition of a correctly formatted Syslog message, it will be discarded. If the Syslog message is verified to be syntactically correct, it will be processed by the ZoneRanger forwarding service. Otherwise, it will be discarded.

Syslog filters may be use to further refine the forwarding of messages within a particular syslog forwarding rule. When configuring a syslog type Forwarding Rule, the Edit button allows for the syslog filter specification. ZoneRanger has the ability to specially process syslog messages sent from Cisco devices (Cisco Syslog). Specific syslog filters may be created related to Cisco syslog messages. The following conditions may be used in the creation of syslog filter specification:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
</table>

*ZoneRanger 5.6 User's Guide* 65
<table>
<thead>
<tr>
<th>Program Name</th>
<th>Name of the program that generated the syslog message, as the name appears in the message.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Search</td>
<td>Search string that the syslog message must contain. The search string can be a regular expression search. This will forward syslogs if the text matches the given message. <strong>Not</strong> can be applied to the message search preventing messages with specific text from being forwarded.</td>
</tr>
</tbody>
</table>
| Cisco Syslog with Max Severity | Cisco syslog messages of the specified severity or lower.  
**Note:** The severities are more urgent the lower the number, so this filter includes the specified severity and those that are more urgent. |
| Syslog with Max Severity | Syslog messages of the specified severity or lower.  
**Note:** Syslog severities are more urgent the lower the number, so this filter includes the specified severity and those that are more urgent. |
| Syslog with Facility | Syslog messages of the specified severity or lower.  
**Note:** Syslog severities are more urgent the lower the number, so this filter includes the specified severity and those that are more urgent. |

If multiple criteria are selected, a Syslog message must match all selected criteria to be forwarded. Also, syslog filters allow messages to be forwarded as a syslog message or forwarded as an SNMP trap. If the **Cisco Syslog with Max Severity** criteria is chosen, the correct Cisco trap for the severity is generated. Otherwise, a Syslog trap with the specified **Specific Type** is generated.

**NetFlow and sFlow Forwarding**

ZoneRanger has the capability to receive NetFlow and sFlow packets from managed devices and forward those packets through a Ranger Gateway to another application. When a NetFlow or sFlow packet is received by ZoneRanger, the packet is inspected to determine whether or not to be syntactically correct. For NetFlow, version 5 and version 9 packets will be verified. For sFlow, version 4 and version 5 packets will be verified. Any other version will be discarded. If the NetFlow or sFlow packet is verified to be syntactically correct, it will be processed by the ZoneRanger forwarding service. Otherwise, the packet is discarded.

**Generic UDP Forwarding**

ZoneRanger has the capability to receive generic UDP traffic from managed devices and forward those packets through a Ranger Gateway to another application. Since there is no configured format for this UDP traffic, no verification occurs before the packets are processed by the ZoneRanger forwarding service.
Chapter 22: FTP Proxy

The basic intent of the FTP protocol is to allow client applications to transfer files to/from a remote server. The FTP protocol is based on TCP and separates control and data into separate TCP connections. In cases where the client and server are separated by a firewall, this separation of control and data connections creates a problem. While control connections are always directed at a well-known port, data connections use dynamically assigned ports, making it difficult to configure the firewall to allow only the needed ports. Making matters worse, the direction in which the data connection is initiated depends on whether requested transfer mode is active or passive, making it difficult to implement a policy preventing initiation of connections from less secure network zones to more secure network zones.

The ZoneRanger FTP proxy service provides an effective solution for these problems, acting as an application-layer proxy firewall for FTP traffic, enabling FTP clients to exchange files with servers located within firewall-partitioned networks.

The following figure provides a high-level overview of an FTP proxy transaction. Note that the Management Application Server in this figure is acting as an FTP client, and one or more managed devices may act as FTP servers.

Figure 22-1. ZoneRanger FTP Proxy

Note that the put file or get file requests and associated responses shown in this figure are exchanged via the control connection, while the transfer of the actual file content takes place over a separate TCP data connection. The ZoneRanger FTP proxy service carefully inspects all FTP control connection traffic, and only those data connections that are matched with known outstanding transfer requests, are allowed to pass.

The ZoneRanger FTP proxy feature supports all FTP protocol transactions defined in RFC 959, including:

- Get File Request (from devices in a firewall-partitioned zone)
- Put File Request (to devices in a firewall-partitioned zone)
- List Directory Request
- Delete File Request
- Rename File Request
In addition to supporting active and passive mode file transfers, the ZoneRanger FTP proxy feature (Configuration > Inbound Proxy page TCP tab) also includes an optional active-to-passive conversion feature, allowing an FTP client’s active mode transfer requests to be presented to clients as passive mode requests, so that clients that only support active mode are able to exchange files with servers that only support passive mode. The active-to-passive conversion feature is enabled on a per-ZoneRanger basis. When this feature is enabled the ZoneRanger will present all FTP proxy requests to managed devices in the form of passive requests, regardless of whether to FTP client’s request is active or passive.

In cases where the FTP servers reside in a less secure network than the FTP clients, enabling active-to-passive conversion, where possible, is recommended from a security perspective, because it ensures that all FTP data connections are originated by the ZoneRanger, rather than by the managed devices.

While the ZoneRanger FTP proxy service is able to mitigate a variety of risks associated with passing FTP traffic between firewall-partitioned networks, it should be noted that the FTP protocol exchanges user ID and password information over an unencrypted TCP connection, which results in a certain amount of unavoidable risk. As a result, it is recommended that newer, more secure file transfer protocols such as SCP and SFTP alternatives be used wherever possible. The ZoneRanger FTP proxy service is recommended for use only in cases where the managed devices reside in a highly secure network, or where FTP is the only solution available, given the management applications and/or managed devices involved.
Chapter 23: HTTP/HTTPS Proxy

A Ranger Gateway and one or more joined ZoneRangers can provide an HTTP/HTTPS proxy service, enabling access to web servers located in firewall-partitioned networks, without requiring the firewall to be configured to pass HTTP or HTTPS.

The following figure provides a high-level overview of an HTTP/HTTPS proxy transaction. Note that the Management Application Server in this figure is acting as a web browser, and one or more managed devices may act as web servers.

![Figure 23-1. ZoneRanger HTTP/HTTPS](image)

In addition to using HTTP/HTTPS proxy to communicate with managed devices, the HTTP and HTTPS proxy services can also be used to access the ZoneRanger web interface for joined ZoneRangers.

While the ZoneRanger is able to proxy both HTTP and HTTPS protocols, HTTPS will typically be the preferred protocol for most applications, because the HTTP protocol may exchange user ID and password information over an unencrypted TCP connection, and therefore is less secure. As a result, HTTPS proxy is enabled by default and HTTP is disabled by default for managed devices.

Web browsers can access HTTP and HTTPS Proxy services in a variety of ways, as described in the following sections.

GVI/RGVI

When using GVI or RGVI, the web browser sends HTTP or HTTPS requests intended for a managed device to the actual address of the target device, or an address that can be uniquely mapped to the target device. The management application server is configured with static routing rules, so that traffic destined for devices located in firewall-partitioned networks is routed to a virtual interface, which then forwards the traffic to the Ranger Gateway.

When the Ranger Gateway receives the initial TCP connection request for an HTTP or HTTPS session, it will check the Proxy Access Control configuration to verify that the request should be allowed, and to identify the proxy service to which the request should be forwarded. The Ranger Gateway will then consult the Proxy Map service in order to identify a ZoneRanger that is able to relay the request to the target device. The request is then forwarded to the selected ZoneRanger, which in turn, establishes a TCP connection to the target device. Once this TCP connection is established, the ZoneRanger will inform the Ranger Gateway, and the Ranger Gateway will complete the establishment of the initial TCP connection (i.e. the connection between the web browser and the Ranger Gateway). From this point on, the Ranger Gateway and selected ZoneRanger will relay HTTP or HTTPS data between the web browser’s TCP connection to the Ranger Gateway and the ZoneRanger’s TCP connection to the target device, until one of the connections is disconnected.
The primary advantage of GVI and RGVI is that the existence of the HTTP/HTTPS proxy is completely transparent to the management application. Common routing mechanisms within the underlying operating system are used to intercept traffic bound for devices in firewall-partitioned networks, so there is no need to modify or reconfigure the management application in any way. Another advantage is that the same mechanism can be used for other proxy services, such as ICMP proxy, or SNMP proxy.

**SOCKS**

SOCKS is a standard protocol for generic TCP and UDP proxy services that can be used to redirect management traffic from the management application to a SOCKS server integrated within the Ranger Gateway. In order to use SOCKS, either the management application must include built-in support for SOCKS, or generic SOCKS “shim” software must be installed on the management application server. The shim software inserts itself between the management application and the server’s TCP/IP stack, and redirects traffic for specified IP addresses and ports to a SOCKS server, based on configuration information.

In order to access a managed device through HTTP or HTTPS proxy, a SOCKS-aware web browser initially establishes a TCP connection to the SOCKS port (by default, 4855) on the Ranger Gateway. After this connection is established, the client application sends a SOCKS connection request to the Ranger Gateway, indicating the managed device and port to which the client would like to connect.

The SOCKS server on the Ranger Gateway will check the Proxy Access Control configuration to verify that the request should be allowed, and to identify the proxy service to which the request should be forwarded. The Ranger Gateway will then consult the Proxy Map service in order to identify a ZoneRanger that is able to proxy traffic to the target device, and to translate the target address, if necessary, then forwards the connection request to the selected ZoneRanger, which attempts to connect to the target device. If this connection is successfully established, the ZoneRanger notifies the Ranger Gateway, which in turn notifies the web browser.

From this point, the Ranger Gateway and selected ZoneRanger simply relay data between the client application’s TCP connection to the Ranger Gateway and the ZoneRanger’s TCP connection to the target device, allowing the web browser and target device to exchange HTTP/HTTPS requests and responses. The Ranger Gateway and ZoneRanger continue to relay data until one of the connections is disconnected. Most web browsers support the SOCKS protocol.

As an example, the following steps would be used to configure Internet Explorer 6.0 to use the SOCKS server on the Ranger Gateway (port 4855).

1. Select **Internet Options...** from the **Tools** menu and click the **Connections** tab. The resulting dialog should be as shown in the following figure.
2. Click the **LAN Settings** button. A dialog box will open as shown in the following figure.

![Internet Explorer Connections tab](image)

*Figure 23-2. Internet Explorer ... Connections tab*

3. Check the **Use a proxy server for your LAN** box, then click the **Advanced...** button. A dialog box will open as shown in the following figure.

![Local Area Network (LAN) Settings](image)

*Figure 23-3. Internet Explorer ... LAN Settings*
4. Enter the IP address or hostname of the Ranger Gateway and the SOCKS server port in the Socks row of the Servers table as shown in the figure. Note that the Use the same proxy server for all protocols box should be left unchecked, and all of the other rows in the Servers table should be left blank as shown in the figure.

5. Click OK in the three dialog boxes to save your changes.

**Dedicated HTTP/HTTPS Ports**

When a ZoneRanger is joined to a Ranger Gateway, the Ranger Gateway allocates dedicated ports that can be used to access various services (for example, HTTP, HTTPS, SQL, Telnet, and SSH) on the newly joined ZoneRanger.

You can use the `list TcpPorts` command on the Ranger Gateway command interface or the Ranger Gateway Viewer to identify the ports that have been allocated for each ZoneRanger.

For example, the following figure shows the dedicated ports associated with a selected ZoneRanger listed on the Information tab of the main window of the Ranger Gateway Viewer. Note that the HTTP Port in this case is 20000 and the HTTPS Port is 20001.
A web browser can establish a proxy connection to a joined ZoneRanger simply by connecting to the Ranger Gateway’s address, specifying the dedicated HTTP or HTTPS port associated with that ZoneRanger as the destination port, as shown in the following figure.
As a shortcut, the user can automatically launch their default browser and browse via a dedicated HTTP or HTTPS port to the web interface of the selected ZoneRanger by clicking the **Browse (HTTP)** or **Browse (HTTPS)** buttons on the **Status** tab of the Ranger Gateway Viewer’s main window as shown in the following figure.

**Figure 23-7. Ranger Gateway Main Window**
Chapter 24: ICMP Proxy

A Ranger Gateway and one or more joined ZoneRangers can provide an ICMP proxy service, enabling management applications to send ICMP echo requests (a.k.a. “ping”) to devices in firewall-partitioned networks (e.g. DMZs), and to receive associated ICMP echo responses, without requiring the firewall to be configured to allow ICMP traffic. The ICMP proxy service will only proxy ICMP echo requests and responses.

The following figure provides a high-level overview of an ICMP proxy transaction.

The basic steps of an ICMP proxy transaction are as follows:

1. A management application generates an ICMP echo request, intended for a specific managed device.
2. The Ranger Gateway intercepts the ICMP echo request via the GVI or RGVI interface.
3. The Ranger Gateway checks the Proxy Access Control configuration to verify that the ICMP echo request should be allowed, and to identify the proxy service to which the ICMP echo request should be forwarded (i.e. ICMP Proxy).
4. The ICMP Proxy service in the Ranger Gateway consults with the Proxy Map service to select a ZoneRanger that is able to relay the ICMP echo request to the target device, then forwards the ICMP echo request to the selected ZoneRanger.
5. The ZoneRanger forwards the ICMP echo request to the managed device.
6. The managed device generates an ICMP echo response and sends it to the requesting ZoneRanger.
7. The ZoneRanger forwards the ICMP echo response to the Ranger Gateway.
8. The Ranger Gateway forwards the ICMP echo response to the management application.

In order to use the ICMP proxy service, the GVI or RGVI service, as described in Chapter 7, on the Ranger Gateway must be enabled, and configured to intercept traffic destined for managed devices located in firewall-partitioned networks.

The ICMP proxy service can optionally be configured to perform proxy caching as described in Chapter 33.
Chapter 25: NTP Proxy

The Network Time Protocol (NTP) allows network devices and servers to synchronize their clocks with one or more centralized time servers, across a variable-latency network. In applications where time synchronization across devices is important, the ability to administer time across a large number of devices from a small number of centralized time servers using NTP is a significant advantage.

A ZoneRanger, joined to a Ranger Gateway, can provide an NTP proxy service, enabling devices in firewall-partitioned networks to synchronize their clocks with NTP servers located in other network zones, without requiring configuration of firewall rules to allow NTP traffic. ZoneRanger’s NTP proxy service can be configured to operate in either of two modes:

1. The ZoneRanger can obtain its time from a centralized NTP server (either directly or via a joined Ranger Gateway), and can act as a secondary time server, responding autonomously to NTP requests from client devices, as illustrated in the following figure.

![Figure 25-1. ZoneRanger NTP Proxy -- Mode 1](image)

2. The ZoneRanger can act as straight NTP protocol proxy, inspecting NTP requests received from client devices, relaying valid requests via a joined Ranger Gateway to a centralized timer server, and relaying server responses back to the requesting clients, as illustrated in the following figure.

![Figure 25-2. ZoneRanger NTP Proxy -- Mode 2](image)
Configuring ZoneRanger to Act as a Secondary NTP Server

The following steps are required to configure ZoneRanger to act as a secondary NTP time server and can be found on the web interface on the Configuration > System page Time tab:

1. The **Time Synchronization Enabled** option should be enabled.
2. The **Server Type** for time synchronization should be set to NTP.
3. If NTP authentication is required, a set of one or more (index, key) pairs should be configured in the **NTP Keys** table. This set of (index, key) pairs may include:
   - Pairs that match the values configured on the NTP servers, as described in part “c” of the following step.
   - Additional pairs that managed devices may use when authenticating NTP traffic to/from the ZoneRanger.
4. A list of NTP time servers should be configured in the **NTP Servers** table. The following information is required for each entry in the list:
   - The Ranger Gateway through which the ZoneRanger will access the NTP server. The value “None (direct)” can be specified if the ZoneRanger should access the NTP server directly.
   - The IP address or hostname of the NTP server.
   - The authentication (index, key) to be used when accessing the NTP server. The selected (index, key) must match the configuration of the specified server. That is, the set of (index, key) pairs configured on the specified server must include the selected (index, key) pair.
5. The **ZoneRanger Acts as NTP Server** option should be enabled.
6. Optionally, the **Authenticate Client Requests** option may be enabled.
7. Managed devices should be configured to use the ZoneRanger as their NTP time server. For high availability, multiple ZoneRangers configured to act as an NTP server can be deployed in the same network. If the redundant ZoneRangers are configured with the virtual IP address feature enabled, the configured NTP server list for each managed device can simply include the virtual IP address. If the virtual IP address feature is not enabled, the configured NTP server list for each managed device should include the specific address for each NTP-enabled ZoneRanger.

Configuring ZoneRanger to Act as an NTP Proxy

The following steps are required to configure ZoneRanger to act as an NTP proxy and can be found on the web interface on the Configuration > Inbound Proxy page NTP tab:

1. The ZoneRanger must be configured not to act as an NTP server. This can be accomplished in a variety of ways:
   - Time Synchronization on the Configuration > System page Time tab can be disabled.
   - The **Server Type** for time synchronization can be set to a value other then NTP.
   - The **ZoneRanger Acts as NTP Server** option can be disabled.
2. A list of NTP time servers should be configured in the **Proxy NTP Servers**. The following information is required for each entry in the list:
a. The **Ranger Gateway** through which the ZoneRanger will access the NTP server. Note that the option for the ZoneRanger to access an NTP server directly is not supported in this case.

b. The IP address or hostname of the **NTP Server**.

3. Optionally, the **Validate Authentication** option may be enabled. If this option is enabled, a set of (index, key) pairs must be configured in the **Keys** table. These values must match the corresponding values configured on the NTP servers that were specified in the previous step. In the case of NTP proxy, the managed device acting as an NTP client does not have any control over which NTP server will be chosen to handle each request. As such, the table of (index, key) values in the specified NTP servers must be identical, at least for the set of indexes that the managed devices may use.

4. By default, when the ZoneRanger proxies a NTP proxy request through the joined Ranger Gateway to the NTP server, the source of the NTP request will appear to the NTP server as if the NTP client is the Ranger Gateway server and not the actual ZoneRanger managed device. It is possible to configure the Ranger Gateway to spoof the source address in the NTP proxy request to be that of the ZoneRanger managed device instead of the address of the Ranger Gateway. To configure NTP proxy spoofing, either use the Ranger Gateway Viewer **Configure > Gateway Settings** menu, **NTP Proxy** area or the `configGateway` command on the Ranger Gateway.

**Additional Note**

If the Ranger Gateway is communicating with an NTP server installed on the same server using GVI or on a different server using RGVI, it may be necessary to restart the NTP server after the GVI service or RGVI client on that server are enabled. The startup procedure of some NTP servers is to enumerate the IP addresses on a given server, and then to specifically bind a listener for each of these addresses. Enabling GVI or starting an RGVI client effectively creates a new IP address, associated with an underlying virtual point-to-point interface, and if this interface was not configured at the point when the NTP server was started, the NTP server will be unaware of the additional interface, and will not be able to receive NTP traffic from that interface. Where the Ranger Gateway and an NTP server are co-located on the same server, it is recommended that the server be configured to start the Ranger Gateway before starting the NTP server when the server starts up. Similarly, where an RGVI client and an NTP server are co-located on the same server, it is recommended that the server be configured to start the RGVI client before starting the NTP server.
Chapter 26: Polling

ZoneRanger has the capability to monitor managed devices for status changes. ZoneRanger supports the following types of polling:

- ICMP polling (often called “ping”) to determine the status of IP interfaces
- SNMP querying of ifTable information (iOperStatus) of SNMP enabled nodes to determine the status of individual interfaces
- TCP polling of TCP ports on nodes to determine the status of processes listening on the ports

The Configuration > Polling page on the ZoneRanger provides the interface for polling configuration. By default, when a device is discovered, it is automatically configured to be polled for status changes. This behavior may be changed on the Configuration > Discovery page Options tab by unchecking Auto managing newly discovered nodes.

ICMP Polling

ZoneRanger uses ICMP polling to monitor the status of IP interfaces. By default, ZoneRanger polls interfaces on managed devices every 5 minutes (300 seconds). Using the Configuration > Polling page Interface Settings tab, different polling rates as well as ICMP timeouts and retries may be configured for nodes, specific IP interfaces, or groups of IP interfaces.

Whether or not ICMP polling is enabled or disabled for a particular IP interface may be configured on the Configuration > Polling page Enable/Disable tab. By choosing the appropriate device in the list, ICMP polling for each IP interface may be disabled by setting Poll Interface to None.

When ZoneRanger determines the status of an IP interface has changed, a tscZRIfUp or tscZRIfDown SNMP trap is generated as appropriate. If a device is determined to have changed state, a tscZRNodeUp or tscZRNodeDown trap will be generated. If at least one interface of a device whose previous status was Up is determined to be Down, a tscZRNodeMarginal trap will be generated.

ICMP Latency

As the ZoneRanger ICMP polls IP interfaces, the polling service caches the last ICMP round-trip time for each IP interface. This value may be retrieved for all IP addresses or specific IP addresses using the viewIcmpLatency command from a joined Ranger Gateway. The polling service only retains the last ICMP latency measurement for each polled IP interface.

SNMP Polling

For those interfaces which do not have an IP address, ZoneRanger can be configured to use SNMP polling to monitor the status. The SNMP status of an interface is determined by the SNMP querying of ifTable information (iOperStatus) of SNMP enabled nodes. By default, ZoneRanger does NOT automatically configure any interfaces to use SNMP polling. When SNMP polling is configured, ZoneRanger polls those interfaces on managed devices every 5 minutes (300 seconds). Using the Configuration > Polling page Interface Settings tab, different polling rates for nodes, specific interfaces, or groups of interfaces may be configured. If the interface is an SNMP interface without an IP address, the node name in the database should be used as the pattern.

Whether or not SNMP polling is enabled or disabled for a particular interface may be configured on the Configuration > Polling page Enable/Disable tab. By choosing the appropriate device in the list, SNMP polling for each interface may be enabled by setting Poll Interface to SNMP.
If the SNMP option under Poll Interface is unavailable for the interfaces on a device (i.e. there is no ifIndex listed), that means the device did not respond to an SNMP request when it was discovered. If the device is configured to support SNMP, verify the settings on the Configuration > SNMP page are correct for this device. Then use the Diagnostics > SNMP walk page to test whether or not ZoneRanger can use SNMP to query the device. Once ZoneRanger can successfully make SNMP requests to the device, the device needs to be rediscovered using the Configuration > Discovery page.

When ZoneRanger determines the status of an interface has changed, a tscZRIfUp or tscZRIfDown SNMP trap is generated as appropriate. If a device is determined to have changed state, a tscZRNodeUp or tscZRNodeDown trap will be generated. If at least one interface of a device whose previous status was Up is determined to be Down, a tscZRNodeMarginal trap will be generated.

**TCP Polling**

ZoneRanger uses TCP polling of TCP ports on nodes to determine the status of processes listening on those TCP ports. By default, ZoneRanger polls TCP ports on managed devices every 5 minutes (300 seconds). Use the Configuration > Polling page TCP Settings tab to configure different polling rates for an individual TCP Port as well as modifying the default TCP port polling rate.

Polling for a particular TCP port may be enabled or disabled using the Polling Enabled column on the Configuration > Polling page TCP Settings tab. When devices are discovered, the TCP ports listed on this tab will automatically be polled. If a Polling Enabled checkbox is unchecked, polling for the corresponding TCP port is disabled for all nodes, regardless of the per-node Poll TCP settings.

TCP port status propagation for specific services can be enabled or disabled for all nodes. If status propagation is enabled, TCP port status effects the overall status of nodes. Thus, if a TCP port does not, the TCP port will be in the Down state and the node will be in the Marginal state. When ZoneRanger determines the status of an TCP port has changed, a tscZRTcpUp, tscZRTcpBusy, tscZRTcpRefused, or tscZRTcpTimeout SNMP trap is generated as appropriate.

All of the TCP ports listed on the Configuration > Polling page TCP Settings tab, are configured on the Configuration > Discovery page TCP Ports tab. If a TCP is added or removed on the Configuration > Discovery page TCP Ports tab, that TCP port will be added or removed from the Configuration > Polling page TCP Settings tab.
Chapter 27: Root Cause

Though the use of a sophisticated root cause analysis service, ZoneRanger will automatically determine the root cause of a device outage. Depending on the network topology, a problem with one device, such as a router or switch, can affect status polling results for multiple devices. To address this problem, when ZoneRanger is triggered by status polling failures as described in Chapter 24, the root cause service determines which device is the root cause of the problem, and which devices are impacted by the root cause device.

The root cause service divides root cause analysis into two categories: IP (related to interface status polling failures) and TCP (related to TCP status polling failures). The Configuration > Root Cause page IP tab is used to configure the reporting of IP root causes and the Configuration > Root Cause page TCP tab is used to configure reporting TCP root causes.

After the polling service determines an IP interface has failed a status poll, the root cause service will enter a verification period to determine if the IP interface is no longer replying or if it was a transitory outage. The verification time period is configured on the Configuration > Root Cause page IP tab from the Advanced button. The default is to verify an outage using four ICMP requests over a two minute period.

Once the device is verified to have failed, a tscZRV erifyDown SNMP trap is generated. Then once the root cause is determined, the tscZRSourceDown SNMP trap is generated indicating the root cause of the outage. Using the Configuration > Root Cause page IP tab, ZoneRanger can be configured to send an email indicating the root cause either through a joined Ranger Gateway or directly from the ZoneRanger.

If the polling service determines additional IP devices fail to respond, whose outage was caused by a root cause device, a tscZRI nferredDown SNMP trap will be generated indicating this device is impacted by a root cause outage. The list of current root causes can be displayed using the View > Root Causes page on the ZoneRanger.

When a root cause device is successfully polled by the polling service, the root cause service verifies the device is responding and generates a tscZRVerifyUp SNMP trap. A tscZRSourceUp SNMP trap and optional email, if configured, are then generated indicating the root cause outage has been restored.

The root cause service describes root causes outages in terms of entities which are IP, TCP, or cloud. The IP entity refers to an IP address. The TCP entity refers to a TCP port. The cloud entity refers to an area of the network which the ZoneRanger could not determine any specific IP addressable information. One example of a cloud is a dumb network hub.
Chapter 28: SNMP Proxy

A Ranger Gateway and one or more joined ZoneRangers can provide an SNMP proxy service, enabling management applications to have SNMP access to devices in firewall-partitioned networks (e.g. DMZs) without requiring the firewall to be configured to allow SNMP traffic.

The following figure provides a high-level overview of an SNMP proxy transaction.

![Figure 28-1. ZoneRanger SNMP Proxy](image)

The basic steps of an SNMP proxy transaction are as follows:

1. A management application generates an SNMP Get or SNMP Set request, intended for a specific managed device.
2. The Ranger Gateway receives/intercepts the request.
3. The Ranger Gateway checks the Proxy Access Control configuration to verify that the request should be allowed, and to identify the proxy service to which the request should be forwarded (i.e. SNMP Proxy).
4. The SNMP Proxy service in the Ranger Gateway consults with the Proxy Map service in order to select a ZoneRanger that is able to relay the request to the target device.
5. The ZoneRanger forwards the request to the target device.
6. The target device generates a response and sends it to the requesting ZoneRanger.
7. The ZoneRanger forwards the response to the Ranger Gateway.
8. The Ranger Gateway forwards the response to the management application.

Management applications can access the SNMP Proxy service in a variety of ways, as described in the following sections.

GVI/RGVI

When using GVI or RGVI, the management application sends SNMP requests intended for a managed device to the actual address of the target device, or an address that can be uniquely mapped to the target device. The management application server is configured with static routing rules, so that traffic destined for devices located in firewall-partitioned networks is routed to a virtual interface, which then forwards the traffic to the Ranger Gateway.

Consider the network example in the following figure. Two DMZ’s are shown. The first DMZ has one ZoneRanger (ZR-1) and the second one has two (ZR-2, ZR-3). The IP addresses in the two DMZ’s do not overlap.
The messaging flow for an SNMP proxy request using GVI is illustrated in the following figure.

Note the following from this example:

- The management application requests that a UDP datagram containing the SNMP GetRequest message be sent to the address of the target device (10.4.1.2) [1].
- The routing table in the management application server is preconfigured to route traffic destined for the 10.4.1.2 address to the GVI driver.
- The GVI driver forwards the request to the Ranger Gateway [2], which checks the Proxy Access Control configuration to verify that the request should be allowed and to identify the proxy service to which the request should be forwarded (i.e. SNMP Proxy).
• The SNMP Proxy service consults the Proxy Map service in the Ranger Gateway to determine the list of ZoneRangers that manage the target device (ZR-2, and ZR-3). One of the ZoneRangers (ZR-2) is selected, and the request is forwarded to the selected ZoneRanger [3].

• The selected ZoneRanger forwards the request to the target device [4].

• The target device replies back to the ZoneRanger [5], which relays the response to the Ranger Gateway [6]. The SNMP Proxy service relays the response to the GVI driver [7].

• The GVI driver forwards the response to the management application [8].

The primary advantage of GVI and RGVI is that the existence of the SNMP proxy is completely transparent to the management application. Common routing mechanisms within the underlying operating system are used to intercept traffic bound for devices in firewall-partitioned networks, so there is no need to modify or reconfigure the management application in any way. Another advantage is that the same mechanism can be used for other proxy services, such as ICMP proxy, or TCP proxy.

SOCKS

SOCKS is a standard protocol for generic TCP and UDP proxy services that can be used to redirect management traffic from the management application to a SOCKS server integrated within the Ranger Gateway. In order to use SOCKS, either the management application must include built-in support for SOCKS, or generic SOCKS “shim” software must be installed on the management application server. The shim software inserts itself between the management application and the server’s TCP/IP stack, and redirects traffic for specified IP addresses and ports to a SOCKS server, based on configuration information.

When using the SOCKS mechanism, the management application sends SNMP requests intended for a managed device to the actual address of the target device, or an address that can be uniquely mapped to the target device. The built-in SOCKS client or the SOCKS shim redirects these requests to the SOCKS server integrated within the Ranger Gateway.
The following figure shows a SOCKS shim inserted between the management application and the operating system.

Figure 28-3. ZoneRanger SNMP Proxy with SOCKS

The messaging flow for an SNMP proxy request using a SOCKS shim is illustrated in the following figure.

Note the following from this example:

- The management application requests that a UDP datagram containing the SNMP Get request message be sent to the address of the target device (10.4.1.2) [1].
• The SOCKS shim intercepts the request, performs a SOCKS protocol handshake with the
  SOCKS server in the Ranger Gateway, to establish a “UDP association” [2, 3], then
  forwards the SNMP request, to the SOCKS server, along with a header indicating that the
datagram is intended for address 10.4.1.2 [4].

• The SOCKS server in the Ranger Gateway checks the Proxy Access Control
  configuration to verify that the request should be allowed and to identify the proxy
  service to which the request should be forwarded (i.e. SNMP Proxy).

• The SNMP Proxy service consults the Proxy Map service to determine the list of
  ZoneRangers that manage the target device (ZR-2, and ZR-3). One of the ZoneRangers
  (ZR-2) is selected, and the request is forwarded to the selected ZoneRanger [5].

• The selected ZoneRanger forwards the request to the target device [6].

• The target device replies back to the ZoneRanger [7], which relays the response to the
  Ranger Gateway [8]. The SNMP Proxy service relays the response to the SOCKS server,
  which forwards the response to the SOCKS shim along with a header indicating that the
  response was received from 10.4.1.2 [9].

• The SOCKS shim forwards the response to the management application [10].

One advantage of SOCKS over GVI/RGVI is that it is typically possible to configure the SOCKS
client to route traffic for certain ports to the Ranger Gateway, while traffic destined for other ports
is routed normally. In addition, some SOCKS clients can be configured to only intercept traffic
sent from specified applications. A disadvantage of SOCKS is that many management
applications do not provide built-in support for SOCKS and reliable SOCKS shims may not be
available for the operating system being used. In these cases, an alternative SNMP proxy access
mechanism will need to be selected.

**IP Address Aliasing**

Most operating systems provide a means to associate multiple IP addresses with each network
interface (i.e. a primary address, and one or more “aliases”). If IP address aliases, corresponding
to managed devices located in firewall-partitioned networks, are defined on the management
application server, all traffic generated by the management application and destined for these
devices will be routed as local traffic to the interface where the IP address aliases have been
defined. If an SNMP proxy port has been configured, the SNMP Proxy service on the Ranger
Gateway will listen on that port for requests destined for any of these IP addresses. As a result,
when a management application sends an SNMP request intended for a managed device to one of
the configured alias addresses, with the destination port set to the configured SNMP proxy port,
the Ranger Gateway will receive the request.

If the management application and the Ranger Gateway software have been installed on the same
server, the IP address aliases can usually be added to the server’s loopback interface. In such
cases, it may be possible to configure the IP address aliases for managed devices to be the same as
the actual IP addresses of those devices. If the management application and the Ranger Gateway
software have been installed on different servers, the IP address aliases must be added to an
appropriate network interface on the Ranger Gateway server, and static routes will need to be
defined on the management application server to ensure that SNMP requests are routed to the
Ranger Gateway server.
Consider the network example in the following figure. In this example, the management application and the Ranger Gateway have been installed on the same server. Two DMZ’s are shown. The first DMZ has one ZoneRanger (ZR-1) and the second one has two (ZR-2, ZR-3). The IP addresses in the two DMZ’s do not overlap.

Figure 28-4. ZoneRanger SNMP Proxy with IP Aliasing

In order to manage this network, addresses 10.2.1.1, 10.2.1.2, 10.4.1.1, 10.4.1.2, and 10.4.1.3 would be configured as IP address aliases on the management application server, corresponding to the five managed devices, and the Proxy Map service in the Ranger Gateway would be configured as shown in the figure. An example of the messaging flow for an SNMP proxy request is shown in the following figure.

Note the following from this example:
• The management application requests that a UDP datagram containing the SNMP Get request message be sent to the address of the target device (10.4.1.2), using the destination SNMP proxy port [1]. Assuming that the specified destination IP address has been defined as an IP address alias on the management application server, the request will be delivered to the SNMP Proxy service within the Ranger Gateway.

• The SNMP Proxy service in the Ranger Gateway will check the Proxy Access Control configuration to verify that the request should be allowed, then will consult the Proxy Map service to determine the list of ZoneRangers that manage the target device (ZR-2, and ZR-3). One of the ZoneRangers (ZR-2) is selected, and the request is forwarded to the selected ZoneRanger [2].

• The selected ZoneRanger forwards the request to the target device [3].

• The target device replies back to the ZoneRanger [4], which relays the response to the Ranger Gateway [5].

• The Ranger Gateway forwards the response to the management application [6].

In some cases it may be necessary to configure the SNMP proxy service to use a non-standard port value in order to avoid conflict with an SNMP agent on the Ranger Gateway server that may be listening on port 161. The port that the ZoneRanger will use to present the request to the managed device can be configured on a per-device basis. This allows different managed devices in the same firewall-partitioned network to listen for SNMP requests on different ports. By default, the ZoneRanger will forward SNMP requests to destination port 161.

IP address aliasing can be used on all operating systems where the Ranger Gateway software is supported. The main disadvantage of the IP address aliasing technique is the administrative effort required to add and maintain IP address aliases for all managed devices on the Ranger Gateway server. Another concern is that operating systems may limit the number of IP address aliases that can be defined. As a result, this technique may not be able to support the required number of managed devices for some applications.

Community String Conventions

In all of the previously described mechanisms, the Ranger Gateway determines the address of the target device for each SNMP request based on the address to which the management application sent the request. An alternative is to configure the management application to send the SNMP request to an arbitrary IP address on the Ranger Gateway server, and for the SNMP Proxy service within the Ranger Gateway to determine the target device address based on additional information embedded into the SNMP request's community string, according to specified conventions. For example, the following community string format can be used:

    community@ZoneRanger@device

where community is the actual community string that the target device is expecting (e.g. public), ZoneRanger is the name or IP address of a ZoneRanger that is managing the target device, and device is the name or IP address of the target device. In this case, the Ranger Gateway would extract the ZoneRanger and device values from the community string, and would forward the request to the specified ZoneRanger. The ZoneRanger would then send the request to the target device. Note that the ZoneRanger and device values are removed before the request is forwarded to the target device, so the target device only sees the community value that it is expecting.
Consider the network example in the following figure. In this example, the management application and the Ranger Gateway have been installed on different servers. Two DMZ’s are shown. The first DMZ has one ZoneRanger (ZR-1) and the second one has two (ZR-2, ZR-3). The IP addresses in the two DMZ’s do not overlap. The IP address of the Ranger Gateway Server is 10.254.1.1.

**Figure 28-5. ZoneRanger SNMP Proxy with Community String Conventions**

An example of the messaging flow for an SNMP proxy request is shown in the following figure.

---

9 The community string conventions mechanism can also be used in cases where the Ranger Gateway is installed on the management application server. In such cases, the management application would need to send SNMP requests to a local IP address.
Note the following from this example:

- The management application directs the SNMP request to the Ranger Gateway's IP address (10.254.1.1), using the SNMP proxy port [1]. The ZoneRanger to which the request should be forwarded (ZR-2), and the target device’s actual IP address (10.4.1.2) are embedded in the community string, along with the community string value that the target device expects (i.e. public).

- The SNMP Proxy service in the Ranger Gateway receives the request, parses the community string, verifies that the request should be allowed, then forwards the request to the specified ZoneRanger [2].

- The ZoneRanger forwards the request to the target device [3], with the ZoneRanger and device portions of the community string removed.

- The target device replies back to the ZoneRanger [4], which relays the response to the Ranger Gateway [5].

- The Ranger Gateway forwards the response to the management application [6].

The SNMP Proxy service can be configured to use different community string formats. The following formats are supported:

1. community@ZoneRanger@device
2. device@ZoneRanger@community
3. community@device
4. device@community
5. community

Formats 1 and 2 require management applications to specify the ZoneRanger (or ZoneRanger group) that will relay the SNMP request to the target device. When using grouping, the ZoneRanger field in the community string can be replaced with a group name that identifies a group of ZoneRangers, and the SNMP Proxy service in the Ranger Gateway will automatically select a ZoneRanger from this group to relay the request. The only difference between formats 1 and 2 is the order of the fields. The ability to configure the SNMP Proxy service to use different field orders has been provided in order to handle situations where management applications and managed devices are using their own community string prefix or suffix conventions.

Formats 3 and 4 do not require the management application to specify a ZoneRanger. Instead, the SNMP Proxy service consults the Proxy Map service, in order to identify a ZoneRanger that is able to relay traffic to the target device, and then forwards the SNMP request to the selected ZoneRanger. The only difference between formats 3 and 4 is the order of the fields. The ability to use different field orders has been provided in case management applications and managed devices are using their own community string prefix or suffix conventions.

When the Proxy Map service is used, the responsibility for identifying the ZoneRanger to relay each request is essentially moved from the management application to the Ranger Gateway. The advantages of this approach are:

- Associations between ZoneRangers and DMZ devices, and any required address translations for DMZ devices (e.g. if NAT is in effect) are configured in one place, and can be shared by multiple proxy services across multiple management applications.

- The Proxy Map service can be configured to balance proxy requests across a set of ZoneRanger candidates, resulting in a more even distribution of proxy traffic in situations where DMZ devices are being managed by multiple ZoneRangers.
The following figure shows a message flow example, based on the previous sample network, using the `community@device` convention.

![Message Flow Diagram]

Note the following from this example:

- The management application directs the SNMP request to the Ranger Gateway's IP address (10.254.1.1), using the SNMP proxy port [1]. The target device’s actual IP address (10.4.1.2) is embedded in the community string, along with the community string value that the target device expects (e.g. public).

- The SNMP Proxy service verifies that the request should be allowed, then consults the Proxy Map service in the Ranger Gateway to determine the list of ZoneRangers that manage the target device (ZR-2, and ZR-3). One of the ZoneRangers (ZR-2) is selected, and the request is forwarded to the selected ZoneRanger [2].

- The selected ZoneRanger uses the actual IP address of the target device (10.4.1.2) to forward the request to the target device [3], with the `device` portion of the community string removed.

- The target device replies back to the ZoneRanger [4], which relays the response to the Ranger Gateway [5].

- The Ranger Gateway forwards the response to the management application [6].

Format 5 implies that no special information is embedded in the community string, and is used in conjunction with the IP address aliasing mechanism.

In some cases it may be necessary to configure the SNMP proxy service to use a non-standard port value in order to avoid conflict with an SNMP agent on the Ranger Gateway server that may be listening on port 161. The port that the ZoneRanger will use to present the request to the managed device can be configured on a per-device basis. This allows different managed devices in the same firewall-partitioned network to listen for SNMP requests on different ports. By default, the ZoneRanger will forward SNMP requests to destination port 161. When community string conventions are being used, the management application can optionally override the configured port for a given device, by adding `":port"` to the `device` part of the community string, where `port` is the desired port number.

When community string conventions are used, a simplified form of Proxy Access Control is used to determine whether or not requests should be allowed. Instead of using the portMap and portConfig tables, as described in Chapter 14, the SNMP Proxy service simply verifies that the source address associated with the request matches the configured `SNMP proxy client address`. The `SNMP proxy client address` can be configured using the Ranger Gateway Viewer or by using the `configGateway` command.
Community string conventions are best suited for management applications that can be configured to send SNMP requests for all managed devices to a single address\(^{10}\). The primary advantage of community string conventions is that there is no need to install a GVI driver, an RGVI client, or a SOCKS shim on the management application server. The three-part community string format (e.g. `community@ZoneRanger@device`) is also useful when managing networks with overlapping addresses. The primary disadvantage is that the management application must be configured in an atypical way in order to use the proxy. Some management applications require unique addresses for each managed device, and do not support the concept of a common proxy address. In these cases, an alternative SNMP proxy mechanism will need to be selected.

### SNMPv3 Conversion

The ZoneRanger SNMP Proxy service can be used to proxy SNMPv1 and SNMPv2c requests to managed devices. In addition, ZoneRanger can be configured to translate SNMPv1 or SNMPv2c requests to SNMPv3 requests, as illustrated in the following figure.

![SNMPv3 Conversion Diagram](image)

This feature enables authentication and encryption of SNMP messages in firewall-partitioned networks, such as a DMZ, where enhanced security is arguably most needed, while avoiding the need to configure or upgrade existing management applications to support SNMPv3. SNMPv3 conversion can be configured on a per-device basis, so that the additional administrative effort required for SNMPv3 can be limited only to those devices where security is most needed.

It is recommended that SNMPv3 users change the authentication and encryption passwords associated with management devices on a regular basis. To facilitate this, the ZoneRanger web interface includes a tool for automatically updating SNMPv3 passwords on managed devices. This tool is located on the **Administration > SNMP** page **SNMPv3 Passwords** tab of the ZoneRanger web interface.

---

\(^{10}\) Community string conventions can also be used when the management application uses different addresses for different target devices. However, the GVI/RGVI, SOCKS, or IP address aliasing mechanisms are likely to be preferred in such cases, because the need to configure special community strings for each device is eliminated.
Chapter 29: TCP Proxy

A Ranger Gateway and one or more joined ZoneRangers can provide a TCP proxy service, enabling management applications to establish TCP connections to devices located in firewall-partitioned networks, without requiring the firewall to be configured to allow TCP connections.

The following figure provides a high-level overview of a TCP proxy transaction.

![TCP Proxy Diagram](image)

The TCP proxy service is intended for use only in cases where the application protocol being carried over the TCP connection is not supported by one of the more specific TCP-based proxy services (e.g. Telnet, SSH, HTTP, HTTPS). Given that the application protocol being used is not identified, the TCP proxy service is unable to perform any application layer protocol screening or filtering. As such, TCP proxy is disabled by default, and should only be enabled for those devices/ports where it is absolutely needed.

Management applications can access the TCP Proxy service in a variety of ways, as described in the following sections.

**GVI/RGVI**

When using GVI or RGVI, the management application sends TCP connection requests intended for a managed device to the actual address of the target device, or an address that can be uniquely mapped to the target device. The management application server is configured with static routing rules, so that traffic destined for devices located in firewall-partitioned networks is routed to a virtual interface, which then forwards the traffic to the Ranger Gateway.

When the Ranger Gateway receives the TCP connection request, it will check the Proxy Access Control configuration to verify that the request should be allowed, then will consult the Proxy Map service in order to identify a ZoneRanger that is able to relay the request to the target device. The request is forwarded to the selected ZoneRanger, which in turn, establishes a TCP connection to the target device. Once this TCP connection is established, the ZoneRanger will inform the Ranger Gateway, and the Ranger Gateway will complete the establishment of the initial TCP connection (i.e. the connection between the management application and the Ranger Gateway). From this point on, the Ranger Gateway and selected ZoneRanger will relay data between the management application’s TCP connection to the Ranger Gateway and the ZoneRanger’s TCP connection to the target device, until one of the connections is disconnected.
The primary advantage of GVI and RGVI is that the existence of the TCP proxy is completely transparent to the management application. Common routing mechanisms within the underlying operating system are used to intercept traffic bound for devices in firewall-partitioned networks, so there is no need to modify or reconfigure the management application in any way. Another advantage is that the same mechanism can be used for other proxy services, such as ICMP proxy, or SNMP proxy.

**SOCKS**

SOCKS is a standard protocol for generic TCP and UDP proxy services that can be used to redirect management traffic from the management application to a SOCKS server integrated within the Ranger Gateway. In order to use SOCKS, either the management application must include built-in support for SOCKS, or generic SOCKS “shim” software must be installed on the management application server. The shim software inserts itself between the management application and the server’s TCP/IP stack, and redirects traffic for specified IP addresses and ports to a SOCKS server, based on configuration information.

In order to access a managed device through TCP proxy, a SOCKS-aware web browser initially establishes a TCP connection to the SOCKS port (by default, 4855) on the Ranger Gateway. After this connection is established, the client application sends a SOCKS connection request to the Ranger Gateway, indicating the managed device and port to which the client would like to connect.

The SOCKS server on the Ranger Gateway checks the Proxy Access Control configuration to verify that the request should be allowed, then consults the Proxy Map service to identify a ZoneRanger that is able to proxy traffic to the target device, and to translate the target address, if necessary. The request is then forwarded to the selected ZoneRanger, which attempts to connect to the target device. If this connection is successfully established, the ZoneRanger notifies the Ranger Gateway, which in turn notifies the management application.

From this point, the Ranger Gateway and selected ZoneRanger simply relay data between the client application’s TCP connection to the Ranger Gateway and the ZoneRanger’s TCP connection to the target device. The Ranger Gateway and ZoneRanger continue to relay data until one of the connections is disconnected.

Most web browsers support the SOCKS protocol. If a SOCKS-enabled web browser is not available, you can use SOCKS “shim” software, which effectively inserts itself between the client application and the networking layer on the host where the client application is running, and redirects connection requests to a configured SOCKS server.
Chapter 30: Telnet/SSH Proxy

A Ranger Gateway and one or more joined ZoneRangers can provide an Telnet and SSH proxy service, enabling Telnet and/or SSH client applications to have command line access to devices located in firewall-partitioned networks, without requiring the firewall to be configured to pass Telnet or SSH traffic.

The following figure provides a high-level overview of a Telnet/SSH proxy transaction. Note that the Management Application Server in this figure is acting as a Telnet/SSH client, and one or more managed devices may act as Telnet/SSH servers.

Telnet/SSH clients can range from simple command-line tools, to configuration management or security management applications that use Telnet or SSH to communicate with managed devices. In addition to using Telnet/SSH proxy to communicate with managed devices, the Telnet and SSH proxy services can also be used to access the ZoneRanger text interface for joined ZoneRangers.

While the ZoneRanger is able to proxy both Telnet and SSH protocols, SSH will typically be the preferred protocol for most applications, because the Telnet protocol, which exchanges user ID and password information over an unencrypted TCP connection, is less secure. As a result, SSH proxy is enabled by default and Telnet is disabled by default for managed devices.

Management applications can access Telnet and SSH Proxy services in a variety of ways, as described in the following sections.

GVI/RGVI

When using GVI or RGVI, the management application sends Telnet or SSH requests intended for a managed device to the actual address of the target device, or an address that can be uniquely mapped to the target device. The management application server is configured with static routing rules, so that traffic destined for devices located in firewall-partitioned networks is routed to a virtual interface, which then forwards the traffic to the Ranger Gateway.
When the Ranger Gateway receives the initial TCP connection request for a Telnet or SSH connection, it will check the Proxy Access Control configuration to verify that the request should be allowed, then will consult the Proxy Map service in order to identify a ZoneRanger that is able to relay the request to the target device. The request is then forwarded to the selected ZoneRanger, which in turn, establishes a TCP connection to the target device. Once this TCP connection is established, the ZoneRanger will inform the Ranger Gateway, and the Ranger Gateway will complete the establishment of the initial TCP connection (i.e. the connection between the management application and the Ranger Gateway). From this point on, the Ranger Gateway and selected ZoneRanger will relay Telnet or SSH data between the management application’s TCP connection to the Ranger Gateway and the ZoneRanger’s TCP connection to the target device, until one of the connections is disconnected.

The following figure illustrates how PuTTY, a freely available Telnet/SSH client application, can be used to establish an SSH session with a managed device via GVI or RGVI.

Figure 30-2. Ranger Gateway with PUTTY with GVI or RGVI

The steps to establish a connection, in this case are:

1. Enter the IP address or hostname of the target device.
2. Select the protocol to be used (SSH in this example).
3. Verify the port value. PuTTY will automatically set this value based on the selected protocol. You may need to modify this value if the target device uses a non-standard port for SSH or Telnet.
4. Click the Open button to establish the SSH session.
The primary advantage of GVI and RGVI is that the existence of the Telnet/SSH proxy is completely transparent to the management application. Common routing mechanisms within the underlying operating system are used to intercept traffic bound for devices in firewall-partitioned networks, so there is no need to modify or reconfigure the management application in any way. Another advantage is that the same mechanism can be used for other proxy services, such as ICMP proxy, or SNMP proxy.

**SOCKS**

SOCKS is a standard protocol for generic TCP and UDP proxy services that can be used to redirect management traffic from the management application to a SOCKS server integrated within the Ranger Gateway. In order to use SOCKS, either the management application must include built-in support for SOCKS, or generic SOCKS “shim” software must be installed on the management application server. The shim software inserts itself between the management application and the server’s TCP/IP stack, and redirects traffic for specified IP addresses and ports to a SOCKS server, based on configuration information.

In order to access a managed device through Telnet or SSH proxy, a SOCKS-aware Telnet/SSH client application can initially establishes a TCP connection to the SOCKS port (by default, 4855) on the Ranger Gateway. After this connection is established, the client application sends a SOCKS connection request to the Ranger Gateway, indicating the DMZ device and port to which the client would like to connect.

The SOCKS server on the Ranger Gateway will check the Proxy Access Control configuration to verify that the request should be allowed, then will consult the Proxy Map service to identify a ZoneRanger that is able to proxy traffic to the target device, and to translate the target address, if necessary. The request is then forwarded to the selected ZoneRanger, which attempts to connect to the target device. If this connection is successfully established, the ZoneRanger notifies the Ranger Gateway, which in turn notifies the Telnet/SSH client.

From this point, the Ranger Gateway and selected ZoneRanger simply relay data between the client application’s TCP connection to the Ranger Gateway and the ZoneRanger’s TCP connection to the target device, allowing the SSH client and target device to establish a Telnet or SSH session. The Ranger Gateway and ZoneRanger continue to relay data until one of the connections is disconnected.

Most commercial Telnet/SSH client applications support the SOCKS protocol. If a SOCKS-enabled SSH client is not available, you can use SOCKS “shim” software, which effectively inserts itself between the client application and the networking layer on the host where the client application is running, and redirects connection requests to a configured SOCKS server.

The following figures how PuTTY can be used to establish an SSH session with a managed device through the SOCKS server in the Ranger Gateway.
Figure 30-3. Ranger Gateway with PUTTY with SOCKS

The steps to establish a connection, in this case are:

1. Enter the IP address or hostname of the target device.
2. Select the protocol to be used (SSH in this example).
3. Verify the port value. PuTTY will automatically set this value based on the selected protocol. You may need to modify this value if the target device uses a non-standard port for SSH or Telnet.
4. Click Proxy in the Category pane on the left hand side of the window. The following page is displayed.
5. Specify the **Proxy type** (SOCKS 5, for this example), **Proxy hostname**, which should be the host name or IP address of the Ranger Gateway (for example, mygateway.company.com), and the **Port**, which should be the SOCKS server port on the Ranger Gateway (for example, 4855).

6. Depending on your DNS configuration, you may need to change the **Do DNS name lookup at proxy end** setting. The **Username** and **Password** fields should be left blank.

7. Click the Open button to establish the SSH session.

One advantage of SOCKS over GVI/RGVI is that it is typically possible to configure the SOCKS client to route traffic for certain ports to the Ranger Gateway, while traffic destined for other ports is routed normally. In addition, some SOCKS clients can be configured to only intercept traffic sent from specified applications. A disadvantage of SOCKS is that many management applications do not provide built-in support for SOCKS and reliable SOCKS shims may not be available for the operating system being used. In these cases, an alternative Telnet/SSH proxy access mechanism will need to be selected.
IP Address Aliasing

Most operating systems provide a means to associate multiple IP addresses with each network interface (i.e. a primary address, and one or more “aliases”). If IP address aliases, corresponding to managed devices located in firewall-partitioned networks, are defined on the management application server, all traffic generated by the management application and destined for these devices will be routed as local traffic to the interface where the IP address aliases have been defined. If an SSH proxy port has been configured, the SSH Proxy service on the Ranger Gateway will listen on that port for requests destined for any of these IP addresses. As a result, when a management application sends a SSH request intended for a managed device to one of the configured alias addresses, with the destination port set to the configured SSH proxy port, the Ranger Gateway will receive the request.

If the management application and the Ranger Gateway software have been installed on the same server, the IP address aliases can usually be added to the server’s loopback interface. In such cases, it may be possible to configure the IP address aliases for managed devices to be the same as the actual IP addresses of those devices. If the management application and the Ranger Gateway software have been installed on different servers, the IP address aliases must be added to an appropriate network interface on the Ranger Gateway server, and static routes will need to be defined on the management application server to ensure that traffic related to SSH session requests is routed to the Ranger Gateway server.

The SSH proxy port can be configured using the `configGateway` command or the Ranger Gateway Viewer > Gateway Settings window. Note that by default, this feature is disabled and the SSH proxy port is undefined. In addition, an SSH proxy destination port must be defined to indicate the port on managed devices that the ZoneRanger should use to establish SSH proxy sessions. The SSH proxy destination port can be defined using the `configGateway` command or the Ranger Gateway Viewer > Gateway Settings window. The default value is 22. Note that the IP address aliasing mechanism does not support Telnet proxy.

To access a managed device using SSH proxy, an SSH client application would establish a TCP connection to the IP address on the Ranger Gateway that is associated with the target device, specifying the configured SSH proxy port as the destination port. After this connection is established, the Ranger Gateway will check the Proxy Access Control configuration to verify that the request should be allowed, then will consult the Proxy Map service to identify the target device, and to select a ZoneRanger that is able to proxy traffic to the target device. The connection request is then forwarded to the selected ZoneRanger, which attempts to connect to the target device.

If this connection is successfully established, the ZoneRanger notifies the Ranger Gateway. From this point, the Ranger Gateway and selected ZoneRanger simply relay data between the client application’s TCP connection to the Ranger Gateway and the ZoneRanger’s TCP connection to the target device, enabling the SSH client and target device to establish an SSH session. The Ranger Gateway and ZoneRanger continue to relay data until one of the connections is disconnected.
The following figure illustrates how to use PuTTY to establish an SSH session with a managed device, using the SSH proxy port on the Ranger Gateway.

![PuTTY Configuration](image)

**Figure 30-5. Ranger Gateway with PUTTY with IP Address Aliasing**

The steps to establish a connection, in this case are:

1. Enter the IP address or hostname of the target device (64.10.37.5 in this case).
2. Select the protocol to be used (SSH in this example).
3. Enter the configured SSH proxy port value (4822 in this example) in the Port field.
4. Click the Open button to establish the SSH session.

The IP address aliasing approach for SSH proxy has the following advantages:

- It can be used in cases where the Ranger Gateway software is installed on a server with an operating system that does not support GVI.
- It does not require a SOCKS-enabled SSH client or SOCKS shim.

The main disadvantage of the IP address aliasing technique is the administrative effort required to add and maintain IP address aliases for all managed devices on the Ranger Gateway server. Another concern is that operating systems may limit the number of IP address aliases that can be defined. As a result, this technique may not be able to support the required number of managed devices for some applications.
Dedicated Telnet/SSH Ports

When a ZoneRanger is joined to a Ranger Gateway, the Ranger Gateway allocates dedicated ports that can be used to access various services, HTTP, HTTPS, SQL, Telnet, and SSH, on the newly joined ZoneRanger. You can use the `listTcpPorts` command on the Ranger Gateway command interface or the Ranger Gateway Viewer to identify the ports that have been allocated for each ZoneRanger.

A Telnet/SSH client application can establish a proxy connection to a joined ZoneRanger simply by connecting to the Ranger Gateway’s address, specifying the dedicated Telnet or SSH port associated with that ZoneRanger as the destination port.

The following figure illustrates how to use PuTTY to establish an SSH session with a joined ZoneRanger, using that ZoneRanger’s dedicated SSH proxy port on the Ranger Gateway.

![PuTTY Configuration](image)

*Figure 30-6. Ranger Gateway with PuTTY with dedicated ports*

The steps to establish a connection, in this case are:

3. Enter the IP address or hostname of the target device (`mygateway.company.com`, in this example).
4. Select the protocol to be used (SSH in this example).
5. Enter the configured SSH proxy port value (20004 in this example) in the **Port** field.
6. Click the Open button to establish the SSH session.
Note that dedicated ports can be used only to access to a ZoneRanger text interface. Dedicated ports cannot be used to access other managed devices. A significant disadvantage with SSH proxy using dedicated Ranger Gateway ports is that the same destination address (the Ranger Gateway’s host name or IP address) can be used to establish SSH sessions with different ZoneRangers, which typically confuses SSH clients that are configured to verify host keys. When an SSH client is configured to verify host keys, it typically maintains a table that associates host addresses with their corresponding SSH host keys. Assuming you have not already used SSH to access the Ranger Gateway itself, the first time you access a ZoneRanger using a Ranger Gateway dedicated port, an entry is created associating the Ranger Gateway address with the ZoneRanger host key. If you then try to access a different ZoneRanger, the SSH client will notice that the new ZoneRanger host key does not match the saved value, and might conclude that the new ZoneRanger is masquerading as the old one.

The only solutions to this problem are to configure the SSH client to ignore this condition or to use a different form of SSH proxy access. Note that host keys are used to protect against Man-In-The-Middle attacks, so before deciding to disable or relax host key verification, you will need to ensure that your company’s security requirements will not be compromised.
Chapter 31: TACACS+/RADIUS Proxy

A ZoneRanger and one or more joined Ranger Gateways can provide a TACACS+ and/or RADIUS proxy service, allowing devices located in a firewall-partitioned network zone to make TACACS+ and/or RADIUS Authentication, Authorization and Accounting (a.k.a. AAA) requests to a ZoneRanger, which forwards the requests to a Ranger Gateway, which in turn forwards the requests to a TACACS+ and/or RADIUS server. Replies from the server follow the reverse path through the Ranger Gateway, the ZoneRanger, and back to the device that made the initial request.

The following figure provides a high-level overview of a TACACS+/RADIUS proxy transaction.

The TACACS+/RADIUS proxy enables devices located in firewall-partitioned networks to use TACACS+ and/or RADIUS services, without requiring the firewall to be configured to pass TACACS+ or RADIUS messages. You can also use the TACACS+/RADIUS proxy to control access to the ZoneRanger itself.

Configuring TACACS+/RADIUS Proxy on a ZoneRanger

In order for the ZoneRanger to be able to proxy TACACS+ and/or RADIUS traffic, it must be joined to one or more Ranger Gateways, and one or more server groups must be defined. A server group is a named set of TACACS+/RADIUS server entries, each of which contains the following information:

- The joined Ranger Gateway to be used to relay traffic to a given TACACS+/RADIUS server.
- The host name or IP address of the TACACS+/RADIUS server.
- The TACACS+ port on the given server.
- The RADIUS authentication and accounting ports on the given server.

Server groups can be configured with multiple entries, in order to provide high availability. Multiple server groups can be defined, allowing TACACS+/RADIUS traffic for different devices to be routed to different groups of servers. Once a set of server groups has been defined, proxy rules must be configured for each protocol, associating managed devices, or groups of managed devices, with the server group that should be used for those devices. Each proxy rule associates an IP address, or range of IP addresses, with a server group name. Separate proxy rule tables are provided for TACACS+ and RADIUS. As an example, the simplest possible configuration would be as follows:

- Define a single server group named “MyServerGroup”
- Add the following proxy rule to the TACACS+ table:
Add the following proxy rule to the RADIUS table:

```
.*.*.* *MyServerGroup
```

Using this configuration, the ZoneRanger will select a server from the MyServerGroup group to handle TACACS+ and RADIUS requests from all managed devices. In order to configure a second server group to handle requests originated by specific devices, the following steps would be required:

- Define a new server group (e.g. “MyOtherServerGroup”)
- Insert proxy rules for the specific IP addresses or IP address ranges to the top of the TACACS+ table:

  
  10.254.1.1 MyOtherServerGroup
  10.254.2.10-20 MyOtherServerGroup
  .*.*.* *MyServerGroup

- Insert proxy rules for the specific IP addresses or IP address ranges to the top of the RADIUS table:

  
  10.254.1.1 MyOtherServerGroup
  10.254.2.10-20 MyOtherServerGroup
  .*.*.* *MyServerGroup

When handling a TACACS+ or RADIUS request from a given device, the ZoneRanger will search through the proxy rules table associated with the protocol being used for the first rule that matches the requesting device’s address. As such, it is important to ensure that specific address rules are placed ahead of overlapping range or wild-card rules.

Server Groups are configured on the Configuration > Access Control page Server Groups tab on the ZoneRanger web interface. Proxy rules for TACACS+ and RADIUS are configured on the TACACS+ and RADIUS tabs.

**Configuring ZoneRanger to use TACACS+/RADIUS**

It is also possible to configure the ZoneRanger to use TACACS+ or RADIUS to authenticate and authorize access to the ZoneRanger web and text interfaces. In effect, the ZoneRanger acts as a TACACS+ or RADIUS client, using its own proxy service to relay authentication and authorization requests to a configured server group. TACACS+ can be enabled and configured on the TACACS+ tab, and RADIUS can be enabled and configured from the RADIUS tab. Note that enabling both TACACS+ and RADIUS at the same time is not allowed.

If the ZoneRanger is configured to use RADIUS, you will need to specify the server group to be used for ZoneRanger authentication and authorization requests. If the ZoneRanger is configured to use TACACS+, you can specify the server group to be used, the authentication login type (ASCII or PAP), privilege levels associated with admin and operator status, and the service and protocol arguments to be used in the authorization process.

If the ZoneRanger has been configured to use TACACS+ or RADIUS, and an authentication request is rejected by the configured TACACS+ or RADIUS server, the ZoneRanger will check to see if the specified user name and password match a locally configured user (see the Configuration > Access Control page Users tab on the ZoneRanger web interface).
In order to configure the ZoneRanger to authenticate with Windows IAS, a specific Resource Policy must be added in IAS. The Resource Policy must have a Policy condition where Service-Type matches “Authenticate Only”. An Attribute needs to be added to the Profile that matches the following:

- **Attribute name:** Vendor-Specific
- **Attribute number:** 26
- **Attribute format:** OctetString
- **Attribute values:**
  - **Vendor:** Vendor code: 2668
  - **Value:** admin

In order to configure the ZoneRanger to authenticate with FreeRadius, the following needs to be added to the FreeRadius dictionary:

```
VENDOR
BEGIN-VENDOR
Tavve
ATTRIBUTE
SecurityLevel 1 string
END-VENDOR
```

**Configuring TACACS+/RADIUS Proxy on a Ranger Gateway**

The Ranger Gateway can be configured to interact with a TACACS+ or RADIUS server in a variety of ways. Where possible, the most convenient method is to install the Ranger Gateway software on the same server where the TACACS+/RADIUS server application has been installed. In this case, the Ranger Gateway can optionally be configured to spoof the source address in requests forwarded to the TACACS+/RADIUS server, so that these requests appear to be coming directly from the managed device. This is an important feature, because TACACS+/RADIUS servers typically can be configured so that users have different privileges on different devices, and the source address in the request is used to identify the device being accessed. Note that the spoofing feature requires GVI or RGVI to be enabled and configured to intercept replies directed back to the managed devices.

When the spoofing feature is disabled, TACACS+ and RADIUS requests will appear to the server as having been sent by the IP address of the Ranger Gateway rather than by specific managed devices. This option is easier to configure, but is valid only in cases where the access privileges for given users are the same across all managed devices.

Another option is to install the Ranger Gateway and the TACACS+/RADIUS server application on different servers. In this case, if source address spoofing is enabled, additional configuration will be required:

- IP Forwarding will need to be enabled on the Ranger Gateway server.
- Static routes will need to be configured on the TACACS+/RADIUS server, so that traffic destined for managed devices is routed to the Ranger Gateway. Note that this requires the TACACS+/RADIUS server and the Ranger Gateway server to be in the same subnet.
- The Ranger Gateway must have GVI or RGVI enabled and configured to intercept traffic destined for managed devices.

As before, disabling source address spoofing is a simpler option (much simpler in this case, as there is no need for IP forwarding, static routes, or GVI/RGVI), but is valid only in cases where the access privileges for given users are the same across all managed devices, because all requests will appear to the server as having been originated by the Ranger Gateway.
Source address spoofing for TACACS+ and RADIUS can be configured in the Ranger Gateway Viewer on the Configure > Gateway Settings dialog Access Control tab, or by using the configGateway command on the Ranger Gateway to set the variables radius_proxy_spoof and/or tacacs_proxy_spoof.

Where the Ranger Gateway and TACACS+/RADIUS server are not installed on the same server, it may be useful to use two or more Ranger Gateways, in order to provide high availability. In this case, each server group may have multiple entries for each TACACS+/RADIUS server, one for each Ranger Gateway that can be used to relay requests to that server. For example, if there are two equivalent TACACS+/RADIUS servers, acs1 and acs2, and two Ranger Gateways, rg1 and rg2, that can be used to relay requests to those servers, the corresponding server group would contain four entries:

- rg1 acs1
- rg2 acs1
- rg1 acs2
- rg2 acs2

**Configuration Example**

In order to illustrate the configuration required for TACACS+/RADIUS proxy, consider the following sample network:

![Figure 31-2. ZoneRanger TACACS+/RADIUS Proxy Configuration](image)

Note that there are four TACACS+/RADIUS servers shown in this diagram: acs1, acs2, acs3, and acs4. In the case of acs1 and acs2, the Ranger Gateway software is installed on the same server as the TACACS+/RADIUS server application. In the case of acs3 and acs4, the Ranger Gateway software has been installed on separate servers.
For the purposes of this example, assume that all DMZ devices are in the 10.1.1.0/255.255.255.0 subnet, that access to the router (10.1.1.1) and the two ZoneRangers (10.1.1.100 and 10.1.1.101) are to be authenticated and authorized through servers acs1 and acs2, using TACACS+ only, and that all other devices are to use acs3 and acs4, and may use TACACS+ or RADIUS. It will also be assumed that source address spoofing will be used when relaying requests to acs1 and acs2, but will not be used for acs3 and acs4.

In order to support this scenario, the following configuration would be required:

1. On both ZoneRangers, define the following server groups:
   
   serverGroup1:
   
   acs1 acs1
   
   acs2 acs2

   serverGroup2:
   
   rg3 acs3
   
   rg4 acs3
   
   rg3 acs4
   
   rg4 acs4

2. On both ZoneRangers, define the following TACACS+ proxy rules:
   
   10.1.1.1 serverGroup1
   
   10.1.1.[100-101] serverGroup1
   
   *.*.*.* serverGroup2

3. On both ZoneRangers, define the following RADIUS proxy rule:
   
   *.*.*.* serverGroup2

4. On acs1 and acs2, enable source address spoofing for TACACS+. In addition, the GVI or RGVI service should be enabled and configured to intercept traffic destined for 10.1.1.0/255.255.255.0.

5. On rg3 and rg4, ensure that source address spoofing for TACACS+ and RADIUS is disabled.

**Additional Configuration Options**

In order to troubleshoot any difficulties associated with the use of TACACS+ or RADIUS proxy services, the ZoneRanger can be configured to log all TACACS+ and/or RADIUS transactions. TACACS+ logging is configured on the **Configuration -> Access Control** page **TACACS+** tab on the ZoneRanger web interface, and RADIUS logging is configured on the **RADIUS** tab.

In addition, a number of advanced configuration settings are provided for each protocol. For TACACS+ the available settings are:

- **Client timeout** – the amount of time, in seconds, that the ZoneRanger will maintain information about an inactive TACACS+ authentication or authorization session.
- **Server timeout** – the amount of time that the Ranger Gateway will wait for a response from a TACACS+ server.
- **Maximum Message Size** – the maximum size, in bytes, of a valid TACACS+ message.

For RADIUS, the available settings are:
• **Client timeout**– the amount of time, in seconds, that the ZoneRanger will maintain information about an inactive RADIUS authentication or authorization session.

• **Server timeout** – the amount of time that the Ranger Gateway will wait for a response from a RADIUS server.

Server groups can also be configured with a number of protocol-specific options. For TACACS+, the available server group options are:

• **TACACS+ Shared Key** – the key used for encrypting TACACS+ messages. If this key is configured, the ZoneRanger will decrypt and validate all TACACS+ messages. Note that in order to use this option for a given server group, all devices managed by a given ZoneRanger that are mapped to that server group will need to be configured to use the same encryption key.

• **Insert IP Address** – If the TACACS+ Shared Key has been enabled, it is possible to configure the ZoneRanger to insert the requesting device’s address into the `rem_addr` field of a TACACS+ request, so that this address can be logged by the TACACS+ server. This option may be useful in the case where the Ranger Gateway is not configured to spoof the source address.

For RADIUS, the available server group options are:

• **RADIUS Shared Key** – the key used for authenticating and encrypting RADIUS messages. If this key is configured, the ZoneRanger will verify that all RADIUS messages have been signed with the shared key. Note that in order to use this option for a given server group, all devices managed by a given ZoneRanger that are mapped to that server group will need to be configured to use the same key.
Chapter 32: TFTP Proxy

TFTP (Trivial File Transfer Protocol) is a common protocol used in the management of network device configurations. The majority of network devices provide mechanisms whereby the devices can be instructed to transfer their configurations to/from a TFTP server. A growing number of management applications have been developed to use this mechanism to provide advanced configuration management for larger numbers of network devices.

ZoneRanger can be configured to proxy TFTP requests to the Ranger Gateway or through the Ranger Gateway to another TFTP server in the secure environment. Thus, ZoneRanger provides a secure mechanism for the TFTP protocol to manage the configuration files of ZoneRanger managed devices. ZoneRanger can also be configured to be a TFTP server for its managed devices.

Using Configuration > Inbound Proxy page TFTP tab, ZoneRanger can be configured how to handle TFTP requests from managed devices. When a TFTP proxy request is received by the ZoneRanger from a managed device, ZoneRanger uses the incoming Client Address to determine the appropriate rule from the TFTP Proxy Rules table. Each TFTP Proxy request can be processed in one of three ways indicated by Proxy Option:

1. None – Handle the TFTP requests locally on the ZoneRanger
2. To Gateway – Send the TFTP Requests to the specified Ranger Gateway
3. Thru Gateway – Send the TFTP Requests through the specified Ranger Gateway to the port on a remote TFTP server.

When the To Gateway option is used, the default Read and Write directory on the Ranger Gateway is install_dir/store/zr/tftpproxy for TFTP files. The Read and Write directories may be changed from the Ranger Gateway Viewer menu Configure > Gateway Settings window on the TFTP Proxy tab on the Ranger Gateway.

The Permissions configured in each TFTP Proxy rule specifies if the client is allowed to make read or write requests, and for write requests, if the user is allowed to create new files. Note, the : Create permission option is limited to "None" and "To Gateway" proxy options.

The TFTP proxy rule to be used for a given client device for an incoming TFTP request is identified by searching through the set of configured TFTP proxy rules in order (that is, from top to bottom as they are displayed in the table) until a match is found. Therefore, the order in which the rules appear in the table is very important.
Thus a set of static TFTP proxy rules may be configured to indicate how ZoneRanger processes incoming TFTP requests from managed devices. ZoneRanger TFTP proxy feature also can be used together with the SNMP proxy feature to handle the situation where managed devices are instructed to transfer configuration files using an SNMP set request. When the ZoneRanger sees an SNMP set request that instructs a device to perform a configuration file transfer, it can modify the request, effectively redirecting the request to use the ZoneRanger’s TFTP server, then when the managed device initiates the request, ZoneRanger will proxy the file through the Ranger Gateway to the originally requested TFTP server.

This capability can be enabled by checking the **Enable Single-Use SNMP triggered rules** checkbox. In this case, ZoneRanger generates a single use TCP proxy rule based on the SNMP set proxied via the Ranger Gateway. Note, this feature is triggered by sets using the CISCO-CONFIG-COPY-MIB (Cisco IOS software release 12.0) and the OLD-CISCO-SYSTEM-MIB/OLD-CISCO-FLASH-MIB (Cisco IOS software release 10.2 and later).

The **SNMP triggered rules timeout** field specifies the maximum life span for the SNMP triggered single use rules. If the ZoneRanger managed device uses the TFTP server on the ZoneRanger, that rule will be used, then removed so it will not be used again. If the rule is not used within this life span, the rule will expire and be discarded.

Files available via TFTP located on the ZoneRanger can be managed using the Ranger Gateway. The **Tools > TFTP Manager** window from the **Ranger Gateway Viewer** as well as Ranger Gateway commands `downloadTftpFile` and `uploadTftpFile` can be used to add and remove files from the ZoneRanger when it is acting as a TFTP server.
Chapter 33: Traffic Monitoring

ZoneRanger, in combination with Ranger Gateway, can receive and proxy different protocols from devices in the network to management applications. Due to the transparent nature of a ZoneRanger deployment, it can be difficult to determine how much traffic is being received and proxied through a particular ZoneRanger and Ranger Gateway. Also, it can be difficult to determine if a particular device is frequently sending information to a ZoneRanger or is frequently being queried by the ZoneRanger.

By default, ZoneRanger and Ranger Gateway will monitor the traffic they are receiving and proxying over the past configured interval for each Traffic Type (SNMP Trap, Syslog, ICMP, SNMP, etc). During the configured measurement interval on the ZoneRanger and the Ranger Gateway, the total amount of traffic for each Traffic Type will be measured.

On the ZoneRanger, the amount of traffic for each Traffic Type for each IP address will also be measured. If Traffic logging is enabled to Short, the total amounts of traffic will be logged. In addition the amount of traffic for the 25 “Top Talkers” will be recorded provided their total traffic exceeds 1% of the ZoneRanger's total traffic. If the Traffic logging is enabled to Full, the amount of traffic for each IP address will be recorded in descending order.

On the Ranger Gateway, the amount of traffic for each Traffic Type to each joined ZoneRanger will also be measured. If Traffic logging is enabled to Short, the total amounts of traffic will be logged. If Traffic logging is enabled to Full, the amount of traffic for each ZoneRanger will be recorded.

On the ZoneRanger web interface View > Traffic Information page, overall received traffic and proxy traffic, traffic per IP address and peak traffic per IP address may be viewed.

The ZoneRanger and Ranger Gateway may be configured to check thresholds for each Traffic Type over the measurement interval. If a threshold is exceeded, a message will be logged in the system log file and if enabled, a SNMP trap notification will also be sent. On the ZoneRanger, thresholds may be set for the overall Traffic Type and on a per IP address basis. On the Ranger Gateway, thresholds may be set for the overall Traffic Type and on a per ZoneRanger basis.
Chapter 34: Whitelisting

Whitelisting is the ability to restrict information to be only from a specific set of addresses. The ZoneRanger may be configured to only accept information from (Inbound) or send information to (Outbound) a specific set of IP addresses. Especially for the ZR-SPX model ZoneRanger, the configuration of a whitelist for Inbound information (SNMP Traps, Syslogs, etc) provides a security measure for the ZoneRanger to only process information from a known set of IP addresses. When whitelisting is enabled, only Inbound information which has a source address specified in the whitelist will be processed by the ZoneRanger. All other Inbound information with source addresses which are not in the whitelist will be ignored. This includes telnet, SSH, HTTP, and HTTPS requests.

It is able possible to configure the ZoneRanger to apply the whitelist to Outbound information. If enabled, Outbound requests (SNMP requests, ICMP, requests, etc) must have their source addresses specified in the whitelist. If the source address is not specified in the whitelist, the request will be discarded. Special care needs to be taken when enforcing the whitelist for Outbound requests. This enforcement will apply to all ZoneRanger initiated requests which include discovery, polling, joining, diagnostics, as well as proxy requests. This also includes network services such as DNS and NTP that will need to be added to the whitelist. However, joined Ranger Gateways and Redundant ZoneRangers are automatically whitelisted even though they will not appear in the whitelist.
Part IV. ZoneRanger and Ranger Gateway Interfaces

ZoneRanger has four user interfaces which may be used to interact with the system and are described in detail in the following chapters:

- ZoneRanger Web Interface  Chapter 35
- Ranger Gateway Viewer  Chapter 36
- ZoneRanger Text Interface  Chapter 37
- Ranger Gateway Command Interface  Chapter 38
Chapter 35: ZoneRanger Web Interface

ZoneRanger Dashboard

The ZoneRanger dashboard enables you to quickly view ZoneRanger activity, status, and general information. The dashboard consists of four sections; Activity, Inventory, Statistics, and Information. Each of the section's location, visibility, and contents may be configured on a per user basis from the View > Preferences page.

Activity Section

The Activity Section consists of a set of activity indicators which give an indication when a particular ZoneRanger service is in use. When an activity indicator flashes, ZoneRanger is performing tasks associated with the indicated service. If a indicator is dark, the associated activity is idle. Activity indicators provide a general indication of service activity, but are controlled to increase visibility and minimize performance impact. The number of times an activity indicator flashes typically does not indicate the number of transactions performed by the associated service.
<table>
<thead>
<tr>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>Flashes while ZoneRanger performs discovery</td>
</tr>
<tr>
<td>ICMP Poller</td>
<td>Flashes intermittently while ZoneRanger sends ICMP (ping) traffic to managed nodes to test whether nodes are responding</td>
</tr>
<tr>
<td>SNMP Poller</td>
<td>Flashes intermittently while ZoneRanger sends SNMP traffic to managed nodes to test whether node interfaces are up</td>
</tr>
<tr>
<td>TCP Poller</td>
<td>Flashes intermittently while ZoneRanger sends TCP connection requests to managed nodes to test whether nodes are accepting connections</td>
</tr>
<tr>
<td>HP OM Proxy</td>
<td>Flashes intermittently while ZoneRanger proxies HP OM requests</td>
</tr>
<tr>
<td>ICMP Proxy</td>
<td>Flashes intermittently while ZoneRanger proxies an ICMP (ping) request from the Ranger Gateway</td>
</tr>
<tr>
<td>NTP Proxy</td>
<td>Flashes intermittently while ZoneRanger proxies an NTP request from the Ranger Gateway</td>
</tr>
<tr>
<td>RADIUS Proxy</td>
<td>Flashes intermittently while ZoneRanger sends RADIUS requests and responses</td>
</tr>
<tr>
<td>SNMP Proxy</td>
<td>Flashes intermittently while ZoneRanger proxies a SNMP request from a Ranger Gateway, or a response from an SNMP agent</td>
</tr>
<tr>
<td>TACACS+ Proxy</td>
<td>Flashes intermittently while ZoneRanger sends TACACS+ requests and responses</td>
</tr>
<tr>
<td>TCP Proxy (Outbound)</td>
<td>Flashes intermittently while ZoneRanger sends TCP requests and responses</td>
</tr>
<tr>
<td>TFTP Proxy</td>
<td>Flashes intermittently while ZoneRanger sends TFTP requests and responses</td>
</tr>
<tr>
<td>Web File Proxy</td>
<td>Flashes intermittently while ZoneRanger proxies Web File HTTP or HTTPS requests</td>
</tr>
<tr>
<td>Generic Forwarding</td>
<td>Flashes intermittently while ZoneRanger forwards uncharacterized UDP packets</td>
</tr>
<tr>
<td>NetFlow Forwarding</td>
<td>Flashes intermittently while ZoneRanger forwards received NetFlow data</td>
</tr>
<tr>
<td>sFlow Forwarding</td>
<td>Flashes intermittently while ZoneRanger forwards received sFlow data</td>
</tr>
<tr>
<td>Syslog Forwarding</td>
<td>Flashes intermittently while ZoneRanger forwards received Syslog messages</td>
</tr>
<tr>
<td>Trap Forwarding</td>
<td>Flashes intermittently while ZoneRanger forwards SNMP traps</td>
</tr>
<tr>
<td>Data Diode Reader</td>
<td>Flashes intermittently while ZoneRanger reads files from Data Diode</td>
</tr>
<tr>
<td>Data Diode Writer</td>
<td>Flashes intermittently while ZoneRanger writes files from Data Diode</td>
</tr>
</tbody>
</table>

**Inventory Section**

The Inventory Section consists of the root cause status indicator and the inventory bars. The root cause indicator displays the current root cause information for the ZoneRanger managed nodes. The inventory bars display the status of the ZoneRanger managed nodes.
The root cause status indicator enables you to view a list of nodes that have been identified as the root cause of an outage. If the indicator is green with a check mark, there are no root causes associated with managed nodes. If the indicator is red with an exclamation point, it provides a link to the View > Root Causes page. If the indicator is red and the exclamation point is flashing, at least one root cause has occurred since the most recent viewing of the View > Root Causes page.

The inventory bars display the current status of the ZoneRanger managed nodes categorized by the following types:

- Routers
- Switches
- Web servers
- Servers

**Note:** Some nodes may appear in multiple categories. For example, a Solaris server that is also running a web server appears in both the Server and Web Server categories.

Each inventory bar represents a type of node and uses color to display the status of nodes of that type. The proportion of each color in an inventory bar represents the proportion of nodes of that type that have a particular status.

For example, if ZoneRanger is managing 20 routers and a quarter of the Routers inventory bar is yellow, five of the routers are marginal (the following section describes colors that can appear in the inventory bars). The total number of devices in a category appears in a number at the right of the inventory bar for the category. You can mouse over the inventory bar to see a count of devices associated with each status.

<table>
<thead>
<tr>
<th>Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>Represents nodes that are up. This node status is called normal.</td>
</tr>
<tr>
<td>Yellow</td>
<td>Represents nodes for which at least one interface or TCP port is down, but at least one interface or TCP port is up. This node status is called marginal.</td>
</tr>
<tr>
<td>Red</td>
<td>Represents nodes that are down. This node status is called critical.</td>
</tr>
<tr>
<td>Blue</td>
<td>Represent nodes whose status is unknown. Such devices are configured for polling but have not yet been polled, or are not configured for polling.</td>
</tr>
</tbody>
</table>

**Statistics Section**

The Statistics section displays various charts showing proxy and forwarding traffic over the previous 4.5 hour period, as well as information about the ZoneRanger itself. The information displayed is based on statistics values which are cached by the ZoneRanger application software. These statistics values can be viewed on the View > Statistics page. The displayed values are since the ZoneRanger started or since the values were reset on the View > Statistics page. These values are not historically ed. These Statistics are also available via the ZoneRanger SNMP agent.

The View > Preferences page can be used to customize which Chart appears first on the ZoneRanger Dashboard and which protocols are displayed on the Forwarding chart. Preferences are configured on a per user basis.
Information Section

The Information section displays pertinent information about this particular ZoneRanger. By default, this information includes:

- Audit Status
- Hostname
- IP address(es)
- Version
- Model
- Time
- Ranger Gateways

If the Audit status is green with a check mark, the ZoneRanger has not detected any irregularities from its self-check. If the Audit status is red with an exclamation point, it provides a link to the View > System Audit page which will describe the condition.

The View > Preferences page may be used to change what information is displayed in the Information Section on a per user basis.

The ZoneRanger menu

The ZoneRanger menu appears on all ZoneRanger pages. Users having administrator access authority have access to the entire ZoneRanger menu. Users with Operator access have access to the Home and View menus. The items available in each category are described in the following sections.

Administration

Backup/Restore

You can use the Administration > Backup/Restore page to backup and restore a ZoneRanger and to manage existing ZoneRanger backups. A ZoneRanger backup contains the ZoneRanger database and configuration information. Backups do not contain ZoneRanger system specific information such as hostname and IP address. A backup may be stored on the ZoneRanger itself, or on a joined Ranger Gateway. A backup can be used when replacing one ZoneRanger with another (for example, a hardware replacement or upgrade), or to restore a ZoneRanger back to a known state (for example, if a the ZoneRanger was misconfigured or to add a ZoneRanger to a pool).

![Figure 35-2 Administration > Backup/Restore page](image-url)
When a backup is restored, the ZoneRanger software is automatically restarted.

**Discovery**

In order for nodes to be managed by ZoneRanger, the nodes must be discovered. The Administration > Discovery page can be used to manually start the Discovery process.

![ZoneRanger Discovery Page](image)

*Figure 35-3. Administration > Discovery page*

When discovery is in progress, the Administration > Discovery page displays discovery progress, and the Discovery activity indicator on the ZoneRanger dashboard flashes to show activity. The length of time discovery takes depends on your network size and how discovery is configured.
The discovery progress page shows:

- The number of hosts found and scanned, and the number of host TCP ports scanned
- The number of IP addresses found and scanned, and the number of IP addresses that were resolved
- The number of subnets found, and the number of subnets for which Layer 2 and Layer 3 information has been scanned.

This page will automatically update while it is visible or if you return to it while discovery is in progress. As discovery progresses, the numbers in the Counts column converge for each of the entities begin scanned.

The Recent Events table shows the 25 most recently reported discovery events. When discovery finishes, the top row contains “Discovery Complete.” After discovery finishes, you can click View Last Discovery Log on the Administration > Discovery page to display log entries from the most recent discovery run. While viewing the log, you can filter messages by message text and limit the number of returned log entries.
Adding/Scanning Individual Nodes

Individual nodes may be added directly to the ZoneRanger database without running a full discovery. By using the Add/Scan Nodes section, up to 10 IP addresses or hostnames may be directly added to the ZoneRanger database. When the Start Scan button is pressed, the ZoneRanger will scan each entered IP address or hostname using SNMP and TCP and store the resulting information in the ZoneRanger database. If an existing IP address or hostname is scanned, the ZoneRanger database information for that device will be updated.

Note, when adding nodes individually, no subnet or path information for these devices will be available. Thus, root cause information will not be available for these newly added devices until a full discovery is executed.

Profiles

You can use the Administration > Profiles page to load, save, and manage ZoneRanger profiles. You can save a ZoneRanger profile on the ZoneRanger, or on a joined Ranger Gateway. Profiles enable you to save ZoneRanger configurations for later use. For example, having saved profiles available can save time when configuring other ZoneRangers requiring similar configurations.

A ZoneRanger profile comprises most of its configuration information, but does not include the contents of the ZoneRanger database. As such, a profile does not contain information associated with specific managed entities (for example, polling configuration for specific nodes, interfaces, and TCP ports). The Administration > Profiles page enables you to load saved profiles, store profiles, and delete profiles.

Figure 35-5. Adding Node via the Administration > Discovery page
When a profile is loaded on a ZoneRanger, the ZoneRanger software is automatically restarted.

**License Activation**

The Administration > License Activation page may be used to activate a ZoneRanger VM so that it may process management traffic. A ZoneRanger VM may obtain a license by either retrieving a license from a Ranger Gateway License Server or by using an Activation Key.

**Use a Ranger Gateway License Server**
A ZoneRanger VM may obtain its license from a Ranger Gateway License Server. When the
Use a Ranger Gateway License Server button is initially selected, the list of licenses from
each joined Ranger Gateway is presented under the Choose A License section. The table
displays the following attributes:

<table>
<thead>
<tr>
<th>Select</th>
<th>Ranger Gateway</th>
<th>Available</th>
<th>Used</th>
<th>Expiration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-MSP, HP Operations Manager</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>0</td>
<td>1</td>
<td>Jul 20, 2012</td>
<td>ZR-MSP, Web File</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-MSP</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-100</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-20</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-MSP, Data Diode, HP Operations Manager</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-50, Web File</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-MSP, Data Diode Anchor, Web File, HP Operations Manager</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-100, HP Operations Manager</td>
</tr>
<tr>
<td></td>
<td>sam.tavve.com</td>
<td>1</td>
<td>0</td>
<td>Jul 20, 2012</td>
<td>ZR-200, Data Diode</td>
</tr>
</tbody>
</table>

The Refresh License List button will request the current set of licenses from each of the
joined Ranger Gateways and update the list displayed in the Choose A License section. When a license is selected in the table, the Get License button is used to request the license
from the Ranger Gateway License Server.
If there is a current license allocated to this ZoneRanger VM and a new license is selected, the current license will be released if the new license is acquired. If the selected license is no longer available, the current license will be maintained. If there is a current license allocated to this ZoneRanger and a license with fewer managed nodes is selected and acquired, all of the managed nodes under the old licenses will be unmanaged.

Selection of a license may cause the ZoneRanger software to restart.

**Use an Activation Key**

When the **Use an Activation Key** button is selected, an activation key, based on the Pending Token, must be entered in the Load License Activation Key section. The Pending Token must be provided to Tavve Software which will return an activation key. When the activation key is entered and the **Load Activation Key** button is clicked, the activation key will be verified against the current pending token. If the verification is successful, the license will be activated and the ZoneRanger software will be restarted.

**Proxy Cache**

The **Administration > Proxy Cache** page may be used to clear the ZoneRanger SNMP Proxy and ICMP Proxy caches and to view the current cache sizes. The SNMP and ICMP proxy caches store past response information to be used to fulfill future proxy requests within a configured time period.
The size of the SNMP and ICMP proxy caches is the number of entries currently stored in the cache. The memory used by the proxy cache gives the amount of memory currently used by the caches (in kilobytes) and also lists the amount of memory at which the caches are considered full. When the caches are full, they will not store any new values (until enough entries time out for it to not be full), but will continue to return entries that have already been stored. Even when there are no entries in either cache the memory used will not be zero since there is some cache overhead.

**Route Management**

The Administration > Route Management page may be used to add and remove network routes from the ZoneRanger.
When a network route is added to the ZoneRanger, a confirmation dialog will be displayed indicating whether or not to commit this route. The route is temporarily added to the ZoneRanger but will be removed if the route is not committed within 60 seconds. This step is needed for routes which are added that cause the ZoneRanger to no longer be reachable. In case a route is inadvertently added to the ZoneRanger that would cause it to become unreachable, the route will expire in 60 seconds and the ZoneRanger would be reachable again.

![ZoneRanger Route Management](image)

**Figure 35-11. Administration > Route Management page for IPv6**

If the ZoneRanger has a ZR-SPX license, it will also support the ability to be attached to an IPv6 network. IPv6 route management, as seen in Figure 35-11, will become available after the ZoneRanger is configured with an IPv6 address on the **Configuration > System page IP tab**. When a IPv6 network route is added to the ZoneRanger, a confirmation dialog will be displayed indicating whether or not to commit this route. The route is temporarily added to the ZoneRanger but will be removed if the route is not committed within 60 seconds. This step is needed for routes which are added that cause the ZoneRanger to no longer be reachable. In case a route is inadvertently added to the ZoneRanger that would cause it to become unreachable, the route will expire in 60 seconds and the ZoneRanger would be reachable again.

**Service Dump**

If you are troubleshooting a potential ZoneRanger related problem, Tavve Support may require additional information for problem diagnosis and you may be asked to perform a **service dump**. Service dumps can provide Tavve Support with useful diagnostic information which includes detailed information about ZoneRanger configuration, status, and history.
When you perform a service dump, the ZoneRanger builds a service dump file and transfers the file to a joined Ranger Gateway. After a service dump file is transferred to a Ranger Gateway, you will be given detailed instructions for sending the service dump file to Tavve Support. Usually, a standard service dump contains all necessary troubleshooting data. However, Tavve Support might occasionally request a targeted service dump that contains specific data.

After a service dump is initiated, the ZoneRanger updates the page with the results of the service dump, including:

- File destination
- File name
- File size
- Approximate file transfer time remaining

If the file transfer is estimated to take less than 20 seconds, the results are displayed after the file transfer finishes. After a service dump finishes, an entry is made in the ZoneRanger system log indicating the success or failure of the service dump.

**Shut Down**

The **Administration > Shut Down** page provides a mechanism to restart the ZoneRanger application software, reboot the ZoneRanger, and shut down the ZoneRanger.
When restarting the ZoneRanger software, the ZoneRanger will stop processing requests during the restart period which may last several minutes. During that time, the ZoneRanger web interface will be unavailable.

SNMP

The Administration > SNMP page shows if there are any duplicate SNMPv3 Engine IDs and provides a mechanism that facilitates the process of updating SNMPv3 passwords.

The Administration > SNMP page Duplicate Engine IDs tab displays the list of IP addresses which ZoneRanger has determined to have the same SNMPv3 Engine ID. Each SNMP Agent that supports v3 has a unique Engine ID associated with that agent. When a ZoneRanger issues an SNMP v3 proxy request or receives an SNMPv3 notification, the SNMP Engine ID is cached by IP address. IP addresses on the same device may use the same SNMP Engine ID. When more than one device has the same SNMP Engine ID, SNMP v3 packets from the device with the lower SNMP Engine Reboots and SNMP Engine Time will most likely be discarded.

The Refresh button will re-check the cache for duplicate Engine Ids.

The Clear Engine IDs button will clear all SNMP Engine Ids from the ZoneRanger cache. The SNMP Engine Ids will be added to the cache if they are discovered again.

Changing user passwords for SNMPv3 devices

Since it is recommended that SNMPv3 passwords be changed periodically, the Administration > SNMP page SNMPv3 Passwords tab provides a mechanism that facilitates the process of updating SNMPv3 passwords on ZoneRanger managed nodes.
When updating SNMPv3 passwords, it is important to ensure that the passwords for each user are updated on all devices on which that user has been configured. If some devices are updated and some devices are not, the ZoneRanger will no longer be able to access devices that were not updated until their SNMPv3 passwords are manually reconfigured.

To ensure that all devices are updated properly, the SNMPv3 password configuration tool walks the user through a series of steps, performing tests along the way, and provides the option to revert changes if problems are encountered.

Note: The targets and SNMPv3 users must be configured on the Configuration > SNMP Manager tab.

To change SNMPv3 passwords using the configuration tool, perform the following steps from the Administration > SNMP page SNMPv3 Passwords tab:

Step 1: Click Analyze SNMPv3 Devices.

This tests all managed nodes for which SNMPv3 was configured. The analysis ensures that all such devices are reachable, and that their SNMPv3 user configuration information is accessible.

Step 2. Inspect the results of the analysis.

The analysis displays current passwords for each configured user, and the set of users configured on each device. If any devices were unreachable, or if SNMPv3 user configuration information was not available for a device, error messages are displayed. If error messages are reported for any devices, it is recommended that you resolve the underlying issue and repeat the analysis before proceeding with password updates.

Step 3. Update passwords.

If you decide to proceed, modify the authentication and privacy passwords for various users as desired and click Update Passwords to apply the changes. You can click Randomize to generate random passwords for all users, if desired.

Step 4. Inspect results.

Inspect the results of the password update. If the update succeeded for all devices, a table of updated nodes and users is displayed. If the update failed for one or more devices, error messages are displayed.

If error messages are present, you can:
a) Click **Undo Changes** to back out the changes. This resets user passwords to their previous values for any devices that were successfully updated. After backing out the changes, you can resolve the underlying issue, repeat the analysis, and try updating the passwords again.

b) Note which devices did not update properly, and modify the passwords manually. Typically, this is accomplished by logging in to the device and using the configuration interface for the device to modify the passwords.

Note that when the SNMPv3 passwords configuration tool successfully updates the passwords for a set of SNMPv3 users, corresponding configuration entries in the SNMPv3 users table (Configuration > SNMP page on the Users tab) are automatically updated to match the new passwords. If the **Undo Changes** button is used to reset the passwords to their previous values, corresponding configuration entries in the SNMPv3 users table are also automatically reset to their previous values.

### SSL Certificates

ZoneRanger uses SSL certificates for secure communication between Ranger Gateways and ZoneRangers, as well as communications between ZoneRanger and web browsers, for authentication and encryption of transmitted data.

The SSL configuration on each ZoneRanger or Ranger Gateway consists of two parts:

1. Configuring a ZoneRanger or Ranger Gateway with private encryption keys, and with corresponding certificates that it will use to identify itself, and to pass public encryption key material to other entities.
2. Configuring a ZoneRanger, Ranger Gateway, or web browser with the identities or “trusted subjects” with which it is authorized to communicate.

By default, all ZoneRangers and Ranger Gateways are configured with certificates issued by Tavve’s internal certificate authority. In order to provide increased security, some users may wish to obtain their own unique SSL certificates, either from Tavve’s internal certificate authority, or from a well known external certificate authority, such as VeriSign, Thawte, or Entrust. In these cases, it will be necessary to modify the SSL configuration on each Ranger Gateway and ZoneRanger, both with the new certificates, and with updated trusted subject lists.

### SSL Certificate Formats

New SSL public key certificate and private key may be installed on the ZoneRanger in the following formats:

1. PKCS #12
2. X.509 Certificate and Private Key
3. JKS Keystore

For PKCS #12, you will need the following:

1. The PKCS #12 file containing the private key and public key certificate. The public certificate must be signed by a trusted CA.
2. The password used to read the file.

For X.509 Certificate and Private Key, you will need the following:

1. The PEM file containing the public/private key pair for the new certificate.
2. The password to read the PEM file.
3. The new certificate, signed by a trusted CA

For JKS Keystore, you will need the following:

1. The Java keystore file containing the new certificate signed by a trusted CA.
2. The password to read the keystore file.
3. The password to read the key entry in the keystore file.

If the Trusted Certificate Authority used to sign the new SSL certificate is not in the ZoneRanger's list of Trusted Certificate Authorities it must be added before the new certificate is installed.

HTTPS SSL Certificates

The Administration > SSL Certificates page HTTPS tab provides a mechanism to change the private key and corresponding certificate on a ZoneRanger used to secure the communications between the ZoneRanger's web server and web browsers.

![ZoneRanger SSL Certificates](image)

**Figure 35-15. Administration -> SSL Certificates page HTTPS tab**
If the Trusted Certificate Authority used to sign the new SSL certificate is not in the ZoneRanger's list of Trusted Certificate Authorities it must be added before the new certificate is installed. The Configuration > Ranger Gateway page SSL Trust tab provides a mechanism to add new Trusted Certificate Authorities.

A new SSL public key certificate and private key may be installed using the formats specified above.

If the needed, use the web browser's specific interface to add the new Trusted Certificate Authority to the browser's the list of Trusted Root Certificate Authorities.

The Administration > SSL Certificates page HTTPS tab also provides a mechanism to revert the SSL certificate and private key back to its original setting if needed.

**Important Note.** You should use HTTPS when installing new keys and certificates using the ZoneRanger web interface to reduce the risk of unauthorized disclosure of sensitive encryption material.

**Messaging SSL Certificates**

The Administration > SSL Certificates page Messaging tab provides a mechanism to install the private key and corresponding certificates on a ZoneRanger for communications between the ZoneRanger and Ranger Gateways.

*Figure 35-16. Administration > SSL Certificates page Messaging tab*

If the Trusted Certificate Authority used to sign the new SSL certificate is not in the ZoneRanger's list of Trusted Certificate Authorities it must be added before the new certificate is installed. The Configuration > Ranger Gateway page SSL Trust tab provides a mechanism to add new Trusted Certificate Authorities.
A new SSL public key certificate and private key may be installed using the formats specified above.

Once the new messaging SSL certificate and private key are loaded on the ZoneRanger, all communications with joined Ranger Gateways will stop until the Trusted Subject for the new ZoneRanger certificate is added to the Ranger Gateways using the trustedSSL command.

The Administration > SSL Certificates tab Messaging page also provides a mechanism to revert the SSL certificate and private key back to its original setting if needed.

**Important Note.** You should use HTTPS when installing new keys and certificates using the ZoneRanger web interface to reduce the risk of unauthorized disclosure of sensitive encryption material.

**Trap Definitions**

ZoneRanger can be configured to forward SNMP traps based on trap specific information. To accomplish this, ZoneRanger must be configured with the particular SNMP trap definitions. The Administration > Trap Definitions page provides a mechanism to upload trap definitions via a trap definitions XML file. ZoneRanger is initially configured with a trap definitions file that contains all Tavve-defined traps.

![ZoneRanger Trap Definitions](image)

**Figure 35-17. Administration > Trap Definitions page**

The initial trap definitions file is also available on the Ranger Gateway. The location where the file is installed depends on your operating system:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris</td>
<td>install_dir/ZRCustom/Trap-Definitions.xml</td>
</tr>
<tr>
<td>Linux</td>
<td>install_dir/ZRCustom/Trap-Definitions.xml</td>
</tr>
<tr>
<td>Windows</td>
<td>install_dir\ZRCustom\Trap-Definitions.xml</td>
</tr>
</tbody>
</table>

where <install_dir> is the directory where the Ranger Gateway software was installed.

The trap definition file uses a simple NMS-neutral XML format. To add your own trap definitions, you can manually edit the trap definition file, or you can use the trapdToXml command, which can be used to merge trap definitions from OpenView/NetView trapd.conf files into the XML trap definition file. The trapdToXml tool is included in the Ranger Gateway install. When editing the trap definition file, be aware that the trap definitions file will be replaced on new or migration installations of the Ranger Gateway.

The currently configured trap definitions may be viewed on the Configuration > Forwarding page in the Trap Filters tab. When a new trap definition file is uploaded to the ZoneRanger, it is validated, and if valid, is installed on the ZoneRanger.
**Note:** Uploading a new trap definition file replaces the previous version. Be sure that the new trap definition file includes all previously defined traps.

**XML trap definition example**

ZoneRanger uses an NMS-neutral XML format for trap definitions. An example of an XML trap definition follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<trap-definitions>
  <trap-definition name="tscZRIfDown">
    <enterprise-oid>1.3.6.1.4.1.2668.1.1.16</enterprise-oid>
    <generic-type>6</generic-type>
    <specific-type>51</specific-type>
    <format>Interface $1 ($2) down.</format>
    <description>The interface was not able to be reached by ZoneRanger.</description>
  </trap-definition>
  ...
</trap-definitions>
```

**Configuration**

**Access Control**

Any user interaction with ZoneRanger requires logging in with a user name and password. The method of authentication and the determination of the valid set of user names and passwords is configured on the Configure > Access Control page. The Configure > Access Control page is also used to configure ZoneRanger to proxy TACACS+ and RADIUS requests for managed devices.

**Authentication using TACACS+ and RADIUS**

ZoneRanger can proxy TACACS+ and RADIUS requests from its managed devices to a TACACS+ or RADIUS server for authentication and authorization of user names and passwords. These requests and responses are proxied through a joined Ranger Gateway.

User authentication is organized through the use of Server Groups. Incoming authentication requests are sent to the TACACS+ or RADIUS server determined by the node’s membership in one or more server groups. In the case of a node being configured in multiple server groups, the request will be sent in order to each TACACS+ or RADIUS server until a success or failure response is received.

The ZoneRanger itself can be configured to either use TACACS+ or RADIUS for authentication but not both. The ZoneRanger may be configured to communicate directly with a TACACS+ or RADIUS server or it can be proxied through a Ranger Gateway. If the authentication fails on the ZoneRanger from a central authority, ZoneRanger will attempt to authenticate the user locally.

**Security levels**

ZoneRanger has two security(authorization) levels which determine the amount of user access.
### Security Level

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin</td>
<td>The Admin security level has access to all ZoneRanger configuration pages</td>
</tr>
<tr>
<td>Operator</td>
<td>The Operator security level limits access to viewer pages; the Administration, Diagnostics, and Configure menus and pages are hidden.</td>
</tr>
</tbody>
</table>

#### Updating user information

The **Configuration > Access Control** page **Users** tab is used to configure user access to the ZoneRanger. User access is controlled for the web interface, Telnet, and SSH. By default, ZoneRanger is initially configured with two users, **operator** and **admin**. The default password for operator is **operator**, and the default password for admin is **admin**. You can modify the security level and change passwords for existing users, delete existing users, and add new users. For new users, you can configure user IDs, security levels, and passwords. Passwords must be at least five alphanumeric characters.

In addition to user authentication, ZoneRanger maintains two security levels to access particular ZoneRanger functions; remote database access and the text configuration menu. The Setup User security level is used to access the text configuration menu for initial setup of the ZoneRanger via telnet, SSH and a connected console. By default the user is **setup** and the password is **setup**. The user and the password for the Setup User security level may both be changed. The password must contain at least five alphanumeric characters. Special characters are not accepted.

The database user security level allows for remote, read-only database access using a joined Ranger Gateway. The user name is **ranger_ro**. By default, the password is **readonly**. The password may be changed and must contain at least five alphanumeric characters. Special characters are not accepted. When accessing the database, the database name is **rangerDb**.
Configuring TACACS+

The Configuration > Access Control page TACACS+ tab allows for the configuration of ZoneRanger TACACS+ proxy for authentication of managed nodes as well as TACACS+ authentication on the ZoneRanger itself. At least one Server Group (see Chapter 17) must be created before TACACS+ proxy configuration can be accomplished. TACACS+ authentication of the ZoneRanger itself may be proxied through a Ranger Gateway, which requires at least one Server Group, or may be configured to communicate directly to a TACACS+ server.

Configuring TACACS+ proxy

![Configuration > Access Control page TACACS+ tab](Figure 35-19. Configuration > Access Control page TACACS+ tab)

The Proxy Rules section is used to define which Server Group is selected for each incoming TACACS+ request. Thus, ZoneRanger managed nodes can be organized across multiple TACACS+ servers depending on an organization’s user authentication strategy. For example, network devices may authenticate to one TACACS+ server while servers authenticate to another TACACS+ server. The Source Address field must be IP address or may be an address pattern or Node Group (see Chapter 2).

TACACS+ requests received by a ZoneRanger, and TACACS+ responses sent by a ZoneRanger can be written to a log file, called /log/tacacsProxy.log. This log can be downloaded using the `downloadFile` command on a Ranger Gateway. This can affect the performance of TACACS+ proxy. The log file may also be viewed on the View > Service Logs page. The Log Levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Message header is logged</td>
</tr>
<tr>
<td>Full</td>
<td>Entire message is logged</td>
</tr>
</tbody>
</table>

ZoneRanger 5.6 User's Guide 136
You can use the **Show Advanced** button to access and configure the following advanced options:

<table>
<thead>
<tr>
<th>Advanced Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Timeout</td>
<td>Amount of time in seconds a ZoneRanger waits for a message from a TACACS+ client before closing the TCP connection for that client.</td>
</tr>
<tr>
<td>Server Timeout</td>
<td>Amount of time in seconds a ZoneRanger or Ranger Gateway waits for a response from a TACACS+ server.</td>
</tr>
<tr>
<td>Maximum Message Size</td>
<td>Maximum allowed TACACS+ message size, in bytes.</td>
</tr>
</tbody>
</table>

**Configuring TACACS+ for ZoneRanger via proxy**

The **Use TACACS+ for ZoneRanger access control** checkbox enables ZoneRanger to authenticate and authorize web, Telnet, and SSH users using TACACS+. ZoneRanger may be configured to authenticate directly to a TACACS+ server or through a Ranger Gateway using TACACS+ proxy. The **Access Mode** dropdown determines which method the ZoneRanger should use to authenticate with a TACACS+ server.

![Figure 35-20. Configuring ZoneRanger to authenticate via TACACS+ proxy](image)
When authenticating the ZoneRanger itself using TACACS+ proxy, a **Server Group** must be specified along with **Login Type**. The privilege levels corresponding to the operator and administrator privileges must be set to those configured on the TACACS+ server. The ZoneRanger uses an authorization request to retrieve the privilege level of the user from the TACACS+ server. This request contains a number of authorization arguments one of which must be the primary service. Additional arguments may be required by the TACACS+ server in order to return the privilege level of the user.

**Configuring TACACS+ for ZoneRanger direct**

The **Use TACACS+ for ZoneRanger access control** checkbox enables ZoneRanger to authenticate and authorize web, Telnet, and SSH users using TACACS+. ZoneRanger may be configured to authenticate directly to a TACACS+ server or through a Ranger Gateway using TACACS+ proxy. The **Access Mode** dropdown determines which method the ZoneRanger should use to authenticate with a TACACS+ server.

---

![ZoneRanger User Interface](image)

**Figure 35-21. Configuring ZoneRanger to authenticate via TACACS+ directly**
When authenticating the ZoneRanger itself directly to a TACACS+ server, at least one TACACS server must be specified along with Login Type. Use the Add TACACS+ Server button to add additional TACACS+ servers. ZoneRanger will choose from the listed TACACS+ servers with which it has most recently authenticated successfully. If the current authentication fails, the ZoneRanger will use additional servers if a timeout has not yet occurred. The privilege levels corresponding to the operator and administrator privileges must be set to those configured on the TACACS+ server. The ZoneRanger uses an authorization request to retrieve the privilege level of the user from the TACACS+ server. This request contains a number of authorization arguments one of which must be the primary service. Additional arguments may be required by the TACACS+ server in order to return the privilege level of the user.

Cisco ACS servers, beginning with version 5, require a TACACS+ authorization request to include a Command argument if the Service argument is shell. If the Command box is checked and a Command argument value is specified, a Command argument with the specified value will be added to the authorization request. If the Command box is checked and no Command argument value is specified, an empty Command argument will be added to the authorization request.

Configuring RADIUS

The Configuration > Access Control page RADIUS tab allows for the configuration of ZoneRanger RADIUS proxy for authentication of managed nodes as well as RADIUS authentication on the ZoneRanger itself. At least one Server Group (see Chapter 16) must be created before RADIUS proxy configuration can be accomplished. RADIUS authentication of the ZoneRanger itself may be proxied through a Ranger Gateway, which requires at least one Server Group, or may be configured to communicate directly to a RADIUS server.

The Proxy Rules section is used to define which server group is selected for each incoming RADIUS request. Thus, ZoneRanger managed nodes can be organized across multiple RADIUS servers depending on an organization’s user authentication strategy. For example, network devices may authenticate to one RADIUS server while servers authenticate to another RADIUS server. The Source Address field must be IP address and may be an address pattern or Node Group (see Chapter 2).
RADIUS requests received by a ZoneRanger, and RADIUS responses sent by a ZoneRanger can be written to a log file, called log/radiusProxy.log. This log can be downloaded using the downloadFile command on a Ranger Gateway. This can affect the performance of RADIUS proxy. The log file may also be viewed on the View > Service Logs page. The Log Levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Message header is logged</td>
</tr>
<tr>
<td>Full</td>
<td>Entire message is logged</td>
</tr>
</tbody>
</table>

You can use the Show Advanced Options button to access and configure the following advanced options:

<table>
<thead>
<tr>
<th>Advanced Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Timeout</td>
<td>Amount of time in seconds a ZoneRanger waits for a message from a RADIUS client before closing the TCP connection for that client.</td>
</tr>
<tr>
<td>Server Timeout</td>
<td>Amount of time in seconds a ZoneRanger or Ranger Gateway waits for a response from a RADIUS server</td>
</tr>
</tbody>
</table>

Configuring RADIUS for ZoneRanger via proxy

The Use RADIUS for ZoneRanger access control checkbox enables ZoneRanger to authenticate and authorize web, Telnet, and SSH users using RADIUS. ZoneRanger may be configured to authenticate directly to a RADIUS server or through a Ranger Gateway using RADIUS proxy. The Access Mode dropdown determines which method the ZoneRanger should use to authenticate with a RADIUS server.
When authenticating the ZoneRanger itself using RADIUS proxy, a Server Group must be specified.

**Configuring RADIUS for ZoneRanger direct**

The Use RADIUS for ZoneRanger access control checkbox enables ZoneRanger to authenticate and authorize web, Telnet, and SSH users using RADIUS. ZoneRanger may be configured to authenticate directly to a RADIUS server or through a Ranger Gateway using RADIUS proxy. The Access Mode dropdown determines which method the ZoneRanger should use to authenticate with a RADIUS server.

Figure 35-24. Configuring ZoneRanger to authenticate via RADIUS directly

When authenticating the ZoneRanger itself directly to a RADIUS server at least one RADIUS server must be specified. Use the Add RADIUS Server button to add additional RADIUS servers. ZoneRanger will choose from the listed TACACS+ servers with which it has most recently authenticated successfully. If the current authentication fails, the ZoneRanger will use additional servers if a timeout has not yet occurred. An optional RADIUS Shared Key may be specified in the RADIUS Shared Key field to be used to encrypt and decrypt RADIUS messages.

**Configuring Server Groups**

Server groups are used by TACACS+ and RADIUS proxy to configure the set of destination servers for proxied requests. At least one Server Group must be defined before TACACS+ and RADIUS proxy can be configured.
The **Server Group Entries** section defines the set of ports, authentication server, and Ranger Gateway for this server group. Multiple Server Group entries may be assigned to a Server Group. If a specified authentication server fails to respond to an incoming authentication request, the next authentication server will be sent the request.

**TACACS+ Shared Key** defines the optional key used to decrypt TACACS+ messages. If a shared default key was specified and **Insert IP Address** is checked, ZoneRanger inserts the source IP address into the `rem_addr` field of authentication START, authorization REQUEST, and accounting REQUEST messages so that TACACS+ servers can log the original source of the TACACS+ request. In order to use this option, all devices managed by a ZoneRanger configured to use the same server group must be configured with the same key.

If no key is specified, TACACS+ message bodies are not decrypted or modified as they pass through the ZoneRanger.

**RADIUS Shared Key** defines the key used for encryption when this server group is selected as the server group to use for ZoneRanger access control.
Discovery

In order for devices to be managed by ZoneRanger, they must be discovered. The Configuration > Discovery page provides the ability to adjust a variety of configuration options associated with the discovery process.

Figure 35-27. Configuration > Discovery page Options tab

Configuring Discovery Options

The Periodic discovery interval checkbox enables ZoneRanger to run discovery on a periodic basis. The value in the Periodic discovery interval field specifies the periodic discovery interval in days, hours, and minutes. The default interval is 7 days. If you change the interval, discovery runs again after the interval you specified passes. The interval begins when you save the discovery configuration.

When Search for additional nodes is checked, discovery can use several methods to find nodes that are not specifically configured using seed nodes and ping ranges. Minimally, discovery examines the network connectivity of nodes specified by the seed list and ping ranges to find intermediate nodes. When Search for additional nodes is unchecked, no extra searching is performed, and only those nodes specified by the seed list and ping ranges are discovered.

Note: To use root cause correlation, you must check Search for Additional Nodes. This examination of network connectivity is required to build the root cause correlation rules.

Using additional advanced methods, discovery can search for nodes that fall outside of the seed node list, ping ranges and connectivity constraints. The following methods are available:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read IP route table</td>
<td>This method uses SNMP to examine IP routing tables on discovered nodes to discover additional IP addresses</td>
</tr>
<tr>
<td>Read ARP Cache</td>
<td>This method uses SNMP to examine ARP caches on discovered nodes to discover additional IP addresses</td>
</tr>
<tr>
<td>Broadcast ping enabled</td>
<td>This method sends a broadcast ping to discovered subnets, to discover additional IP addresses</td>
</tr>
</tbody>
</table>
The **Auto enable polling for newly discovered nodes** checkbox enables polling of newly discovered nodes, using parameters set on the **Configuration > Polling** page. Unchecking this box disables polling of newly discovered nodes. You can use the **Configuration > Polling** page to enable polling manually. In order for a node to be polled, it must also be managed by the ZoneRanger.

The **Auto manage newly discovered nodes** checkbox causes all newly discovered nodes, up to the license limit, to be managed by ZoneRanger. Unchecking this box causes all newly discovered nodes to initially be designated as unmanaged. To manage them, use the **Configuration > Node Management** page.

The **Ignored IP Addresses** section lists the set of IP address to ignore while performing network discovery. When a node is scanned, if any listed address is found, it is not stored in the database. This can be useful for administrative IP addresses that are duplicated in the network.

### Configuring Seed Nodes

The **Configuration > Discovery** page **Seed Nodes** tab lists the nodes to explicitly add to the ZoneRanger databases. If **Search for additional nodes** is enabled on the **Options** tab, the list of **seed nodes** will also be used as a starting point for additional discovery.

You can add seed nodes using the **table view** or the **quick entry view** of the seed nodes list. The quick entry view can be used to paste from a file. For a small network that changes infrequently, a seed node list and **manual discovery** may be all that is necessary to populate the ZoneRanger database.

### Configuring Networks

The **Configuration > Discovery** page **Networks** tab, is used to limit additional nodes that ZoneRanger can potentially discover. This helps control the number and address ranges of devices that will be added to the ZoneRanger managed node list, and helps limit the time spent in discovery.
The **Include Networks** section defines the outer bounds of the network to be discovered. If any include networks are defined, only nodes that have an interface in at least one include network are discovered.

The **Exclude Networks** section defines areas in the network to avoid during discovery. Interfaces in an exclude network are added only if they are on a node that has other interfaces that are in an include network.

The filtering provided by the include network and exclude network lists applies to ping ranges, addresses found in ARP caches, route tables, and interface tables, and to addresses found by broadcast pings and root cause path analysis.

Filtering does not apply to addresses specified in seed nodes. In effect, seed nodes override filtering.

**Note:** If the include network list is empty, only the exclude network list limits discovery. If the exclude network list is also empty, discovery is not limited.

### Configuring Ping Ranges

The **Configuration > Discovery** page **Ping Ranges** tab is used to specify a list of IP addresses to ping as a part of discovery. During discovery, ZoneRanger pings (sends one or more ICMP echo requests) all addresses in any specified ping ranges that pass the include/exclude network filtering criteria. Any address that responds is added to the database.
A ping range is an IP address in which each octet can be a number, or a range of numbers enclosed in square brackets.

The following examples illustrate how ping ranges are interpreted:

<table>
<thead>
<tr>
<th>Ping Range</th>
<th>IP Addresses Pinged</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.10.10.[12-15]</td>
<td>10.10.10.12, 10.10.10.13, 10.10.10.14, and 10.10.10.15.</td>
</tr>
<tr>
<td>10.10.10.*</td>
<td>10.10.10.1 ... 10.10.10.255.</td>
</tr>
<tr>
<td>10.[10-11].20.[3-5]</td>
<td>10.10.20.3, 10.10.20.4, 10.10.20.5, 10.11.20.3, 10.11.20.4, and 10.11.20.5.</td>
</tr>
</tbody>
</table>

**Note:** Using large ping ranges greatly increases discovery time.

**Configuring TCP Ports**

The **Configuration > Discovery** page TCP Ports tab defines the different TCP services that the ZoneRanger will discover and monitor. The TCP port list on this tab is initially populated with common TCP services.

The **TCP Port Scan Timeout** field specifies how long to wait for a response from the TCP port before timing out.

If **Auto Discover** is checked for a service, the corresponding TCP port is tested on all discovered nodes to determine whether the node supports the service. To keep nodes that do not have the service from having the port scanned, uncheck the corresponding **Auto Discover** checkbox. You can use the **Configuration > Polling** page to manually configure the service to be polled on a node.
If Auto Discover is checked for a service, the associated TCP port is tested on all discovered nodes to see which nodes support the service. You can use the Configuration > Polling page TCP Settings tab to manually add TCP ports to selected managed nodes, delete TCP ports from selected managed nodes, and to enable/disable polling for specific ports on selected managed nodes.

When a TCP service is deleted, all instances of this service are deleted from the ZoneRanger database, and any TCP polling of the deleted service is stopped.

Configuring Device Types

The Configuration > Discovery page Device Types tab, provide a mechanism to map the sysObjectID of discovered devices to a device type, enabling you to identify routers and switches.

ZoneRanger can use the SNMP sysObjectID of a device to determine its type. If the sysObjectID of a device matches an entry in the Device Type Mappings list, the node is assumed to be of the specified type. After you add or edit device type mappings, ZoneRanger does not update its device inventory until discovery runs again.

Forwarding

Using the Configuration > Forwarding page, ZoneRanger can be configured to listen for traffic on specified ports, and forward the traffic through a Ranger Gateway to a specified destination host and port that is reachable from the Ranger Gateway.
Configuring forwarding rules

The Configuration > Forwarding page Forwarding Rules tab allows the creation of forwarding rules associated with each Ranger Gateway or Destination Group. Each entry in the forwarding rules table consists of the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Whether or not this forwarding rule is currently active</td>
</tr>
<tr>
<td>Type</td>
<td>Trap, Syslog, NetFlow, sFlow, Generic</td>
</tr>
<tr>
<td>Source Addresses</td>
<td>Source addresses of the data coming in to ZoneRanger. Addresses are separated by commas and may be an address pattern or Node Group.</td>
</tr>
<tr>
<td>Local Port</td>
<td>The local port ZoneRanger is listening on for this traffic</td>
</tr>
<tr>
<td>Destination Host</td>
<td>Hostname or IP address that will ultimately receive the data. Must be reachable by the Ranger Gateway.</td>
</tr>
<tr>
<td>Destination Port</td>
<td>Port on the Destination Host where the data will be sent.</td>
</tr>
<tr>
<td>Filter</td>
<td>Additional filtering options for Trap and Syslog.</td>
</tr>
</tbody>
</table>

Additional Trap Filtering

When configuring a trap forwarding rule, the Filter column for the corresponding row in the rules table contains a pulldown list of defined trap filters, and an Options button, that can be used to configure additional forwarding options.

The pulldown list contains the list of trap filters defined on the ZoneRanger. ZoneRanger provides a base set of trap filters which cannot be deleted. You can use this list to select the trap filter to be applied for the rule you are defining.

The Options button provides the following set of trap conversion options:
### Conversion

<table>
<thead>
<tr>
<th>Conversion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do Not Convert</td>
<td>All traps and SNMPv2c informs are forwarded in the same form in which they were received. SNMPv3 informs are converted to SNMPv3 traps.</td>
</tr>
<tr>
<td>Convert to SNMPv1</td>
<td>SNMPv1 traps are forwarded in the same form in which they were received, SNMPv2c traps and informs are converted to SNMPv1 traps, and SNMPv3 traps and informs are converted to SNMPv1 traps.</td>
</tr>
<tr>
<td>Convert to SNMPv2c</td>
<td>SNMPv1 traps are converted to SNMPv2c traps, SNMPv2c traps and informs are forwarded in the same form in which they were received, and SNMPv3 traps and informs are converted to SNMPv2c traps.</td>
</tr>
</tbody>
</table>

In cases where an SNMPv2c or SNMPv3 inform is converted to a trap, the ZoneRanger will automatically respond to the device that sent the inform with the appropriate response. In the case of SNMPv3 traps and informs, the user defined in the incoming notification must be configured under the SNMPv3 Users in order to completely process the notification.

### Additional Syslog Filtering

Syslog messages can be filtered using several different criteria. If no criteria is specified, all Syslog messages are forwarded. To filter on a certain type of criteria, check the box to the left of the criteria label and enter the desired filtering criteria. If multiple criteria are selected, a Syslog message must match all selected criteria to be forwarded.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Name</td>
<td>Name of the program that generated the Syslog message, as the name appears in the message.</td>
</tr>
<tr>
<td>Message Search</td>
<td>Search string that the Syslog message must contain. The search string can be a regular expression search. The Syslog message will be forwarded when the search criteria is met. The <strong>Not Checkbox</strong> will prevent messages from being forwarded if they meet the search criteria.</td>
</tr>
<tr>
<td>Cisco Syslog with Max Severity</td>
<td>Cisco Syslog messages of the specified severity or lower.</td>
</tr>
<tr>
<td>Syslog with Max Severity</td>
<td>Syslog messages of the specified severity or lower.</td>
</tr>
<tr>
<td>Syslog with Facility</td>
<td>Syslog messages of the specified facility.</td>
</tr>
</tbody>
</table>

When **Forward as Syslog** is selected, incoming Syslog messages are relayed to the destination. When **Forward as Trap** is selected, incoming Syslog messages are converted into and forwarded as traps.

If the **Cisco Syslog with Max Severity** criteria is chosen, the correct Cisco trap for the severity is generated. Otherwise, a Syslog trap with the specified **Specific Type** is generated.
If the **Program Name** condition or the **Message Search** condition is enabled and the associated values is blank, that is equivalent to disabling the condition.

The **Forwarding Logging levels** are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Message header or length is logged</td>
</tr>
<tr>
<td>Full</td>
<td>Entire message is logged</td>
</tr>
</tbody>
</table>

This Forwarding log can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file may also be viewed on the **View > Service Logs** page. The log file is called `/log/udpFwd.log`.

**Configuring destination groups**

The **Configuration > Forwarding** page **Destination Groups** tab provides the ability to create destination groups, which are a named set of rules comprised of Ranger Gateways or Data Diodes and the ultimate destination for UDP datagrams. Destination Groups are a mechanism to more concisely organize multiple forwarding rules with the same set of destinations.

Each destination group is comprised of a set of rules. Each rule is comprised of a Ranger Gateway or Data Diode and the hostname or IP address of the ultimate destination of the UDP datagram. Destination groups may also contain other destination groups. Once a destination group is created, it will appear within the drop-down on the Forwarding Rules tab.

![Configuration > Forwarding page Destinations tab](image)

*Figure 35-34. Configuration > Forwarding page Destinations tab*

The first step in the creation of a Destination Group is entering the Destination Group name and clicking the **Add** button.
The next step is to add the required number of Destination Rules by clicking the **Add Destination Rule** button and entering the appropriate Ranger Gateway and Destination for each Destination Rule.

Destination Groups can contain other Destination Groups. Thus, if there are already defined Destination Groups, each can be added using the **Add Destination Group** button.

**Configuring trap filters**

The **Configuration > Forwarding** page **Trap Filters** tab provides the ability to create trap filters, which are a named set of conditions for matching traps. ZoneRanger uses trap filters as the filtering criteria for forwarding traps in forwarding rules.
ZoneRanger provides a set of pre-defined trap filters. Using the Add Custom Trap Filter button, additional trap filters may be created. Trap filters are configured with a set of conditions which either all conditions must be true or at least one condition must be true. Trap filters may use the following conditions:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trap</td>
<td>Previously defined trap</td>
</tr>
<tr>
<td>Enterprise OID</td>
<td>SNMPv1 Enterprise OID of a trap or an OID prefix</td>
</tr>
<tr>
<td>Generic Trap</td>
<td>SNMPv1 Generic Type of a trap</td>
</tr>
<tr>
<td>Specific Trap</td>
<td>SNMPv1 Specific Type of a trap</td>
</tr>
<tr>
<td>Trap OID</td>
<td>SNMPv2c Trap OID of a trap or an OID prefix</td>
</tr>
<tr>
<td>Variable Binding</td>
<td>Variable binding value of a trap, defined by an index starting at 1. An '*' may be used at the beginning and/or end of the value to denote a wildcard match</td>
</tr>
<tr>
<td>Agent</td>
<td>SNMPv1 agent of a trap (IP address or IP address range) Multiple agents can be listed using commas between IP addresses and may be an address pattern.</td>
</tr>
<tr>
<td>Version</td>
<td>SNMP version of the trap</td>
</tr>
<tr>
<td>Community</td>
<td>SNMP community string, or user name for SNMPv3</td>
</tr>
<tr>
<td>Trap Filter</td>
<td>previously defined trap filter</td>
</tr>
</tbody>
</table>

Configuring syslog options

The Configuration > Forwarding page Syslog tab provides the ability to modify the strictness of syslog forwarding.
ZoneRanger provides the capability to configure additional checking of syslog messages to further enhance the security of syslog forwarding through the ZoneRanger. When checking the **Require Printable Characters** button, only those syslog messages with printable characters will be forwarded. The determination of printable characters will be based on the format of the syslog message. For BSD or RFC 3164 formatted syslog messages, those syslog messages that only contain printable ASCII characters (Decimal 32 – Decimal 126) will be forwarded. For RFC 5424 formatted syslog messages, those syslog messages which only contain printable UTF-8 characters will be forwarded. All syslog message which contain non-printable characters will be discarded. Warning messages will be logged in the **View > System Log** for discarded syslog messages. Discarded messages will be logged if Forwarding logging is set to Short or Full.

**Inbound Proxy**

Using the **Configuration > Inbound Proxy** page, ZoneRanger can be configured to process various inbound proxy services.

The TFTP rules determine whether a file is transferred from the ZoneRanger managed device to the ZoneRanger itself, through the ZoneRanger to a joined Ranger Gateway, or through the ZoneRanger and joined Ranger Gateway to a specified TFTP server. TFTP rules can be assigned to specific interfaces, specific nodes, or groups of interfaces having IP addresses that match a specified pattern. Each **TFTP Proxy Rule** contains the following information:
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Address</td>
<td>IP address pattern, Node Group, the IP address of an interface, or the fully qualified hostname of a TFTP client which is a ZoneRanger managed device. The pattern, address, or hostname identifies the interface, set of interfaces, or node to which settings in the group are applied.</td>
</tr>
<tr>
<td>Permissions</td>
<td>Permissions determining whether the TFTP client can make read or write requests, and for write requests, if the TFTP client is allowed to create new files. Create permission is limited to None and To Gateway proxy options.</td>
</tr>
<tr>
<td>Proxy Option</td>
<td>Proxy option determines how read and write requests are to be processed locally on ZoneRanger (None), on a Ranger Gateway (To Gateway) or on another TFTP server in the trusted network (Through Gateway). See table below.</td>
</tr>
<tr>
<td>Gateway</td>
<td>Ranger Gateway to use for proxied requests</td>
</tr>
<tr>
<td>Remote TFTP Server</td>
<td>Remote TFTP server/port to use in the trusted network. This option is only valid for Through Gateway transactions.</td>
</tr>
<tr>
<td>Port</td>
<td>Port to use on the remote TFTP server.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Proxy Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>All TFTP transfers are between the client and the ZoneRanger</td>
</tr>
<tr>
<td>To Gateway</td>
<td>Client initiates a TFTP session with the ZoneRanger and files are transferred to and from the Gateway through the ZoneRanger. The default directory used on the Gateway to for the files is install_dir/store/zr/tftpproxy</td>
</tr>
<tr>
<td>Through Gateway</td>
<td>Client initiates a TFTP session with the ZoneRanger and files are transferred to and from the specified TFTP client tftp directory through the ZoneRanger and Gateway. No files are stored on the Gateway.</td>
</tr>
</tbody>
</table>

The TFTP rule to be used for a given client device is identified by searching through the set of configured rules in order (that is, from top to bottom as they are displayed in the table) until a match is found. Therefore, the order in which the rules appear in the table is very important. The arrow buttons in each row can be used to modify the order of the rules.

The Enable Block Number Wrapping checkbox allows a file larger than 33553919 bytes to be transferred (max block number is 65535, and blocks are 512 bytes long, except for the last which must be < 512). When not checked, it prevents a transfer session from endlessly sending a file until the file system is full.
The **Enable Single-Use SNMP Triggered Rules** checkbox allows the TFTP proxy service to auto-generate single use rules based on SNMP Set requests proxied via the Ranger Gateway. For example, some devices can be instructed to save their configuration using an SNMP set. The contents of the set will specify where the file should be saved. With this feature enabled, the proxied SNMP set is inspected and recognized and the parameters are used to define a one time only TFTP proxy rule, specific to the target device. The SNMP packet is then updated to make the ZoneRanger the destination for the transfer. When the target device uses the ZoneRanger for the TFTP transfer, the ZoneRanger finds the single use rule and proxies the transfer to the actual destination originally specified in the SNMP set.

**Note:** This feature is triggered by sets using the CISCO-CONFIG-COPY-MIB (Cisco IOS software release 12.0) and the OLD-CISCO-SYSTEM-MIB/OLD-CISCO-FLASH-MIB (Cisco IOS software release 10.2 and later).

The **SNMP triggered rules timeout** field specifies the maximum life span for the SNMP triggered single use rules. After the timeout has expired, the rule will be discarded.

The **TFTP Proxy Logging** levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Basic information about each TFTP request, including auto-generated single use rule creation/usage/expiration events.</td>
</tr>
<tr>
<td>Full</td>
<td>Additional information, such as the rule used for each TFTP request</td>
</tr>
</tbody>
</table>

This TFTP Proxy log can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file may also be viewed on the **View > Service Logs** page. The log file is called `/log/tftp.log`.

**Configuring NTP Proxy**

ZoneRanger has the capability to proxy NTP traffic from a ZoneRanger managed device to the Ranger Gateway and return the response. The **Configuration > Inbound Proxy** page **NTP** tab allows for the configuration of the NTP Proxy Servers as well as the logging of NTP proxy requests.
The Proxy NTP Servers section is used to create the list of Ranger Gateway/NTP Servers pairs to process NTP requests from ZoneRanger managed devices. In the case of multiple entries in the list, ZoneRanger will continue to use a Ranger Gateway/NTP Server pair that successfully responds to NTP requests. If a Ranger Gateway/NTP Server fails to respond, another entry in the list is chosen.

This Configuration > Proxy page NTP tab will be disabled if the ZoneRanger has been configured to act as an NTP server on the Configuration > System page Time tab using the ZoneRanger Acts As NTP Server checkbox.

The NTP Proxy Logging levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Message header is logged</td>
</tr>
<tr>
<td>Full</td>
<td>Entire message is logged</td>
</tr>
</tbody>
</table>

This NTP Proxy log can be downloaded by the downloadFile command on a Ranger Gateway. The log file is called /log/ntpProxy.log. The log file may also be viewed on the View > Service Logs page.

If Validate Authentication is checked, all NTP proxy requests are first validated to ensure that they match one of the listed keys. This is useful to eliminate spurious NTP requests from unsecure devices. The Keys section is used to create the list of specific NTP keys ZoneRanger can use to authenticate the incoming NTP request.

You can use the Show Advanced button to access and configure the following advanced options:
### Advanced Option | Description
--- | ---
Client Timeout | Amount of time in seconds a ZoneRanger waits for a message from a NTP client before closing the TCP connection for that client.
Server Timeout | Amount of time in seconds a ZoneRanger or Ranger Gateway waits for a response from a NTP server

**Configuring Secure TCP Proxy**

ZoneRanger has the capability to proxy TCP traffic from a ZoneRanger managed device to the Ranger Gateway and return the response. The Configuration > Inbound Proxy page Secure TCP tab allows for the configuration of TCP Ports as well as the logging of TCP proxy requests. SSH on the Configuration -> System -> Ports Tab must be set to “Disabled” or “Ranger Gateway Only” before Secure TCP can be proxied.

![Configuration > Inbound Proxy page](image)

**Figure 35-41. Configuration > Node Management page**

Before configuring Secure TCP proxy rules, at least one TCP category must be created. Categories are used to group like proxy rules together for logging and statistical purposes. The category value associated with a proxy rule is included in each log entry in the inboundTcpProxy.log file on the ZoneRanger and the inboundTcpProxy.log file on the Ranger Gateway. The category value is also used to sub-divide the TCP (Inbound) usage statistics on the View Statistics page.

The **Proxy Rules** section manages how inbound secure TCP Proxy connections are treated. Each entry defines the list of possible application servers a given connection should use for proxy. To create a new entry, click the **Add** button, fill in the **Source Address**, **Destination**, **Category** and click the **Edit** button to select at least one management application server and port. The **Category** field defines the category used for Inbound TCP statistics and logging. At least one Category must be created before a proxy rule can be added.

The order of the **Proxy Rules** is important. ZoneRanger uses this list by selecting the first matching rule, starting from the top of the list. Use the **Move Up** and **Move Down** buttons to change the order.

**Note:** The **Source Address** may be a hostname, IP address, IP address range, or a node group.
4. A range ( [10-70] ) or wildcard ( * ) can be specified for any octet in the IP address. Example: 10.50.[5-15].*

5. Node group names must be prefixed with '@'. For example, @webServers

**Note:** Port 22 Tcp Proxy requires that the SSH Port is disabled for eth0 and eth0 interfaces. SSH to the ZoneRanger via the Ranger Gateway can remain enabled if desired.

The **Secure TCP Proxy Logging** levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Message header is logged</td>
</tr>
<tr>
<td>Full</td>
<td>Entire message is logged</td>
</tr>
</tbody>
</table>

This Secure TCP Proxy log can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/inboundTcpProxy.log`. The log file may also be viewed on the **View > Service Logs** page.

**Node Management**

ZoneRanger proxies requests and forwards information if the information references a managed node in its databases. The **Configuration > Node Management** page is used to configure device and management information for ZoneRanger.

![Figure 35-42. Configuration > Node Management page](image)

### Configuring managed nodes

The **Configuration > Node Management** page **Managed Nodes** tab enables the ability to set which nodes are managed within ZoneRanger. When a node is moved from the **Managed** list to the **Unmanaged** list and the **Save** button is clicked, ZoneRanger management services become unavailable for that node. For example, the node, along with its interfaces and TCP ports, is no longer polled. The node does not appear in the Root Cause report, and its status color becomes blue. An unmanaged node cannot be accessed using SNMP proxy. Traps, syslog, NetFlow, sFlow, and general UDP data received from unmanaged nodes is not forwarded.
Removing managed nodes

The Configuration > Node Management page Node Removal tab enables the ability to delete nodes from the ZoneRanger database. When a node is removed, ZoneRanger management services become unavailable to that node. For example, the node, along with its interfaces and TCP ports, is no longer polled. The node does not appear in the Root Cause report, and is not included in status bars. A node that is removed will not be sent any proxy requests via ZoneRanger. Any data received from nodes that have been removed will not be forwarded.

Node removal is appropriate for nodes which no longer exist, or will otherwise not be discovered again. Nodes for which database information is no longer required, but which will continue to be discovered, should be unmanaged rather than removed.

After removing a node, you should check any configuration pages where specific IP addresses or hostnames might be specified (for example, Configuration > Forwarding, Configuration > Polling, etc) and remove or modify any rules that refer to the removed node.

Configuring node groups

Node Groups represent a collection of address patterns that can be applied to IP based configurations such as forwarding rules and proxy rules. The Configuration > Node Management page Node Groups tab is used to create, delete, and modify Node Groups. Node Groups may contain an arbitrary number of valid address patterns as well as other Node Groups. When specifying a Node Group as part of a configuration or within another Node Group, the name of the Node Group must be prefixed with '@'. For example, the Node Group webservers would be represented in configurations as @webservers.

Note that Node Groups may not contain hostnames.

Configuring device types

During discovery, ZoneRanger uses SNMP sysObjectId to determine the device type of each node. The Configuration > Node Management page Device Types tab enables you to manually assign device types to discovered nodes. If the device type of a router or switch was manually changed to "server," its device type might change back to the actual type when discovery is performed again.

Outbound Proxy

The Configuration > Outbound Proxy page is used to customize the configuration of various proxy services including TCP, FTP, and ICMP.

Configuring TCP Proxy

The Configuration > Outbound Proxy page TCP Proxy tab allows for the configuration of the level of logging for TCP proxy services.
TCP Proxy Logging specifies the level of logging for TCP proxy requests through the ZoneRanger. This can affect the performance of TCP proxy.

The TCP Proxy Logging levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Basic information about each TCP proxy transaction request is logged, including entries for Opened, Closed, and Failed events.</td>
</tr>
<tr>
<td>Full</td>
<td>Additional information is added to the log, including: An Opening event entry as the proxy is connecting, enhanced proxy path details with ports.</td>
</tr>
</tbody>
</table>

The TCP Proxy logging level applies for all TCP proxied connections between the Ranger Gateway and the ZoneRanger which include HTTP, HTTPS, Telnet, SSH, and FTP. This TCP Proxy log can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/tcpProxy.log`. The log file may also be viewed on the View > Service Logs page.

**Configuring FTP Proxy**

ZoneRanger is capable of proxying FTP connections initiated through the Ranger Gateway to ZoneRanger managed nodes. The Configuration > Outbound Proxy page FTP tab provides configuration for those connections.
The **Enable Active-to-Passive Translation** checkbox enables TCP proxy sessions processed as FTP to translate "Active" data transfer requests to "Passive" data transfer requests. The "Passive" (PASV) mode is a more secure form of data transfer, allowing the FTP client to initiate all connections. For older FTP clients that do not support PASV, enabling this option will allow them to operate with FTP servers that require PASV.

**Configuring ICMP Proxy**

ZoneRanger has the capability to proxy ICMP requests through the Ranger Gateway to a ZoneRanger managed device and return the response. The **Configuration > Proxy** page **ICMP** tab allows for the configuration of the ICMP caching of those responses as well as the logging of ICMP proxy requests.

![ZoneRanger dashboard](image)

**Figure 35-45. Configuration > Outbound Proxy page ICMP tab**

The **ICMP Proxy Logging** levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Basic information about each ICMP packet is logged</td>
</tr>
<tr>
<td>Full</td>
<td>Information about each ICMP packet is logged</td>
</tr>
</tbody>
</table>

This ICMP Proxy log can be downloaded by the **downloadFile** command on a Ranger Gateway. The log file is called `/log/icmpProxy.log`. The log file may also be viewed on the **View > Service Logs** page.

ZoneRanger can be configured to cache (store) previous ICMP responses to be used for subsequent ICMP requests within a particular time period. The **ICMP Proxy Cache Enabled** checkbox enables this capability.

The **Caching Rules** section allows for the creation of rules to indicate which addresses should have their ICMP responses cached based on the following information:
**Setting** | **Description**
---|---
Address | Source address of the incoming ICMP request. Address may be an address pattern or Node Group.
Positive Cache Enabled | Store successful ICMP responses
Time to Cache | Length of time to store successful ICMP responses
Time Units | Units of time to store successful ICMP responses
Negative Cache Enabled | Store unsuccessful ICMP responses
Time to Cache | Length of time to store unsuccessful ICMP responses
Time Units | Units of time to store unsuccessful ICMP responses

If an address matches more than one entry then the first entry in the table that it matches is used.

The **ICMP Proxy Cache Logging** levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Basic information about each proxy cache transaction request including if it is configured to be cached or is a cache hit or miss.</td>
</tr>
<tr>
<td>Full</td>
<td>Additional information, including: which Ranger Gateway a request came from and whether the response is positive or negative.</td>
</tr>
</tbody>
</table>

This ICMP Proxy Cache log can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/icmpProxyCache.log`. The log file may also be viewed on the View > Service Logs page.

**Peers**

Multiple ZoneRangers may be configured to work in concert to achieve greater management reliability. The **Configuration > Peers** page is used to configure multiple ZoneRangers to work in this environment.

![Configuration > Peers page Group tab](image)
Configuring the ZoneRanger group

On the Configuration > Peers page Group tab, the Group Name is used to filter duplicate information from multiple ZoneRangers reporting to the same Ranger Gateway. Redundant ZoneRangers always have the same group name. You can give multiple nonredundant ZoneRangers the same group name if they cannot become redundant, but filtering of duplicate information is still desired.

Configuring redundancy

Redundancy links two or more ZoneRangers so that services can be provided continuously, even if one of the ZoneRangers become unavailable. Redundant ZoneRangers share configuration information and database information. The Configuration > Peers page Redundancy tab is used to establish redundant ZoneRangers.

![Configuration > Peers page Redundancy tab](image)

Figure 35-47. Configuration > Peers page Redundancy tab

When configuring redundancy, the ZoneRanger becoming redundant (the target ZoneRanger) gets its settings and information from the source ZoneRanger specified in the Source ZoneRanger for Redundancy field. The Polling Interval defines how frequently each redundant ZoneRanger verifies the availability of all the other redundant ZoneRangers. The passcodes must match between the source and target ZoneRangers. Enabling redundancy will result in a restart of the target ZoneRanger.

Once redundancy has been established between ZoneRangers, the Configuration > Peers page Redundancy tab will list all redundant ZoneRangers and the status of each ZoneRanger.

Configuring virtual IP

A virtual IP address may be shared by redundant ZoneRangers. A virtual IP address is a secondary IP address which one of the redundant ZoneRangers is configured to support. If that ZoneRanger becomes unavailable, another ZoneRanger will automatically begin supporting that IP address. This functionality allows for ZoneRanger managed devices to be configured with a single IP address for forwarding rather than the primary ZoneRanger IP address. Thus, if the primary IP address of the ZoneRanger needs to be changed, the ZoneRanger managed devices will not need to be updated with a new IP address. The Configuration > Peers page Virtual IP tab is used to establish the virtual IP address of the redundant ZoneRangers.
The virtual IP address may be created on either eth0 or eth1 on the ZoneRanger but that must be consistent on all redundant ZoneRangers.

The **Port** is used by the redundant ZoneRangers to communicate the availability of the virtual IP address. All messages sent between the redundant ZoneRangers on this port are encrypted. The **Port** must be consistent on all redundant ZoneRangers.

The **Heartbeat Interval** is the frequency the redundant ZoneRangers check for the availability of the virtual IP address. It must be greater than 1 second. The **Heartbeat Interval** must be consistent on all redundant ZoneRangers.

The **Heartbeat Timeout** is the amount of time the ZoneRanger will wait for a positive response indicating the availability of the virtual IP address. If this timeout is reached, another redundant ZoneRanger will assume control of the virtual IP address. The **Heartbeat Timeout** must be at least twice the **Heartbeat Interval**.

**Polling**

ZoneRanger can be used to status poll specific nodes, interfaces and TCP ports. ZoneRanger polling can be configured on the **Configuration > Polling** page.
Configuring polling for a single node

The **Enable/Disable** tab displays a list of all managed nodes sorted by IP address or fully qualified host name. When a single node or IP address is selected, polling for that individual node may be configured. If the **Polling Enabled checkbox** is unchecked, all polling for that node is disabled.

Configuring polling for multiple nodes

To configure polling for multiple nodes, select the desired nodes in the list. All of the available buttons will then apply to all of the selected nodes.
Figure 35-51. Configuration > Polling page Interface Settings tab

Configuring polling settings

The Configuration > Polling page Interface Settings tab displays a table of interface polling settings groups. Settings groups may be assigned to a specific interface, node, or group of interfaces whose IP addresses match a specified pattern.

Settings groups specify the following information:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td>An IP address pattern, Node Group, the IP address of an interface, or the fully qualified hostname of a node. The address pattern, address, Node Group, or hostname identifies the interface, set of interfaces, or node to which settings in the group are applied.</td>
</tr>
<tr>
<td>Polling Interval</td>
<td>Time interval, in seconds, between polling cycles</td>
</tr>
<tr>
<td>ICMP Timeout</td>
<td>Length of time that the poller waits for an ICMP response</td>
</tr>
<tr>
<td>ICMP Retries</td>
<td>Number of times that polling is attempted before an ICMP failure is reported</td>
</tr>
</tbody>
</table>

Settings groups provide a convenient way to associate interface polling settings with individual interfaces, or common interface polling settings with groups of interfaces (for example, all interfaces on a specified node, or all interfaces with IP addresses matching a specified wild card pattern).

The interface polling settings for a given interface are identified by searching through the set of settings groups in order (from top to bottom as they are displayed in the table) until the first match is found. Therefore, the order in which the settings groups are shown in the table is very important. The arrows may be used to change the order of the settings groups.

The Polling Interval applies to both ICMP and SNMP polling. SNMP timeouts and retries may be configured on the Configuration > SNMP page Managers tab.
Configuring TCP Polling Settings

The Configuration > Polling page TCP Settings tab configures general TCP port polling behavior on the ZoneRanger. You can configure the following aspects of TCP port polling:

- The default polling interval for all TCP services and all nodes. If no polling interval is set for a specific service, the default polling interval is used.
- Polling intervals for specific services can be configured for all nodes.
- Polling for specific services (that is, name and port number) can be enabled or disabled for all nodes.
- TCP port status propagation for specific services can be enabled or disabled for all nodes. If status propagation is enabled, TCP port status affects node status. Thus, if polling fails for one or more TCP ports, in the absence of other more critical issues, the node is degraded to the marginal state.

**Note:** Ports in the list are those that were previously configured in the Configure > Discovery page TCP Ports tab.

The Default Polling Interval field specifies the time interval, in seconds, between TCP port polling cycles. The default polling interval applies to all TCP ports on all nodes unless an override is configured.

Figure 35-52. Configuration > Polling page TCP Settings tab
Ranger Gateway

Joining Ranger Gateways

The Configuration > Ranger Gateway Configuration page is used to configure communications between ZoneRangers and Ranger Gateways. This configuration is used to join and unjoin ZoneRangers and Ranger Gateways, to configure SSL trust between ZoneRanger and Ranger Gateways, and to restrict the addresses to which the ZoneRanger can initiate connections.

![Figure 35-53. Configuration > Ranger Gateway page](image)

The Ranger Gateway section is used display currently joined ranger gateways and provide a utility for joining to new ones. Users may use the add feature and enter Gateway addresses and passcodes to initiate new join requests. When the page is saved, the ZoneRanger will attempt to join any new Ranger Gateways entered via this process. Any joined gateway will then be displayed under the Joined Ranger Gateways table providing what the current Passcode of that gateway is and the current version of the software it is running.

The passcode for any joined Gateway will display under the Gateway Passcode column. The Refresh button can be used to refresh this data and display the most up to date passcode on that Gateway.

The final field is used for displaying and changing the Zone Rangers passcode. This is the passcode needed if a gateway is to initiate the join request to a Zone Ranger.

The ZoneRanger passcode can be changed to a different value without affecting the join status of Ranger Gateways that were joined previously.

The View > System Log may provide additional information about why the join attempt was unsuccessful.
Configuring Management Application Servers

The Mgmt App Servers tab allows users to configure management applications servers that can be reached via traffic through a Ranger Gateway. These configured management application servers can then be used by the ZoneRanger within the configuration of Inbound TCP Proxy. To add a new Management Application Server, first enter the IP address or the Hostname of the server and press the Add button. This will allow for the selection of which Ranger Gateway traffic must flow through in order to reach the Management Application Server.

Configuring Ranger Gateway restrictions

The ZoneRanger may be configured to restrict the list of Ranger Gateway addresses to which it can attempt to join. The Configuration > Ranger Gateway page Restrictions tab may be used to list these addresses.
A "messaging connection" is an SSL connection used to allow secure communication between a ZoneRanger and a Ranger Gateway, or between redundant ZoneRangers. The **Restricted Addresses** section is used to prevent the ZoneRanger from initiating messaging connections to specified addresses or address ranges. Restricted address may be specified as address patterns.

The ZoneRanger will accept incoming messaging connections regardless of any configured restrictions. For example, if the address corresponding to a Ranger Gateway is restricted, the Ranger Gateway will be allowed to initiate a messaging connection to the ZoneRanger, but the ZoneRanger will not be allowed to initiate a connection to the Ranger Gateway.

The typical application of restricted addresses is the case where a ZoneRanger is located in a DMZ, the Ranger Gateway is located on the other side of a firewall, and security policy dictates that all connections through the firewall be initiated from outside the DMZ.

**Configuring SSL Trust between ZoneRanger and Ranger Gateway**

All communication between Ranger Gateways and ZoneRangers is protected using SSL, in an effort to authenticate the communicating entities, and to ensure that the information being communicated remains confidential. The **Configuration > Ranger Gateways page SSL Trust** tab establishes the credentials needed for a Ranger Gateway to communicate with the ZoneRanger.

![Figure 35-56. Configuration > Ranger Gateway page SSL Trust tab](image)

The SSL configuration on a ZoneRanger or Ranger Gateway consists of two parts:

1. Configuring a ZoneRanger or Ranger Gateway with private encryption keys, and with corresponding certificates that it will use to identify itself, and to pass public encryption key material to other entities.

2. Configuring a ZoneRanger or Ranger Gateway with the identities or "trusted subjects" with which it is authorized to communicate.

By default, each ZoneRanger is configured with a certificate issued by the Tavve internal certificate authority, with the following subject (identity):
Similarly, each Ranger Gateway is configured with a certificate with the following subject:

CN = Ranger Gateway, OU = Engineering, O = Tavve, L = Morrisville, ST = North Carolina, C = US

ZoneRangers are configured, by default, to permit communication with both subjects, in order to support communication with joined Ranger Gateways, and with redundant peers.

To authorize a ZoneRanger to communicate with a device that was configured with a particular SSL certificate, you must enter that certificate subject into the ZoneRanger Subjects table.

The Trusted Certificate Authorities section lists the certificate authorities to use for authentication of a Ranger Gateway connection. If the Ranger Gateway private key is not signed by one of the trusted certificate authorities, the connection will be rejected. The Add Certificate Authority section may be used to install additional trusted certificate authorities. The certificate must be either an X.509 Certificate or a JKS Keystore format.

**Root Cause**

ZoneRanger includes a sophisticated root cause analysis service, which when triggered by status polling failures, that determines which device is the root cause of the problem, and which devices are impacted by the root cause device.

The root cause service divides root cause analysis into two categories: IP and TCP. The Configuration > Root Cause page is used to configure the reporting of IP and TCP root causes.

**Configuring IP Root Cause**

ZoneRanger is automatically configured to determine the root cause of an IP outage and generate the associated SNMP Trap. ZoneRanger can also be configured to send an email after determining the root cause of an outage. You can configure the following settings with respect to root cause outages:
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Description to be used as the subject in trap notifications.</td>
</tr>
<tr>
<td><strong>Email Notification Recipients</strong></td>
<td>Email addresses for those who receive email notifications. Use commas or semicolons to separate multiple addresses.</td>
</tr>
<tr>
<td><strong>Email Return Address</strong></td>
<td>&quot;Reply-To&quot; address for email notifications. Use an address that helps you to identify the reporting ZoneRanger, such as ZoneRanger1@dmz1. <strong>Note:</strong> The domain in this address cannot be localhost, which Ranger Gateway does not recognize.</td>
</tr>
<tr>
<td><strong>Ranger Gateway for sending Email Notifications</strong></td>
<td>Ranger Gateway through which to forward email. If no Ranger Gateway is selected, the ZoneRanger sends the notifications.</td>
</tr>
</tbody>
</table>

The Ranger Gateway sends its email destination to the configured mail server. The **Send Test Email** button can be used to verify that the configuration parameters are correct.

The **Show Advanced Options** button can be used to specify the actions that ZoneRanger takes to verify the status of a device or interface. After verification, ZoneRanger generates a trap and sends notification emails.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Down Verify Pings</strong></td>
<td>Number of times that a device or interface is pinged before it is considered to be down.</td>
</tr>
<tr>
<td><strong>Down Verify Time</strong></td>
<td>Number of seconds during which a device or interface is pinged before it is considered to be down.</td>
</tr>
<tr>
<td><strong>Up Verify Pings</strong></td>
<td>Number of times that a device or interface previously reported to be down must be successfully pinged before it is considered to be up.</td>
</tr>
<tr>
<td><strong>Up Verify Time</strong></td>
<td>Number of seconds during which a device or interface is pinged before it is considered to be up.</td>
</tr>
</tbody>
</table>
ZoneRanger is automatically configured to determine the root cause of an TCP outage and generate the associated SNMP Trap. ZoneRanger can also be configured to send an email after determining the root cause of an outage. You can configure the following settings with respect to root cause outages:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description to be used as the subject in trap notifications</td>
</tr>
<tr>
<td>Email Notification Recipients</td>
<td>Email addresses for those who receive email notifications. Use commas or semicolons to separate multiple addresses</td>
</tr>
<tr>
<td>Email Return Address</td>
<td>“Reply-To” address for email notifications. Use an address that helps you to identify the reporting ZoneRanger, such as ZoneRanger1@dmz1. Note: The domain in this address cannot be localhost, which Ranger Gateway does not recognize.</td>
</tr>
<tr>
<td>Ranger Gateway for sending Email Notifications</td>
<td>Ranger Gateway though which to forward email. If no Ranger Gateway is selected, the ZoneRanger sends the notifications.</td>
</tr>
</tbody>
</table>

The Ranger Gateway sends its email destination to the configured mail server. The Send Test Email button can be used to verify that the configuration parameters are correct.

The Show Advanced Options button can be used to specify the actions that ZoneRanger takes to verify the status of a TCP port. After verification, ZoneRanger generates a trap and sends notification emails.
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down Verify Pools</td>
<td>Number of times that a TCP port is polled before it is considered to be down</td>
</tr>
<tr>
<td>Down Verify Time</td>
<td>Number of seconds during which a TCP port is polled before it is considered to be down</td>
</tr>
</tbody>
</table>

**SNMP**

ZoneRanger makes extensive use of the SNMP protocol to both manage devices and to allow other applications to access ZoneRanger managed devices. The **Configuration > SNMP** page is used to configure various SNMP settings that the ZoneRanger uses to make SNMP requests to network devices.

```
Figure 35-59. Configuration > SNMP page
```

### Configuring SNMP Options

The **Configuration > SNMP** page **Options** tab is used to specify the level of logging for SNMP proxy requests and responses on the ZoneRanger, whether or not SNMPv3 users must be configured in order to proxy SNMPv3 requests, and can be used to specify an optional SNMP configuration file used to upload target rules and user information.

The **SNMP Proxy Logging** levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Basic information about each SNMP packet is logged</td>
</tr>
<tr>
<td>Full</td>
<td>Information about each SNMP packet is logged, including all variable bindings</td>
</tr>
</tbody>
</table>

This SNMP Proxy log can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/snmpProxy.log`. The log file may also be viewed on the **View > Service Logs** page.
The **Require SNMPv3 users to be configured for proxy requests** checkbox determines whether or not an SNMPv3 user must be configured on the ZoneRanger in order to proxy SNMPv3 requests. If checked, when the ZoneRanger receives a SNMPv3 proxy request, a valid SNMPv3 user must be configured in order for the ZoneRanger to proxy the request. If no valid SNMPv3 user is found, the proxy request is discarded. If unchecked, when the ZoneRanger receives an SNMPv3 proxy request, it will attempt to validate the request using the configured SNMPv3 users. However, if no valid SNMPv3 user is found, the ZoneRanger will still proxy the request. Note that SNMP caching is only possible for SNMPv3 requests with configured users or if the proxy request uses noAuthNoPriv Security Level.

The ZoneRanger is only able to process an incoming SNMPv3 Inform if there is a configured SNMPv3 user or the Inform is using noAuthNoPriv Security Level. When the ZoneRanger is able to process an incoming SNMPv3 Inform, the ZoneRanger will convert the Inform to an SNMPv3 Trap, forward the trap based on any configured forwarding rules, and respond to the client that the Inform was received. ZoneRanger can forward SNMPv3 traps which use any Security Level regardless of whether or not there is a configured SNMPv3 user.

The **Require SNMPv3 users to be configured for notifications** checkbox determines whether or not an SNMPv3 user must be configured on the ZoneRanger in order to validate SNMPv3 traps and informs. If checked, when the ZoneRanger receives a SNMPv3 trap or inform, a valid SNMPv3 user must be configured in order for ZoneRanger to process the notification. If no valid SNMPv3 user is found, the trap or inform is discarded. If unchecked, when the ZoneRanger receives an SNMPv3 trap or inform, it will attempt to validate the notification using the configured SNMPv3 users. However, if no valid SNMPv3 user is found, the ZoneRanger will still process the notification.

However, there are some limitations when SNMPv3 users are not configured for SNMPv3 traps and informs:

1. The type of notification (trap or inform) cannot be determined for encrypted notifications.
2. Encrypted notifications will not match any trap filters using properties of the PDU with the exception of version.
3. The ZoneRanger will not return responses to the client when it receives an SNMPv3 Inform.
4. Duplicate encrypted notifications will not be discarded on the Ranger Gateway.

Some users may prefer to use an external management application, such as CiscoWorks, to manage SNMPv3 configuration for DMZ devices. In such cases, you can export the SNMPv3 configuration from the management application, convert the information to the Tavve-specified XML format, and upload the resulting file to the ZoneRanger.

The uploaded configuration information updates existing target rules and users and adds new target rules and users. An example (`snmp-rules.xml`) and the schema (`snmp-rules.xsd`) can be found on the Ranger Gateway in the ZRCustom Directory. Uploading a new SNMP rules files adds new rules and modifies existing rules. The new file does not replace the previous set of rules.

**Configuring SNMP Manager information**

The **Configuration > SNMP** page **SNMP Manager** tab displays a list of target rules that contain parameters that ZoneRanger requires to obtain SNMP information from managed devices.
The **Target Rules** section lists the SNMP parameters used when making SNMP requests to ZoneRanger managed devices. The order of target rules is important. ZoneRanger selects the first matching rule in the list, starting at the top. Use the arrow buttons to change the order of the rules.

Discovery and SNMP polling use target rule parameters when sending SNMP Requests to devices. The first rule whose **Target** matches the device address is used.

SNMP requests proxied through a Ranger Gateway use target rule parameters to govern conversion between SNMP versions. When a proxied request is received from a Ranger Gateway, the first rule whose **Target** matches the destination address and whose **Community** matches the request community is used. If no rule is matched, SNMPv1 is used; that is, SNMPv2c requests are converted to SNMPv1. Wildcards, specified by *, may be used at the end of the community string. When using wildcards, the preceding portion of the community string will be used for matching. If there is a match, the entire request community string is used.

Trap forwarding converts all SNMPv3 notifications to SNMPv2c or SNMPv1. The target rules determine the community string to use for this conversion. The first rule that matches the source address and user name from the SNMPv3 trap determines the community string in the forwarded trap.

The **Target Rule** parameters are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target</strong></td>
<td>Target IP address of the SNMP request. Target may be an address pattern or Node Group.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>SNMP version to use</td>
</tr>
<tr>
<td><strong>User</strong></td>
<td>SNMPv3 User</td>
</tr>
<tr>
<td><strong>Community</strong></td>
<td>Community String</td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
<td>Length of time ZoneRanger waits for a response to an individual SNMP request.</td>
</tr>
<tr>
<td><strong>Retries</strong></td>
<td>Number of times to retry an SNMP request</td>
</tr>
<tr>
<td><strong>Remote Port</strong></td>
<td>Port number on the host used for SNMP queries</td>
</tr>
</tbody>
</table>
Configuring SNMPv3 Users

In order to validate communications with SNMPv3 agents, ZoneRanger must be configured with a set of SNMPv3 Users. The Configuration > SNMP page Users tab is used to manage SNMPv3 users.

The SNMPv3 Users section is the set of entries each of which contains the following:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>SNMPv3 user name</td>
</tr>
<tr>
<td>Security Level</td>
<td>noAuthNoPriv; authentication and privacy (encryption) are not applied. authNoPriv; only authentication is applied. authPriv; authentication and privacy are applied.</td>
</tr>
<tr>
<td>Auth Protocol</td>
<td>Authentication protocol</td>
</tr>
<tr>
<td>Auth Password</td>
<td>Authentication password used if authentication is applied.</td>
</tr>
<tr>
<td>Privacy Protocol</td>
<td>Encryption Protocol</td>
</tr>
<tr>
<td>Privacy Password</td>
<td>Encryption password used if privacy is applied.</td>
</tr>
<tr>
<td>Context Name</td>
<td>SNMPv3 context name to use with this user. This field is optional.</td>
</tr>
</tbody>
</table>

Note: All user names must be unique, and all passwords must contain at least eight characters.

Using the SNMP Access Test

The Configuration > SNMP page Access Test tab enables you to test different combinations of SNMP versions, community string values, and user parameters on ZoneRanger devices.
The `Show Accessible` and `Show Disabled` check boxes control which nodes are listed as testable. The `Show Accessible` check box includes all the SNMP accessible nodes. The `Show Disabled` check box includes all the disabled targets under the `Disallowed` tab.

You can use the `Community Strings` and `Users` tables to specify community strings and users. The `Defaults` button will initialize the list of community strings or users with those already configured for other nodes.

After the test finishes, you can click `Update Configuration` to automatically add configuration rules to the Manager and User tables, based on successful test results.

**Configuring the SNMP Preferred Address**

Some devices having multiple IP interfaces might be configured so that only one IP interface responds to SNMP requests. You can use the `Configuration > SNMP page Preferred Address` tab to configure the IP address that the ZoneRanger uses when sending SNMP requests on behalf of services such as SNMP polling and SNMP proxy.

When SNMP accessible nodes are first discovered, a default preferred SNMP interface is chosen for each node. To modify the preferred SNMP interface, select a node in the `Nodes` list and use the appropriate radio button to select the IP address.
Configuring the SNMP disallowed list

There may be managed devices to which ZoneRanger should not make any SNMP requests. The Configuration > SNMP page Disallowed tab can be used to list the devices which ZoneRanger should not query via SNMP. This list takes precedence over the rules on the Manager tab.

In the Disallowed Targets section, list the devices to which ZoneRanger should not make SNMP requests. The Address may be an address pattern or Node Group.

For some devices, it is desirable to prevent a set of SNMP OIDs from being accessed via SNMP Proxy. In the Configure disallowed OIDs for SNMP proxy section, a set of Targets, OID trees, and whether or not to disallow an Get or Set to those Targets. When an SNMP Proxy Get or Set request is received with a disallowed OID, the OID is removed from the request before the request is forwarded to the target. When the SNMP response is received from the target, the disallowed OIDs are added to the response and the value of the OID is set to null. The SNMP error status is set to NoSuchName and the error index is set to the index of the first disallowed OID.

For SNMP v3 requests, the request will be evaluated only if the user is defined or the request uses the security level noAuthNoPriv.

SNMP GetNext and GetBulk requests are not supported.

The Target Rule parameters are:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>Target IP address of the SNMP request. Target may be an address pattern or Node Group.</td>
</tr>
<tr>
<td>OID</td>
<td>SNMP OID to disallowed. The requested OID matches the disallowed OID if the requested OID starts with the disallowed OID.</td>
</tr>
<tr>
<td>Disallow Get</td>
<td>Disallowed SNMP Get Requests</td>
</tr>
<tr>
<td>Disallow Set</td>
<td>Disallowed SNMP Set Requests</td>
</tr>
</tbody>
</table>

**Configuring the ZoneRanger SNMP Agent**

ZoneRanger provides information via its read-only SNMP agent. The **Configuration > SNMP** page **Agent** tab can be used to configure specific SNMP agent information.

![Figure 35-65. Configuration > SNMP page Agent tab](image)

The **Community String** defines the community string to respond to when using SNMPv1 or SNMPv2c. The **Users** list defines which users the ZoneRanger agent will respond to when using SNMPv3. The users are defined on the **Configuration > SNMP** page **Users** tab. To disable the SNMP agent on the ZoneRanger, uncheck all three versions next to **Agent Responds To**.

**Configuring the SNMP Proxy Cache**

ZoneRanger has the capability to proxy SNMP requests through the Ranger Gateway to a ZoneRanger managed device and return the response. The **Configuration > SNMP** page **Proxy Cache** tab allows for the configuration of the SNMP caching of those responses as well as the logging of SNMP proxy requests.
ZoneRanger can be configured to cache (store) previous SNMP responses to be used for subsequent SNMP requests within a particular time period. The **SNMP Proxy Cache Enabled** checkbox enables this capability.

The **SNMP Proxy Cache Logging** levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Basic information about each proxy cache transaction.</td>
</tr>
<tr>
<td>Long</td>
<td>Information about each requested OID and identity.</td>
</tr>
</tbody>
</table>

This SNMP Proxy Cache log can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/snmpProxyCache.log`. The log file may also be viewed on the **View > Service Logs** page.

The **Caching Settings** section allows for the creation of rules to indicate which addresses should have their SNMP responses cached based on the following information:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OID</td>
<td>OID of SNMP request beginning with the listed OID</td>
</tr>
<tr>
<td>Cache</td>
<td>Whether or not cache this particular OID</td>
</tr>
<tr>
<td>Time to Cache</td>
<td>Length of time to store successful SNMP responses</td>
</tr>
<tr>
<td>Time Units</td>
<td>Units of length of time to store successful SNMP responses</td>
</tr>
</tbody>
</table>

If an OID in the SNMP request matches more than one entry then the first entry in the table that it matches is used. Thus if a more specific OID entry is meant to override a less specific entry then the more specific entry should be listed first. For example to cache all of the System table except for sysUpTime then first add an entry for OID .1.3.6.1.2.1.1 set to not cache and then add one for .1.3.6.1.2.1.1.3 set to be cached.
System

After ZoneRanger is installed, configured, and running, Configuration > System page can be used to modify the ZoneRanger system configuration.

Figure 35-67. Configuration > System page IP tab

Modifying the network settings

ZoneRanger has two Ethernet interfaces which may be configured using the Configuration > System page IP tab.

By default, the Ethernet interfaces on the ZoneRanger are configured to automatically set the interface speed. However, if necessary, the interface speed and duplex type may be specified on the IP tab.

If the ZoneRanger has a ZR-SPX license, it can be configured to use an IPv6 address. Check the Enable IPv6 checkbox to enable the ZoneRanger to use IPv6 addresses. A default route for all interfaces and static IPv6 addresses for each interface can be configured. The IPv6 address must be specified in CIDR format. For example, IPv6 block 2001:db8::/48 represents the IPv6 addresses from 2001:db8:0:0:0:0:0:0 to 2001:db8:0:ffff:ffff:ffff:ffff:ffff.

If the ZoneRanger is connected to a 802.1q VLAN trunk, the Connect to VLAN trunk check box should be selected. Primary VLAN ID defines the VLAN id the ZoneRanger should use for its communications. Use blank or 0 (zero) as the Primary VLAN ID to denote native mode. Use the Add VLAN button to add each VLAN that the ZoneRanger will participate in.

Note: Saving any changes on this tab results in a ZoneRanger restart.
Modifying DNS settings

ZoneRanger may be configured to use DNS servers for name resolution as well as to act as a Secondary DNS server for managed devices. The **DNS Servers** section lists the set of DNS servers ZoneRanger should use for name resolution. The **Search Domains** section lists the search domains, if any, used to resolve unqualified names. The **Hostnames** section provides a mechanism to map individual IP addresses to hostnames. This is useful when no DNS server is available. If the **Secondary DNS enabled** check box is checked, ZoneRanger acts as a caching DNS server.

**Note:** Saving any changes on this tab results in a ZoneRanger restart. You must run discovery to update the ZoneRanger database to reflect any DNS configuration changes.

Setting the ZoneRanger time

The **Configuration > System** page **Time** tab is used to set the ZoneRanger time-of-day clock and optionally, to act as an NTP server for its managed devices. ZoneRanger can either use its local time or it can synchronize with an external source via the Ranger Gateway, a Time Server which supports RFC 868, or an NTP server.
When specifying **Time Server** in the **Server Type** field, enter the IP address or fully qualified hostname of a time server supporting RFC 868 in the **Server** field.

When specifying **Ranger Gateway** in the **Server Type** field, ZoneRanger will synchronize its time with that of the Ranger Gateway specified in the **Server** field.

When specifying **NTP** in the **Server Type** field, ZoneRanger will synchronize its time with at least one NTP server listed in the **NTP Servers** table. When multiple NTP servers are listed in the **NTP Servers** table, ZoneRanger will use the best time provided by those set of NTP Servers. The **NTP Servers** table has the following settings:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranger Gateway</td>
<td>None (direct) - communicate with the NTP server directly, or communicate to the NTP server through the specified Ranger Gateway</td>
</tr>
<tr>
<td>NTP Server</td>
<td>NTP server to use for time synchronization</td>
</tr>
<tr>
<td>Key</td>
<td>Authentication key to use to validate time with NTP server. Values are retrieved from <strong>NTP Keys</strong> table.</td>
</tr>
</tbody>
</table>

The **NTP Keys** table lists the set of authentication keys used by ZoneRanger when it is communicating with an NTP server or when it is acting as an NTP server for its managed devices.

When **ZoneRanger Acts as NTP Server** is checked, this ZoneRanger will accept time requests from its managed devices. When ZoneRanger is in this NTP mode, it cannot be configured to proxy NTP requests through the Ranger Gateway on the **Configuration > Proxy** page **NTP** tab. When ZoneRanger is acting as an NTP server to serve time for its managed devices, authentication keys can be configured. When **Authenticate Client Requests** is selected, the keys defined in the **NTP Keys** section will be used for authentication.

**Note:** Saving changes to these settings results in a ZoneRanger restart.
Configuring ZoneRanger ports

The **Configuration > System** page **Ports** tab is used to enable and disable various ZoneRanger ports. Changing values determines how ports for the corresponding service respond.

Each ZoneRanger port has multiple service options depending on the type of port. Below are the service options:

<table>
<thead>
<tr>
<th>Service Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>eth0 and eth1</strong></td>
<td>Enabled for both interfaces.</td>
</tr>
<tr>
<td>disabled</td>
<td>Port is disabled.</td>
</tr>
<tr>
<td>Ranger Gateway</td>
<td>Port is disabled but service is still accessible through joined Ranger Gateways. This option is not available for all services.</td>
</tr>
<tr>
<td>Only</td>
<td></td>
</tr>
<tr>
<td><strong>eth0 only</strong></td>
<td>Port is only enabled on eth0.</td>
</tr>
<tr>
<td><strong>eth1 only</strong></td>
<td>Port is only enabled on eth1.</td>
</tr>
<tr>
<td>ZoneRanger Port</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>HTTP</strong></td>
<td>HTTP ports (80 and 8080) for the ZoneRanger web server. The Ranger Gateway Only option disables direct access to the web server, but permits proxy access through a joined Ranger Gateway.</td>
</tr>
<tr>
<td><strong>HTTPS</strong></td>
<td>HTTPS ports (443 and 8443) for the secure ZoneRanger web server. The Ranger Gateway Only option disables direct access to the secure web server, but permits proxy access through a joined Ranger Gateway.</td>
</tr>
<tr>
<td><strong>ICMP</strong></td>
<td>Enables and disables ICMP from the specified interface. The Ranger Gateway Only option disables direct access via ICMP, but permits proxy access through a joined Ranger Gateway.</td>
</tr>
<tr>
<td><strong>Messaging Port</strong></td>
<td>Port to use for communication between ZoneRanger and Ranger Gateway. If changed, the messaging ports of all joined Ranger Gateways and redundant ZoneRangers must be updated</td>
</tr>
<tr>
<td><strong>NTP</strong></td>
<td>NTP server port (123)</td>
</tr>
<tr>
<td><strong>RADIUS Proxy</strong></td>
<td>RADIUS ports (1812 and 1813) used to proxy RADIUS requests</td>
</tr>
<tr>
<td><strong>SNMP Agent</strong></td>
<td>SNMP agent port (161).</td>
</tr>
<tr>
<td><strong>SNMP Trap Port</strong></td>
<td>SNMP trap port (162) used to receive external SNMP traps.</td>
</tr>
<tr>
<td><strong>SSH</strong></td>
<td>SSH port (22) used to connect to the ZoneRanger configuration. The Ranger Gateway Only option disables direct access to SSH, but permits proxy access through a joined Ranger Gateway</td>
</tr>
<tr>
<td><strong>Syslog Port</strong></td>
<td>Syslog port (514) used to receive Syslog messages.</td>
</tr>
<tr>
<td><strong>TACACS+ Proxy</strong></td>
<td>TACACS+ port (49) used to proxy TACACS+ requests.</td>
</tr>
<tr>
<td><strong>Telnet</strong></td>
<td>Telnet port (23) used to connect to ZoneRanger configuration.</td>
</tr>
<tr>
<td><strong>TFTP Server</strong></td>
<td>TFTP server port (69).</td>
</tr>
</tbody>
</table>

**Note:** When Virtual IP is enabled, it will open a UDP port on the configured interface. This port is used for communication between redundant ZoneRangers. This UDP Port is configured on the Configuration > Peers page Virtual IP tab.

**Note:** If the ZoneRanger web interface is unavailable, you can run the portControl command from a joined Ranger Gateway to enable and disable the listed ports, except for the messaging port.

**Configuring ZoneRanger properties**

Properties are used to configure or display some aspect of ZoneRanger operation. The Configuration > System page Properties tab list the current set of ZoneRanger properties. In general, properties are used for advanced tuning and analysis.
The Properties section lists the set of ZoneRanger name/value pairs which define the properties for this ZoneRanger.

Traffic

ZoneRanger can receive and proxy many different types on network traffic. It is often difficult to determine how much traffic the ZoneRanger is receiving and proxying. The ZoneRanger automatically tracks how much data is received and proxied. The ZoneRanger can also be configured to monitor thresholds by Traffic Type and to send an SNMP trap if a threshold is exceeded. The Configuration > Traffic page is used to configure the thresholds by Traffic Type for received and proxied traffic.

Configuring Traffic Options

The Configuration > Traffic page Options tab is used to specify the level of logging for the Traffic service on the ZoneRanger, and can be used to specify the measurement interval for Traffic threshold calculations.

The Traffic Logging levels are:
<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Logging is off</td>
</tr>
<tr>
<td>Short</td>
<td>Traffic totals are logged each time the thresholds are checked.</td>
</tr>
<tr>
<td>Full</td>
<td>Traffic totals along with counts for each IP address are included.</td>
</tr>
</tbody>
</table>

This Traffic log can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/traffic.log`. The log file may also be viewed on the View > Service Logs page.

The Traffic measurement interval defines how frequently the ZoneRanger will check if any thresholds are exceeded. The traffic rate is evaluated within this interval. Traffic will be counted and logged (if logging is enabled) even if thresholds are not configured to be checked. The minimum traffic measurement interval is 60 seconds.

The traffic rate is calculated for each one second interval and the highest rate is saved and used to compare with the configured thresholds. As an example, if the SNMP threshold is configured for 100 requests/sec and the interval is 5 minutes, if a burst of 105 requests occurs during one second and even if no other requests are received within the 5 minute interval, the maximum one second traffic rate is 105 requests/sec which exceeds the threshold.

Received and proxied traffic is measured for both managed and unmanaged IP addresses. Also, traffic is counted even if it does not pass a configured forwarding rule.

**Configuring Received Traffic**

The Configuration > Traffic page Received Traffic tab is used to enable threshold monitoring by traffic type or IP address, set threshold values by traffic type and IP address, and whether or not to send an SNMP trap if a configured threshold is exceeded for traffic received by the ZoneRanger.
The **Enable Threshold Monitoring** checkboxes are used to configure whether or not the ZoneRanger should monitor thresholds by traffic type or by IP address. When enabled by traffic type, the total traffic received by traffic type will be compared with the configured threshold during the measurement interval. When enabled by IP address, the traffic received from each IP address will be compared with the configured threshold for each traffic type during the measurement interval.

If threshold monitoring is enabled and a threshold is exceeded, a ZoneRanger audit message will be displayed as well as a message will be logged in the ZoneRanger System log. If the **Send a trap when a threshold is exceeded** checkbox is checked, the ZoneRanger will also generate an SNMP trap containing information about the exceeded threshold.

### Configuring Proxied Traffic

The **Configuration > Traffic** page **Proxied Traffic** tab is used to enable threshold monitoring by traffic type or IP address, set threshold values by traffic type and IP address, and whether or not to send an SNMP trap if a configured threshold is exceeded for traffic proxied by the ZoneRanger.

---

**Figure 35-73: Configuration > Traffic page Received Traffic tab**

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Threshold (packets/second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>500</td>
</tr>
<tr>
<td>Generic</td>
<td>250</td>
</tr>
<tr>
<td>NetFlow</td>
<td>250</td>
</tr>
<tr>
<td>sFlow</td>
<td>250</td>
</tr>
<tr>
<td>Syslog</td>
<td>250</td>
</tr>
<tr>
<td>Traps</td>
<td>250</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Threshold (packets/second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>100</td>
</tr>
<tr>
<td>Generic</td>
<td>50</td>
</tr>
<tr>
<td>NetFlow</td>
<td>50</td>
</tr>
<tr>
<td>sFlow</td>
<td>50</td>
</tr>
<tr>
<td>Syslog</td>
<td>50</td>
</tr>
<tr>
<td>Traps</td>
<td>50</td>
</tr>
</tbody>
</table>

---

**ZoneRanger 5.6 User’s Guide**

189
The **Enable Threshold Monitoring** checkboxes are used to configure whether or not the ZoneRanger should monitor thresholds by traffic type or by IP address. When enabled by traffic type, the total traffic proxied by traffic type will be compared with the configured threshold during the measurement interval. When enabled by IP address, the traffic proxied to each IP address will be compared with the configured threshold for each traffic type during the measurement interval.

If threshold monitoring is enabled and a threshold is exceeded, a ZoneRanger audit message will be displayed as well as a message will be logged in the ZoneRanger System log. If the **Send a trap when a threshold is exceeded** checkbox is checked, the ZoneRanger will also generate an SNMP trap containing information about the exceeded threshold.

### Whitelist

ZoneRanger can receive data from many different network sources as well as send requests to many different network sources. For node licensed ZoneRangers, only information and requests from managed nodes will be processed. In the case of ZR-SPX, all information and requests are processed since there is no check for node management. For additional security and performance, a set of devices (“whitelist”) may be configured to restrict the set of IP addresses. The **Configuration > Whitelist** page is used to configure the specific set of devices from which ZoneRanger will receive information or to which ZoneRanger will send information.
The Enable button enables and activates the Whitelist feature. Once enabled, only information from those devices with a source address specified in the whitelist will be processed. All information from other devices will be ignored. This includes new Join requests from Ranger Gateways and new Redundancy requests from other ZoneRangers.

The Enforce Whitelist For Outbound Requests checkbox configures the ZoneRanger to apply the whitelist to all outbound ZoneRanger requests. This applies to Discovery, Root Cause, Diagnostics, Join, Redundancy requests as well as proxy requests received from joined Ranger Gateways. Network servers should as DNS and NTP must also be added to the whitelist in order to allow for those services to continue to operate.

You can add address patterns using the table view or the quick entry view. The quick entry view can be used to paste from a file.

**Diagnostics**

**Address Resolver**

ZoneRanger provides a diagnostic tool to resolve and reverse resolve addresses via the Diagnostics > Address Resolver page.

**Figure 35-76. Diagnostics > Address Resolver**
The **Diagnostics > Address Resolver** page is used to resolve hostnames to IP addresses, or to “reverse resolve” IP addresses to hostnames. The hostname does not have to be managed by the ZoneRanger.

**FindRoute**

ZoneRanger provides a diagnostic tool to display the route information from a source host to a destination host via the **Diagnostics > FindRoute** page.

![FindRoute screenshot](image)

**Figure 35-77. Diagnostics > FindRoute**

The FindRoute tool displays the route information from a source host to a destination host. Unlike the Traceroute tool, the FindRoute tool uses SNMP to determine the route between hosts. The SNMP settings used are those configured on the **Configuration > SNMP** page.

**Insertion Tools**

ZoneRanger provides a diagnostic tool to insert traps, UDP packets, and Syslog messages to test forwarding and Ranger Gateway configuration via the **Diagnostics > Insertion Tools** page.

![Insertion Tools screenshot](image)

**Figure 35-78. Diagnostics > Insertion Tools page**
The insertion diagnostic tools enable you to insert traps, UDP packets, and Syslog messages to test forwarding and Ranger Gateway configuration. The activity indicators flash only when traps, packets, or messages are forwarded to a Ranger Gateway.

When using **Insert SNMP Trap**, the SNMPv1 test trap `tscZRTTestTrap` with an enterprise of `1.3.6.1.4.1.26868.1.1.16` and type of `91` is generated. If the **Trap Forwarding** activity indicator flashes, this indicates that the inserted trap was forwarded to a Ranger Gateway.

When using **Insert UDP Packet**, a UDP datagram with the payload containing the string ZoneRanger UDP test, followed by a count of the number of test datagrams sent since you most recently logged into ZoneRanger is generated. If the **UDP Forwarding** activity indicator flashes, this indicates that an inserted UDP datagram was forwarded to a Ranger Gateway.

When using **Insert Syslog message**, a Syslog INFO message with a program name of ZoneRanger and the string Syslog test as the message text is generated. The test message is followed by a count of the number of test messages sent since you most recently logged into ZoneRanger. If the **Syslog Forwarding** activity indicator flashes, this indicates that an inserted Syslog message was forwarded to a Ranger Gateway.

**Ping/Scan**

ZoneRanger provides a diagnostic tool to ping IP addresses and scan TCP ports and SNMP interfaces via the **Diagnostics > Ping/Scan** page.

![Figure 35-79. Diagnostics > Ping/Scan page ICMP Ping tab](image)

The **ICMP Ping** diagnostic enables you to ping an address and view the results. The address does not have to be a ZoneRanger managed device. By default, the ICMP ping diagnostic reports the number of transmitted ICMP echo requests, the number of received echo replies, and the total elapsed time of the ping. If the ZoneRanger is IPv6 capable, IPv6 pings can be made using this utility.

**Using the TCP port scan diagnostic**

The **TCP Port Scan** diagnostic scans the specified address for open TCP ports and displays the results.
The TCP Port Scan diagnostic can either be used to scan the Well Known Ports (0 - 1023) or a set of individual ports. The valid Port values are 0 – 65535. The Address does not need to be a ZoneRanger managed device. After unchecking the Scan Well Known Ports checkbox, a single TCP port or a TCP port range may be specified.

By default, the TCP Port Scan diagnostic will ping the specified address before scanning TCP ports. This prevents the scanning of an unreachable device. However, some networks may filter unreachable messages which causes the scan to stop prematurely. By checking the Suppress Ping checkbox, the scan will not test for the availability of the device. If the device is unreachable, there could be a significant timeout for each tested TCP port.

Using the SNMP interface scan diagnostic

The SNMP interface scan diagnostic scans the interface table on the specified address and displays admin and oper status. If the ZoneRanger is IPv6 capable, IPv6 scans can be made using this utility.

If Use Configured Settings is checked, the configured SNMP rules are used. If Use Configured Settings is not checked, the SNMP rules for the scan must be specified. A specific interface index may be entered by unchecking Scan All Interfaces.

SNMP

The Diagnostic > SNMP page provides tools to perform SNMP GetNext requests to perform the function of the popular snmpwalk command, displaying a tree of information for the specified device. It also provides a tool to discover SNMPv3 Engine Ids. If the ZoneRanger is IPv6 capable, IPv6 requests can be made using this utility.
To use the SNMP settings configured in **Configuration > SNMP**, check the **Use Configured Settings** check box. Otherwise, uncheck the box to specify alternate SNMP information. For IPv6 requests, no default configuration settings are available.

### Using the SNMP Engine IDs diagnostic

The SNMP Engine IDs diagnostic discovers the SNMP v3 Engine ID for the specified node and determines whether or not this Engine ID has previously been discovered on another device.
The **Diagnostic > SNMP page Engine IDs tab** provides the ability to discover the SNMPv3 Engine ID, SNMP Engine Reboots and SNMP Engine Time of the specified node. If the SNMP Engine ID has been previously discovered, the specified node’s reported SNMP Engine Reboots and SNMP Engine Time are compared to the expected values. If the result is outside of the expected values, SNMP proxy requests for this device will be discarded.

ZoneRanger maintains a cache of SNMPv3 Engine IDs that it has previously discovered. It uses this cache to verify SNMPv3 agents. When using this diagnostic, any other IP addresses using the SNMP Engine ID of the specified device will also be reported. SNMP Engine IDs discovered using this diagnostic page will not be cached. SNMP Engine IDs are only cached when discovered via SNMP Proxy or SNMP notifications.

Note: IP addresses on the same device can use the same SNMP Engine ID. In addition, different ports on the same node may have different SNMP Engine IDs. When more than one device has the same SNMP Engine ID, SNMPv3 packets from the device with the lower SNMP Engine Reboots and SNMP Engine Time may be discarded.

**TACACS+/RADIUS**

The **Diagnostic > TACACS+/RADIUS Test page** performs an authorization request that can be used to test ZoneRanger TACACS+ and RADIUS proxy configuration.
The TACACS+/RADIUS diagnostic can be used to validate the TACACS+ and RADIUS proxy service configuration and to perform sample authentication transactions. The **Source Address** is used to find the configured proxy rule and associated server group. If the address is 127.0.0.1, the server group for ZoneRanger access control is used.

If **Use Shared Key from Server Group** is selected, the shared key defined in selected server group will be used for encryption. If not selected, an alternate **Shared Key** may be specified.

If **Perform authorization** is selected, if the protocol is TACACS+, an authorization request is performed if the authentication request was successful. If not selected, no authorization request is performed.

If **use the ZoneRanger’s configured values** is selected, the values for Service, Protocol, and Command already configured on the ZoneRanger will be used in the authorization request. If not selected, specified values for Service, Protocol, and Command will be used for the authorization request.

Cisco ACS servers, beginning with version 5, require a TACACS+ authorization request to include a Command argument if the Service argument is shell. If the Command box is checked and a Command argument value is specified, a Command argument with the specified value will be added to the authorization request. If the Command box is checked and no Command argument value is specified, an empty Command argument will be added to the authorization request.

**Traceroute**

The **Diagnostic > Traceroute** performs the function of the `traceroute` command, displaying the route between the ZoneRanger and a host.
The Traceroute diagnostic performs the function of the popular traceroute command. No results are displayed until the traceroute finishes. The hostname does not need to be a ZoneRanger managed device. If the **Do not map IP addresses to host names** checkbox is checked, the command will not attempt to resolve any of the returned IP addresses. This can improve performance of the command. If the ZoneRanger is IPv6 capable, IPv6 routes can be traced using this utility.

**View**

**Database**

During discovery, ZoneRanger builds a database containing information about discovered nodes, interfaces, networks, and TCP ports. The **View > Database** page enables you to display information about these entities.

The **View > Database Viewer** page has a tab for each of the following types of entities: nodes, interfaces, networks, and TCP ports. Each tab provides a means of querying and filtering queries for a specific entity type.
The **Displayed Columns** list is a multi-select list that limits which columns of the table are displayed. To select multiple columns, hold down the **Ctrl** key while clicking items in the list.

**Query Filter** is used to return only rows that meet specified criteria. To add criteria, click **Add**, select the **Column Name**, and enter a **Value**. If you add multiple criteria, each row returned from a query matches all criteria.

Values returned by a query may be links. Clicking on these links displays either the corresponding database query results or a **Node Reports** page.

The **Reset** button may be used to set all of the fields on the page back to their original values.

**Network Reports**

Network reports provide a set of useful reports about specific network configuration topics. The **View > Network Reports** page enables you to view network reports for resolved IP addresses, resolved nodes, and devices that support SNMP.

![ZoneRanger Network Reports](image)

**Figure 35-87. View > Network Reports page**

**Viewing the Resolved IP Addresses report**

The **View > Network Reports** page **Resolved IP Addresses** tab displays two lists, one of resolved IP addresses and one of unresolved IP addresses. The lists are built using IP addresses captured during discovery.

During discovery, ZoneRanger uses a variety of techniques to discover IP addresses. ZoneRanger then attempts to resolve each IP address to a hostname. Hostnames could be resolved for the IP addresses listed in the **Resolved IP Addresses** list, but not for the IP addresses listed in the **Unresolved IP Addresses** list.

When the **Resolved IP Addresses** report is initially displayed, the resulting report is generated by performing a database query, and indicates which IP addresses could be resolved and which could not as of the last time discovery was executed.
The Test button can be used to update the Resolved IP Addresses report based on the current DNS configuration. The update process can take a few minutes as ZoneRanger attempts to resolve hostnames for IP addresses in the database. When the test finishes, the Resolved IP Addresses and Unresolved IP Addresses lists are refreshed. Unmanaged nodes appear as plain text.

**Note:** The test does not update the database. As a result, when you next view the Resolved IP Addresses report, the results revert to displaying status as of the last time discovery was performed.

**Viewing the Resolved Nodes report**

The View > Network Reports page Resolved Nodes tab displays the list of resolved nodes and the list of unresolved nodes. The lists are built using IP addresses captured during discovery.

During discovery, ZoneRanger uses ICMP requests to discover nodes. ZoneRanger could resolve hostnames for the nodes listed in the Resolved Nodes list, but could not resolve hostnames for the IP addresses listed in the Unresolved Nodes list.

Because no hostname could be resolved for the IP addresses in the Unresolved Nodes list, only IP addresses appear in this list. In both lists, information for managed nodes appear as hyperlinks to node reports. Unmanaged nodes appear as plain text.

**Viewing the SNMP Accessible report**

The View > Network Reports page SNMP Accessible tab displays the list of SNMP accessible nodes and the list of SNMP inaccessible nodes. The lists are built using the discovered nodes.
During discovery, ZoneRanger uses the configured SNMP settings to query certain SNMP information from discovered nodes. ZoneRanger received responses to these requests in the nodes listed in the **SNMP Accessible Nodes** list, but did not receive a response for the nodes listed in **SNMP Inaccessible Nodes** list.

The two lists display an IP address next to each node, and the SNMP version currently configured for the SNMP accessible nodes. The IP address is the configured preferred interface for SNMP requests. Unmanaged nodes appear as plain text.

When the **SNMP Accessible** report is initially displayed, the resulting report is generated by performing a database query, and indicates which nodes were accessible using SNMP and which were not, as of the last time discovery was performed. To update the report based on current device status and configuration, click **Test**.

The update process can take a few minutes as ZoneRanger performs an SNMP Get request of `sysObjectID` for each node. When the test finishes, the **SNMP Accessible Nodes** and **SNMP Inaccessible Nodes** lists are refreshed.

**Note:** The test does not update the database. As a result, when you next view the **SNMP Accessible** report, the results revert to displaying status as of the last time discovery was performed.

**Node Reports**

The **View > Node Reports** page displays detailed information for the nodes that a ZoneRanger manages. The displayed node information enables you to view the status of individual interfaces and TCP ports on managed nodes, and to view the status of managed and unmanaged nodes in the path between a ZoneRanger and a managed node of interest.
The View > Node Reports page displays all ZoneRanger managed devices and by their current status. Within each tab, the devices of a particular type may be viewed by using the dropdown. Each report displays information based on the last time discovery has run. The wrench icon provides a link to the Diagnostics > Ping/Scan page for the indicated node or interface.

Status colors have the following meaning:

<table>
<thead>
<tr>
<th>Color</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>Normal</td>
</tr>
<tr>
<td>Yellow</td>
<td>Marginal</td>
</tr>
<tr>
<td>Red</td>
<td>Critical</td>
</tr>
<tr>
<td>Blue</td>
<td>Unknown (Not configured for polling)</td>
</tr>
<tr>
<td>Orange</td>
<td>Impacted</td>
</tr>
</tbody>
</table>

**Preferences**

The View > User Preferences page is used to configure preferences associated with the individual web users. Each web user that was configured on the Configuration > Access Control page has a corresponding set of user preferences which are applied whenever the user logs in.
The **View > Preferences** page allows the user to completely control their view of the ZoneRanger dashboard. By unchecking an item, that item will be removed from the dashboard. Each section may also be moved or removed.

**Root Causes**

The **View > Root Causes** page displays information about outstanding root causes. A root cause is the *entity* underlying one or more symptoms of a network problem. An entity may be a node, interface, TCP service, or cloud.

Root cause details appear to the right of the root cause list, displayed in two sections. The upper section displays information about the root cause entity. Displayed information varies, depending on entity type and the amount of available information about the entity.

The lower section displays information about impacted entities. Impacted entities are those affected by the root cause.
Service Logs

ZoneRanger logs information about the various ZoneRanger services in the form of log files. The View > Service Logs page allows the contents of the service logs to be displayed.

Many of the ZoneRanger services, such as ICMP Proxy, SNMP Proxy, TACACS+ Proxy, etc, may be configured to log information to service-specific log files.

After you specify any filtering criteria, click Show Matching Log Entries to display any log entries from the specified log file that match your filtering criteria. The From and To check boxes enable you to specify the time period for which you want to view log entries. If you uncheck the From check box, the start time is unbounded; in other words, the start of the period is the time of the oldest log entry.

ZoneRanger will store up to 7 days of log entries. Log files can be downloaded by the downloadFile command on a Ranger Gateway. The log file names are service-specific.

The Show Matching Log Entries and Automatically Update button may be used to view current log entries which match the specified criteria as well as to automatically update the display when a new log entry is received by the ZoneRanger which matches the criteria. To stop automatically updating the display with the specified criteria, click the Stop Updating button. The View > Service Logs page will remain in the automatic update mode until the Stop Updating button is clicked or the web browser is exited.

Statistics

ZoneRanger collects statistics for various services. The View > Statistics page displays statistics on a variety of ZoneRanger services.
When the ZoneRanger application software is started or restarted, its services cache statistics which are appropriate for each service. These statistics are useful in determining the level of processing a particular service is experiencing. Statistics may be updated by using the **Refresh Selected** button. Statistics may be reset (set to 0) by using the **Reset Selected** button.

**Syslog**

ZoneRanger logs all Syslog messages. The **View > Syslog** page displays logged messages that meet the filtering criteria.
ZoneRanger will store up to 7 days of received syslog messages. Syslog log files can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/syslog.log`.

The `Show Matching Syslog and Automatically Update` button may be used to view current syslog messages which match the specified criteria as well as to automatically update the display when a new syslog message is received by the ZoneRanger which matches the criteria. To stop automatically updating the display with the specified criteria, click the `Stop Updating` button. The `View > Syslog` page will remain in the automatic update mode until the `Stop Updating` button is clicked or the web browser is exited.

**System Audit**

Every five minutes, a ZoneRanger audits its overall status. The audit comprises tests that verify whether the ZoneRanger is operating correctly, and can communicate with any joined Ranger Gateways or redundant ZoneRangers. The `View > System Audit` page reports abnormal events detected during the previous audit.

There are three levels of audit results:

- **Warnings/Notes**
- **Major Errors**
Where possible, ZoneRanger attempts to correct the causes of major errors, using preconfigured escalating corrective action sequences. To review the log of corrective actions, see the system log (View > System Log).

**System Information**

The View > System Information page displays information about the ZoneRanger system, the status of joined Ranger Gateways, and the patch history.

![System Information](image)

**System Log**

The View > System Log page displays significant ZoneRanger events that have been logged in the ZoneRanger system log.
You can specify criteria used to filter the log entries for display, these may include regular expressions, max entries, and text search. After you specify any filtering criteria, click **Show Matching Log Entries** to display any system log entries that match your filtering criteria.

Some entries in the system log may be throttled to avoid filling the system log with repeated messages. When throttling occurs, multiple similar entries are combined into a single entry, with the number of entries that were skipped displayed in square brackets at the end of the entry.

The system log contains the following types of entries:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INFO</strong></td>
<td>Information entries generally report events that occur normally</td>
</tr>
<tr>
<td><strong>WARN</strong></td>
<td>Warning entries report events that might indicate problems</td>
</tr>
<tr>
<td><strong>ERROR</strong></td>
<td>Error entries report events during which some action needs to be take</td>
</tr>
</tbody>
</table>

The **From** and **To** check boxes enable you to specify the time period for which you want to view the system log. If you uncheck the **From** check box, the start time is unbounded; in other words, the start of the period is the time of the oldest log entry.
ZoneRanger will store up to 7 days of system log messages. System log files can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/ranger.log`.

The Show Matching Log Entries and Automatically Update button may be used to view current system messages which match the specified criteria as well as to automatically update the display when a new system message is received by the ZoneRanger which matches the criteria. To stop automatically updating the display with the specified criteria, click the Stop Updating button. The View > System Log page will remain in the automatic update mode until the Stop Updating button is clicked or the web browser is exited.

Traffic Information

ZoneRanger monitors the amount of received and proxied traffic it has processed in the last two traffic measurement intervals as well as the peak traffic by traffic type within the measurement interval. The View > Traffic Information page displays the amount of traffic based on IP address.
Received Traffic

The View > Traffic Information page Received Traffic tab shows the amount of total received traffic and received traffic per IP address over the last two measurement intervals as configured in the Configuration > Traffic page Options tab. Traffic amounts will only be displayed for those IP addresses that have data greater than 0. Up to 500 IP addresses will be displayed. Note that if there are no Forwarding Rules configured NetFlow, sFlow, or Generic, then no data will be received by ZoneRanger since ZoneRanger only listens for those protocols if there is a Forwarding Rule configured.

The Reset Traffic Data button will reset all traffic counts, both received and proxied. The Update Traffic button will update the currently displayed traffic information. If the Update Traffic button is clicked within the current measurement interval, there will be no changes to the display.

The Automatically Update Traffic Data button will cause the display to update when the current measurement interval has changed. To stop automatically updating the display, click the Stop Updating button. The View > Traffic Information page Received Traffic tab will remain in the automatic update mode until the Stop Updating button is clicked or the web browser is exited.

Proxied Traffic

The View > Traffic Information page Proxied Traffic tab shows the amount of total proxied traffic and proxied traffic per IP address over the last two measurement intervals as configured in the Configuration > Traffic page Options tab. Traffic amounts will only be displayed for those IP addresses that have data greater than 0. Up to 500 IP addresses will be displayed.

The Reset Traffic Data button will reset all traffic counts, both forwarded and proxied. The Update Traffic Data button will update the currently displayed traffic information. If the Update Traffic Data button is clicked within the current measurement interval, there will be no changes to the display.
The **Automatically Update Traffic Data** button will cause the display to update when the current measurement interval has changed. To stop automatically updating the display, click the **Stop Updating** button. The **View > Traffic Information** page **Forward Traffic** tab will remain in the automatic update mode until the **Stop Updating** button is clicked or the web browser is exited.

**Peak Rates**

The **View > Traffic Information** page **Peak Rates** tab shows the most recent peak traffic analysis for each traffic type.

![ZoneRanger Traffic Information](image)

*Figure 35-103. View > Traffic Information page Peak Rates tab*

Peak traffic rate analysis calculates the average traffic rate during the busiest time interval within the most recent measurement interval (typically one hour). For example, the one second peak traffic rate is the average traffic (in transactions per second) during the busiest one second interval within the most recent periodic measurement interval. The two second peak traffic rate is the average traffic during the busiest contiguous two second interval. This is reported for up to a 60 second interval.

Peak traffic analysis can be used to measure the magnitude and duration of traffic bursts. A high one second rate accompanied by decreasing rates for the longer intervals indicates a transient burst of traffic. If the rates for the longer intervals are also high, this indicates a more sustained traffic burst.

The **Update Traffic Data** Button will update the currently displayed traffic information. If the **Update Traffic Data** button is clicked within the current measurement interval, there will be no changes to the display.

The **Automatically Update Traffic Data** button will cause the display to update when the current measurement interval has changed. To stop automatically updating the display, click the **Stop Updating** button. The **View > Traffic Information** page **Forward Traffic** tab will remain in the automatic update mode until the **Stop Updating** button is clicked or the web browser is exited.

**Traps**

ZoneRanger logs all received traps. The **View > Traps** page displays logged traps that meet the filtering criteria.
After you specify any filtering criteria, click **Show Matching Traps** to display any traps that match your filtering criteria. The **From** and **To** check boxes enable you to specify the time period for which you want to view traps. If you uncheck the **From** check box, the start time is unbounded; in other words, the start of the period is the time of the oldest log entry.

ZoneRanger will store up to 7 days of traps. Trap log files can be downloaded by the `downloadFile` command on a Ranger Gateway. The log file is called `/log/trapd.log`.

The **Show Matching Traps and Automatically Update** button may be used to view current traps which match the specified criteria as well as to automatically update the display when a new trap is received by the ZoneRanger which matches the criteria. To stop automatically updating the display with the specified criteria, click the **Stop Updating** button. The **View > Traps** page will remain in the automatic update mode until the **Stop Updating** button is clicked or the web browser is exited.
User's Guide

The ZoneRanger User's Guide will be displayed in a separate window or tab.
Chapter 36: Ranger Gateway Viewer

The Ranger Gateway Viewer provides a GUI for convenient configuration and use of Ranger Gateway services.

Starting the Ranger Gateway Viewer

To start the Ranger Gateway Viewer on Linux and Solaris systems, run the RangerGateway command on a command line.

On Windows systems, use the Ranger Gateway Viewer menu item in the Windows Start menu. Select All Programs > Tavve > Ranger Gateway Viewer.

A splash screen will be displayed briefly while the Ranger Gateway Viewer is starting up, then the main Ranger Gateway Viewer window will be displayed, as shown in the following figure.

![Figure 36-1. Ranger Gateway Viewer](image)

The main window consists of the following parts:

- Menu bar, located near the top of the window.
- Left-hand pane, consisting of a toolbar, and a list box showing all joined ZoneRangers.
- Right-hand pane, which contains Status and Information tabs associated with whatever joined ZoneRanger is selected in the list box in the left-hand pane.
Using the Ranger Gateway Viewer toolbar

The Ranger Gateway Viewer toolbar includes the following controls:

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Ranger Gateway, based on the most recent audit: Green - normal; Yellow - warning; Red - error</td>
</tr>
<tr>
<td>Join</td>
<td>Initiate a join request to a ZoneRanger</td>
</tr>
<tr>
<td>Unjoin</td>
<td>Initiate an unjoin request to a ZoneRanger</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refresh the list of joined ZoneRangers, and the information for the selected ZoneRanger</td>
</tr>
</tbody>
</table>

Displayed information on the Status tab

All ZoneRangers that are currently joined to the Ranger Gateway are listed on the Ranger Gateway Viewer. When a ZoneRanger in this list is selected, the Status and Information tabs associated with that ZoneRanger are displayed. This information is updated based on the refresh interval specified on the Configure > Gateway Viewer Settings... window.

The Status tab provides the following:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZoneRanger Status</td>
<td>ZoneRanger status from the most recent audit. Mouse-over displays a brief summary of audit results. Clicking displays a complete summary of audit results.</td>
</tr>
<tr>
<td>Root Cause Status</td>
<td>Current ZoneRanger root cause status. Mouse-over displays a brief summary of root cause information. Clicking displays a list of current root cause devices/entities.</td>
</tr>
<tr>
<td>Inventory Bars</td>
<td>Current ZoneRanger inventory organized by type. Mouse-over displays number of devices in each status.</td>
</tr>
<tr>
<td>Browse (HTTP)</td>
<td>Opens a browser window for the selected joined ZoneRanger</td>
</tr>
<tr>
<td>Browse (HTTPS)</td>
<td>Opens a secure browser window for the selected joined ZoneRanger</td>
</tr>
</tbody>
</table>

If the Browse (HTTP) or Browse (HTTPS) buttons are clicked, the Ranger Gateway will launch a web browser on the selected ZoneRanger. On Windows systems, the Ranger Gateway will launch the default browser for that system. On Linux and Solaris systems, the Ranger Gateway will search for a web browser, based on the configured browser path\(^\text{11}\), and will launch the selected browser. Note that the Browse (HTTP) and Browse (HTTPS) may be disabled if the corresponding port is disabled on the selected ZoneRanger.

---

\(^{11}\)See Configure > Gateway Viewer Settings...
Displayed information on the Information tab

Figure 36-2. Ranger Gateway Viewer Information tab

The **Information** tab displays system and configuration information about the selected ZoneRanger. This information is updated based on the refresh interval specified on the **Configure > Gateway Viewer Settings...** window. The message “Loading..” will appear when the Ranger Gateway is requesting information from the selected ZoneRanger. This message normally appears for only a few seconds. If it appears for a significant amount of time, then the Ranger Gateway is no longer able to communicate with the selected ZoneRanger.

### The Ranger Gateway Viewer menu

#### View

**Gateway Status...**

The **View > Gateway Status...** displays a window with the current status of the Ranger Gateway software.
Similar to the ZoneRanger, the Ranger Gateway software performs a self-audit every five minutes. If a condition is found which cannot be automatically resolved, information will be available in the **Gateway Status** window indicating the nature of the issue.

One such condition is the loss of communication between the ZoneRanger and the Ranger Gateway. In this case, the audit would indicate the last time the Ranger Gateway was able to communicate with the ZoneRanger.

Given that the Ranger Gateway audit runs on a five minute interval, it can take up to five minutes from the point where a problem condition occurs before the audit process on the Ranger Gateway detects the problem and generates an audit result. Given that the Ranger Gateway Viewer refreshes its information on a configurable time interval (the default is 30 seconds), there can also be a delay between the point where an audit result is generated on the Ranger Gateway and when it is reflected on the Ranger Gateway Viewer. The same delays can occur when a problem is resolved. It can take up to five minutes from the point when a problem is resolved for the Ranger Gateway to remove the corresponding audit result, and it can take up to a full Ranger Gateway Viewer refresh cycle for the removal of that result to be reflected on the Ranger Gateway Viewer.

**Gateway Log...**

The Ranger Gateway software maintains a log file of information regarding its status and processing. The **Gateway Log...** menu item will display this log using an operating system specific application. On Windows systems, the Ranger Gateway log is displayed using NotePad. On Linux and Solaris systems, the Ranger Gateway will search for a web browser, based on the configured browser path, and will display the log in the selected web browser. The Ranger Gateway Log file is located under the Ranger Gateway installation directory `log/gateway.log`.

**Configure Gateway Settings**

When **Gateway Settings...** is selected from the Configure menu, the **Gateway Settings** dialog will be displayed. This dialog contains the following parts:

- A list of settings categories, displayed on the left hand side of the dialog.
- Settings pane, displayed on the right hand side of the dialog. The content of the settings pane is based on the selected category.
- An **OK** button; clicking this button will save any settings changes and close the **Gateway Settings** dialog.

---

12See **Configure > Gateway Viewer Settings...**
• A **Cancel** button; clicking this button will close the Gateway Settings dialog without saving any changes. Clicking the **close** box in the upper right hand corner of the dialog has the same effect as clicking the **Cancel** button.

The settings pane content for each of the listed categories is described in the following sections.

**Gateway Settings…General**

The **Gateway Settings…General** window provides basic configuration information for the Ranger Gateway.

![Gateway Settings Window](image)

*Figure 36-4. Gateway Settings Window*

The configuration settings are the following:
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passcode</td>
<td>Default passcode used when joining to ZoneRangers if none is specified during the join request, or if the join was requested from the ZoneRanger web interface.</td>
</tr>
<tr>
<td>Mail Server Address</td>
<td>Hostname or IP address of the mail server the Ranger Gateway should use (e.g. for e-mail notifications related to root cause)</td>
</tr>
<tr>
<td>Messaging Port</td>
<td>Destination port used for communications between ZoneRangers and Ranger Gateways. All joined ZoneRangers and Ranger Gateways must use the same port.</td>
</tr>
<tr>
<td>HTTP Port Enabled</td>
<td>When checked, the Ranger Gateway will listen for HTTP requests on the configured <strong>HTTP Port</strong>, and will return a web page listing all of the currently joined ZoneRangers, along with links to their web interfaces via proxy through the Ranger Gateway.</td>
</tr>
<tr>
<td>HTTP Port</td>
<td>Ranger Gateway HTTP port which can be used to access the web interface of joined ZoneRangers.</td>
</tr>
</tbody>
</table>

**Gateway Settings…Access Control**

The **Gateway Settings…Access Control** window provides the configuration information for the Ranger Gateway handling of TACACS+ and RADIUS requests.
The Access Control window can be used to configure whether or not TACACS+ and RADIUS client requests from ZoneRanger managed devices, will be presented to the TACACS+/RADIUS server with the source address of the Ranger Gateway or the source address of the ZoneRanger managed device.

When the Spoof TACACS+ Client Requests checkbox is enabled, the source address in TACACS+ requests sent from the Ranger Gateway to the TACACS+ server will be the source address of the original sending device managed by the ZoneRanger. If the checkbox is disabled, the source address in these requests will be the address of the Ranger Gateway. The Spoof RADIUS Client Requests checkbox governs the source address in RADIUS requests in a similar manner.

Gateway Settings...Device Groups

The Gateway Settings...Device Groups window provides the configuration information for the creation and management of Device Groups. A device group is a named set of IP addresses, or address patterns, used in the configuration of the Proxy Access Control and Proxy Map services. Device groups are described in detail in Chapter 7.
Device Groups enable configuration settings to be associated with an arbitrary list of devices with disjoint IP addresses, as opposed to address ranges and wild cards which can only refer to contiguous IP address spaces.

The **Add** button displays a new window which is used to create a new Device Group. Once the Device Group has been created, additional addresses may be added under the **Address** column. The **Organize** button can be used to reorder the list of addresses in numerical ascending order.

To modify the name of a Device Group, right-click on the group name in the **Device Group** list, and click **Modify**. To remove a Device Group, right-click on the group name in the **Device Group** list, and click **Delete**.

Due to the frequency that addresses will be queried by other system services, the Ranger Gateway maintains a cache of Device Group information to increase system performance at the cost of memory usage. The **Device Groups Cache Size** is used to set the maximum number of entries in the cache. The default is 100 entries with a valid range of 0 – 10000.

**Gateway Settings...Forwarding**

The **Gateway Settings...Forwarding** window provides the configuration information for the Ranger Gateway handling of forwarding requests.
The Forwarding window allows users to configure whether or not forwarded SNMP Traps, Syslog messages, and other UDP traffic, will be sent from the Ranger Gateway to the receiving application with the source address of the Ranger Gateway or the source address of the ZoneRanger managed device.

When the **Spoof Source Address** checkbox is enabled, the source address in the UDP traffic will be the source address of the original sending device managed by the ZoneRanger. When the **Spoof Source Address** checkbox is disabled, the source address will be the address of the Ranger Gateway.

Note: The mechanism that the Ranger Gateway uses to spoof source addresses may be prevented by Windows XP security updates, so the Spoof Source Address option may need to be disabled for Ranger Gateways running on Windows XP.

**Gateway Settings…GVI**

The **Gateway Settings…GVI** window provides the configuration information for the Ranger Gateway Virtual Interface (GVI). This is the mechanism whereby the Ranger Gateway is able to intercept management application requests directed toward ZoneRanger managed devices. The GVI service is described in detail in Chapter 8.
The GVI Enabled checkbox can be used to configure whether or not the GVI service is enabled. When the gateway settings are saved (i.e. by clicking the **OK** button in the **Gateway Settings** dialog), if the GVI Enabled checkbox was enabled, the Ranger Gateway will create a virtual point-to-point interface on the management application server, and will receive all traffic that is routed to this virtual interface.

The **GVI Routes** list specifies the set of addresses that should be routed to the virtual interface. An address may be specified as a entire subnet (e.g. 10.0.0.0/255.255.255.0) or a specific address (e.g. 10.2.5.6). To add a new address, select the empty entry at the end of list and enter the address information. To delete entries, select the appropriate entries and click the **Delete** button.

**Gateway Settings...ICMP Proxy**

The **Gateway Settings...ICMP Proxy** window provides the configuration information for the Ranger Gateway ICMP Proxy service.
Management applications can use the ICMP proxy service to send ICMP requests through the Ranger Gateway to ZoneRanger managed devices. The **Timeout** value is the number of seconds to wait for a response from the ZoneRanger for each ICMP request. When **Use TTL** is checked, the ICMP proxy service will reuse the Time To Live value from the received packet. This is useful for supporting network utilities such as ICMP based **traceroute**. If this option is disabled, the ZoneRanger will use its own TTL for sending proxied ICMP requests.

**Gateway Settings...Inbound TCP Proxy**

The **Gateway Settings...Inbound TCP Proxy** window provides the mechanism to manage the inbound TCP proxy facilities on the Ranger Gateway.
The Inbound TCP Proxy window can be used to configure whether or not TCP proxy requests from the ZoneRanger managed devices, will be presented to the source address of the Ranger Gateway or the source address of the ZoneRanger managed device.

When the Inbound TCP Proxy Spoof Enabled checkbox is enabled, the source address in TCP proxy requests from ZoneRanger sent from the Ranger Gateway to an application will be the source address of the original sending device managed by the ZoneRanger. If the checkbox is disabled, the source address in these requests will be the address of the Ranger Gateway.

Gateway Settings...Logging

The Gateway Settings...Logging window provides the mechanism to manage the logging facilities on the Ranger Gateway.
The Ranger Gateway software has an extensive set of logging capabilities. These logs are useful in understanding how the Ranger Gateway is operating and in diagnosing problems. The following are the list of logs:

<table>
<thead>
<tr>
<th>Service/Protocol</th>
<th>Log file name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic UDP Forwarding</td>
<td>genericUDP.log</td>
</tr>
<tr>
<td>GVI</td>
<td>gvi.log</td>
</tr>
<tr>
<td>ICMP Proxy</td>
<td>icmpProxy.log</td>
</tr>
<tr>
<td>NetFlow Forwarding</td>
<td>netflow.log</td>
</tr>
<tr>
<td>NTP Proxy</td>
<td>ntpProxy.log</td>
</tr>
<tr>
<td>Port Map</td>
<td>portMap.log</td>
</tr>
<tr>
<td>Proxy Map</td>
<td>proxyMap.log</td>
</tr>
<tr>
<td>RADIUS Proxy</td>
<td>radiusProxy.log</td>
</tr>
<tr>
<td>RGVI</td>
<td>rgvi.log</td>
</tr>
<tr>
<td>sFlow Forwarding</td>
<td>sFlow.log</td>
</tr>
<tr>
<td>SNMP Proxy</td>
<td>snmpProxy.log</td>
</tr>
<tr>
<td>Syslog Forwarding</td>
<td>syslog.log</td>
</tr>
<tr>
<td>TACACS+ Proxy</td>
<td>tacacsProxy.log</td>
</tr>
<tr>
<td>TCP Inbound Proxy</td>
<td>inboundTcpProxy.log</td>
</tr>
<tr>
<td>TCP Proxy</td>
<td>tcpProxy.log</td>
</tr>
<tr>
<td>TFTP Proxy</td>
<td>tftpProxy.log</td>
</tr>
<tr>
<td>Traffic</td>
<td>traffic.log</td>
</tr>
<tr>
<td>Trap Forwarding</td>
<td>trap.log</td>
</tr>
</tbody>
</table>

Figure 36-11. Gateway Settings .. Logging Window
### Log Level

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No messages are logged</td>
</tr>
<tr>
<td>Short</td>
<td>Only message headers are logged</td>
</tr>
<tr>
<td>Long</td>
<td>Entire messages are logged</td>
</tr>
</tbody>
</table>

All log files are stored in the log directory under the base Ranger Gateway install directory.

**Gateway Settings…NTP Proxy**

The **Gateway Settings…NTP Proxy** window provides the configuration information for the Ranger Gateway handling of NTP requests.

The **NTP Proxy** window can be used to configure whether or not NTP client requests from ZoneRanger managed devices, will be presented to the NTP server with the source address of the Ranger Gateway or the source address of the ZoneRanger managed device.

When the **Spoof NTP Client Requests** checkbox is enabled, the source address in NTP requests sent from the Ranger Gateway to the NTP server will be the source address of the original sending device managed by the ZoneRanger. If the checkbox is disabled, the source address in these requests will be the address of the Ranger Gateway.

**Gateway Settings…Proxy Map**

The **Gateway Settings…Proxy Map** window provides the mechanism to configure the Proxy Map service on the Ranger Gateway. The Proxy Map service is described in detail in Chapter 17.
The Ranger Gateway provides the Proxy Map service to assist in processing proxy requests. When multiple ZoneRangers are joined to a Ranger Gateway, or when the use of address translation (NAT) is in effect, the Proxy Map service determines which ZoneRanger should service the incoming request.

When **Resolve Host Names** is checked, the Proxy Map service will resolve hostnames in the proxy requests to IP addresses prior to searching for matching configuration rules. If disabled, the Proxy Map service will search for configuration rules which match the specified hostname.

When **Allow Unconfigured Routes** is checked, in the absence of a configured route for a given destination address, the Proxy Map service will simply select any joined ZoneRanger to proxy the request. If this option is disabled, the Proxy Map service will only select ZoneRangers to handle proxy requests based on configured rules. If no matching rule is found for a given proxy request, that request will fail.

When **Balance ZoneRanger Selection** is checked, the Proxy Map service will attempt to balance proxy requests which can be serviced by multiple ZoneRangers across those ZoneRangers over time. If disabled, proxy requests will be sent to the ZoneRanger that most recently responded to this Ranger Gateway (i.e. the ZoneRanger from which the Ranger Gateway has most recently observed evidence of healthy activity).

**Route List Cache Size** is the number of Proxy Map routing rules cached by the Proxy Map service. The default is 1000 entries with a valid range of 0 – 10000.
The **Proxy Map** window’s **Route** tab allows for the creation of proxy routing rules that map the destination IP address of the incoming request to the Ranger Gateway (i.e. the **RG Address**), to a ZoneRanger that can proxy the request to the target device (i.e. the **ZoneRanger**), and, in cases where address translation is required, the IP address that the ZoneRanger should use to send traffic to the target device (i.e. the **ZR Address**).

To add a new route, click the entry fields at the bottom of the list and fill in the appropriate information. The **RG Address** field may be a Device Group or an IP address pattern. The **ZR Address** field may be an IP address pattern. To delete a route, select the route and click the **Delete** button.

The **Weight** field indicates the relative cost of each proxy map route. If there are more than one proxy map routes which match an incoming request, the lowest cost proxy map entry will be chosen if that ZoneRanger is responsive. The default weight, if not specified, is zero, which is the least cost.

To reorder the set of routes, select a route and use the **Up** and **Down** buttons to move the particular rules. The **Organize** button will order the rules with more specific routes to less specific routes.

**Gateway Settings…Proxy Ports**

The **Gateway Settings…Proxy Ports** window provides the mechanism to configure the Proxy Access Control service on the Ranger Gateway. The Proxy Access Control Service is described in detail in Chapter 15.
The **Proxy Ports** window **Port Config** tab allows configuration of Port Config rulesets (a.k.a. port configurations), which are named sets of rules specifying which transport protocol/port combinations are allowed, what management protocol will be used for a given transport/port, and any port translations that should be applied. Once defined, Port Config rulesets can be referenced in the Port Map rules, as configured on the **Port Map** tab.

To add a new Port Config ruleset, click the **Add** button. The **Transport** field indicates whether the incoming request is using **ICMP**, **TCP**, or **UDP**. The **RG Port** field is the destination port associated with the incoming request as received by the Ranger Gateway. The RG Port field may be a single port number (eg 22) or a port range (eg 300-310). The **Protocol** field is the management protocol to be used for the incoming request. The **ZR Port** field is the destination port that the ZoneRanger should use when forwarding the request to the target device, or a translation rule that can be used to calculate the port that should be used based on the rg-port. When the **Transport** field is **ICMP**, all of the other fields are ignored.

To modify the name of a Port Config ruleset, right-click on the name in the **Port Config** list, and click **Modify**. To remove a **Port Config** ruleset, right-click on the name in the **Port Config** list, and click **Delete**.

To add a rule to a Port Config ruleset, click the empty field at the bottom of the **Transport** column. To modify any of the fields in a rule, click on the individual field in the rule. To remove a rule from a Port Config ruleset, select the rule and click the **Delete** button.

The **Up** and **Down** button can be used organize the Port Config rules. The **Organize** button will reorder the rules so that they are grouped by transport protocol, with TCP rules listed first, followed by UDP rules, followed by ICMP.

**Port Config Cache Size** is the maximum number of table entries cached by the Range Gateway in memory to improve performance. The default is 100 entries with a valid range of 0 – 10000.
The **Proxy Ports** window **Port Map** tab allows configuration of Port Map rules. The Port Map rules determine which Port Config ruleset should be used based on the Source IP address associated with the requesting client, and Destination IP address of the target managed device.

To add a Port Map rule, click the empty field at the bottom of the **Source** column. The **Source** field may be a Device Group, IP address pattern, or special Device Group @Local. The Device Group @Local consists of all addresses that are local to the Ranger Gateway server. The **Destination** field may be a Device Group, IP address pattern, or special Device Group @ZoneRanger. The Device Group @ZoneRanger consists of the IP addresses of all joined ZoneRangers. The **Port Config** field is the name of the Port Config rule in the **Port Config** tab.

To change a Port Map rule, select any of the fields in the rule and modify it. To delete a Port Map rule, select any field in the rule and click the **Delete** button.

The order of Port Map rules is important since the Port Config ruleset to be used for an incoming request will be the first matching rule, according to the order the rules appear in the table. Use the **Up** and **Down** buttons to reorder the Port Map rules. The default Port Map rule should be last.

**Port Map Cache Size** is the maximum number of table entries cached by the Range Gateway in memory to improve performance. The default is 100 entries with a valid range of 0 – 10000.

**Gateway Settings...RGVI**

The **Gateway Settings...RGVI** window provides the mechanism to configure the Remote GVI settings on the Ranger Gateway. RGVI is the mechanism whereby the Ranger Gateway is able to intercept management application requests directed toward ZoneRanger managed devices, in cases where the management application and the Ranger Gateway have been installed on separate servers. The RGVI service is described in detail in Chapter 8.
The **RGVI Enabled** checkbox can be used to configure whether or not the RGVI service is enabled. When the gateway settings are saved (i.e. by clicking the **OK** button in the **Gateway Settings** dialog), if the **RGVI Enabled** checkbox was enabled, the Ranger Gateway will create a virtual point-to-point interface on the Ranger Gateway server, and will begin listening for connection requests from RGVI clients installed on management application servers.

RGVI clients communicate with the Ranger Gateway via UDP on a configured port (default: 1194). The **RGVI Port** text box can be used to specify a different UDP port. Note that if a value other than the default is used, all RGVI clients that are configured to connect to this Ranger Gateway will need to have their configurations modified to use the new port. RGVI client configuration is described in Appendix J.

The **RGVI Clients** list is used to configure the set of RGVI clients that are allowed to connect to this Ranger Gateway. The RGVI service on the Ranger Gateway will only accept client connections from addresses that are configured in this list.

The **Add** button displays a new **Add RGVI Client** dialog window which is used to configure a new entry in the **RGVI Clients** list. Each entry consists of two parts:

1. An IP address, address pattern, or device group, used to identify the client or clients for which this entry will be applied.

2. A list of subnet and/or host addresses to be intercepted by the corresponding RGVI client(s), or an indication that the list of addresses should be identical to that configured for the GVI service.

---

*Figure 36-17. Gateway Settings .. RGVI Window*
An initial set of subnet/host addresses to be intercepted can be configured in the Add RGVI Client window. Once the RGVI client entry has been created, you can select an entry in the RGVI Clients list and use the Use GVI Routes check box and Subnet/Host table to modify the intercept settings associated with the selected client. To add a Subnet/Host to be intercepted, click the empty field at the bottom of the Subnet/Host table, and enter the subnet or host address to be intercepted. To modify a host or subnet address, select the corresponding entry in the table and edits its value. To delete a host or subnet address, select the corresponding entry and click the Delete button.

To modify the address, address pattern, or device group used to identify an RGVI Clients table entry, right-click on the entry in the RGVI Clients list, and click Modify. This will open the Modify RGVI Client dialog window, which also can be used to modify any settings associated with the entry. To remove an RGVI Clients table entry, right-click on the entry and click Delete.

The Up and Down buttons, located below the RGVI Clients list, are used to change the order of the entries in the RGVI Clients list. When an RGVI client attempts to connect to the Ranger Gateway, the set of host and subnet addresses to be pushed to the client will be based on the first matching entry in the RGVI Clients list. If specific host addresses are always used to identify RGVI clients, there will only be one matching entry for any given client, so the order of the entries does not matter. However, if address patterns or device groups are used to identify RGVI clients, it is possible that multiple entries may match a given client, so the order of the entries in the RGVI Clients list becomes important.

Important Note: The set or host/subnet addresses to be intercepted by an RGVI client is pushed to the RGVI client at the point where the client connects with the Ranger Gateway, and cannot be modified after the connection is established. As a result, whenever the set of host/subnet addresses to be intercepted by a client is modified on the Ranger Gateway, it will be necessary to restart any affected clients.

Gateway Settings…SNMP Proxy

The Gateway Settings…SNMP Proxy window provides the mechanism to configure the SNMP Proxy settings on the Ranger Gateway.
If the IP Address Aliasing or Community String Conventions mechanisms for SNMP proxy are being used (see Chapter 28), the Ranger Gateway will need to be configured to listen for SNMP proxy requests on a configured port. In order to configure this port, and any associated settings, the **Community String SNMP Proxy Enabled** checkbox must be checked. The following settings can be configured when this checkbox is enabled:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Port on which the Ranger Gateway listens for SNMP proxy requests. The default is 4852.</td>
</tr>
<tr>
<td>Community String Format</td>
<td>Format that applications use to specify the SNMP community string when making SNMP proxy requests.</td>
</tr>
<tr>
<td>Client Address</td>
<td>The set of client addresses from which SNMP proxy requests will be accepted. May be specified as a Device Group, IP address, or IP address pattern</td>
</tr>
</tbody>
</table>

The **SNMP Timeout** field is the amount of time, in seconds, the Ranger Gateway waits for a response to an SNMP proxy request.

**Gateway Settings…SOCKS Server**

The Gateway Settings…SOCKS Server window provides the mechanism to configure the SOCKS Server on the Ranger Gateway. SOCKS is described in further detail in Appendix C.
SOCKS is a standard networking proxy protocol that enables SOCKS-aware applications to communicate TCP and UDP protocols through a SOCKS server without requiring direct IP-reachability to the end device. The Ranger Gateway provides a built-in SOCKS server which can be used to access ZoneRanger proxy services for TCP and UDP-based management protocols such as Telnet, SSH, HTTP, HTTPS, and SNMP. A SOCKS-aware application can direct proxy requests to the SOCKS server in the Ranger Gateway, which will relay these requests to managed devices via the ZoneRanger.

When **SOCKS Server Enabled** is checked, the **Server Port** setting can be defined. The **Server Port** field specifies the port on which Ranger Gateway will listen for SOCKS requests. The default is 4855.

**Gateway Settings…SSH Proxy**

The **Gateway Settings…SSH Proxy** window provides the mechanism to configure the SSH Proxy settings on the Ranger Gateway.
If the IP Address Aliasing mechanism is being used for SSH proxy, (see Chapter 28), the Ranger Gateway must be configured to listen for SSH proxy requests on a configured port.

When **SSH Proxy Enabled** is checked, the **SSH Proxy Port** and **SSH Proxy Destination Port** settings can be defined. The **SSH Proxy Port** field specifies the port on which Ranger Gateway will listen for SSH Proxy requests. The default is 4822. The **SSH Proxy Destination Port** field specifies the destination port which the ZoneRanger should use when sending SSH proxy traffic to managed devices. The default is 22.

**Gateway Settings...Status Traps**

The **Gateway Settings...Status Traps** window can be used to configure the Ranger Gateway to send status traps to a specified destination address and port.
Gateway status traps are generated by the internal audit process in the Ranger Gateway when problems are detected. When Send Gateway Status Traps is checked, the Destination Host Address and Destination Port fields can be defined. The Destination Host Address field specifies the hostname or IP address to which status traps should be sent. The Destination Port field specifies the destination port that should be used when sending status traps.

**Gateway Settings…TFTP Proxy**

The Gateway Settings…TFTP Proxy window provides the mechanism to configure TFTP Proxy destinations on the Ranger Gateway. The TFTP proxy service is described in detail in Chapter 32.
The Ranger Gateway has the ability to use TFTP proxy to transfer files to and from ZoneRanger managed devices.

The **Read Directory** field specifies the directory where TFTP files should be read when proxying files to ZoneRanger managed devices. The **Write Directory** field specifies the directory where TFTP files should be written when proxying files from ZoneRanger managed devices.

**Gateway Settings...Traffic**

The **Gateway Settings...Traffic** window provides the mechanism to configure thresholds and notifications for traffic proxied through and received by the Ranger Gateway.
The Ranger Gateway has the ability to monitor the amount of traffic received from and proxied to its joined ZoneRangers. The Traffic measurement interval defines how frequently the Ranger Gateway will check if a threshold has been exceeded. The traffic rate is calculated as an average over the measurement interval in seconds.

The Ranger Gateway monitors traffic in two categories. The Overall category is all of the traffic of a particular type either received from or proxied to all joined ZoneRangers. The Per ZoneRanger category is all of the traffic of a particular type either received from or proxied to an individual ZoneRanger.

In order for the Ranger Gateway to log traffic, Traffic Logging must be set in the Gateway Settings..Logging Window. If Traffic logging is set to Short, then Overall traffic will be logged. If Traffic logging is set to Full, then Overall and Per ZoneRanger traffic will be logged.

Figure 36-23. Gateway Settings..Traffic window Options
The Gateway Settings..Traffic window Received Traffic tab allows for the configuration of thresholds for received traffic from its joined ZoneRangers. When Enable monitoring the Overall thresholds or Enable monitoring the Per ZoneRanger thresholds is checked, if a threshold is exceeded, an entry will be logged in the Ranger Gateway Log. If the Send notifications checkbox is checked for Overall or Per ZoneRanger thresholds, then an SNMP Trap will be generated if a threshold is exceeded.

The traffic rate is calculated for each one second interval and the highest rate is compared with the thresholds. For example, if the Trap threshold is set for 100 traps/sec, the interval is 5 minutes, and a burst of 105 traps occurs during one second and even if no other traps are received during the 5 minutes, the maximum one second traffic rate is 105 requests/sec which exceeds the threshold.

---

**Figure 36-24: Gateway Settings..Traffic window Received Traffic**

---

**ZoneRanger 5.6 User's Guide**  
240
The **Gateway Settings..Traffic** window **Proxied Traffic** tab allows for the configuration of thresholds for proxied traffic through the Ranger Gateway to its joined ZoneRangers. When **Enable monitoring the Overall thresholds** or **Enable monitoring the Per ZoneRanger thresholds** is checked, if a threshold is exceeded, an entry will be logged in the Ranger Gateway Log. If the **Send notifications** checkbox is checked for Overall or Per ZoneRanger thresholds, then an SNMP Trap will be generated if a threshold is exceeded.

The traffic rate is calculated for each one second interval and the highest rate is compared with the thresholds. For example, if the SNMP threshold is set for 100 requests/sec, the interval is 5 minutes, and a burst of 105 proxy requests occurs during one second and even if no other SNMP requests are received during the 5 minutes, the maximum one second traffic rate is 105 proxy requests/sec which exceeds the threshold.

**Gateway Viewer Settings...**

The **Gateway Viewer Settings** windows allows configuration of Gateway Viewer specific information. Note that the appearance of the Gateway Viewer Settings dialog depends on the operating system being used for the Ranger Gateway. The Windows version of the dialog is shown in the following figure.

The Solaris/Linux version of the Gateway Viewer Settings dialog provides additional settings, as shown in the following figure.
The **Refresh Interval** field, common to both versions of the dialog, specifies how frequently the list of joined ZoneRangers and the status information for the currently selected ZoneRanger are updated. The default is 30 seconds.

The **Browser Path** list, shown only on Solaris/Linux systems, specifies a list of paths where the Ranger Gateway Viewer should look when trying to locate a web browser to be used when viewing the Ranger Gateway log or when the **Browse (HTTP)** or **Browse (HTTPS)** button have been clicked. When searching for a web browser, the Ranger Gateway viewer will try each path entry in the list, until the first web browser is found. Note that path entries can be absolute or relative. In the case of relative entries, the Ranger Gateway Viewer will search based on the configured operating system PATH environment variable. As such, with the configuration shown in the previous figure, the Ranger Gateway Viewer will simply look to see if any of netscape, mozilla, firefox, or opera can be found using the configured operating system path, and will return the first one found. In the case of absolute entries, the Ranger Gateway Viewer will simply search for the specified file. The simplest way to force a specific web browser to be used is to configure this list with a single absolute path entry (e.g. `/usr/bin/firefox`).

The **Move Up** and **Move Down** buttons may be used to change the order of browser path entries. The **Test** button can be used to identify the browser to be chosen based on the current browser path configuration. To remove a browser from the list, select the entry and click the **Delete** button.

**Tools**

**TFTP Manager...**

The **Tools > TFTP Manager...** window is used to upload files from the Ranger Gateway to the TFTP server on the selected ZoneRanger, download files from the TFTP server on the selected ZoneRanger to the Ranger Gateway, and to delete files on the TFTP server on the selected ZoneRanger.
The **Upload Directory** and **Download Directory** lists correspond to directories on the Ranger Gateway:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux Upload</td>
<td><code>install_dir/upload/tftp</code></td>
</tr>
<tr>
<td>Linux Download</td>
<td><code>install_dir/download/zoneranger/tftp</code></td>
</tr>
<tr>
<td>Solaris Upload</td>
<td><code>install_dir/upload/tftp</code></td>
</tr>
<tr>
<td>Solaris Download</td>
<td><code>install_dir/download/zoneranger/tftp</code></td>
</tr>
<tr>
<td>Windows Upload</td>
<td><code>install_dir\upload\tftp</code></td>
</tr>
<tr>
<td>Windows Download</td>
<td><code>install_dir\download\zoneranger\tftp</code></td>
</tr>
</tbody>
</table>

Where `zoneranger` is the name of the ZoneRanger from which the files in the directory have been downloaded.

Where `install_dir` is the directory where the Ranger Gateway software is installed.

To transfer a file from the Ranger Gateway to the selected ZoneRanger, select the file in the **Upload Directory** list. Then, click **Upload File**. The selected file is copied to the selected ZoneRanger.

To transfer a file from the selected ZoneRanger to the Ranger Gateway, select the file in the **TFTP Directory** list. Then, click **Download File**. The selected file is copied from the selected ZoneRanger.

To remove a file from the TFTP directory of the selected ZoneRanger, select the file in the **TFTP Directory** list. Then, click **Remove File**.

You can use the **Refresh** button to update the lists.

**Patch Manager…**

The Tools > **Patch Manager…** is used to apply and remove patches on the selected ZoneRanger.
The Available Patches list contains all patch files in the patch directory on the Ranger Gateway, minus any patches that have already been uploaded or installed on the selected ZoneRanger. The Uploaded Patches list contains all patch files which have been uploaded to the ZoneRanger but have yet to be applied on the selected ZoneRanger. The Applied Patches list contains all of the patches which have been installed on the selected ZoneRanger.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>install_dir/upload/patch</td>
</tr>
<tr>
<td>Solaris</td>
<td>install_dir/upload/patch</td>
</tr>
<tr>
<td>Windows</td>
<td>install_dir\upload\patch</td>
</tr>
</tbody>
</table>

Where install_dir is the directory where the Ranger Gateway software is installed.

Before attempting to apply a patch, you must copy the corresponding patch file into the Ranger Gateway's patch directory.

To upload an available patch, select the patch in the Available Patches list and click Upload.

To apply an available patch, select the patch in the Uploaded Patches list and click Apply.

To remove an uploaded patch, select the patch in the Uploaded Patches list and click Remove.

To remove an applied patch, select the patch in the Applied Patches list and click Remove.

To view patch specific information, select the patch and click Info.

You can use the Refresh button to update the lists.

**Shutdown ZoneRanger...**

The Tools > Shutdown ZoneRanger... is used to restart, reboot, or shutdown the selected ZoneRanger.
The **Tools > Shutdown ZoneRanger** window provides the ability to restart, reboot, or shutdown a ZoneRanger. Select the appropriate radio button and click **OK**, or click **Cancel** to close the **Shutdown** window.

### Help

#### Help Contents

The **Help > Help Contents** window contains detailed information about the configuration options available from the Ranger Gateway Viewer. This window is also available from each of the individual configuration windows.

### About Ranger Gateway...

The **Help > About Ranger Gateway...** window displays version information for Ranger Gateway Viewer.
Figure 36-32. Help .. About Ranger Gateway Window
Chapter 37: ZoneRanger Text Interface

Using the ZoneRanger Text Interface

The ZoneRanger text interface provides the ability to view and configure a ZoneRanger providing a mechanism for configuration automation. This interface is accessible when using Telnet or SSH to access the ZoneRanger. Only users with admin security level are allowed to access this interface.

The ZoneRanger Text-based interface provides the following set of commands.

Special Character Handling

The ZoneRanger text interface has special characters which have a particular behavior.

? (Question Mark) Question mark produces contextual help based on the preceding text.
\ (Backslash) Escapes the next character to remove any special processing of the next character.
" (Quotation Marks) Quotation marks are used to enclose a literal string especially when entering white space characters.

In order to have one of the above characters entered without special treatment, the backslash (\) characters need to be entered before the special character. For example, to have the literal ? character, \? would need to be entered. Be aware that when escaping a character as the above, the backslash character will no longer appear on the command line.

Command Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ZoneRanger 5.6 User's Guide 247
access-control
access-control-server-group
arp
discovery
exit
findroute
forward
group
history
icmpr
join
message-system
node
ntp
ping
polling
radius
resolve
root-cause
route
scan
shell
show
snmp
snmpwalk
system
tacacs
tcp
tftp
time
traceroute
traffic
trap-filter
vlan
whitelist

access control

To manage the users and passwords on ZoneRanger to access the web and text interfaces, as well as the ZoneRanger database and setup menu.

```
access-control [ database-password db_password | setup-password setup_password | users ]
```

Syntax Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>database-password</td>
<td>Database password settings</td>
</tr>
<tr>
<td>db_password</td>
<td>Password (at least 5 characters)</td>
</tr>
<tr>
<td>setup-password</td>
<td>Setup password settings</td>
</tr>
<tr>
<td>setup_password</td>
<td>Password (at least 5 characters)</td>
</tr>
<tr>
<td>users</td>
<td>ZoneRanger Web and Text Interface users</td>
</tr>
</tbody>
</table>
Usage Guidelines

The **access control users** command will immediately enter a configuration submode. Unless **cancel** is issued, the ZoneRanger software will restart when the users information is saved.

Once you are in the users configuration submode, the following configuration commands are available:

- **cancel** - Exit this mode without saving any changes
- **exit** – exit users mode (saving changes)
- **no** – remove or use default settings
- **user** – user to configure

```
user user_name password [ administrator | operator ]
```

```
no user user_name password [ administrator | operator ]
```

<table>
<thead>
<tr>
<th>command</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>User to configure.</td>
</tr>
<tr>
<td>user_name</td>
<td>User name to configure</td>
</tr>
<tr>
<td>password</td>
<td>Password for user. Must be at least 5 characters</td>
</tr>
<tr>
<td>administrator</td>
<td>User is administrator level</td>
</tr>
<tr>
<td>operator</td>
<td>User is operator level</td>
</tr>
<tr>
<td>no</td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

Example

This example shows how to add users:

```
zr# access control users
zr(user)# user topdog secretpassword administrator
zr(user)#exit
```

**access-control-server-group**

To add a group of access control servers to be used for TACACS+ and RADIUS authentication to ZoneRanger managed devices or the ZoneRanger itself. To remove a group, use the **no** form of this command.

```
access-control-server-group group_name
```

```
no access-control-server-group group_name
```

Syntax Description

<table>
<thead>
<tr>
<th>group_name</th>
<th>Name of the access group.</th>
</tr>
</thead>
</table>
Usage Guidelines

Once you are in the access control configuration submode, you can add any number of server group entries using the `group-entry` clause. You may also add a single `tacacs` and/or `radius` clause for this access control server group. If multiple `tacacs` or `radius` clauses are entered, the last one will be used. Once you are in the access control server group configuration submode, the following configuration commands are available:

- `cancel` - Exit this mode without saving any changes
- `clear-group-entries` – Clear all server group entries
- `exit` – exit server group mode (saving changes)
- `group-entry` – Add a server to this access control server group.
- `no` – remove or use default settings
- `radius` – RADIUS-specific server group settings
- `tacacs` – TACACS+-specific server group settings

```plaintext
group-entry ranger_gateway access_host TACACS_Port RADIUS_Auth_Port RADIUS_Acct_Port

no group-entry ranger_gateway access_host TACACS_Port RADIUS_Auth_Port RADIUS_Acct_Port
```

<table>
<thead>
<tr>
<th>group-entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adds an access control server to the group.</td>
</tr>
<tr>
<td>ranger_gateway</td>
</tr>
<tr>
<td>access_host</td>
</tr>
<tr>
<td>TACACS_Port</td>
</tr>
<tr>
<td>RADIUS_Auth_Port</td>
</tr>
<tr>
<td>RADIUS_Acct_Port</td>
</tr>
<tr>
<td>no</td>
</tr>
</tbody>
</table>

```plaintext
radius key radius_key

no radius key radius_key
```

<table>
<thead>
<tr>
<th>radius key</th>
</tr>
</thead>
<tbody>
<tr>
<td>radius_key</td>
</tr>
<tr>
<td>no</td>
</tr>
</tbody>
</table>

```plaintext
tacacs [ insert-ip | key tacacs_key ]

no tacacs [ insert-ip | key tacacs_key ]
```

| tacacs |
| Adds TACACS+ specific information to this server group |
| insert-ip | Insert source address in rem_addr field of TACACS+ message |
| key | Specifies the TACACS+ key. |
Example

This example shows how to create an access control server group using a TACACS+ server for authentication:

```
zr# access-control-server-group acgroup
zr(server-group)# group-entry rgateway tac1 49 1812 1813
zr(server-group)# tacacs insert-ip
zr(server-group)# tacacs key secretkey
zr(server-group)# exit
```

arp

Perform a diagnostic arp from the ZoneRanger.

arp

Usage Guidelines

Command to perform a diagnostic arp.

Example

This example shows how to issue arp:

```
zr# arp
```

discovery

To configure the discovery settings on a ZoneRanger. To remove a discovery setting, use the no form of this command.

```
discovery [ auto-manage | auto-poll | exclude-network | ignored-address | include-network | period | ping-ranger | search | seed-node | start | tcp-service | tcp-timeout ]
```

```
no discovery [ auto-manage | auto-poll | exclude-network | ignored-address | include-network | period | ping-ranger | search | seed-node | start | tcp-service | tcp-timeout ]
```

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto-manage</td>
<td>Automatically manage newly discovered devices</td>
</tr>
<tr>
<td>auto-poll</td>
<td>Automatically poll newly discovered devices</td>
</tr>
<tr>
<td>exclude-network</td>
<td>Networks excluded from discovery</td>
</tr>
<tr>
<td>ignored-address</td>
<td>Addresses ignored in discovery</td>
</tr>
<tr>
<td>include-network</td>
<td>Networks included in discovery</td>
</tr>
<tr>
<td>period</td>
<td>Periodically perform discovery</td>
</tr>
<tr>
<td>ping-range</td>
<td>Ping ranges to discover</td>
</tr>
</tbody>
</table>

| tacacs_key         | TACACS+ key to use in this server group          |
| no                 | Deletes the TACACS+ key                          |
### Usage Guidelines

Each of the discovery commands will take effect immediately when executed. The `search` clause must include all options on the same statement in the order they are listed.

**discovery auto-manage**

no discovery auto-manage

<table>
<thead>
<tr>
<th>auto-manage</th>
<th>Automatically manage newly discovered devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>Disables automatically managing newly discovered devices</td>
</tr>
</tbody>
</table>

**discovery auto-poll**

no discovery auto-poll

<table>
<thead>
<tr>
<th>auto-poll</th>
<th>Automatically poll newly discovered devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>Disables automatic polling of newly discovered devices</td>
</tr>
</tbody>
</table>

**discovery exclude-network** `ip_address netmask`

no discovery exclude-network `ip_address netmask`

<table>
<thead>
<tr>
<th>exclude-network</th>
<th>Networks excluded from discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ip_address</code></td>
<td>Address of subnet to exclude</td>
</tr>
<tr>
<td><code>netmask</code></td>
<td>Netmask of subnet to exclude</td>
</tr>
<tr>
<td>no</td>
<td>Deletes an exclude address</td>
</tr>
</tbody>
</table>

**discovery ignored-address** `ip_address`

no discovery ignored-address `ip_address`

<table>
<thead>
<tr>
<th>ignored-address</th>
<th>Addresses ignored in discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ip_address</code></td>
<td>Address to ignore</td>
</tr>
<tr>
<td>no</td>
<td>Deletes an ignored address</td>
</tr>
</tbody>
</table>

**discovery include-network** `ip_address netmask`
no discovery include-network ip_address netmask

<table>
<thead>
<tr>
<th>include-network</th>
<th>Networks included from discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address</td>
<td>Address of subnet to include</td>
</tr>
<tr>
<td>netmask</td>
<td>Netmask of subnet to include</td>
</tr>
<tr>
<td>no</td>
<td>Deletes an included address</td>
</tr>
</tbody>
</table>

discovery period discovery_time

no discovery period discovery_time

<table>
<thead>
<tr>
<th>period</th>
<th>Periodically perform discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>discovery_time</td>
<td>Discovery period in hours</td>
</tr>
<tr>
<td>no</td>
<td>Deletes a discovery period</td>
</tr>
</tbody>
</table>

discovery ping-range ip_address_pattern

no discovery ping-range ip_address_pattern

<table>
<thead>
<tr>
<th>ping-range</th>
<th>Ping ranges to discover</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern to look for new devices</td>
</tr>
<tr>
<td>no</td>
<td>Deletes a ping-range</td>
</tr>
</tbody>
</table>

discovery search [ ip-route & arp-cache & broadcast-ping ]

no discovery search [ ip-route & arp-cache & broadcast-ping ]

<table>
<thead>
<tr>
<th>search</th>
<th>Search for additional nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_route</td>
<td>Search IP route tables</td>
</tr>
<tr>
<td>arp-cache</td>
<td>Search ARP caches</td>
</tr>
<tr>
<td>broadcast-ping</td>
<td>Send broadcast pings</td>
</tr>
<tr>
<td>no</td>
<td>Deletes a search criteria</td>
</tr>
</tbody>
</table>

discovery seed-node ip_address

no discovery seed-node ip_address

<table>
<thead>
<tr>
<th>seed-node</th>
<th>Seed nodes to be used by discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address</td>
<td>Seed node IP address</td>
</tr>
<tr>
<td>no</td>
<td>Deletes a seed node</td>
</tr>
</tbody>
</table>

discovery start
**start**  
Start discovery now

**discovery tcp-service**  
TCP_port TCP_service [ auto-discover ]

**no discovery tcp-service**  
TCP_port TCP_service [ auto-discover ]

<table>
<thead>
<tr>
<th>tcp-service</th>
<th>TCP services to discover</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP_port</td>
<td>TCP port to discover</td>
</tr>
<tr>
<td>TCP_service</td>
<td>TCP service to discover</td>
</tr>
<tr>
<td>auto-discover</td>
<td>Auto-discover port on newly discovered devices (optional)</td>
</tr>
<tr>
<td>no</td>
<td>Deletes a tcp service from discovery</td>
</tr>
</tbody>
</table>

**discovery tcp-timeout**  
timeout

**no discovery tcp-timeout**  
timeout

<table>
<thead>
<tr>
<th>tcp-timeout</th>
<th>Timeout for TCP connections, in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>Timeout, in seconds</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to create a set of discovery rules:

```
zr# discovery auto-manage
zr# discovery auto-poll
zr# discovery include-network 10.0.0.0 255.0.0.0
zr# discovery exclude-network 11.10.0.0 255.255.0.0
zr# discovery ping-range 10.3.4.*
zr# discovery ignored-address 10.2.3.4
zr# discovery tcp-service 256 custom
zr# discovery tcp-timeout 3
zr# discovery seed-node 10.10.10.1
zr# discovery period 24
zr# discovery search ip-route arp-cache broadcast-ping
zr# discovery start
```

**findroute**

Perform a diagnostic findroute using SNMP information between two devices.

**findroute** hostname1 hostname2

**Syntax Description**

| hostname1          | Hostname or IP address of starting device |
Usage Guidelines

Command to determine the route using SNMP information between the two specified devices.

Example

This example shows how to find the route between zr1 and node2:

```plaintext
zr# findroute zr1 node2
```

**forward**

To add a forwarding rule to forward UDP data from the ZoneRanger to the indicated Ranger Gateway. To remove a forwarding rule, use the `no` form of this command.

Forward [ dest-group | log-level | netflow | generic | sflow | syslog | syslog-options | trap ] options

no forward [ dest-group | log-level | netflow | generic | sflow | syslog | syslog-options | trap ] options

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dest-group</td>
<td>Destination Groups</td>
</tr>
<tr>
<td>log-level</td>
<td>Logging level for forwarding</td>
</tr>
<tr>
<td>netflow</td>
<td>Netflow forwarding rules</td>
</tr>
<tr>
<td>generic</td>
<td>Generic forwarding rules</td>
</tr>
<tr>
<td>sflow</td>
<td>Sflow forwarding rules</td>
</tr>
<tr>
<td>snmp</td>
<td>SNMPv3 options</td>
</tr>
<tr>
<td>syslog</td>
<td>Syslog forwarding rules</td>
</tr>
<tr>
<td>syslog-options</td>
<td>Options for syslog forwarding</td>
</tr>
<tr>
<td>trap</td>
<td>Trap forwarding rules</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Each of the forward commands will take effect immediately when executed. The `syslog`, `trap` and `dest-group` clauses use configuration submodes to further define the forwarding rule.

**forward** dest-group *group_name*

**no forward** dest-group *group_name*

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>group_name</td>
<td>Destination group name.</td>
</tr>
<tr>
<td>no</td>
<td>Deletes destination group</td>
</tr>
</tbody>
</table>
Once you have entered a destination group name, the destination group submode allows you to enter destination group entries. Once you are in the destination group submode, the following configuration commands are available:

- **cancel** - Exit this mode without saving any changes
- **clear-group-entries** – Clear all destination group entries
- **exit** – exit destination group mode (saving changes)
- **group-entry** – Destination group entry
- **list** – List the current set of destination group entries.
- **no** – remove or use default settings

**group-entry [ gateway_name hosts | dest-group group_name | data-diode ]**

**no group-entry [ gateway_name hosts | dest-group group_name | data-diode ]**

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>gateway_name</td>
<td>Hostname or IP address of a joined Ranger Gateway.</td>
</tr>
<tr>
<td>hosts</td>
<td>Comma separated list of destination IP addresses or hostnames</td>
</tr>
<tr>
<td>dest-group</td>
<td>Adds an already defined destination group as a rule</td>
</tr>
<tr>
<td>group_name</td>
<td>Destination group name to add as a rule</td>
</tr>
<tr>
<td>data-diode</td>
<td>Adds Data Diode as the destination rule</td>
</tr>
<tr>
<td>no</td>
<td>Removes a destination group entry</td>
</tr>
</tbody>
</table>

**forward log-level [ none | short | full ]**

**no forward log-level [ none | short | full ]**

<table>
<thead>
<tr>
<th>log-level</th>
<th>Configure logging level for forwarding</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No logging (default)</td>
</tr>
<tr>
<td>short</td>
<td>Message header or length is logged</td>
</tr>
<tr>
<td>full</td>
<td>Entire message is logged</td>
</tr>
<tr>
<td>no</td>
<td>Delete forwarding log level</td>
</tr>
</tbody>
</table>

**forward netflow local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ] destination_host_port [ source_addresses | enable | disable ]**

**no forward netflow local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ] destination_host_port [ source_addresses | enable | disable ]**

<table>
<thead>
<tr>
<th>local_port</th>
<th>ZoneRanger port to receive NetFlow packets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ranger_gateway</td>
<td>Hostname or IP address of a joined Ranger Gateway.</td>
</tr>
<tr>
<td>destination_host</td>
<td>Hostname or IP address to which ZoneRanger should forward NetFlow packets</td>
</tr>
<tr>
<td>dest-group</td>
<td>Forward to destination group</td>
</tr>
</tbody>
</table>
### NetFlow Forwarding Rules

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>group-name</strong></td>
<td>Destination group to which to forward NetFlow packets</td>
</tr>
<tr>
<td><strong>data-diode</strong></td>
<td>Forward to Data Diode</td>
</tr>
<tr>
<td><strong>destination_host_port</strong></td>
<td>Port on hostname or IP address to which ZoneRanger should forward NetFlow packets</td>
</tr>
<tr>
<td><strong>source_addresses</strong></td>
<td>Source addresses of NetFlow packets to forward. IP address pattern or comma separated list.</td>
</tr>
<tr>
<td><strong>enable</strong></td>
<td>Enable this forwarding rule</td>
</tr>
<tr>
<td><strong>disable</strong></td>
<td>Disable this forwarding rule</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Deletes a NetFlow forwarding rule</td>
</tr>
</tbody>
</table>

**forward generic**

```plaintext
local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ]

destination_host_port [ source_addresses | enable | disable ]
```

**no forward generic**

```plaintext
local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ]

destination_host_port [ source_addresses | enable | disable ]
```

<table>
<thead>
<tr>
<th><strong>local_port</strong></th>
<th>ZoneRanger port to receive generic UDP packets.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ranger_gateway</strong></td>
<td>Hostname or IP address of a joined Ranger Gateway.</td>
</tr>
<tr>
<td><strong>destination_host</strong></td>
<td>Hostname or IP address to which ZoneRanger should forward Generic UDP packets</td>
</tr>
<tr>
<td><strong>dest-group</strong></td>
<td>Forward to destination group</td>
</tr>
<tr>
<td><strong>group-name</strong></td>
<td>Destination group to which to forward Generic UDP packets</td>
</tr>
<tr>
<td><strong>data-diode</strong></td>
<td>Forward to Data Diode</td>
</tr>
<tr>
<td><strong>destination_host_port</strong></td>
<td>Port on hostname or IP address to which ZoneRanger should forward Generic UDP packets</td>
</tr>
<tr>
<td><strong>source_addresses</strong></td>
<td>Source addresses of Generic UDP packets to forward. IP address pattern or comma separated list.</td>
</tr>
<tr>
<td><strong>enable</strong></td>
<td>Enable this forwarding rule</td>
</tr>
<tr>
<td><strong>disable</strong></td>
<td>Disable this forwarding rule</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Deletes a generic forwarding rule</td>
</tr>
</tbody>
</table>

**forward sflow**

```plaintext
local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ]

destination_host_port [ source_addresses | enable | disable ]
```

**no forward sflow**

```plaintext
local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ]

destination_host_port [ source_addresses | enable | disable ]
```

<table>
<thead>
<tr>
<th><strong>local_port</strong></th>
<th>ZoneRanger port to receive sFlow packets.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ranger_gateway</strong></td>
<td>Hostname or IP address of a joined Ranger Gateway.</td>
</tr>
<tr>
<td><strong>destination_host</strong></td>
<td>Hostname or IP address to which ZoneRanger should forward sFlow packets</td>
</tr>
<tr>
<td><strong>dest-group</strong></td>
<td>Forward to destination group</td>
</tr>
</tbody>
</table>
group-name | Destination group to which to forward sFlow packets
--- | ---
data-diode | Forward to Data Diode
destination_host_port | Port on hostname or IP address to which ZoneRanger should forward sFlow packets
source_addresses | Source addresses of sFlow packets to forward. IP address pattern or comma separated list.
enable | Enable this forwarding rule
disable | Disable this forwarding rule
no | Deletes as sFlow forwarding rule

**forward snmp v3-require notifications**

**no forward snmp v3-require notifications**

notifications | Require SNMPv3 users to be configured for processing SNMPv3 traps and informs
--- | ---
no | Do not require SNMPv3 users to be configured.

**forward syslog**

local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ] destination_host_port [ source_addresses | enable | disable ]

**no forward syslog**

local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ] destination_host_port [ source_addresses | enable | disable ]

**local_port** | ZoneRanger port to receive syslog messages.
--- | ---
ranger_gateway | Hostname or IP address of a joined Ranger Gateway.
destination_host | Hostname or IP address to which ZoneRanger should forward syslog messages
dest-group | Forward to destination group
group-name | Destination group to which to forward syslog messages
data-diode | Forward to Data Diode
destination_host_port | Port on hostname or IP address to which ZoneRanger should forward syslog messages
source_addresses | Source addresses of syslog messages to forward. IP address pattern or comma separated list.
enable | Enable this forwarding rule
disable | Disable this forwarding rule
no | Deletes as syslog forwarding rule

Once you have entered a syslog message forwarding rule, the syslog message forwarding submode allows you to enter a syslog filter. Once you are in the syslog message forwarding submode, the following configuration commands are available:

- **cancel** - Exit this mode without saving any changes
- **cisco-severity** – Forward only Cisco sylogs up to a given severity
- **convert** – Forward syslog as another type
- **exit** – exit server group mode (saving changes)
- **facility** – Forward syslogs up to a given facility
- **message** – Forward only syslogs with a given text.
- **not-message** – Forward syslogs that do not have the given text
- **no** – remove or use default settings
- **program** – Forward only syslogs with a given program name
- **severity** – Forward only syslogs up to a given severity

### cisco-severity severity

<table>
<thead>
<tr>
<th>cisco-severity</th>
<th>Forward only Cisco syslogs up to a given severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>severity</td>
<td>Cisco severity level [ 0 -- 7 ]</td>
</tr>
<tr>
<td>no</td>
<td>Deletes the cisco-severity filter</td>
</tr>
</tbody>
</table>

### convert trap trap_type

<table>
<thead>
<tr>
<th>convert</th>
<th>Forward syslog as another type</th>
</tr>
</thead>
<tbody>
<tr>
<td>trap_type</td>
<td>Trap specific type for non-Cisco traps</td>
</tr>
<tr>
<td>no</td>
<td>Deletes the convert filter</td>
</tr>
</tbody>
</table>

### facility facility

<table>
<thead>
<tr>
<th>facility</th>
<th>Forward syslogs up to a given facility</th>
</tr>
</thead>
<tbody>
<tr>
<td>facility</td>
<td>Facility level [ 0 -- 23 ]</td>
</tr>
<tr>
<td>no</td>
<td>Deletes the facility filter</td>
</tr>
</tbody>
</table>

### message message_text regex

<table>
<thead>
<tr>
<th>message</th>
<th>Forward only syslogs with a given text</th>
</tr>
</thead>
<tbody>
<tr>
<td>message_text</td>
<td>Message text to search for. May be regular expression</td>
</tr>
<tr>
<td>regex</td>
<td>Treat message_text as a regular expression</td>
</tr>
<tr>
<td>no</td>
<td>Deletes the message filter</td>
</tr>
</tbody>
</table>

### not-message message_text regex

<table>
<thead>
<tr>
<th>not-message</th>
<th>Forward only syslogs that do not have the given text</th>
</tr>
</thead>
</table>
**message_text**
Message text to search for. May be regular expression

**regex**
Treat message_text as a regular expression

**program** `program_name`

**no program** `program_name`

| **program** | Forward only syslogs with a given program name |
| **program_name** | Program name to search for. |
| **no** | Deletes the program filter |

**severity** `severity_level`

**no severity** `severity_level`

| **severity** | Forward only syslogs up to a given severity |
| **severity_level** | Maximum severity level [ 0 – 7 ] |
| **no** | Deletes the severity filter |

**forward syslog-options require-printable-ascii**

**no forward syslog-options require-printable-ascii**

| **syslog-options** | Configure syslog forwarding options |
| **require-printable-ascii** | Require printable ascii in syslog messages |
| **no** | Do not require printable ascii in syslog messages |

**forward trap** `local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ] destination_host_port [ source_addresses | enable | disable ]`

**no forward trap** `local_port [ ranger_gateway destination_host | dest-group group-name | data-diode ] destination_host_port [ source_addresses | enable | disable ]`

| **local_port** | ZoneRanger port to receive SNMP traps. |
| **ranger_gateway** | Hostname or IP address of a joined Ranger Gateway. |
| **destination_host** | Hostname or IP address to which ZoneRanger should forward SNMP Traps |
| **dest-group** | Forward to destination group |
| **group-name** | Destination group to which to forward SNMP Traps |
| **data-diode** | Forward to Data Diode |
| **destination_host_port** | Port on hostname or IP address to which ZoneRanger should forward SNMP Traps |
| **source_addresses** | Source addresses of SNMP Traps to forward. IP address |
Once you are have entered a SNMP trap forwarding rule, the SNMP trap forwarding submode allows you to enter a specific trap filter. Once you are in the SNMP trap forwarding submode, the following configuration commands are available:

- `cancel` - Exit this mode without saving any changes
- `convert` - Forward syslog as another type
- `exit` - exit server group mode (saving changes)
- `no` - remove or use default settings
- `trap-filter` - Forward traps matching a filter

**Examples**

This example shows how to create a netflow forwarding rule for ZoneRanger port 9996 through Ranger Gateway rg1 to hostname collector at port 999 for all sources matching the IP address ranger 10.1.2.*.

```
zr# trap forward netflow 9996 rg1 collector 999 10.1.2.*
```

This example shows how to create a syslog forwarding rule for ZoneRanger port 512 through Ranger Gateway rg1 to hostname syslog at port 512 for all sources matching the IP address ranger 10.1.2.*, which have a severity of 5 or less and to convert those syslog message to SNMP trap with a specific type of 60.

```
zr# trap forward syslog 512 rg1 syslog 512 10.1.2.*
zr(syslog)# severity 5
zr(syslog)# convert 60
```
zr(syslog)# exit

**group**

To set the group name for this ZoneRanger

```
group groupname
```

**Syntax Description**

<table>
<thead>
<tr>
<th>groupname</th>
<th>Name of the group</th>
</tr>
</thead>
</table>

**Usage Guidelines**

To set the name of the group for this ZoneRanger.

**Example**

This example shows how to change the group name:

```
zr# group newgroup
```

**history**

To set the number of previous commands to cache for the ZoneRanger text interface

```
history number
```

**Syntax Description**

<table>
<thead>
<tr>
<th>number</th>
<th>Number of command to cache greater than zero</th>
</tr>
</thead>
</table>

**Usage Guidelines**

To set the number of previous commands to recall.

**Example**

This example shows how to set the number of commands to recall:

```
zr# history 50
```

**icmp**

To manage the ICMP proxy settings for this ZoneRanger. To remove a ICMP proxy setting, use the `no` form of this command.

```
icmp [ cache | log-level ]
```

```
no icmp [ cache | log-level ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>cache</th>
<th>Manage the ICMP cache settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>log-level</td>
<td>Level of logging on the ZoneRanger for ICMP proxy</td>
</tr>
</tbody>
</table>
Usage Guidelines

Each of the icmp commands will take effect immediately when executed. The clauses are positional significant. Thus each clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

icmp cache cache

no icmp cache cache

<table>
<thead>
<tr>
<th>cache</th>
<th>Enable ICMP proxy caching for this ZoneRanger</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>Disable ICMP proxy caching for this ZoneRanger</td>
</tr>
</tbody>
</table>

icmp cache log-level [ none | short | full ]

no icmp cache log-level [ none | short | full ]

<table>
<thead>
<tr>
<th>log-level</th>
<th>Configure logging level for ICMP proxy caching</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No logging (default)</td>
</tr>
<tr>
<td>short</td>
<td>Basic information is logged</td>
</tr>
<tr>
<td>full</td>
<td>Additional information is logged</td>
</tr>
<tr>
<td>no</td>
<td>Delete ICMP proxy caching log level</td>
</tr>
</tbody>
</table>

icmp cache rule ip_address_pattern positive-cache cache_time [ seconds | minutes | hours ]

negative-cache cache_time [ seconds | minutes | hours ] position index

no icmp rule ip_address_pattern positive-cache cache_time [ seconds | minutes | hours ]

negative-cache cache_time [ seconds | minutes | hours ] position index
<table>
<thead>
<tr>
<th><strong>rule</strong></th>
<th>ICMP proxy caching rule for this ZoneRanger</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ip_address_pattern</strong></td>
<td>IP address pattern to use for this ICMP proxy rule</td>
</tr>
<tr>
<td><strong>positive-cache</strong></td>
<td>Set positive response caching time for this rule</td>
</tr>
<tr>
<td><strong>cache_time</strong></td>
<td>Length of time to cache positive responses</td>
</tr>
<tr>
<td><strong>seconds</strong></td>
<td>Positive cache time in seconds</td>
</tr>
<tr>
<td><strong>minutes</strong></td>
<td>Positive cache time in minutes</td>
</tr>
<tr>
<td><strong>hours</strong></td>
<td>Positive cache time in hours</td>
</tr>
<tr>
<td><strong>negative-cache</strong></td>
<td>Set negative response caching time for this rule</td>
</tr>
<tr>
<td><strong>cache_time</strong></td>
<td>Length of time to cache negative responses</td>
</tr>
<tr>
<td><strong>seconds</strong></td>
<td>Negative cache time in seconds</td>
</tr>
<tr>
<td><strong>minutes</strong></td>
<td>Negative cache time in minutes</td>
</tr>
<tr>
<td><strong>hours</strong></td>
<td>Negative cache time in hours</td>
</tr>
<tr>
<td><strong>position</strong></td>
<td>Position to place ICMP proxy caching rule (optional)</td>
</tr>
<tr>
<td><strong>index</strong></td>
<td>Index position of ICMP proxy caching rule starting at 1</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Delete ICMP proxy caching log level</td>
</tr>
</tbody>
</table>

```
icmp log-level [ none | short | full ]
no icmp log-level [ none | short | full ]
```

<table>
<thead>
<tr>
<th><strong>log-level</strong></th>
<th>Configure logging level for ICMP proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No logging (default)</td>
</tr>
<tr>
<td>short</td>
<td>Basic information is logged</td>
</tr>
<tr>
<td>full</td>
<td>Additional information is logged</td>
</tr>
<tr>
<td>no</td>
<td>Delete ICMP proxy log level</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to set an ICMP proxy caching rule for addresses 10.1.10.* and set the log level to full.

```
zr# icmp cache rule 10.1.10.* positive-cache 30 seconds negative-cache 20 seconds
zr# icmp log-level full
```

**join**

To set the join passcode for this ZoneRanger.

```
join passcode passcode
```
Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passcode</td>
<td>Specifies the passcode</td>
</tr>
<tr>
<td>passcode</td>
<td>Passcode to use for this ZoneRanger</td>
</tr>
</tbody>
</table>

Usage Guidelines

To set the passcode for this ZoneRanger:

Example

This example shows how to set the passcode:

```
zr# join passcode passcode1
```

message-system

To configure the access restrictions and SSL configuration of the ZoneRanger messaging system. To remove a access restrictions and SSL configuration, use the `no` form of this command.

```
message-system [ restricted-address | ssl ]
```

```
no message-system [ restricted-address | ssl ]
```

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>restricted-address</td>
<td>Configure the addresses to which ZoneRanger cannot initiate communications</td>
</tr>
<tr>
<td>ssl</td>
<td>Configure the SSL Trust for ZoneRanger communications</td>
</tr>
</tbody>
</table>

Usage Guidelines

Each of the message-system commands will take effect immediately when executed. The clauses are positional significant. Thus each clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

```
message-system restricted-address ip_address_pattern position index
```

```
no message-system restricted-address ip_address_pattern position index
```

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>restricted-address</td>
<td>Configure the addresses to which ZoneRanger cannot initiate communications</td>
</tr>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern to use for this restriction</td>
</tr>
<tr>
<td>position</td>
<td>Position to place restricted-address rule (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position of rule starting at 1</td>
</tr>
<tr>
<td>no</td>
<td>Delete message-system restricted-address rule</td>
</tr>
</tbody>
</table>

```
message-system ssl [ trusted-subject word position index ]
```
no message-system ssl [ trusted-subject word position index ]

<table>
<thead>
<tr>
<th>ssl</th>
<th>Configure the SSL Trust for ZoneRanger communications</th>
</tr>
</thead>
<tbody>
<tr>
<td>trusted-subject</td>
<td>Specifies a Trusted Subject for ZoneRanger</td>
</tr>
<tr>
<td>word</td>
<td>Specifies the Trusted Subject for ZoneRanger to use</td>
</tr>
<tr>
<td>position</td>
<td>Position to place message-system rule (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position of rule starting at 1</td>
</tr>
<tr>
<td>no</td>
<td>Delete message-system ssl rule</td>
</tr>
</tbody>
</table>

Example

This example shows how to restrict the ZoneRanger from initiating communications with IP addresses 10.1.10.*.

```
zr# message-system restricted-address 10.1.10.*
```

This example shows how to specify a new trusted subject for ZoneRanger to allow for communications.

```
zr# message-system ssl trusted-subject "CN=Ranger Gateway,OU=Engineering,O=Tavve,L=Morrisville,ST=North Carolina,C=US"
```

node

To manage Node Groups for this ZoneRanger. To remove a group, use the **no** form of this command.

```
node group group_name

no node group group_name
```

Syntax Description

| group_name | Name of the Node Group. |

Usage Guidelines

Once you are in the node group configuration submode, you can add any number of node group entries using the **group-entry** clause. Once you are in the node group configuration submode, the following configuration commands are available:

- **cancel** - Exit this mode without saving any changes
- **clear-group-entries** – Clear all node group entries
- **exit** – exit server group mode (saving changes)
- **group-entry** – Add an IP address pattern to this node group.
- **list** – List the server in this node group
- **no** – remove or use default settings

```

group-entry ip_address_pattern
```
**no group-entry ip_address_pattern**

<table>
<thead>
<tr>
<th><strong>group-entry</strong></th>
<th>Adds an ip address pattern to the group.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ip_address_pattern</strong></td>
<td>IP address pattern or another node group (prefix @).</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Deletes an access control server</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to create a node group:

```
  zr# node group ngroup
  zr(node-group)# group-entry 10.1.1.*
  zr(node-group)# group-entry @anotherNodeGroup
  zr(node-group)# exit
```

**ntp**

To configure the NTP proxy settings for this ZoneRanger and its managed devices. To remove a NTP proxy setting, use the **no** form of this command.

```
  ntp [ client-timeout | key | log-level | proxy-server | server-timeout | validate-authentication ]
  no ntp [ client-timeout | key | log-level | proxy-server | server-timeout | validate-authentication ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th><strong>client-timeout</strong></th>
<th>Client session timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>key</strong></td>
<td>NTP authentication key</td>
</tr>
<tr>
<td><strong>log-level</strong></td>
<td>NTP proxy logging level</td>
</tr>
<tr>
<td><strong>proxy-server</strong></td>
<td>Ranger Gateway and server to which to proxy traffic</td>
</tr>
<tr>
<td><strong>server-timeout</strong></td>
<td>Server session timeout</td>
</tr>
<tr>
<td><strong>validate-authentication</strong></td>
<td>Validate authentication keys</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Each of the NTP proxy commands will take effect immediately when executed. The **proxy-server** clauses are positional significant. Thus each **proxy-server** clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

```
  ntp client-timeout timeout
  no ntp client-timeout timeout
```

| **client-timeout** | Amount of time a ZoneRanger waits for a message from an NTP client before closing connection |
**timeout**

NTP client timeout in seconds

**no**

Delete client timeout rule

**ntp key** key_value key position index

**no ntp key** key_index key_value position index

<table>
<thead>
<tr>
<th>key</th>
<th>NTP keys used to validate NTP proxy requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>key_index</td>
<td>NTP index value for this key</td>
</tr>
<tr>
<td>key_value</td>
<td>NTP key value to be used for validation</td>
</tr>
<tr>
<td>position</td>
<td>Position to place key (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position of key starting at 1</td>
</tr>
<tr>
<td>no</td>
<td>Delete NTP key</td>
</tr>
</tbody>
</table>

**ntp log-level** [ none | short | full ]

**no ntp log-level** [ none | short | full ]

| log-level | Configure logging level for NTP proxy     |
| none      | No logging (default)                      |
| short     | Message header is logged                  |
| full      | Entire message is logged                  |
| no        | Delete NTP log level                      |

**ntp proxy-server** ranger_gateway ntp_server position index

**no ntp proxy-server** ranger_gateway ntp_server position index

| proxy-server | Configure Ranger Gateway/NTP server pairs which will process NTP proxy requests from Zoneranger managed devices |
| ranger_gateway | Ranger Gateway through which to send proxy request |
| ntp_server | NTP server to which to send NTP request |
| position | Position to place NTP proxy rule (optional) |
| index | Index position of NTP proxy rule starting at 1 |
| no | Delete NTP proxy server rule |

**ntp server-timeout** timeout
no ntp server-timeout timeout

<table>
<thead>
<tr>
<th>server-timeout</th>
<th>Amount of time a ZoneRanger waits for a message from an NTP server</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>NTP server timeout in seconds</td>
</tr>
<tr>
<td>no</td>
<td>Delete server timeout rule</td>
</tr>
</tbody>
</table>

ntp validate-authentication

no validate-authentication

<table>
<thead>
<tr>
<th>validate-authentication</th>
<th>Validate incoming NTP proxy requests with configured keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>Disable validation</td>
</tr>
</tbody>
</table>

Example

This example shows how to add a NTP proxy rule to proxy NTP requests through gateway1 to an NTP server called ntpserver1.

```
zr# ntp proxy-server gateway1 ntpserver1 position 2
```

ping

Perform a diagnostic ping from the ZoneRanger to an device.

ping address

Syntax Description

| address | Hostname or IP address to ping |

Usage Guidelines

Command to perform a diagnostic ping to a device.

Example

This example shows how to ping device node1:

```
zr# ping node1
```

polling

To configure the ICMP and TCP polling intervals for managed devices . To remove a polling setting, use the no form of this command.

polling [ interface | tcp ]

no polling [ interface | tcp ]
Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>Configure the ICMP polling intervals</td>
</tr>
<tr>
<td>tcp</td>
<td>Configure the TCP polling intervals</td>
</tr>
</tbody>
</table>

Usage Guidelines

Each of the polling commands will take effect immediately when executed. The `interface` clauses are positional significant. Thus each `interface` clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

In the `tcp service` clause, the options must be specified in the order they are displayed.

```
polling interface ip_address_pattern interval timeout retries position index

no polling interface ip_address_pattern interval timeout retries position index
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>Configure the ICMP polling intervals</td>
</tr>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern to use for this polling rule</td>
</tr>
<tr>
<td>interval</td>
<td>Polling interval in seconds</td>
</tr>
<tr>
<td>timeout</td>
<td>Polling timeout in seconds</td>
</tr>
<tr>
<td>retries</td>
<td>Number of retries after unsuccessful poll</td>
</tr>
<tr>
<td>position</td>
<td>Position to place polling rule (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position of polling rule starting at 1</td>
</tr>
<tr>
<td>no</td>
<td>Delete polling rule</td>
</tr>
</tbody>
</table>

```
polling tcp default-interval interval

polling tcp service TCP_Port enabled propagate-status interval interval
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default-interval</td>
<td>Configure the TCP default polling interval</td>
</tr>
<tr>
<td>interval</td>
<td>Default TCP polling interval</td>
</tr>
<tr>
<td>service</td>
<td>TCP service to be configured</td>
</tr>
<tr>
<td>enabled</td>
<td>Enable polling for this TCP port</td>
</tr>
<tr>
<td>propagate-status</td>
<td>Propagate the status of this TCP port to the device upon which it resides.</td>
</tr>
<tr>
<td>interval</td>
<td>Configure the interval to poll this TCP port</td>
</tr>
<tr>
<td>interval</td>
<td>Polling interval for this TCP port in seconds</td>
</tr>
</tbody>
</table>

Examples

This example shows how to add ICMP polling rules:

```
zr# polling interface 10.1.*.* 300 2 1
```
This example shows how to add TCP polling rules:

```
zr# polling interface 11.4.3.[1-3] 500 3 2 1
```

radius

To configure the RADIUS access control settings on the ZoneRanger. To remove a RADIUS access control setting, use the `no` form of this command.

```
radius [ access-control | client-timeout | log-level | proxy-rule | server-timeout ]
```

```
no radius [ access-control | client-timeout | log-level | proxy-rule | server-timeout ]
```

Syntax Description

<table>
<thead>
<tr>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-control</td>
<td>Configure the RADIUS access-control for the ZoneRanger itself</td>
</tr>
<tr>
<td>client-timeout</td>
<td>Timeout for RADIUS client session</td>
</tr>
<tr>
<td>log-level</td>
<td>Level of logging on the ZoneRanger for RADIUS</td>
</tr>
<tr>
<td>proxy-rule</td>
<td>Specify which server group is selected for an incoming RADIUS request from</td>
</tr>
<tr>
<td></td>
<td>a managed device.</td>
</tr>
<tr>
<td>server-timeout</td>
<td>Timeout for RADIUS server sessions</td>
</tr>
</tbody>
</table>

Usage Guidelines

Each of the radius commands will take effect immediately when executed. The `proxy-rule` clauses are positional significant. Thus each `proxy-rule` clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

```
radius access-control server-group group_name
```

```
no radius access-control server-group group_name
```

<table>
<thead>
<tr>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-control</td>
<td>Configure ZoneRanger access control using RADIUS</td>
</tr>
<tr>
<td>server-group</td>
<td>Specify the server group ZoneRanger will use for RADIUS authentication</td>
</tr>
<tr>
<td>group_name</td>
<td>Access control server group name</td>
</tr>
<tr>
<td>no</td>
<td>Delete ZoneRanger access control using RADIUS</td>
</tr>
</tbody>
</table>

```
radius client-timeout timeout
```

```
no radius client-timeout timeout
```

<table>
<thead>
<tr>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>client-timeout</td>
<td>Configure RADIUS client timeout</td>
</tr>
<tr>
<td>timeout</td>
<td>RADIUS client timeout in seconds.</td>
</tr>
<tr>
<td>no</td>
<td>Delete RADIUS client timeout</td>
</tr>
</tbody>
</table>
radius log-level  [none|short|full]

no radius log-level

<table>
<thead>
<tr>
<th>log-level</th>
<th>Configure logging level for RADIUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No logging (default)</td>
</tr>
<tr>
<td>short</td>
<td>RADIUS message header is logged</td>
</tr>
<tr>
<td>full</td>
<td>RADIUS message is logged</td>
</tr>
<tr>
<td>no</td>
<td>Delete RADIUS log level</td>
</tr>
</tbody>
</table>

radius proxy-rule  ip_address_pattern server_group position index

no radius proxy-rule  ip_address_pattern server_group position index

<table>
<thead>
<tr>
<th>proxy-rule</th>
<th>Specify server group to process incoming RADIUS message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern of managed devices</td>
</tr>
<tr>
<td>server_group</td>
<td>Server group to proxy incoming RADIUS message to</td>
</tr>
<tr>
<td>position</td>
<td>Position to place proxy rule (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position of rule starting at 1</td>
</tr>
<tr>
<td>no</td>
<td>Delete RADIUS proxy rule</td>
</tr>
</tbody>
</table>

radius server-timeout  timeout

no radius server-timeout  timeout

<table>
<thead>
<tr>
<th>server-timeout</th>
<th>Configure RADIUS server timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>RADIUS server timeout in seconds.</td>
</tr>
<tr>
<td>no</td>
<td>Delete RADIUS server timeout</td>
</tr>
</tbody>
</table>

Examples

This example shows how to configure RADIUS on ZoneRanger:

```
  zr# radius access-control server-group rgroup
  zr# radius client-timeout 300
  zr# radius log-level short
  zr# radius server-timeout 20
  zr# radius proxy-rule 10.*.*.* rgroup1
  zr# radius proxy-rule 10.1.3.* rgroup2 1
```

resolve

Perform a diagnostic name resolution of a hostname or IP address from the ZoneRanger.

resolve address
Syntax Description

| address | Hostname or IP address to resolve |

Usage Guidelines

Command to perform a diagnostic name resolution of a hostname or IP address.

Example

This example shows how to resolve device node1:

`zr# resolve node1`

root-cause

To modify the root-cause configuration of the ZoneRanger. To remove a root-cause setting, use the **no** form of this command.

`root-cause [ ip | tcp ]`

Syntax Description

| ip       | Configure the IP root cause settings on this ZoneRanger |
| tcp      | Configure the TCP root cause settings on this ZoneRanger |

Usage Guidelines

Each of the root-cause commands will take effect immediately when executed.

```
root-cause ip [ description descr | down-verify-pings down_pings | down-verify-time down_time ]
email [ from email_addr | ranger-gateway rg | recipients addresses ]
| up-verify-pings up_pings | up-verify-time up_time ]
```

```
no root-cause ip [ description descr | down-verify-pings down_pings | down-verify-time down_time ]
email [ from email_addr | ranger-gateway rg | recipients addresses ]
| up-verify-pings up_pings | up-verify-time up_time ]
```

<p>| description | Description used for trap notifications |
| descr       | Description used as part of root cause traps |
| down-verify-pings | Number of ICMP pings to send to verify a device is down |
| down_pings  | Number of pings greater than 1 |
| down-verify-time | Time in seconds to verify a device is down |
| down_time   | Time in seconds |
| email       | Email configuration to send root cause notification |
| from        | Send email directly from the ZoneRanger |
| email_addr  | Email addresses separated by commas |
| ranger-gateway | Send email through specified Ranger Gateway |
| rg          | Joined Ranger Gateway through which to send email |</p>
<table>
<thead>
<tr>
<th><strong>recipients</strong></th>
<th>List of email recipients</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>addresses</strong></td>
<td>Email addresses separated by commas</td>
</tr>
<tr>
<td><strong>up-verify-pings</strong></td>
<td>Number of ICMP pings to send to verify a device is up</td>
</tr>
<tr>
<td><strong>up_pings</strong></td>
<td>Number of pings greater than 1</td>
</tr>
<tr>
<td><strong>up-verify-time</strong></td>
<td>Time in seconds to verify a device is up</td>
</tr>
<tr>
<td><strong>up_time</strong></td>
<td>Time in seconds</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Delete the ip root cause rule</td>
</tr>
</tbody>
</table>

**root-cause tcp** [ description descr | down-verify-polls down_polls | down-verify-time down_time | email [ from email_addr | ranger-gateway rg | recipients addresses ] ]

**no root-cause tcp** [ description descr | down-verify-polls down_polls | down-verify-time down_time | email [ from email_addr | ranger-gateway rg | recipients addresses ] ]

<table>
<thead>
<tr>
<th><strong>description</strong></th>
<th>Description used for trap notifications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>descr</strong></td>
<td>Description used as part of root cause traps</td>
</tr>
<tr>
<td><strong>down-verify-polls</strong></td>
<td>Number of TCP polls to send to verify a TCP port is down</td>
</tr>
<tr>
<td><strong>down_polls</strong></td>
<td>Number of TCP polls greater than 1</td>
</tr>
<tr>
<td><strong>down-verify-time</strong></td>
<td>Time in seconds to verify a TCP port is down</td>
</tr>
<tr>
<td><strong>down_time</strong></td>
<td>Time in seconds</td>
</tr>
<tr>
<td><strong>email</strong></td>
<td>Email configuration to send root cause notification</td>
</tr>
<tr>
<td><strong>from</strong></td>
<td>Send email directly from the ZoneRanger</td>
</tr>
<tr>
<td><strong>email_addr</strong></td>
<td>Email addresses separated by commas</td>
</tr>
<tr>
<td><strong>ranger-gateway</strong></td>
<td>Send email through specified Ranger Gateway</td>
</tr>
<tr>
<td><strong>rg</strong></td>
<td>Joined Ranger Gateway through which to send email</td>
</tr>
<tr>
<td><strong>recipients</strong></td>
<td>List of email recipients</td>
</tr>
<tr>
<td><strong>addresses</strong></td>
<td>Email addresses separated by commas</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Delete the TCP root cause rule</td>
</tr>
</tbody>
</table>

**Example**

This example shows how to send root cause notifications through Ranger Gateway ranger1:

```
zr# root-cause ip email ranger-gateway ranger1
```

**route**

To modify the routing table of the ZoneRanger.

```
route [ add | commit | delete | view ]
```
Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a route temporarily to the ZoneRanger routing table</td>
</tr>
<tr>
<td>commit</td>
<td>Commit a route permanently to the ZoneRanger routing table</td>
</tr>
<tr>
<td>delete</td>
<td>Delete a route from the ZoneRanger routing table</td>
</tr>
<tr>
<td>view</td>
<td>View the ZoneRanger routing table</td>
</tr>
</tbody>
</table>

Usage Guidelines

Each of the route commands will take effect immediately when executed. The `add` clause will temporarily add the specified route to the ZoneRanger for a period of 60 seconds. Within that 60 seconds, a corresponding `commit` clause must be executed to permanently add the route to the ZoneRanger routing table. If the 60 seconds expires without a corresponding `commit` clause, the previously adding route will automatically be removed.

```plaintext
route add net_address net_mask gateway_address metric
```

<table>
<thead>
<tr>
<th>add</th>
<th>Add a route temporarily to the ZoneRanger routing table</th>
</tr>
</thead>
<tbody>
<tr>
<td>net_address</td>
<td>Network address to add to the routing table</td>
</tr>
<tr>
<td>net_mask</td>
<td>Network mask for this route</td>
</tr>
<tr>
<td>gateway_address</td>
<td>Gateway address for this route</td>
</tr>
<tr>
<td>metric</td>
<td>Metric for this route (optional)</td>
</tr>
</tbody>
</table>

```plaintext
route commit net_address net_mask gateway_address metric
```

<table>
<thead>
<tr>
<th>commit</th>
<th>Commit a route permanently to the ZoneRanger routing table</th>
</tr>
</thead>
<tbody>
<tr>
<td>net_address</td>
<td>Network address to commit to the routing table</td>
</tr>
<tr>
<td>net_mask</td>
<td>Network mask for this route</td>
</tr>
<tr>
<td>gateway_address</td>
<td>Gateway address for this route</td>
</tr>
<tr>
<td>metric</td>
<td>Metric for this route (optional)</td>
</tr>
</tbody>
</table>

```plaintext
route delete net_address net_mask gateway_address metric
```

<table>
<thead>
<tr>
<th>delete</th>
<th>Remove route permanently from the ZoneRanger routing table</th>
</tr>
</thead>
<tbody>
<tr>
<td>net_address</td>
<td>Network address to remove to the routing table</td>
</tr>
<tr>
<td>net_mask</td>
<td>Network mask for this route</td>
</tr>
<tr>
<td>gateway_address</td>
<td>Gateway address for this route</td>
</tr>
<tr>
<td>metric</td>
<td>Metric for this route (optional)</td>
</tr>
</tbody>
</table>

```plaintext
route view
```

| view | View the current ZoneRanger routing table |
Example

This example shows how to add and remove a route from the ZoneRanger:

```
zr# route add 10.1.2.3 255.255.255.255 10.1.2.1
zr# route commit 10.1.2.3 255.255.255.255 10.1.2.1
zr# route view
zr# route delete 10.1.2.3 255.255.255.255 10.1.2.1
```

scan

Perform a diagnostic TCP or scan from the ZoneRanger to an device.

```
scan [ snmp address | tcp address port_range ]
```

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp</td>
<td>Perform an SNMP scan of the interface table</td>
</tr>
<tr>
<td>tcp</td>
<td>Perform a TCP scan</td>
</tr>
<tr>
<td>address</td>
<td>Hostname or IP address to ping or scan</td>
</tr>
<tr>
<td>port_range</td>
<td>TCP port or TCP port range to scan. Comma separated list or dashed list (eg 22,34 or 100-120) (optional)</td>
</tr>
</tbody>
</table>

Usage Guidelines

Command to perform either an SNMP scan of the interface table of the device specified using the ZoneRanger's configured SNMP settings or a TCP scan of the ZoneRanger configured TCP ports or a set of specified TCP ports

Example

This example shows how to scan device node1:

```
zr# scan snmp node1
zr# scan tcp node1 22,45
zr# scan tcp node1 100-110
```

shell

Modify the options of the text interface shell.

```
shell [ output-lines num_lines | prompt new_prompt | debug level ]
```

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>output-lines</td>
<td>Number of lines to display before pausing output</td>
</tr>
<tr>
<td>num_lines</td>
<td>Number of lines before pausing output</td>
</tr>
<tr>
<td>prompt</td>
<td>Prompt the shell should display</td>
</tr>
<tr>
<td>new_prompt</td>
<td>New string to display as prompt</td>
</tr>
<tr>
<td>debug</td>
<td>Specify debug level of command shell</td>
</tr>
<tr>
<td>level</td>
<td>Debug level which is 1-15</td>
</tr>
</tbody>
</table>
Usage Guidelines

Command to modify options of the text interface shell.

Example

This example shows how to modify command shell options:

```
zr# shell output-lines 15
zr# shell prompt “zr shell >”
zr# shell debug 5
```

show

Display current configuration values or ZoneRanger information.

**show command**

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-control</td>
<td>Display user configuration</td>
</tr>
<tr>
<td>access-control-server-group</td>
<td>Display server group configuration</td>
</tr>
<tr>
<td>configuration</td>
<td>Display all ZoneRanger configuration</td>
</tr>
<tr>
<td>discovery</td>
<td>Display discovery configuration</td>
</tr>
<tr>
<td>forward</td>
<td>Display forwarding configuration</td>
</tr>
<tr>
<td>group</td>
<td>Display group name</td>
</tr>
<tr>
<td>hostname</td>
<td>Display ZoneRanger hostname</td>
</tr>
<tr>
<td>icmp</td>
<td>Display ICMP proxy configuration</td>
</tr>
<tr>
<td>join</td>
<td>Display join configuration</td>
</tr>
<tr>
<td>message-system</td>
<td>Display messaging configuration</td>
</tr>
<tr>
<td>node</td>
<td>Display Node Group configuration</td>
</tr>
<tr>
<td>ntp</td>
<td>Display NTP proxy configuration</td>
</tr>
<tr>
<td>polling</td>
<td>Display polling configuration</td>
</tr>
<tr>
<td>radius</td>
<td>Display RADIUS configuration</td>
</tr>
<tr>
<td>root-cause</td>
<td>Display root cause configuration</td>
</tr>
<tr>
<td>routes</td>
<td>Display ZoneRanger network routes</td>
</tr>
<tr>
<td>shell</td>
<td>Display shell settings</td>
</tr>
<tr>
<td>smtp</td>
<td>Display SNMP configuration</td>
</tr>
<tr>
<td>system</td>
<td>Display ZoneRanger system configuration</td>
</tr>
<tr>
<td>system-information</td>
<td>Display ZoneRanger system information</td>
</tr>
<tr>
<td>tacacs</td>
<td>Display TACACS+ configuration</td>
</tr>
<tr>
<td>tcp</td>
<td>Display TCP proxy configuration</td>
</tr>
<tr>
<td>tftp</td>
<td>Display TFTP configuration</td>
</tr>
<tr>
<td>traffic</td>
<td>Display Traffic configuration</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>trap-filter</td>
<td>Display Trap Filter configuration</td>
</tr>
<tr>
<td>version</td>
<td>Display ZoneRanger version</td>
</tr>
<tr>
<td>whitelist</td>
<td>Display Whitelist configuration</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Command to view the current configuration of the ZoneRanger.

**Example**

This example shows how to show the ZoneRanger system information:

```
zr# show system-information
```

**snmp**

To modify the SNMP configuration of the ZoneRanger. To remove a polling setting, use the `no` form of this command.

```
snmp [ agent | cache | disallowed | disallowed-oid | log-level | manager-rule | user | v3-require ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>Configure the ZoneRanger SNMP agent</td>
</tr>
<tr>
<td>cache</td>
<td>Configure SNMP proxy caching</td>
</tr>
<tr>
<td>disallowed</td>
<td>Configure the list of IP addresses disallowed SNMP access</td>
</tr>
<tr>
<td>disallowed-oid</td>
<td>Configure OIDs that will not be SNMP Proxied</td>
</tr>
<tr>
<td>log-level</td>
<td>Configure the SNMP logging level</td>
</tr>
<tr>
<td>manager-rule</td>
<td>Configure SNMP device management rules</td>
</tr>
<tr>
<td>user</td>
<td>Configure SNMP v3 users</td>
</tr>
<tr>
<td>v3-require</td>
<td>Configure whether or not SNMPv3 users are required</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Each of the `snmp` commands will take effect immediately when executed. The `manager-rules` clauses are positional significant. Thus each `manager-rules` clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

```
snmp agent [ community comm_string | contact contact_string | location loc_string user user_name | v1 | v2c | v3 ]
```

```
no snmp agent [ community comm_string | contact contact_string | location loc_string user user_name | v1 | v2c | v3 ]
```

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>Configure the ZoneRanger SNMP agent</td>
</tr>
<tr>
<td>community</td>
<td>Configure the ZoneRanger SNMP community string</td>
</tr>
<tr>
<td>Param</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>comm_string</td>
<td>ZoneRanger SNMP community string</td>
</tr>
<tr>
<td>contact</td>
<td>Configure the ZoneRanger SNMP sysContact</td>
</tr>
<tr>
<td>contact_string</td>
<td>ZoneRanger SNMP sysContact</td>
</tr>
<tr>
<td>location</td>
<td>Configure the ZoneRanger SNMP sysLocation</td>
</tr>
<tr>
<td>loc_string</td>
<td>ZoneRanger SNMP sysLocation</td>
</tr>
<tr>
<td>user</td>
<td>SNMP v3 users allowed access to ZoneRanger SNMP agent</td>
</tr>
<tr>
<td>user_name</td>
<td>Name of SNMP v3 user allowed access to SNMP agent</td>
</tr>
<tr>
<td>v1</td>
<td>Enable ZoneRanger SNMP v1 agent support</td>
</tr>
<tr>
<td>v2</td>
<td>Enable ZoneRanger SNMP v2 agent support</td>
</tr>
<tr>
<td>v3</td>
<td>Enable ZoneRanger SNMP v3 agent support</td>
</tr>
<tr>
<td>no</td>
<td>Delete ZoneRanger SNMP agent configuration</td>
</tr>
</tbody>
</table>

**snmp cache cache**

**no snmp cache cache**

| cache | Enable SNMP proxy caching for this ZoneRanger |
| no   | Disable SNMP proxy caching for this ZoneRanger |

**snmp cache log-level [ none | short | full ]**

**no snmp cache log-level [ none | short | full ]**

| log-level | Configure logging level for SNMP proxy caching |
| none      | No logging (default)                           |
| short     | Basic information is logged                    |
| full      | Additional information is logged               |
| no        | Delete SNMP proxy caching log level            |

**snmp cache rule object_id cache cache_time [ seconds | minutes | hours ] position index**

**no snmp cache rule object_id cache cache_time [ seconds | minutes | hours ] position index**
rule
object_id
cache
cache_time
seconds
minutes
hours
position
index
no

**snmp disallowed**  *ip_address_pattern*

**no snmp disallowed**  *ip_address_pattern*

<table>
<thead>
<tr>
<th>disallowed</th>
<th>Configure the list of IP addresses disallowed SNMP access</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ip_address_pattern</em></td>
<td>IP address pattern to disallow ZoneRanger SNMP access</td>
</tr>
<tr>
<td>no</td>
<td>Delete ZoneRanger SNMP disallowed configuration</td>
</tr>
</tbody>
</table>

**snmp disallowed-oid**  *ip_address_pattern object_id get_disallowed set_disallowed index*

**no snmp disallowed-oid**  *ip_address_pattern object_id get_disallowed set_disallowed index*

<table>
<thead>
<tr>
<th>disallowed-oid</th>
<th>Configure the list of Object IDs disallowed via SNMP proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ip_address_pattern</em></td>
<td>IP address pattern from which to disallow Object IDs</td>
</tr>
<tr>
<td><em>object_id</em></td>
<td>Object ID to be disallowed</td>
</tr>
<tr>
<td><em>get_disallowed</em></td>
<td>Boolean whether or not SNMP Gets are not allowed</td>
</tr>
<tr>
<td><em>set_disallowed</em></td>
<td>Boolean whether or not SNMP Sets are not allowed</td>
</tr>
<tr>
<td>index</td>
<td>Index position</td>
</tr>
<tr>
<td>no</td>
<td>Delete disallowed Object IDs configuration</td>
</tr>
</tbody>
</table>

**snmp log-level [ none | short | full ]**

**no snmp log-level [ none | short | full ]**

<table>
<thead>
<tr>
<th>log-level</th>
<th>Configure the SNMP logging level</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No SNMP logging</td>
</tr>
<tr>
<td>short</td>
<td>Log basic SNMP information</td>
</tr>
<tr>
<td>full</td>
<td>Log full SNMP information include varbinds.</td>
</tr>
</tbody>
</table>
no
Disable SNMP logging.

**snmp manager** ip_address_pattern comm_string [ v1 | v2c | v3 user] timeout timeout retries retries port port position index

**no snmp manager** ip_address_pattern comm_string [ v1 | v2c | v3 user] timeout timeout retries retries port port position index

<table>
<thead>
<tr>
<th>manager-rules</th>
<th>Configure SNMP device management rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern to disallow ZoneRanger SNMP access</td>
</tr>
<tr>
<td>comm_string</td>
<td>ZoneRanger SNMP community string</td>
</tr>
<tr>
<td>v1</td>
<td>Use SNMP v1 for this rule</td>
</tr>
<tr>
<td>v2</td>
<td>Use SNMP v2 for this rule</td>
</tr>
<tr>
<td>v3</td>
<td>Use SNMP v3 for this rule</td>
</tr>
<tr>
<td>user</td>
<td>SNMP v3 user to use with this rule</td>
</tr>
<tr>
<td>timeout</td>
<td>Specify how long SNMP request should wait</td>
</tr>
<tr>
<td>timeout</td>
<td>How long in seconds a SNMP request waits before a device is reported as unreachable.</td>
</tr>
<tr>
<td>retries</td>
<td>Specify the number of times a SNMP request is retried</td>
</tr>
<tr>
<td>retries</td>
<td>Number of times that a SNMP request is retried.</td>
</tr>
<tr>
<td>port</td>
<td>Specify the port to which to make SNMP requests</td>
</tr>
<tr>
<td>port</td>
<td>Port number used on the host for a SNMP request.</td>
</tr>
<tr>
<td>position</td>
<td>Position to place manager rule (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position</td>
</tr>
<tr>
<td>no</td>
<td>Delete SNMP management rule</td>
</tr>
</tbody>
</table>

**snmp oidsDisallowed** ip_address_pattern comm_string port port position index

**no snmp oidsDisallowed** ip_address_pattern comm_string port port position index

<table>
<thead>
<tr>
<th>oidsDisallowed</th>
<th>Configure OIDs which are not allowed to be SNMP Proxied</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern to disallow ZoneRanger SNMP access</td>
</tr>
<tr>
<td>comm_string</td>
<td>ZoneRanger SNMP community string</td>
</tr>
<tr>
<td>port</td>
<td>Specify the port to which to make SNMP requests</td>
</tr>
<tr>
<td>port</td>
<td>Port number used on the host for a SNMP request.</td>
</tr>
<tr>
<td>position</td>
<td>Position to place manager rule (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position</td>
</tr>
<tr>
<td>no</td>
<td>Delete SNMP management rule</td>
</tr>
</tbody>
</table>

**snmp user** user_name authentication [ md5 | sha ] auth_password privacy [ des | aes128 | aes192 | aes256 ] priv_password
no user  user_name authentication [ md5 | sha ] auth_password privacy [ des | aes128 | aes192 | aes256 ] priv_password

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Configure SNMP v3 users</td>
</tr>
<tr>
<td>user_name</td>
<td>SNMP v3 user name</td>
</tr>
<tr>
<td>authentication</td>
<td>SNMP v3 authentication type</td>
</tr>
<tr>
<td>md5</td>
<td>Use MD5 authentication</td>
</tr>
<tr>
<td>sha</td>
<td>Use SHA authentication</td>
</tr>
<tr>
<td>auth_password</td>
<td>Authentication password (must be at least 8 characters)</td>
</tr>
<tr>
<td>privacy</td>
<td>SNMP v3 privacy type.</td>
</tr>
<tr>
<td>des</td>
<td>Use DES privacy</td>
</tr>
<tr>
<td>aes128</td>
<td>Use AES128 privacy</td>
</tr>
<tr>
<td>aes192</td>
<td>Use AES192 privacy</td>
</tr>
<tr>
<td>aes256</td>
<td>Use AES256 privacy</td>
</tr>
<tr>
<td>priv_password</td>
<td>Privacy password (must be at least 8 characters)</td>
</tr>
<tr>
<td>no</td>
<td>Delete SNMP user</td>
</tr>
</tbody>
</table>

snmp v3-require proxy-requests

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v3-require</td>
<td>Configure whether or not SNMPv3 users are required</td>
</tr>
<tr>
<td>proxy-requests</td>
<td>Require that SNMPv3 users be configured in order to proxy any SNMPv3 requests</td>
</tr>
<tr>
<td>no</td>
<td>Do not require SNMPv3 users.</td>
</tr>
</tbody>
</table>

Examples

This example shows how to set the ZoneRanger SNMP agent settings:

```
zr# snmp agent community new_community
zr# snmp agent location “New York City”
zr# snmp agent user john
zr# snmp agent v3
```

This example shows how to add IP addresses to which the ZoneRanger will not make SNMP requests:

```
zr# snmp disallowed 10.1.2.*
```

This example shows how to add some SNMP management rules:

```
zr# snmp manager-rule 10.1.10.10 1 Community1 timeout 1 retries 1 port 161
zr# snmp manager-rule 10.1.10.10 3 SecureUser Community1 timeout 1 retries 1 port 161 1
```

This example shows how to add some SNMP v3 users:

```
zr# snmp user user1 authentication md5 authpass privacy des privpass
```
snmpwalk

Perform a diagnostic snmpwalk to a hostname or IP address from the ZoneRanger.

**snmpwalk address [ v1 | v2c | v3 ]**

**Syntax Description**

<table>
<thead>
<tr>
<th>address</th>
<th>Hostname or IP address to which to make SNMP request</th>
</tr>
</thead>
</table>

**Usage Guidelines**

Command to perform a diagnostic snmpwalk to a hostname or IP address.

**snmpwalk address [ v1 | v2 ] community_string oid port**

<table>
<thead>
<tr>
<th>address</th>
<th>Hostname or IP address to which to make SNMP request</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>SNMP v1</td>
</tr>
<tr>
<td>v2</td>
<td>SNMP v2</td>
</tr>
<tr>
<td>community_string</td>
<td>SNMP community string to use for this SNMP request</td>
</tr>
<tr>
<td>oid</td>
<td>Requested SNMP oid</td>
</tr>
<tr>
<td>port</td>
<td>Port to which to make request (optional)</td>
</tr>
</tbody>
</table>

**snmpwalk address v3 user_name noAuthNoPriv oid port**

**snmpwalk address v3 user_name authNoPriv [md5|sha] auth_password oid port**

**snmpwalk address v3 user_name authPriv [md5|sha] auth_password dex priv_password oid port**

<table>
<thead>
<tr>
<th>address</th>
<th>Hostname or IP address to which to make SNMP request</th>
</tr>
</thead>
<tbody>
<tr>
<td>v3</td>
<td>SNMP v3</td>
</tr>
<tr>
<td>user_name</td>
<td>SNMP v3 user to use for this request</td>
</tr>
<tr>
<td>noAuthNoPriv</td>
<td>No authentication and no privacy for this request</td>
</tr>
<tr>
<td>authNoPriv</td>
<td>Authentication and no privacy for this request</td>
</tr>
<tr>
<td>authPriv</td>
<td>Authentication and privacy for this request</td>
</tr>
<tr>
<td>auth_password</td>
<td>Authentication password</td>
</tr>
<tr>
<td>priv_password</td>
<td>Privacy password</td>
</tr>
<tr>
<td>oid</td>
<td>Requested SNMP OID</td>
</tr>
<tr>
<td>port</td>
<td>Port to which to make request (optional)</td>
</tr>
</tbody>
</table>
Example

This example shows how to SNMP walk the system table:

```
zr# snmpwalk router1 v1 public 1.3.6.1.2.1.1
```

system

Change the system configuration on the ZoneRanger.

```
system [ dns | host | port | property | reboot | restart | shutdown ]
```

Syntax Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dns</td>
<td>Configure ZoneRanger DNS settings</td>
</tr>
<tr>
<td>host</td>
<td>Configure ZoneRanger host name list</td>
</tr>
<tr>
<td>port</td>
<td>Configure ZoneRanger ports</td>
</tr>
<tr>
<td>property</td>
<td>Configure ZoneRanger properties</td>
</tr>
<tr>
<td>reboot</td>
<td>Reboot ZoneRanger</td>
</tr>
<tr>
<td>restart</td>
<td>Restart ZoneRanger software</td>
</tr>
<tr>
<td>shutdown</td>
<td>Shutdown ZoneRanger</td>
</tr>
</tbody>
</table>

Usage Guidelines

Each of the system commands will take effect immediately when executed. The `dns` clauses are positional significant. Thus each `dns` clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

```
system dns [ search-domain domain_name position index | secondary-dns | server dns_server position index ]
```

```
o system dns [ search-domain domain_name position index | secondary-dns | server dns_server position index ]
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dns</td>
<td>Configure ZoneRanger DNS settings</td>
</tr>
<tr>
<td>search-domain</td>
<td>Domain to search to resolve host names</td>
</tr>
<tr>
<td>domain_name</td>
<td>Domain name to search</td>
</tr>
<tr>
<td>position</td>
<td>Position to place domain name (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position</td>
</tr>
<tr>
<td>secondary-dns</td>
<td>Enable ZoneRanger as a secondary DNS server</td>
</tr>
<tr>
<td>server</td>
<td>Specify a DNS server for ZoneRanger to use for name resolution</td>
</tr>
<tr>
<td>dns_server</td>
<td>DNS server name</td>
</tr>
<tr>
<td>position</td>
<td>Position to place DNS server (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position</td>
</tr>
<tr>
<td>no</td>
<td>Delete DNS specification</td>
</tr>
</tbody>
</table>

ZoneRanger 5.6 User's Guide 284
**system host** `ip_address hostname alias_list`

**no system host** `ip_address hostname alias_list`

<table>
<thead>
<tr>
<th>host</th>
<th>Configure ZoneRanger host name list</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ip_address</code></td>
<td>IP address with which to associate a hostname</td>
</tr>
<tr>
<td><code>hostname</code></td>
<td>Hostname to associate with IP address</td>
</tr>
<tr>
<td><code>alias_list</code></td>
<td>List of aliases to associate with IP address. May be a space separated list enclosed in quotation marks</td>
</tr>
</tbody>
</table>

**system port [ http | https | icmp | messaging | ntp | radius | snmp-agent | snmp-trap | ssh | syslog | tacacs | telnet | tftp | [ disabled | eth0 | eth1 | enabled | ranger-gateway ]**

<table>
<thead>
<tr>
<th>port</th>
<th>Configure ZoneRanger port access</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>http</code></td>
<td>HTTP port 80</td>
</tr>
<tr>
<td><code>https</code></td>
<td>HTTP port 443</td>
</tr>
<tr>
<td><code>icmp</code></td>
<td>ICMP for specified interface</td>
</tr>
<tr>
<td><code>messaging</code></td>
<td>ZoneRanger/Ranger Gateway communications (default 4854)</td>
</tr>
<tr>
<td><code>ntp</code></td>
<td>NTP port 123</td>
</tr>
<tr>
<td><code>radius</code></td>
<td>Radius port 1812 and 1813</td>
</tr>
<tr>
<td><code>snmp-agent</code></td>
<td>SNMP agent port 161</td>
</tr>
<tr>
<td><code>snmp-trap</code></td>
<td>SNMP trap port 162</td>
</tr>
<tr>
<td><code>ssh</code></td>
<td>SSH port 22</td>
</tr>
<tr>
<td><code>syslog</code></td>
<td>Syslog port 514</td>
</tr>
<tr>
<td><code>tacacs</code></td>
<td>TACACS+ port 49</td>
</tr>
<tr>
<td><code>telnet</code></td>
<td>Telnet port 23</td>
</tr>
<tr>
<td><code>tftp</code></td>
<td>TFTP port 69</td>
</tr>
<tr>
<td><code>disabled</code></td>
<td>Port is disabled</td>
</tr>
<tr>
<td><code>eth0</code></td>
<td>Port only available on ZoneRanger eth0 interface</td>
</tr>
<tr>
<td><code>eth1</code></td>
<td>Port only available on ZoneRanger eth1 interface</td>
</tr>
<tr>
<td><code>enabled</code></td>
<td>Port is enabled</td>
</tr>
<tr>
<td><code>ranger-gateway</code></td>
<td>Port is only available from the Ranger Gateway</td>
</tr>
</tbody>
</table>

**system property** `property_name property_value`

**no system property** `property_name property_value`

<table>
<thead>
<tr>
<th>property</th>
<th>Configure ZoneRanger properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>property_name</code></td>
<td>Property name to change/set</td>
</tr>
<tr>
<td><code>property_value</code></td>
<td>Property value to name/set</td>
</tr>
</tbody>
</table>
tacacs

To configure the TACACS+ access control settings on the ZoneRanger. To remove a TACACS+ access control setting, use the no form of this command.

```
tacacs [access-control|client-timeout|log-level|max-size|proxy-rule|server-timeout]
no tacacs [access-control|client-timeout|log-level|max-size|proxy-rule|server-timeout]
```

Syntax Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>access-control</td>
<td>Configure the TACACS+ access-control for the ZoneRanger itself</td>
</tr>
<tr>
<td>client-timeout</td>
<td>Timeout for TACACS+ client session</td>
</tr>
<tr>
<td>log-level</td>
<td>Level of logging on the ZoneRanger for TACACS+</td>
</tr>
<tr>
<td>max-size</td>
<td>Maximum size of a TACACS+ message</td>
</tr>
<tr>
<td>proxy-rule</td>
<td>Specify which server group is selected for an incoming TACACS+ request from a managed device.</td>
</tr>
<tr>
<td>server-timeout</td>
<td>Timeout for TACACS+ server sessions</td>
</tr>
</tbody>
</table>

Usage Guidelines

Each of the TACACS+ commands will take effect immediately when executed. The `proxy-rule` clauses are positional significant. Thus each `proxy-rule` clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

```
tacacs access-control admin-level admin_level
```

```
tacacs access-control command cmd
```

```
tacacs access-control command-required
```

```
tacacs access-control direct-server-entry address port index
```

```
tacacs access-control direct-server-key key
```

```
tacacs access-control login-type [ ascii | pap ]
```

```
tacacs access-control method [ direct | proxy ]
```

```
tacacs access-control operator-level oper_level
```

```
tacacs access-control protocol protocol_string
```

```
tacacs access-control server-group group_name
```

```
tacacs access-control service service_name
```
no tacacs access-control …

<table>
<thead>
<tr>
<th><strong>access-control</strong></th>
<th>Configure access control using TACACS+ for ZoneRanger</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>admin-level</strong></td>
<td>Level associated with admin users</td>
</tr>
<tr>
<td><strong>admin_level</strong></td>
<td>Level (1-15) associated with admin users</td>
</tr>
<tr>
<td><strong>command</strong></td>
<td>Send command on authorization requests</td>
</tr>
<tr>
<td><strong>cmd</strong></td>
<td>Command to be sent on authorization requests</td>
</tr>
<tr>
<td><strong>command-required</strong></td>
<td>Require TACACS+ authorization request to include command</td>
</tr>
<tr>
<td><strong>direct-server-entry</strong></td>
<td>Authenticate directly to TACACS+ server</td>
</tr>
<tr>
<td><strong>address</strong></td>
<td>Hostname or IP address of TACACS+ server</td>
</tr>
<tr>
<td><strong>port</strong></td>
<td>Port to use for authentication on TACACS+ server</td>
</tr>
<tr>
<td><strong>index</strong></td>
<td>Position of direct server starting at 1 (optional)</td>
</tr>
<tr>
<td><strong>direct-server-key</strong></td>
<td>Define shared key used for message encryption</td>
</tr>
<tr>
<td><strong>key</strong></td>
<td>Shared key used to encrypt and decrypt TACACS+ messages</td>
</tr>
<tr>
<td><strong>login-type</strong></td>
<td>Login type to use (ASCII or PAP)</td>
</tr>
<tr>
<td><strong>method</strong></td>
<td>Define method to use to contact TACACS+ server</td>
</tr>
<tr>
<td><strong>direct</strong></td>
<td>Contact TACACS+ server directly</td>
</tr>
<tr>
<td><strong>proxy</strong></td>
<td>Contact TACACS+ server via proxy</td>
</tr>
<tr>
<td><strong>operator-level</strong></td>
<td>Level associated with operator users</td>
</tr>
<tr>
<td><strong>operator_level</strong></td>
<td>Level (1-15) associated with operator users</td>
</tr>
<tr>
<td><strong>protocol</strong></td>
<td>Protocol to which this ZoneRanger login is associated</td>
</tr>
<tr>
<td><strong>server-group</strong></td>
<td>Specify the server group ZoneRanger will use for TACACS+ authentication.</td>
</tr>
<tr>
<td><strong>group_name</strong></td>
<td>Access control server group name</td>
</tr>
<tr>
<td><strong>service</strong></td>
<td>Service to which this ZoneRanger login is associated</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Delete ZoneRanger access control using TACACS+</td>
</tr>
</tbody>
</table>

**tacacs client-timeout**  
*timeout*

**no tacacs client-timeout**  
*timeout*

<table>
<thead>
<tr>
<th><strong>client-timeout</strong></th>
<th>Configure TACACS+ client timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>timeout</strong></td>
<td>TACACS+ client timeout in seconds.</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Delete TACACS+ client timeout</td>
</tr>
</tbody>
</table>

**tacacs log-level** [ none | short | full ]
no tacacs log-level

<table>
<thead>
<tr>
<th>log-level</th>
<th>Configure logging level for TACACS+</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No logging (default)</td>
</tr>
<tr>
<td>short</td>
<td>TACACS+ message header is logged</td>
</tr>
<tr>
<td>full</td>
<td>TACACS+ message is logged</td>
</tr>
<tr>
<td>no</td>
<td>Delete TACACS+ log level</td>
</tr>
</tbody>
</table>

tacacs proxy-rule ip_address_pattern server_group position index

no tacacs proxy-rule ip_address_pattern server_group position index

<table>
<thead>
<tr>
<th>proxy-rule</th>
<th>Specify server group to process incoming TACACS+ message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern of managed devices</td>
</tr>
<tr>
<td>server_group</td>
<td>Server group to proxy incoming TACACS+ message to</td>
</tr>
<tr>
<td>index</td>
<td>Position of rule starting at 1 (optional)</td>
</tr>
<tr>
<td>no</td>
<td>Delete TACACS+ proxy rule</td>
</tr>
</tbody>
</table>

tacacs server-timeout timeout

no tacacs server-timeout timeout

<table>
<thead>
<tr>
<th>server-timeout</th>
<th>Configure TACACS+ server timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>TACACS+ server timeout in seconds.</td>
</tr>
<tr>
<td>no</td>
<td>Delete TACACS+ server timeout</td>
</tr>
</tbody>
</table>

Examples

This example shows how to configure TACACS+ on ZoneRanger:

```
zr# tacacs access-control server-group rgroup
zr# tacacs client-timeout 300
zr# tacacs log-level short
zr# tacacs server-timeout 20
zr# tacacs proxy-rule 10.*.*.* rgroup1
zr# tacacs proxy-rule 10.1.3.* rgroup2 1
```

tcp

To modify the TCP proxy configuration of the ZoneRanger. To remove a TCP proxy setting, use the no form of this command.

tcp log-level [none | short | full] | ftp-active-to-passive
Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftp-active-to-passive</td>
<td>Convert active FTP sessions to passive sessions</td>
</tr>
<tr>
<td>log-level</td>
<td>Configure the TCP proxy logging level</td>
</tr>
</tbody>
</table>

Usage Guidelines

Each of the TCP commands will take effect immediately when executed.

```plaintext
tcp ftp-active-to-passive
no tcp ftp-active-to-passive
tcp log-level [ none | short | full ]
no tcp log-level
```

<table>
<thead>
<tr>
<th>log-level</th>
<th>Configure logging level for TCP proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No logging (default)</td>
</tr>
<tr>
<td>short</td>
<td>Basic information is logged</td>
</tr>
<tr>
<td>full</td>
<td>Additional information is logged</td>
</tr>
<tr>
<td>no</td>
<td>Disable TCP proxy logging</td>
</tr>
</tbody>
</table>

Examples

This example shows how to set the TCP proxy log level.

```
  zr# tcp log-level full
```

**tftp**

To modify the TFTP proxy configuration of the ZoneRanger. To remove a TFTP proxy polling setting, use the `no` form of this command.

```
tftp [ blocknum-wrapping | log-level | proxy-rule | snmp-triggered-rules ]
```

Syntax Description

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>log-level</td>
<td>Configure the TFTP proxy logging level</td>
</tr>
<tr>
<td>blocknum-wrapping</td>
<td>Enable or disable block number wrapping</td>
</tr>
<tr>
<td>proxy-rule</td>
<td>Configure TFTP proxy rules</td>
</tr>
<tr>
<td>snmp-triggered-rules</td>
<td>Enable or disable SNMP triggered TFTP proxy</td>
</tr>
</tbody>
</table>
Usage Guidelines

Each of the TFTP proxy commands will take effect immediately when executed. The **proxy-rule** clauses are positional significant. Thus each **proxy-rule** clause takes an optional index position which determines its place relative to the other rules. The indices start at 1. If no index position is specified, the rule is placed at the bottom of the list.

**tftp log-level**  [ none | short | full ]

**no tftp log-level**

<table>
<thead>
<tr>
<th>log-level</th>
<th>Configure logging level for TFTP proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No logging (default)</td>
</tr>
<tr>
<td>short</td>
<td>Basic information is logged</td>
</tr>
<tr>
<td>full</td>
<td>Additional information, including TFTP rule, is logged</td>
</tr>
<tr>
<td>no</td>
<td>Disable TFTP proxy logging</td>
</tr>
</tbody>
</table>

**tftp proxy-rule**  **ip_address_pattern**  **read**  **write**  **create**  [ to ranger_gateway remote_host remote_port ]  **position**  **index**

**no tftp proxy-rule**  **ip_address_pattern**  **read**  **write**  **create**  [ to ranger_gateway remote_host remote_port ]  **position**  **index**

<table>
<thead>
<tr>
<th>proxy-rule</th>
<th>Configure TFTP proxy rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>Allow TFTP proxy read access</td>
</tr>
<tr>
<td>write</td>
<td>Allow TFTP proxy write access</td>
</tr>
<tr>
<td>create</td>
<td>Allow TFTP proxy create access</td>
</tr>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern to disallow ZoneRanger SNMP access</td>
</tr>
<tr>
<td>to</td>
<td>Specify Ranger Gateway to TFTP proxy directly to or through</td>
</tr>
<tr>
<td>ranger_gateway</td>
<td>Ranger Gateway to TFTP proxy to or through</td>
</tr>
<tr>
<td>remote_host</td>
<td>Hostname or IP address of remote host to which TFTP proxy should send requests</td>
</tr>
<tr>
<td>remote_port</td>
<td>Port on hostname or IP address of remote host to which TFTP proxy should send requests</td>
</tr>
<tr>
<td>position</td>
<td>Position to place TFTP proxy rule (optional)</td>
</tr>
<tr>
<td>index</td>
<td>Index position of rule starting at 1</td>
</tr>
<tr>
<td>no</td>
<td>Delete TFTP proxy rule</td>
</tr>
</tbody>
</table>

**tftp snmp-triggered-rules**  **timeout**

**no tftp snmp-triggered-rules**  **timeout**

<table>
<thead>
<tr>
<th>snmp-triggered-rules</th>
<th>Enable or disable SNMP triggered TFTP proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>Length of time in seconds a temporary rules exists</td>
</tr>
</tbody>
</table>
Examples

This example shows how to add a TFTP proxy rule to allow 10.1.1.1 to read and write TFTP files through Ranger Gateway gateway1 to a TFTP server tftpserver on port 69.

```
zr# tftp proxy-rule 10.1.1.1 read write to gateway1 tftpserver 69
```

**time**

To configure the time setting on the ZoneRanger itself. To remove a time setting, use the `no` form of this command.

```
time [ gateway | ntp | time-protocol ]
```

```
no time [ gateway | ntp | time-protocol ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gateway</td>
<td>Retrieve time from Ranger Gateway. May cause restart.</td>
</tr>
<tr>
<td>ntp</td>
<td>Retrieve time from an NTP server.</td>
</tr>
<tr>
<td>time-protocol</td>
<td>Retrieve time from server supporting RFC 868. May cause restart.</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Each of the time commands will take effect immediately when executed. The `gateway` and `time-protocol` clauses may cause the ZoneRanger software to restart.

```
time gateway ranger_gateway
```

```
no time gateway ranger_gateway
```

```
gateway  | Retrieve time from a joined Ranger Gateway
ranger_gateway | Joined Ranger Gateway from which to retrieve time
no     | Disables retrieving time from a Ranger Gateway
```

```
time ntp [ direct-server server_name key_index | enabled | proxy-server ranger_gateway ntp_server key_index | key key_index key_string | server-authentication-enabled | server-enabled ]
```

```
no time ntp [ direct-server server_name key_index | enabled | proxy-server ranger_gateway ntp_server key_index | key key_index key_string | server-authentication-enabled | server-enabled ]
```

```
network | Retrieve ZoneRanger time from a NTP server
server | Retrieve ZoneRanger from a NTP server directly.
s_server_name | NTP server name from which to directly retrieve time
```
**key_index**
- Authentication key index which must already be defined

**enabled**
- Synchronize ZoneRanger time using NTP

**proxy-server**
- Retrieve ZoneRanger time through Ranger Gateway.

**ranger_gateway**
- Retrieve ZoneRanger time from a NTP server through this joined Ranger Gateway

**ntp_server**
- NTP server name from which to retrieve time

**key_index**
- Authentication key index which must already be defined

**key**
- Defines an NTP authentication key

**key_index**
- Authentication key index

**key_string**
- Authentication key string

**server-authentication-enabled**
- ZoneRanger authenticates client requests

**server_enabled**
- ZoneRanger acts as a NTP server.

**no**
- Disables retrieving time from a NTP server

### time time-protocol server_name

**no time time-protocol server_name**

<table>
<thead>
<tr>
<th>time-protocol</th>
<th>Retrieve time from a host that supports RFC 868</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_name</td>
<td>Name of server that supports RFC 868</td>
</tr>
<tr>
<td>no</td>
<td>Disables retrieving time from server that supports RFC 868</td>
</tr>
</tbody>
</table>

### traceroute

Perform a diagnostic traceroute from the ZoneRanger to an device.

**traceroute address**

**Syntax Description**

<table>
<thead>
<tr>
<th>address</th>
<th>Hostname or IP address to ping</th>
</tr>
</thead>
</table>

**Usage Guidelines**

Command to perform a network traceroute from the ZoneRanger to the specified device.

**Example**

This example shows how to traceroute device node1:

```
zr# traceroute node1
```
**traffic**

To configure the traffic configuration settings on the ZoneRanger. To remove a traffic configuration setting, use the `no` form of this command.

```
traffic [ forwarded | interval | log-level | proxied ]
```

```
no traffic [ forwarded | interval | log-level | proxied ]
```

**Syntax Description**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>forwarded</code></td>
<td>Configure the forwarded traffic thresholds</td>
</tr>
<tr>
<td><code>interval</code></td>
<td>Interval to check traffic thresholds (in seconds)</td>
</tr>
<tr>
<td><code>log-level</code></td>
<td>Level of traffic logging on the ZoneRanger</td>
</tr>
<tr>
<td><code>proxied</code></td>
<td>Configure the proxied traffic thresholds</td>
</tr>
</tbody>
</table>

**Usage Guidelines**

Each of the traffic commands will take effect immediately when executed.

```
traffic forwarded [ all | per ] [ <cr> | notify [ <cr> [ [ total | generic | netflow | sflow | syslog | trap ] threshold ] ] ]
```

```
no traffic forwarded ...
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>forwarded</code></td>
<td>Configure forwarded threshold information for ZoneRanger</td>
</tr>
<tr>
<td><code>all</code></td>
<td>All traffic</td>
</tr>
<tr>
<td><code>per</code></td>
<td>Per IP address traffic</td>
</tr>
<tr>
<td><code>&lt;cr&gt;</code></td>
<td>Enable threshold monitoring</td>
</tr>
<tr>
<td><code>notify</code></td>
<td>Traffic notification and thresholds</td>
</tr>
<tr>
<td><code>&lt;cr&gt;</code></td>
<td>Enable notification if threshold exceeded</td>
</tr>
<tr>
<td><code>total</code></td>
<td>All traffic</td>
</tr>
<tr>
<td><code>generic</code></td>
<td>Generic UDP traffic</td>
</tr>
<tr>
<td><code>netflow</code></td>
<td>NetFlow traffic</td>
</tr>
<tr>
<td><code>sflow</code></td>
<td>sflow traffic</td>
</tr>
<tr>
<td><code>syslog</code></td>
<td>syslog traffic</td>
</tr>
<tr>
<td><code>trap</code></td>
<td>SNMP Trap traffic</td>
</tr>
<tr>
<td><code>threshold</code></td>
<td>Threshold value for traffic type</td>
</tr>
<tr>
<td><code>no</code></td>
<td>Delete traffic configuration</td>
</tr>
</tbody>
</table>

```
traffic interval value
```
**interval**
Measurement interval over which thresholds are calculated

**value**
Value of interval in seconds (minimum is 60).

**traffic log-level [ none | short | full ]**

**no traffic log-level**

<table>
<thead>
<tr>
<th>log-level</th>
<th>Configure logging level for traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No logging (default)</td>
</tr>
<tr>
<td>short</td>
<td>Traffic totals are logged at each measurement interval</td>
</tr>
<tr>
<td>full</td>
<td>Traffic counts per IP address are logged at each measurement interval</td>
</tr>
<tr>
<td>no</td>
<td>Delete traffic log level</td>
</tr>
</tbody>
</table>

**traffic proxied [ all | per ] [ <cr> | notify [ <cr> [ [ icmp | ntp | radius | snmp | tacacs ] threshold ] ] ]**

**no traffic proxied …**

<table>
<thead>
<tr>
<th>forwarded</th>
<th>Configure proxied threshold information for ZoneRanger</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>All traffic</td>
</tr>
<tr>
<td>per</td>
<td>Per IP address traffic</td>
</tr>
<tr>
<td>&lt;cr&gt;</td>
<td>Enable threshold monitoring</td>
</tr>
<tr>
<td>notify</td>
<td>Traffic notification and thresholds</td>
</tr>
<tr>
<td>&lt;cr&gt;</td>
<td>Enable notification if threshold exceeded</td>
</tr>
<tr>
<td>icmp</td>
<td>ICMP proxy requets</td>
</tr>
<tr>
<td>ntp</td>
<td>NTP proxy requests</td>
</tr>
<tr>
<td>radius</td>
<td>RADIUS proxy requests</td>
</tr>
<tr>
<td>snmp</td>
<td>SNMP proxy requests,</td>
</tr>
<tr>
<td>tacacs</td>
<td>TACACS+ proxy requests</td>
</tr>
<tr>
<td>threshold</td>
<td>Threshold value for IP address</td>
</tr>
<tr>
<td>no</td>
<td>Delete traffic configuration</td>
</tr>
</tbody>
</table>

**trap-filter**

To add a trap filter to be used by trap forwarding rules. To remove a trap filter, use the **no** form of this command.

**trap-filter filter_name**

**no trap-filter filter_name**
Syntax Description

| filter_name | Name of the filter.

Usage Guidelines

Trap filters are created within the trap filter configuration submode. Once you are in the trap filter configuration submode, the following configuration commands are available:

- **all-conditions** – All conditions must be true to pass filter
- **any-condition** – At least one condition must be true to pass filter
- **cancel** – Exit this mode without saving any changes
- **clear-conditions** – Clear all conditions
- **condition** – Define a new condition
- **exit** – exit server group mode (saving changes)
- **no** – remove or use default settings
- **not** – Negate a condition

**condition agent ip_address_pattern**

**no condition agent ip_address_pattern**

<table>
<thead>
<tr>
<th>condition</th>
<th>Adds a filtering condition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>Specify agent condition</td>
</tr>
<tr>
<td>ip_address_pattern</td>
<td>IP address pattern to match against trap agent</td>
</tr>
<tr>
<td>no</td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

**condition community string**

**no condition community string**

<table>
<thead>
<tr>
<th>condition</th>
<th>Adds a filtering condition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>community</td>
<td>Specify community string condition</td>
</tr>
<tr>
<td>string</td>
<td>String to match against trap community string</td>
</tr>
<tr>
<td>no</td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

**condition enterprise oid**

**no condition enterprise oid**

<table>
<thead>
<tr>
<th>condition</th>
<th>Adds a filtering condition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>enterprise</td>
<td>Specify enterprise condition</td>
</tr>
<tr>
<td>oid</td>
<td>oid to match against trap enterprise oid</td>
</tr>
<tr>
<td>no</td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

**condition filter filter_name**
### no condition filter filter_name

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>condition</strong></td>
<td>Adds a filtering condition.</td>
</tr>
<tr>
<td><strong>filter</strong></td>
<td>Specify trap filter condition</td>
</tr>
<tr>
<td><strong>filter_name</strong></td>
<td>Specify an already defined trap filter</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

### condition generic type

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>condition</strong></td>
<td>Adds a filtering condition.</td>
</tr>
<tr>
<td><strong>generic</strong></td>
<td>Specify generic trap condition</td>
</tr>
<tr>
<td><strong>type</strong></td>
<td>Generic trap type to match against trap type</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

### condition oid oid

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>condition</strong></td>
<td>Adds a filtering condition.</td>
</tr>
<tr>
<td><strong>oid</strong></td>
<td>Specify OID condition</td>
</tr>
<tr>
<td><strong>oid</strong></td>
<td>OID to match against trap OID</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

### condition specific type

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>condition</strong></td>
<td>Adds a filtering condition.</td>
</tr>
<tr>
<td><strong>specific</strong></td>
<td>Specify specific trap condition</td>
</tr>
<tr>
<td><strong>type</strong></td>
<td>Specific trap type to match against trap type</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

### condition trap trap_name

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>condition</strong></td>
<td>Adds a filtering condition.</td>
</tr>
<tr>
<td><strong>trap</strong></td>
<td>Specify trap condition</td>
</tr>
<tr>
<td><strong>trap_name</strong></td>
<td>Specific trap name to match against trap</td>
</tr>
<tr>
<td><strong>no</strong></td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>
condition variable-binding index value

no condition variable-binding index value

<table>
<thead>
<tr>
<th>condition</th>
<th>Adds a filtering condition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable-binding</td>
<td>Specify variable-binding condition</td>
</tr>
<tr>
<td>index</td>
<td>Specific variable binding index to use to match against trap. Starts with 1.</td>
</tr>
<tr>
<td>value</td>
<td>Value of variable binding to use to match against trap.</td>
</tr>
<tr>
<td>no</td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

condition version [ 1 | 2c | 3 ]

no condition version [ 1 | 2c | 3 ]

<table>
<thead>
<tr>
<th>condition</th>
<th>Adds a filtering condition.</th>
</tr>
</thead>
<tbody>
<tr>
<td>trap</td>
<td>Specify version condition</td>
</tr>
<tr>
<td>1</td>
<td>Specific SNMP v1 to match against trap</td>
</tr>
<tr>
<td>2c</td>
<td>Specific SNMP v2c to match against trap</td>
</tr>
<tr>
<td>3</td>
<td>Specific SNMP v3 to match against trap</td>
</tr>
<tr>
<td>no</td>
<td>Deletes this condition</td>
</tr>
</tbody>
</table>

Example

This example shows how to create a trap filter which will only allow traps with agent addresses matching 10.1.10.*.

```
zr# trap-filter agentfilter
zr(trap-filter)# condition agent 10.1.10.*
zr(trap-filter)# exit
```

vlan

To disable all VLAN trunks.

vlan disable

whitelist

To change the whitelist configuration on the ZoneRanger. To disable whitelisting, use the no form of this command.

whitelist

no whitelist
Usage Guidelines

Whitelist changes are made within the whitelist configuration submode. Once you are in the whitelist configuration submode, the following configuration commands are available:

- **add** – Add an IP address pattern
- **cancel** - Exit whitelist without saving any changes
- **clear** – Clear all whitelist entries
- **delete** – Delete an IP address pattern
- **enforce-outbound-requests** – Blocks all traffic with addresses outside of the whitelist
- **exit** – exit server group mode (saving changes)
- **list** – Lists the IP Address patterns
- **no** – Disable settings
- **test** – Apply the IP address patterns for 60 seconds

### add ip_address_pattern

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>add</strong></td>
<td>Adds a IP address pattern to the whitelist.</td>
</tr>
<tr>
<td><strong>ip_address_pattern</strong></td>
<td>IP address pattern to add</td>
</tr>
</tbody>
</table>

### delete ip_address_pattern

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>delete</strong></td>
<td>Removes an IP address pattern from the whitelist.</td>
</tr>
<tr>
<td><strong>ip_address_pattern</strong></td>
<td>IP address pattern to delete</td>
</tr>
</tbody>
</table>

### Example

This example shows how to add whitelist entry.

```
zr# whitelist
zr(whitelist settings)# add 10.1.10.*
zr(trap-filter)# exit
```
Chapter 38: Ranger Gateway Command Interface

The Ranger Gateway command interface provides an interface for Ranger Gateway configuration, as well as interaction with joined ZoneRangers.

Running the commands

The commands are installed in the following directories, depending on the platform:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>install_dir/bin</td>
</tr>
<tr>
<td>Solaris</td>
<td>install_dir/bin</td>
</tr>
<tr>
<td>Windows</td>
<td>install_dir\bin</td>
</tr>
</tbody>
</table>

where <install_dir> is the directory where the Ranger Gateway software was installed (by default: C:\Program Files\Tavve\Ranger Gateway).

Command Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>


<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addRoute</td>
<td>Temporarily adds an entry to the ZoneRanger routing table</td>
</tr>
<tr>
<td>auditResult</td>
<td>Displays the timestamp and results of the most recent audit</td>
</tr>
<tr>
<td>backup</td>
<td>Manages backups of a ZoneRanger configuration profile and database</td>
</tr>
<tr>
<td>changeRouteTestPeriod</td>
<td>Changes the length of time that addRoute temporarily adds an entry to the ZoneRanger routing table</td>
</tr>
<tr>
<td>commitRoute</td>
<td>Permanently commits a previously added route to the ZoneRanger routing table</td>
</tr>
<tr>
<td>configGateway</td>
<td>Displays, changes, or resets a variety of the Ranger Gateway configuration</td>
</tr>
<tr>
<td>configLicenses</td>
<td>Loads and displays the set of licenses for the Ranger Gateway to provide for ZoneRanger VMs</td>
</tr>
<tr>
<td>configSSL</td>
<td>Installs SSL certificates on the Ranger Gateway</td>
</tr>
<tr>
<td>configTacacServers</td>
<td>Lists, adds, and removes TACACS+ servers connected to the Ranger Gateway, and sets the timeout and log level.</td>
</tr>
<tr>
<td>configTraffic</td>
<td>Configures the Traffic thresholds on the Ranger Gateway</td>
</tr>
<tr>
<td>createSecurityKey</td>
<td>Creates security keys used to restrict access to Ranger Gateway commands</td>
</tr>
<tr>
<td>debugDaemons</td>
<td>Manages specific debugging options on a ZoneRanger.</td>
</tr>
<tr>
<td>debugFilter</td>
<td>Manages specific debugging filters on a ZoneRanger or Ranger Gateway</td>
</tr>
<tr>
<td>debugLevel</td>
<td>Manages the overall debugging level of the ZoneRanger or Ranger Gateway</td>
</tr>
<tr>
<td>debugString</td>
<td>Displays debugging information from particular ZoneRanger and Ranger Gateway services</td>
</tr>
<tr>
<td>deleteRoute</td>
<td>Removes an entry from the ZoneRanger routing table</td>
</tr>
<tr>
<td>deviceGroup</td>
<td>Manages named groups of devices to be used with ProxyMap and portMap commands</td>
</tr>
<tr>
<td>discovery</td>
<td>Used to initiate or check the status of a manual run of the discovery process.</td>
</tr>
<tr>
<td>downloadFile</td>
<td>Used to download log files, such as trapd.log and syslog.log, from a ZoneRanger</td>
</tr>
<tr>
<td>downloadTftpFile</td>
<td>Copies a file from a ZoneRanger TFTP directory to the Ranger Gateway download directory</td>
</tr>
<tr>
<td>echoTest</td>
<td>Verifies communication with a ZoneRanger</td>
</tr>
<tr>
<td>GatewayStart</td>
<td>Starts the Ranger Gateway software (Windows only)</td>
</tr>
<tr>
<td>GatewayStop</td>
<td>Stops the Ranger Gateway software (Windows only)</td>
</tr>
<tr>
<td>gateway.start.ksh</td>
<td>Starts the Ranger Gateway software (Linux, and Solaris only)</td>
</tr>
<tr>
<td>gateway.stop.ksh</td>
<td>Stops the Ranger Gateway software (Linux, and Solaris only)</td>
</tr>
<tr>
<td>gvi</td>
<td>Manages and tests the configuration of the gateway virtual interface which provides transparent communications to joined ZoneRangers</td>
</tr>
<tr>
<td>joinRequest</td>
<td>Attempts to join to a ZoneRanger</td>
</tr>
<tr>
<td>listJoined</td>
<td>Lists all ZoneRangers joined to the Ranger Gateway</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>listStatistics</td>
<td>Lists statistics collected from various services since the Ranger Gateway or ZoneRanger most recently started</td>
</tr>
<tr>
<td>listTcpPorts</td>
<td>Lists the Ranger Gateway ports that proxy to the HTTP, HTTPS, Telnet, SSH, and SQL services on each joined ZoneRanger</td>
</tr>
<tr>
<td>listTftpFiles</td>
<td>Lists the files in a ZoneRanger TFTP directory</td>
</tr>
<tr>
<td>localMacs</td>
<td>Lists the current set of Mac Addresses on this Ranger Gateway</td>
</tr>
<tr>
<td>mibToXml</td>
<td>Produces a ZoneRanger trap destination XML file from a trap definitions in the specified SNMP MIB file</td>
</tr>
<tr>
<td>patchinstall</td>
<td>Installs a patch on a Ranger Gateway (Linux and Solaris Only)</td>
</tr>
<tr>
<td>patchstatus</td>
<td>Lists the installed patches on a Ranger Gateway (Linux and Solaris Only)</td>
</tr>
<tr>
<td>patchuninstall</td>
<td>Uninstalls a patch on a Ranger Gateway (Linux and Solaris Only)</td>
</tr>
<tr>
<td>patchZR</td>
<td>Uploads, applies, or remove ZoneRanger patches</td>
</tr>
<tr>
<td>portConfig</td>
<td>Manages named sets of rules which describe allowable ports, protocols, and port translations used in Proxy Access Control</td>
</tr>
<tr>
<td>portControl</td>
<td>Enables and disables various ZoneRanger services</td>
</tr>
<tr>
<td>portMap</td>
<td>Manages sets of rules which determine the Port Config ruleset to use based on the source address of the requesting client and the destination address of the target device used in Proxy Access Control</td>
</tr>
<tr>
<td>profile</td>
<td>Manages profiles of a ZoneRanger configuration.</td>
</tr>
<tr>
<td>propertyGet</td>
<td>Retrieves the value of the specified property from the ZoneRanger or Ranger Gateway</td>
</tr>
<tr>
<td>propertyList</td>
<td>Lists all of properties the ZoneRanger or Ranger Gateway</td>
</tr>
<tr>
<td>propertySet</td>
<td>Sets the value of the specified property on the ZoneRanger or Ranger Gateway</td>
</tr>
<tr>
<td>propertyUnset</td>
<td>Clears the specified property on the ZoneRanger or Ranger Gateway</td>
</tr>
<tr>
<td>proxyMap</td>
<td>Manages the contents of the active proxy map as well as the configurations setting of the Proxy Map service.</td>
</tr>
<tr>
<td>RangerGateway</td>
<td>Starts the Ranger Gateway Viewer GUI</td>
</tr>
<tr>
<td>removeTftpFile</td>
<td>Removes a file from the ZoneRanger TFTP directory</td>
</tr>
<tr>
<td>rgBackup</td>
<td>Creates or restores a backup of the Ranger Gateway configuration</td>
</tr>
<tr>
<td>rgvi</td>
<td>Manages the configuration of the remote gateway virtual interface which provides transparent communications from Ranger Gateway to OpenVPN clients</td>
</tr>
<tr>
<td>servicedump</td>
<td>Generates a file containing diagnostic information about the Ranger Gateway or a ZoneRanger</td>
</tr>
<tr>
<td>setPasscode</td>
<td>Changes the passcode of the Ranger Gateway. The passcode is used to join to ZoneRangers</td>
</tr>
<tr>
<td>shutdownSystem</td>
<td>Restarts, reboots, and shuts down a ZoneRanger</td>
</tr>
<tr>
<td>snmpRequest</td>
<td>Performs SNMP requests, and listens for notifications</td>
</tr>
</tbody>
</table>
### sqlQuery
Queries SQL database tables

### trapdToXml
Converts an OpenView or NetView `trapd.conf` trap definition file to the ZoneRanger XML format

### trapFwdLogParser
Used to display in a human readable form, the trap forward log on the Ranger Gateway

### trapXmlValidator
Validates an XML trap definitions file

### troubleshootNetwork
Executes `ping`, `nslookup`, `traceroute` and `snmpget` commands on a ZoneRanger. `Ping`, `traceroute` and `snmpget` support IPv6 addresses.

### trustedSSL
Configures which ZoneRanger certificates the Ranger Gateway trusts

### tuntap
Installs, removes, or displays the status of the driver required to be installed to use the `gvi` on Windows

### unjoinAll
Unjoins all joined ZoneRanger

### unjoinRequest
Unjoins a ZoneRanger

### uploadConfig
Uploads an updated trap definition file

### uploadTftpFile
Uploads a file to the ZoneRanger TFTP directory

### viewIcmpLatency
Retrieves last ICMP latency time for ZoneRanger polled devices

### viewRoutes
Displays the routing table of the specified ZoneRanger

### viewRouteTestPeriod
Displays how long `addRoute` temporarily adds routes to the ZoneRanger routing table

### addRoute

```
addRoute zoneranger network_addr network_mask gateway_addr [metric]

zoneranger ipv6 network_addr gateway_addr [metric]
```

- `zoneranger` specifies the name of the ZoneRanger to add the route
- `ipv6` specifies that the route is an IPv6 route
- `network_addr` specifies the destination network IP address. An IPv6 address uses CIDR format
- `network_mask` specifies the network mask of the route for IPv4 routes
- `gateway_addr` specifies the IPv4 gateway
- `next-hop` specifies the IPv6 next hop address
- `metric` specifies the optional metric for this route

`addRoute` temporarily adds an entry to the specified ZoneRanger routing table, then removes it after 60 seconds. This enables route testing before making permanent routing table updates. To make a route permanent, use the `commitRoute` command before 60 seconds has elapsed.

### auditResult

```
auditResult [zoneranger]
```

- `zoneranger` specifies the name of the ZoneRanger to display most recent audit (optional)

`auditResult` displays the timestamp and results of the most recent audit.

If `zoneranger` is not given, the command displays the most recent Ranger Gateway audit.
backup

backup [ create zoneranger [ comment ]] |

[ listRG ] | [ deleteFromRG backup_name ] |

[ restoreFromRG zoneranger backup_name ] |

[ listZR zoneranger ] | [ deleteFromZR zoneranger backup_name ] |

[ restoreFromZR zoneranger backup_name ]

zoneranger specifies the name of the ZoneRanger from which to take the backup
comment specifies an optional comment to be stored with the backup.

backup_name specifies the backup to use

create creates a backup of the ZoneRanger configuration profile and database in the Ranger
Gateway store/zr/backup directory

listRG lists the ZoneRanger backups stored on this Ranger Gateway

deleteFromRG is used to delete a ZoneRanger backup stored on this Ranger Gateway

restoreFromRG is used to restore a backup to a ZoneRanger from a backup stored on this
Ranger Gateway

listZR list the backups stored on a ZoneRanger

deleteFromZR is used to delete a backup stored on the ZoneRanger

restoreFromZR is used to restore a backup from a backup stored on the ZoneRanger

backup is used to manage the creation, restoration and storage of ZoneRanger configuration and
database backups.

changeRouteTestPeriod

changeRouteTestPeriod zoneranger test_period

zoneranger specifies the name of the ZoneRanger on which to change the value
test_period specifies the test period in seconds

changeRouteTestPeriod changes the length of time that addRoute temporarily adds an entry to
the ZoneRanger routing table.

commitRoute

commitRoute zoneranger network_addr network_mask gateway_addr

zoneranger ipv6 network_addr gateway_addr [metric]

zoneranger specifies the name of the ZoneRanger to add the route
ipv6 specifies that the route is an IPv6 route
network_addr specifies the destination network IP address. An IPv6 address uses CIDR format.

network_mask specifies the network mask of the route for IPv4 routes.

gateway_addr specifies the IPv4 gateway.

next-hop specifies the IPv6 next hop address.

commitRoute permanently adds an entry to the ZoneRanger routing table. The route must first be added using the addRoute command.

configGateway

configGateway -view [config_item] | -change config_item new_value | -default config_item

cfg Item is one of:

<table>
<thead>
<tr>
<th>config_item</th>
<th>description</th>
</tr>
</thead>
</table>

ZoneRanger 5.6 User’s Guide
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gateway_status_trap_dest_address</td>
<td>Destination hostname or IP address to which Ranger Gateway status traps are sent</td>
</tr>
<tr>
<td>generic_forward_log</td>
<td>Level of logging for generic UDP forwarding - values: none, short, full</td>
</tr>
<tr>
<td>http_forward_port</td>
<td>Ranger Gateway HTTP port which can be used to access the web interface of joined ZoneRangers. The default is 4853.</td>
</tr>
<tr>
<td>http_forward_port_enabled</td>
<td>Whether or not the http_forward_port should be available on the Ranger Gateway.</td>
</tr>
<tr>
<td>icmp_proxy_log</td>
<td>Level of logging for ICMP proxy - values: none, short, full</td>
</tr>
<tr>
<td>icmp_proxy_timeout</td>
<td>Number of seconds to wait for a response from the ZoneRanger for each ICMP request</td>
</tr>
<tr>
<td>icmp_proxy_ttl</td>
<td>Enable ICMP proxy to include handling the TTL field in the received packet. Must be enabled to use network utilities such as ICMP based traceroute.</td>
</tr>
<tr>
<td>mail_server</td>
<td>Hostname or IP address of the mail server the Ranger Gateway should use to send root cause emails</td>
</tr>
<tr>
<td>messaging_port</td>
<td>Port used for communications between ZoneRangers and Ranger Gateways. All joined ZoneRangers and Ranger Gateways and redundant ZoneRangers must use the same port.</td>
</tr>
<tr>
<td>netflow_forward_log</td>
<td>Level of logging for NetFlow forwarding - values: none, short, full</td>
</tr>
<tr>
<td>ntp_proxy_log</td>
<td>Level of logging for NTP Proxy - values: none, short, full</td>
</tr>
<tr>
<td>ntp_proxy_spoof</td>
<td>Whether or not the source address in the NTP client requests is the source address of the original sending device managed by the ZoneRanger or the address of the Ranger Gateway.</td>
</tr>
<tr>
<td>radius_proxy_log</td>
<td>Level of logging for RADIUS proxy - values: none, short, full</td>
</tr>
<tr>
<td>radius_proxy_spoof</td>
<td>Whether or not the source address in the RADIUS client requests is the source address of the original sending device managed by the ZoneRanger or the address of the Ranger Gateway.</td>
</tr>
<tr>
<td>sflow_forward_log</td>
<td>Level of logging for sFlow forwarding - values: none, short, full</td>
</tr>
<tr>
<td>Configuration Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>snmp_proxy_client_address</td>
<td>Address pattern or device group which lists the set of client addresses from which to accept SNMP proxy requests using the Community String Convention method.</td>
</tr>
<tr>
<td>snmp_proxy_log</td>
<td>Level of logging for generic UDP forwarding - values: none, short, full</td>
</tr>
<tr>
<td>snmp_proxy_mode</td>
<td>Format that applications use to specify the SNMP community string when making SNMP proxy requests via the SNMP Proxy port.</td>
</tr>
<tr>
<td>snmp_proxy_port</td>
<td>Port on which the Ranger Gateway listens for SNMP proxy requests. The default is 4852.</td>
</tr>
<tr>
<td>snmp_proxy_port_enabled</td>
<td>Whether or not the snmp_proxy_port should be available on the Ranger Gateway.</td>
</tr>
<tr>
<td>snmp_proxy_timeout</td>
<td>Amount of time, in seconds, the Ranger Gateway waits for a response to an SNMP proxy request.</td>
</tr>
<tr>
<td>socks_server_port</td>
<td>Port on which Ranger Gateway listens for SOCKS requests.</td>
</tr>
<tr>
<td>socks_server_port_enabled</td>
<td>Whether or not the Ranger Gateway will listen for SOCKS server requests.</td>
</tr>
<tr>
<td>ssh_proxy_dest_port</td>
<td>Port on which ZoneRanger managed devices to which an SSH proxy session should be established when using ssh_proxy_port.</td>
</tr>
<tr>
<td>ssh_proxy_port</td>
<td>Port on which Ranger Gateway listens for SSH Proxy requests. The default is 4822.</td>
</tr>
<tr>
<td>ssh_proxy_port_enabled</td>
<td>Whether or not the Ranger Gateway will listen for SSH proxy requests.</td>
</tr>
<tr>
<td>syslog_forward_log</td>
<td>Level of logging for syslog forwarding - values: none, short, full.</td>
</tr>
<tr>
<td>tacacs_proxy_log</td>
<td>Level of logging for TACACS+ proxy - values: none, short, full.</td>
</tr>
<tr>
<td>tacacs_proxy_spoof</td>
<td>Whether or not the source address in the TACACS+ client requests is the source address of the original sending device managed by the ZoneRanger or the address of the Ranger Gateway.</td>
</tr>
<tr>
<td>tcp_proxy_log</td>
<td>Level of logging for TCP proxy - values: none, short, full.</td>
</tr>
<tr>
<td>tcp_inbound_proxy_log</td>
<td>Level of inbound logging for TCP proxy - values: none, short, full.</td>
</tr>
<tr>
<td>Configuration Item</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>tcp_inbound_proxy_spoof</td>
<td>Whether or not the inbound source address in the TCP client requests is the source address of the original sending device managed by the ZoneRanger or the address of the Ranger Gateway.</td>
</tr>
<tr>
<td>tftp_proxy_log</td>
<td>Level of logging for TFTP proxy - values: none, short, full</td>
</tr>
<tr>
<td>tftp_proxy_read_dir</td>
<td>Directory where TFTP files should be read when proxying files to ZoneRanger managed devices</td>
</tr>
<tr>
<td>tftp_proxy_write_dir</td>
<td>Directory where TFTP files should be written when proxying files from ZoneRanger managed devices</td>
</tr>
<tr>
<td>trap_forward_log</td>
<td>Level of logging for trap forwarding - values: none, short, full</td>
</tr>
<tr>
<td>udp_forward_spoof</td>
<td>Whether or not the source address in the UDP traffic is the source address of the original sending device managed by the ZoneRanger or the address of the Ranger Gateway.</td>
</tr>
</tbody>
</table>

- **view** displays the current value of the specified configuration item. If there is no current value specified, all configuration values are displayed.
- **change** changes the current value of the specified configuration item to new_value.
  
  **config_item** specifies the configuration item to change
  
  **new_value** specifies the new value to use for this configuration item.
- **default** resets the specified configuration item back to its original value.

**configGateway** displays, changes, or resets the Ranger Gateway configuration.

**configLicenses**

**configLicenses [ list ] | [ load <filename> ] | [ export <filename> ]**

- **list** displays the list of licenses loaded on this Ranger Gateway
- **load** can be used to load a new set of licenses on the Ranger Gateway
  
  **filename** specifies the path and filename of licenses to load.
- **export** can be used to export the current set of licenses on the Ranger Gateway
  
  **filename** specifies the path and filename in which to store the current set of licenses.
- **listMacAddresses** can be used to list the mac addresses that tie a set of licenses to the particular Ranger Gateway.

**configLicenses** can be used to list, load, or export the set of ZoneRanger VM licenses on this Ranger Gateway. When loading a new set of licenses from the specified file, all of the current licenses will be removed prior to loading the new licenses.
configSSL


cfgSSL can be used to install SSL certificates used for communication between the Ranger Gateway and joined ZoneRangers. If configSSL is executed with no parameters, the following options list is displayed:

1. Display current certificate
2. Install by PKCS #12
3. Install by key and SSL certificate
4. Install by keystore
5. Revert to original SSL certificate
6. Display usage

Option 1: Display current certificate

configSSL [ -show ]

List the details for the current certificate.

Option 2: Install by PKCS #12

configSSL [ -pkcs12 pkcs_file [ -password password ] ]

pkcs_file specifies the file containing an SSL certificate in PKCS #12 format.

Password specifies the password needed to access the SSL certificate.

This option is used to load a new SSL certificate on the Ranger Gateway using a PKCS #12 file as the source of the security information.

Option 3: Install by key and SSL Certificate

configSSL [ -certificate cert_file [ -pem pem_file ] [ -pemPassword password ] ]

cert_file specifies the file containing an signed SSL certificate.

pem_file specifies the file containing the private key in X.509 format.

password specifies the password needed to access the private key.

This option is used to load a new SSL certificate on the Ranger Gateway using a X.509 file as the source of the security information.

Option 4: Install by keystore

configSSL [ -keystore key_file [ -keystorePassword kp_password ] [ -keyEntryPassword ke_password ] ]

key_file specifies the file in keystore format containing the SSL keys and certificates
kp_password specifies the password to access the keystore file.

ke_password specifies the password needed to access the private key.

This option is used to load a new SSL certificate on the Ranger Gateway using a Java Keystore as the source of the security information.

Option 5: Revert to original SSL certificate

This option reverts the presently used SSL certificate back to the Tavve original SSL certificate.

After a certificate is installed on the Ranger Gateway, you must use the ZoneRanger web interface to configure joined ZoneRangers to accept connections using the new certificate. If not already present, the Trusted Subject which is associated with the new SSL Certificate must be added on the Configuration > Ranger Gateway page SSL Trust tab on the ZoneRanger.

configTacacsServers

cfgTacacsServers [ -list ] | [ -remove tacacsServer [ port ] ] | [ -spoof on|off ] | [ -log none | short | full]

- list displays the list of TACACS+ servers needed prior to Ranger Gateway 5.0 - Deprecated
- remove can be used to remove the specified TACACS+ servers from the list - Deprecated
tacacsServer specifies the name of the TACACS+ server to remove
port specifies the port the TACACS+ server is using.

- spoof can be used to indicate whether or not the source address in the TACACS+ client requests is the source address of the original sending device managed by the ZoneRanger or the address of the Ranger Gateway.
  • on: spoofing is enabled
  • off: spoofing is disabled

- log specifies the level of logging for TACACS+ proxy requests
  • none: Logging is turned off.
  • short: Only message headers are logged.
  • full: Entire messages are logged.

cfgTacacsServers can be used to modify the TACACS+ configuration settings on the Ranger Gateway. Prior to Ranger Gateway 5.0, in order to configure TACACS+ proxy support, TACACS+ servers had to be listed in the Ranger Gateway configuration. Beginning with Ranger Gateway 5.0, that is no longer the case. Any TACACS+ servers already configured may be displayed or removed by the cfgTacacsServers command.

configTraffic

cfgTraffic subcommand [arguments]

configTraffic configures traffic thresholds, enables and disables notification, and sets the traffic log level.

subcommand can be one of the following:
  • view
• **set [per_zr] forwarded [total|generic|netflow|sflow|syslog|trap] value**
• **set [per_zr] proxied [icmp|ntp|radius|snmp|tacacs] value**
• **check_thresholds [per_zr] [forwarded|proxied] [on|off]**
• **notify [per_zr] [forwarded|proxied] [on|off]**
• **interval value**
• **log [none|short|full]**

**view** displays the current traffic configuration settings

**set..forwarded** sets the thresholds for forwarded traffic
  - **per_zr** specifies the threshold is on a per IP address basis.
  - **total** specifies the threshold is for total traffic
  - **generic** specifies the threshold is for generic UDP only.
  - **netflow** specifies the threshold is for netFlow packets only.
  - **sflow** specifies the threshold is for sFlow packets only.
  - **syslog** specifies the threshold is for syslog messages only.
  - **trap** specifies the threshold is for SNMP traps only.
  - **value** specifies the threshold value in seconds.

**set..proxied** sets the thresholds for proxied traffic
  - **per_zr** specifies the threshold is on a per IP address basis.
  - **icmp** specifies the threshold is for ICMP proxy only
  - **ntp** specifies the threshold is for NTP proxy only.
  - **radius** specifies the threshold is for RADIUS proxy only.
  - **snmp** specifies the threshold is for SNMP proxy only.
  - **tacacs** specifies the threshold is for TACACS+ proxy only.
  - **value** specifies the threshold value in seconds.

**check_thresholds** enables and disables threshold checking
  - **per_zr** specifies the threshold check is on a per IP address basis.
  - **forwarded** specifies the threshold check is for forwarded traffic.
  - **proxied** specifies the threshold check is for proxied traffic.
  - **on** enables threshold checking.
  - **off** disables threshold checking.

**notify** enables and disables notification when a threshold is exceeded.
  - **per_zr** specifies the threshold notification is on a per IP address basis.
  - **forwarded** specifies the threshold notification is for forwarded traffic.
  - **proxied** specifies the threshold notification is for proxied traffic.
  - **on** enables threshold notification.
  - **off** disables threshold notification.
interval specifies the measurement interval for threshold checking.  

value specifies the measurement interval in seconds. Minimum value is 60.

log specifies the level of logging for traffic monitoring. 

- none specifies no logging will occur. 
- short specifies logging of Overall traffic counts 
- full specifies logging of Overall and Per ZoneRanger traffic counts

createSecurityKey

createSecurityKey security-admin|admin|operator [-p] [-d dir] [-t]

- p prompts for a passphrase when creating the security key
- d specifies destination directory to write the security key.
  dir Fully qualified directory
- t tests for support of MAC/HMAC algorithms on this system.

createSecurityKey allows for the creation of security keys which may be applied to Ranger Gateway commands or directories to control the access to Ranger Gateway commands. There are three levels of security; security-admin, admin and operator, which may be specified. If a destination directory is not specified, the security key is written to the install_dir/gateway/security directory.

debugDaemons

debugDaemons [zoneranger] -g debug-level -s proxy_name

- zoneranger specifies the name of the ZoneRanger
- g sets the debug level. 0 turns off debugging.
- s specifies the service name.

debugDaemon is used, as directed by Tavve Support personnel, to record debugging information for a particular service or services on the ZoneRanger. Debugging filters can negatively effect the performance of the specified service so debugging filters should be cleared after they are no longer needed by Tavve Support personnel.

debugFilter

debugFilter [zoneranger] [-list | -add name level(1-15) | -remove name | -clear ]

- zoneranger specifies the name of the ZoneRanger
- list displays the list of currently configured debugging filters.
- add specifies the service name and debug level for that service to add.
- remove specifies the service name to remove from debugging.
- clear removes all debugging filters.

debugFilter is used, as directed by Tavve Support personnel, to record debugging information for a particular service or services on the ZoneRanger and Ranger Gateway. Debugging filters can negatively effect the performance of the specified service so debugging filters should be cleared after they are no longer needed by Tavve Support personnel. If a zoneranger is not specified, the debugging filter will be applied to the Ranger Gateway.
**debugLevel**

`debugFilter [ [zoneranger] [ -set level(1-15)] ] | -jni [ -set level(1-15)]`

- `zoneranger` specifies the name of the ZoneRanger
- `-set` sets the overall debug level (default is 4).
- `-jni` sets the debug level for the jni on the Ranger Gateway (default is 0).

`debugLevel` is used, as directed by Tavve support personnel, to set the overall debugging level for the entire ZoneRanger or Ranger Gateway. This should only be used under the direction of Tavve Support personnel. The use of non-default debugging levels will cause performance degradation for the system. When `debugLevel` is used with no parameters, it displays the current debug level of the Ranger Gateway. When `debugLevel` is used with only the `zoneranger` parameter, it displays the current debug level of the specified ZoneRanger.

**debugString**

`debugString [ -t timeout ] [ -x ] [ -v ] -list | -s service | -all`

- `-t` sets the timeout in seconds for the command to complete.
- `-x` outputs information in XML format. Not valid with `-list`
- `-v` outputs verbose information
- `-list` outputs the available service names
- `-s` outputs information about the specified service
- `-all` outputs information about all available services

`debugString` is used, as directed by Tavve support personnel, to display information about services on the Ranger Gateway. This should only be used under the direction of Tavve Support personnel.

**deleteRoute**

`deleteRoute zoneranger network_addr network_mask gateway_addr`

`zoneranger ipv6 network_addr next-hop`

- `zoneranger` specifies the name of the ZoneRanger to add the route
- `ipv6` specifies that the route is an IPv6 route
- `network_addr` specifies the destination network IP address. An IPv6 address uses CIDR format
- `network_mask` specifies the network mask of the route for IPv4 routes
- `gateway_addr` specifies the IPv4 gateway
- `next-hop` specifies the IPv6 next hop address

`deleteRoute` removes an entry from the ZoneRanger routing table.

**deviceGroup**

`deviceGroup subcommand [arguments]`
deviceGroup creates named groups of devices which can be used in the creation of proxyMap
and portMap rules. deviceGroup command is organized as a set of subcommands, each of which
supports different parameters and options. Most deviceGroup subcommands provide the option to
operate directly on the active device group table of a running Ranger Gateway in real time, or to
work offline with text files, which can be inspected and edited using a text editor, then installed
on the Ranger Gateway when required modifications have been completed.

As a convenience, a device group called ZoneRanger is available which includes any IP addresses
that map to a joined ZoneRanger based on the proxyMap configuration.

subcommand can be one of the following:

- **copy** [-in input_file] [-out output_file]
- **add** [-in input_file] [-out output_file] group-name address [address...]
- **remove** [-in input_file] [-out output_file] group-name address [address...]
- **merge** [-in input_file] [-out output_file] merge_file
- **list** [-in input_file] [group-name] [address]
- **clear** [-f]
- **config** [item [value]]
- **test** [address]

**deviceGroup copy** [-in input_file] [-out output_file]

- **in** indicates the name of the input file containing device group information
- **out** indicates the name of the output file to write device group information

**deviceGroup copy** can be used for the following:

- To copy the content of the active device group table to a specified text file.
- To copy the content of a specified text file to the active device group table.
- To copy the content of one specified text file to another.

If no input file is specified, the active device group table is used as the source of the copy. If
no output file is specified, the input configuration is automatically copied to the active device
group table. If an output file is specified, the input configuration is written to the specified file
and the active device group table is unchanged.

Note that the copy subcommand always outputs XML. The input format can be XML, or a
simple text format.

**deviceGroup add** [-in input_file] [-out output_file] group-name

- **in** indicates the name of the input file containing device group information
- **out** indicates the name of the output file to write device group information
- **group-name** indicates the name of the group being created or modified.
- **address** specifies the members to be added to the group which may be an individual
address, an address patterns range, another device group.

**deviceGroup add** can be used for the following:

- To create a new device group, with one or more members
• To add one or more members to an existing device group.

If the group-name parameter indicates a device group that already exists, the specified addresses will be added to the existing group. Otherwise a new device group will be created containing the specified member addresses.

deviceGroup add can read input from the active device group table, or from a specified text file. If no input file is specified, the active device group table is used. If no output file is specified, the resulting configuration is automatically copied to the active device group table. If an output file is specified, the resulting configuration is written to the specified file and the active device group table is unchanged.

deviceGroup remove [-in input_file] [-out output_file] group-name

    address [address...]

-in indicates the name of the input file containing device group information
-out indicates the name of the output file to write device group information

    group-name indicates the name of the group to be removed or modified.

    address indicates the members to be removed from the group. If no address values are specified, the entire group will be removed.

deviceGroup remove can be used to remove members from a device group, or to remove an entire device group from the active device group table, or from an offline file.

If the last member is removed from a group, the group will automatically be removed as well. If the specified group does not exist, or does not contain the specified members, the input configuration will be unchanged.

deviceGroup remove can read input from the active device group table, or from a specified text file. If no input file is specified, the active device group table is used. If no output file is specified, the resulting configuration is automatically copied to the active device group table. If the output file is specified, the resulting configuration is written to the specified file and the active device group table is unchanged.

deviceGroup merge [-in input_file] [-out output_file] merge_file

    -in indicates the name of the input file containing device group information

    -out indicates the name of the output file to write device group information

    merge-file specifies the name of a file that contains a set of entries to be merged to the input configuration.

deviceGroup merge can be used for the following:

• To add one or more new device groups.

• To add members to one or more existing device groups.

The logic for merging is similar to the logic for the add subcommand. For each entry in the merge file, if the group-name parameter indicates a device group that already exists, the specified addresses will be added to the existing group. Otherwise a new device group will be created containing the specified member addresses.
deviceGroup can read input from the active device group table, or from a specified text file. If no input file is specified, the active device group table is used. If no output file is specified, the resulting configuration is automatically copied to the active device group table. If the output file is specified, the resulting configuration is written to the specified file and the active device group table is unchanged.

**deviceGroup list [-in input_file] [group-name [address]]**

- **-in** indicates the name of the input file containing device group information
- **group-name** indicates the name of the group to be listed.
- **address** indicates the members to be listed from the group.

deviceGroup list can read input from the active device group table, or from a specified text file. If no input file is specified, the active device group table is used.

deviceGroup clear [-f]

deviceGroup clear can be used to remove all device groups from the active device group table. When deviceGroup clear is executed, the user is prompted to confirm that the active device group table should be cleared. If the response is "y" or "yes" (case is ignored), the active device group table will be cleared. Otherwise the active device group table will be unchanged. If the -f option is specified, the user is not prompted.

deviceGroup config [item [value]]

deviceGroup config can be used to display or modify configuration items associated with the device group table.
The configuration items associated with the device group table are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>device_group_cache_size</td>
<td>Determines the maximum number of entries in the cache. Valid values are positive integers in the range 0-10000. The default value is 100.</td>
</tr>
</tbody>
</table>

If no item or value is specified, the values of all configuration items are listed. If an item is specified with no value, the current value of the specified configuration item is displayed. If an item and a value are specified, the value of the specific configuration item is set to the specified value.

**deviceGroup test [address]**

`deviceGroup test` performs a query on the Ranger Gateway to list all device groups from the active device group table that contain a specified address.

The primary difference between the list subcommand and the test subcommand is that the test subcommand also checks to see if the specified address matches any of the automatically defined groups (i.e. Local, ZoneRanger). In addition, the list subcommand can list entries from offline files, but the test subcommand only lists entries from the Ranger Gateway’s active device group table.

**deviceGroup File Formats**

The various `deviceGroup` subcommands that generate configurations (i.e. copy, add, remove, merge) all generate configuration information in an XML format. An example of this format, corresponding to the default Ranger Gateway configuration is as follows:

```xml
<device-group-list>
  <device-group name="group1">
    <address value="10.1.2.*"/>
    <address value="10.10.2.*"/>
  </device-group>
  <device-group name="group2">
    <address value="10.2.2.1"/>
    <address value="10.2.2.2"/>
    <address value="10.2.2.5"/>
  </device-group>
</device-group-list>
```

The `deviceGroup` commands that read configurations (i.e. copy, add, remove, merge, list) are able to read configuration input in the XML format, and also in a simplified text format. An example of this format, corresponding to the XML example above, is as follows:

```
group1 10.1.2.*
group1 10.10.2.*
group2 10.2.2.1
```
group2 10.2.2.2

group2 10.2.2.5

discovery

discovery zoneranger [-start | -status]

zoneranger specifies the name of the ZoneRanger to begin discovery

-start starts the discovery service on the specified ZoneRanger

-status displays the status of the discovery service on the specified ZoneRanger

discovery starts the discovery service on the specified ZoneRanger, or gives the status of a currently running discovery service.

downloadFile

downloadFile zoneranger [-list | filename]

zoneranger specifies the name of the ZoneRanger

-list displays the list of possible files which may be retrieved from the specified ZoneRanger.

filename specifies the name of the file as it appears in the -list option.

downloadFile can be used to download log files from the ZoneRanger containing the following information:

- SNMP traps
- Syslog messages
- Discovery results
- Patch application results
- General system operation

downloadTftpFile

downloadTftpFile zoneranger filename

zoneranger specifies the name of the ZoneRanger from which to retrieve file

filename specifies the name of the file to retrieve.

downloadTftpFile copies a file from the ZoneRanger TFTP directory to the Ranger Gateway download directory.

echoTest

echoTest [zoneranger]

zoneranger specifies the name of the ZoneRanger

echoTest verifies communication with a ZoneRanger. If executed without any parameters, echoTest verifies communication with the Ranger Gateway software. Use Ctrl-C to stop running echoTest.
**GatewayStart**

*GatewayStart*

GatewayStart starts the Ranger Gateway software. (Windows only)

**GatewayStop**

*GatewayStop*

GatewayStop stops the Ranger Gateway software. (Windows only)

**gateway.start.ksh**

`gateway.start.ksh`

starts the Ranger Gateway software. This command ignores any arguments. (Linux, and Solaris only)

**gateway.stop.ksh**

`gateway.stop.ksh`

stops the Ranger Gateway software. This command ignores any arguments. (Linux, and Solaris only)

**gvi**

`gvi subcommand [arguments]`

gvi manages the routes for the gateway virtual interface to provide transparent communications to joined ZoneRangers. gvi command is organized as a set of subcommands, each of which supports different parameters and options.

subcommand can be one of the following:

- enable
- disable
- status
- **generate-routes -subnet|-address -database|-proxyMap**
- **add-route <subnet> [<subnet>...]**
- **remove-route <subnet> [<subnet>...]**
- **merge-routes <input-file>**
- **list-routes [-all]**
- **clear-routes [-f]**
- **config [item[=value]]**
- **test [address]**

`gvi enable`
By default, the GVI service is disabled. `gvi enable` subcommand enables the GVI service. When the `gvi enable` subcommand is executed, the Ranger Gateway will create the virtual interface, add any required ZoneRanger host routes and any configured virtual interface routes to the system routing table, and will begin handling any management traffic that is received on the virtual interface.

**Note:** On Windows systems, the `tuntap` command must be executed before the GVI service is enabled after Ranger Gateway installation. Also, once the GVI is enabled, run the Windows command `ncpa.cpl` to display the current list of network interfaces. Using the **Advanced** > **Advanced Settings...** menu item, verify that the GVI virtual interface is last in the access order for network services.

**gvi disable**

`gvi disable` subcommand disables the GVI service. When the `gvi disable` subcommand is executed, the Ranger Gateway will stop handling management traffic received on the virtual interface, will delete the virtual interface routes and ZoneRanger host routes, and will remove the virtual interface.

**gvi status**

`gvi status` subcommand displays the current status of the GVI service. The `gvi status` subcommand indicates whether the GVI service is currently enabled or disabled and displays any errors or warnings that were generated during the most recent route manager operation.

**gvi generate-routes**

- `subnet` indicates to list subnets as potential additions as GVI routes
- `address` indicates to list individual IP addresses or IPv6 addresses in CIDR format as potential additions as GVI routes
- `database` indicates to use the databases of all joined ZoneRangers as the source of potential additions as GVI routes
- `proxyMap` indicates to use the Ranger Gateway Proxy Map service as the source of potential additions as GVI routes

`gvi generate-routes` subcommand incorporates other Ranger Gateway configuration information, or ZoneRanger database information, to identify subnets or individual IP addresses or IPv6 addresses in CIDR format that may be considered as candidates for the GVI route list.

The intent of the `gvi generate-routes` subcommand is to facilitate the process of identifying the subnets or IP addresses for which virtual interface routes may need to be created. The options for the `gvi generate-routes` command are used to specify the type of information that will be listed (i.e. subnets or individual IP addresses), and the source of the information (i.e. the databases of all joined ZoneRangers, of the Ranger Gateway's proxy map configuration).

Note that if NAT is in effect between the Ranger Gateway and the ZoneRanger, querying the databases of joined ZoneRangers will not produce useful results, because the listed subnets or addresses will reflect the ZoneRangers' perspective, as opposed to the Ranger Gateway's perspective. The output of the `gvi generate-routes` command can be redirected to a file, so that the resulting routes can be merged with the GVI route list using the `gvi merge-routes` subcommand.
It is highly recommended that the output of the `gvi generate-routes` subcommand be manually inspected and verified before the resulting routes are merged with the GVI route list. This is especially true if the `-database` option is used because the ZoneRanger discovery process may have discovered addresses and subnets that are beyond the scope of the network being managed, and the creation of virtual interface routes for such addresses would interfere with the management application’s ability to communicate with devices using those addresses.

It should also be noted that the `-subnet` option should, in general, be preferred over the `-address` option, because the resulting list will typically be much smaller, resulting in a corresponding decrease in the number of virtual interface routes that will need to be created.

```
gvi add-route <subnet> [<subnet>...]
```

*subnet* indicates the subnet or individual IP address or an IPv6 address in CIDR format to add to the GVI route list.

gvi add-route subcommand adds one or more subnets or individual IP addresses or IPv6 addresses in CIDR format to GVI route list.

The route manager within the GVI service maintains a persistent list of subnets and individual IP addresses that correspond to DMZ devices, and therefore, should be routed to the virtual interface. The `gvi add-route` subcommand can be used to add one or more subnets or individual IP addresses or an IPv6 address in CIDR format to this list. If the GVI service is enabled, the route manager will create a corresponding static route for each subnet or individual IP address in the list.

Each parameter after the add-route subcommand name can either be a specific IP address or an IPv6 address in CIDR format, or a subnet description. Any of the following formats can be used to describe a subnet:

- 10.1.10.*
- 10.1.10.[0-255]
- 10.1.10.0/255.255.255.0
- 10.1.10.0/24
- 2001:db8::/48

*Note:* IPv6 must be enabled in for GVI in order for the IPv6 routes to be asserted.

```
gvi remove-route <subnet> [<subnet>...]
```

*subnet* indicates the subnet or individual IP address or an IPv6 address in CIDR format to remove from the GVI route list.

gvi remove-route subcommand removes one or more subnets or individual IP addresses or IPv6 addresses in CIDR format from the GVI route list.

If the GVI service is enabled, the route manager will delete the corresponding static route for each subnet or individual IP address or an IPv6 address in CIDR format that has been removed from the list.

```
gvi merge-routes <input-file>
```

*input-file* specifies the file containing the list of subnets or IP addresses or IPv6 addresses in CIDR format to add to the GVI route list.

gvi merge-routes subcommand adds virtual interface routes for subnets and individual IP addresses or IPv6 addresses in CIDR format listed in the specified input file.
The required format for subnet or IP address or an IPv6 address in CIDR format entries in the input file is the same as for the `gvi add-route` command with one entry per line. Note, if there is an error adding a GVI route from a specific line of input file, processing will continue on subsequent lines.

**gvi list-routes [-all]**

- **all** lists all routes which include any virtual interface routes or ZoneRanger host routes in the system routing table that appear to have been created manually.

`gvi list-routes` command lists all IP addresses or IPv6 addresses in CIDR format and subnets in the GVI route list.

If the GVI service is enabled, the listed routes will also include any ZoneRanger host routes that have been created by the route manager. If the `-all` option is specified, the listed routes will also include any virtual interface routes or ZoneRanger host routes in the system routing table that appear to have been created manually. Each such address will be prefixed by a + (plus) character in the resulting list, in order to distinguish these routes from the routes that have been created by the GVI route manager.

**gvi clear-routes [-f]**

`gvi clear-routes` command removes all IP addresses and subnets in the GVI route list. If the `-f` option is specified, the user is not prompted for confirmation.

**gvi config [item [value]]**

`gvi config` can be used to display or modify configuration items associated with the GVI service.

The configuration items associated with the GVI service are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>log_level</td>
<td>Determines the level of logging for the GVI service - values: none, short, full. Default is none.</td>
</tr>
<tr>
<td>mtu_size</td>
<td>Determines the MTU size for outgoing packets.</td>
</tr>
<tr>
<td>ipv6_enable</td>
<td>Determines if IPv6 is enabled. The default is false.</td>
</tr>
</tbody>
</table>

If no `item` or `value` is specified, the values of all configuration items are listed. If an `item` is specified with no `value`, the current value of the specified configuration item is displayed. If an `item` and a `value` are specified, the value of the specific configuration item is set to the specified value.

**gvi test <address>**

`address` indicates the individual IP address or an IPv6 address in CIDR format to verify.

`gvi test` can be used to verify that the GVI and Proxy Map services have been configured properly so that traffic send to a specified address will be intercepted, routed to an appropriate ZoneRanger, and forwarded to the intended device. The `gvi test` subcommand also verifies whether or not the specified address is managed by a ZoneRanger where applicable.

**joinRequest**

**joinRequest address [passcode]**

`address` indicates the hostname or IP address of the ZoneRanger to contact.
passcode indicates the security passcode to use with the specified ZoneRanger joinRequest attempts to join to a ZoneRanger.

The passcode must match the ZoneRanger passcode exactly. If passcode is not given, the Ranger Gateway passcode is used.

listJoined

listJoined [-p] [zonerranger]

-p include the passcodes for the joined ZoneRanger or Ranger Gateways

<zonerranger> specifies the name of the ZoneRanger from which to list Ranger Gateways (optional)

listJoined lists all ZoneRangers joined to the Ranger Gateway

listStatistics

listStatistics [zonerranger] [-reset] [-queue]

zonerranger specifies the name of the ZoneRanger from which to list statistics (optional)

-reset resets all of the statistics to zero.

-queue displays the queue's statistics.

listStatistics manages statistics collected from various services since the Ranger Gateway or ZoneRanger most recently started.

If zonerranger is not specified, the listStatistics command operates on the Ranger Gateway statistics.

listTcpPorts

listTcpPorts [zonerranger]

<zonerranger> specifies the name of the ZoneRanger from which to list TCP ports (optional)

listTcpPorts lists the Ranger Gateway ports that proxy to the HTTP, HTTPS, SQL, SSH, and Telnet services on each joined ZoneRanger.

When a Ranger Gateway and ZoneRanger are joined, a set of TCP ports on the Ranger Gateway are allocated to communicate with the HTTP, HTTPS, SQL, SSH, and Telnet services on the ZoneRanger. Thus, applications which understand those protocols, may communicate to the assigned port on the Ranger Gateway and that communications will be proxied directed to the same service on the ZoneRanger. If however, the intended service has been disabled on the ZoneRanger, the communication will fail.

The listTcpPorts command displays the specific TCP ports assigned for each join ZoneRanger. Those ports would be used with a particular application.

If zonerranger is not specified, the listTcpPorts command displays the TCP ports for all joined ZoneRangers.

listTftpFiles

listTftpFiles zonerranger [-detail]
zoneranger specifies the name of the ZoneRanger from which to list TFTP files
- detail displays the size and date of each TFTP file.

listTftpFiles lists the files in the ZoneRanger TFTP directory.

localMacs
  localMacs
  listMacs lists the MAC addresses on this Ranger Gateway.

mibToXml
  mibToXml<mib_file> <output_file> ]

  mib_filename specifies the path and filename of SNMP MIB file to process.
  output_file specifies the path and filename in which to store trap definitions.
mibToXml can be used to produce an SNMP Trap definitions XML file which may be uploaded to a
ZoneRanger from the trap definitions listed in the specified SNMP MIB file.

patchinstall
  patchinstall patchfile [ -noserver ] [ -nosave ]

  patchfile Ranger Gateway patch filename as provided by Tavve Support.
  -noserver specifies to not check if the Ranger Gateway is running before installation.
  -nosave specifies to not save backup of changes indicating patch could not be removed
patchinstall is used, under the direction of Tavve Support personnel, to install patches on the
Ranger Gateway. This is for Linux and Solaris Ranger Gateways only.

patchstatus
  patchstatus

  patchstatus is used to display the list of installed patches on the Ranger Gateway. This is for Linux
and Solaris Ranger Gateways only.

patchuninstall
  patchuninstall patchid [ -noserver ]

  patchid id of the patch to remove..
  -noserver specifies to not check if the Ranger Gateway is running before installation.
patchuninstall is used, under the direction of Tavve Support personnel, to remove a patch from
the Ranger Gateway. This is for Linux and Solaris Ranger Gateways only.
patchZR

patchZR zoneranger subcommand [arguments] | infoAvailable [-timeout seconds] patch_number | listAvailable [-timeout seconds]

patchZR uploads, applies, or removes a patch from a joined ZoneRanger.

zoneranger specifies the name of the ZoneRanger

subcommand can be one of the following:

- upload [-timeout seconds] patch_number
- apply [-timeout seconds] patch_number
- remove [-timeout seconds] patch_number
- infoApplied [-timeout seconds] patch_number
- infoUploaded [-timeout seconds] patch_number
- listApplied [-timeout seconds]
- listUploaded [-timeout seconds]
- removeUploaded [-timeout seconds] patch_number

infoAvailable displays specific information about the specified patch

  -timeout specifies the number of seconds to wait for a response from the ZoneRanger. The default is 30 seconds.
  
  patch_number specifies the patch number for which to view the information

listAvailable lists what patches are available to be installed

  -timeout specifies the number of seconds to wait for a response from the ZoneRanger. The default is 30 seconds.

Note, patch_number should NOT include the .pat file extension. All patches contain an internal timeout so in most cases, the timeout does not need to be specified.

patchZR zoneranger upload [-timeout seconds] patch_number

  -timeout specifies the number of seconds to wait for any response from the ZoneRanger. The default is 30 seconds.
  
  patch_number specifies the patch number to be uploaded

The patchZR upload subcommand uploads the specified patch to the specified ZoneRanger. The patch will not be installed until the patchZR apply subcommand is issued. The patchZR upload subcommand will continue uploading the file as long as it receives periodic responses from the ZoneRanger within the specified timeout. If no response is received within the timeout period, the upload will fail.

patchZR zoneranger apply [-timeout seconds] patch_number

  -timeout specifies the number of seconds to wait for the patch to be applied. The default is 30 seconds.
  
  patch_number specifies the patch number to be applied
The `patchZR` apply subcommand applies (installs) the specified patch to the indicated ZoneRanger. If the patch has not been completely applied within the specified timeout period, the command will exit. The patch will be applied to completion.

```bash
patchZR zoneranger apply [-timeout seconds] patch_number
```

- `--timeout` specifies the number of seconds to wait for the patch to be applied. The default is 30 seconds.
- `patch_number` specifies the patch number to be applied

The `patchZR` remove subcommand removes the specified patch from the indicated ZoneRanger. If the patch has not been completely removed within the specified timeout period, the command will exit but the patch will still be removed.

```bash
patchZR zoneranger remove [-timeout seconds] patch_number
```

- `--timeout` specifies the number of seconds to wait for the patch to be removed. The default is 30 seconds.
- `patch_number` specifies the patch number to be removed

The `patchZR` infoApplied subcommand displays information about an applied patch from the indicated ZoneRanger. If the patch information has not been retrieved within the specified timeout period, the command will exit.

```bash
patchZR zoneranger infoApplied [-timeout seconds] patch_number
```

- `--timeout` specifies the number of seconds to wait for the patch information to be returned. The default is 30 seconds.
- `patch_number` specifies the patch number for which to view information

The `patchZR` infoUploaded subcommand displays information about an uploaded patch from the indicated ZoneRanger. If the patch information has not been retrieved within the specified timeout period, the command will exit.

```bash
patchZR zoneranger infoUploaded [-timeout seconds] patch_number
```

- `--timeout` specifies the number of seconds to wait for the patch information to be returned. The default is 30 seconds.
- `patch_number` specifies the patch number for which to view information

The `patchZR` listApplied subcommand displays the list of all applied patches from the indicated ZoneRanger. If the patch information has not been retrieved within the specified timeout period, the command will exit.

```bash
patchZR zoneranger listApplied [-timeout seconds]
```

- `--timeout` specifies the number of seconds to wait for the list of patches to be returned. The default is 30 seconds.

The `patchZR` listUploaded subcommand displays the list of all uploaded patches from the indicated ZoneRanger. If the patch information has not been retrieved within the specified timeout period, the command will exit.

```bash
patchZR zoneranger listUploaded [-timeout seconds]
```

- `--timeout` specifies the number of seconds to wait for the list of patches to be returned. The default is 30 seconds.

The `patchZR` remove Uploaded subcommand removes the specified patch from the indicated ZoneRanger. If the patch has not been completely removed within the specified timeout period, the command will exit but the patch will still be removed.

```bash
patchZR zoneranger remove Uploaded [-timeout seconds] patch_number
```

- `--timeout` specifies the number of seconds to wait for the patch to be removed. The default is 30 seconds.
- `patch_number` specifies the patch number to remove
The `patchZR removeUploaded` subcommand removes the specified patch from the list of all uploaded patches from the indicated ZoneRanger. If the patch information has not been retrieved within the specified timeout period, the command will exit.

**portConfig**

**portConfig** subcommand [arguments]

portConfig manages named sets of rules which describe allowable ports, protocols, and port translations used in Proxy Access Control. The portConfig command is organized as a set of subcommands, each of which supports different parameters and options. Most portConfig subcommands provide the option to operate directly on the active portConfig table of a running Ranger Gateway in real time, or to work offline with text files, which can be inspected and edited using a text editor, then installed on the Ranger Gateway when required modifications have been completed.

subcommand can be one of the following:

- **copy** [-in *input_file*] [-out *output_file*]
  
- **add** [-in *input_file*] [-out *output_file*] <port-config-name transport rg-port protocol [zr-port]>
  
- **remove** [-in *input_file*] [-out *output_file*] <port-config-name [transport] [rg-port] [protocol] [zr-port]>
  
- **merge** [-in *input_file*] [-out *output_file*] merge_file
  
- **list** [-in *input_file*] [port-config-name [transport [rg-port]]]
  
- **clear** [-f]

- **config** [item [value]]

- **test** port-config-name transport rg-port

**portConfig copy** [-in *input_file*] [-out *output_file*]

-in indicates the name of the input file containing portConfig information

-out indicates the name of the output file to write portConfig information

**portConfig copy** can be used for the following:

- To copy the content of the active portConfig table to a specified text file.
- To copy the content of a specified text file to the active portConfig table.
- To copy the content of one specified text file to another.

If no input file is specified, the active portConfig table is used as the source of the copy. If no output file is specified, the input configuration is automatically copied to the active portConfig table. If an output file is specified, the input configuration is written to the specified file and the active portConfig table is unchanged.

Note that the **portConfig copy** subcommand always outputs XML. The input format can be XML, or a simple text format. See **PortConfig File Formats** for more details.

**portConfig add** [-in *input_file*] [-out *output_file*] <port-config-name> <transport> <rg-port> <protocol> [<zr-port>]

-in indicates the name of the input file containing portConfig information

-out indicates the name of the output file to write portConfig information
port-config-name specifies the name of the port config ruleset

transport specifies the protocol of ICMP, UDP or TCP

rg-port specifies the destination port associated with the incoming request as received by the Ranger Gateway

protocol specifies the management protocol to be used for the incoming request

zrport specifies the destination port that the ZoneRanger should use when forwarding the request to the target device, or a translation rule that can be used to calculate the port that should be used based on the rg-port

The portConfig add subcommand can be used for the following purposes:

- To add a new portConfig table rule.
- To modify an existing portConfig table rule.

The port-config-name, transport, rg-port, protocol, and optional zrport parameters specify the content of the rule to be added or modified. If the input configuration already contains a rule with the matching port-config-name, transport, rg-port, and protocol, the existing rule will be replaced. Otherwise the new rule is added.

The portConfig add subcommand can read input from the active portConfig table, or from a specified text file. If no input file is specified, the active portConfig table is used. If no output file is specified, the resulting configuration is automatically copied to the active portConfig table. If an output file is specified, the resulting configuration is written to the specified file and the active portConfig table is unchanged.

portConfig remove [-in input-file] [-out output-file] <port-config-name> [
transport> [
rg-port> [protocol> [zr-port>]]]]

-in indicates the name of the input file containing portConfig information

-out indicates the name of the output file to write portConfig information

port-config-name specifies the name of the port config ruleset

transport specifies the protocol of ICMP, UDP or TCP

rg-port specifies the destination port associated with the incoming request as received by the Ranger Gateway

protocol specifies the management protocol to be used for the incoming request

zrport specifies the destination port that the ZoneRanger should use when forwarding the request to the target device, or a translation rule that can be used to calculate the port that should be used based on the rg-port.

The portConfig remove subcommand can be used to remove one or more rules from the active portConfig table, or from an offline file.

The port-config-name, and optional transport, rg-port, protocol, and zr-port parameters specify the rule to be removed. If no transport, rg-port, protocol, or zr-port values are specified, all rules that match the specified port-config-name will be removed. If a transport, rg-port, protocol, or zr-port value is specified, only the rules that contain the matching values will be removed. If no matching rules are found, the input configuration will be unchanged.
The `portConfig remove` subcommand can read input from the active portConfig table, or from a specified text file. If no input file is specified, the active portConfig table is used. If no output file is specified, the resulting configuration is automatically copied to the active portConfig table. If an output file is specified the resulting configuration is written to the specified file and the active portConfig table is unchanged.

```
portConfig merge [-in input-file] [-out output-file] <merge-file>
```

- **-in** indicates the name of the input file containing device group information
- **-out** indicates the name of the output file to write device group information
- **merge-file** specifies the name of a file that contains a set of entries to be merged to the input configuration.

`portConfig merge` can be used for the following:

- To add new portConfig table rules.
- To modify existing portConfig table rules.

The logic for merging is similar to the logic for adding a single rule. If the input configuration already contains a rule with the matching `portconfig-name`, `transport`, `rg-port`, and `protocol` to one of the rules to be merged, the existing rule will be replaced. Otherwise the new rule is added. One way in which the `portConfig merge` subcommand differs from the add subcommand is that when the merge subcommand replaces an existing rule, the existing rule is removed, and the rule that replaces it is added to the end of the configuration. As a result, the `portConfig merge` subcommand can be used to rearrange the order of rules within a configuration. If a file containing a set of rules is merged onto a portConfig table configuration, the merged rules will appear in the resulting configuration in the same order they appear in the merge file.

The `portConfig merge` subcommand can read input from the active portConfig table, or from a specified text file. If no input file is specified, the active portConfig table is used. If no output file is specified, the resulting configuration is automatically copied to the active portConfig table. If the output file is specified, the resulting configuration is written to the specified file and the active portConfig table is unchanged.

```
portConfig list [-in input_file] [<port-config-name> [<transport> [</rg-port>]]]
```

- **-in** indicates the name of the input file containing port information
- **port-config-name** specifies the name of the port config ruleset
- **transport** specifies the protocol of ICMP, UDP or TCP
- **rg-port** specifies the destination port associated with the incoming request as received by the Ranger Gateway

The `portConfig list` subcommand can be used for the following purposes:

- To list all rules in a configuration.
- To list the rules in a configuration that match a given `port-config-name`.
- To list the rules in a configuration that match a given `port-config-name` and `transport`.
- To list the rules in a configuration that match a given `port-config-name`, `transport`, and `rg-port`.
If no `port-config-name`, `transport`, or `rg-port` value is specified, all of the rules in the input configuration will be listed. Otherwise, only those rules that match specified `port-config-name`, `transport`, and `rg-port` values will be listed.

The `portConfig list` subcommand can read input from the active portConfig table, or from a specified text file. If no input file is specified, the active portConfig table is used.

**portConfig clear [-f]**

`portConfig clear` can be used to remove all rules from the active portConfig table. When `portConfig clear` is executed, the user is prompted to confirm that the active portConfig table should be cleared. If the response is “y” or “yes” (case is ignored), the active portConfig table will be cleared. Otherwise the active portConfig table will be unchanged. If the `-f` option is specified, the user will not be prompted.

**portConfig config [item [value]]**

The configuration items associated with the portConfig table are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>port_config_cache_size</code></td>
<td>Determines the maximum number of entries in the cache. Valid values are positive integers in the range 0-10000. The default value is 100.</td>
</tr>
</tbody>
</table>

`portConfig config` can be used to display or modify configuration items associated with the portConfig table.

If no `item` or `value` is specified, the values of all configuration items are listed. If an `item` is specified with no `value`, the current value of the specified configuration item is displayed. If an `item` and a `value` are specified, the value of the specific configuration item is set to the specified value.

**portConfig test <port-config-name> <transport> <rg-port>**

- `port-config-name` specifies the name of the port config ruleset
- `transport` specifies the protocol of ICMP, UDP or TCP
- `rg-port` specifies the destination port associated with the incoming request as received by the Ranger Gateway

`portConfig test` can be used to display the rule which will be used on the Ranger Gateway when presented with the specified information.

Unlike the `portConfig test` subcommand, which will list all matching rules in a portConfig table configuration for a given `port-config-name`, `transport`, and `rg-port`, the `portConfig test` subcommand performs an ordered search for the first matching rule in the active portConfig table, similar to the approach that the Ranger Gateway will use to process specific proxy requests.

**portConfig File Formats**

The various `portConfig` subcommands that generate configurations (i.e. generate, copy, add, remove, merge) all generate configuration information in an XML format. An example of this format, corresponding to the default Ranger Gateway configuration is as follows:

```xml
<port-config-list>
  <port-config name="Default">
```
<rule transport="TCP" rg-port="22" protocol="SSH"/>  
<rule transport="TCP" rg-port="443" protocol="HTTPS"/>  
<rule transport="UDP" rg-port="161" protocol="SNMP"/>  
<rule transport="ICMP"/>
</port-config>

<port-config name="ZoneRangerDefault">
    <rule transport="TCP" rg-port="22" protocol="SSH"/>
    <rule transport="TCP" rg-port="23" protocol="TELNET"/>
    <rule transport="TCP" rg-port="80" protocol="HTTP"/>
    <rule transport="TCP" rg-port="443" protocol="HTTPS"/>
    <rule transport="TCP" rg-port="5432" protocol="SQL"/>
    <rule transport="UDP" rg-port="161" protocol="SNMP"/>
    <rule transport="ICMP"/>
</port-config>

</port-config-list>

The portConfig commands that read configurations (i.e. copy, add, remove, merge, list) are able to read configuration input in the XML format, and also in a simplified text format. An example of this format, corresponding to the XML example above, is as follows:

Default TCP 22 SSH
Default TCP 443 HTTPS
Default UDP 161 SNMP
Default ICMP
ZoneRangerDefault TCP 22 SSH
ZoneRangerDefault TCP 23 TELNET
ZoneRangerDefault TCP 80 HTTP
ZoneRangerDefault TCP 443 HTTPS
ZoneRangerDefault TCP 5432 SQL
ZoneRangerDefault UDP 161 SNMP
ZoneRangerDefault ICMP

portControl

portControl zoneranger [ -list | portName setting ]

zoneranger  specifies the name of the ZoneRanger

- list displays the current port settings in the specified ZoneRanger

portName specifies the port to be configured. Supported values are: http, https, ntp, radius, snmpAgent, ssh, syslog, tacacs, telnet, tftp, trap
setting specifies the incoming value. Supported values are: disabled (not applicable for snmpAgent), eth0, eth1, both, gateway (for http, https, ssh, telnet, snmpAgent only).

portControl enables and disables various ZoneRanger services.

ZoneRanger has numerous ports which may be configured to be accessible and to determine how that port may be accessed. The snmpAgent (port 161) may not be disabled since it is needed for ZoneRanger discovery. However, to disable external access, use the Configuration > SNMP page Agent tab on the ZoneRanger web interface and uncheck all three SNMP versions next to Agent Responds To.

portMap

portMap subcommand [arguments]

portMap Manages sets of rules which determine the Port Config ruleset to use based on the source address of the requesting client and the destination address of the target device used in Proxy Access Control. The portMap command is organized as a set of subcommands, each of which supports different parameters and options. Most portMap subcommands provide the option to operate directly on the active portMap table of a running Ranger Gateway in real time, or to work offline with text files, which can be inspected and edited using a text editor, then installed on the Ranger Gateway when required modifications have been completed. Use this command to enable the Ranger Gateway to include IPv6 addresses for SNMP and ICMP.

subcommand can be one of the following:

- **copy** [-in input_file] [-out output_file]
- **add** [-in input_file] [-out output_file] src-address dest-address port-config-name
- **remove** [-in input_file] [-out output_file] src-address dest-address port-config-name
- **merge** [-in input_file] [-out output_file] merge_file
- **list** [-in input_file]
- **clear** [-f]
- **config** [item value]
- **test** src-address dest-address transport rg-port

**portMap copy** [-in input_file] [-out output_file]

- **in** indicates the name of the input file containing portMap information
- **out** indicates the name of the output file to write portMap information

portMap copy can be used for the following:

- To copy the content of the active portMap table to a specified text file.
- To copy the content of a specified text file to the active portMap table.
- To copy the content of one specified text file to another.

If no input file is specified, the active portMap table is used as the source of the copy. If no output file is specified, the input configuration is automatically copied to the active portMap table. If an output file is specified, the input configuration is written to the specified file and the active portMap table is unchanged.
Note that the `portMap copy` subcommand always outputs XML. The input format can be XML, or a simple text format. See `portMap File Formats` for examples.

```
portMap add [-in input-file] [-out output-file] <src-address> <dest-address> <port-config-name>
```

- `-in` indicates the name of the input file containing portMap information
- `-out` indicates the name of the output file to write portMap information
- `src-address` indicates the source IP address of the incoming request.
- `dest-address` indicates the destination IP address of the managed device
- `port-config-name` name of the Port Config rule

The `portMap add` subcommand can be used for the following purposes:

- To add a new portMap table rule.
- To modify an existing portMap table rule.

The `src-address`, `dest-address`, and `port-config-name` parameters specify the content of the rule to be added or modified. If the input configuration already contains a rule with the matching `src-address` and `dest-address`, the existing rule will be replaced. Otherwise the new rule is added.

The `portMap add` subcommand can read input from the active portMap table, or from a specified text file. If no input file is specified, the active portMap table is used. If no output file is specified, the resulting configuration is automatically copied to the active portMap table. If an output file is specified, the resulting configuration is written to the specified file and the active portMap table is unchanged.

To add Ipv6 addresses use the command: `portMap add ::/0 ::/0 Default`

```
portMap remove [-in input-file] [-out output-file] [<src-address> [<dest-address> [<port-config-name>]]]
```

- `-in` indicates the name of the input file containing portMap information
- `-out` indicates the name of the output file to write portMap information
- `src-address` indicates the source IP address of the incoming request.
- `dest-address` indicates the destination IP address of the managed device
- `port-config-name` name of the Port Config rule

The `portMap remove` subcommand can be used to remove one or more rules from the active portMap table, or from an offline file.

The `src-address` and optional `dest-address` and `port-config-name` parameters specify the rule to be removed. If no `dest-address`, or `port-config-name` values are specified, all rules that match the specified `port-config-name` will be removed. If a `dest-address` and `port-config-name` value are specified, only the rules that contain the matching values will be removed. If no matching rules are found, the input configuration will be unchanged.

The `portMap remove` subcommand can read input from the active portMap table, or from a specified text file. If no input file is specified, the active portMap table is used. If no output file is specified, the resulting configuration is automatically copied to the active portMap table. If an output file is specified the resulting configuration is written to the specified file and the active portMap table is unchanged.

```
portMap merge [-in input-file] [-out output-file] <merge-file>
```

The `portMap merge` subcommand can read input from the active portMap table, or from a specified text file. If no input file is specified, the active portMap table is used. If no output file is specified, the resulting configuration is automatically copied to the active portMap table. If an output file is specified the resulting configuration is written to the specified file and the active portMap table is unchanged.
- **in** indicates the name of the input file containing device group information
- **out** indicates the name of the output file to write device group information

*merge-file* specifies the name of a file that contains a set of entries to be merged to the input configuration.

**portMap merge** can be used for the following:

- To add new portMap table rules.
- To modify existing portMap table rules.

The logic for merging is similar to the logic for adding a single rule. If the input configuration already contains a rule with the matching **src-address** and **dest-address** to one of the rules to be merged, the existing rule will be replaced. Otherwise the new rule is added. One way in which the **portMap merge** subcommand differs from the add subcommand is that when the merge subcommand replaces an existing rule, the existing rule is removed, and the rule that replaces it is added to the end of the configuration. As a result, the **portConfig merge** subcommand can be used to rearrange the order of rules within a configuration. If a file containing a set of rules is merged onto a portMap table configuration, the merged rules will appear in the resulting configuration in the same order they appear in the merge file.

The **portMap merge** subcommand can read input from the active portMap table, or from a specified text file. If no input file is specified, the active portMap table is used. If no output file is specified, the resulting configuration is automatically copied to the active portMap table. If the output file is specified, the resulting configuration is written to the specified file and the active portMap table is unchanged.

**portMap list [-in input_file]**

- **in** indicates the name of the input file containing portMap information

**portMap list** can be used to list all rules in a configuration.

The **portMap list** subcommand can read input from the active portMap table, or from a specified text file. If no input file is specified, the active portMap table is used. Otherwise, the specified input file is used.

**portMap clear [-f]**

**portMap clear** can be used to remove all rules from the active portMap table. When **portMap clear** is executed, the user is prompted to confirm that the active portMap table should be cleared. If the response is “y” or “yes” (case is ignored), the active portMap table will be cleared. Otherwise the active portMap table will be unchanged. If the -f option is specified, the user is not prompted.

**portMap config [item [value]]**

The configuration items associated with the portMap table are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>log_level</strong></td>
<td>Determines the level of logging for the portMap service – values: none, short, full. Default is none.</td>
</tr>
<tr>
<td><strong>port_map_cache_size</strong></td>
<td>Determines the maximum number of entries in the cache. Valid values are positive integers in the range 0-10000. The default value is 1000</td>
</tr>
</tbody>
</table>
portMap config can be used to display or modify configuration items associated with the portMap table.

If no item or value is specified, the values of all configuration items are listed. If an item is specified with no value, the current value of the specified configuration item is displayed. If an item and a value are specified, the value of the specific configuration item is set to the specified value.

**portMap test** `<src-address> <dest-address> <transport> <rg-port>`

*src-address* indicates the source IP address of the incoming request.
*dest-address* indicates the destination IP address of the managed device
*transport* specifies the protocol of TCP, UDP or ICMP
*rg-port* specifies the destination port associated with the incoming request as received by the Ranger Gateway

portConfig test can be used to display the proxy rule which will be used on the Ranger Gateway when presented with the specified information.

Unlike the portMap list subcommand, which performs a simple lookup in a portMap table configuration, the portMap test subcommand executes full lookups of the portMap and portConfig tables, in order to verify how the Ranger Gateway should respond to specific proxy requests.

**portMap File Formats**

The various portMap subcommands that generate configurations (i.e. copy, add, remove, merge) all generate configuration information in an XML format. An example of this format, corresponding to the default Ranger Gateway configuration is as follows:

```xml
<port-map>
  <rule src-address="*.*.*.*" dest-address="@ZoneRanger" port-config-name="ZoneRangerDefault" />
  <rule src-address="*.*.*.*" dest-address="*.*.*.*" port-config-name="Default" />
</port-map>
```

The portMap commands that read configurations (i.e. copy, add, remove, merge, list) are able to read configuration input in the XML format, and also in a simplified text format. An example of this format, corresponding to the XML example above, is as follows:

```
*.*.*.* @ZoneRanger ZoneRangerDefault
*.*.*.* *.*.*.* Default
```

**profile**

```
profile [ create zoneranger [ comment ] ] |
[ listRG ] | [ deleteFromRG profile_name ] |
[ restoreFromRG zoneranger profile_name ] |
[ listZR zoneranger ] | [ deleteFromZR zoneranger profile_name ] |
```
**[restoreFromZR zoneranger profile_name]**

- **zoneranger** specifies the name of the ZoneRanger from which to take the profile
- **comment** specifies an optional comment to be stored with the profile
- **profile_name** specifies the profile to use

**create** creates a profile of the ZoneRanger configuration in the Ranger Gateway store/zr/profile directory

**listRG** lists the ZoneRanger profiles stored on this Ranger Gateway

**deleteFromRG** is used to delete a ZoneRanger profile stored on this Ranger Gateway

**restoreFromRG** is used to restore a profile from a profile stored on this Ranger Gateway

**listZR** list the profiles stored on a ZoneRanger

**deleteFromZR** is used to delete a profile stored on the ZoneRanger

**restoreFromZR** is used to restore a profile from a profile stored on the ZoneRanger

Profile is used to manage the creation, restoration and storage of ZoneRanger configurations.

**propertyGet**

**propertyGet [zoneranger] property_name**

- **zoneranger** specifies the name of the ZoneRanger
- **property_name** indicates the property to retrieve

**propertyGet** retrieves the value of the specified **property_name** from the ZoneRanger or Ranger Gateway. If the specified **property_name** does not exist, nothing is returned. The **property_name** is case-sensitive.

**propertyList**

**propertyList [zoneranger]**

- **zoneranger** specifies the name of the ZoneRanger

**propertyList** retrieves the current set of properties from the ZoneRanger or Ranger Gateway.

**propertySet**

**propertySet [zoneranger] property_name property_value**

- **zoneranger** specifies the name of the ZoneRanger
- **property_name** indicates the property to set
property_value indicates the value to assign to the property

propertySet assigns the specified property_value to the property_name on the specified ZoneRanger or Ranger Gateway. If the specified property_name does not exist, it is created. The property_name is case-sensitive.

**propertyUnset**

```
propertyUnset [zoneranger] property_name
```

- zoneranger specifies the name of the ZoneRanger
- property_name indicates the property to unset

propertyUnset clears the specified property_name on the specified ZoneRanger or Ranger Gateway. If the specified property_name does not exist, nothing is returned. The property_name is case-sensitive.

**proxyMap**

**proxyMap** subcommand [arguments]

proxyMap manages the contents of the active proxy map as well as the configurations setting of the Proxy Map service. The proxyMap command is organized as a set of subcommands, each of which supports different parameters and options. Most proxyMap subcommands provide the option to operate directly on the active proxy map of a running Ranger Gateway in real time, or to work offline with text files, which can be inspected and edited using a text editor, then installed on the Ranger Gateway when required modifications have been completed.

Subcommand can be one of the following:

- **generate [–out output_file]**
- **copy [–in input_file] [–out output_file]**
- **add [–in input_file] [–out output_file] rg-address zoneranger [zr-address] [–weight weight]**
- **remove [–in input_file] [–out output_file] rg-address [zoneranger]**
- **merge [–in input_file] [–out output_file] merge_file**
- **list [–in input_file] [rg-address]**
- **clear [-f]**
- **config [item [value]]**
- **test rg-address**
- **test zoneranger zr-address**

**proxyMap generate [–out output_file]**

- **out** indicates the name of the output file to write proxy map information

The proxyMap generate subcommand can automatically generate a basic proxy map configuration, by querying the databases of joined ZoneRangers to identify the devices they are managing.
If no output file is specified, the generated configuration is automatically copied to the active proxy map. Otherwise, the generated configuration is written to the specified file and the active proxy map is unchanged.

**Note:** The `proxyMap generate` subcommand is not as useful when NAT is in effect, because the Ranger Gateway and its joined ZoneRangers cannot identify the address mappings. However, it might still be helpful to auto-generate to a file, manually edit the resulting file to add the necessary `zr-address` values, and then install the resulting file.

```
proxyMap copy [-in input_file] [-out output_file]
```

- `-in` indicates the name of the input file containing proxy map information
- `-out` indicates the name of the output file to write proxy map information

The `proxyMapTool copy` subcommand can be used to:

- Copy the content of the active proxy map to a specified text file.
- Copy the content of a specified text file to the active proxy map.
- Copy the content of one specified text file to another.

If no input file is specified, the active proxy map is used as the source of the copy. If no output file is specified, the input configuration is automatically copied to the active proxy map. If an output file is specified, the input configuration is written to the specified file and the active proxy map is unchanged.

**Note:** The `copy` subcommand always outputs XML. The input format can be XML, or a simple text format. See **ProxyMap File Formats** for examples.

```
proxyMap add [-in input_file] [-out output_file] rg-address zoneranger [zr-address] [-weight weight ]
```

- `-in` indicates the name of the input file containing proxy map information
- `-out` indicates the name of the output file to write proxy map information
- `-weight` indicates the weight of a ZoneRanger relative to other ZoneRangers
- `rg-address` specifies the IP address or an IPv6 address in CIDR format of the incoming request to the Ranger Gateway
- `zoneranger` specifies the ZoneRanger to which to send the request
- `zr-address` specifies the IP address to use on the ZoneRanger in the case of address translation for this request
- `weight` specifies the weight of this ZoneRanger. This must be a positive integer.

The `proxyMap add` subcommand can be used to:

- Add a new proxy map entry.
- Modify an existing proxy map entry.

The `rg-address`, `zoneranger`, and optional `zr-address` parameters specify the content of the entry to be added or modified. If the input configuration already contains an entry with the matching `rg-address` and `zoneranger`, the existing entry is replaced. Otherwise, the new entry is added.

The `proxyMap add` subcommand can read input from the active proxy map, or from a specified text file. If no input file is specified, the active proxy map is used. If no output file is specified, the resulting configuration is automatically copied to the active proxy map. Otherwise, the resulting configuration is written to the specified file and the active proxy map is unchanged.
The weight option must be a positive integer which indicates the weight of each ZoneRanger relative to other ZoneRangers when choosing proxy map entries. Precedence is determined by the lowest cost (0) value. The weight is only evaluated after there are no responsive ZoneRangers at the current lowest weight.

**proxyMap remove** [-in _input_file_] [–out _output_file_] _rg-address_ [zoneranger]

- **-in** indicates the name of the input file containing proxy map information
- **-out** indicates the name of the output file to write proxy map information
- **rg-address** specifies the IP address or an IPv6 address in CIDR format of the incoming request to the Ranger Gateway
- **zoneranger** specifies the ZoneRanger to which to send the request

The `proxyMap remove` subcommand can be used to remove one or more entries from the active proxy map, or from an offline file.

The **rg-address** and optional **zoneranger** parameters specify the entry to be removed.

If no **zoneranger** value is specified, all entries that match the specified **rg-address** are removed. If a **zoneranger** value is specified, only the entry that contains the matching **rg-address** and **zoneranger** values is removed.

If no matching entries are found, the input configuration will be unchanged. The `proxyMap remove` subcommand can read input from the active proxy map, or from a specified text file. If no input file is specified, the active proxy map is used. If no output file is specified, the resulting configuration is copied to the active proxy map. Otherwise, the resulting configuration is written to the specified file and the active proxy map is unchanged.

**proxyMap merge** [-in _input_file_] [–out _output_file_] _merge_file_

- **-in** indicates the name of the input file containing proxy map information
- **-out** indicates the name of the output file to write proxy map information
- **merge_file** specifies the name of a file that contains a set of entries to be merged to the input configuration

The `proxyMap merge` subcommand can be used to:

- Add new proxy map entries.
- Modify existing proxy map entries.

The logic for merging is similar to the logic for adding an entry. If the input configuration already contains an entry with the matching **rg-address** and **zoneranger** to one of the entries to be merged, the existing entry is replaced. Otherwise the new entry is added.

One way that the `proxyMap merge` subcommand differs from the `add` subcommand is that when the `proxyMap merge` subcommand replaces an existing entry where **rg-address** is an address pattern, the existing entry is removed, and the entry that replaces it is added to the end of the configuration. As a result, the `proxyMap merge` subcommand can rearrange the order of address pattern entries in a configuration.

If a file containing a set of address pattern entries is merged onto a proxy map configuration, the merged entries will appear in the resulting configuration in the same order they appear in the merge file.
The `proxyMap merge` subcommand can read input from the active proxy map, or from a specified text file. If no input file is specified, the active proxy map is used. If no output file is specified, the resulting configuration is automatically copied to the active proxy map. If an output file is specified, the resulting configuration is written to the specified file and the active proxy map is unchanged.

**proxyMap list [-in input_file] [rg-address]**

- `-in` indicates the name of the input file containing proxy map information
- `-out` indicates the name of the output file to write proxy map information
- `rg-address` specifies the IP address or an IPv6 address in CIDR format of the incoming request to the Ranger Gateway

The `proxyMap list` subcommand can be used to:

- List all entries in a configuration.
- List the entries in a configuration that match a given `rg-address`.

If no `rg-address` value is specified, all entries in the input configuration are listed. Otherwise, only those entries that match the specified `rg-address` are listed.

**Note:** The resulting list might include entries with a specific `rg-address` value that exactly matches the input value, as well as entries where the `rg-address` value is a matching address pattern. The `proxyMap list` subcommand can read input from the active proxy map, or from a specified text file. If no input file is specified, the active proxy map is used. Otherwise, the specified input file is used.

**proxyMap clear [-f]**

The `proxyMap clear` subcommand is used to clear (remove all entries from) the active proxy map. If the `-f` option is specified, the user is not prompted for confirmation.

When the `proxyMap clear` subcommand is executed, the user is prompted to confirm that the active proxy map should be cleared. If the response is `y` or `yes` (case is ignored), the active proxy map is cleared. Otherwise, the active proxy map is not changed.

**proxyMap config [[item [value]]]**

The configuration items associated with the proxy map service are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>log_level</code></td>
<td>Determines the level of logging for the proxy map service – values: <code>none</code>, <code>short</code>, <code>full</code>. Default is <code>none</code>.</td>
</tr>
<tr>
<td><code>resolve_host_names</code></td>
<td>Specifies whether <code>rg-address</code> values should be resolved before searching for matching entries in the active proxy map. The default value is <code>true</code>.</td>
</tr>
<tr>
<td><code>allow_unconfigured_routes</code></td>
<td>Specifies whether the proxy map service should simply select the best available ZoneRanger to relay requests in the absence of a matching entry in the active proxy map. The default value is <code>true</code>.</td>
</tr>
<tr>
<td><code>balance_zoneranger_selection</code></td>
<td>Specifies whether the proxy map service should attempt to balance proxy transaction load across the available ZoneRangers. The default value is <code>true</code>.</td>
</tr>
<tr>
<td><code>route_list_cache_size</code></td>
<td>Determines the maximum number of entries in the cache. Valid values are positive integers in the range 0-10000. The default value is 1000.</td>
</tr>
</tbody>
</table>
proxyMap config: can be used to display or modify configuration items associated with the proxy map service.

If no item or value is specified, the values of all configuration items are listed. If an item is specified with no value, the current value of the specified configuration item is displayed. If an item and a value are specified, the value of the specific configuration item is set to the specified value.

Note, For maximum performance, the value of the route_list_cache_size configuration item should be set as follows:

- If all active proxy map entries have rg-address values which are address patterns, the route_list_cache_size should be set to a number greater than or equal to the number of distinct address patterns.
- Otherwise, the route_list_cache_size should be set to a number greater than or equal to the number of DMZ devices to which proxy requests may be directed via this Ranger Gateway.

proxyMap test rg-address

The proxyMap test subcommand performs a query on the proxy map service to test how it responds to requests to select zoneranger and zr-address values for a given rg-address.

Unlike the proxyMap list subcommand, which performs a simple lookup in proxy map configuration, the proxyMap test subcommand executes the full selection algorithm taking into account configuration settings, the content of the active proxy map, and ZoneRanger status and history.

ProxyMap File formats

The various proxyMap subcommands that generate configurations (generate, copy, add, remove, merge) generate configuration information in an XML format.

An example of this format, corresponding to the example configuration in follows:

```xml
<proxy-map>
  <route-list rg-addr="62.1.25.15">
    <route zr="ZR-1"/>
  </route-list>

  <route-list rg-addr="62.1.25.30">
    <route zr="ZR-1"/>
  </route-list>

  <route-list rg-addr="62.2.37.1">
    <route zr="ZR-2" zr-addr="192.168.1.1"/>
    <route zr="ZR-3" zr-addr="192.168.1.1"/>
  </route-list>

  <route-list rg-addr="62.2.37.2">
    <route zr="ZR-2" zr-addr="192.168.1.2"/>
  </route-list>
</proxy-map>
```
The proxyMap commands that read configurations (copy, add, remove, merge, list) can read configuration input in the XML format, and also in a simplified text format.

An example of the text format corresponding to the preceding XML example follows:

```
62.1.25.15 ZR-1
62.1.25.30 ZR-1
62.2.37.1 ZR-2 192.168.1.1
62.2.37.1 ZR-3 192.168.1.1
62.2.37.2 ZR-2 192.168.1.2
62.2.37.2 ZR-3 192.168.1.2
62.2.37.3 ZR-2 192.168.1.3
62.2.37.3 ZR-3 192.168.1.3
```

**Note:** All subcommands that generate configurations automatically group entries that have the same rg-address value into route-list elements, and order the resulting route-list elements according to the following rules:

- All elements where the rg-address value is not an address pattern are listed first, in lexicographical order by rg-address.
- All elements where the rg-address value is an address pattern are listed last, in the order in which they were originally created.

Note that there is a slight difference to the way that the proxyMap add and proxyMap merge subcommands handle ordering of modified address pattern elements. The proxyMap add subcommand leaves modified elements in their original order, while the proxyMap merge subcommand effectively moves the modified element to the end of the configuration. The reason for this is to allow the proxyMap merge subcommand to be used to reorder the address pattern elements in a configuration.

The order of address pattern elements within the active proxy map is significant because it controls how the proxy map service algorithm selects a route-list for a given rg-address. In general, the algorithm for looking up a route-list for a given rg-address uses the following rules:

- If there are any route-list elements in the active proxy map with an rg-address value that is not an address pattern and exactly matches the specified address, that route-list is selected, and any matching address pattern root-list elements are ignored.
- Otherwise, the first route-list element with a matching address pattern rg-address value is selected.
RangerGateway

RangerGateway

RangerGateway starts the Ranger Gateway Viewer GUI.

removeTftpFile

removeTftpFile

```
removeTftpFile zoneranger filename
```

- **zoneranger** specifies the name of the zoneranger to remove the file
- **filename** specifies the file to remove.

removeTftpFile removes a file from the ZoneRanger TFTP directory.

rgBackup

```
rgBackup [ -d directory ] backup | restore filename [ -nostart ]
```

- **directory** specifies the name of the directory to use for backups
- **filename** specifies the backup file to restore.

rgBackup creates or restores a backup of the Ranger Gateway configuration. If a directory is not specified, the `<install_dir>/backup` directory will be used. Only a backup of the same Ranger Gateway version may be restored. The `-nostart` option causes the Ranger Gateway to NOT restart after the backup is restored.

rgvi

```
rgvi subcommand [arguments]
```

rgvi manages the allowable clients and routes for the remote gateway virtual interface to provide communications with OpenVPN clients. rgvi command is organized as a set of subcommands, each of which supports different parameters and options. The minimum release of the OpenVPN client required to support IPv6 is 2.3.

Subcommand can be one of the following:

- **enable**
- **disable**
- **status**
- **add-client <client-address>**
- **remove-client <client-address>**
- **clear-clients [-f]**
- **list-clients**
- **use-gvi-routes <client-address>**
- **add-route <client-address> <subnet> [<subnet>...]**
- **remove-route <client-address> <subnet> [subnet>...]**
- **list-routes [<client-address>]**
- **clear-routes <client-address> [-f]**
- **config [item [value]]**

rgvi enable
By default, the RGVI service is disabled. `rgvi enable` subcommand enables the RGVI service. When the `rgvi enable` subcommand is executed, the RGVI service in the Ranger Gateway will begin listening for OpenVPN client connections. The RGVI service will not accept connections from OpenVPN clients unless those clients have been added by the `rgvi add-client` subcommand. If the RGVI service had previously been configured with client-addresses and routes then disabled using the `rgvi disable` subcommand, that configuration will be reasserted when the `rgvi enable` subcommand is executed.

`rgvi disable`

`rgvi disable` subcommand disables the RGVI service. When the `rgvi disable` subcommand is executed, the RGVI service on the Ranger Gateway will disconnect any current OpenVPN client connections and stop listening for future OpenVPN client connections.

`rgvi status`

`rgvi status` subcommand displays the current status of the RGVI service. The `rgvi status` subcommand indicates whether the RGVI service is currently enabled or disabled.

`rgvi add-client <client-address>`

`<client-address>` indicates the set of OpenVPN client addresses which may connect to this Ranger Gateway. It may be an IP address or an IPv6 address in CIDR format, an address pattern, or a device group.

`rgvi add-client` subcommand specifies which OpenVPN clients may connect to the RGVI service on the Ranger Gateway.

`rgvi remove-client <client-address>`

`<client-address>` indicates the set of OpenVPN client addresses which will no longer be allowed to connect to this Ranger Gateway. It may be an IP address or an IPv6 address in CIDR format, an address pattern, or a device group.

`rgvi remove-client` subcommand removes one or more OpenVPN client addresses that were allowed to connect to the RGVI service on the Ranger Gateway.

`rgvi clear-clients [-f]`

`[-f]` prompts for user confirmation before clearing the OpenVPN client address list.

`rgvi clear-clients` subcommand removes all OpenVPN client addresses that were allowed to connect to the RGVI service on the Ranger Gateway.

`rgvi list-clients`

`rgvi list-clients` subcommand displays the current list of OpenVPN client addresses that are allowed to connect to the RGVI service on the Ranger Gateway.

`rgvi use-gvi-routes <client-address>`

`<client-address>` indicates the set of OpenVPN client addresses which will use routes defined in the GVI.

`rgvi use-gvi-routes` subcommand applies the set of routes defined in the GVI to this set of OpenVPN client addresses.

`rgvi add-route <client-address> <subnet> [...]`

`<client-address>` indicates the set of OpenVPN client addresses to which to add routes.
**subnet** indicates the subnet or individual IP address or an IPv6 address in CIDR format to add to the OpenVPN client address.

**rgvi add-route** subcommand adds one or more subnets or individual IP addresses or IPv6 addresses in CIDR format to the specified OpenVPN client address.

The route manager within the RGVI service maintains a persistent list of subnets and individual IP addresses or IPv6 addresses in CIDR format that correspond to DMZ devices, and therefore, should be routed to an OpenVPN client address. The **rgvi add-route** subcommand can be used to add one or more subnets or individual IP addresses or IPv6 addresses in CIDR format to this list. When an OpenVPN client address connects to the RGVI service, the RGVI service sends the routes that are configured for that client address.

**Important Note:** The set or host/subnet addresses to be intercepted by an RGVI client is pushed to the RGVI client at the point where the client connects with the Ranger Gateway, and cannot be modified after the connection is established. As a result, whenever the set of host/subnet addresses to be intercepted by a client is modified on the Ranger Gateway, it will be necessary to restart any affected clients.

Each parameter after the add-route subcommand name can either be a specific IP address or an IPv6 address in CIDR format, or a subnet description. Any of the following formats can be used to describe a subnet:

- 10.1.10.*
- 10.1.10.[0-255]
- 10.1.10.0/255.255.255.0
- 2001:db8::/48

**rgvi remove-route** `<client-address><subnet>[<subnet>...]`

`<client-address>` indicates the set of OpenVPN client addresses to which to remove routes.

`subnet` indicates the subnet or individual IP address or an IPv6 address in CIDR format to remove from the OpenVPN client address.

**rgvi remove-route** subcommand removes one or more subnets or individual IP addresses or IPv6 addresses in CIDR format from the specified OpenVPN client address. If the RGVI service is enabled and the OpenVPN client address is already connected, the OpenVPN client address must disconnect and reconnect to the RGI service to receive an updated route list.

**rgvi list-routes** `<client-address>`

`<client-address>` indicates the set of OpenVPN client addresses for which to display routes (optional).

**rgvi list-routes** command lists all routes for each OpenVPN client address or only the routes of the specified OpenVPN client address.

**rgvi clear-routes** `<client-address>` [-f]

`<client-address>` indicates the set of OpenVPN client addresses for which to clear routes.

[-f] skips the prompt for user confirmation before clearing the OpenVPN client address list.
rgvi clear-routes command removes all routes from the specified OpenVPN client address.

**rgvi config [item [value]]**

rgvi config can be used display or modify configuration items associated with the RGVI service.

The configuration items associated with the RGVI service are:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>log_level</td>
<td>Determines the level of logging for the RGVI service – values: none, short, full. Default is none.</td>
</tr>
<tr>
<td>encrypted</td>
<td>Whether or not communications between the OpenVPN client and the Ranger Gateway is encrypted. Default is true.</td>
</tr>
<tr>
<td>inactivity_threshold</td>
<td>Sets the time for the connection to timeout if there is no activity.</td>
</tr>
<tr>
<td>mtu_size</td>
<td>Sets the MTU for the tunnel.</td>
</tr>
<tr>
<td>ping_interval</td>
<td>Sets the interval in seconds for the activity ping</td>
</tr>
<tr>
<td>port</td>
<td>Port which the RGVI services is listening for OpenVPN clients. Default is 1194</td>
</tr>
</tbody>
</table>

If no item or value is specified, the values of all configuration items are listed. If an item is specified with no value, the current value of the specified configuration item is displayed. If an item and a value are specified, the value of the specific configuration item is set to the specified value.

**servicedump**

**servicedump zoneranger [-i[info] | -s[top] | -t[arget] location name from to]**

**servicedump –rg**

-rg creates a Ranger Gateway service dump

**servicedump zoneranger [-i[info] | -s[top] | -t[arget] location name from to]**

zeranger specifies the name of the zoneranger to perform the service dump

-[-i[info]] reports the status of the service dump.
-[-s[top]] stops the service dump.
-[-t[arget]] performs a targeted service dump.

servicedump generates a file containing diagnostic information about Ranger Gateway or ZoneRanger system problems. servicedump is useful when working with Tavve Technical Support.

File creation using servicedump continues across Ranger Gateway restarts. If you kill the shell or command window that spawned a service dump, or if the shell or command window is otherwise stopped, file creation continues.
setPasscode

```
setPasscode passcode
```

`passcode` indicates the default passcode the Ranger Gateway should use.

`setPasscode` changes the passcode of the Ranger Gateway. The passcode is used when joining to ZoneRangers.

shutdownSystem

```
shutdownSystem zoneranger -restart | -reboot | -shutdown
```

`zoneranger` specifies the name of the ZoneRanger

- `restart` indicates that the ZoneRanger application software be restarted.
- `reboot` indicates that the ZoneRanger be rebooted.
- `shutdown` indicates that the ZoneRanger software be turned off.

`shutdownSystem` restarts, reboots, and shuts down the specified `zoneranger`.

**Note:** This command *does not* prompt for confirmation.

snmpRequest

`snmpRequest` is an SNMP utility that can perform SNMP Set, Get, and GetNext requests, and listens for traps and informs. The command is useful if you are working on a machine that does not run SNMP commands.

Some `snmpRequest` examples follow:

- Listen to port 162 for SNMPv1 and SNMPv2 traps:
  ```
  snmpRequest -O I 0.0.0.0/162
  ```

- Walk the system MIB using the community `public` and SNMPv1:
  ```
  snmpRequest -p GETNEXT -O w -v 1 -c public ZR500/161 1.3.6.1.2.1.1
  ```

- Walk the system MIB using SNMPv3, user `andy`, authentication password `authpass1`, privacy password `authpass2`, and an increased timeout of 15 seconds:
  ```
  snmpRequest -p GETNEXT -O w -v 3 -u andy -A authpass1 -a MD5 -x privpass1
  ```

- Send an SNMPv1 trap with community `public`, enterprise 1.2.3, generic 6, specific 42, and variable binding 1.2.3.1.0 => "Test 1."
  ```
  snmpRequest -p V1TRAP -v 1 -c public -Ce 1.2.3 -Cg 6 -Cs 42
  ```

sqlQuery

```
sqlQuery zoneranger [-s separator] [ [-tables] | [-cols tablename] | [sql_query] ]
```

`zoneranger` specifies the name of the ZoneRanger.
- \( s \) specifies the string which separator returned columns
- \( \text{tables} \) returns the list of tables form the ZoneRanger database
- \( \text{cols} \) returns the columns from the specified table

```
sql_query
```

indicates the SQL query to perform

\( sqlQuery \) queries SQL database tables on the specified ZoneRanger. The \( sqlQuery \) command can be used to retrieve database information stored in the specified ZoneRanger which includes node, interface, and general network connectivity of managed devices.

You can query the following database tables: cloud entity interface, node, otherrrelationentity, rootcause, subnet, tcpport

Note, on some operating systems, if the \( sql_query \) contains *, then it must be escaped or the \( sql_query \) must be enclosed in quotes.

**trapFwdLogParser**

```
trapFwdLogParser [input_file [output_file]]
```

- \( \text{input_file} \) specifies the file to containing hex encoded traps
- \( \text{output_file} \) specifies the file to which to write readable traps

\( \text{trapFwdLogParser} \) translates the internally formatted traps that appear in the trap.log file, when the trap forwarding logging level is set to \textbf{full}, to a readable format.

If no \( \text{input_file} \) is specified, the standard input is used. If no \( \text{output_file} \) is specified, the standard output is used.

**trapdToXml**

```
trapdToXml [trapd_file] [(-preserve|-nopreserve) input_file [output_file]]
```

- \( \text{trapd_file} \) is the trapd.conf style file (i.e. OpenView NNM, Tivoli NetView)
- \( \text{-preserve} \) preserves original trap definitions in \( \text{input_file} \) if conflicts occur
- \( \text{-nopreserve} \) overwrites original trap definitions in \( \text{input_file} \) if conflicts occur

\( \text{input_file} \) is an optional trap-definitions.xml style file, which is merged with the traps from the trapd.conf file.

\( \text{output_file} \) specifies file to write trap definitions in xml format.

\( \text{trapdToXml} \) converts a NetView or OpenView trapd.conf trap definition file to a format that ZoneRanger can use.

If \( \text{output_file} \) is not specified, the created file name is \texttt{trap-definitions.xml}. You can then use the ZoneRanger web interface or the uploadConfig command to upload the converted trap definitions.

**trapXmlValidator**

```
trapXmlValidator trap_definitions_xml_file
```

- \( \text{trap_definitions_xml_file} \) is a trap-definitions.xml style file.

\( \text{trapXmlValidator} \) verifies that the syntax of the input file is valid so that a ZoneRanger can load the trap definitions file.
troubleshootNetwork

troubleshootNetwork zoneranger [-timeout seconds] command [arguments]

zoneranger specifies the name of the ZoneRanger

-timeout specifies the number of seconds for the command to complete

command is ping, nslookup, traceroute or snmpget

arguments indicates specific command arguments

troubleshootNetwork executes ping, nslookup, traceroute and snmpget commands on a ZoneRanger. Ping, traceroute and snmpget support IPv6 addresses.

trustedSSL

trustedSSL -listMessagingSubjects | -addMessagingSubject [-subject value] | -removeMessagingSubject [-number index] | -listRgviSubjects | -addRgviSubject [-subject value] | -removeRgviSubject [-number index] | -listCas [-v] | -addCa [-file cert_file] [-keystore file] | -removeCa [-number indices] | -revertCas

trustedSSL configures which certificates the Ranger Gateway trusts. If trustedSSL is executed with no parameters, the following options list is displayed:

1. List trusted messaging subjects
2. Add trusted messaging subject
3. Remove trusted messaging subject
4. List trusted RGVI subjects
5. Add trusted RGVI subject
6. Remove trusted RGVI subject
7. List trusted certificate authorities
8. Add trusted certificate authority
9. Remove trusted certificate authority
10. Revert to factory certificate authorities.
11. Display usage

Option 1: List trusted messaging subjects

trustedSSL -listMessagingSubjects

This option is used to display the current list of trusted messaging subjects on the Ranger Gateway. This is used for ZoneRanger communications.

Option 2: Add trusted messaging subject

trustedSSL -addMessagingSubject [-subject value]

-subject specifies the trusted messaging subject to add to the Ranger Gateway list.

This option is used to add the specified messaging subject to the Ranger Gateway configuration. This is used for ZoneRanger communications.

Option 3: Remove trusted messaging subject

trustedSSL -removeMessagingSubject [-number index]
-number specifies the index number of the trusted messaging subject as returned from the listSubjects subcommand to be removed from the Ranger Gateway list.

This option is used to remove the specified messaging subject from the Ranger Gateway configuration.

Option 4: List trusted RGVI subjects

trustedSSL -listRgviSubjects

This option is used to display the current list of trusted RGVI subjects on the Ranger Gateway. This is used for OpenVPN communications.

Option 5: Add trusted RGVI subject

trustedSSL -adRgviSubject [-subject value]

-subject specifies the trusted RGVI subject to add to the Ranger Gateway list.

This option is used to add the specified RGVI subject to the Ranger Gateway configuration. This is used for OpenVPN communications.

Option 6: Remove trusted RGVI subject

trustedSSL -removeRgviSubject [-number index]

-number specifies the index number of the trusted RGVI subject as returned from the listSubjects subcommand to be removed from the Ranger Gateway list.

This option is used to remove the specified RGVI subject from the Ranger Gateway configuration.

Option 7: List trusted certificate authorities

trustedSSL -listCas [-v]

-v displays expanded information for each trusted certificate authority

This option is used to list the currently trusted certificate authorities for the Ranger Gateway.

Option 8: Add trusted certificate authority

trustedSSL -addCa [-file cert_file ] | [-keystore file]

-file specifies the name of X.509 certificate file containing the trusted certificate authority information.

-keystore specifies the name of the keystore file containing the trusted certificate authority information.

This option is used to add a new trusted certificate authority from the specified file to the Ranger Gateway configuration.

Option 9: Remove trusted certificate authority

trustedSSL -removeCa [-number indices]

-number specifies the index number of the trusted subject as returned from the listSubjects subcommand to be removed from the Ranger Gateway list.

This option is used to remove a trusted certificate authority from the Ranger Gateway configuration.
Option 10: Revert to factory certificate authorities

trustedSSL -revertCas

This option is restores the trusted certificate authority list to that of the original Ranger Gateway installation.

tuntap

tuntap -install | -remove | -status

tuntap installs, removes, or displays the status of the driver required to be installed to use the gvi on Windows. This must be installed upon completion of Ranger Gateway installation before running the gvi command. (Windows only)

unjoinAll

unjoinAll

unjoinAll unjoins from all joined ZoneRangers.

unjoinRequest

unjoinRequest zoneranger

<zoneranger> specifies the name of the ZoneRanger

unjoinRequest unjoins from a ZoneRanger.

uploadConfig

uploadConfig zoneranger (-hosts | -traps) configFile

zoneranger specifies the name of the ZoneRanger

-hosts indicates that configFile is in the hostname mappings format.

-traps indicates that configFile is in the trap definitions format.

configFile specifies the file to be uploaded to the indicated ZoneRanger.

uploadConfig uploads a configuration file, for either trap definitions or hostname to IP address mappings, to a ZoneRanger.

configFile must be located in the upload directory.

A sample hostname mappings formatted file, named hosts-config.xml, and the XML schema describing the format, named hosts-config.xsd, are located in the ZRCustom directory.

A sample trap definitions formatted file, named trap-definitions.xml, contains the Tavve traps. The XML schema describing the format is named trap-definitions.xsd. Both are located in the ZRCustom directory.

uploadTftpFile

uploadTftpFile zoneranger file

zoneranger specifies the name of the ZoneRanger

uploadTftpFile uploads a file to the ZoneRanger TFTP directory.

viewIcmpLatency

viewIcmpLatency zoneranger [ipAddress1 ... ipAddressN]
zoneranger specifies the name of the ZoneRanger

ipAddress indicates a specific set of IP addresses from which to request ICMP latency.

viewIcmpLatency lists the ICMP latency for the addresses polled by the specified ZoneRanger.

ZoneRanger keeps the latency in memory from the last ICMP poll for all devices it is polling.
viewIcmpLatency can be used to retrieve the ICMP latency for the specified addresses. If no addresses are specified, the ICMP latency for all polled devices will be returned.

viewRoutes

viewRoutes zoneranger

zoneranger specifies the name of the ZoneRanger

viewRoutes displays the routing table of the specified ZoneRanger.

viewRouteTestPeriod

viewRouteTestPeriod zoneranger

zoneranger specifies the name of the ZoneRanger

viewRouteTestPeriod displays how long addRoute temporarily adds routes to the ZoneRanger routing table. The changeRouteTestPeriod command can be used to change this value.
Part V. ZoneRanger Applications

This section introduces and describes separately licensed feature sets which provide application specific functionality in ZoneRanger.

Separately licensed features are distributed by Tavve as ZoneRanger patches. Each Tavve license patch is specific to the ZoneRanger upon which it can be installed and may not be installed on another ZoneRanger.
Chapter 39: HP OM

ZoneRanger has the ability, via a separately licensed application, to support Hewlett-Packard's Operations Manager (OM) application. The HP OM application licensed on ZoneRanger allows HP OM agents residing on ZoneRanger managed devices to securely initiate communications with HP OM servers residing within the secure area of the network.

In order for the ZoneRanger and Ranger Gateway to proxy requests between the HP OM agents and HP OM server, the HP OM agents need to be configured to send requests to a ZoneRanger configured port. Basically, the ZoneRanger will appear to be the HP OM server to the HP OM agents. The ZoneRanger will then proxy those requests through a Ranger Gateway to the HP OM server in the secure network. The HP OM server responses will be proxied through the Ranger Gateway and ZoneRanger back to the HP OM agents.

**HP OM Certificates**

HP OM agents communicate with an HP OM server using the HTTPS protocol. This requires that SSL certificates be used between the HP OM server and HP OM agents. However, ZoneRanger breaks the HTTPS connection between the HP OM agent and the HP OM server. Thus, in order to have successful communications, the ZoneRanger must be configured with the proper SSL certificate and private key in order to connect with the HP OM agent.

The HP OM server may also be configured to require client authentication with the HP OM agents. In this case, the SSL certificate and private key will also be used to authenticate to the HP OM server.
On the Configuration > Inbound Proxy page HP OM tab, the Add HP OM Options button allows for the specification of the SSL Certificate, Private Key, and Trusted CA Certificates to be used by ZoneRanger when communicating with HP OM agents and HP OM servers. The Certificate and Private Key section define the information necessary to authenticate communications between HP OM agents and HP OM servers and the ZoneRanger.

The Trusted Certificate Authorities section defines which certificates will be trusted. The certificate will be verified with one of the configured Certificate Authorities. The Add Certificate Authority section may be used to add Trusted Certificate Authorities.

The HTTP Enabled checkbox specifies whether or not HTTP communications with HP OM agents is allowed. This is useful for HP OM agents that initially acquire their HTTPS SSL certificates using HTTP.
Management Application Servers

A management application server is the hostname or IP address of a server on which a particular management application resides to which the ZoneRanger should proxy information coming from a ZoneRanger managed node. In this case, these would be the hostnames or IP addresses of the HP OM servers to which the HP OM agents need to communicate. One or more Ranger Gateways may be used to reach a particular management application server, in this case, HP OM server. The Ranger Gateways may be installed on the HP OM server itself or on another system. The set of Ranger Gateways needed to reach each HP OM server must be defined on the ZoneRanger. The path to each HP OM server is used when defining Inbound Proxy rules on the Configuration > Inbound Proxy page HP OM tab..
The **Proxy Rules** section defines the association of HP OM agent addresses and the local ZoneRanger TCP port to which those addresses are sending requests, the HP OM options which should be used with each rule, and the destination list of management application servers (HP OM servers) to which to proxy requests. The **Source Address** must be an IP address or an address pattern. If more than one application server is specified as a destination, ZoneRanger will attempt each destination until it can successfully proxy the request. Once ZoneRanger determines a successful management application server destination, it will continue to use that destination until a proxy request fails.

HP OM requests received by a ZoneRanger, and HP OM responses sent by a ZoneRanger can be written to a log file, called `/log/hpomProxy.log`. This log can be downloaded using the `downloadFile` command on a Ranger Gateway. This can affect the performance of HP OM proxy. The Log Levels are:
<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Logging is off</td>
</tr>
<tr>
<td>short</td>
<td>Basic information about each HP OM proxy transaction request is logged, including entries for Opened, Closed, and Failed events.</td>
</tr>
<tr>
<td>full</td>
<td>Additional information is added to the log, including: an Opening event entry as the proxy is connecting, enhanced proxy path details with ports.</td>
</tr>
</tbody>
</table>

Destinations

Each HP OM Proxy Rule has a set of **Destination Management Application Servers** which are paths to HP OM Servers. The Management Application Server may either be a joined Ranger Gateway, as indicated by a preceding “RG:” or a path to a Management Application Server as configured on the **Configuration > Ranger Gateway** page **Mgmt App Servers** tab. The Port is the TCP port to which HP OM requests will be sent.
Indicators and Statistics

When the HP OM application is licensed on the ZoneRanger, there are several mechanisms which indicate the status of proxy requests between HP OM agents and HP OM servers.

In the Statistics section on the ZoneRanger dashboard, there is a chart which is specific to HP OM proxy requests. As with other charts, this chart shows the last 4 hours of HP OM proxy traffic based on the statistics recorded by the ZoneRanger. Specific ZoneRanger statistics are also available on the View > Statistics page when viewing the TCP Proxy service. A Status Indicator in the Activity section called HP OM Proxy will light when HP OM proxy requests are processed by the ZoneRanger. As with other status indicators, the light will flash as a result of HP OM proxy traffic on an occasional basis and not flash the exact number of HP OM proxied requests.
Chapter 40: Web File

ZoneRanger has the ability, via a separately licensed application, to support the placement and retrieval of files by using HTTP or HTTPS Get and Put requests. The Web File application licensed on ZoneRanger allows applications to initiate communications with web servers residing within the secure area of the network in order to put or retrieve files.

Figure 40-1. Web File Proxy Requests

In order for the ZoneRanger and Ranger Gateway to proxy requests between the Web File agents and web server, the Web File agents need to be configured to send HTTP or HTTPS requests to a ZoneRanger configured port. Basically, the ZoneRanger will appear to be the Web server to the Web File agents. The ZoneRanger will then proxy those requests through a Ranger Gateway to the Web server in the secure network. The Web server responses will be proxied through the Ranger Gateway and ZoneRanger back to the Web File agents.

Web File Options

Web File agents communicate with the Web server using either HTTP or HTTPS protocols. In the case of a HTTP Get File or HTTP Put File request, the ZoneRanger will validate the name of the file in the request based on a configuration setting. In the case of a HTTPS request, since HTTPS connections are encrypted, the ZoneRanger will proxy the request through to the Ranger Gateway directly.
On the Configuration > Inbound Proxy page Web File tab, the Add Web File Options button allows for the specification of which protocol to use (HTTP or HTTPS) and in the case of HTTP, the Get and Put Filename pattern to use for validation.

The HTTP Enabled checkbox indicates HTTP communications with Web File agents is required. The Get Filename Pattern and Put Filename Pattern allow a limited set of pattern matching special characters. The set of allowed characters in filenames are: a-z, A-Z, 0-9, . (period), - (dash), _ (underscore) and space. The * (star) special character represents one or more valid filename characters. The [] special characters list the possible single valid characters. For example, [a-c] would be valid either an a, b or c.

The HTTPS Enabled checkbox indicates HTTPS communications with Web File agents is required. The HTTP connection is directly proxied to the ultimate destination.

It is valid to allow both HTTP and HTTPS communications in the same Web File Options configuration. It is also valid to specify neither HTTP or HTTPS communications (both unchecked) in the same Web File Options configuration. In the later case, no connections will be proxied.
Management Application Servers

A management application server is the hostname or IP address of a server on which a particular management application resides to which the ZoneRanger should proxy information coming from a ZoneRanger managed node. In this case, these would be the hostnames or IP addresses of the Web servers to which the Web File agents need to communicate. One or more Ranger Gateways may be used to reach a particular management application server, in this case, Web server. The Ranger Gateways may be installed on the Web server itself or on another system. The set of Ranger Gateways needed to reach each Web server must be defined on the ZoneRanger. The path to each Web server is used when defining Web File Proxy rules on the Configuration > Inbound Proxy page Web File tab.

Figure 40-3. Configure > Ranger Gateway page Mgmt App Servers tab
Web File Proxy

Proxy Rules

The Proxy Rules section defines the association of Web File agent addresses and the local ZoneRanger TCP port to which those addresses are sending requests, the Web File options which should be used with each rule, and the destination list of management application servers (Web servers) to which to proxy requests. The Source Address must be an IP address or an address pattern. If more than one application server is specified as a destination, ZoneRanger will attempt each destination until it can successfully proxy the request. Once ZoneRanger determines a successful management application server destination, it will continue to use that destination until a proxy request fails.

Web File requests received by a ZoneRanger, and Web File responses sent by a ZoneRanger can be written to a log file, called /log/webFileProxy.log. This log can be viewed using the View > Service Logs on the ZoneRanger web GUI and it can be downloaded using the downloadFilecommand on a Ranger Gateway. This can affect the performance of Web File proxy. The Log Levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Logging is off</td>
</tr>
<tr>
<td>short</td>
<td>Basic information about each file transfer request is logged, as well as failed attempts to proxy and connections closed due to protocol violations or invalid file names.</td>
</tr>
<tr>
<td>full</td>
<td>Additional information is added to the log, including state change events for the underlying TCP proxy connection, and enhanced proxy path details with ports.</td>
</tr>
</tbody>
</table>
Destinations

Each Web File Proxy Rule has a set of Destination Management Application Servers which are paths to Web Servers. The Management Application Server may either be a joined Ranger Gateway, as indicated by a preceding “RG:” or a path to a Management Application Server as configured on the Configuration > Ranger Gateway page Mgmt App Servers tab. The Port is the TCP port to which Web server requests will be sent.
Indicators and Statistics

When the Web File application is licensed on the ZoneRanger, there are several mechanisms which indicate the status of proxy requests between Web File agents and Web servers.

In the Statistics section on the ZoneRanger dashboard, there is a chart which is specific to Web File proxy requests. As with other charts, this chart shows the last 4 hours of Web File proxy traffic based on the statistics recorded by the ZoneRanger. Specific ZoneRanger statistics are also available on the View > Statistics page when viewing the TCP Proxy service. A Status Indicator in the Activity section called Web File Proxy will light when Web File proxy requests are processed by the ZoneRanger. As with other status indicators, the light will flash as a result of Web File proxy traffic on an occasional basis and not flash the exact number of Web File proxied requests.

Figure 40-6. ZoneRanger Dashboard, Web File Proxy chart

When the Web File application is licensed on the ZoneRanger, there are several mechanisms which indicate the status of proxy requests between Web File agents and Web servers.

In the Statistics section on the ZoneRanger dashboard, there is a chart which is specific to Web File proxy requests. As with other charts, this chart shows the last 4 hours of Web File proxy traffic based on the statistics recorded by the ZoneRanger. Specific ZoneRanger statistics are also available on the View > Statistics page when viewing the TCP Proxy service. A Status Indicator in the Activity section called Web File Proxy will light when Web File proxy requests are processed by the ZoneRanger. As with other status indicators, the light will flash as a result of Web File proxy traffic on an occasional basis and not flash the exact number of Web File proxied requests.

ZoneRanger 5.6 User's Guide
Chapter 41: Generic Inbound TCP Proxy

A ZoneRanger, which does not have the Generic Inbound TCP feature licensed, does not allow a TCP connection to be initiated to it from a ZoneRanger managed device. However, TCP connections from applications such as SSH, may be initiated through a Ranger Gateway to a ZoneRanger and ultimately to a ZoneRanger managed device. This is part of the security design of a ZoneRanger.

The Generic Inbound TCP Proxy feature, when licensed on a ZoneRanger, allows TCP agent applications residing on ZoneRanger managed devices to initiate communications with TCP application servers residing within the secure area of the network through a Ranger Gateway.

In order for the ZoneRanger and Ranger Gateway to proxy TCP packets between the TCP applications, the TCP agent applications need to be configured to send requests to a ZoneRanger configured TCP port. Basically, the ZoneRanger will appear to be the TCP application server to the TCP application agents. The ZoneRanger will then proxy those packets through a Ranger Gateway to the TCP application server in the secure network. The TCP application server packets will be proxied through the Ranger Gateway and ZoneRanger back to the TCP application agents.

Figure 1. Generic Inbound TCP Proxy Open Requests
Categories

On the Configuration > Inbound Proxy page Generic TCP tab, the Add Generic TCP Category button creates a category name which is used to group like proxy rules together for logging and statistical purposes. The category name associated with a proxy rule is included in each log entry in the genericInboundTcpProxy.log file on the ZoneRanger and the inboundTcpProxy.log file on the Ranger Gateway. The category value is also used to sub-divide the Generic TCP (Inbound) usage statistics on the View Statistics page.

Category names may include any alphanumeric characters including spaces. A category may not be deleted if it is currently associated with a Proxy Rule. At least one category must be configured before any Proxy Rules can be created.

Figure 2. Generic TCP Category Configuration
Management Application Servers

A management application server is the hostname or IP address of a server on which a particular management application resides to which the ZoneRanger should proxy information coming from a ZoneRanger managed node. In this case, these would be the hostnames or IP addresses of the TCP application servers to which the TCP application agents need to communicate. One or more Ranger Gateways may be used to reach a particular management application server, in this case, TCP application server. The Ranger Gateways may be installed on the TCP application server itself or on another system. The set of Ranger Gateways needed to reach each TCP application server must be defined on the ZoneRanger. The path to each TCP application server is used when defining Proxy Rules on the Configuration > Inbound Proxy page Generic TCP tab.
Proxy Rules

The **Proxy Rules** section defines the association of TCP application agents and the local ZoneRanger TCP port to which those agents are sending requests, the category which should be used with each rule, and the destination list of management application servers (TCP application servers) to which to proxy requests. The **Source Address** must be an IP address or an address pattern. If more than one management application server is specified as a destination, ZoneRanger will attempt each destination until it can successfully proxy the request. Once ZoneRanger determines a successful management application server destination, it will continue to use that destination until a proxy request fails.

TCP requests received by a ZoneRanger, and TCP responses sent by a ZoneRanger can be written to a log file, called `/log/genericInboundTcpProxy..log`. This log can be downloaded using the `downloadFile` command on a Ranger Gateway or viewed using the **View > Service Logs** page. This can affect the performance of Generic TCP proxy. The Log Levels are:

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Logging is off</td>
</tr>
<tr>
<td>short</td>
<td>Basic information about each TCP proxy transaction request is logged, including entries for <em>Opened</em>, <em>Closed</em>, and <em>Failed</em> events.</td>
</tr>
<tr>
<td>full</td>
<td>Additional information is added to the log, including: an Opening event entry as the proxy is connecting, enhanced proxy path details with ports.</td>
</tr>
</tbody>
</table>

![Figure 3. Configure > Inbound TCP Configuration Tab](image)

ZoneRanger 5.6 User's Guide
Destinations

Each Generic TCP Proxy Rule has a set of **Destination Management Application Servers** which are paths to TCP application servers. The Management Application Server may either be a joined Ranger Gateway, as indicated by a preceding “RG:” or a path to a Management Application Server as configured on the **Configuration > Ranger Gateway** page **Mgmt App Servers** tab. The Port is the TCP port to which TCP requests will be sent.

Indicators and Statistics

Each Generic TCP Proxy Rule has a set of **Destination Management Application Servers** which are paths to TCP application servers. The Management Application Server may either be a joined Ranger Gateway, as indicated by a preceding “RG:” or a path to a Management Application Server as configured on the **Configuration > Ranger Gateway** page **Mgmt App Servers** tab. The Port is the TCP port to which TCP requests will be sent.

Figure 4. Configure > Inbound Proxy page Generic TCP tab, Destination Management Servers

Figure 5. ZoneRanger Dashboard, Generic Inbound TCP Proxy chart
When the Generic Inbound TCP feature is licensed on the ZoneRanger, there are several mechanisms which indicate the status of proxy requests between TCP agents and TCP application servers.

In the Statistics section on the ZoneRanger dashboard, there is a chart which is specific to Generic Inbound TCP proxy requests. As with other charts, this chart shows the last 4 hours of Generic Inbound TCP proxy traffic based on the statistics recorded by the ZoneRanger. Specific ZoneRanger statistics are also available on the View > Statistics page when viewing the TCP Proxy service. A Status Indicator in the Activity section called Generic Inbound TCP Proxy will light when Generic Inbound TCP proxy requests are processed by the ZoneRanger. As with other status indicators, the light will flash as a result of Generic Inbound TCP proxy traffic on an occasional basis and not flash the exact number of Generic Inbound TCP proxied requests.
Security and Generic Inbound TCP

The use of the Generic Inbound TCP feature on a ZoneRanger decreases the security profile of the ZoneRanger of protecting management applications for two reasons:

1. Any arbitrary TCP agent application can initiate a TCP connection through the ZoneRanger to the TCP application server.
2. The contents of the TCP packets are not validated.

Thus, care must be taken when using this feature. Security of the TCP application servers and of the TCP applications themselves should be reviewed so the above risks are mitigated.

Suggestions to help mitigate risk on the ZoneRanger

1. When configuring the Proxy Rules on the Configuration > Inbound Proxy page, Generic TCP tab, the source addresses should be specific as possible. This narrows the possibility of unintended source addresses from initiating TCP connections using the Proxy Rule.
2. Maintain a strict managed node policy on the ZoneRanger. Only IP addresses from managed nodes will be allowed to initiate connections. When an unmanaged IP address attempts to connect to a configured Generic Inbound TCP port, a syslog message will be generated.
3. Use the Configuration > Whitelist page to only allow configured IP addresses to send data or initiate connections with the ZoneRanger. The use of a whitelist is a very good mechanism to closely define the IP addresses which interact with the ZoneRanger.
4. Where possible, use non-standard TCP ports for incoming connections. Non-standard TCP ports are less noticeable to the casual observer.
Appendices

A. SNMP Agent

ZoneRanger provides specific system and traffic information via its SNMP agent. The MIB can be found in the ZRCustom directory on the Ranger Gateway installation as the file ZONERANGER-AGENT.mib.

-- This is the MIB for Tavve Software Co.'s ZoneRanger agent.
-- Copyright 2007-2009 Tavve Software Co. All Rights Reserved.
-- Reproduction of this document is authorized on
-- condition that this copyright notice is included.
-- This MIB document embodies Tavve Software Co.'s proprietary
-- intellectual property. Tavve Software Co. retains all
-- title and ownership in this MIB, including any
-- revisions.
--
-- It is Tavve Software Co.'s intent to encourage the widespread
-- use of this MIB in connection with the management of
-- Tavve Software Co. products. Tavve Software Co. grants vendors,
-- end-users, and other interested parties a non-exclusive
-- license to use this MIB in connection with the
-- management of Tavve Software Co. products.
--
-- This MIB document is supplied "AS IS," and Tavve Software Co.
-- makes no warranty, either express or implied, as to the
-- use operation, condition, or performance of the MIB.

ZONERANGER-AGENT-MIB DEFINITIONS ::= BEGIN

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, enterprises, Integer32, Counter32 FROM SNMPv2-SMI
  DisplayString, DateAndTime, TEXTUAL-CONVENTION FROM SNMPv2-TC
  MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF;

tscZRAgent MODULE-IDENTITY
  LAST-UPDATED "200903160000Z"
  ORGANIZATION "Tavve Software"
  CONTACT-INFO
    "postal:  One Copley Parkway
              Suite 480
              Morrisville, NC 27560"
    phone: 919 460 1489
    email: support@tavve.com
    web: http://www.tavve.com"
  DESCRIPTION
    "This MIB module defines a MIB which provides mechanisms
to query information from a ZoneRanger."
  REVISION "200802010000Z"
  DESCRIPTION
    "First revision."
  REVISION "200903160000Z"
  DESCRIPTION
    "Added percentage CPU utilization."
::= { tscMibs 16}

  tavve OBJECT IDENTIFIER ::= { enterprises 2668 }
  tscMibs OBJECT IDENTIFIER ::= { tavve 2 }
  tscZRObjects OBJECT IDENTIFIER ::= { tscZRAgent 1 }
  tscZRConformance OBJECT IDENTIFIER ::= { tscZRAgent 2 }
-- Textual conventions

KBytes ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d"
STATUS current
DESCRIPTION "Storage size, expressed in units of 1024 bytes."
SYNTAX Integer32 (0..2147483647)

-- Objects

tscZRInformation OBJECT IDENTIFIER ::= { tscZRObjects 1 }
tscZRMessagingStats OBJECT IDENTIFIER ::= { tscZRObjects 2 }
tscZRForwardStats OBJECT IDENTIFIER ::= { tscZRObjects 3 }
tscZRSnmpProxyStats OBJECT IDENTIFIER ::= { tscZRObjects 4 }
tscZRIcmpProxyStats OBJECT IDENTIFIER ::= { tscZRObjects 5 }

tscZRVersion OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A textual description of the ZoneRanger software version. This value should include the numeric version as well as SP level if set."
::= { tscZRInformation 1 }

tscZRModel OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A textual description of the ZoneRanger model. For example, ZR-200, ZR-SPX, etc."
::= { tscZRInformation 2 }

tscZRManagedNodes OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of managed nodes."
::= { tscZRInformation 3 }

tscZRManagedRemaining OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of managed nodes remaining."
::= { tscZRInformation 4 }

tscZRCPUsage OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The number of centi-seconds of the total system's CPU resources consumed."
::= { tscZRInformation 5 }

tscZRDiskUsage OBJECT-TYPE
SYNTAX Integer32 (0..100)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The percentage of disk capacity that is currently in use
rounded to the nearest integer."
::= { tscZRInformation 6 }

tscZRLastStartTime OBJECT-TYPE
SYNTAX      DateAndTime
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The last time that the ZoneRanger software started."
::= { tscZRInformation 7 }

tscZRAppServerFreeMemory OBJECT-TYPE
SYNTAX      KBytes
UNITS       "KB"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The amount of free JVM memory in kilobytes."
::= { tscZRInformation 8 }

tscZRFreeMemory OBJECT-TYPE
SYNTAX      KBytes
UNITS       "KB"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"The amount of free system memory in kilobytes."
::= { tscZRInformation 9 }

tscZRPatchStatusTable OBJECT-TYPE
SYNTAX      SEQUENCE OF TscZRPatchStatusEntryEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"This (conceptual) table contains a list of applied patches."
::= { tscZRInformation 10 }

tscZRPatchStatusEntry OBJECT-TYPE
SYNTAX      TscZRPatchStatusEntryEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"Entry for tscZRPatchStatusTable."
INDEX       { tscZRPatchStatusIndex }
::= { tscZRPatchStatusTable 1 }

TscZRPatchStatusEntryEntry ::= SEQUENCE {
    tscZRPatchStatusIndex  Integer32,
    tscZRPatchStatusName   DisplayString
}

tscZRPatchStatusIndex OBJECT-TYPE
SYNTAX      Integer32 (1..2147483647)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"The index of a particular patch status. They are indexed
from 1, oldest to newest."
::= { tscZRPatchStatusEntry 1 }

ZoneRanger 5.6 User's Guide 374
tscZRPatchStatusName OBJECT-TYPE
SYNTAX       DisplayString
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "A textual name of the patch."
::= { tscZRPatchStatusEntry 2 }

tscZRPercentCpuUsage OBJECT-TYPE
SYNTAX       Integer32 (0..100)
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "The percentage of CPU utilization that was recently in use rounded to the nearest integer. For more accurate information, use tscZRCpuUsage instead."
::= { tscZRInformation 11 }

tscZRRangerGatewayTable OBJECT-TYPE
SYNTAX       SEQUENCE OF TscZRRangerGatewayEntryEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION  "This (conceptual) table contains a list of Ranger Gateways."
::= { tscZRInformation 100 }

tscZRRangerGatewayTable OBJECT-TYPE
SYNTAX       SEQUENCE OF TscZRRangerGatewayEntryEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION  "This (conceptual) table contains a list of Ranger Gateways."
::= { tscZRInformation 100 }

tscZRRangerGatewayAddress OBJECT-TYPE
SYNTAX       IpAddress
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION  "The IP address of a particular Ranger Gateway."
::= { tscZRRangerGatewayEntry 1 }

tscZRRangerGatewayName OBJECT-TYPE
SYNTAX       DisplayString
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "The name of a particular Ranger Gateway."
::= { tscZRRangerGatewayEntry 2 }

tscZRRangerGatewayConnectionStatus OBJECT-TYPE
SYNTAX       INTEGER {
    up(1),
    down(2),
    unknown(3)
  }
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "The connection status of a particular Ranger Gateway."
::= { tscZRRangerGatewayEntry 4 }

tscZRMessagesExternalSent OBJECT-TYPE
SYNTAX       Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of messages sent off box since the last restart."
::= { tscZRMessagingStats 1 }

tscZRMessagesExternalReceived OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of messages received off box since the last restart."
::= { tscZRMessagingStats 2 }

tscZRMessagesDiscarded OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of messages discarded since the last restart."
::= { tscZRMessagingStats 3 }

tscZRForwardStatsTable OBJECT-TYPE
SYNTAX SEQUENCE OF TscZRForwardStatsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This (conceptual) table contains a list of forwarded UDP data."
::= { tscZRForwardStatsTable 1 }

tscZRForwardStatsEntry OBJECT-TYPE
SYNTAX TscZRForwardStatsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Entry for tscZRForwardStatsTable."
INDEX { tscZRForwardStatsProtocol }
::= { tscZRForwardStatsTable 1 }

TscZRForwardStatsEntry ::= SEQUENCE {
  tscZRForwardStatsProtocol Integer32,
  tscZRForwardStatsName DisplayString,
  tscZRForwardStatsCount Counter32
}

tscZRForwardStatsProtocol OBJECT-TYPE
SYNTAX INTEGER {
  generic(1),
  trap(2),
  syslog(3),
  sflow(4),
  netflow(5)
}
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The protocol filter type."
::= { tscZRForwardStatsEntry 1 }

tscZRForwardStatsName OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"A descriptive label for the protocol type."
::= { tscZRForwardStatsEntry 2 }

tscZRForwardStatsCount OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of forwarded datagrams. This may be more than
the number of received datagrams, as a single one may be
forwarded to multiple destinations."
::= { tscZRForwardStatsEntry 3 }

tscZRSnmpProxyRequests OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of SNMP proxy requests."
::= { tscZRSnmpProxyStats 1 }

tscZRSnmpProxyResponses OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of SNMP proxy responses."
::= { tscZRSnmpProxyStats 2 }

tscZRSnmpProxyDiscards OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of SNMP proxy requests discarded. One possible reason
is requests to unmanaged devices."
::= { tscZRSnmpProxyStats 3 }

tscZRIcmpProxyRequests OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of ICMP proxy requests."
::= { tscZRIcmpProxyStats 1 }

tscZRIcmpProxyResponses OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of ICMP proxy responses."
::= { tscZRIcmpProxyStats 2 }

tscZRIcmpProxyDiscards OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The count of ICMP proxy requests discarded. One possible reason
is requests to unmanaged devices."
::= { tscZRIcmpProxyStats 3 }

-- Conformance
tscZRCompliances OBJECT IDENTIFIER ::= { tscZRConformance 1 }
tscZRGroups OBJECT IDENTIFIER ::= { tscZRConformance 2 }

tscZRCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION "The compliance statement for the ZoneRanger Agent MIB."
  MODULE -- this module
  MANDATORY-GROUPS {
    tscZRInformationGroup, tscZRMessagingGroup, tscZRForwardingGroup, tscZRSnmpProxyGroup, tscZRIcmpProxyGroup
  }
  ::= { tscZRCompliances 1 }

tscZRInformationGroup OBJECT-GROUP
  OBJECTS {
    tscZRVersion, tscZRModel, tscZRManagedNodes, tscZRManagedRemaining,
    tscZRCpuUsage, tscZRDiskUsage, tscZRLastStartTime,
    tscZRApServerFreeMemory, tscZRFreeMemory, tscZRPatchStatusName
  }
  STATUS current
  DESCRIPTION "The ZoneRanger Information Group."
  ::= { tscZRGroups 1 }

tscZRMessagingGroup OBJECT-GROUP
  OBJECTS {
    tscZRMessagesDiscarded, tscZRMessagesExternalReceived, tscZRMessagesExternalSent
  }
  STATUS current
  DESCRIPTION "The ZoneRanger Messaging Group."
  ::= { tscZRGroups 2 }

tscZRForwardingGroup OBJECT-GROUP
  OBJECTS {
    tscZRForwardStatsName, tscZRForwardStatsCount
  }
  STATUS current
  DESCRIPTION "The ZoneRanger UDP Forwarding Group."
  ::= { tscZRGroups 3 }

tscZRSnmpProxyGroup OBJECT-GROUP
  OBJECTS {
    tscZRSnmpProxyRequests, tscZRSnmpProxyResponses, tscZRSnmpProxyDiscards
  }
  STATUS current
  DESCRIPTION "The ZoneRanger SNMP Proxy Group."
  ::= { tscZRGroups 4 }

tscZRIcmpProxyGroup OBJECT-GROUP
  OBJECTS {
    tscZRIcmpProxyRequests, tscZRIcmpProxyResponses, tscZRIcmpProxyDiscards
  }
  STATUS current
  DESCRIPTION "The ZoneRanger ICMP Proxy Group."
  ::= { tscZRGroups 5 }

END

ZoneRanger 5.6 User's Guide
B. ZoneRanger and Ranger Gateway Traps

ZoneRanger and Ranger Gateway generate SNMP traps to indicate changes to managed devices as well as the ZoneRanger and Ranger Gateway itself.

Traps are defined in the file `tavve.mib` which is located in the ZRCustom directory on the Ranger Gateway.

### Root cause traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tscZRInferredDown</td>
<td>Sent after ZoneRanger verifies that a device is down and is not correlated to a network problem.</td>
</tr>
<tr>
<td>tscZRInferredUp</td>
<td>Complements tscZRInferredDown to report that a root cause node that was down is now up.</td>
</tr>
<tr>
<td>tscZRSourceDown</td>
<td>Sent after ZoneRanger verifies that a device is down and is correlated to a network problem.</td>
</tr>
<tr>
<td>tscZRSourceUp</td>
<td>Complements tscZRSourceDown to report that a source node that was down is now up.</td>
</tr>
<tr>
<td>tscZRVerifyDown</td>
<td>Sent after ZoneRanger reports that a root cause node is down.</td>
</tr>
<tr>
<td>tscZRVerifyUp</td>
<td>Sent after ZoneRanger reports that a device is again up after being verified down.</td>
</tr>
</tbody>
</table>

### Test trap

<table>
<thead>
<tr>
<th>Trap</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tscZRTestTrap</td>
<td>Sent by ZoneRanger to verify that ZoneRanger can receive and send traps.</td>
</tr>
</tbody>
</table>

### Syslog traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tscZRCtSeverity0</td>
<td>A Cisco device reported a severity 0 event</td>
</tr>
<tr>
<td>tscZRCtSeverity1</td>
<td>A Cisco device reported a severity 1 event</td>
</tr>
<tr>
<td>tscZRCtSeverity2</td>
<td>A Cisco device reported a severity 2 event</td>
</tr>
<tr>
<td>tscZRCtSeverity3</td>
<td>A Cisco device reported a severity 3 event</td>
</tr>
<tr>
<td>tscZRCtSeverity4</td>
<td>A Cisco device reported a severity 4 event</td>
</tr>
<tr>
<td>tscZRCtSeverity5</td>
<td>A Cisco device reported a severity 5 event</td>
</tr>
<tr>
<td>tscZRCtSeverity6</td>
<td>A Cisco device reported a severity 6 event</td>
</tr>
<tr>
<td>tscZRCtSeverity7</td>
<td>A Cisco device reported a severity 7 event</td>
</tr>
<tr>
<td>tscZRCtSeverity8</td>
<td>A Cisco device reported a severity 8 event</td>
</tr>
<tr>
<td>tscZRSltTrap</td>
<td>Syslog message encapsulated as an SNMP trap</td>
</tr>
</tbody>
</table>
### Polling status traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcsZRTcpRefused</td>
<td>The TCP service refused a connection attempt.</td>
</tr>
<tr>
<td>tscRedZRIfDown</td>
<td>Sent after ZoneRanger determines that a redundant interface is no longer reachable.</td>
</tr>
<tr>
<td>tscRedZRIfUnknown</td>
<td>Sent after ZoneRanger determines that a redundant interface is no longer known.</td>
</tr>
<tr>
<td>tscRedZRIfUp</td>
<td>Sent after ZoneRanger determines that a redundant interface is reachable.</td>
</tr>
<tr>
<td>tscZRIfDown</td>
<td>Sent after ZoneRanger determines that an interface is no longer reachable.</td>
</tr>
<tr>
<td>tscZRIfUnknown</td>
<td>Sent after ZoneRanger determines that an interface is no longer known.</td>
</tr>
<tr>
<td>tscZRIfUp</td>
<td>Sent after ZoneRanger determines that an interface is reachable.</td>
</tr>
<tr>
<td>tscZRNodeDown</td>
<td>Sent after ZoneRanger determines that all interfaces on a node are down.</td>
</tr>
<tr>
<td>tscZRNodeMarginal</td>
<td>Sent after ZoneRanger determines that some interfaces on the node are down and some interfaces on the node are up.</td>
</tr>
<tr>
<td>tscZRNodeUnknown</td>
<td>Sent after ZoneRanger determines that all interfaces on a node are unknown.</td>
</tr>
<tr>
<td>tscZRNodeUp</td>
<td>Sent after ZoneRanger determines that all interfaces on a node are up.</td>
</tr>
<tr>
<td>tscZRTcpBusy</td>
<td>The TCP service is busy during a connection attempt.</td>
</tr>
<tr>
<td>tscZRTcpRefused</td>
<td>The TCP service refused the connection attempt.</td>
</tr>
<tr>
<td>tscZRTcpTimeout</td>
<td>The TCP service timed out a connection attempt.</td>
</tr>
<tr>
<td>tscZRTcpUnknown</td>
<td>The TCP service connection attempt status is unknown; possible network error</td>
</tr>
<tr>
<td>tscZRTcpUp</td>
<td>The TCP service is up and receiving connections.</td>
</tr>
</tbody>
</table>

### Ranger Gateway status traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tscRGCommDown</td>
<td>Sent by Ranger Gateway to report that a Ranger Gateway lost communication with a joined ZoneRanger</td>
</tr>
</tbody>
</table>
### Configuration change traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tscZHRHostnameChanged</td>
<td>Sent by ZoneRanger to report that it changed the reported hostname</td>
</tr>
<tr>
<td>tscZRInterfaceAdded</td>
<td>Sent by ZoneRanger to report that it added the reported interface</td>
</tr>
<tr>
<td>tscZRInterfaceDeleted</td>
<td>Sent by ZoneRanger to report that it deleted the reported interface</td>
</tr>
<tr>
<td>tscZRIPAddressAdded</td>
<td>Sent by ZoneRanger to report that it added the reported IP address.</td>
</tr>
<tr>
<td>tscZRIPAddressDeleted</td>
<td>Sent by ZoneRanger to report that it deleted the reported IP address.</td>
</tr>
<tr>
<td>tscZRNNodeAdded</td>
<td>Sent by ZoneRanger to report that it added the reported node</td>
</tr>
<tr>
<td>tscZRNNodeDeleted</td>
<td>Sent by ZoneRanger to report that it deleted the reported node</td>
</tr>
<tr>
<td>tscZRNNodeMerged</td>
<td>Sent by ZoneRanger to report that it merged two hostnames</td>
</tr>
<tr>
<td>tscZRSysContactChanged</td>
<td>Sent by ZoneRanger to report that it changed the reported SNMP sysContact</td>
</tr>
<tr>
<td>tscZRSysLocationChanged</td>
<td>Sent by ZoneRanger to report that it changed the reported SNMP sysLocation</td>
</tr>
<tr>
<td>tscZRSysNameChanged</td>
<td>Sent by ZoneRanger to report that it changed the reported SNMP sysName.</td>
</tr>
<tr>
<td>tscZRSysOIDChanged</td>
<td>Sent by ZoneRanger to report that it changed the reported SNMP sysObjectId</td>
</tr>
<tr>
<td>tscZRTcpPortAdded</td>
<td>Sent by ZoneRanger to report that it added the reported TCP port.</td>
</tr>
<tr>
<td>tscZRTcpPortDeleted</td>
<td>Sent by ZoneRanger to report that it deleted the reported TCP port.</td>
</tr>
</tbody>
</table>
## Audit Traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tscCertificateHasExpired</td>
<td>A certificate on the ZoneRanger or Ranger Gateway has expired.</td>
</tr>
<tr>
<td>tscCertificateIsExpiring</td>
<td>A certificate on the ZoneRanger or Ranger Gateway is expiring soon.</td>
</tr>
<tr>
<td>tscCertificateIsNotTrusted</td>
<td>A certificate on the ZoneRanger or Ranger Gateway does not have valid Trusted Certificate Authority.</td>
</tr>
<tr>
<td>tscCertificateIsNotYetValid</td>
<td>A certificate on the ZoneRanger or Ranger Gateway has a start date in the future.</td>
</tr>
<tr>
<td>tscDataDiodeFileTransferError</td>
<td>The ZoneRanger could not communicate with the Data Diode.</td>
</tr>
<tr>
<td>tscDataDiodeInvalidData</td>
<td>The ZoneRanger discarded a Data Diode file due to invalid data.</td>
</tr>
<tr>
<td>tscDataDiodeMountError</td>
<td>The ZoneRanger could not communicate with the Data Diode.</td>
</tr>
<tr>
<td>tscDataDiodeNoActivityTrap</td>
<td>The ZoneRanger detected no activity on the Data Diode.</td>
</tr>
<tr>
<td>tscDataDiodeReportTransferError</td>
<td>The ZoneRanger detected no activity on the Data Diode.</td>
</tr>
<tr>
<td>tscDataDiodeSubtendingDuplicateID</td>
<td>The ZoneRanger detected a duplicate subtending ZoneRanger ID as itself.</td>
</tr>
<tr>
<td>tscDataDiodeSubtendingDuplicateTrap</td>
<td>The ZoneRanger detected a duplicate subtending ZoneRanger ID in the chain.</td>
</tr>
<tr>
<td>tscDataDiodeSubtendingVmActivationExpiring</td>
<td>The ZoneRanger detected a subtending ZoneRanger activation that will soon expire.</td>
</tr>
<tr>
<td>tscDataDiodeSubtendingVmNotActivated</td>
<td>The ZoneRanger detected a subtending ZoneRanger that is not activated.</td>
</tr>
<tr>
<td>tscInitialProfileLoadError</td>
<td>An error while loading a profile after initial configuration</td>
</tr>
<tr>
<td>tscInternalError</td>
<td>An internal error occurred</td>
</tr>
<tr>
<td>tscJoinedPingFailed</td>
<td>A joined ZoneRanger and Ranger Gateway cannot communicate</td>
</tr>
<tr>
<td>tscJoinedVersionInvalid</td>
<td>The ZoneRanger version does not match the reporting Ranger Gateway, or the Ranger Gateway version does not match reporting ZoneRanger.</td>
</tr>
<tr>
<td>Trap</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>tscJoinFailed</td>
<td>The ZoneRanger and Ranger Gateway failed to join.</td>
</tr>
<tr>
<td>tscMessageQueueIsFull</td>
<td>A queue on the ZoneRanger or Ranger Gateway is full</td>
</tr>
<tr>
<td>tscMessagesDiscarded</td>
<td>The ZoneRanger or Ranger Gateway discarded messages</td>
</tr>
<tr>
<td>tscMessagesDiscardedMaxQueueTimeExceeded</td>
<td>The ZoneRanger or Ranger Gateway discarded messages because the queue time was exceeded</td>
</tr>
<tr>
<td>tscMessagesDiscardedQueueFull</td>
<td>The ZoneRanger or Ranger Gateway discarded messages</td>
</tr>
<tr>
<td>tscMessagingProtocolViolation</td>
<td>The ZoneRanger or Ranger Gateway detected a protocol violation.</td>
</tr>
<tr>
<td>tscMessagingSecurityViolation</td>
<td>The ZoneRanger or Ranger Gateway detected a security violation</td>
</tr>
<tr>
<td>tscNotJoined</td>
<td>The ZoneRanger behaves as if it is not joined to the reporting Ranger Gateway, or the Ranger Gateway behaves as if it is not joined to the reporting ZoneRanger</td>
</tr>
<tr>
<td>tscRedundantPeerNotReporting</td>
<td>A redundant peer of a ZoneRanger is not responding to pings.</td>
</tr>
<tr>
<td>tscRGForwardError</td>
<td>Sent by ZoneRanger to report that a Ranger Gateway failed to forward a packet</td>
</tr>
<tr>
<td>tscServiceDegraded</td>
<td>A ZoneRanger or Ranger Gateway service is degraded</td>
</tr>
<tr>
<td>tscServiceFailed</td>
<td>A ZoneRanger or Ranger Gateway service failed</td>
</tr>
<tr>
<td>tscSnmpProtocolViolation</td>
<td>The ZoneRanger detected SNMP protocol violations.</td>
</tr>
<tr>
<td>tscThresholdExceeded</td>
<td>A system threshold limit was exceeded.</td>
</tr>
<tr>
<td>tscTrafficIpThresholdExceeded</td>
<td>The ZoneRanger determined a traffic threshold was exceeded for a particular IP address</td>
</tr>
<tr>
<td>tscTrafficThresholdExceeded</td>
<td>The ZoneRanger or Ranger Gateway determined a traffic threshold was exceeded</td>
</tr>
<tr>
<td>tscUnauthorizedMessageClient</td>
<td>The ZoneRanger or Ranger Gateway discarded messages from an unauthorized client</td>
</tr>
<tr>
<td>tscUnauthorizedMessageDest</td>
<td>The ZoneRanger or Ranger Gateway discarded messages to an unauthorized destination</td>
</tr>
<tr>
<td>tscUnauthorizedMessageSrc</td>
<td>The ZoneRanger or Ranger Gateway discarded messages from an unauthorized source</td>
</tr>
<tr>
<td>tscVmActivationExpiring</td>
<td>The ZoneRanger activation will soon expire.</td>
</tr>
<tr>
<td>Trap</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>tscVMNotActivated</td>
<td>The ZoneRanger is not activated.</td>
</tr>
</tbody>
</table>
C. SOCKS

SOCKS is an Internet standards-track protocol for generic TCP and UDP proxy services, defined in RFC 1928. SOCKS can be used to redirect management traffic from the management application to a SOCKS server integrated within the Ranger Gateway. In order to use SOCKS, either the management application must include built-in support for SOCKS, or generic SOCKS “shim” software must be installed on the management application server. The shim software inserts itself between the management application and the server’s TCP/IP stack, and redirects traffic for specified IP addresses and ports to a SOCKS server, based on configuration information.

The two most prevalent versions of the SOCKS protocol are Version 4 and Version 5. SOCKS v4 and SOCKS v5 both support the ability for a SOCKS client to request a TCP connection to a target device. The SOCKS v5 protocol also defines two additional features:

- The ability for a SOCKS client to send UDP datagrams and receive associated responses.
- The ability for a SOCKS client to bind a port and receive incoming TCP connections.

The SOCKS server in the Ranger Gateway supports only the ability for a SOCKS client to originate TCP connections to managed devices or joined ZoneRangers and the ability to send or receive UDP datagrams, but does not support the ability for a SOCKS client to receive incoming TCP connections.

A significant advantage of SOCKS is that it provides a mechanism for applications running on one server to use the services of a Ranger Gateway installed on a different server. The SOCKS server on the Ranger Gateway currently does not support client authentication, but Proxy Access Control can be used to limit the set of servers that are allowed to use the proxy services provided by a Ranger Gateway and its joined ZoneRangers. While SOCKS can be very useful for certain applications, such as SSH proxy, its overall usefulness tends to be somewhat limited given the number of prevalent management applications that do not provide built-in support. SOCKS shims can be used as an alternative in such cases, especially when the management application is installed on a Windows operating system, but it can be difficult to find a reliable, fully-featured SOCKS shim for certain other operating systems.

Establishing a TCP connection using SOCKS proxy

The process to establish a connection to a managed device using SOCKS proxy is as follows:

1. A SOCKS-aware client application (or SOCKS shim) establishes a TCP connection to the SOCKS port on the Ranger Gateway (the default is 4855).
2. After the connection is established, the client application sends a SOCKS connect request to the Ranger Gateway, indicating the target device and port.
3. The SOCKS server on the Ranger Gateway identifies the source address, destination address, transport (i.e. TCP, in this case) and destination port associated with the connection request, and uses the Proxy Access Control tables to determine whether the request should be allowed, and if so, what protocol is expected (e.g. for validation, or special processing), and what port translation rule, if any, should be applied before presenting the request to the target device.
4. If the request is allowed, the SOCKS server on the Ranger Gateway consults the Proxy Map service to identify a ZoneRanger that is able to proxy traffic to the target device, and to translate the target address to the address that the ZoneRanger must use to access the target device if NAT is in effect, then forwards the connection request to the selected ZoneRanger.

---

Most Telnet/SSH client applications and web browsers do provide built-in support for SOCKS. Most of the more specialized management applications do not.
5. The selected ZoneRanger attempts to establish a TCP connection to the target device. If successful, the ZoneRanger informs the SOCKS server on the Ranger Gateway.

6. The SOCKS server on the Ranger Gateway sends a response message to the client application, indicating that the connect request was successful.

7. From this point on, the Ranger Gateway and selected ZoneRanger will relay data between the client application’s TCP connection to the Ranger Gateway and the ZoneRanger’s TCP connection to the target device, until one of the ends of the connection is disconnected.

Performing a UDP Protocol Transaction using SOCKS proxy

The process to perform a UDP protocol transaction (e.g. an SNMP Get Request/Response) using SOCKS proxy is as follows:

1. A SOCKS-aware client application (or SOCKS shim) establishes a TCP connection to the SOCKS port on the Ranger Gateway (the default is 4855).

2. After the connection is established, the client application sends a SOCKS UDP Associate request to the Ranger Gateway, optionally specifying the source address and port that it will use when sending UDP datagrams to the SOCKS server.

3. The SOCKS server replies to the UDP Associate message, indicating the address and port to which the SOCKS-aware client should send datagrams that are to be relayed through the SOCKS server.

4. When the client has a datagram (e.g. an SNMP Get Request) to send to a managed device, it prepends a SOCKS header indicating the target device address and port, and sends the resulting datagram to the address and port that was indicated by the SOCKS server in the previous step.

5. The SOCKS server receives the datagram, identifies the source address, destination address, transport (i.e. UDP, in this case) and destination port associated with the datagram, and uses the Proxy Access Control tables to determine whether the datagram should be forwarded to a managed device, and if so, what protocol is expected (e.g. for validation, or special processing), and what port translation rule, if any, should be applied before presenting the request to the target device.

6. If the request is allowed, the SOCKS server consults the Proxy Map service to identify a ZoneRanger that is able to proxy traffic to the target device, and to translate the target address to the address that the ZoneRanger must use to access the target device if NAT is in effect, removes the prepended header, then forwards the request to the selected ZoneRanger.

7. The selected ZoneRanger will forward the request to the target device. If the target device replies, the ZoneRanger will relay the reply to the SOCKS server on the Ranger Gateway.

8. The SOCKS server will prepend the address and port corresponding to the target device to the reply datagram, then will forward the resulting datagram to the client application.

9. At this point the UDP transaction is complete. The client application can continue to use the UDP association that was created in steps 2 and 3 for additional transactions as long as the TCP connection that was created in step 1 remains established. When the TCP connection is cleared, the SOCKS server will automatically remove the UDP association.

SOCKS Server Configuration

The SOCKS port on the Ranger Gateway can be specified or changed at any time using the configGateway command, or the Configure Gateway Settings dialog on the Ranger Gateway Viewer. The default port is 4855.
D. IP Address Aliasing

IP address aliasing is one of a number of alternative mechanisms (e.g. GVI, SOCKS) that can be used to enable the Ranger Gateway to intercept management protocol traffic originated by management applications and destined for managed devices.

Most operating systems provide a means to associate multiple IP addresses with each network interface (i.e. a primary address, and one or more “aliases”). If IP address aliases, corresponding to managed devices located in firewall-partitioned networks, are defined on the management application server, and the Ranger Gateway is configured to listen on a variety of ports for TCP or UDP traffic destined for any of these IP addresses, all traffic generated by the management application and destined for these devices on the configured ports will be received by the Ranger Gateway.

If the management application and the Ranger Gateway software have been installed on the same server, the IP address aliases can usually be added to the server’s loopback interface.

For example consider the network shown in the following figure:

![Figure D-1. IP Address Aliasing](image)

In this network, the management application is managing devices in two DMZs via a Ranger Gateway and a set of three ZoneRangers. There are five devices to be managed in all: 10.2.1.1, 10.2.1.2, 10.4.1.1, 10.4.1.2, and 10.4.1.3. In order to enable the Ranger Gateway to intercept traffic destined for these devices, five IP address aliases are defined on the management application server. The addresses in this case are identical to the actual IP addresses of the managed devices.
If the management application and the Ranger Gateway software have been installed on different servers, the IP address aliases must be added to an appropriate network interface on the Ranger Gateway server, and static routes will need to be defined on the management application server to ensure that SNMP requests are routed to the Ranger Gateway server.

For example consider the network shown in the following figure:

Figure D-2. Proxy Map using IP Address Aliasing

In this network, the management application is managing devices in two DMZs via a Ranger Gateway and a set of three ZoneRangers. The Ranger Gateway and management application are installed on different servers. There are five devices to be managed in all: 10.2.1.1, 10.2.1.2, 10.4.1.1, 10.4.1.2, and 10.4.1.3. In order to enable the Ranger Gateway to intercept traffic destined for these devices, five IP address aliases are defined on the Ranger Gateway application server: 10.10.1.21, 10.10.1.22, 10.10.1.23, 10.10.1.24, and 10.10.1.25. The management application server is configured with static routes so that all traffic destined for the alias addresses will be routed to the Ranger Gateway server (10.1.1.2). The Proxy Map in the Ranger Gateway is configured to translate the alias addresses to the actual device addresses. Note that the management application server routing table in the figure above could be simplified by configuring a single subnet route (10.10.1.0/24 - 10.2.1.2), provided that the there are no devices with addresses in the specified subnet that need to be routed normally (e.g. to the management application server’s default gateway). In general, alias addresses should be chosen so as to avoid confusion with actual device addresses.
It should be noted that in the case where the management application and the Ranger Gateway are installed in different servers, the need for static routing rules in the management application server can typically be eliminated if the IP addresses alias values lie within the same subnet as the management application server and Ranger Gateway server. From the example above, assuming that management application server IP address and the primary address associated with the Ranger Gateway are both in the 10.1.1.0/24 subnet, if sufficient unused addresses in this subnet could be found, these addresses could be used as the alias addresses.

A major disadvantage of the IP address aliasing technique is the administrative effort required to add and maintain IP address aliases for all managed devices on the Ranger Gateway server, as well as any corresponding static routing rules in management servers, where applicable. Another concern is that operating systems may limit the number of IP address aliases that can be defined. As a result, this technique may not be able to support the required number of managed devices for some applications. Lastly, the number of proxy protocols that are supported by this technique is fairly limited (i.e. SNMP and SSH).
E. SSL Communications between ZoneRanger and Ranger Gateway

Communication between joined ZoneRangers and Ranger Gateways, and redundant ZoneRangers is secured using the Secure Sockets Layer (SSL) protocol. SSL provides both encryption and authentication. At the beginning of each SSL session both parties involved in the session authenticate each other by exchanging SSL certificates. In order for the session to be established, each party (i.e. a ZoneRanger or a Ranger Gateway) must validate the other party's SSL certificate based on the following criteria:

- The SSL certificate presented by the remote party must have been signed by a certificate authority that the receiving party is configured to trust.
- The distinguished name associated with the SSL certificate presented by the remote party must identify a subject/entity that the receiving party is configured to trust.

By default, each ZoneRanger is configured with a certificate issued by the Tavve internal certificate authority, with the following distinguished name:

CN=ZoneRanger,OU=Engineering,O=Tavve,L=Morrisville,ST=North Carolina,C=US

Similarly, each Ranger Gateway is configured with a certificate with the following distinguished name:

CN=RangerGateway,OU=Engineering,O=Tavve,L=Morrisville,ST=North Carolina,C=US

The given distinguished names essentially identify two subjects: the generic ZoneRanger subject and the generic RangerGateway subject. ZoneRangers are configured, by default, to allow communication with both subjects, in order to support communication with joined Ranger Gateways, and with redundant peers. Ranger Gateways are configured only to allow communication with the ZoneRanger subject.

This initial SSL configuration is provided so that ZoneRangers and Ranger Gateways are able to communicate right out of the box. In environments where a high degree of security is required, it is recommended that the Ranger Gateways and ZoneRangers be reconfigured to use customer-specific certificates. The process to replace the Tavve Messaging SSL configuration for both the ZoneRanger and Ranger Gateway with customer specific security credentials is as follows:

1. Replace ZoneRanger Messaging SSL Certificate

   Using the Administration > SSL Certificate page Messaging tab on the ZoneRanger, install the new public key certificate and private key specific to your security environment. The SSL Certificate can be in PKCS #12, X509, or Keystore format. If a problem occurs, the original Tavve SSL certificate may be restored.

2. Replace Ranger Gateway SSL Certificate

   Using the configSSLcommand on the Ranger Gateway, install the new public key certificate and private key specific to your security environment. The SSL certificate can be in PKCS #12, X509, or Keystore format. If a problem occurs, the original Tavve SSL certificate may be restored.

3. Update ZoneRanger Certificate Authorities and Trusted Subjects
Using the **Configuration > Ranger Gateway** page **SSL Trust** tab on the ZoneRanger, first add the distinguished name identified in the SSL certificate which was installed on the Ranger Gateway by using the **Add Subject** button\(^{14}\). The default Subjects may be removed if desired.

Second, add the Certificate Authority which authorized the SSL certificate which was installed on the Ranger Gateway if it is not already specified in the Trusted Certificate Authorities table. The Certificate Authority may be added from a file in X509 or JKS Keystore format by using the **Add Certificate** button. The other Trusted Certificate Authorities may be removed if desired. If a problem occurs, the original Tavve trusted certificate authorities may be restored.

### 4. Update Ranger Gateway Certificate Authorities and Trusted Subjects

Using the **trustSSL** command on the Ranger Gateway, first add the distinguished name identified in the SSL certificate which was installed on the ZoneRanger by using the **Add trusted subject** option. The default Subjects may be removed if desired.

Second, add the Certificate Authority which authorized the SSL certificate which was installed on the ZoneRanger if it is not already listed under the List trusted certificate authorities option. The Certificate Authority may be added from a file in X509 or JKS Keystore format using the **Add trusted certificate authorities** option. The other Trusted Certificate Authorities may be removed if desired. If a problem occurs, the original Tavve trusted certificate authorities may be restored.

\(^{14}\)In the terminology of SSL certificates, a **distinguished name** is used to identify a **subject** or entity. The Ranger Gateway and ZoneRanger user interfaces do not differentiate between a subject and a subject's distinguished name. As such, when configuring a list of trusted subjects, the values that are entered are in fact the distinguished names of the subjects that are to be trusted.
F. SSL Communications between ZoneRanger's Web Server and Web Browsers

Communication between the ZoneRanger's web server and web browsers is secured using the Secure Sockets Layer (SSL) protocol. SSL provides both encryption and authentication. At the beginning of each SSL session both parties involved in the session authenticate each other by exchanging SSL certificates. In order for the session to be established, each party (i.e. a ZoneRanger or a web browser) must validate the other party's SSL certificate based on the following criteria:

- The SSL certificate presented by the remote party must have been signed by a certificate authority that the receiving party is configured to trust.
- The distinguished name associated with the SSL certificate presented by the remote party must identify a subject/entity that the receiving party is configured to trust.

By default, each ZoneRanger's web server is configured with a certificate issued by the Tavve internal certificate authority, with the following distinguished name:

```
CN=ZoneRanger,OU=Engineering,O=Tavve,L=Morrisville ST=North Carolina,C=US
```

Since, by default, web browsers are not configured to trust the Tavve Certificate, the user is prompted by the web browser at each initial session HTTPS session is as to whether or not to trust the connection.

This initial SSL configuration is provided so that ZoneRanger's web server and web browsers are able to communicate right out of the box. In environments where a high degree of security is required, it is recommended that the web browsers and ZoneRanger's web server be reconfigured to use customer-specific certificates. The process to replace the Tavve HTTPS SSL configuration for both the ZoneRanger's web server and web browser with customer specific security credentials is as follows:

1. Replace ZoneRanger HTTPS SSL Certificate

   Using the Administration > SSL Certificate page HTTPS tab on the ZoneRanger, install the new public key certificate and private key specific to your security environment. The SSL Certificate can be in PKCS #12, X509, or Keystore format. If a problem occurs, the original Tavve SSL certificate may be restored.

2. Update each web browser's Certificate Authorities and Trusted Subjects

   Using the web browser's specific interface, add the Certificate Authority which authorized the SSL certificate which was installed on the ZoneRanger. If a problem occurs, the original Tavve trusted certificate authorities may be restored.
G. Accessing ZoneRanger Though the Ranger Gateway

Some ZoneRanger ports and services may be accessed securely by proxy through the Ranger Gateway. The Ranger Gateway assigns a set of TCP ports for each joined ZoneRanger for a particular set of services (HTTP, HTTPS, SQL, SSH, and Telnet). The `listTcpPorts` command and the Information tab on the Ranger Gateway Viewer display the port mapping for each joined ZoneRanger.

**Using a Ranger Gateway to access a ZoneRanger web interface**

By default, the ZoneRanger web interface is accessed using HTTP on port 80, or using HTTPS on port 443. However, if those ports cannot be accessed because of security considerations, you can access the ZoneRanger web interface through a Ranger Gateway. A Ranger Gateway provides proxy access to the HTTP and HTTPS ports for each joined ZoneRanger.

For example, suppose the `listTcpPorts` command returned HTTP port 20012 and the HTTPS port 20013 for a particular ZoneRanger. You would browse to one of the following URLs to access that ZoneRanger’s web interface through the Ranger Gateway:

http://RangerGatewayName:20012/
https://RangerGatewayName:20013/

These ports (in this case, 20012 and 20013) are assigned when the Ranger Gateway and ZoneRanger are joined and will remain the same while the ZoneRanger and Ranger Gateway remain joined. If they are unjoined and then joined later, the ports may change.

**Using Ranger Gateway to access and query the ZoneRanger database**

The ZoneRanger maintains information in an SQL database about discovered devices. The Ranger Gateway provides read-only access to the SQL database of each joined ZoneRanger. You can access the database using any application that supports remote SQL database connections. The SQL application you use would connect to the Ranger Gateway on the port that has been assigned to the ZoneRanger whose database you are needing to access.

For example, suppose the `listTcpPorts` command returned SQL port 20014 for a particular ZoneRanger. An SQL application could then access the ZoneRanger database by making SQL queries to `RangerGatewayName Port 20014`.

The following additional information would also be needed in order to access the database:

- **Database Name:** rangerDb
- **Database User:** ranger_ro
- **Database Password:** readonly

The Database password can be changed in the **Configuration > Access Control** page **Passwords** tab on the ZoneRanger web interface.

This port (in this case, 20014) is assigned when the Ranger Gateway and ZoneRanger are joined and will remain the same while the ZoneRanger and Ranger Gateway remain joined. If they are unjoined and then joined later, the port may change.
Using Ranger Gateway to remotely access the ZoneRanger Text Interface

By default, the ZoneRanger can be accessed using SSH on port 22, or using Telnet on port 23. However, if those ports cannot be accessed because of security considerations, you can access the ZoneRanger through a Ranger Gateway. A Ranger Gateway provides proxy access to the SSH and Telnet ports for each joined ZoneRanger.

For example, suppose the listTcpPorts command returned SSH port 20014 and the Telnet port 20015 for a particular ZoneRanger. You would access that ZoneRanger’s text interface through the Ranger Gateway:

```
telnet RangerGatewayName 20015

ssh -p 20014 RangerGatewayName
```

These ports (in this case, 20014 and 20015) are assigned when the Ranger Gateway and ZoneRanger are joined and will remain the same while the ZoneRanger and Ranger Gateway remain joined. If they are unjoined and then joined later, the ports may change.
**H. ZoneRanger Technician Access**

Even with all of the ZoneRanger audit, logging, and diagnostic capabilities, there may be rare times when Tavve Support must access a ZoneRanger at the operating system level to diagnose problems.

To enable access to a ZoneRanger while preserving ZoneRanger security, ZoneRanger provides a highly secure technician access method. The process of establishing technician access follows:

1. The customer sets up terminal access directly to the ZoneRanger.
2. The technician logs in using the terminal access setup user ID and password, as described in *ZoneRanger Installation and Configuration Guide*.  
   
   **Note:** The user ID for terminal access is always `setup`, and the initially configured password is `setup`. You should change this password as soon as initial configuration is finished.

   **Note:** The MAC address of the ZoneRanger that is displayed at the top of the Main Menu screen. Users must communicate this MAC address to Tavve Support personnel so that a time-limited, secure passcode can be generated. This passcode can only be used on the ZoneRanger having the matching MAC address, and only for a limited time after the passcode is generated. Technician access passcodes are generated at Tavve, using a custom application that encrypts and digitally signs the resulting passcode.

3. The customer enters the keyword `technicianaccess` on the main screen.
4. At the passcode prompt, the customer enters the provided passcode.
5. After a valid passcode is entered, a shell prompt appears. The customer then has operating system level access to the ZoneRanger. This level of access remains active until the technician access session is exited.

**ZoneRanger technician access security**

The ZoneRanger technician access mechanism, though cumbersome, was designed according to the following principles:

- Technician access can only take place with the cooperation of both the ZoneRanger owner and Tavve Support. ZoneRanger owners cannot use technician access without contacting Tavve Support. Technician access is possible only when using a passcode generated by Tavve Support, for a ZoneRanger having a specific MAC address, and for a specific time period.

- Technician access passcodes are highly secure. The passcodes are very difficult to break, are valid only for a ZoneRanger having a specific MAC address, and only for a specific time period. Technician access passcode generation and verification is based on public key encryption technology. The passcode is generated using a private key known only to Tavve and verified using the corresponding public key that is configured in all ZoneRangers. Passcodes generated by Tavve Support are very long and very secure.

In the unlikely event that an attacker or another ZoneRanger owner were able to obtain a technician access passcode and access the configuration menu, the attacker could not use the passcode on other ZoneRangers, or at any time outside the valid passcode period. Configuration access is difficult because it requires physical access to a ZoneRanger, and knowledge of the configuration password for the ZoneRanger.
I. Installation

This appendix describes how to install the Ranger Gateway software, which is distributed on a CD-ROM labeled Ranger Gateway.

Operating system requirements

Ranger Gateway runs on any hardware that supports the following operating systems:

- Red Hat Enterprise Linux version 4.0 or higher
- Solaris 2.8 or higher
- SuSE Linux version 11.1 or higher
- Centos 5.2 or later
- Windows 2008 Server R2, Windows 2012 Server

Notes:

- Some Unix systems might have insufficient space in the /tmp directory to perform the install. The installation requires approximately 254MB of free space in /tmp. You can set the IATEMPDIR environment variable to a directory that has sufficient space, if necessary. Run the following command to set IATEMPDIR:

  export IATEMPDIR=directory

- Perl, version 5.5 or higher, must be installed, executable, and in the PATH environment variable on Linux and Solaris systems.

- Ranger Gateway requires at least 256MB of RAM.

- In order for the Ranger Gateway software to start properly, it must be possible for the software to identify the local host address.

Installing Ranger Gateway on Linux and Solaris

You can use the GUI or console versions of the installation software to install the Ranger Gateway software on Linux or Solaris.

To use the GUI version of the installation software, follow these steps:

1. Insert the Ranger Gateway CD-ROM into the CD drive and mount the drive.
2. Change your working directory to the mounted CD.
3. Run the following command:

   ./install.ksh

   This displays the Ranger Gateway splash screen and the Welcome window for the installer.

4. Follow the installation software prompts. Ranger Gateway Viewer automatically launches after you exit the installer.

To use the console version of the installation software, follow these steps:

1. Insert the Ranger Gateway CD-ROM into the CD drive and mount the drive.
2. Change your working directory to the mounted CD.
3. Run the following command:
   
   `/install.ksh -i console`

4. Follow the installation software prompts.

**Installing Ranger Gateway on Windows**

To install Ranger Gateway on a Windows system, follow these steps:

1. Insert the Ranger Gateway software into the CD drive.
2. Open the CD drive.
3. Run `install.bat`
   
   This displays the Ranger Gateway splash screen and the **Welcome** window for the installer.
4. Follow the installation software prompts.
5. Ranger Gateway Viewer automatically launches after you exit the installer.

**Uninstalling Ranger Gateway on Linux and Solaris**

To uninstall the Ranger Gateway software on Linux and Solaris systems, run the following command:

```
install_dir/UninstallerData/Uninstall_Ranger_Gateway
```

**Note:** To run the console version of the uninstall software, run the following command:

```
install_dir/UninstallerData/Uninstall_Ranger_Gateway -i console
```

To completely remove the Ranger Gateway software, you can remove the `install_dir` directory after running

```
Uninstall_Ranger_Gateway
```

**Uninstalling Ranger Gateway on Windows**

To uninstall the Ranger Gateway software on Windows systems, use the Windows **Add/Remove Programs** control panel.
**J. Installing Ranger Gateway in Solaris 10 Zones**

Solaris 10 zones are used as a means of virtualization for applications running on Sun hardware systems. Solaris zones are used to isolate applications from each other while running on the same hardware platform. This implementation can be used to run multiple management applications on the same Sun server with each application installed in distinct zones.

There is one root or *global* zone. The *global* zone has complete access to the system. All other zones are created to have only access to specific resources on the system. There are some operating system capabilities which are only available from the *global* zone. One of those capabilities is the manipulation of network routes.

If one or more zones are running management applications, it is possible to install the Ranger Gateway with those management applications in order for those applications to manage ZoneRanger managed devices. However, due to the inability for non-*global* zones to manage network routes, the Ranger Gateway GVI will not install in non-*global* zones. In order for the Ranger Gateways installed in the non-*global* zones to use GVI in a Solaris 10 zones environment, the GVI must be installed in the *global* zone and the GVI driver made available to the non-*global* zones in which the Ranger Gateway is installed.

To install the GVI in the *global* zone, copy the file gvi.zip from the gvi directory on the Ranger Gateway installation CD to a local directory on the solaris 10 server. For this example, the gvi.zip file was placed in the /usr/gvi directory. As root,

```
cd /usr/gvi
unzip gvi.zip
./gviinstall.ksh
```

This will install a new device at /dev/tun. This device needs to be made available to any zones which will have the Ranger Gateway installed. Please see Solaris 10 documentation on how this is accomplished.

To verify the /dev/tun driver is available for a Solaris zone, run the command:

```
zonecfg -z ZoneNameHere info
```

The results should include the following:

```
device
    match: /dev/tun
```

Once the /dev/tun is available in the zone, the Ranger Gateway can be installed in that zone as the root user.
K. RGVI Client Installation and Configuration

The RGVI service within the Ranger Gateway has been designed to emulate an OpenVPN\textsuperscript{15} server, so that the freely available OpenVPN client can be used as the RGVI client. The advantages of this approach are as follows:

- OpenVPN is a widely used, award-winning SSL VPN solution, with over 3 million users and 150,000 downloads per month.
  
  (see http://www.openvpn.net/index.php/about/openvpn-facts.html)

- The OpenVPN client has a relatively small processor/memory footprint, resulting in minimal impact to management application servers.

- The OpenVPN client is supported on a wide variety of operating system versions. Pre-built Windows versions are available directly from the OpenVPN web site. Solaris packages and Linux RPMs are available from other sources.

- The OpenVPN protocol is highly secure, with two-way SSL/TLS-based authentication, message integrity protection, and replay attack prevention.

- The OpenVPN client and the GVI service on the Ranger Gateway use the same underlying technology for intercepting management traffic, so RGVI and GVI can provide very similar functionality.

- The minimum OpenVPN client that supports IPv6 is 2.3.

Even though RGVI makes use of the OpenVPN client, this does not imply that the resulting system is essentially a VPN. Rather, VPN technology is employed to intercept traffic and to route this traffic to the Ranger Gateway, at which point the traffic is inspected, processed, and proxied via one or more ZoneRangers in the same manner as locally-intercepted traffic (e.g. via GVI). So the end result is an application layer proxy firewall with a VPN-based front-end, as opposed to a simple VPN.

In general, the process for installing and configuring the OpenVPN client consists of four steps:

1. Obtain the OpenVPN software from the OpenVPN web site, or a trusted third-party web site that provides pre-built OpenVPN versions for the operating system you are using.

2. Install the OpenVPN software on the management application server.

3. Copy configuration file and certificate file samples from the Ranger Gateway install CD, to the appropriate directory, and edit the configuration file to point the client to the Ranger Gateway server(s) you are using.

4. Configure the management application server to automatically start the OpenVPN client.

In order to configure the client with the list of Ranger Gateway candidates, as described in Step 3 above, you will need to locate the following placeholder in the copied configuration file:

```plaintext
# Replace the following address with Ranger Gateway's address
remote 192.168.1.1
```

The remote \texttt{192.168.1.1} line must be replaced with one or more lines indicating the available Ranger Gateway candidates. For example, if there are two candidates with addresses \texttt{10.254.12.1} and \texttt{10.254.12.2}, the placeholder should be replaced with the following lines:

```plaintext
# Replace the following address with Ranger Gateway's address
remote 10.254.12.1
remote 10.254.12.2
```

\textsuperscript{15}OpenVPN is a freely-available open source SSL VPN solution (see http://www.openvpn.net)
Note that the Ranger Gateway install CD provides sample SSL/TLS certificates that are matched to the default configuration of the RGVI service on the Ranger Gateway. If you prefer to use your own certificates, you will need to modify the OpenVPN client configuration to use your certificates, and you will need to modify the RGVI configuration on the Ranger Gateway, to accept your certificates.

The remaining details for performing the installation and configuration steps are dependent on the operating system being used. Additional information for a number of supported operating systems is provided in the following sections. If no information has been provided for the operating system you are using, please contact Tavve technical support.

**Solaris**

A pre-built Solaris OpenVPN package can be downloaded from https://www.opencsw.org/package/openvpn/, an open source Solaris software site. In order to install packages from the opencsw site, you will need to have the pkgutil tool, installed on your server. You can test to see if the pkgutil tool is already installed by looking for the following executable:

```
/opt/csw/bin/pkgutil
```

If the pkgutil executable is not found, follow the instructions for downloading and installing pkgutil, as described on the following web page:


Once pkgutil has been installed, you can install OpenVPN by simply executing the following command:

```
/opt/csw/bin/pkgutil/pkgutil --install openvpn
```

The installation process installs the openvpn executable in the /opt/csw/sbin directory, and creates the following directory for OpenVPN configuration files:

```
/etc/csw/openvpn
```

The next step is to copy the following sample files from the rgvi directory on the Ranger Gateway install CD to the /etc/csw/openvpn directory. The specific files to be copied, and the associated configuration instructions depend on whether you prefer to start the OpenVPN client manually or intend to configure the OpenVPN client to start automatically when the operating system is restarted (i.e. via an init.d script), as described in the following sections.

**Starting the OpenVPN Client Manually**

If you prefer to run the OpenVPN client manually, copy the following files from the rgvi directory on the Ranger Gateway install CD to the /etc/csw/openvpn directory:

- rgviClient.conf
- rgviClient.crt
- rgviClientWithPassword.key
- tavveCA.crt

After the files have been copied, you will need to edit the rgviClient.conf file to specify the list of Ranger Gateway candidates, as described above. Once this step has been completed, you can run the OpenVPN client by executing the following commands:
cd /etc/csw/openvpn
/opt/csw/sbin/openvpn rgviClient.conf

The rgviClientWithPassword.key file is password-protected, so when OpenVPN is started, you will be prompted to enter the password. The password for the default rgviClientWithPassword.p12 file is rgvi.

Note that the Ranger Gateway should be running and the RGVI service should be enabled and configured to accept connections from this client before the RGVI client is started. You can verify that the RGVI client has successfully started and connected to the RGVI service on the Ranger Gateway by verifying that the IP address associated with RGVI client is listed in the output of the following command (executed on the Ranger Gateway server):

```
/opt/tavve/gateway/bin/rgvi status
```

### Starting the OpenVPN Client Automatically

If you prefer to have the OpenVPN client start automatically when the operating system is restarted, copy the following files from the rgvi directory on the Ranger Gateway install CD to the /etc/csw/openvpn directory:

- rgviClient.conf
- rgviClient.crt
- rgviClientNoPassword.key
- tavveCA.crt

After the files have been copied, you will need to edit the rgviClient.conf file to specify the list of Ranger Gateway candidates, as described above. In addition, you will need to modify the rgviClient.conf file to indicate that the rgviClientNoPassword.key key file should be used, because there is no way to provide a key file password when the client is started automatically. Two changes are required:

1. Comment out the “key rgviClientWithPassword.key” line.
2. Uncomment the “# key rgviClientNoPassword.key” line.

Note that the “#” character denotes a comment line. The resulting two lines should be as follows:

```
# key rgviClientWithPassword.key
key rgviClientNoPassword.key
```

The recommended steps for configuring the OpenVPN client to be started automatically when the operating system is restarted are as follows:

1. Create a copy of the /etc/init.d/openvpn script (initially created by the OpenVPN installer), in the same directory, by executing the following command:

```
cp /etc/init.d/openvpn /etc/init.d/rgviClient
```

2. Edit the newly-created /etc/init.d/rgviClient file to indicate that the sample RGVI configuration should be used. To do this, replace the line that reads:

```
OPENVPN_CONF=/etc/csw/openvpn/openvpn.conf
```

with:

```
OPENVPN_CONF=/etc/csw/openvpn/rgviClient.conf
```
3. Create a symbolic link in the /etc/rc3.d directory to the /etc/init.d/rgviClient file, by executing the following command:

```
ln -s /etc/init.d/rgviClient /etc/rc3.d/S95rgviClient
```

4. Create a symbolic link in the /etc/rc1.d directory to the /etc/init.d/rgviClient file by executing the following command:

```
ln -s /etc/init.d/rgviClient /etc/rc1.d/K16rgviClient
```

After completing these configuration steps, you will be able to start the OpenVPN client by executing the following command:

```
/etc/init.d/openvpn start
```

Similarly, you can stop the OpenVPN service by executing the following command:

```
/etc/init.d/openvpn stop
```

If you are using Solaris 10, you also have the option of using the new Service Management Facility (SMF) to configure the OpenVPN client to start automatically. Instructions for configuring services using SMF are provided in the Solaris 10 documentation.

**Linux**

As indicated on the OpenVPN downloads page, pre-built OpenVPN RPM’s for a number of Linux variants can be downloaded from the following web page:

```
http://dag.wieers.com/rpm/packages/openvpn
```

As an alternative to downloading a pre-built package, you can download the OpenVPN source code and build/install using the `./configure` convention, as described in the “Linux Notes (without RPM)” section on the following web page:

```
```

In summary, the required steps are:

1. Download openvpn-[version].tar.gz from the OpenVPN download page.

2. Expand the downloaded file by executing the following command:

   ```
tar xzf openvpn-[version].tar.gz
```

3. `cd` to the top-level directory and execute the following commands:

   ```
./configure
   make
   make install
   ```

By default, the installation process installs the `openvpn` executable in the `/usr/sbin` directory, and creates the following directory for OpenVPN configuration files:

```
/etc/openvpn
```

The next step is to copy the following sample files from the `rgvi` directory on the Ranger Gateway install CD to the `/etc/openvpn` directory. The specific files to be copied, and the associated configuration instructions depend on whether you prefer to start the OpenVPN client manually or intend to configure the OpenVPN client to start automatically when the operating system is restarted (i.e. via an `init.d` script), as described in the following sections.
Starting the OpenVPN Client Manually

If you prefer to run the OpenVPN client manually, copy the following files from the \texttt{rgvi} directory on the Ranger Gateway install CD to the \texttt{/etc/openvpn} directory:

- \texttt{rgviClient.conf}
- \texttt{rgviClient.crt}
- \texttt{rgviClientWithPassword.key}
- \texttt{tavveCA.crt}

After the files have been copied, you will need to edit the \texttt{rgviClient.conf} file to specify the list of Ranger Gateway candidates, as described above. Once this step has been completed, you can run the OpenVPN client by executing the following commands:

\begin{verbatim}
  cd /etc/openvpn
  /usr/sbin/openvpn rgviClient.conf
\end{verbatim}

The \texttt{rgviClientWithPassword.key} file is password-protected, so when OpenVPN is started, you will be prompted to enter the password. The password for the default \texttt{rgviClientWithPassword.p12} file is \texttt{rgvi}.

Note that the Ranger Gateway should be running and the RGVI service should be enabled and configured to accept connections from this client before the RGVI client is started. You can verify that the RGVI client has successfully started and connected to the RGVI service on the Ranger Gateway by verifying that the IP address associated with RGVI client is listed in the output of the following command (executed on the Ranger Gateway server):

\begin{verbatim}
  /usr/tavve/gateway/bin/rgvi status
\end{verbatim}

Starting the OpenVPN Client Automatically

If you prefer to have the OpenVPN client start automatically when the operating system is restarted, copy the following files from the \texttt{rgvi} directory on the Ranger Gateway install CD to the \texttt{/etc/openvpn} directory:

- \texttt{rgviClient.conf}
- \texttt{rgviClient.crt}
- \texttt{rgviClientNoPassword.key}
- \texttt{tavveCA.crt}

After the files have been copied, you will need to edit the \texttt{rgviClient.conf} file to specify the list of Ranger Gateway candidates, as described above. In addition, you will need to modify the \texttt{rgviClient.conf} file to indicate that the \texttt{rgviClientNoPassword.key} key file should be used, because there is no way to provide a key file password when the client is started automatically. Two changes are required:

3. Comment out the \texttt{"key rgviClientWithPassword.key"} line.
4. Uncomment the \texttt{"# key rgviClientNoPassword.key"} line.

Note that the \texttt{"#"} character denotes a comment line. The resulting two lines should be as follows:

\begin{verbatim}
  # key rgviClientWithPassword.key
  key rgviClientNoPassword.key
\end{verbatim}
The recommended steps for configuring the OpenVPN client to be started automatically when the operating system is restarted are as follows:

1. Create a symbolic link in the /etc/rc3.d directory to the /etc/init.d/openvpn file, by executing the following command:
   
   ```
   ln -s /etc/init.d/openvpn /etc/rc3.d/S95openvpn
   ```

2. Create a symbolic link in the /etc/rc1.d directory to the /etc/init.d/openvpn file by executing the following command:
   
   ```
   ln -s /etc/init.d/openvpn /etc/rc1.d/K16openvpn
   ```

Note that the /etc/init.d/openvpn script will start an instance of OpenVPN for every .conf file that it finds in the /etc/openvpn directory. Before starting OpenVPN the first time, you should inspect the /etc/openvpn directory to ensure that there are no unexpected .conf files. If the OpenVPN client is only being used for RGVI, the only .conf file in the directory should be rgviClient.conf.

After completing these configuration steps, you will be able to start the OpenVPN client by executing the following command:

   ```
   /etc/init.d/openvpn start
   ```

Similarly, you can stop the OpenVPN service by executing the following command:

   ```
   /etc/init.d/openvpn stop
   ```

The chkconfig utility, if available on your Linux system, can simplify the process of configuring and managing Linux services. Information describing this utility can be found at the following URLs:


   [http://www.netadmintools.com/art94.html](http://www.netadmintools.com/art94.html)

**Microsoft Windows**

A pre-built installer for Microsoft Windows is available on the Downloads page of the official OpenVPN web site ([http://www.openvpn.net/index.php/open-source/downloads.html](http://www.openvpn.net/index.php/open-source/downloads.html)). According to the information on the OpenVPN web site, Windows 2000 and all later versions are currently supported. Note that if you are using Windows Vista, or a 64-bit Windows variant, you will need to use the OpenVPN 2.1 or later client.

To install the OpenVPN client, simply execute the downloaded installer, accepting all default settings. On most Windows installations, the OpenVPN software will be installed in the following directory:

   ```
   C:\Program Files\OpenVPN
   ```

However, please note that on 64 Bit Windows systems, the OpenVPN software may be installed in the following directory:

   ```
   C:\Program Files(x86)\OpenVPN
   ```

The next step is to copy sample files from the rgvi directory on the Ranger Gateway install CD to the C:\Program Files\OpenVPN\config directory. The specific files to be copied, and the associated configuration instructions depend on whether you prefer to start the OpenVPN client manually or intend to run the OpenVPN client as a Windows service, as described in the following sections.
Starting the OpenVPN Client Manually

If you prefer to run the OpenVPN client manually, copy the following files from the rgvi
directory on the Ranger Gateway install CD to the C:\Program Files\OpenVPN\config
directory:

- rgviClient.conf
- rgviClient.crt

After the files have been copied, you will need to edit the rgviClient.conf file to specify the
list of Ranger Gateway candidates, as described above. Once this step has been completed, you
can run the OpenVPN client by executing the following commands:

```
cd C:\Program Files\OpenVPN\config
..\bin\openvpn rgviClient.conf
```

The rgviClient.crt file is password-protected, so when OpenVPN is started, you will be
prompted to enter the password. The password for the default rgviClient.crt file is rgvi.

Note that the Ranger Gateway should be running and the RGVI service should be enabled and
configured to accept connections from this client before the RGVI client is started. You can verify
that the RGVI client has successfully started and connected to the RGVI service on the Ranger
Gateway by verifying that the IP address associated with RGVI client is listed in the output of the
following command (executed on the Ranger Gateway server):

```
$RangerGatewayInstallDir\bin\rgvi status
```

Running the OpenVPN Client as a Windows Service

If you prefer to run the OpenVPN client as a Windows service, copy the following files from the
rgvi directory on the Ranger Gateway install CD to the C:\Program Files\OpenVPN\config
directory:

- rgviClientWindowsService.ovpn
- rgviClientWindowsService.p12
- tavveCA.crt
- LocalComputerAccountPersonalCertificatesConsole.msc

After copying these files, you will need to edit the rgviClientWindowsService.ovpn file
to specify the list of Ranger Gateway candidates, as described above. Note that you should
remove any other files with the .ovpn extension in the config directory before starting the
OpenVPN service.

Note that the rgviClientWindowsService.p12 file is password-protected, but there is no
way to specify the required password at the point when the OpenVPN service is started. The
recommended solution is to load the key and certificate information from the
rgviClientWindowsService.p12 file into the Windows Certificate Store, using the
following steps:

1. Open the pre-configured certificate management console, by executing the following
   commands:

   ```
cd C:\Program Files\OpenVPN\config
   mmc LocalComputerAccountPersonalCertificatesConsole.msc
   ```

2. The Local Computer Account Personal Certificates console window will open, as shown
   in the following figure.
3. Right-click on the **Personal** icon in the left-hand panel, and select **All Tasks → Import...** as shown in the following figure.

4. The welcome page for the **Certificate Import Wizard** will be displayed. Read the information on the welcome page, then click the **Next** button. The File to Import page will be displayed as shown in the following figure.
5. Click the **Browse** button. An **Open** file dialog will be displayed. Browse to the `C:\Program Files\OpenVPN\config` directory, select **Personal Information Exchange (*.pfx *.p12)** from the **Files of type** drop down list, then select the `rgviClientWindowsService.p12` file, as shown in the following figure.

6. Click the **Open** button. The **File to Import** page will be re-displayed. Click the **Next** button. The **Password** page will be displayed, as shown in the following figure.
7. Enter the private key password associated with the rgviClientWindowsService.p12 file (i.e. rgvi) in the Password text box, then click the Next button. The Certificate Store page will be displayed as shown in the following figure.

8. Click the Next button. The Completing the Certificate Import Wizard page will be displayed. Click the Finish button. A confirmation dialog will be displayed, indicating that the import was successful. Click the OK button. The Local Computer Account Personal Certificates console will be re-displayed. Click on the Certificates item in the left hand panel, to display the personal certificates that have been configured for the local computer account, as shown in the following figure.
9. Click the close box or select **File → Exit** to close the console.

The OpenVPN installer automatically configures a new windows service named “Open VPN Service”, but by default, the service is not started and the **Startup type** for the service is Manual. Before attempting to start this service, verify that the **rgviClientWindowsService.ovpn** file is the only file with the .ovpn extension in the C:\Program Files\OpenVPN\config directory. Any other files with this extension should be deleted, renamed to have a different extension, or moved to a different directory. To start the OpenVPN service, open the **Services** control panel tool (located in the **Administrative Tools** group), scroll to locate the entry for OpenVPN Service, right-click on this entry, and select **Properties** from the resulting drop-down menu. The properties page for the OpenVPN Service will be displayed as shown in the following figure.

![OpenVPN Service Properties](image)

Select **Automatic** in the **Startup type** drop-down, click the **Start** button to start the service, then click the **OK** button to save your settings and close the window.
The OpenVPN service should now be started, and should automatically restart whenever the server is rebooted. You can inspect the status of the OpenVPN service by looking in the log file at the following location:

C:\Program Files\OpenVPN\log\rgviClientWindowsService.log

If OpenVPN started and connected to the Ranger Gateway successfully, the following message should be displayed in the log file:

Initialization Sequence Completed