Gateway Virtual Interface Primer

Introduction

In order for the ZoneRanger to proxy management traffic to managed DMZ 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 DMZ devices to the Ranger Gateway.

2. Provide a mechanism whereby management traffic destined for DMZ devices is intercepted by the Ranger Gateway.

The primary advantage of the latter approach is that the management application can send requests to DMZ devices as if it was communicating with those devices directly. The fact that the requests are subsequently 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.* There are a number of mechanisms that can be used to implement transparent proxy. One such mechanism, the Gateway Virtual Interface (GVI), is described in this primer.

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 DMZ devices is routed to this virtual interface. The Ranger Gateway, as the creator/owner of the virtual interface, receives all traffic that is routed to the virtual interface, and forwards this traffic to configured proxy services (e.g. SNMP proxy, TCP proxy) within the Ranger Gateway. The proxy services consult 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 forward the traffic to the selected ZoneRanger, which in turn, forwards the traffic to the target DMZ device. Where applicable, proxy services may also perform validation and filtering of the management traffic, as appropriate for the service in question.

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 DMZ 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, if the Ranger Gateway is communicating with a joined ZoneRanger using the IP address `10.1.10.100`, via gateway `10.2.1.1`, and the route manager is configured to route the `10.1.10.0/255.255.255.0` subnet to the virtual interface, the route manager will automatically create the following routes:

```
10.1.10.100 255.255.255.255 10.2.1.1
10.1.10.0 255.255.255.0 192.168.48.2
```

Similarly, if the GVI service is enabled, and a request to join to a given ZoneRanger is received by the Ranger Gateway, the 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 is controlled and configured using the `gvi` command. The various subcommands and options provided by the `gvi` command are described in the following section.

Command Reference

The `gvi` command includes a variety of subcommands that can be used to enable/disable the GVI, view the GVI status, manage static routes, configure GVI service options, and test the GVI service, to verify that it has been configured correctly. Each subcommand, and its associated parameters and options, are described in the following sections.

Enabling and Disabling the GVI Service

By default, the GVI service is disabled. The `gvi enable` subcommand can be used to enable the GVI service:

```
gvi enable
```

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.

The `gvi disable` command can be used to disable the GVI service:

```
gvi disable
```

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.

The `gvi status` subcommand can be used to display the current status of the GVI service:

```
gvi status
```

The output of the `gvi status` subcommand will indicate whether the GVI service is currently enabled or disabled. In addition, the `gvi status` subcommand will display any errors or warnings that were generated during the most recent route manager operation.

Managing Virtual Interface Routes

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 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.

```
gvi add-route <subnet>|<address> <subnet>|<address> ...
```

Each parameter after the `add-route` subcommand name can either be a specific IP address, 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

The `gvi merge-routes` subcommand is similar to the `gvi add-routes` subcommand. The primary difference is that the `gvi merge-routes` subcommand adds virtual interface routes for subnets and individual IP addresses listed in a specified input file.

```
gvi merge-routes <input-file>
```

The required format for subnet or IP address entries in the input file is the same as for the `gvi add-route` command, except that only one entry per line is allowed.

The `gvi remove-route` subcommand can be used to remove one or more subnets or individual IP addresses 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 that has been removed from the list.

```
gvi remove-route [<subnet>|<address] [<subnet>|<address] ...
```

The `gvi-list-routes` subcommand can be used to list all IP addresses and subnets in the GVI route list:

```
gvi list-routes [-all]
```

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 (i.e. they look like virtual interface routes or ZoneRanger host routes, but do not appear to have been created by the route manager). 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 route manager.

The `gvi-clear-routes` subcommand can be used to clear the GVI route list:

```
gvi clear-routes
```

When the GVI route list is cleared, all virtual interface and ZoneRanger host routes will be removed from the system routing table.

The `gvi generate-routes` subcommand makes use of other Ranger Gateway configuration information, or ZoneRanger database information, to identify subnets or individual IP addresses that may be considered as candidates for the GVI route list.

```
gvi generate-routes [-subnet|-address] [-database|-proxyMap]
```

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` subcommand 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` subcommand 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.

WARNING! 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 DMZ being managed, and the creation of virtual interface routes for such addresses would interfere with the management application's ability to communicate with non-DMZ 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.

Configuring GVI Options

The `gvi config` subcommand is used to configure GVI service options:

```
gvi config [<item> [<value>]]
```

The following options can be configured:

Item	Valid Values	Default Value
`log_level`	`none, short, full`	`none`

Examples:

1. To display all configuration items and their current values:

   ```
   gvi config
   ```

2. To display the current value of the `log_level` item:

   ```
   gvi config log_level
   ```

3. To set the value of the `log_level` item to `full`:

   ```
   gvi config log_level full
   ```

Testing the GVI Service

The `gvi test` command can be used to verify that the GVI and proxy map services in the Ranger Gateway have been configured properly so that traffic send to a specified address can be intercepted, routed to an appropriate ZoneRanger, and forwarded to the intended device.

```
gvi test <address>
```

Example:

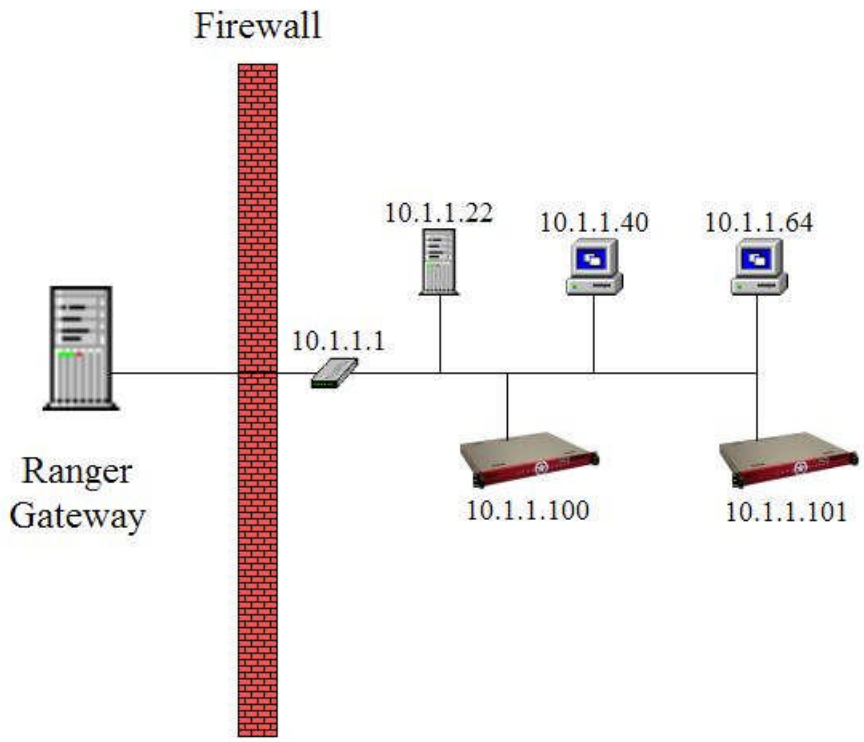
```
gvi test 10.99.99.1
```

Sample output:

```
The Gateway Virtual Interface is enabled.
The Gateway Virtual Interface responded to request to 10.99.99.1.
Request will be proxied through ZoneRanger 10.1.10.20 to 127.0.0.1.
ZoneRanger 10.1.10.20 is responding and 127.0.0.1 is managed.
SUCCESS
```

Simple Setup Examples

Assume a simple network, with no NAT in effect, as illustrated in the following diagram:



The following commands would be required to configure and test the GVI service:

1. Ensure that the Ranger Gateway is joined to the two ZoneRangers:

   ```
   joinRequest 10.1.1.100
   joinRequest 10.1.1.101
   ```

2. Add proxy map rules indicating that both ZoneRangers are managing devices in the `10.1.10.0/255.255.255.0` subnet:

   ```
   proxyMapTool add 10.1.1.* 10.1.1.100
   proxyMapTool add 10.1.1.* 10.1.1.101
   ```

3. Add a route for the `10.1.1.0/255.255.255.0` subnet to the GVI route list:

   ```
   gvi add-route 10.1.1.*
   ```

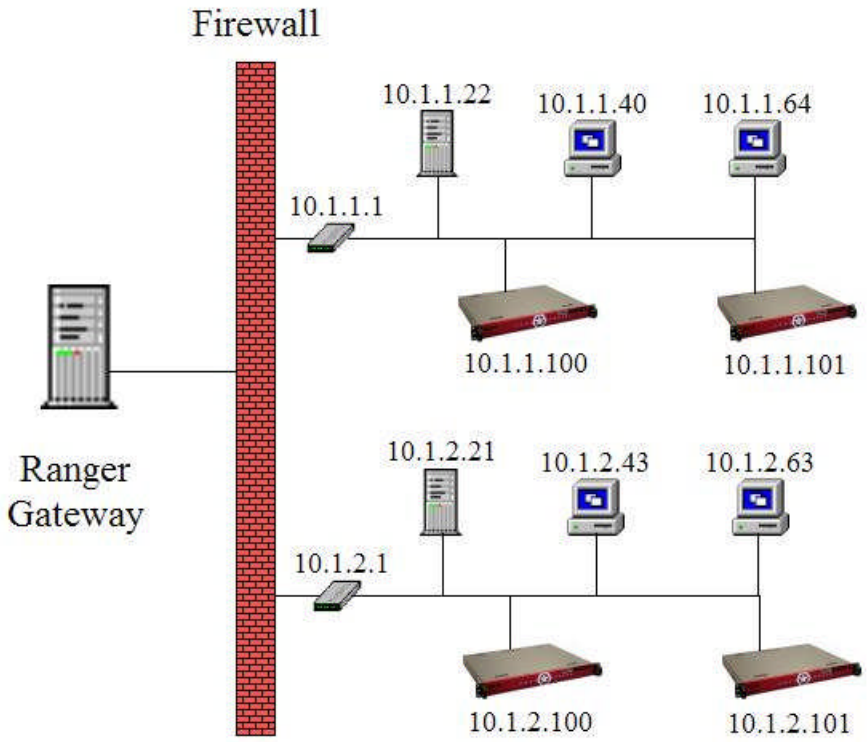
4. Enable the GVI service:

   ```
   gvi enable
   ```

5. Test the configuration, specifying the address of a managed device:

   ```
   gvi test 10.1.1.40
   ```

As a slightly more complicated example, assume that the Ranger Gateway is joined to two pairs of ZoneRangers, where each pair is managing a different network, as shown in the following figure.



In this case, the following commands would be required to configure and test the GVI service:

1. Ensure that the Ranger Gateway is joined to the four ZoneRangers:

```
joinRequest 10.1.1.100
joinRequest 10.1.1.101
joinRequest 10.1.2.100
joinRequest 10.1.2.101
```

2. Add proxy map rules indicating the subnets being managed by each ZoneRanger:

```
proxyMapTool add 10.1.1.* 10.1.1.100
proxyMapTool add 10.1.1.* 10.1.1.101
proxyMapTool add 10.1.2.* 10.1.2.100
proxyMapTool add 10.1.2.* 10.1.2.101
```

3. Add routes for the `10.1.1.0/255.255.255.0` and `10.1.2.0/255.255.255.0` subnets to the GVI route list:

   ```
   gvi add-route 10.1.1.*
   gvi add-route 10.1.2.*
   ```

   Note that in this case, the two subnets could be optionally combined into a single route:

   ```
   gvi add-route 10.1.0.0/255.255.252.0
   ```

4. Enable the GVI service:

   ```
   gvi enable
   ```

5. Test the configuration, specifying the address of a few managed devices:

   ```
   gvi test 10.1.1.40
   gvi test 10.1.2.63
   ```