Mandiant

CONFIGURE SOURCE-BASED ROUTING FOR NETWORK ACTORS

Linux-based Network Actors can have multiple network interfaces that are used to separate management traffic from test and monitoring traffic. You may want the Actor-Director communication (*management* traffic) on distinct IPs or even distinct networks from where Action traffic (Actor-to-Actor traffic when running security content like Network Actions or File Transfer Actions) runs (*test* traffic). This separation lets you distinguish between network traffic that is part of Security Validation command and control (*management*) traffic and network traffic that is intended to be exercising security controls (*test*).

> If only a single interface or single IP is used on an Actor, network traces don't distinguish which network packets are part of Actor-Director communications and which are part of running security content.

For Network Actors with multiple interfaces, additional network configuration is required to ensure that network traffic enters and leaves through the desired network interface.

For example, Red Hat Enterprise Linux 8 (RHEL8) uses Network Manager for configuring network interfaces through the NetworkManager Command Line Interface (`nmcli`) tool.

As an example of configuring source-based routing with multiple interfaces, consider a Linux Actor with three network interfaces configured through `vsetnet` :

- `eth0` : Configured as the *management* interface with example IP address `10.10.1.0`
- `eth1` : Configured as the *test* interface with example IP address `10.20.2.20`
- `eth2` : Configured as the *monitor* interface with example IP address `10.30.3.30`

You must run additional `nmcli` commands after running `vsetnet` .

> In `vsetnet` , if you only configure one interface, `eth0` as management, no additional `nmcli` commands are required.

Identify Device and Connection Mappings

The following command shows the association Network Manager has between devices and connections. Subsequent `nmcli` commands require a connection name.

Run the command as follows:

```
% nmcli –fields device,name connection show
```

Then, the device and name mapping appears in the output:

```
DEVICE NAME
eth0   Wired Connection 1
eth1   Wired Connection 2
eth2   Wired Connection 3
```

Separate Traffic Based on Configured Interfaces

- If you configure all three interfaces (for example, `eth0` as management, `eth1` as test, and `eth2` as monitor), you need to run all the commands that follow.
- If you only configure two interfaces, `eth0` as management, and `eth1` as test, you only need the `nmcli` commands in the example that references `Wired Connection 2`.

Taking the connection names from the preceding example, use the following commands to separate management traffic from test and monitor traffic:

```
% nmcli con mod "Wired Connection 2" ipv4.route-table 10
% nmcli con mod "Wired Connection 2" ipv4.routing-rules "priority 10 iif eth1 table 10"
% nmcli con mod "Wired Connection 2" +ipv4.routing-rules "priority 10 from 10.20.2.20 table 10"
```

Specifically, for monitor traffic separation, run these commands:

```
% nmcli con mod "Wired Connection 3" ipv4.route-table 20
% nmcli con mod "Wired Connection 3" ipv4.routing-rules "priority 20 iif eth2 table 20"
% nmcli con mod "Wired Connection 3" +ipv4.routing-rules "priority 20 from 10.30.3.30 table 20"
```