MANRS Training Lab

Open source tooling to provide hands-on labs
Context

• Developed for MANRS
• But wider applicability kept in mind all the time

• Initial development funded by ISOC (thanks!)
• Released under GPL3 after completion
Design goals

• Lab content goals
  - Allow student to get hands-on experience with different platforms: Cisco, Junos, Mikrotik, etc…
  - Automatic validation of the results
  - Teach student to use IRR
Design goals

• Management goals
  - Allow self-signup by students (optional)
  - Set a time limit for a lab (optional)
  - Allow teacher to extend time limit, pause, restart, export etc.
  - Allow teacher to view what a student is doing and help them
Component choices

• GNS3 for virtual lab
• Django front-end (uWSGI application)
• Web based interface using web sockets
• Redis for live communication channels
• PostgreSQL database for storage
Hardware requirements

• Heavy hardware preferred
  - Every student gets their own clone
  - Every clone needs a few GB of memory and some CPU cycles
  - Juniper vMX needs a little bit more GB and CPU cores
Workflow

• Build a lab template in GNS3
• Link it to the management system
• Configure description, instructions and goals

• Each student gets a clone of the template
  - Either created by teacher or by self-signup
Example: MANRS Lab
Example: MANRS Lab - GNS3 template
Example: MANRS Lab - Linking to training system
Example: MANRS Lab - Goal definition

### MANRS Lab Manager

**Home > Lab > Monitor templates > MANRS AS64501**

#### Change monitor template

**Name:** MANRS AS64501

**Instructions:**

The customer (AS64501)

Customer 64501 should announce the following prefixes to you:

- `2001:db8:1001::/48`
- `192.0.2.0/24`

For testing purposes you can ping them on addresses 2001:db8:1001::1 and 192.0.2.1.

Use markdown for styling

---

### MONITOR GOALS

**Monitor goal: MANRS AS64501: Received traffic**

<table>
<thead>
<tr>
<th>Goal type:</th>
<th>Received traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal content:</td>
<td>SRC=108.51.109.1 DST=192.0.2.2</td>
</tr>
</tbody>
</table>
Monitor goals: MANRS AS64501

**Monitor goal: MANRS AS64501: Received traffic**

**Goal type:** Received traffic

**Goal content:**

```
SRC=198.51.100.1 DST=192.0.2.1
SRC=10.0.0.1 DST=192.0.2.1
```

**Monitor goal: MANRS AS64501: IPv4 routes**

**Goal type:** IPv4 routes

**Goal content:**

```
BIRD 1.5.0 ready.
10.0.0.0/8 via 203.0.113.252 on eth0 [bgp1 11:48:05] * (100) [AS65001]
    Type: BGP unicast univ
    BGP.origin: IGP
    BGP.as_path: 64500 64510 65000 65001
    BGP.nexthop: 203.0.113.252
    BGP.local pref: 100
192.168.0.0/16 via 203.0.113.252 on eth0 [bgp1 11:48:05] * (100) [AS65003]
    Type: BGP unicast univ
    BGP.origin: IGP
```

**Monitor goal: MANRS AS64501: IPv6 routes**

**Goal type:** IPv6 routes

**Goal content:**

```
BIRD 1.5.0 ready.
2001:0db8:2002::/48 via 2001:db8:1000:6ffe::a on eth0 [bgp1 11:44:26] * (100) [AS64502i]
    Type: BGP unicast univ
    BGP.origin: IGP
    BGP.as_path: 64500 64602
```
Example: MANRS Lab - Student's view

**MANRS Lab Manager**
Dashboard: MANRS-Cisco for Sander Steffann

**MANRS for Cisco**
Welcome to the MANRS for Cisco lab. This lab consists of a transit, a peer, two customers, and your very own Cisco router in the middle. The goal is to implement MANRS on your router so that the other routers cannot send you hijacked routes or traffic with spoofed source addresses. And they will try.
The layout of this lab is based on the MANRS Implementation Guide. The addresses and prefixes used in this lab correspond to those used in that document.

**Background information**
At the start of the lab all links are configured and BGP sessions exist for both IPv4 and IPv6. There is no filtering in place. That is your task.

**Your router (AS64500)**
You have full console access to your router. Configure it so it has MANRS.
You should announce the following prefixes from your own router:
- 2001:0db8:1000::/36
- 203.1.11.1/24

**The transit (AS64510)**
The transit will send you the most routes. But it isn't behaving completely correct. Some of its routes are your own! Make sure you don't accept them, or someone on the internet might hijack you. There is also traffic coming from the transit with source addresses that don't exist in the routing table. Those should also be blocked.
For testing purposes you can ping the transit on addresses 2001:db8::1 and 18.0.6.1.

**The peer (AS64511)**
The peer will do the same as the transit, except that of course it's only allowed to attract traffic for itself. So make sure that you filter what they announce to you, and also make sure they don't use you as a free transit!
The peer should announce the following prefixes to you:
- 2001:0db8:1000::/36
For testing purposes you can ping the peer on address 2001:db8:1001::1.

**The customers (AS64501 and AS64502)**
You have two customers in this lab. Both of them have IPv4 and IPv6 address space. AS64501 has address space from your aggregated blocks, AS64502 has provider independent space. But beware: there is also some hijacking going on! Make sure both of your customers behave and don't interfere with each other or the rest of the internet.
Customer 64501 should announce the following prefixes to you:
- 2001:db8::1001::/48
- 192.8.2.0/24
Example: MANRS Lab - Student's view
**Example: MANRS Lab - Student's view**

The customer (AS64501)

Customer AS64501 should announce the following prefixes to you:

- 2001:db8:1001::/48
- 192.0.2.0/24

For testing purposes you can ping them on addresses 2001:db8:1001::1 and 192.0.2.1.

**Looking glass from this router's viewpoint**

**Received traffic (last change at 3:36:18)**

<table>
<thead>
<tr>
<th>Expected</th>
<th>Currently seen</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1 to 192.0.2.1</td>
<td>10.0.0.1 to 192.0.2.1</td>
</tr>
<tr>
<td>These packets shouldn't be received</td>
<td>192.0.2.3 to 192.0.2.1</td>
</tr>
<tr>
<td>These packets shouldn't be received</td>
<td>192.88.99.2 to 192.0.2.1</td>
</tr>
<tr>
<td>These packets shouldn't be received</td>
<td>192.88.99.10 to 192.0.2.1</td>
</tr>
<tr>
<td>198.51.100.1 to 192.0.2.1</td>
<td>198.51.100.1 to 192.0.2.1</td>
</tr>
<tr>
<td>These packets shouldn't be received</td>
<td>198.51.100.3 to 192.0.2.1</td>
</tr>
<tr>
<td>2001:db8::1 to 2001:db8:1001::1</td>
<td>2001:db8::1 to 2001:db8:1001::1</td>
</tr>
<tr>
<td>These packets shouldn't be received</td>
<td>2001:db8::1001:1 to 2001:db8:1001::1</td>
</tr>
<tr>
<td>These packets shouldn't be received</td>
<td>2001:db8::1001:3 to 2001:db8:1001::1</td>
</tr>
<tr>
<td>These packets are missing</td>
<td>3ffe::2 to 2001:db8:1001::1</td>
</tr>
<tr>
<td>These packets shouldn't be received</td>
<td>3ffe::10 to 2001:db8:1001::1</td>
</tr>
</tbody>
</table>

**IPv4 routes (last change at 3:38:29)**

<table>
<thead>
<tr>
<th>Expected</th>
<th>Currently seen</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0/8</td>
<td>10.0.0.0/8</td>
</tr>
<tr>
<td>AS-Peered: 64500</td>
<td>AS-Peered: 64500</td>
</tr>
<tr>
<td>65000 65000 65000 65000 65000</td>
<td>65000 65000 65000 65000 65000</td>
</tr>
<tr>
<td>172.16.0.0/12</td>
<td>172.16.0.0/12</td>
</tr>
</tbody>
</table>
Sander Steffann, RIPE78, 2019-05-22

Ubuntu 18.04 Server
uWSGI labmgr
module
WebSocket data
HTTP requests
GNS Lab Instance
Create
Start
Manage
Manage
Query & Update
Status update
Status update
Status update
Status update
Telnet connection
Lab data
Project state
Store last state
Update live state
Cache
Authentication
Telnet in/output
Redis
PostgreSQL
uWSGI Emperor
uWSGI web socket
GNS3 Server
Django web service
Telnet relay service
Lab Monitor
AS64510
Lab Monitor
AS64511
Lab Monitor
AS64501
Lab Monitor
AS64502
Lab Monitor
AS64510
Lab Monitor
AS64511
IRR node
Work node
AS64500
Teacher workstation
OpenVPN Client
Telnet client
GNS3 Client
OpenVPN
Server
Lab Monitor
AS64511
OpenVPN
Server
Lab Monitor
AS64511
Lab Monitor
AS64511
Lab Monitor
AS64511
Software architecture
Future developments

• Add other types of router for the student
  - Juniper cRIPD, Alcatel/Nokia etc…

• Add RPKI validation to the lab

• Optimise performance of the IRR node under load
Questions?