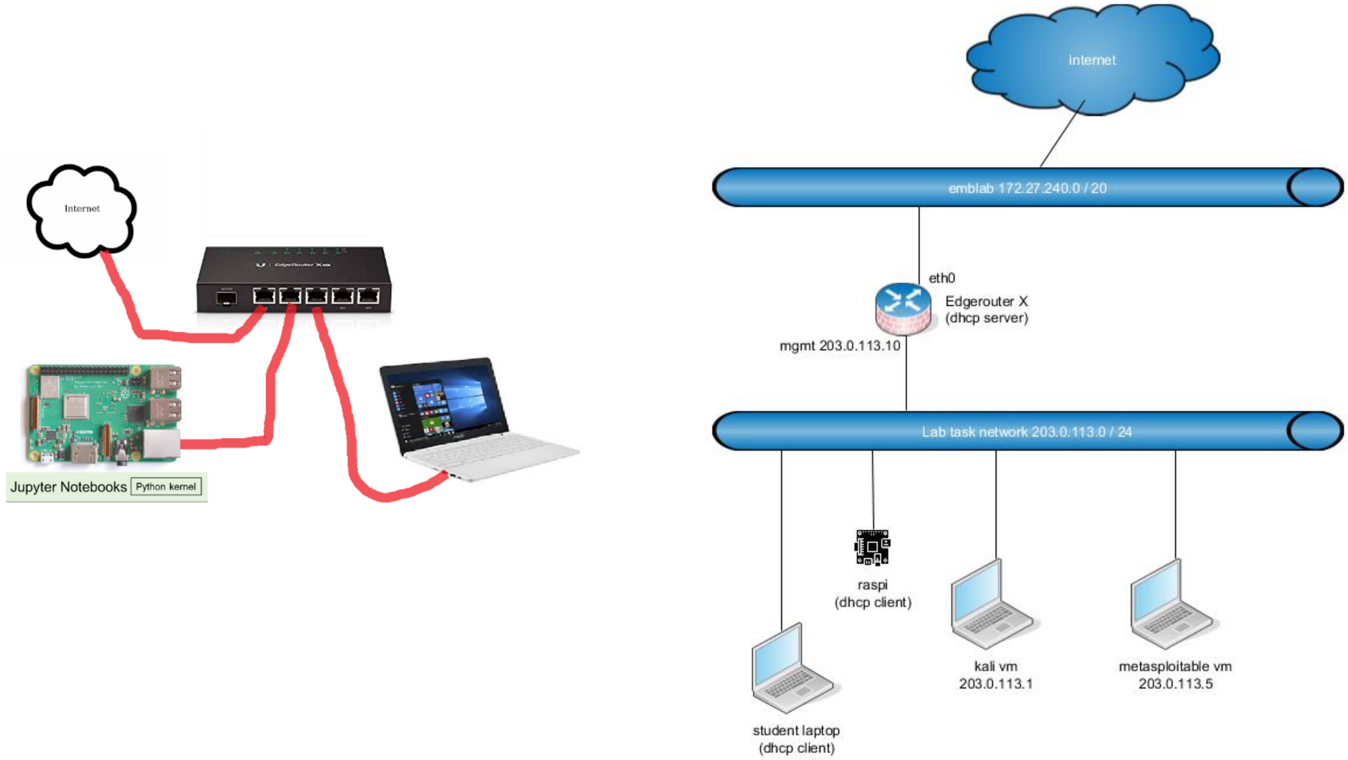
Lab - Packet Crafting to Exploit Unsecured Ports



Topology

Objectives

Part 1: Using hping3 for Port Scanning

Part 2: Crafting Different Types of ICMP Messages

Part 3: Launching DoS Attacks

Background / Scenario

hping3 is a tool used to send custom-crafted TCP/IP packets to a network target in order to elicit a response. Many values in IP packets and TCP headers can be specified in hping3 and the resulting packets sent out on the network. Like Nmap, hping3 can use the TCP header flag fields URG, ACK, PSH, RST, SYN, and FIN to accomplish its scans. It can also craft packets with other protocols such as UDP and ICMP. Unlike Nmap, however, hping3 can use its ability to craft packets to attack a target. hping3 is included in Kali or can be downloaded from http://www.hping.org/. Because hping3 can be used for malicious purposes, avoid using it on production networks unless you have permission to do so.

<table>
<tr><td colspan="4">Source Port</td><td colspan="5">Destination Port</td></tr>
<tr><td colspan="9">Sequence Number</td></tr>
<tr><td colspan="9">Acknowledgement Number</td></tr>
<tr><td>Data Offset</td><td>Reserved</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window</td></tr>
<tr><td colspan="4">Checksum</td><td colspan="5">Urgent Pointer</td></tr>
<tr><td colspan="4">Options</td><td colspan="5">Padding</td></tr>
<tr><td colspan="9">Data</td></tr>
</table>

Required Resources

□ Raspberry Pi 3 Model B or later

□ 8GB Micro SD card (minimum required)

□ PC with IoTSec Kali VM

□ Network connectivity between PC and Raspberry Pi

Part 1: Using hping3 for Port Scanning

a. Set up the topology by connecting the Raspberry Pi to the PC.

b. Start and log into IoTSec Kali VM.

c. Verify that the Kali VM is assigned an IP address on eth0.

```
└─$ ifconfig
```

d. Determine the IP address of your Raspberry Pi.

e. Open the man page for hping3 in Kali VM and review the features and options that are available in hping3.

```
└─$ man hping3
```

f. In a Kali VM terminal, start Wireshark to monitor what hping3 is doing when we are scanning.

```
└─$ wireshark
```

g. In Wireshark, select the eth0 interface in packet capture.

h. You may have captured network traffic that is not relevant to this lab. We are going to restrict the type and of packets we see by using a display filter.

Apply the following filter in Wireshark using IP address of Kali VM as the source address and IP address of your Raspberry Pi as the destination address. In this example, 203.0.113.1 is IP address for Kali VM and 203.0.113.13 is the IP address of your Raspberry Pi.

i. We will first craft packets to do a port scan against the IP address of your Raspberry Pi.

```
└─$ hping3 -8 0-100 -S 203.0.113.13
```

Refer to the Wireshark capture, the man pages, and other sources on the Internet. What do the options 8, 0-100 and -S do?

R: 8 means it is in scan mode, -S is to set the SYN flag and 0-100 means port 0 to 100.

What ports are shown as open?

R: 22 and 80.

j. Expand your scan to include ports up to 1000.

```
└─$ hping3 -8 0-1000 -S 203.0.113.13
```

Did you find any additional ports?

R: No.

What TCP flag was set in the shown in Wireshark?

R: SYN flag.

Part 2: Crafting Different Types of ICMP Messages

ICMP has different message types that we can use to probe a target. For example, message types *8 - echo request* and *0 - echo reply* are used with the TCP/IP tool “ping.” However, if a target is configured not to respond to these ICMP message types, we can use other ICMP message types to attempt to get a response.

a. Open the man page for ICMP in Kali VM and review the features and options that are available in ICMP.

```
└─$ man icmp
```

What is the RFC for ICMP?

R: RFC 792.

b. Start a new Wireshark capture. Click **Continue without Saving** when prompted to save the capture. Apply the same display filter as in the previous part.

c. In the terminal, enter the **hping3** command followed by -1 to scan in ICMP mode. Add the scan target IP address, and enter -C followed by 13 to indicate that ICMP type 13 timestamp request messages should be sent.

```
└─$ hping3 -1 203.0.113.13 -C 13
```

d. Review the Wireshark results and confirm that the ICMP timestamp request packets were sent out. To stop the requests, press Ctrl-C in the Kali VM terminal.

e. Start a new Wireshark capture. Click **Continue without Saving** when prompted to save the capture.

f. Apply the following filter in Wireshark using IP address of Kali VM as the source address and IP address of your Raspberry Pi as the destination address.

```
ip.src == 203.0.113.1 && ip.dst == 203.0.113.13
```

g. Repeat the hping3 command above, but this time send ICMP code 17.

h. Review the Wireshark results. Which ICMP message was sent?

R: Adress mask request.

Part 3: Launching DoS Attacks

Hping3 can launch DoS attacks against ports you found previously in this lab. Using hping3 for this purpose is a good way to test how a network will react to various types of DoS attacks.

a. Start a new Wireshark capture. Click **Continue without Saving** when you are prompted to save the capture. To see two-way TCP traffic from between the Kali VM or the Raspberry Pi, enter only **tcp** as a display filter.

b. In the Kali VM terminal, enter the **hping3** command to send a DoS attack.

```
└─$ hping3 -S 203.0.113.13 -p 88 --flood
```

Looking at Wireshark and the hping3 documentation, what type of TCP messages were sent in this DoS attack? What was the destination TCP port of the attack?

R: SYN messages in port 88.

Look at the source ports that hping3 uses to conduct the DoS flood. How does this scan assign source TCP ports?

R: It starts at one port and increments every time it sends a package.

c. Press Ctrl-C to stop the flood.

d. Start a new Wireshark capture. Click **Continue without Saving** when prompted to save the capture. Display only traffic that has source or destination IP addresses that match the IP address of the Raspberry Pi. (Hint: Edit the ip.src and ip.dest display filter to both use the IP address of the Raspberry Pi. Instead of the && operator, use the || (or) operator.

e. In the Kali VM terminal, enter the **hping3** command to send a DoS Land Attack. This attack sends a packet with the same source IP/port combination as the destination IP/port. In other words, the source IP address is "spoofed" by replacing the Kali VM address another value in the packets.

```
└─$ hping3 -S 203.0.113.13 -a 203.0.113.13 -k -s 89 -p 89 --flood
```

Compare this scan with the SYN flood that you just ran. How were source ports used in this scan? What info does Wireshark report about the packets?

R: The port doesn’t change this time, it uses port 89. It says the tcp port numbers are reused.

f. Press Ctrl-C to stop the flood.

g. Start a new Wireshark capture. Click **Continue without Saving** when prompted to save the capture. Apply the display filter that specifies the Kali VM as the source and the Raspberry Pi as the destination, as was done previously in this lab.

h. In the Kali VM terminal, enter the **hping3** command to send a flood attack.

```
└─$ hping3 --flood —-icmp -p 22 203.0.113.13
```

Look at Wireshark what type of ICMP messages are you seeing?

R: We see echo pings, so type 8.

i. Press Ctrl-C to stop the flood.

j. Complete the following table for the hping3 options that you used in this lab. Use the hping3 man page or other information resources.

Option	Name	Description
-8	Scan	Scan mode, this option expects an argument to describe groups of ports to scan. The port groups are comma separated.
-S	SYN	Set SYN tcp flag
-1	ICMP Mode	ICMP mode, by default hping3 will send ICMP echo-request
-C	ICMP Type	Set the ICMP packet type
--flood	DoS	Send packets as fast as possible, without taking care to show incoming replies.
-a	Spoof hostname	This option is used to set a fake IP source address and ensures that the target will not gain your real address. However, replies will be sent to spoofed address, so you won't be able to see them
-p	Destport	Set PUSH tcp flag
-s	Baseport	Hping3 uses source port to guess replies sequence number. It starts with a base source port number, and increase this number for each packet sent. When the package is received sequence number can be computed as replies.dest.port – base.source.port. Default base source port is random, using this option you are able to set different number.
-k	Keep	Same as -s, but when you need that source port not to be increased for each sent package