IP Source & Destination Spoofing Attacks

There are many cases where we might see irregular traffic for IPv4 and IPv6 packets. In many such cases, this might be done through the source and destination IP fields. We should always consider the following when analyzing these fields for our traffic analysis efforts.

1. `The Source IP Address should always be from our subnet` - If we notice that an incoming packet has an IP source from outside of our local area network, this can be an indicator of packet crafting.
2. `The Source IP for outgoing traffic should always be from our subnet` - If the source IP is from a different IP range than our own local area network, this can be an indicator of malicious traffic that is originating from inside our network.

An attacker might conduct these packet crafting attacks towards the source and destination IP addresses for many different reasons or desired outcomes. Here are a few that we can look for:

1. `Decoy Scanning` - In an attempt to bypass firewall restrictions, an attacker might change the source IP of packets to enumerate further information about a host in another network segment. Through changing the source to something within the same subnet as the target host, the attacker might succeed in firewall evasion.
2. `Random Source Attack DDoS` - Through random source crafting an attacker might be able to send tons of traffic to the same port on the victim host. This in many cases, is used to exhaust resources of our network controls or on the destination host.
3. `LAND Attacks` - LAND Attacks operate similarly to Random Source denial-of-service attacks in the nature that the source address is set to the same as the destination hosts. In doing so the attacker might be able to exhaust network resources or cause crashes on the target host.
4. `SMURF Attacks` - Similar to LAND and Random Source attacks, SMURF attacks work through the attacker sending large amounts of ICMP packets to many different hosts. However, in this case the source address is set to the victim machines, and all of the hosts which receive this ICMP packet respond with an ICMP reply causing resource exhaustion on the crafted source address (victim).
5. `Initialization Vector Generation` - In older wireless networks such as wired equivalent privacy, an attacker might capture, decrypt, craft, and re-inject a packet with a modified source and destination IP address in order to generate initialization vectors to build a decryption table for a statistical attack. These can be seen in nature by noticing an excessive amount of repeated packets between hosts.

It is important to note, that unlike ARP poisoning, the attacks we will be exploring in this section derive from IP layer communications and not ARP poisoning necessarily. However, these attacks tend to be conducted in tandem for most nefarious activities.

Finding Decoy Scanning Attempts

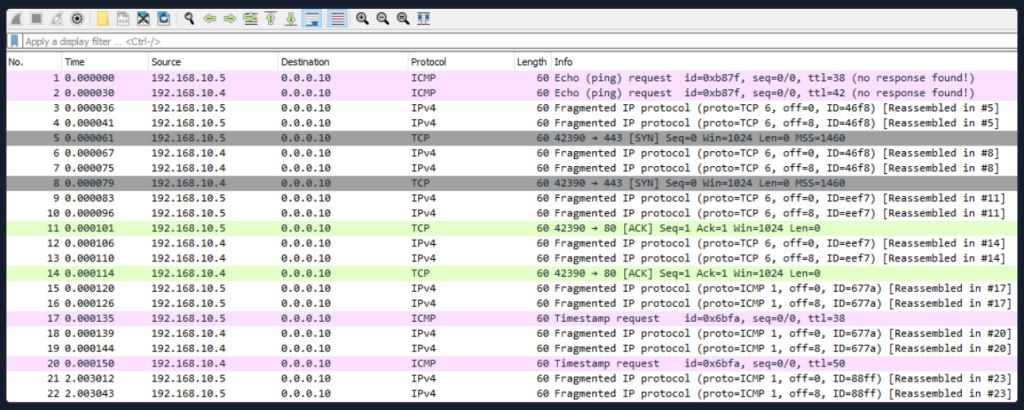
Related PCAP File(s):

- `decoy_scanning_nmap.pcapng`

Simply put, when an attacker wants to gather information, they might change their source address to be the same as another legitimate host, or in some cases entirely different from any real host. This is to attempt to evade IDS/Firewall controls, and it can be easily observed.

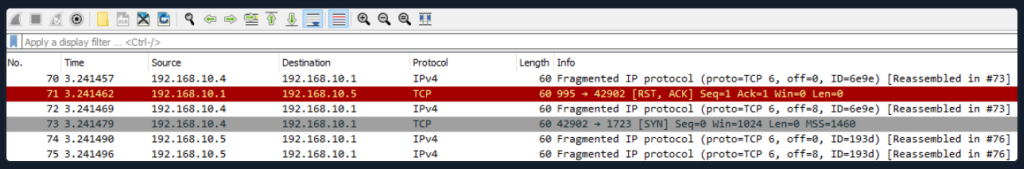
In the case of decoy scanning, we will notice some strange behavior.

1. Initial Fragmentation from a fake address
2. Some TCP traffic from the legitimate source address



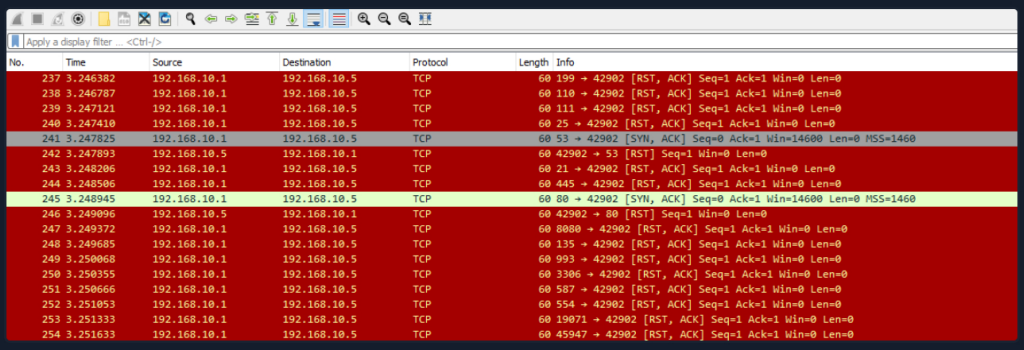
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.5	0.0.0.10	ICMP	60	Echo (ping) request id=0xb87f, seq=0/0, ttl=38 (no response found!)
2	0.000030	192.168.10.4	0.0.0.10	ICMP	60	Echo (ping) request id=0xb87f, seq=0/0, ttl=42 (no response found!)
3	0.000036	192.168.10.5	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=46f8) [Reassembled in #5]
4	0.000041	192.168.10.5	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=46f8) [Reassembled in #5]
5	0.000061	192.168.10.5	0.0.0.10	TCP	60	42390 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
6	0.000067	192.168.10.4	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=46f8) [Reassembled in #8]
7	0.000075	192.168.10.4	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=46f8) [Reassembled in #8]
8	0.000079	192.168.10.4	0.0.0.10	TCP	60	42390 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	0.000083	192.168.10.5	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=eef7) [Reassembled in #11]
10	0.000096	192.168.10.5	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=eef7) [Reassembled in #11]
11	0.000101	192.168.10.5	0.0.0.10	TCP	60	42390 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
12	0.000106	192.168.10.4	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=eef7) [Reassembled in #14]
13	0.000110	192.168.10.4	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=eef7) [Reassembled in #14]
14	0.000114	192.168.10.4	0.0.0.10	TCP	60	42390 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
15	0.000120	192.168.10.5	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=ICMP 1, off=0, ID=677a) [Reassembled in #17]
16	0.000126	192.168.10.5	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=ICMP 1, off=8, ID=677a) [Reassembled in #17]
17	0.000135	192.168.10.5	0.0.0.10	ICMP	60	Timestamp request id=0x6bfa, seq=0/0, ttl=38
18	0.000139	192.168.10.4	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=ICMP 1, off=0, ID=677a) [Reassembled in #20]
19	0.000144	192.168.10.4	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=ICMP 1, off=8, ID=677a) [Reassembled in #20]
20	0.000150	192.168.10.4	0.0.0.10	ICMP	60	Timestamp request id=0x6bfa, seq=0/0, ttl=50
21	2.003012	192.168.10.5	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=ICMP 1, off=0, ID=88ff) [Reassembled in #23]
22	2.003043	192.168.10.5	0.0.0.10	IPv4	60	Fragmented IP protocol (proto=ICMP 1, off=8, ID=88ff) [Reassembled in #23]

Secondarily, in this attack the attacker might be attempting to cloak their address with a decoy, but the responses for multiple closed ports will still be directed towards them with the RST flags denoted for TCP.



No.	Time	Source	Destination	Protocol	Length	Info
70	3.241457	192.168.10.4	192.168.10.1	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=6e9e) [Reassembled in #73]
71	3.241462	192.168.10.1	192.168.10.5	TCP	60	995 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
72	3.241469	192.168.10.4	192.168.10.1	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=6e9e) [Reassembled in #73]
73	3.241479	192.168.10.4	192.168.10.1	TCP	60	42902 → 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
74	3.241490	192.168.10.5	192.168.10.1	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=0, ID=193d) [Reassembled in #76]
75	3.241496	192.168.10.5	192.168.10.1	IPv4	60	Fragmented IP protocol (proto=TCP 6, off=8, ID=193d) [Reassembled in #76]

We will definitely notice this in the case of a large port block which has no services running on the victim host.



No.	Time	Source	Destination	Protocol	Length	Info
237	3.246382	192.168.10.1	192.168.10.5	TCP	60	199 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
238	3.246787	192.168.10.1	192.168.10.5	TCP	60	110 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
239	3.247121	192.168.10.1	192.168.10.5	TCP	60	111 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
240	3.247410	192.168.10.1	192.168.10.5	TCP	60	25 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
241	3.247825	192.168.10.1	192.168.10.5	TCP	60	53 → 42902 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
242	3.247893	192.168.10.5	192.168.10.1	TCP	60	42902 → 53 [RST] Seq=1 Win=0 Len=0
243	3.248206	192.168.10.1	192.168.10.5	TCP	60	21 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
244	3.248506	192.168.10.1	192.168.10.5	TCP	60	445 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
245	3.248945	192.168.10.1	192.168.10.5	TCP	60	80 → 42902 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
246	3.249096	192.168.10.5	192.168.10.1	TCP	60	42902 → 80 [RST] Seq=1 Win=0 Len=0
247	3.249372	192.168.10.1	192.168.10.5	TCP	60	8080 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
248	3.249685	192.168.10.1	192.168.10.5	TCP	60	135 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
249	3.250068	192.168.10.1	192.168.10.5	TCP	60	993 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
250	3.250355	192.168.10.1	192.168.10.5	TCP	60	3306 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
251	3.250666	192.168.10.1	192.168.10.5	TCP	60	587 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
252	3.251053	192.168.10.1	192.168.10.5	TCP	60	554 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
253	3.251333	192.168.10.1	192.168.10.5	TCP	60	19071 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
254	3.251633	192.168.10.1	192.168.10.5	TCP	60	45947 → 42902 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

As such, another simple way that we can prevent this attack beyond just detecting it through our traffic analysis efforts is the following.

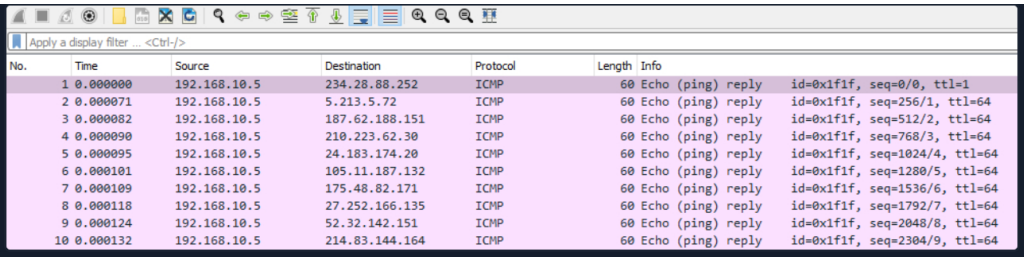
1. Have our IDS/IPS/Firewall act as the destination host would - In the sense that reconstructing the packets gives a clear indication of malicious activity.
2. Watch for connections started by one host, and taken over by another - The attacker after all has to reveal their true source address in order to see that a port is open. This is strange behavior and we can define our rules to prevent it.

Finding Random Source Attacks

Related PCAP File(s):

- ICMP_rand_source.pcapng
- ICMP_rand_source_larg_data.pcapng
- TCP_rand_source_attacks.pcapng

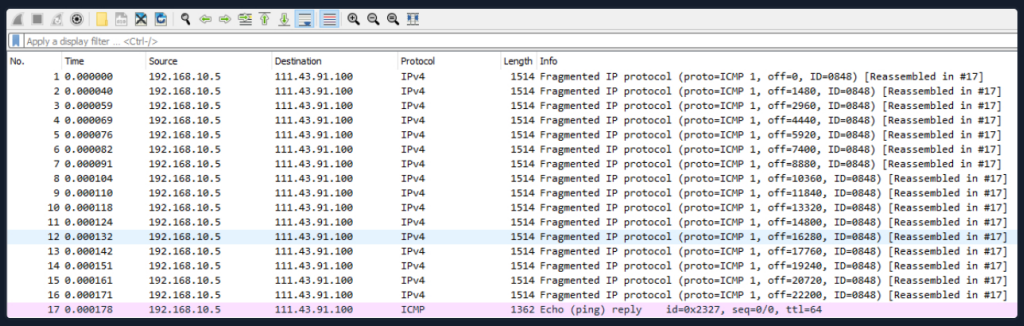
On the opposite side of things, we can begin to explore denial-of-service attacks through source and destination address spoofing. One of the primary and notable examples is random source attacks. These can be conducted in many different flavors. However, notably this can be done like the opposite of a SMURF attack, in which many hosts will ping one host which does not exist, and the pinged host will ping back all others and get no reply.



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.5	234.28.88.252	ICMP	60	Echo (ping) reply id=0x1f1f, seq=0/0, ttl=1
2	0.000071	192.168.10.5	5.213.5.72	ICMP	60	Echo (ping) reply id=0x1f1f, seq=256/1, ttl=64
3	0.000082	192.168.10.5	187.62.188.151	ICMP	60	Echo (ping) reply id=0x1f1f, seq=512/2, ttl=64
4	0.000090	192.168.10.5	210.223.62.30	ICMP	60	Echo (ping) reply id=0x1f1f, seq=768/3, ttl=64
5	0.000095	192.168.10.5	24.183.174.20	ICMP	60	Echo (ping) reply id=0x1f1f, seq=1024/4, ttl=64
6	0.000101	192.168.10.5	105.11.187.132	ICMP	60	Echo (ping) reply id=0x1f1f, seq=1280/5, ttl=64
7	0.000109	192.168.10.5	175.48.82.171	ICMP	60	Echo (ping) reply id=0x1f1f, seq=1536/6, ttl=64
8	0.000118	192.168.10.5	27.252.166.135	ICMP	60	Echo (ping) reply id=0x1f1f, seq=1792/7, ttl=64
9	0.000124	192.168.10.5	52.32.142.151	ICMP	60	Echo (ping) reply id=0x1f1f, seq=2048/8, ttl=64
10	0.000132	192.168.10.5	214.83.144.164	ICMP	60	Echo (ping) reply id=0x1f1f, seq=2304/9, ttl=64

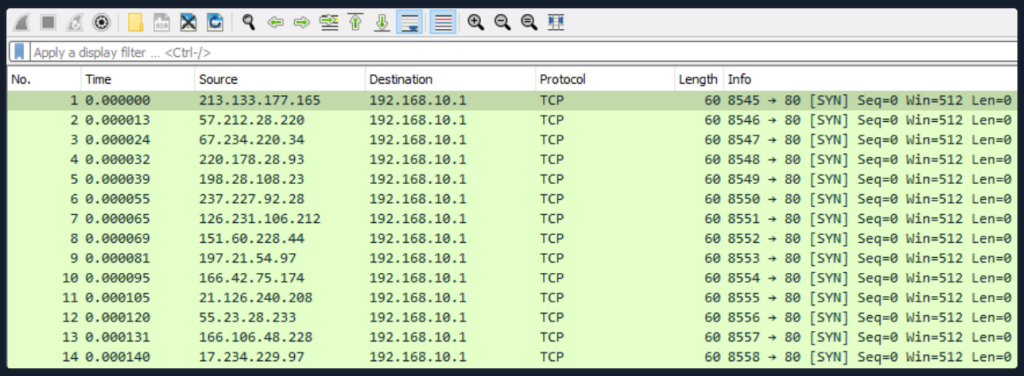
We should also consider that attackers might fragment these random hosts communications in order to draw out more resource exhaustion.



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=0848) [Reassembled in #17]
2	0.000040	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=0848) [Reassembled in #17]
3	0.000059	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=0848) [Reassembled in #17]
4	0.000069	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=0848) [Reassembled in #17]
5	0.000076	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=5920, ID=0848) [Reassembled in #17]
6	0.000082	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=7400, ID=0848) [Reassembled in #17]
7	0.000091	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=8880, ID=0848) [Reassembled in #17]
8	0.000104	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=10360, ID=0848) [Reassembled in #17]
9	0.000110	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=11840, ID=0848) [Reassembled in #17]
10	0.000118	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=13320, ID=0848) [Reassembled in #17]
11	0.000124	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=14800, ID=0848) [Reassembled in #17]
12	0.000132	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=16280, ID=0848) [Reassembled in #17]
13	0.000142	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=17760, ID=0848) [Reassembled in #17]
14	0.000151	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=19240, ID=0848) [Reassembled in #17]
15	0.000161	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=20720, ID=0848) [Reassembled in #17]
16	0.000171	192.168.10.5	111.43.91.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=22200, ID=0848) [Reassembled in #17]
17	0.000178	192.168.10.5	111.43.91.100	ICMP	1362	Echo (ping) reply id=0x2327, seq=0/0, ttl=64

However in many cases, like LAND attacks, these attacks will be used by attackers to exhaust resources to one specific service on a port. Instead of spoofing the source address to be the same as the destination, the attacker might randomize them. We might notice the following.



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	213.133.177.165	192.168.10.1	TCP	60	8545 → 80 [SYN] Seq=0 Win=512 Len=0
2	0.000013	57.212.28.220	192.168.10.1	TCP	60	8546 → 80 [SYN] Seq=0 Win=512 Len=0
3	0.000024	67.234.220.34	192.168.10.1	TCP	60	8547 → 80 [SYN] Seq=0 Win=512 Len=0
4	0.000032	220.178.28.93	192.168.10.1	TCP	60	8548 → 80 [SYN] Seq=0 Win=512 Len=0
5	0.000039	198.28.108.23	192.168.10.1	TCP	60	8549 → 80 [SYN] Seq=0 Win=512 Len=0
6	0.000055	237.227.92.28	192.168.10.1	TCP	60	8550 → 80 [SYN] Seq=0 Win=512 Len=0
7	0.000065	126.231.106.212	192.168.10.1	TCP	60	8551 → 80 [SYN] Seq=0 Win=512 Len=0
8	0.000069	151.60.228.44	192.168.10.1	TCP	60	8552 → 80 [SYN] Seq=0 Win=512 Len=0
9	0.000081	197.21.54.97	192.168.10.1	TCP	60	8553 → 80 [SYN] Seq=0 Win=512 Len=0
10	0.000095	166.42.75.174	192.168.10.1	TCP	60	8554 → 80 [SYN] Seq=0 Win=512 Len=0
11	0.000105	21.126.240.208	192.168.10.1	TCP	60	8555 → 80 [SYN] Seq=0 Win=512 Len=0
12	0.000120	55.23.28.233	192.168.10.1	TCP	60	8556 → 80 [SYN] Seq=0 Win=512 Len=0
13	0.000131	166.106.48.228	192.168.10.1	TCP	60	8557 → 80 [SYN] Seq=0 Win=512 Len=0
14	0.000140	17.234.229.97	192.168.10.1	TCP	60	8558 → 80 [SYN] Seq=0 Win=512 Len=0

In this case, we have a few indicators of nefarious behavior:

1. Single Port Utilization from random hosts
2. Incremental Base Port with a lack of randomization
3. Identical Length Fields

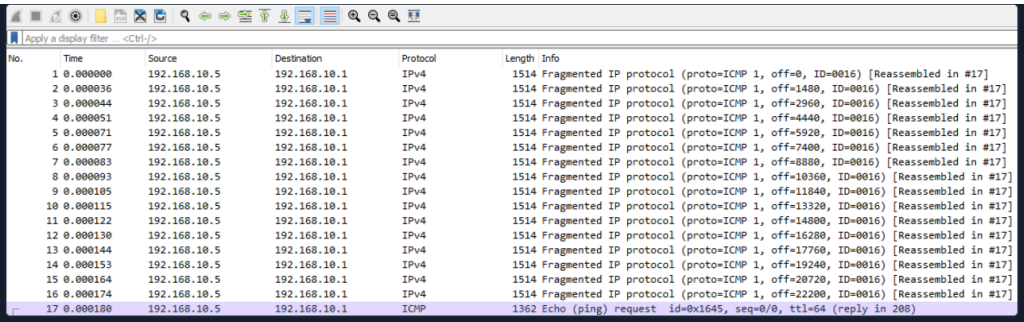
In many real world cases, like a web server, we may have many different users utilizing the same port. However, these requests are contrary of our indicators. Such that they will have different lengths and the base ports will not exhibit this behavior.

Finding Smurf Attacks

SMURF Attacks are a notable distributed denial-of-service attack, in the nature that they operate through causing random hosts to ping the victim host back. Simply put, an attacker conducts these like the following:

1. The attacker will send an ICMP request to live hosts with a spoofed address of the victim host
2. The live hosts will respond to the legitimate victim host with an ICMP reply
3. This may cause resource exhaustion on the victim host

One of the things we can look for in our traffic behavior is an excessive amount of ICMP replies from a single host to our affected host. Sometimes attackers will include fragmentation and data on these ICMP requests to make the traffic volume larger.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=0016) [Reassembled in #17]
2	0.000036	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=0016) [Reassembled in #17]
3	0.000044	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=0016) [Reassembled in #17]
4	0.000051	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=0016) [Reassembled in #17]
5	0.000071	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=5920, ID=0016) [Reassembled in #17]
6	0.000077	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=7400, ID=0016) [Reassembled in #17]
7	0.000083	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=8880, ID=0016) [Reassembled in #17]
8	0.000093	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=10360, ID=0016) [Reassembled in #17]
9	0.000105	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=11840, ID=0016) [Reassembled in #17]
10	0.000115	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=13320, ID=0016) [Reassembled in #17]
11	0.000122	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=14800, ID=0016) [Reassembled in #17]
12	0.000130	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=16280, ID=0016) [Reassembled in #17]
13	0.000144	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=17760, ID=0016) [Reassembled in #17]
14	0.000153	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=19240, ID=0016) [Reassembled in #17]
15	0.000164	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=20720, ID=0016) [Reassembled in #17]
16	0.000174	192.168.10.5	192.168.10.1	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=22200, ID=0016) [Reassembled in #17]
17	0.000180	192.168.10.5	192.168.10.1	ICMP	1362	Echo (ping) request id=0x1645, seq=0/0, ttl=64 (reply in 208)

We might notice many different hosts pinging our single host, and in this case it represents the basic nature of SMURF attacks.

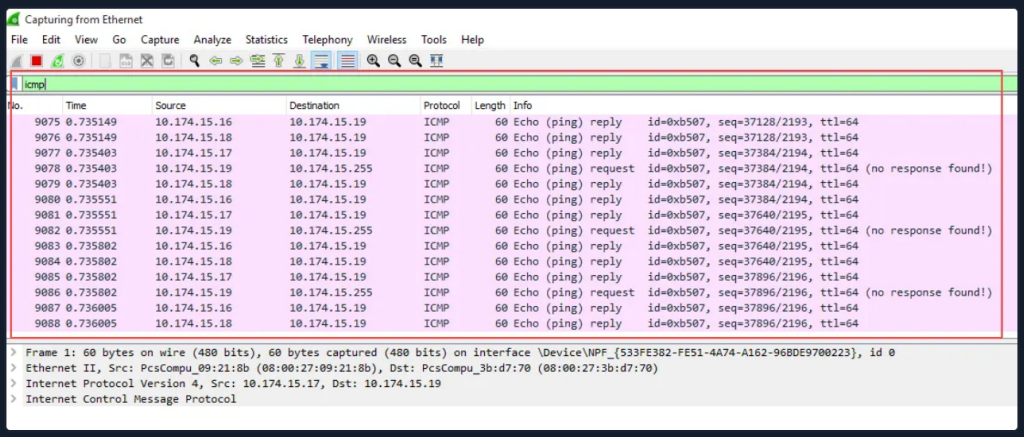


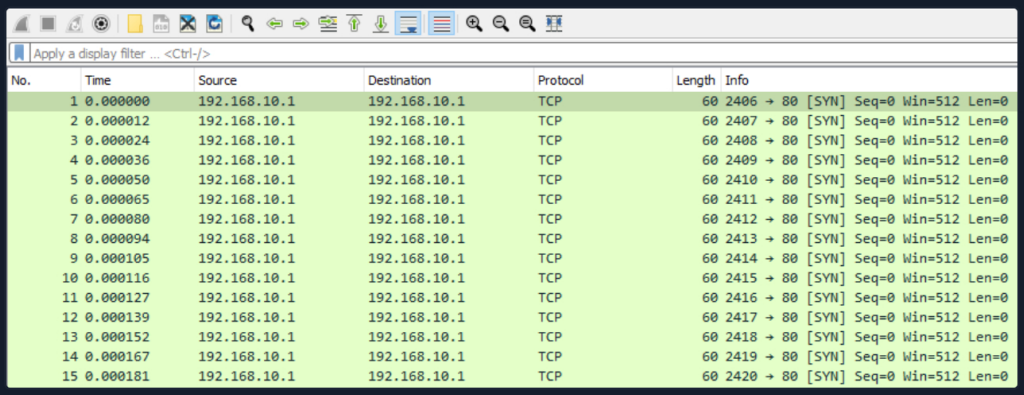
Image From: https://techofide.com/blogs/what-is-smurf-attack-what-is-the-denial-of-service-attack-practical-ddos-attack-step-by-step-guide/

Finding LAND Attacks

Related PCAP File(s):

- `LAND-DoS.pcapng`

LAND attacks operate through an attacker spoofing the source IP address to be the same as the destination. These denial-of-service attacks work through sheer volume of traffic and port re-use. Essentially, if all base ports are occupied, it makes real connections much more difficult to establish to our affected host.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.10.1	192.168.10.1	TCP	60	2406 → 80 [SYN] Seq=0 Win=512 Len=0
2	0.000012	192.168.10.1	192.168.10.1	TCP	60	2407 → 80 [SYN] Seq=0 Win=512 Len=0
3	0.000024	192.168.10.1	192.168.10.1	TCP	60	2408 → 80 [SYN] Seq=0 Win=512 Len=0
4	0.000036	192.168.10.1	192.168.10.1	TCP	60	2409 → 80 [SYN] Seq=0 Win=512 Len=0
5	0.000050	192.168.10.1	192.168.10.1	TCP	60	2410 → 80 [SYN] Seq=0 Win=512 Len=0
6	0.000065	192.168.10.1	192.168.10.1	TCP	60	2411 → 80 [SYN] Seq=0 Win=512 Len=0
7	0.000080	192.168.10.1	192.168.10.1	TCP	60	2412 → 80 [SYN] Seq=0 Win=512 Len=0
8	0.000094	192.168.10.1	192.168.10.1	TCP	60	2413 → 80 [SYN] Seq=0 Win=512 Len=0
9	0.000105	192.168.10.1	192.168.10.1	TCP	60	2414 → 80 [SYN] Seq=0 Win=512 Len=0
10	0.000116	192.168.10.1	192.168.10.1	TCP	60	2415 → 80 [SYN] Seq=0 Win=512 Len=0
11	0.000127	192.168.10.1	192.168.10.1	TCP	60	2416 → 80 [SYN] Seq=0 Win=512 Len=0
12	0.000139	192.168.10.1	192.168.10.1	TCP	60	2417 → 80 [SYN] Seq=0 Win=512 Len=0
13	0.000152	192.168.10.1	192.168.10.1	TCP	60	2418 → 80 [SYN] Seq=0 Win=512 Len=0
14	0.000167	192.168.10.1	192.168.10.1	TCP	60	2419 → 80 [SYN] Seq=0 Win=512 Len=0
15	0.000181	192.168.10.1	192.168.10.1	TCP	60	2420 → 80 [SYN] Seq=0 Win=512 Len=0

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!

+ 1 Inspect the ICMP_smurf.pcapng file, part of this module's resources, and enter the total number of attacking hosts as your answer.

1

Submit

Previous Next

Mark Complete & Next