

Assignment Series #A1 Spoofer

Compile your first Software in Linux (caida spoofer tool)

1. Task description

The receipt of compiling a software with automake is mostly the same:

1. `gzip -d project.tar.gz`
2. `tar -xvf project.tar`
3. `cd project`
4. `configure --prefix=/opt/applic/project`
5. `make`
6. `make install`

After you've downloaded, built and run the spoofer tool from scratch, please start Wireshark and answer the following questions:

1. Explain the idea of IP spoofing
2. Is this a realistic attack over the internet with TCP/IP - explain
3. Is this a realistic attack over the internet with UDP/IP - explain
4. What are the destination addresses of the spoofer tool (see in Wireshark)
5. What protocol is the spoofer tool trying to spoof?
6. Do you find your own results on https://spoofer.caida.org/recent_tests.php

2. Answers

1. The idea behind IP spoofing is the creation of Internet Protocol (IP) packets which have a modified source address in order to either hide the identity of the sender, to impersonate another computer system, or both. It is a technique often used by bad actors to invoke DDoS attacks against a target device or the surrounding infrastructure.
2. Not really. Direct access to traffic is now much more difficult if the intruder's computer is not on the same subnet. This is due to the fact that intercepting a data packet is only possible with the help of the corresponding packet sequence number - a task that is almost impossible today compared to earlier days from outside. In the past, operating systems and network devices still generated these process numbers entered in the TCP header according to a pattern that was always the same. As today's systems output the sequence numbers randomly, these so-called TCP sequence prediction attacks (also called blind spoofing) have basically become ineffective - however, older devices are still at risk.
3. I'd say yes, because UDP doesn't use a `three way handshake` or any `sequence numbers` to establish a connection.

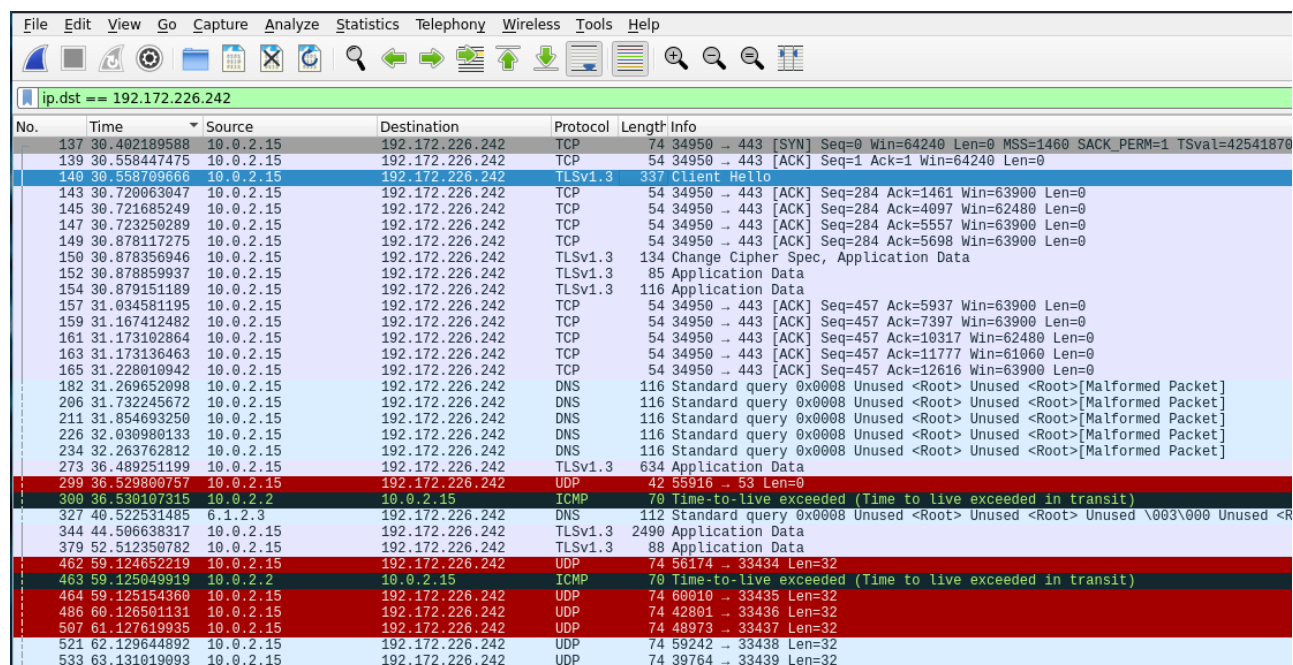
4. The spoofer tool communicates with the ip address 192.172.226.242

```

root@hlkali:/opt/applic/spoofers/bin# netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 10.0.2.15:36498        192.172.226.242:443    ESTABLISHED 32467/spoofers-probe
tcp        0      0 10.0.2.15:45516        34.216.82.69:443      ESTABLISHED 4365/firefox-esr
tcp        0      1 10.0.2.15:56842        136.144.56.255:80     SYN_SENT    32611/curl
root@hlkali:/opt/applic/spoofers/bin#

```

In Wireshark it looks like this (filter `ip.dst == 192.172.226.242`)



No.	Time	Source	Destination	Protocol	Length	Info
137	30.402189588	10.0.2.15	192.172.226.242	TCP	74	34950 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=42541870
139	30.558447475	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
140	30.558709666	10.0.2.15	192.172.226.242	TLSv1.3	337	Client Hello
143	30.720063047	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=284 Ack=1461 Win=63900 Len=0
145	30.721685249	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=284 Ack=4097 Win=62480 Len=0
147	30.723250289	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=284 Ack=5557 Win=63900 Len=0
149	30.878117275	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=284 Ack=5698 Win=63900 Len=0
150	30.878356946	10.0.2.15	192.172.226.242	TLSv1.3	134	Change Cipher Spec, Application Data
152	30.878859937	10.0.2.15	192.172.226.242	TLSv1.3	85	Application Data
154	30.879151189	10.0.2.15	192.172.226.242	TLSv1.3	116	Application Data
157	31.034581195	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=457 Ack=5937 Win=63900 Len=0
159	31.167412482	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=457 Ack=7397 Win=63900 Len=0
161	31.173102864	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=457 Ack=10317 Win=62480 Len=0
163	31.173136463	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=457 Ack=11777 Win=61060 Len=0
165	31.228010942	10.0.2.15	192.172.226.242	TCP	54	34950 → 443 [ACK] Seq=457 Ack=12616 Win=63900 Len=0
182	31.269652098	10.0.2.15	192.172.226.242	DNS	116	Standard query 0x0008 Unused <Root> Unused <Root> [Malformed Packet]
206	31.732245672	10.0.2.15	192.172.226.242	DNS	116	Standard query 0x0008 Unused <Root> Unused <Root> [Malformed Packet]
211	31.854693250	10.0.2.15	192.172.226.242	DNS	116	Standard query 0x0008 Unused <Root> Unused <Root> [Malformed Packet]
226	32.030980133	10.0.2.15	192.172.226.242	DNS	116	Standard query 0x0008 Unused <Root> Unused <Root> [Malformed Packet]
234	32.263762812	10.0.2.15	192.172.226.242	DNS	116	Standard query 0x0008 Unused <Root> Unused <Root> [Malformed Packet]
273	36.489251199	10.0.2.15	192.172.226.242	TLSv1.3	634	Application Data
299	36.529800757	10.0.2.15	192.172.226.242	UDP	42	55916 → 53 Len=0
300	36.530107315	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
327	40.522531485	6.1.2.3	192.172.226.242	DNS	112	Standard query 0x0008 Unused <Root> Unused <Root> Unused \003\000 Unused <R
344	44.506638317	10.0.2.15	192.172.226.242	TLSv1.3	2490	Application Data
379	52.512350782	10.0.2.15	192.172.226.242	TLSv1.3	88	Application Data
462	59.124652219	10.0.2.15	192.172.226.242	UDP	74	56174 → 33434 Len=32
463	59.125049919	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
464	59.125154360	10.0.2.15	192.172.226.242	UDP	74	60010 → 33435 Len=32
486	60.126501131	10.0.2.15	192.172.226.242	UDP	74	42801 → 33436 Len=32
507	61.127619935	10.0.2.15	192.172.226.242	UDP	74	48973 → 33437 Len=32
521	62.129644892	10.0.2.15	192.172.226.242	UDP	74	59242 → 33438 Len=32
533	63.131019093	10.0.2.15	192.172.226.242	UDP	74	39764 → 33439 Len=32

To get a list of all destination addresses I use the **Statistics --> Conversations** menu:

Ethernet · 2		IPv4 · 48	IPv6	TCP · 28	UDP · 223		
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes
10.0.2.15	35417	195.148.124.66	53	5	580	5	
10.0.2.15	38033	193.1.193.136	53	5	580	5	
10.0.2.15	45281	192.172.226.247	53	5	580	5	
10.0.2.15	42869	78.41.116.2	53	5	580	5	
10.0.2.15	50616	203.181.248.51	53	5	580	5	
10.0.2.15	50992	204.235.64.14	53	5	580	5	
10.0.2.15	56949	205.166.205.222	53	5	580	5	
10.0.2.15	59952	192.107.171.130	53	5	580	5	
10.0.2.15	37138	205.189.33.78	53	5	580	5	
10.0.2.15	54676	128.232.97.9	53	5	580	5	
10.0.2.15	45766	128.223.157.13	53	5	580	5	
10.0.2.15	54430	139.18.1.244	53	5	580	5	
10.0.2.15	36038	138.44.13.110	53	5	580	5	
10.0.2.15	57307	192.136.136.30	53	5	580	5	
10.0.2.15	58669	192.172.226.242	53	5	580	5	
10.0.2.15	57297	172.16.16.16	53	4	380	2	
10.0.2.15	48812	205.166.205.222	53	1	42	1	
10.0.2.15	49821	130.206.158.142	53	1	42	1	
10.0.2.15	33867	78.41.116.2	53	1	42	1	
10.0.2.15	50931	203.181.248.51	53	1	42	1	
10.0.2.15	38593	128.223.157.13	53	1	42	1	
10.0.2.15	52906	192.107.171.130	53	1	42	1	
10.0.2.15	38535	192.42.115.98	53	1	42	1	
10.0.2.15	32831	195.148.124.66	53	1	42	1	
10.0.2.15	40333	192.172.226.247	53	1	42	1	
10.0.2.15	36142	128.232.97.9	53	1	42	1	
10.0.2.15	33830	205.189.33.78	53	1	42	1	

5. It's the UDP Protocol. The methodology says:

The spoofer program attempts to send a series of spoofed UDP packets to servers distributed throughout the world. These packets are designed to test:

- Different classes of spoofed IPv4 and IPv6 addresses, including private and routable
- Ability to spoof neighboring, adjacent addresses
- Ability to spoof inbound (towards the client) and outbound (from the client)
- Where along the path filtering is observed
- Presence of a NAT device along the path

6. Yes. <https://spoofer.caida.org/report.php?sessionid=1014179>

Session	Timestamp (UTC)	Client IP Block	ASN	Country	NAT	Outbound Private Status	Outbound Routable Status	Adj Spoof Prefix Len	Results
1014179	2020-10-25 11:22:05	109.106.59.x/24	15600 (FINECOM)	che (Switzerland)	yes	unknown	rewritten	none	Report
1014126	2020-10-25 09:01:05	85.195.251.x/24	13030 (INIT7)	che (Switzerland)	yes	rewritten	rewritten	none	Report
1013902	2020-10-24 22:24:49	109.106.59.x/24	15600 (FINECOM)	che (Switzerland)	yes	unknown	rewritten	none	Report
1013870	2020-10-24 21:02:40	85.195.251.x/24	13030 (INIT7)	che (Switzerland)	yes	rewritten	rewritten	none	Report
1013865	2020-10-24 20:53:26	109.106.59.x/24	15600 (FINECOM)	che (Switzerland)	yes	unknown	rewritten	none	Report
1013376	2020-10-23 21:52:45	77.56.24.x/24	6830 (LGI-UPC)	che (Switzerland)	yes	unknown	rewritten	none	Report
1013369	2020-10-23 21:33:01	77.56.24.x/24	6830 (LGI-UPC)	che (Switzerland)	yes	unknown	rewritten	none	Report
1013365	2020-10-23 21:30:08	77.56.24.x/24	6830 (LGI-UPC)	che (Switzerland)	yes	unknown	rewritten	none	Report
1013362	2020-10-23 21:23:14	77.56.24.x/24	6830 (LGI-UPC)	che (Switzerland)	yes	unknown	rewritten	none	Report