Лабораторна робота №1
Основи комп’ютерних мереж

Виконав:
Студент 3 курсу ФТІ
групи ФІ-92
Поночевний Назар Юрійович
Версія тестової лабораторії: v1.0

Завдання:
Метою даної лабораторної роботи є отримання розуміння роботи базових мережних протоколів канального та мережного рівня (Ethernet, ARP, ICMP, IP), отримання розуміння роботи атаки ARP-spoofing, а також отримання навичок роботи з розповсюдженими утилітами для аналізу і конфігурування у комп’ютерних мережах: ifconfig, ip, netstat, ping, tcpdump, wireshark.

1) Отримайте SSH доступ до машини client_pc1

2) Зберіть інформацію про налаштування мережі, адреси, маршрути тощо

```
root@fd9504d8d51c:~# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.238.10  netmask 255.255.255.0  broadcast 172.16.238.255
        ether ea:e7:b2:4e:88:58  txqueuelen 0  (Ethernet)
        RX packets 2613  bytes 11163970 (11.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2765  bytes 493495 (493.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 243  bytes 32515 (32.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 243  bytes 32515 (32.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@fd9504d8d51c:~#
```

3) Перейдіть за SSH на вузол https-portal (172.11.15.71). Очистіть arp кеш (та упевніться в цьому) та спробуйте пінгувати вузли corp_gateway, corp1, corp2, http_server, ftp_server, проаналізуйте зміни в arp кеші та дайте оцінку отриманому результату

```
root@fd9504d8d51c:~# ssh root@172.11.15.71
The authenticity of host '172.11.15.71 (172.11.15.71)' can't be established.
ED25519 key fingerprint is SHA256:ESk7GQtbCjvBZ0nUNBUr/R0qdnqOZNcG2MEk7h3p95I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.11.15.71' (ED25519) to the list of known hosts.
root@172.11.15.71's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@55df5f83c1c8:~# arp -e
Address                  HWtype  HWaddress           Flags Mask            Iface
corp1.ftilabs_comnetwor  ether   02:42:ac:0b:0f:11   C                     eth0
corp2.ftilabs_comnetwor  ether   02:42:ac:0b:0f:21   C                     eth0
ftilabs_comnetworks_wor  ether   02:42:0a:0a:0f:0f   C                     eth1
dockerhost                       (incomplete)                              eth0
root@55df5f83c1c8:~# ip -s neigh flush all

*** Round 1, deleting 3 entries ***

*** Round 2, deleting 1 entries ***

*** Round 3, deleting 1 entries ***

*** Round 4, deleting 1 entries ***
```

```
*** Round 4, deleting 1 entries ***

*** Round 5, deleting 1 entries ***

*** Round 6, deleting 1 entries ***

*** Round 7, deleting 1 entries ***

*** Round 8, deleting 1 entries ***

*** Round 9, deleting 1 entries ***

*** Round 10, deleting 1 entries ***
*** Flush not complete bailing out after 10 rounds
root@55df5f83c1c8:~# arp -e
Address                  HWtype  HWaddress           Flags Mask            Iface
corp1.ftilabs_comnetwor  ether   02:42:ac:0b:0f:11   C                     eth0
corp2.ftilabs_comnetwor  ether   02:42:ac:0b:0f:21   C                     eth0
ftilabs_comnetworks_wor  ether   02:42:0a:0a:0f:0f   C                     eth1
dockerhost                       (incomplete)                              eth0
root@55df5f83c1c8:~#
```

```
root@55df5f83c1c8:~# ping -c 4 172.11.15.220
PING 172.11.15.220 (172.11.15.220): 56 data bytes
64 bytes from 172.11.15.220: icmp_seq=0 ttl=64 time=0.087 ms
64 bytes from 172.11.15.220: icmp_seq=1 ttl=64 time=0.068 ms
64 bytes from 172.11.15.220: icmp_seq=2 ttl=64 time=0.067 ms
64 bytes from 172.11.15.220: icmp_seq=3 ttl=64 time=0.068 ms
--- 172.11.15.220 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.067/0.073/0.087/0.000 ms
root@55df5f83c1c8:~# ping -c 4 172.11.15.53
PING 172.11.15.53 (172.11.15.53): 56 data bytes
64 bytes from 172.11.15.53: icmp_seq=0 ttl=64 time=0.117 ms
64 bytes from 172.11.15.53: icmp_seq=1 ttl=64 time=0.098 ms
64 bytes from 172.11.15.53: icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from 172.11.15.53: icmp_seq=3 ttl=64 time=0.066 ms
--- 172.11.15.53 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.066/0.089/0.117/0.000 ms
root@55df5f83c1c8:~# ping -c 4 172.11.15.61
PING 172.11.15.61 (172.11.15.61): 56 data bytes
64 bytes from 172.11.15.61: icmp_seq=0 ttl=64 time=0.083 ms
64 bytes from 172.11.15.61: icmp_seq=1 ttl=64 time=0.066 ms
64 bytes from 172.11.15.61: icmp_seq=2 ttl=64 time=0.099 ms
64 bytes from 172.11.15.61: icmp_seq=3 ttl=64 time=0.102 ms
--- 172.11.15.61 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.066/0.088/0.102/0.000 ms
```

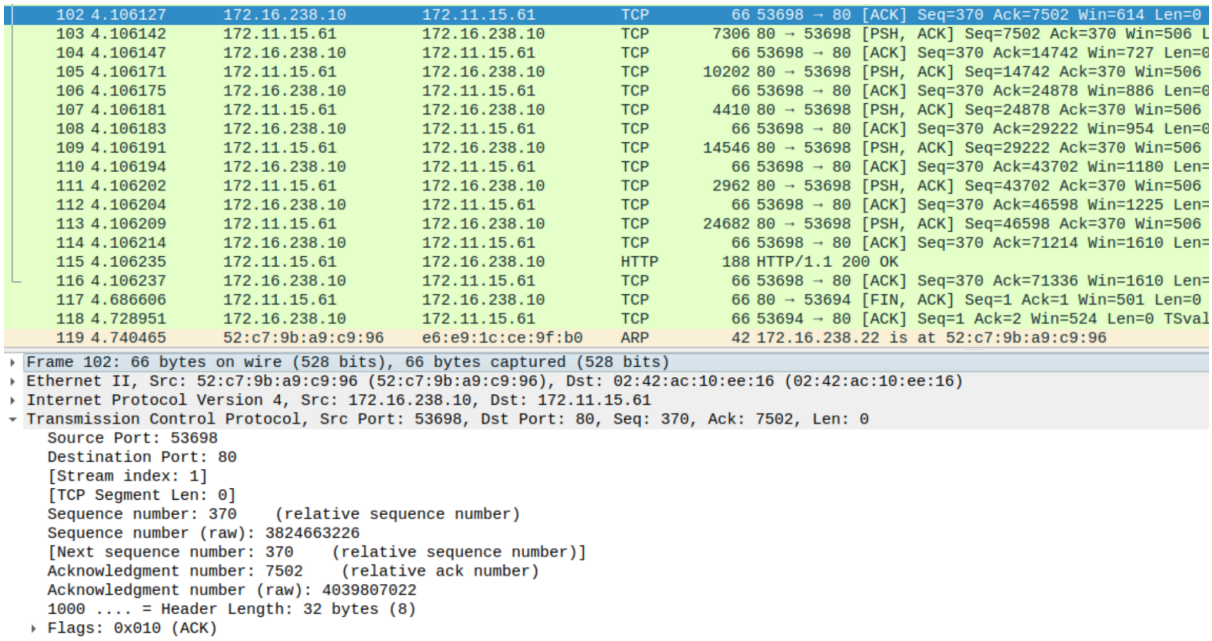
```
root@55df5f83c1c8:~# ping -c 4 172.11.15.17
PING 172.11.15.17 (172.11.15.17): 56 data bytes
64 bytes from 172.11.15.17: icmp_seq=0 ttl=64 time=0.048 ms
64 bytes from 172.11.15.17: icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from 172.11.15.17: icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from 172.11.15.17: icmp_seq=3 ttl=64 time=0.074 ms
--- 172.11.15.17 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.048/0.063/0.074/0.000 ms
root@55df5f83c1c8:~# ping -c 4 172.11.15.33
PING 172.11.15.33 (172.11.15.33): 56 data bytes
64 bytes from 172.11.15.33: icmp_seq=0 ttl=64 time=0.052 ms
64 bytes from 172.11.15.33: icmp_seq=1 ttl=64 time=0.069 ms
64 bytes from 172.11.15.33: icmp_seq=2 ttl=64 time=0.064 ms
64 bytes from 172.11.15.33: icmp_seq=3 ttl=64 time=0.065 ms
--- 172.11.15.33 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.052/0.062/0.069/0.000 ms
root@55df5f83c1c8:~#
```

```
root@55df5f83c1c8:~# arp -e
Address                  HWtype  HWaddress           Flags Mask            Iface
corp1.ftilabs_comnetwor  ether   02:42:ac:0b:0f:11   C                     eth0
corp2.ftilabs_comnetwor  ether   02:42:ac:0b:0f:21   C                     eth0
ftp_server.ftilabs_comn  ether   02:42:ac:0b:0f:35   C                     eth0
ftilabs_comnetworks_wor  ether   02:42:0a:0a:0f:0f   C                     eth1
corp_gateway.ftilabs_co  ether   02:42:ac:0b:0f:dc   C                     eth0
http_server.ftilabs_com  ether   02:42:ac:0b:0f:3d   C                     eth0
dockerhost                       (incomplete)                              eth0
root@55df5f83c1c8:~#
```

4) Відкрийте дві SSH сесії на client_pc1. В одній з них запустіть tcpdump, а у іншій – очистіть arp кеш та спробуйте пінгувати сусідні машини client_pc2-client_pc4. Вивантажте дамп трафіку та проаналізуйте за допомогою wireshark. Відфільтруйте трафік за ICMP та ARP. Опишіть функціонування цих протоколів на прикладі перехоплених пакетів

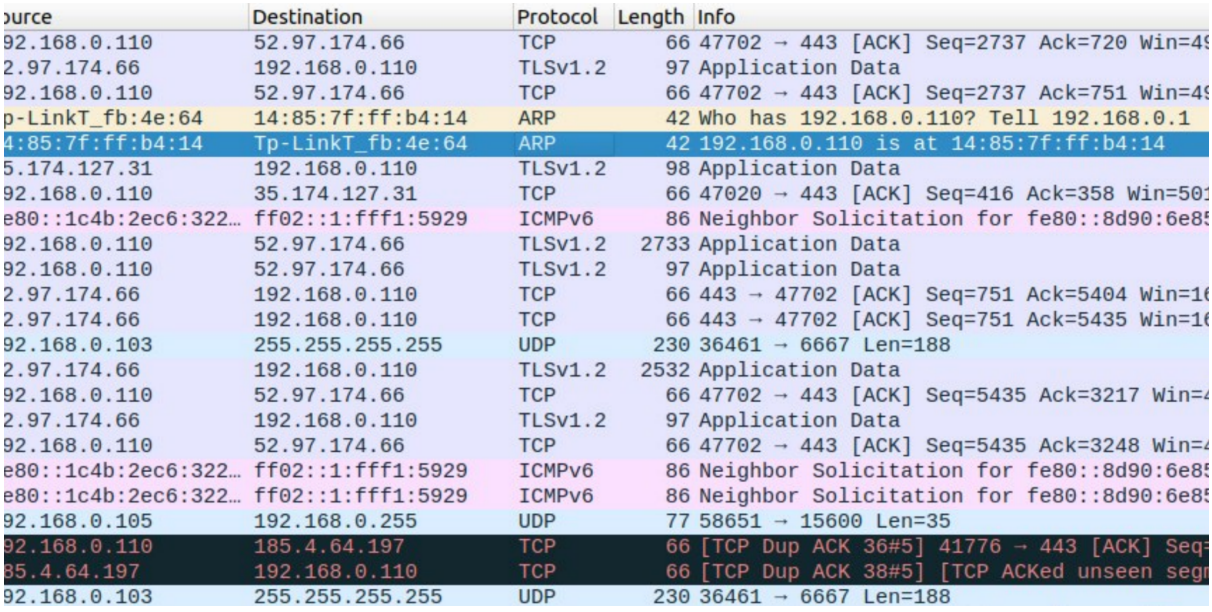
	Time	Source	Destination	Protocol	Length	Info
1	0.000000	d6:db:84:e4:8d:90	Broadcast	ARP	42	Who has 172.16.238.30? Tell 172.16.238.10
2	0.000024	172.16.238.10	172.16.238.30	ICMP	98	Echo (ping) request id=0x0060, seq=0/0, ttl=64 (no re
3	1.001143	172.16.238.10	172.16.238.30	ICMP	98	Echo (ping) request id=0x0060, seq=1/256, ttl=64 (no
4	2.002382	172.16.238.10	172.16.238.30	ICMP	98	Echo (ping) request id=0x0060, seq=2/512, ttl=64 (no
5	3.003074	172.16.238.10	172.16.238.30	ICMP	98	Echo (ping) request id=0x0060, seq=3/768, ttl=64 (no
6	5.139370	d6:db:84:e4:8d:90	56:5c:a5:92:9b:35	ARP	42	172.16.238.10 is at d6:db:84:e4:8d:90
7	8.900051	172.16.238.10	172.24.118.16	ICMP	98	Echo (ping) request id=0x0061, seq=0/0, ttl=64 (no re
8	9.901208	172.16.238.10	172.24.118.16	ICMP	98	Echo (ping) request id=0x0061, seq=1/256, ttl=64 (no
9	10.902483	172.16.238.10	172.24.118.16	ICMP	98	Echo (ping) request id=0x0061, seq=2/512, ttl=64 (no
10	11.903404	172.16.238.10	172.24.118.16	ICMP	98	Echo (ping) request id=0x0061, seq=3/768, ttl=64 (no
11	31.518178	172.16.238.10	172.24.118.15	ICMP	98	Echo (ping) request id=0x0062, seq=0/0, ttl=64 (no re
12	32.519356	172.16.238.10	172.24.118.15	ICMP	98	Echo (ping) request id=0x0062, seq=1/256, ttl=64 (no
13	33.520633	172.16.238.10	172.24.118.15	ICMP	98	Echo (ping) request id=0x0062, seq=2/512, ttl=64 (no
14	34.521886	172.16.238.10	172.24.118.15	ICMP	98	Echo (ping) request id=0x0062, seq=3/768, ttl=64 (no

5) Відкрийте у дампі будь-який TCP пакет та проаналізуйте кожен рівень стеку TCP/IP: опишіть на прикладі перехопленого пакета, які поля містяться у заголовках кожного рівня, яке їх призначення

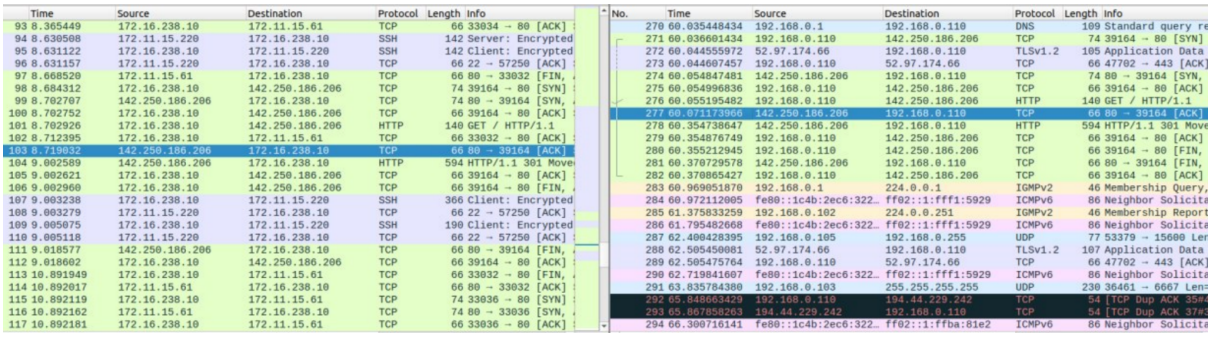


Бачимо відправлений Acknowledge-пакет (ACK) від 172.16.238.10:53698 (source) про прийняття пакета з 172.11.15.61:80 (destination)

6) Знову відкрийте дві сесії на client_pc1 і запустіть tcpdump. Одночасно, запустіть wireshark для перехопленя трафіку зі сторони хосту, на мостовому інтерфейсі з контейнером corp_gateway. У другій SSH сесії спробуйте доступитися до зовнішнього ресурсу (наприклад, через команду curl google.com). Поясніть роботу механізму NAT на прикладі порівняння двох дампів трафіків – зовнішнього і внутрішнього



Бачимо зі скріншота з Wireshark для зовнішнього трафіку, що роутер шле ARP-запити для пошуку фізичної адреси того, кому буде повертати запити “зовні”, тобто від google.com



	Time	Source	Destination	Protocol	Length	Info
93	8.365449	172.16.238.10	172.11.15.61	TCP	66	33034 → 80 [ACK]
94	8.630508	172.11.15.220	172.16.238.10	SSH	142	Server: Encrypted
95	8.631122	172.16.238.10	172.11.15.220	SSH	142	Client: Encrypted
96	8.631157	172.11.15.220	172.16.238.10	TCP	66	22 → 57250 [ACK]
97	8.668520	172.11.15.61	172.16.238.10	TCP	66	80 → 33032 [FIN,
98	8.684312	172.16.238.10	142.250.186.206	TCP	74	39164 → 80 [SYN]
99	8.702707	142.250.186.206	172.16.238.10	TCP	74	80 → 39164 [SYN,
100	8.702752	172.16.238.10	142.250.186.206	TCP	66	39164 → 80 [ACK]
101	8.702926	172.16.238.10	142.250.186.206	HTTP	140	GET / HTTP/1.1
102	8.712395	172.16.238.10	172.11.15.61	TCP	66	33032 → 80 [ACK]
103	8.719032	142.250.186.206	172.16.238.10	TCP	66	80 → 39164 [ACK]
104	9.002589	142.250.186.206	172.16.238.10	HTTP	594	HTTP/1.1 301 Move
105	9.002621	172.16.238.10	142.250.186.206	TCP	66	39164 → 80 [ACK]
106	9.002960	172.16.238.10	142.250.186.206	TCP	66	39164 → 80 [FIN,
107	9.003238	172.16.238.10	172.11.15.220	SSH	366	Client: Encrypted
108	9.003279	172.11.15.220	172.16.238.10	TCP	66	22 → 57250 [ACK]
109	9.005075	172.16.238.10	172.11.15.220	SSH	190	Client: Encrypted
110	9.005118	172.11.15.220	172.16.238.10	TCP	66	22 → 57250 [ACK]
111	9.018577	142.250.186.206	172.16.238.10	TCP	66	80 → 39164 [FIN,
112	9.018602	172.16.238.10	142.250.186.206	TCP	66	39164 → 80 [ACK]
113	10.891949	172.16.238.10	172.11.15.61	TCP	66	33032 → 80 [FIN,
114	10.892017	172.11.15.61	172.16.238.10	TCP	66	80 → 33032 [ACK]
115	10.892119	172.16.238.10	172.11.15.61	TCP	74	33036 → 80 [SYN]
116	10.892162	172.11.15.61	172.16.238.10	TCP	74	80 → 33036 [SYN,
117	10.892181	172.16.238.10	172.11.15.61	TCP	66	33036 → 80 [ACK]

No.	Time	Source	Destination	Protocol	Length	Info
270	60.035448434	192.168.0.1	192.168.0.110	DNS	109	Standard query re
271	60.036601434	192.168.0.110	142.250.186.206	TCP	74	39164 → 80 [SYN]
272	60.044555972	52.97.174.66	192.168.0.110	TLSv1.2	105	Application Data
273	60.044607457	192.168.0.110	52.97.174.66	TCP	66	47702 → 443 [ACK]
274	60.054847481	142.250.186.206	192.168.0.110	TCP	74	80 → 39164 [SYN,
275	60.054996836	192.168.0.110	142.250.186.206	TCP	66	39164 → 80 [ACK]
276	60.055195482	192.168.0.110	142.250.186.206	HTTP	140	GET / HTTP/1.1
277	60.071173966	142.250.186.206	192.168.0.110	TCP	66	80 → 39164 [ACK]
278	60.354738647	142.250.186.206	192.168.0.110	HTTP	594	HTTP/1.1 301 Move
279	60.354876749	192.168.0.110	142.250.186.206	TCP	66	39164 → 80 [ACK]
280	60.355212945	192.168.0.110	142.250.186.206	TCP	66	39164 → 80 [FIN,
281	60.370729578	142.250.186.206	192.168.0.110	TCP	66	80 → 39164 [FIN,
282	60.370865427	192.168.0.110	142.250.186.206	TCP	66	39164 → 80 [ACK]
283	60.969051870	192.168.0.1	224.0.0.1	IGMPv2	46	Membership Query,
284	60.972112005	fe80::1c4b:2ec6:322…	ff02::1:fff1:5929	ICMPv6	86	Neighbor Solicita
285	61.375833259	192.168.0.102	224.0.0.251	IGMPv2	46	Membership Report
286	61.795482668	fe80::1c4b:2ec6:322…	ff02::1:fff1:5929	ICMPv6	86	Neighbor Solicita
287	62.400428395	192.168.0.105	192.168.0.255	UDP	77	53379 → 15600 Len
288	62.505450081	52.97.174.66	192.168.0.110	TLSv1.2	107	Application Data
289	62.505475764	192.168.0.110	52.97.174.66	TCP	66	47702 → 443 [ACK]
290	62.719841607	fe80::1c4b:2ec6:322…	ff02::1:fff1:5929	ICMPv6	86	Neighbor Solicita
291	63.835784380	192.168.0.103	255.255.255.255	UDP	230	36461 → 6667 Len
292	65.848663429	192.168.0.110	194.44.229.242	TCP	54	[TCP Dup ACK 35
293	65.867858263	194.44.229.242	192.168.0.110	TCP	54	[TCP Dup ACK 37
294	66.300716141	fe80::1c4b:2ec6:322…	ff02::1:ffba:81e2	ICMPv6	86	Neighbor Solicita

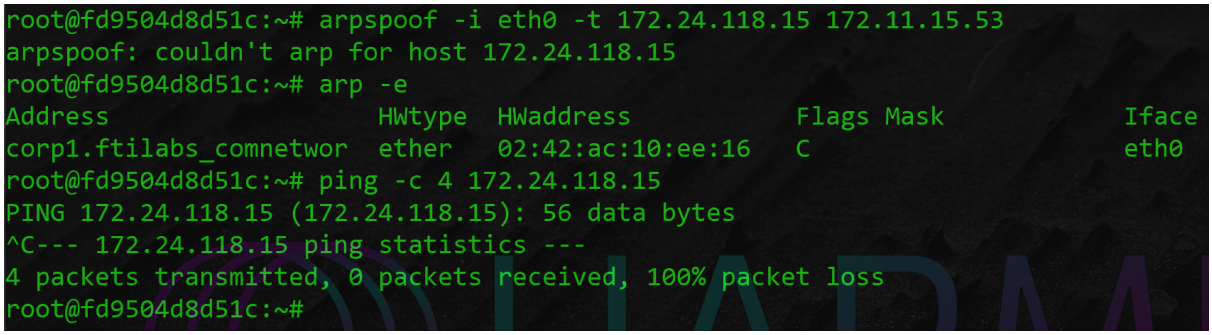
Порівнюючи внутрішній (зліва) та зовнішній (справа) трафіки, бачимо, що source-адреса однакова - 142.250.186.206 (google.com), проте destination-адреси різні (всередині це адреса pc_1, а зовні це локальна адреса хоста - 192.168.0.110).

Взагалі, NAT це такий “хак”, коли під одною білою IPv4 адресою роутера, насправді може бути декілька окремих девайсів, які відправляють запити в мережу, а NAT розкриває пакет, вписує свою адресу і запам’ятовує що він замінив, щоб потім розкрити пакет відповіді, вписати туди справжнього адресата з локальної мережі та передати пакет далі. Таким чином, маскується справжня адреса відправника + якщо підміняти TTL, то можна маскувати навіть сам факт наявності роутера. Це призводить до економії IPv4 адрес, які вже закінчились і збільшує безпеку внутрішніх девайсів локальної мережі.

7) На вузлі client_pc1 запустіть tcpdump та збирайте трафік впродовж 1-2 хвилин. Проведіть arp spoofing атаку на вузли client_pc2 client_pc3, ftp_server, під час кожної з них зберіть трафік впродовж 1- 2 хвилин. Порівняйте отримані дампи та надайте їм оцінку, поясніть, який результат дала атака в кожному випадку і чому саме такий

	Time	Source	Destination	Protocol	Length	Info
191	48.968022	172.11.15.220	172.16.238.10	TCP	66	22 → 57250 [ACK] Seq=77 Ack=241 Win=24555 Len=0 TSval=
192	48.968394	172.16.238.10	172.11.15.220	SSH	150	Client: Encrypted packet (len=84)
193	48.968431	172.11.15.220	172.16.238.10	TCP	66	22 → 57250 [ACK] Seq=77 Ack=325 Win=24555 Len=0 TSval=
194	48.968681	172.16.238.10	172.11.15.220	SSH	190	Client: Encrypted packet (len=124)
195	48.968709	172.11.15.220	172.16.238.10	TCP	66	22 → 57250 [ACK] Seq=77 Ack=449 Win=24555 Len=0 TSval=
196	49.000452	72:d6:79:f6:80:9d	Broadcast	ARP	42	Who has 172.16.238.30? Tell 172.16.238.10
197	49.000481	22:b3:b7:c4:96:2f	72:d6:79:f6:80:9d	ARP	42	172.16.238.30 is at 22:b3:b7:c4:96:2f
198	49.000485	172.16.238.10	172.16.238.30	UDP	42	42648 → 67 Len=0
199	49.000526	172.16.238.30	172.16.238.10	ICMP	70	Destination unreachable (Port unreachable)
200	49.829411	172.16.238.10	172.11.15.61	TCP	66	36140 → 80 [FIN, ACK] Seq=200 Ack=196 Win=64128 Len=0
201	49.829464	172.11.15.61	172.16.238.10	TCP	66	80 → 36140 [ACK] Seq=196 Ack=201 Win=65024 Len=0 TSval=
202	49.829545	172.16.238.10	172.11.15.61	TCP	74	36142 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_P
203	49.829575	172.11.15.61	172.16.238.10	TCP	74	80 → 36142 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=
204	49.829585	172.16.238.10	172.11.15.61	TCP	66	36142 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=298
205	49.829605	172.16.238.10	172.11.15.61	HTTP	256	HEAD /aesiitf HTTP/1.1
206	49.829637	172.11.15.61	172.16.238.10	TCP	66	80 → 36142 [ACK] Seq=1 Ack=191 Win=65024 Len=0 TSval=5
207	49.829832	172.11.15.61	172.16.238.10	HTTP	260	HTTP/1.1 404 Not Found
208	49.829838	172.16.238.10	172.11.15.61	TCP	66	36142 → 80 [ACK] Seq=191 Ack=195 Win=64128 Len=0 TSval=
209	50.001101	72:d6:79:f6:80:9d	22:b3:b7:c4:96:2f	ARP	42	172.11.15.53 is at 72:d6:79:f6:80:9d
210	50.358866	172.16.238.10	172.11.15.61	TCP	327	36138 → 80 [PSH, ACK] Seq=4261 Ack=73125 Win=64128 Len=
211	50.358888	172.11.15.61	172.16.238.10	TCP	66	80 → 36138 [ACK] Seq=73125 Ack=4522 Win=64128 Len=0 TS
212	50.358907	172.16.238.10	172.11.15.61	HTTP	1224	HEAD /common?page=1 HTTP/1.1 (application/x-www-form-
213	50.358915	172.11.15.61	172.16.238.10	TCP	66	80 → 36138 [ACK] Seq=73125 Ack=5680 Win=64128 Len=0 TS
214	50.359076	172.11.15.61	172.16.238.10	HTTP	327	HTTP/1.1 200 OK
215	50.359080	172.16.238.10	172.11.15.61	TCP	66	36138 → 80 [ACK] Seq=5680 Ack=73386 Win=64128 Len=0 TS
216	52.001244	72:d6:79:f6:80:9d	22:b3:b7:c4:96:2f	ARP	42	172.11.15.53 is at 72:d6:79:f6:80:9d
217	54.001501	72:d6:79:f6:80:9d	22:b3:b7:c4:96:2f	ARP	42	172.11.15.53 is at 72:d6:79:f6:80:9d
218	54.008426	22:b3:b7:c4:96:2f	72:d6:79:f6:80:9d	ARP	42	Who has 172.16.238.10? Tell 172.16.238.30
219	54.008434	72:d6:79:f6:80:9d	22:b3:b7:c4:96:2f	ARP	42	172.16.238.10 is at 72:d6:79:f6:80:9d

Неможливо провести arp spoofing атаку на pc_3 та ftp-server, бо pc_3 знаходиться в іншій мережі. Тому утиліта arpspoof не може зробити arp-запит на 172.24.118.15 (pc_3)


root@fd9504d8d51c:~# arpspoof -i eth0 -t 172.24.118.15 172.11.15.53
arpspoof: couldn't arp for host 172.24.118.15
root@fd9504d8d51c:~# arp -e
Address HWtype HWaddress Flags Mask Iface
corp1.ftilabs_comnetwor ether 02:42:ac:10:ee:16 C eth0
root@fd9504d8d51c:~# ping -c 4 172.24.118.15
PING 172.24.118.15 (172.24.118.15): 56 data bytes
^C--- 172.24.118.15 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
root@fd9504d8d51c:~#