

## Лабораторна робота №2

### 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 мережніх протоколів (Ethernet, HTTP, FTP), протоколу сеансового рівня SSL/TLS, а також поглиблення навичок роботи утилітами для аналізу мережного трафіку: tcpdump, wireshark.

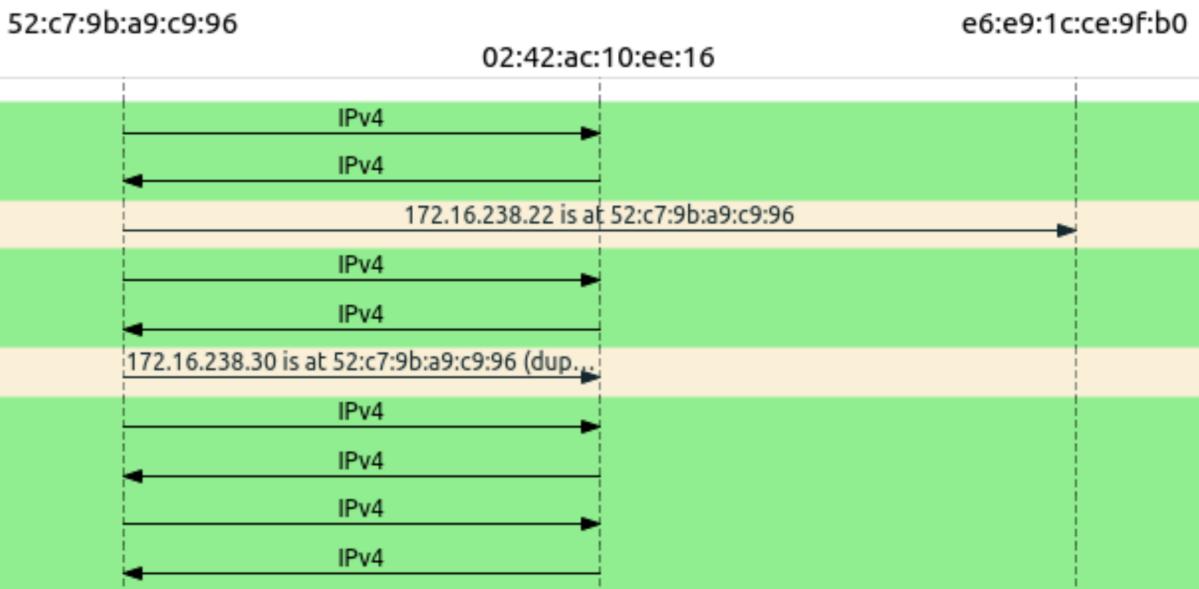
- 1) За допомогою tcpdump перехопіть трафік з вузла client\_pc1 впродовж 1-2 хвилин. Вимкніть аналіз IPv4. Виберіть один з пакетів та опишіть на прикладі перехопленого пакета структуру Ethernet II фрейму

Frame	Time	Source MAC	Destination MAC	Type	Length	Protocol
87	2.501417	02:42:ac:10:ee:16	52:c7:9b:a9:c9:96	0x0800	66	IPv4
90	2.501589	52:c7:9b:a9:c9:96	02:42:ac:10:ee:16	0x0800	66	IPv4
91	2.501604	02:42:ac:10:ee:16	52:c7:9b:a9:c9:96	0x0800	66	IPv4
94	2.740381	52:c7:9b:a9:c9:96	02:42:ac:10:ee:16	0x0800	238	IPv4
95	2.740421	02:42:ac:10:ee:16	52:c7:9b:a9:c9:96	0x0800	66	IPv4
96	2.881376	52:c7:9b:a9:c9:96	02:42:ac:10:ee:16	ARP	42	172.
97	2.881578	52:c7:9b:a9:c9:96	02:42:ac:10:ee:16	0x0800	238	IPv4
98	2.881608	02:42:ac:10:ee:16	52:c7:9b:a9:c9:96	0x0800	66	IPv4
99	4.105633	52:c7:9b:a9:c9:96	02:42:ac:10:ee:16	0x0800	250	IPv4
100	4.105678	02:42:ac:10:ee:16	52:c7:9b:a9:c9:96	0x0800	66	IPv4
101	4.106117	02:42:ac:10:ee:16	52:c7:9b:a9:c9:96	0x0800	7306	IPv4
102	4.106127	52:c7:9b:a9:c9:96	02:42:ac:10:ee:16	0x0800	66	IPv4
103	4.106142	02:42:ac:10:ee:16	52:c7:9b:a9:c9:96	0x0800	7306	IPv4

Frame 87: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)  
Ethernet II, Src: 02:42:ac:10:ee:16 (02:42:ac:10:ee:16), Dst: 52:c7:9b:a9:c9:96 (Destination: 52:c7:9b:a9:c9:96 (52:c7:9b:a9:c9:96))  
Source: 02:42:ac:10:ee:16 (02:42:ac:10:ee:16)

Бачимо у фреймі 2 блоки: Frame 87 - це номер фрейму, його довжина, флаги; Ethernet II - це поля MAC-адреси source і destination

- 2) Проведіть атаку arp spoof на вузол client\_pc2 та перехоплюйте трафік впродовж 1-2 хвилин. Проаналізуйте довільний перехоплений пакет, визначте та опишіть його шлях на канальному рівні, яким чином модифікується ethernet фрейм під час форвардінгу



- 3) Ознайомтесь з CONNECT скануванням за допомогою nmap (наприклад, тут: <https://nmap.org/book/man-port-scanning-techniques.html>), як воно відбувається і в яких випадках застосовується? Проскануйте методом CONNECT вузол https\_portal (172.11.15.71) та перехопіть процес мережним сніфером (tcpdump). На прикладі перехопленого скану, опишіть, як відрізняється реакція закритого і відкритого портів та яка відмінність від SYN сканування

**TCP connect method** – це метод, при якому спостерігач використовує Berkley Socket API та встановлює з'єднання з цільовою машиною в тристоронній послідовності пакетів (three-handshake). Цей метод використовується, коли відправляти сирі пакети в корпоративну мережу неможливо (наприклад, на локальному хості під час проведення penetration test) або застосовується IPv6. Мінусом є легка детекція в логах (зокрема access чи error).

190 9.755738	172.16.238.10	172.11.15.71	TCP	74 38186 → 993 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
191 9.755772	172.11.15.71	172.16.238.10	TCP	54 993 → 38186 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
192 9.755810	172.16.238.10	172.11.15.71	TCP	74 38864 → 199 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
193 9.755839	172.11.15.71	172.16.238.10	TCP	54 199 → 38864 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
194 9.755873	172.16.238.10	172.11.15.71	TCP	74 43660 → 111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
196 9.755937	172.16.238.10	172.11.15.71	TCP	74 37394 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T
197 9.755968	172.11.15.71	172.16.238.10	TCP	74 80 → 37394 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA
198 9.755982	172.16.238.10	172.11.15.71	TCP	66 37394 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=445913089
207 9.756413	172.16.238.10	172.11.15.71	TCP	66 37394 → 80 [RST, ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=44591

Бачимо, що у першому випадку, коли порт закритий, хост відповідає RST|ACK пакетом, а в другому випадку, коли порт відкритий, хост відповідає SYN|ACK, на що спостерігач відповідає ACK пакетом, своєю чергою хост відповідає RST|ACK. Таким чином утиліта NMAP може сканувати всі порти один за одним, перериваючи з'єднання у разі знаходження відкритого порту

- 4) Відкрийте у wireshark дамп з машини client\_pc1 і профільтруйте трафік за http. Опишіть на прикладі перехоплених пакетів: які HTTP методи тут використовуються, які використовуються поля заголовків і для чого кожне з них призначений

132 6.679141	172.16.238.10	172.11.15.61	HTTP	1224 POST /kmu HTTP/1.1 (application/x-www-form-urlencoded)
133 6.679187	172.11.15.61	172.16.238.10	TCP	66 80 → 53700 [ACK] Seq=1 Ack=1403 Win=64128 Len=0 TSval=6
134 6.679671	172.11.15.61	172.16.238.10	TCP	7306 80 → 53700 [PSH, ACK] Seq=1 Ack=1403 Win=64128 Len=7240
135 6.679720	172.16.238.10	172.11.15.61	TCP	66 53700 → 80 [ACK] Seq=1403 Ack=7241 Win=61312 Len=0 TSva
136 6.679778	172.11.15.61	172.16.238.10	TCP	7306 80 → 53700 [PSH, ACK] Seq=7241 Ack=1403 Win=64128 Len=7
137 6.679793	172.16.238.10	172.11.15.61	TCP	66 53700 → 80 [ACK] Seq=1403 Ack=14481 Win=57728 Len=0 TSv
138 6.679860	172.11.15.61	172.16.238.10	TCP	14546 80 → 53700 [PSH, ACK] Seq=14481 Ack=1403 Win=64128 Len=
139 6.679873	172.16.238.10	172.11.15.61	TCP	66 53700 → 80 [ACK] Seq=1403 Ack=28961 Win=50432 Len=0 Tsv
140 6.679902	172.11.15.61	172.16.238.10	TCP	14546 80 → 53700 [PSH, ACK] Seq=28961 Ack=1403 Win=64128 Len=
141 6.679920	172.16.238.10	172.11.15.61	TCP	66 53700 → 80 [ACK] Seq=1403 Ack=43441 Win=43264 Len=0 TSv
142 6.679966	172.11.15.61	172.16.238.10	TCP	2962 80 → 53700 [PSH, ACK] Seq=43441 Ack=1403 Win=64128 Len=
143 6.679980	172.16.238.10	172.11.15.61	TCP	66 53700 → 80 [ACK] Seq=1403 Ack=46337 Win=41728 Len=0 TSv
144 6.680019	172.11.15.61	172.16.238.10	TCP	24682 80 → 53700 [PSH, ACK] Seq=46337 Ack=1403 Win=64128 Len=
145 6.680095	172.16.238.10	172.11.15.61	TCP	66 53700 → 80 [ACK] Seq=1403 Ack=70953 Win=29312 Len=0 TSv
146 6.680145	172.11.15.61	172.16.238.10	HTTP	1406 HTTP/1.1 200 OK

Frame 132: 1224 bytes on wire (9792 bits), 1224 bytes captured (9792 bits)  
 Ethernet II, Src: 52:c7:9b:a9:c9:96 (52:c7:9b:a9:c9:96), Dst: 02:42:ac:10:ee:16 (02:42:ac:10:ee:16)

↳ Destination: 02:42:ac:10:ee:16 (02:42:ac:10:ee:16)  
 ↳ Source: 52:c7:9b:a9:c9:96 (52:c7:9b:a9:c9:96)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 172.16.238.10, Dst: 172.11.15.61

Transmission Control Protocol, Src Port: 53700, Dst Port: 80, Seq: 245, Ack: 1, Len: 1158

Source Port: 53700

Destination Port: 80

[Stream index: 5]

[TCP Segment Len: 1158]

Sequence number: 245 (relative sequence number)

Sequence number (raw): 3646367356

[Next sequence number: 1403 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

Acknowledgment number (raw): 3422429413

1000 .... = Header Length: 32 bytes (8)

Бачимо HTTP метод POST за URL /kmu і заголовком x-www-form-urlencoded.

Скоріше за все це відправляється форма

- 5) Відкрийте у перехоплений трафік з машини client\_pc2 (отриманий в результаті arp spoof) та проаналізуйте ftp трафік. На прикладі перехоплених пакетів, опишіть процедуру аутентифікації та витягніть логін і пароль користувача, під яким передаються файли

19 2.480686	172.11.15.53	172.16.238.30	FTP	148 Response: 220 Welcome Alpine ftp server https://hub.docker
23 2.480754	172.16.238.30	172.11.15.53	FTP	86 Request: USER administrator
27 2.480801	172.11.15.53	172.16.238.30	FTP	100 Response: 331 Please specify the password.
31 2.480822	172.16.238.30	172.11.15.53	FTP	84 Request: PASS Chogory2002
35 2.500389	172.11.15.53	172.16.238.30	FTP	89 Response: 230 Login successful.
39 2.500478	172.16.238.30	172.11.15.53	FTP	74 Request: TYPE I
43 2.500550	172.11.15.53	172.16.238.30	FTP	97 Response: 200 Switching to Binary mode.
47 2.500611	172.16.238.30	172.11.15.53	FTP	72 Request: EPSV
51 2.500709	172.11.15.53	172.16.238.30	FTP	114 Response: 229 Entering Extended Passive Mode (   21009 )
61 2.500832	172.16.238.30	172.11.15.53	FTP	82 Request: STOR ./qui.txt
65 2.501088	172.11.15.53	172.16.238.30	FTP	88 Response: 150 Ok to send data.
75 2.501279	172.11.15.53	172.16.238.30	FTP	90 Response: 226 Transfer complete.
79 2.501343	172.16.238.30	172.11.15.53	FTP	72 Request: QUIT

Бачимо, що аутентифікація проходить у 3 етапи:

- 1) Клієнт відправляє логін "administrator" через команду USER
  - 2) Сервер запитує пароль командою номер 331
  - 3) Клієнт відправляє "Chogory2002" через команду PASS
- 6) Проаналізуйте SSL трафік в цьому ж дампі. Визначте та опишіть, які версії протоколів та які набори шифрів підтримуються клієнтом. Які з цих наборів мають властивість backward secrecy, а які – ні. Який набір шифрів використовується для встановлення TLS з'єднання

1581 46.806608	172.16.238.30	172.11.15.71	TLSv1.2	583 Client Hello
1585 46.806773	172.11.15.71	172.16.238.30	TLSv1.2	832 Server Hello, Certificate, Server Hello Done
1589 46.806997	172.16.238.30	172.11.15.71	TLSv1.2	384 Client Key Exchange, Change Cipher Spec, Finished
1593 46.808473	172.11.15.71	172.16.238.30	TLSv1.2	324 New Session Ticket, Change Cipher Spec, Finished
1597 46.808853	172.16.238.30	172.11.15.71	TLSv1.2	367 [TLS segment of a reassembled PDU]

Бачимо, що клієнт та сервер домовилися про використання TLSv1.2, який дозволяє використання шифрів без backward secrecy, тому цей протокол вважається вразливим

- 7) Розшифруйте HTTPS трафік використовуючи скомпрометований ключ, що знаходиться в репозиторії (.assets/domain.key). Проаналізуйте розшифрований трафік. Знайдіть пакети, що відповідають за аутентифікацію та витягніть логін і пароль

1430 41.670034	172.16.238.30	172.11.15.71	HTTP	198 POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
1434 41.697670	172.11.15.71	172.16.238.30	HTTP	1314 HTTP/1.1 302 Found
1438 41.699637	172.16.238.30	172.11.15.71	HTTP	577 GET /wp-login.php?redirect_to=https%3A%2F%2Fftilabs.com%
1442 41.713630	172.11.15.71	172.16.238.30	HTTP	2012 HTTP/1.1 200 OK (text/html)
1581 46.806608	172.16.238.30	172.11.15.71	TLSv1.2	583 Client Hello
1585 46.806773	172.11.15.71	172.16.238.30	TLSv1.2	832 Server Hello, Certificate, Server Hello Done
1589 46.806997	172.16.238.30	172.11.15.71	TLSv1.2	384 Client Key Exchange, Change Cipher Spec, Finished
1593 46.808473	172.11.15.71	172.16.238.30	TLSv1.2	324 New Session Ticket, Change Cipher Spec, Finished
1597 46.808853	172.16.238.30	172.11.15.71	TLSv1.2	367 [TLS segment of a reassembled PDU]
1601 46.808902	172.16.238.30	172.11.15.71	HTTP	198 POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
1605 46.837091	172.11.15.71	172.16.238.30	HTTP	1314 HTTP/1.1 302 Found
1609 46.838929	172.16.238.30	172.11.15.71	HTTP	577 GET /wp-login.php?redirect_to=https%3A%2F%2Fftilabs.com%
1613 46.852455	172.11.15.71	172.16.238.30	HTTP	2011 HTTP/1.1 200 OK (text/html)
1782 51.947599	172.16.238.30	172.11.15.71	TLSv1.2	583 Client Hello
1786 51.947715	172.11.15.71	172.16.238.30	TLSv1.2	832 Server Hello, Certificate, Server Hello Done
1790 51.947886	172.16.238.30	172.11.15.71	TLSv1.2	384 Client Key Exchange, Change Cipher Spec, Finished
1794 51.948875	172.11.15.71	172.16.238.30	TLSv1.2	324 New Session Ticket, Change Cipher Spec, Finished
1798 51.949215	172.16.238.30	172.11.15.71	TLSv1.2	367 [TLS segment of a reassembled PDU]
1802 51.949244	172.16.238.30	172.11.15.71	HTTP	198 POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
1806 51.976405	172.11.15.71	172.16.238.30	HTTP	1314 HTTP/1.1 302 Found
1810 51.978291	172.16.238.30	172.11.15.71	HTTP	577 GET /wp-login.php?redirect_to=https%3A%2F%2Fftilabs.com%
1814 51.991992	172.11.15.71	172.16.238.30	HTTP	2013 HTTP/1.1 200 OK (text/html)
1936 57.0R94A02	172.16.238.30	172.11.15.71	TLSv1.2	583 Client Hello

```

<Accept-Encoding: gzip, deflate\r\n>
Accept: */*\r\n
<Accept: */*\r\n>
Connection: keep-alive\r\n
<Connection: keep-alive\r\n>
Cookie: wordpress_test_cookie=WP Cookie check\r\n
<Cookie: wordpress_test_cookie=WP Cookie check\r\n>
Content-Length: 103\r\n
<Content-Length: 103\r\n>
Content-Type: application/x-www-form-urlencoded\r\n
<Content-Type: application/x-www-form-urlencoded\r\n>
\...\r\n
[Full request URI: https://ftilabs.com/wp-login.php]
<Request: True>
[HTTP request 1/2]
[Response in frame: 1434]
[Next request in frame: 1438]
File Data: 103 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
  ▶ Form item: "log" = "fti"
  ▶ Form item: "pwd" = "qwerty123"
  ▶ Form item: "wp-submit" = "Log In"
  ▶ Form item: "redirect_to" = "https://ftilabs.com/wp-admin/"
  ▶ Form item: "testcookie" = "1"

```

Бачимо POST-запит на аутентифікацію до адмін-панелі WordPress з логіном “fti” та паролем “qwerty123”