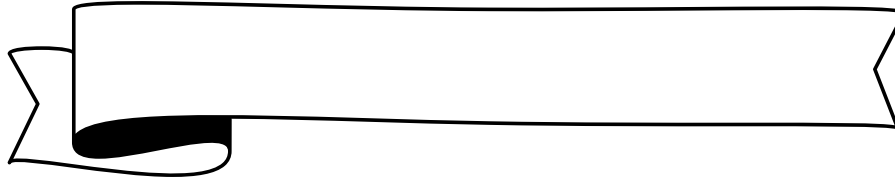


Progetto M1

Elisa Bruno



Esercizio

Traccia e requisiti

Nell'esercizio di oggi metteremo insieme le competenze acquisite finora. Lo studente verrà valutato sulla base della risoluzione al problema seguente.

Requisiti e servizi:

- Kali Linux ☐ IP 192.168.32.100
- Windows 7 ☐ IP 192.168.32.101
- HTTPS server: attivo
- Servizio DNS per risoluzione nomi di dominio: attivo

Traccia:

Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows 7) richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100 (Kali).

Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

Ripetere l'esercizio, sostituendo il server HTTPS, con un server HTTP. Si intercetti nuovamente il traffico, evidenziando le eventuali differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS. Spiegare, motivandole, le principali differenze se presenti.



Imposto l'IP statico su Kali Linux e su Windows 7, come da indicazione. Faccio pingare le due VM dopo aver disattivato il Firewall su Win7 e impostato le VM da Oracle su rete "Internal".

```
(kali@kali)-[~]
$ ping 192.168.32.101
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data:
64 bytes from 192.168.32.101: icmp_seq=1 ttl=128 time=4.73 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.972 ms
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.810 ms
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.848 ms
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.949 ms
^C
--- 192.168.32.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4023ms
rtt min/avg/max/mdev = 0.810/1.662/4.732/1.536 ms
```

Con il comando `/etc/inetsim/inetsim.conf` cambio le impostazioni tengo attivo il protocollo l'HTTP, l'HTTPS e il DNS e disattivo tutti gli altri (mettendo un `#` davanti al comando).

```
time_udp, daytime_tcp, daytime_udp, echo_tcp,
echo_udp, discard_tcp, discard_udp, quotd_tcp,
quotd_udp, chargen_tcp, chargen_udp, finger,
ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
ftp, irc, https

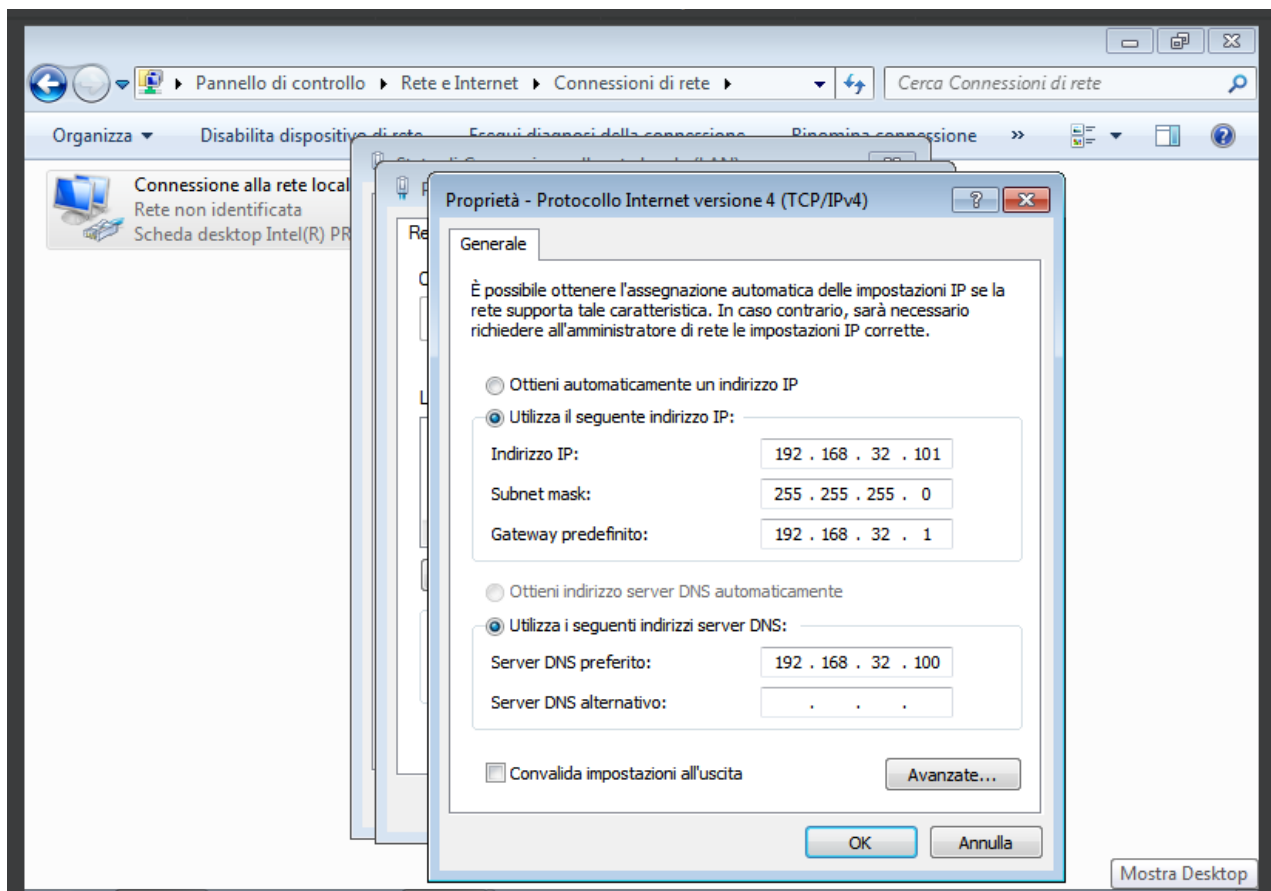
#rt_service dns
rt_service http
rt_service https
#rt_service smtp
#rt_service smtps
#rt_service pop3
#rt_service pop3s
#rt_service ftp
#rt_service ftps
#rt_service tftp
#rt_service irc
#rt_service ntp
#rt_service finger
#rt_service ident
#rt_service syslog
#rt_service time_tcp
#rt_service time_udp
#rt_service daytime_tcp
#rt_service daytime_udp
#rt_service echo_tcp
#rt_service echo_udp
#rt_service discard_tcp
#rt_service discard_udp
#rt_service quotd_tcp
#rt_service quotd_udp
#rt_service chargen_tcp
#rt_service chargen_udp
#rt_service dummy_tcp
#rt_service dummy_udp
```

Attivo, nelle successive impostazioni, il DNS statico, scrivendo `epicode.internal` e l'IP associato (che coincide con la macchina Kali).

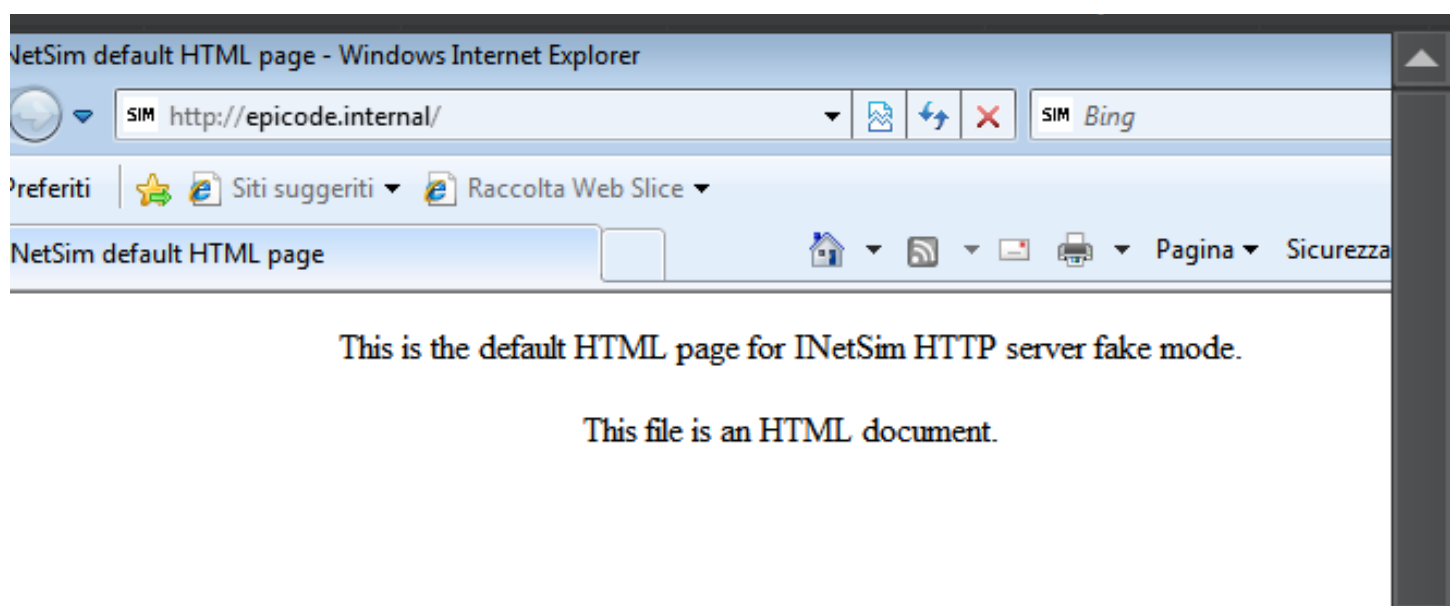
```
#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
dns_static epicode.internal 192.168.32.100
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30

#####
# dns_version
#
# DNS version
#
# Syntax: dns_version <version>
```

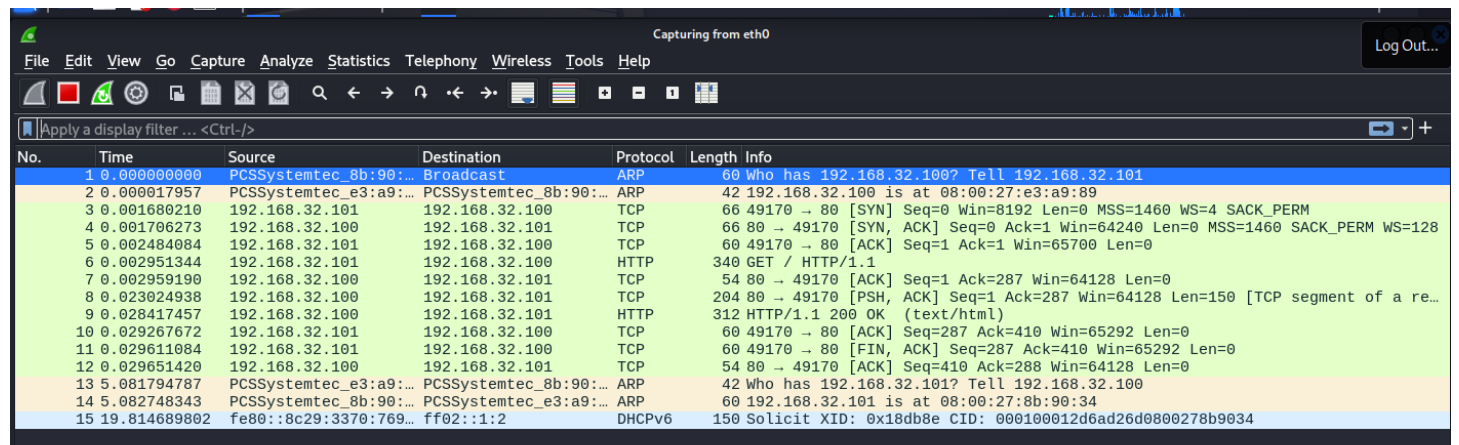
Dalle impostazioni della scheda di rete di Win7, specifico l'IP del DNS.



Con il comando `-sudo inetsim` avvio la simulazione di rete, in Kali. Provo la navigazione, da Explorer di Win7 su `epicode.internal`, prima con il protocollo HTTP, poi con l'HTTPS.



Avvio Wireshark per intercettare la comunicazione tra client-server appena impostati. La prima immagine mostra il traffico sul protocollo HTTP:



The image shows a Wireshark network traffic capture. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with icons for common actions. The main display area shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The packets are as follows:

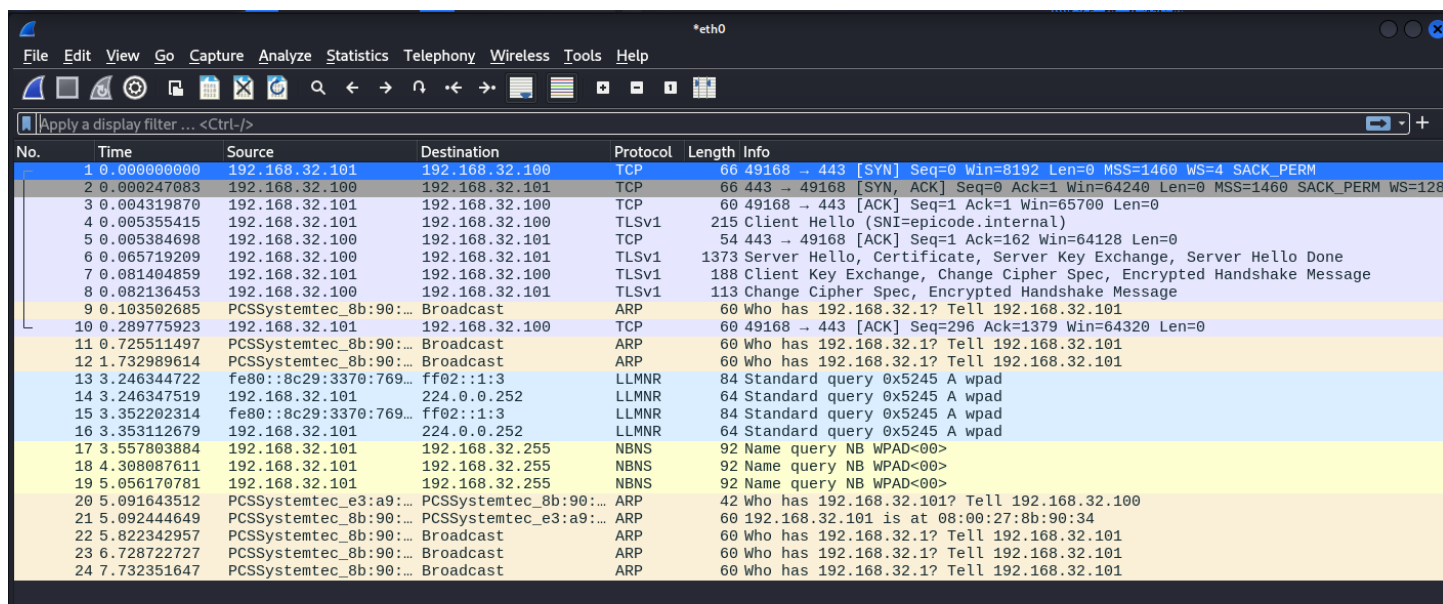
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PCSSystemtec_8b:90:...	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000017957	PCSSystemtec_e3:a9:...	PCSSystemtec_8b:90:...	ARP	42	192.168.32.100 is at 08:00:27:e3:a9:89
3	0.001680210	192.168.32.101	192.168.32.100	TCP	66	49170 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
4	0.001706273	192.168.32.100	192.168.32.101	TCP	66	80 → 49170 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
5	0.002484084	192.168.32.101	192.168.32.100	TCP	60	49170 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.002951344	192.168.32.101	192.168.32.100	HTTP	340	GET / HTTP/1.1
7	0.002959190	192.168.32.100	192.168.32.101	TCP	54	80 → 49170 [ACK] Seq=1 Ack=287 Win=64128 Len=0
8	0.0023024938	192.168.32.100	192.168.32.101	TCP	204	80 → 49170 [PSH, ACK] Seq=1 Ack=287 Win=64128 Len=150 [TCP segment of a re...
9	0.028417457	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
10	0.029267672	192.168.32.101	192.168.32.100	TCP	60	49170 → 80 [ACK] Seq=287 Ack=410 Win=65292 Len=0
11	0.029611084	192.168.32.101	192.168.32.100	TCP	60	49170 → 80 [FIN, ACK] Seq=287 Ack=410 Win=65292 Len=0
12	0.029651420	192.168.32.100	192.168.32.101	TCP	54	80 → 49170 [ACK] Seq=410 Ack=288 Win=64128 Len=0
13	5.081794787	PCSSystemtec_e3:a9:...	PCSSystemtec_8b:90:...	ARP	42	Who has 192.168.32.101? Tell 192.168.32.100
14	5.082748343	PCSSystemtec_8b:90:...	PCSSystemtec_e3:a9:...	ARP	60	192.168.32.101 is at 08:00:27:8b:90:34
15	19.814689802	fe80::8c29:3370:769...	ff02::1:2	DHCPv6	150	Solicit XID: 0x18db8e CID: 000100012d6ad26d0800278b9034

Nel secondo protocollo ARP in uscita si legge l'IP di Kali (192.168.32.100 con il suo MAC address a seguire). Negli ultimi due protocolli ARP, invece, si legge l'IP di Win7 (192.168.32.101 e il suo MAC address).

- **ARP (Address Resolution Protocol):** è un protocollo o una procedura che collega un indirizzo IP (Internet Protocol) in continua evoluzione a un indirizzo fisso del computer fisico, noto anche come indirizzo MAC (Media Access Control), in una rete locale (LAN). Le dimensioni della cache ARP sono limitate per progettazione e gli indirizzi tendono a rimanere nella cache soltanto per pochi minuti. Viene pulita regolarmente per liberare spazio anche con lo scopo di garantire privacy e sicurezza. Lo spoofing ARP è noto anche come routing di avvelenamento dell'ARP o avvelenamento della cache ARP. Si tratta di un tipo di attacco in cui un criminale informatico invia falsi messaggi ARP a una rete LAN con l'intenzione di collegare il proprio indirizzo MAC all'indirizzo IP di un dispositivo o server legittimo all'interno della rete. Il collegamento consente di inviare i dati dal computer della vittima al computer dell'autore di un attacco invece di inviarli alla destinazione originale.

Per queste motivazioni, una comunicazione non crittografata come l'HTTP potrebbe non essere sicura.

La seconda immagine mostra il traffico HTTPS:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.32.101	192.168.32.100	TCP	66	49168 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
2	0.000247083	192.168.32.100	192.168.32.101	TCP	66	443 → 49168 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.004319870	192.168.32.101	192.168.32.100	TCP	60	49168 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.005355415	192.168.32.101	192.168.32.100	TLSv1	215	Client Hello (SNI=epicode.internal)
5	0.005384698	192.168.32.100	192.168.32.101	TCP	54	443 → 49168 [ACK] Seq=1 Ack=162 Win=64128 Len=0
6	0.005719209	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.081404859	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
8	0.082136453	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
9	0.103502685	PCSSystemtec_8b:90:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
10	0.289775923	192.168.32.101	192.168.32.100	TCP	60	49168 → 443 [ACK] Seq=296 Ack=1379 Win=64320 Len=0
11	0.725511497	PCSSystemtec_8b:90:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
12	1.732989614	PCSSystemtec_8b:90:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
13	3.246344722	fe80::8c29:3370:769...	ff02::1:3	LLMNR	84	Standard query 0x5245 A wpad
14	3.246347519	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0x5245 A wpad
15	3.352202314	fe80::8c29:3370:769...	ff02::1:3	LLMNR	84	Standard query 0x5245 A wpad
16	3.35312679	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0x5245 A wpad
17	3.557803884	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
18	4.308087611	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
19	5.056170781	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
20	5.091643512	PCSSystemtec_e3:a9:...	PCSSystemtec_8b:90:...	ARP	42	Who has 192.168.32.101? Tell 192.168.32.100
21	5.092444649	PCSSystemtec_8b:90:...	PCSSystemtec_e3:a9:...	ARP	60	192.168.32.101 is at 08:00:27:8b:90:34
22	5.822342957	PCSSystemtec_8b:90:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
23	6.728722727	PCSSystemtec_8b:90:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
24	7.732351647	PCSSystemtec_8b:90:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101

Accanto ai protocolli ARP si può leggere IP e MAC address del client-server che comunicano. Si potranno intercettare anche altri protocolli, che non sono presenti nel traffico HTTP.

- **TLS (Transport Layer Security):** è un protocollo di sicurezza definito per stabilire canali di crittografia sulle reti. Quando un client si mette in contatto con un server protetto da **TLS**, il server invia al client un certificato che testimonia l'autenticità, attivando la prima fase della sequenza (nota come handshake **TLS**) detta di negoziazione.
- **TCP (Transmission Control Protocol):** presente in entrambe le connessioni HTTP/HTTPS, è un protocollo connection-oriented che opera al livello trasporto della pila OSI. TCP fornisce un servizio full-duplex con conferma e controllo di flusso. TCP viene utilizzato da applicativi che richiedono una trasmissione affidabile, cioè con garanzia di consegna dei dati. Esso stabilisce un canale virtuale bidirezionale fra i due host che creando su ciascun host due connessioni, una in ricezione e una in trasmissione
- **Handshake a 3 vie:** è l'instaurazione della connessione tramite il Transmission Control Protocol che prevede in totale tre passaggi. Nel primo passaggio, il **client** che richiede la connessione invia al server un **pacchetto SYN** con un numero sequenziale individuale e casuale. Questo numero assicura la trasmissione completa nella sequenza corretta.
 1. Dopo che il **server** ha ricevuto il segmento, acconsente all'instaurazione della connessione restituendo un **pacchetto SYN-ACK**, comprensivo del numero sequenziale del client aumentato di 1. Inoltre, trasmette al client il proprio numero sequenziale.
 2. Infine, il **client** conferma la ricezione del segmento SYN-ACK inviando un proprio **pacchetto ACK** che, in questo caso, contiene il numero sequenziale del server aumentato di 1. Al contempo può già trasferire i primi dati al server.

Analizzato il traffico di entrambe le connessioni, è chiaro perchè una connessione HTTPS sia più affidabile di una connessione HTTP. Il protocollo HTTP facilita le attività criminali come lo spionaggio di dati o gli attacchi man in the middle, mentre l'HTTPS utilizza una connessione criptata tramite il TLS, rendendo i dati trasmessi non decifrabili.

