

# Cybersecurity Analyst

Week 4 - W4D4 Pratica

# M1: Progetto

## Requisiti

- **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.

# CONFIGURAZIONE IP E RETE

## KALI LINUX

Da terminale su macchina Kali Linux, è stata impostata la configurazione di rete richiesta tramite modifica del file **/etc/network/interfaces**.

```
# This file describes the network inter
# and how to activate them. For more in

source /etc/network/interfaces.d/*

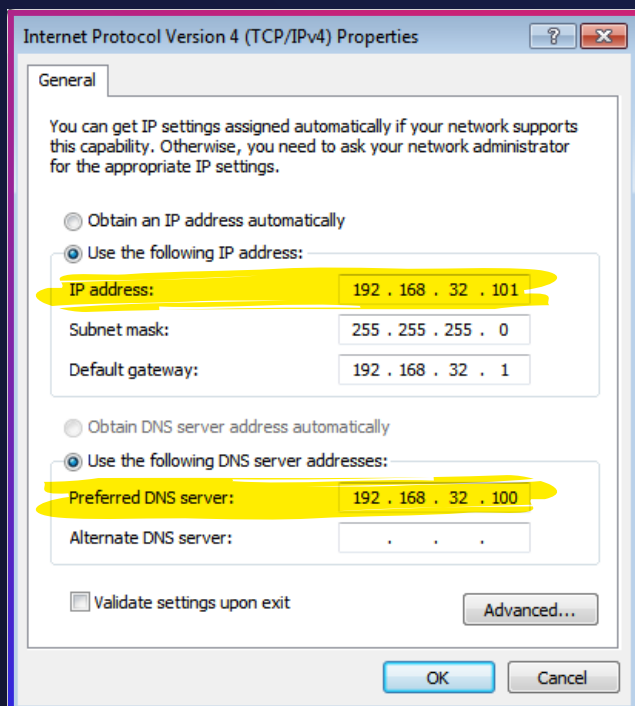
# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.32.100/24
gateway 192.168.32.1
```

## WINDOWS 7

Dalle impostazioni di rete, tramite configurazione del protocollo IPv4, è stato impostato l'IP statico richiesto in esercizio (**192.168.32.101**), la relativa sottorete e il gateway predefinito.

Inoltre, nella sezione relativa al server DNS da utilizzare, è stato impostato l'IP settato su macchina Kali Linux (**192.168.32.100**), perchè sarà questo DNS che andrà a risolvere l'hostname **epicode.internal**.



# CONFIGURAZIONE E AVVIO INETSIN

Da terminale su macchina Kali Linux, è stato configurato il file relativo alle impostazioni di **INetSim** (**inetsim.conf**) escludendo tutti i servizi ad **eccezione di quelli DNS, HTTP e HTTPS** e impostando su 0.0.0.0 (qualsiasi indirizzo) il **service bind address**.

**FIG.1 - IP SERVER DNS**

```
#####
# dns_default_ip
#
# Default IP address to return with DNS repli
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip 192.168.32.100
```

È stata inoltre configurata la sezione relativa al **server DNS** che dovrà essere simulato, impostando l'indirizzo IP di quest'ultimo con quello settato su Kali Linux (**FIG.1**) e un indirizzo DSN statico per associare questo IP all'hostname **epicode.internal** (**FIG.2**).

**FIG.2 - DNS STATICO**

```
#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
dns_static epicode.internal 192.168.32.100
```

Avviando in seguito la simulazione del server virtuale tramite comando "**sudo inetsim**", è possibile verificare la corretta attivazione dei servizi richiesti (**FIG.3**).

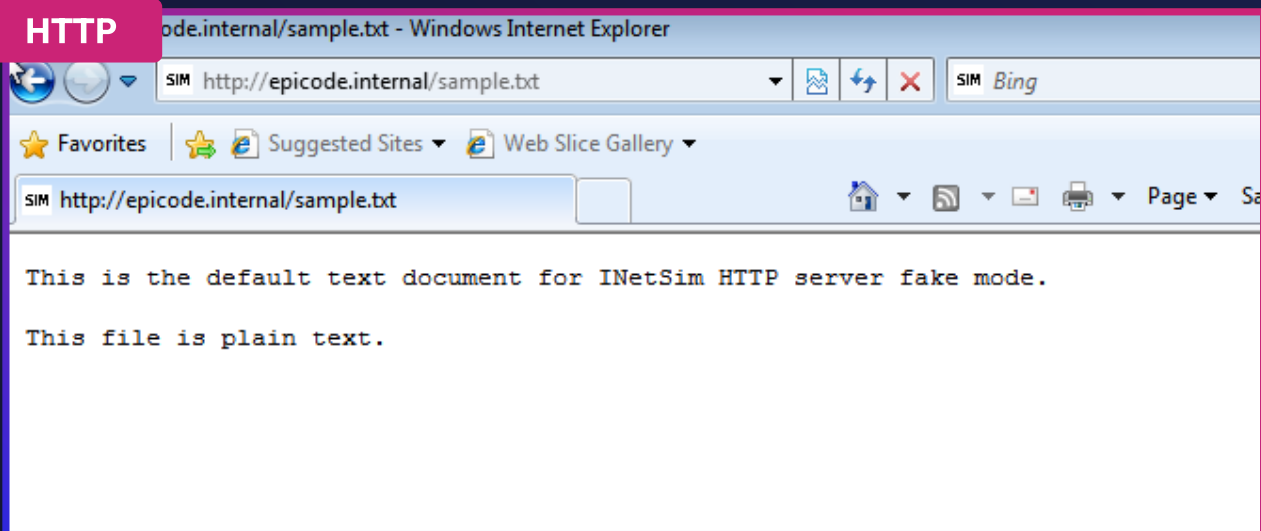
**FIG.3 - AVVIO INETSIM**

```
Listening on: 0.0.0.0
Real Date/Time: 2023-11-19 05:20:20
Fake Date/Time: 2023-11-19 05:20:20 (Delta: 0 sec)
Forking services ...
* dns_53_tcp_udp - started (PID 1949)
print() on closed filehandle MLOG at /usr/share/p
* http_80_tcp - started (PID 1950)
print() on closed filehandle MLOG at /usr/share/p
* https_443_tcp - started (PID 1951)
done.
Simulation running.
```

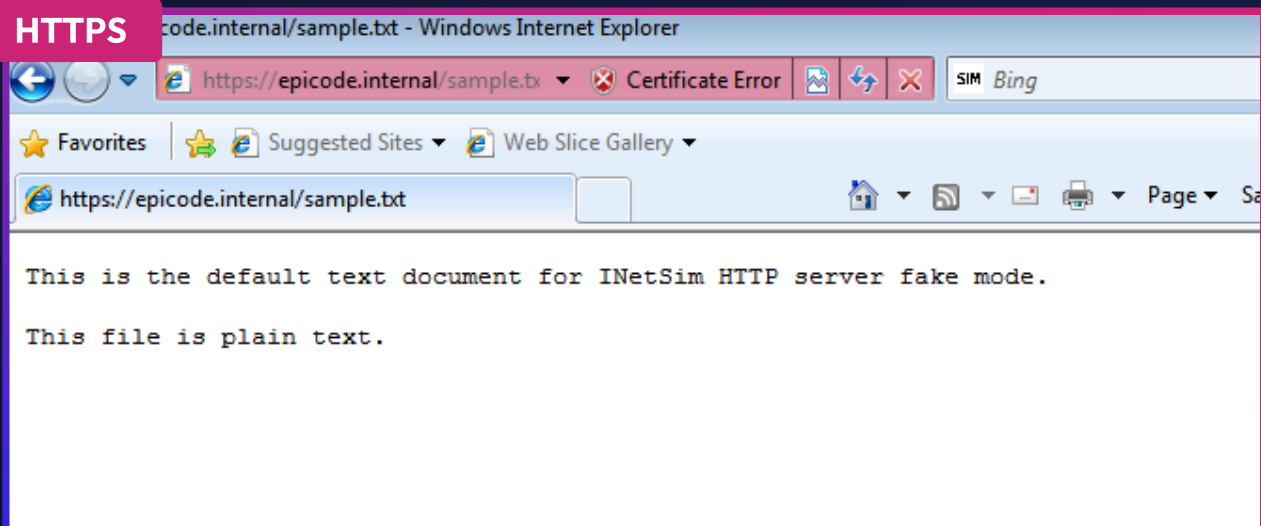
## RICHIESTA FILE SU SERVER VIRTUALE

Tramite Windows Internet Explorer su Windows 7, è possibile verificare il corretto funzionamento della configurazione e del server web virtuale richiedendo la risorsa “**sample.txt**” su host “**epicode.internal**”, sia su server **HTTP** che **HTTPS**.

### HTTP



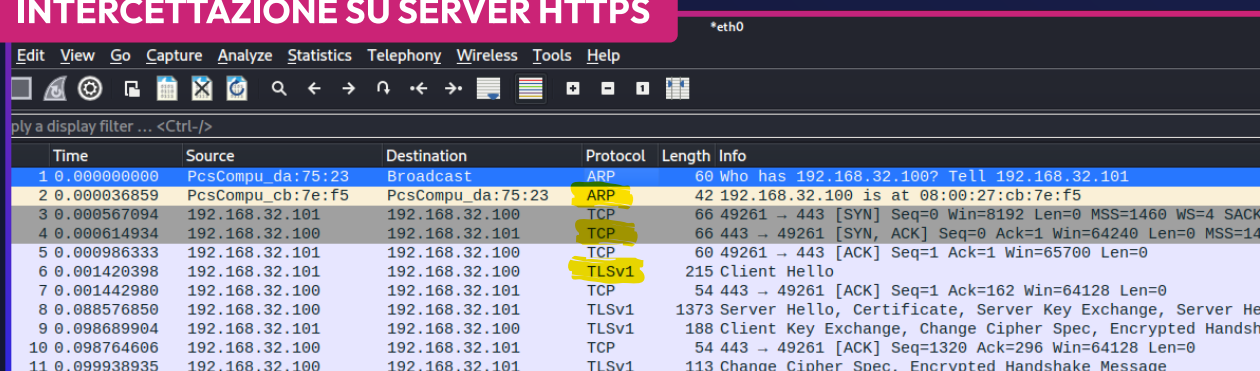
### HTTPS



# INTERCETTAZIONE PACCHETTI SU WIRESHARK

Utilizzando **Wireshark**, uno strumento di analisi del traffico di rete, sono stati **catturati ed esaminati i pacchetti** inviati durante questi processi su rete **eth0**.

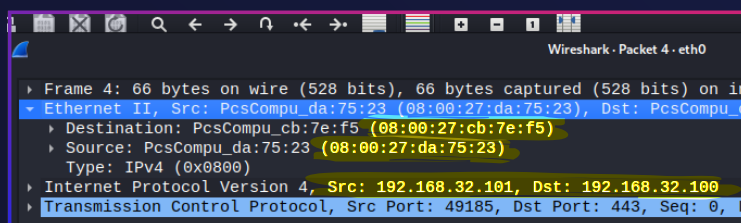
## INTERCETTAZIONE SU SERVER HTTPS



Time	Source	Destination	Protocol	Length	Info
1 0.000000000	PcsCompu_da:75:23	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2 0.000036859	PcsCompu_cb:7e:f5	PcsCompu_da:75:23	ARP	42	192.168.32.100 is at 08:00:27:cb:7e:f5
3 0.000567094	192.168.32.101	192.168.32.100	TCP	66	49261 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK
4 0.000614934	192.168.32.100	192.168.32.101	TCP	66	443 → 49261 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=14
5 0.000986333	192.168.32.101	192.168.32.100	TCP	60	49261 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6 0.001420398	192.168.32.101	192.168.32.100	TLSv1	215	Client Hello
7 0.001442980	192.168.32.100	192.168.32.101	TCP	54	443 → 49261 [ACK] Seq=1 Ack=162 Win=64128 Len=0
8 0.008576850	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exchange, Server He
9 0.008689904	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handsh
10 0.008764606	192.168.32.100	192.168.32.101	TCP	54	443 → 49261 [ACK] Seq=1320 Ack=296 Win=64128 Len=0
11 0.00938935	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message

L'intercettazione inizia con il **protocollo ARP**, tramite cui il sistema effettua la **risoluzione dell'indirizzo IP** a partire dall'indirizzo MAC noto. La comunicazione continua tramite uso del **protocollo TCP** per gestire la trasmissione affidabile dei dati e l'utilizzo del **protocollo TLS** manifestando la protezione dello scambio d'informazioni attraverso la sequenza del **three-way-handshake (SYN-SYN/ACK-ACK)**. Questa rappresenta il momento in cui il client e il server stabiliscono una connessione crittografata mediante lo scambio di chiavi, il che **non consente di vedere in modo chiaro il tipo di richiesta fatta dal client al server e lo scambio d'informazioni tra essi**.

Nonostante questo, analizzando i singoli frame, siamo comunque in grado di vedere gli **indirizzi IP e MAC** coinvolti in questa comunicazione.



Frame 4: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0
Ethernet II, Src: PcsCompu_da:75:23 (08:00:27:da:75:23), Dst: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5)
Source: PcsCompu_da:75:23 (08:00:27:da:75:23)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100
Transmission Control Protocol, Src Port: 49185, Dst Port: 443, Seq: 0, Len: 0



## INTERCETTAZIONE SU SERVER HTTP

\*eth0

Time	Source	Destination	Protocol	Length	Info
1 0.000000000	192.168.32.101	192.168.32.100	TCP	66	49222 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_
2 0.000052574	192.168.32.100	192.168.32.101	TCP	66	80 → 49222 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=146
3 0.000510197	192.168.32.101	192.168.32.100	TCP	60	49222 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4 0.000873243	192.168.32.101	192.168.32.100	HTTP	371	GET /sample.txt HTTP/1.1
5 0.000892515	192.168.32.100	192.168.32.101	TCP	54	80 → 49222 [ACK] Seq=1 Ack=318 Win=64128 Len=0
6 0.054333567	192.168.32.100	192.168.32.101	TCP	204	80 → 49222 [PSH, ACK] Seq=1 Ack=318 Win=64128 Len=150 [TC
7 0.062416412	192.168.32.100	192.168.32.101	HTTP	151	HTTP/1.1 200 OK (text/plain)
8 0.063013035	192.168.32.101	192.168.32.100	TCP	60	49222 → 80 [ACK] Seq=318 Ack=249 Win=65452 Len=0
9 0.063273436	192.168.32.101	192.168.32.100	TCP	60	49222 → 80 [FIN, ACK] Seq=318 Ack=249 Win=65452 Len=0
10 0.063302781	192.168.32.100	192.168.32.101	TCP	54	80 → 49222 [ACK] Seq=249 Ack=319 Win=64128 Len=0

Nel caso di richiesta al server **HTTP**, i frame invece sono evidenti, mostrando la richiesta **GET** del client e la risposta positiva del server (**200 OK**), questo perchè non vi è l'utilizzo del protocollo TLS e il relativo scambio di chiavi cifrate tra le due parti coinvolte; questo fa sì inoltre che la comunicazione HTTP sia **chiaramente leggibile nel formato text/plain**, rendendo agevole l'analisi del contenuto del file preso in oggetto (sample.txt).

## TEXT/PLAIN

```

... on wire (1208 bits), 151 bytes captured (1208 bits) on interface eth0, id 0
  ▶ Ethernet II, Src: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5), Dst: PcsCompu_da:75:23 (08:00:27:da:75:23)
  ▶ Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101
  ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 49159, Seq: 151, Ack: 429, Len: 97
  ▶ [2 Reassembled TCP Segments (247 bytes): #8(150), #9(97)]
  ▶ Hypertext Transfer Protocol
    ▶ Line-based text data: text/plain (5 lines)
      \n
      This is the default text document for INetSim HTTP server fake mode.\n
      \n
      This file is plain text.\n
      \n

```