Report di Laboratorio: Simulazione di Comunicazione Client-Server con Wireshark

Obiettivo: Simulare un'architettura client-server in cui un client (Windows 7) richiede una risorsa a un server (Kali Linux) tramite browser, intercettare la comunicazione con Wireshark e confrontare il traffico HTTPS con HTTP.

Configurazione di Rete:

- Kali Linux: IP 192.168.32.100

- Windows 7: IP 192.168.32.101

- Inserimento dell'IP DNS su Windows 7: IP 192.168.32.100

```
# ident, syslog, dummy_tcp,
# ftps, irc, https
#
start_service dns
#start_service http
start_service https
#start_service smtp
#start_service smtps
#start_service pop3
#start_service pop3s
#start_service ftp
#start_service ftps
#start_service tftp
#start_service irc
#start_service ntp
#start_service finger
#start_service ident
#start_service syslog
#start_service time_tcp
#start_service time_udp
#start_service daytime_tcp
#start_service daytime_udp
#start_service echo_tcp
```

- HTTPS server: Attivo

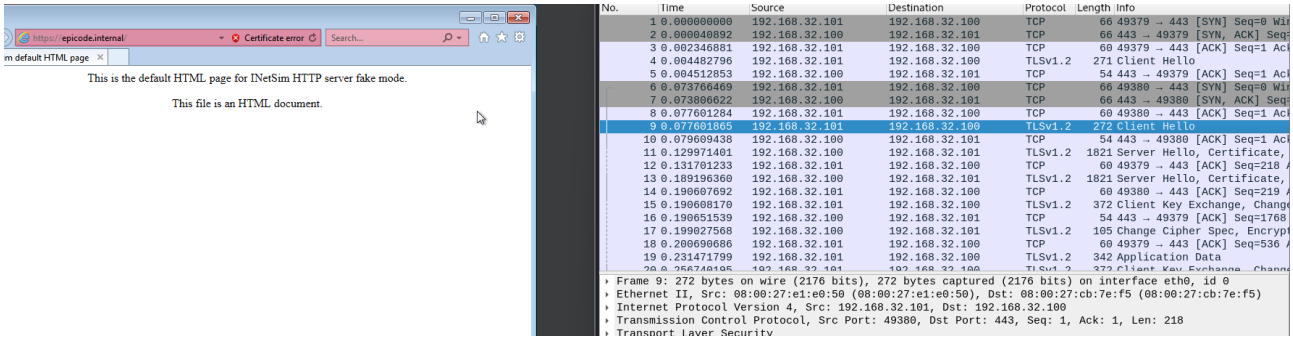
- Servizio DNS: Attivo

- Altri servizi spenti

Fase 1: Comunicazione tramite HTTPS

1. Configurazione Iniziale:

- Kali Linux e Windows 7 sono connessi nella stessa rete virtuale.



2. Richiesta HTTPS:

- Il client (Windows 7) richiede tramite web browser una risorsa all'hostname "epicode.internal" al server HTTPS (Kali Linux).

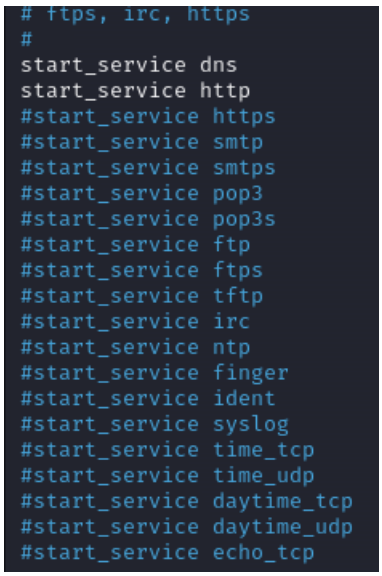
3. Wireshark Capture:

- Utilizzando Wireshark, abbiamo intercettato la comunicazione.

- Sorgente MAC: [MAC_Address_Client], Destinazione MAC: [MAC_Address_Server]

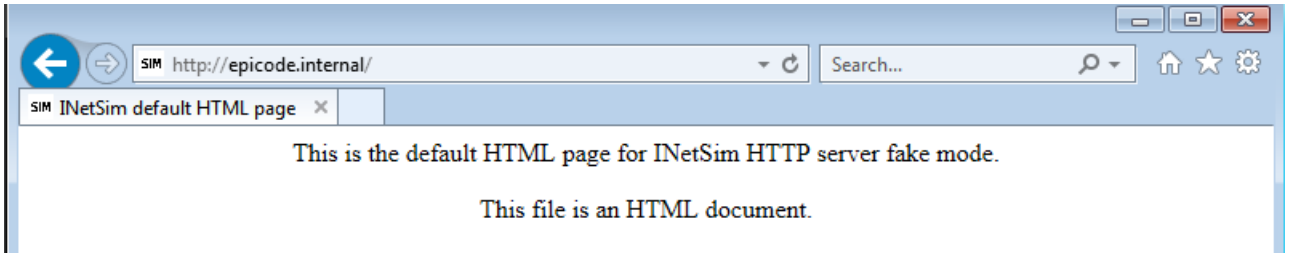
- Contenuto richiesta HTTPS: [Contenuto_Richiesta_HTTPS]

Fase 2: Comunicazione tramite HTTP



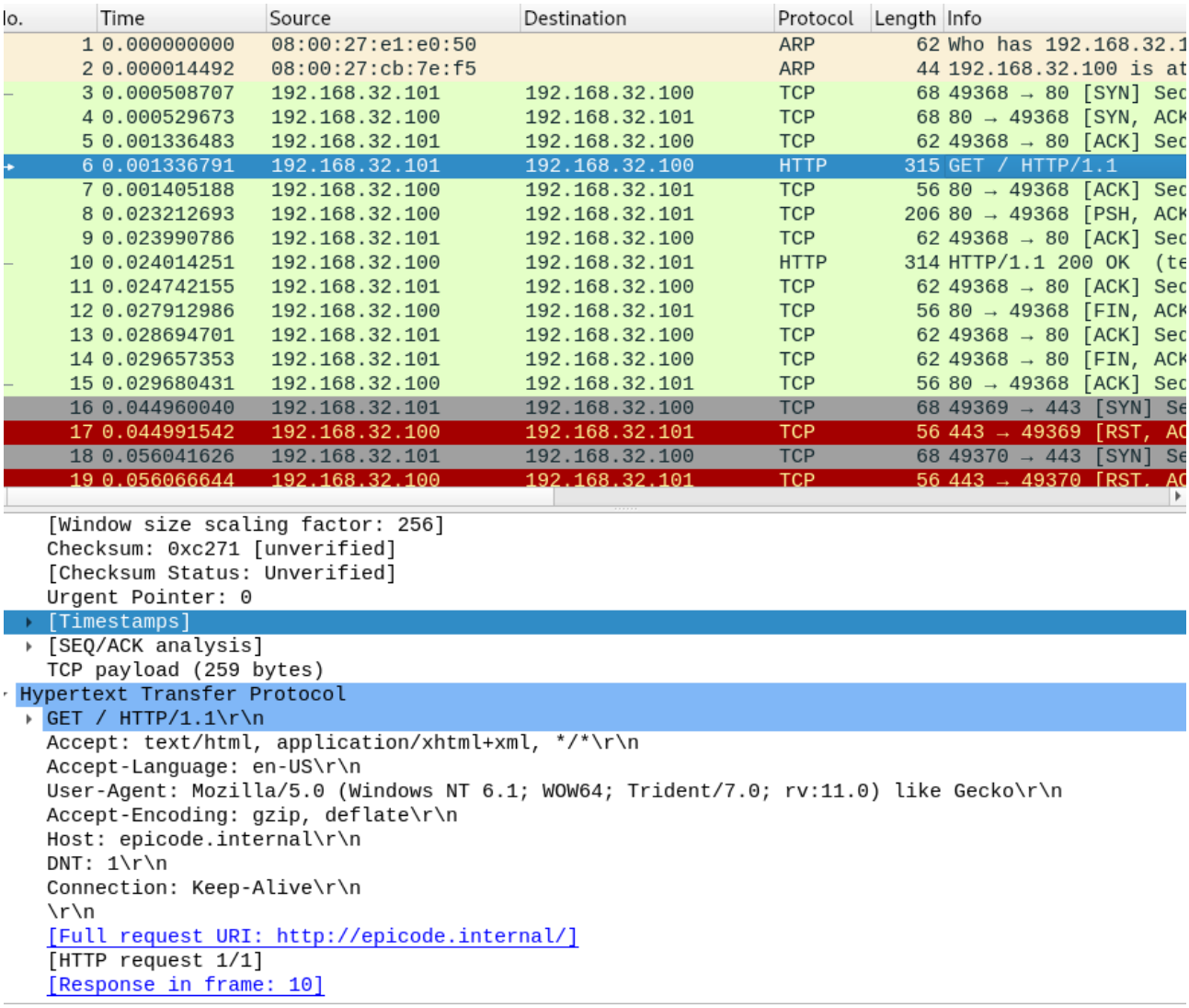
1. Modifica del Server:

- Ho sostituito il server HTTPS con un server HTTP su Kali Linux.



2. Richiesta HTTP:

- Il client (Windows 7) effettua nuovamente la richiesta tramite web browser.



lo.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	08:00:27:e1:e0:50		ARP	62	Who has 192.168.32.1
2	0.000014492	08:00:27:cb:7e:f5		ARP	44	192.168.32.100 is at
3	0.000508707	192.168.32.101	192.168.32.100	TCP	68	49368 → 80 [SYN] Seq
4	0.000529673	192.168.32.100	192.168.32.101	TCP	68	80 → 49368 [SYN, ACK
5	0.001336483	192.168.32.101	192.168.32.100	TCP	62	49368 → 80 [ACK] Seq
6	0.001336791	192.168.32.101	192.168.32.100	HTTP	315	GET / HTTP/1.1
7	0.001405188	192.168.32.100	192.168.32.101	TCP	56	80 → 49368 [ACK] Seq
8	0.023212693	192.168.32.100	192.168.32.101	TCP	206	80 → 49368 [PSH, ACK
9	0.023990786	192.168.32.101	192.168.32.100	TCP	62	49368 → 80 [ACK] Seq
10	0.024014251	192.168.32.100	192.168.32.101	HTTP	314	HTTP/1.1 200 OK (te
11	0.024742155	192.168.32.101	192.168.32.100	TCP	62	49368 → 80 [ACK] Seq
12	0.027912986	192.168.32.100	192.168.32.101	TCP	56	80 → 49368 [FIN, ACK
13	0.028694701	192.168.32.101	192.168.32.100	TCP	62	49368 → 80 [ACK] Seq
14	0.029657353	192.168.32.101	192.168.32.100	TCP	62	49368 → 80 [FIN, ACK
15	0.029680431	192.168.32.100	192.168.32.101	TCP	56	80 → 49368 [ACK] Seq
16	0.044960040	192.168.32.101	192.168.32.100	TCP	68	49369 → 443 [SYN] Se
17	0.044991542	192.168.32.100	192.168.32.101	TCP	56	443 → 49369 [RST, AC
18	0.056041626	192.168.32.101	192.168.32.100	TCP	68	49370 → 443 [SYN] Se
19	0.056066644	192.168.32.100	192.168.32.101	TCP	56	443 → 49370 [RST, AC

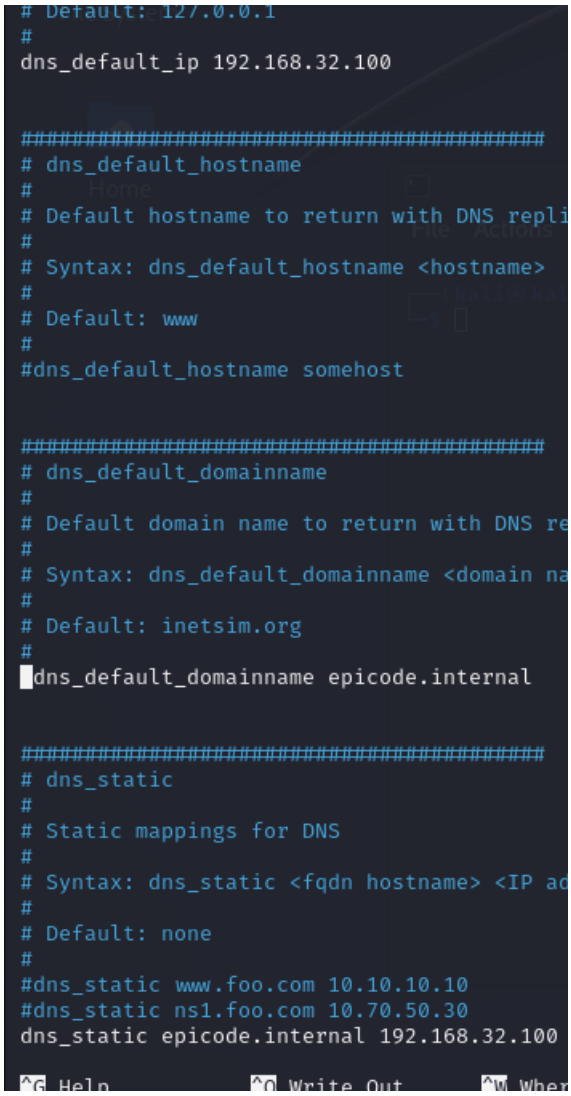
```
    [Window size scaling factor: 256]
    Checksum: 0xc271 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▸ [Timestamps]
  ▸ [SEQ/ACK analysis]
    TCP payload (259 bytes)
▾ Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    Accept: text/html, application/xhtml+xml, */*\r\n
    Accept-Language: en-US\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko\r\n
    Accept-Encoding: gzip, deflate\r\n
    Host: epicode.internal\r\n
    DNT: 1\r\n
    Connection: Keep-Alive\r\n
    \r\n
    [Full request URI: http://epicode.internal/]
    [HTTP request 1/1]
    [Response in frame: 10]
```

3. Wireshark Capture (HTTP):

- La comunicazione è stata intercettata nuovamente.

- Sorgente MAC: [MAC_Address_Client], Destinazione MAC: [MAC_Address_Server]

- Contenuto richiesta HTTP: [Contenuto_Richiesta_HTTP]



I servizi DNS:

-DNS default IP: 192.168.32.100, visto che Kali fa da server, nel nell'IP del DNS va inserito il suo IP

-Il Domainname è epcode.internal come richiesto

-Nel DNS static viene assiciato il Domainname all'IP: epicode.internal

Analisi e Confronto:

- Nel traffico HTTPS, il contenuto della richiesta è crittografato, mentre nel traffico HTTP, il contenuto è leggibile.

- I MAC address di sorgente e destinazione sono gli stessi in entrambe le fasi.

Conclusioni:

- La principale differenza tra il traffico HTTPS e HTTP è la sicurezza.

- HTTPS garantisce una comunicazione crittografata, mentre HTTP trasmette i dati in chiaro.