Report esercitazione finale M1- Architettura Client-Server e Analisi traffico HTTP/HTTPS

Traccia:

Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (windows) 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.

Premessa:

Nelle lezioni precedenti ho configurato kali e windows su virtual box con rete interna. Per lo svolgimento dell'esercizio ho configurato gli IP di entrambi, in modo manuale, assegnando a kali

un IP: 192.168.32.100 e a windows un IP: 192.168.32.101. Subito dopo ho testato la comunicazione tra i due tramite comando "**ping"**, avvenuta con successo. Sapendo che il tool inetsim non permetteva il corretto funzionamento del DNS, ho cercato possibili soluzioni su internet ed ho trovato alternative come "**dnsmasq"**, grazie alla quale ho potuto svolgere l'esercizio per quanto riguarda la parte DNS. Per quanto riguarda la configurazione del server HTTPS e HTTP ho invece individuato il software "**Apache2"**.

Svolgimento:

1. Installazione Apache e dnsmasq su kali

comandi:

sudo apt update install apache2

sudo apt update install dnsmasq

2. Configurazione Apache per HTTPS:

-step1. Ho generato un certificato SSL autofirmato.

Comandi:

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt

-step2. Ho creato un file di configurazione per virtual host Apache2.

Comandi:

sudo nano /etc/apache2/sites-available/epicode.internal.conf

Aggiungendo le stringhe:

```
<VirtualHost *:443>
    ServerName epicode.internal
    DocumentRoot /var/www/epicode
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/epicode.crt
    SSLCertificateKeyFile /etc/ssl/private/epicode.key
</VirtualHost>
```

-step3. Ho abilitato il virtual host e fatto il restart di Apache2 per applicare le modifiche.

Comandi:

sudo a2ensite epicode.internal.conf

sudo systemctl restart apache2

3. Configurazione dnsmaq

-step1. Ho configurato dnsmasq per risolvere epicode.internal.

Comandi:

sudo nano /etc/dnsmasq.conf

Aggiungendo le seguenti stringhe:

```
address=/epicode.internal/192.168.32.100
listen-address=192.168.32.100
```

-step2. Ho avviato il servizio.

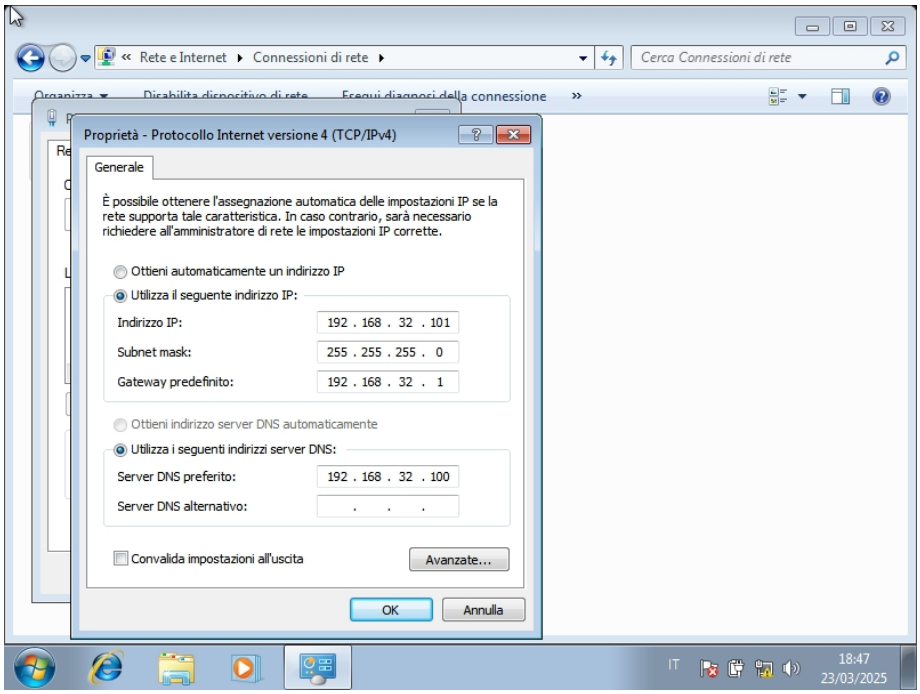
Comandi:

```
sudo systemctl restart dnsmasq
```

4. Verifica DNS server da windows

-step1. Su windows ho configurato l'IP 192.168.32.100 (kali) come dns preferito nelle impostazione di rete della scheda di rete.

Vedi figura:



-step2. Ho verificato la corretta configurazione con il comando:

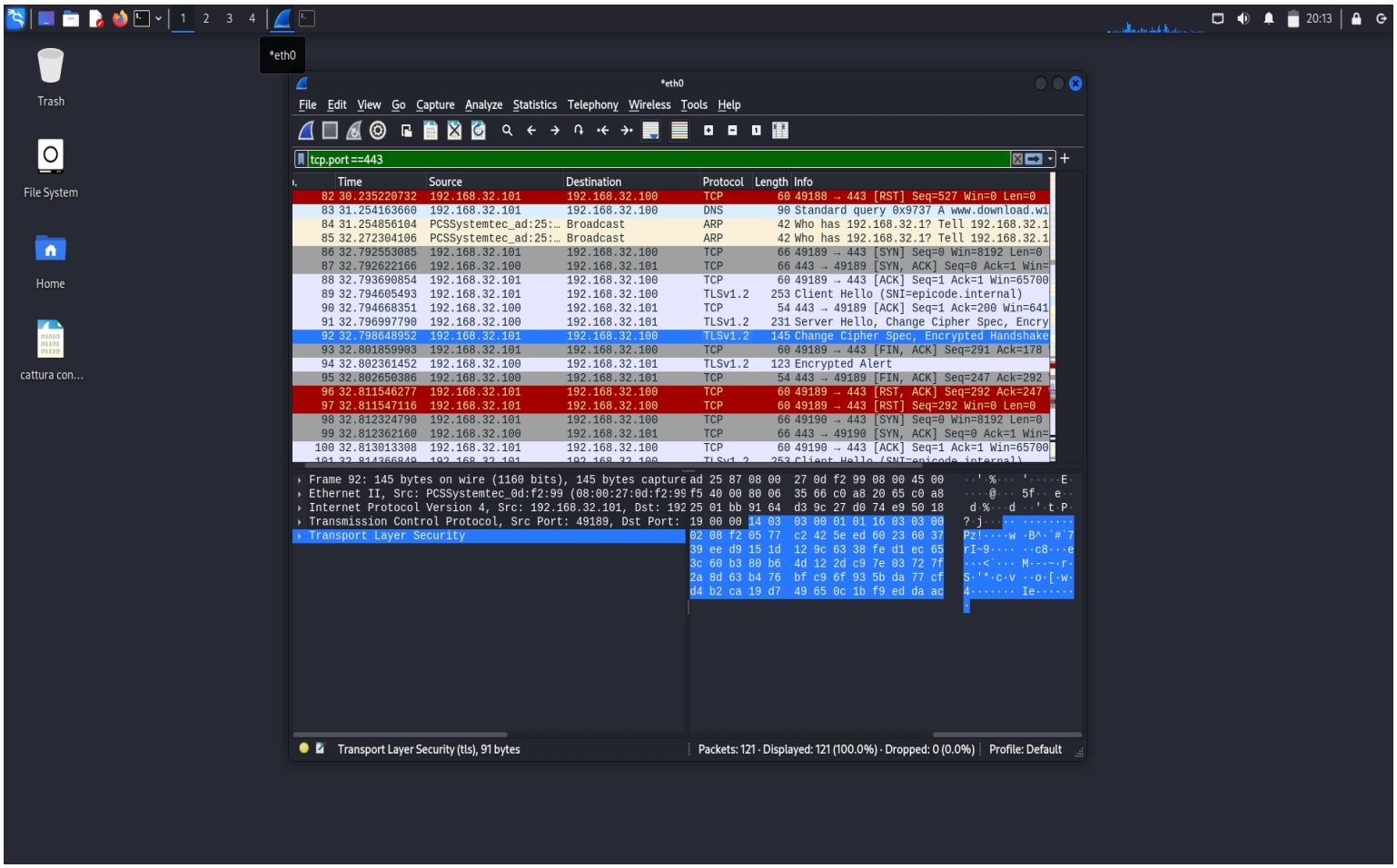
nslookup epicode.internal

5. Test HTTPS da windows

Ho aperto il browser e navigato su https://epicode.internal. Il browser (essendo auto firmato) ha mostrato un avviso di certificato non attendibile. Ho proceduto accettando il rischio ed ho visualizzato la pagina.

6. Analisi traffico HTTPS con Wireshark

Cattura del traffico HTTPS:



Come mostrato in figura abbiamo:

-Indirizzo di sorgente: 192.168.32.101 (windows).

-Indirizzo di destinazione: 192.168.32.100 (kali).

-Handshake TLS.

-Contenuto del messaggio cifrato.

7. Riconfigurazione Apache per HTTP

Ho disabilitato HTTPS eliminando alcune stringhe scritte in precedenza e cambiando la porta “443” con la porta “80” per abilitare HTTP.

Comandi:

sudo nano /etc/apache2/sites-available/epicode.internal.conf

Modificando le stringhe:

<VirtualHost *:80>

ServerName epicode.internal

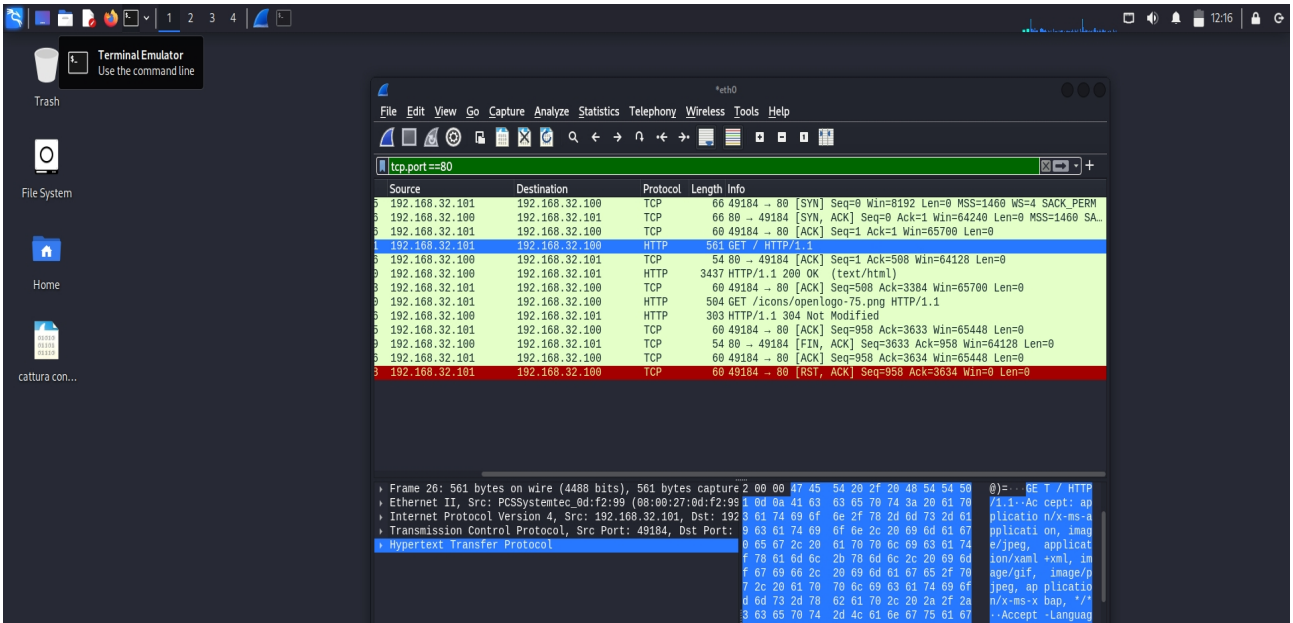
DocumentRoot /var/www/epicode

</VirtualHost>

8. Test HTTP da Windows

Ho nuovamente aperto il browser e stavolta digitando http://epicode.internal, notando che la connessione è stata stabilita senza avvisi di sicurezza.

9. Analisi cattura traffico HTTP con Wireshark



Come mostrato in figura abbiamo:

-Indirizzo di sorgente: 192.168.32.101 (windows).

-Indirizzo di destinazione: 192.168.32.100 (kali).

-Richieste HTTP in chiaro.

-Contenuto della comunicazione visibile.

Conclusione.

Differenze tra HTTPS e HTTP

Le principali differenze riscontrate sono:

-HTTP trasmette tutto in chiaro, permettendo la visualizzazione del contenuto.

-HTTPS utilizza TLS per cifrare la comunicazione, rendendo illeggibile il contenuto dei pacchetti.

-HTTP utilizza la porta 80, mentre HTTPS utilizza la porta 443 e richiede un certificato SSL.

-HTTP non offre nessuna protezione della privacy o integrità dei dati trasmessi.

Si consiglia sempre di usare HTTPS, specialmente in ambienti con dati sensibili.