Progetto Finale - Primo Mese

Indice:

- 1) Configurazione Indirizzi IP Kali e Windows
- 2) Configurazione HTTPS E DNS
- 3) Intercettazione Comunicazione HTTPS e MAC address
- 4) Intercettazione Comunicazione http

1) Configurazione Indirizzi IP Kali e Window

Come primo passaggio fondamentale per una simulazione corretta *client-server* ho configurato i seguenti indirizzi IP delle due macchine virtuali:

Kali: 192.168.32.100

Windows: 192.168.32.101

Successivamente ho eseguito un test del *ping* per controllare il corretto funzionamento dell'ambiente di laboratorio virtuale.

```
(anto⊕ anto)-[~]

$ 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=0.639 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=1.06 ms
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.634 ms
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.549 ms
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.57 ms
64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=0.521 ms
64 bytes from 192.168.32.101: icmp_seq=7 ttl=128 time=0.502 ms
^7

zsh: suspended ping 192.168.32.101

[anto⊕ anto)-[~]
```

```
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\antopc>ping 192.168.32.100

Esecuzione di Ping 192.168.32.100 con 32 byte di dati:
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Statistiche Ping per 192.168.32.100:
Pacchetti: Trasmessi = 4, Ricevuti = 4,
Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
Minimo = 0ms, Massimo = 0ms, Medio = 0ms

C:\Users\antopc>
```

2) Configurazione HTTPS e DNS

Successivamente attraverso lo strumento *inetsim* che permettente di simulare tutto una serie di servizi riguardanti la rete configuro il **DNS** attivando il relativo servizio e l'HTTPS. La configurazione del DNS è un passaggio fondamentale poiché funge da tramite tra la richiesta di un domino in questo caso epicode.internal e l'indirizzo ip 192.168.32.100.

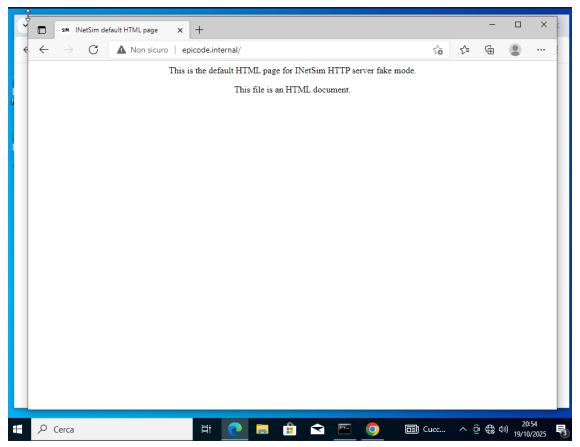
```
# ftps, irc, https
start_service dns
# start_service http
start_service smtn
# start_service smtn
```

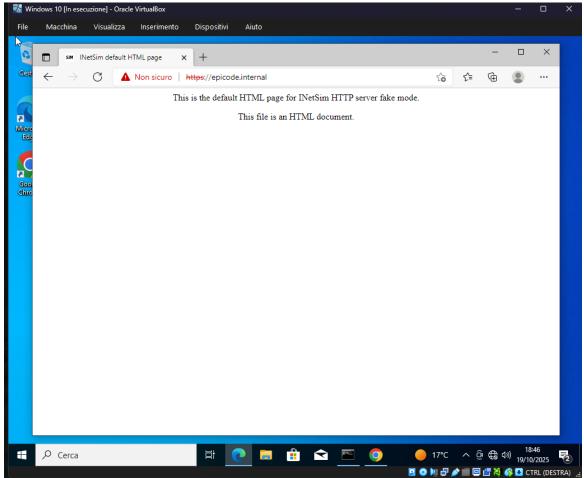
Una volta fatto ciò proseguo con il configurare più nel dettaglio la configurazione DNS:

- come riferimento DNS: 192.168.32.100 e
- -come dominio e nome: epicode.internal

In questo modo le due macchine possono comunicare correttamente.

Una volta completata la configurazione mi sono spostato sulla macchina virtuale Windows per eseguire un test di connessione con il dominio *epicode.internal*, sia in HTTPS che in HTTP, verificando il corretto funzionato e risoluzioni della connessione:





3) Intercettazione Comunicazione HTTPS e MAC address

Dopo aver verificato il corretto indirizzamento con il dominio epicode.internal, ho analizzato nel dettaglio il protocollo **HTTPS** e il **MAC address** delle due macchine virtuali. Per fare questo tipo di analisi dettagliata mi sono servito di un software specifico *wireshark* che analizza tutti i tipi di pacchetti che vengono scambiati tra i due Host.

```
Wireshark · Pacchetto 107 · eth0

Frame 107: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, if the Ethernet II, Src: PCSSystemtec_a2:af:8c (08:00:27:a2:af:8c), Dst: PCSSystemtec_b0:ea:64 (08:00:27:b0:ea:64)

Destination: PCS
```

Come possiamo notare da quest'immagine abbiamo trovato nel livello **Ethernet II** il MAC address di destinazione che quello della fonte. Importante soffermarsi ad analizzare il mac address poichè si tratta di un identificatore univoco legato al singolo dispositivo fisico ed è importate poiché ci permette di far comunicare in modo corretto le due macchine virtuali.

Successivamente grazie a Wireshark possiamo analizzare anche più nel dettaglio il protocollo **HTTPS**:

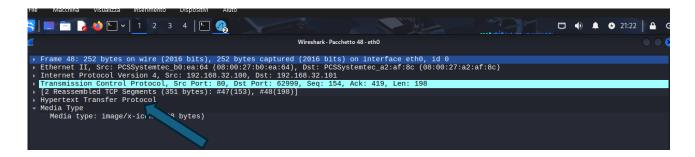
```
108 24.719378119 192.168.32.100 192.168.32.101 TLSv1.2 61 Alert (Level: Fatal, Description: 109 24.720488688 192.168.32.101 192.168.32.100 TCP 60 58885 - 443 [RST, ACK] Seq=196 Ac 110 30.692109496 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 111 31.567355974 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 112 32.567665143 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 112 32.567665143 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystemtec_a2:af:... Broadcast ARP 60 Who has 192.168.32.17 Tell 192.16 PCSSystem PCSS
```

Questo protocollo è essenziale ai fini della trasmissione sicura delle infromazioni su internet, in quanto utilizza la crittografia per proteggere i dati, a differenza del protocollo HTTP.

Come mostrato dell'analisi, l'HTTPS possiede un ulteriore livello chiamato *Transport lateyer* Security (**TLS**) che permette di oscurare le informazioni. Solo chi è in possesso delle chiavi di cifratura può visualizzare le informazioni, garantendo la sicurezza nella trasmissione.

4) Intercettazione Comunicazione http

Analizzando invece il protocollo HTTP è possibile notare come, non essendo cifrato, ci permetta di visualizzare molte più informazioni anche dopo una singola analisi del traffico. Questo dimostra quanto sia importante avere un protocollo cifrato come l'https per proteggere i nostri dati che scambiamo.



Nel caso dell'HTTP, infatti, non è presente il layer di sicurezza e si può chiaramente identificare il protocollo di trasferimento e il **media Type**, ovvero la categoria generale del dato trasmesso. (in questo caso vediamo trattarsi di un'immagine)

Antonio Amatrice 19/10/2025