

# Progetto Finale M1

Intercettare, mediante l'utilizzo di Wireshark, il traffico HTTPS tra Kali e Windows con l'URL `epicode.internal`.

Il fine del progetto è quello di intercettare, mediante Wireshark, il traffico criptato tra la VM Kali e Windows con la connessione all'URL `epicode.internal`.

Per prima cosa imposto un IPV4 statico ad entrambe le VM. Per la Kali viene impostato un IP `192.168.32.100`, mentre per Windows IP `192.168.32.101`

Se provo ad eseguire un ping tra le macchine vedo che funziona quindi riescono a comunicare. (Figura 1).

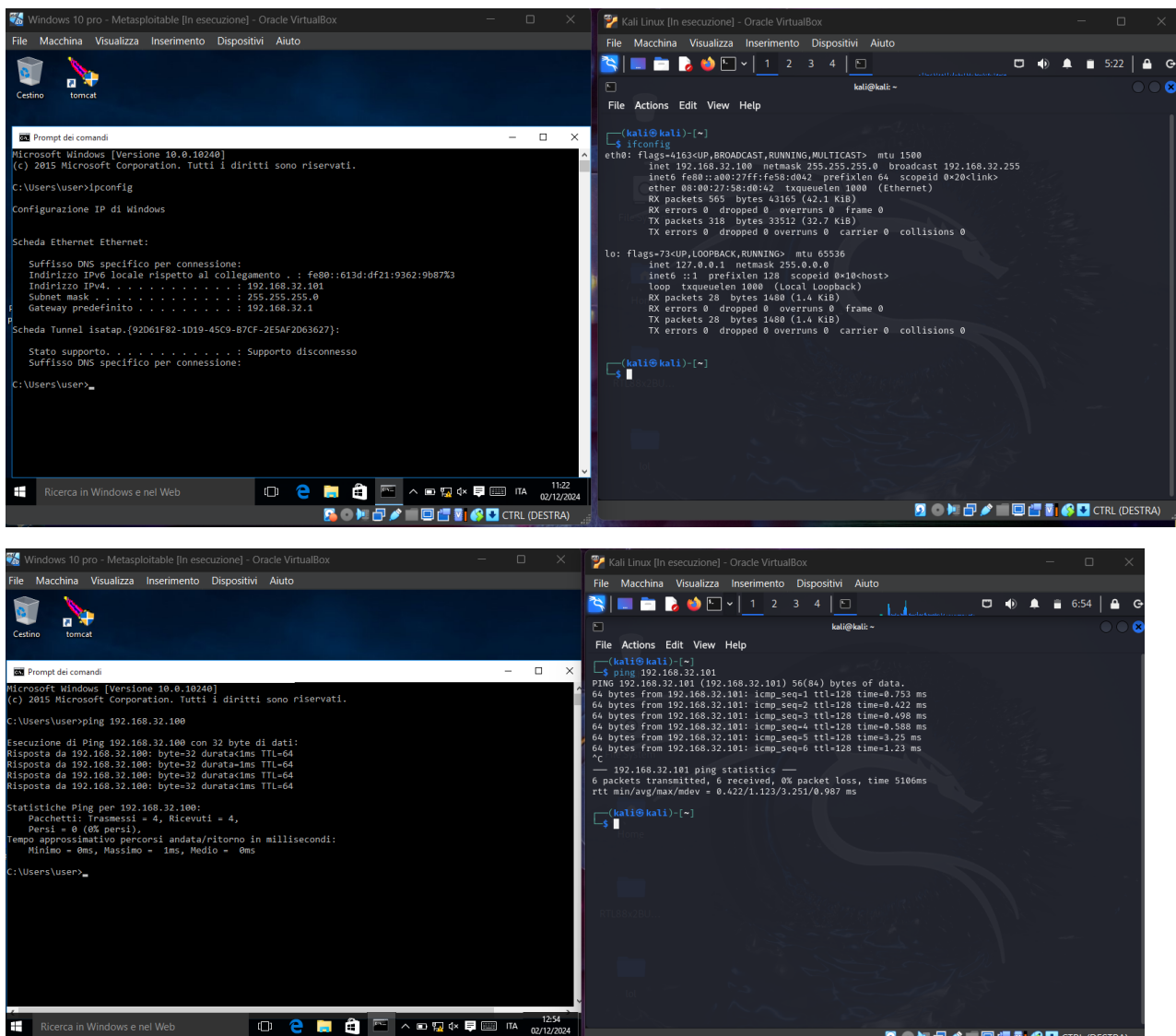
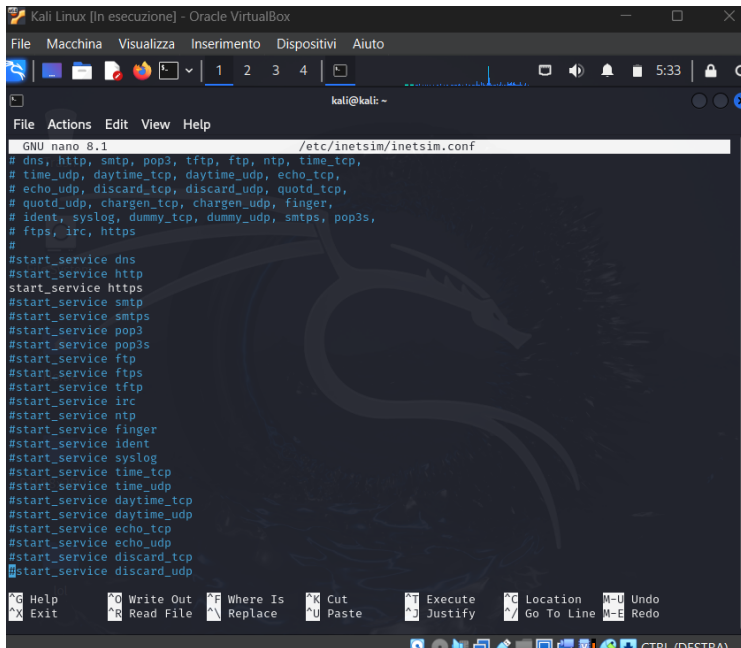


Figura 1

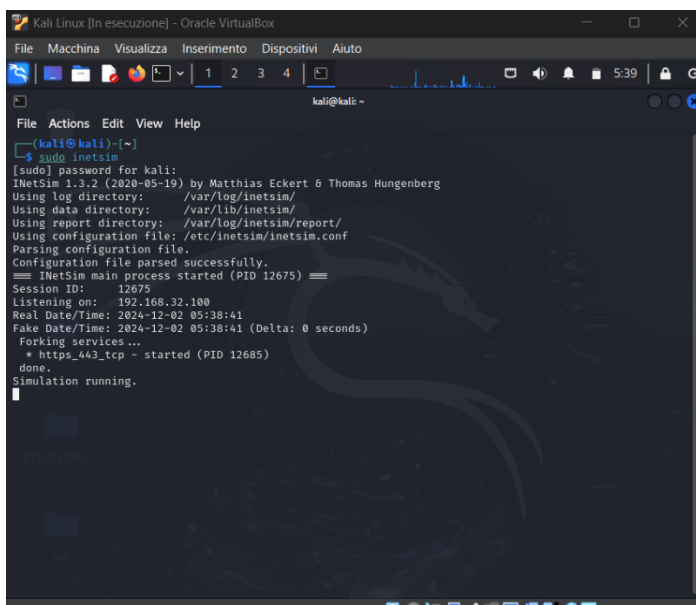
Dovrò simulare il traffico HTTPS quindi, sulla VM Kali avvio utilizzerò l'applicativo **inetsim**. Per usare tale applicativo devo prima modificare il file di configurazione per permettere al servizio di funzionare solamente con il traffico HTTPS. Per fare ciò, sulla VM Kali, userò il comando `sudo nano /etc/inetsim/inetsim.conf`. A questo file di configurazione andrò a disattivare i servizi che, al momento, non sono necessari per lo svolgimento della prova, mediante l'apposizione di un `#` prima del servizio (Figura 2).



```
GNU nano 8.1 /etc/inetsim/inetsim.conf
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# 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,
# 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
#start_service echo_udp
#start_service discard_tcp
#start_service discard_udp
```

Figura 2

Una volta modificato il file di configurazione devo attivare il servizio mediante il comando `sudo inetsim`



```
(kali@kali)~$ sudo inetsim
[sudo] password for kali:
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 12675) ==
Session ID: 12675
Listening on: 192.168.32.100
Real Date/Time: 2024-12-02 05:38:41
Fake Date/Time: 2024-12-02 05:38:41 (Delta: 0 seconds)
Forking services ...
+ https_443_tcp - started (PID 12685)
done.
Simulation running.
```

Figura 3

A questo punto devo attivare il servizio DNS per poter risolvere l'indirizzo **epicode.internal** in indirizzo IP 192.168.32.100. Per fare ciò utilizzo l'applicativo dnscchef mediante il comando

```
dnscchef --fakedomains epicode.internal --fakeip 192.168.32.100 -i 192.168.32.100
```

Adesso utilizzerò la VM Windows per poter verificare se il lavoro svolto funziona correttamente. Apro il browser Microsoft Edge ed inserisco l'indirizzo <https://epicode.internal>. Qui compare l'avviso di sicurezza relativo al certificato ssl (*Figura 4*), clicco su prosegui e vado avanti e mi viene mostrata la pagina HTML del server presente sulla kali (*Figura 5*).

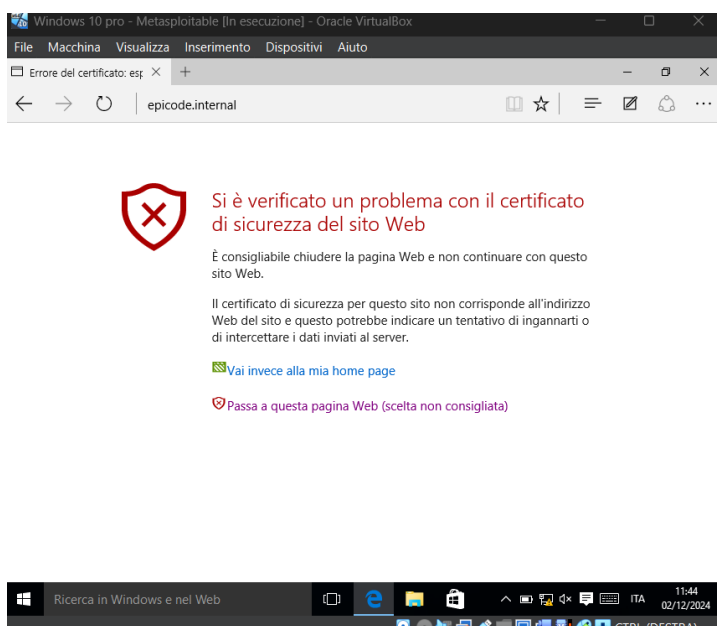


Figura 4

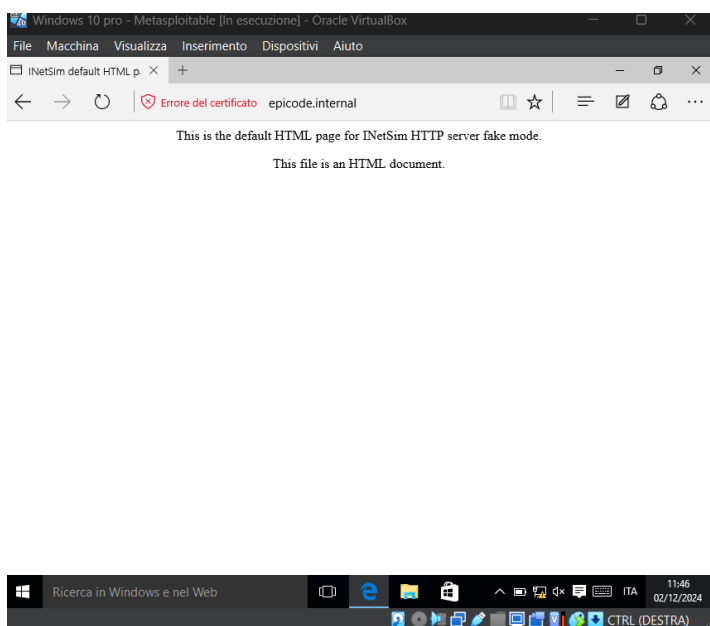


Figura 5

A questo punto chiudo il browser ed attivo Wireshark per poter catturare il traffico HTTPS che avviene tra le 2 VM. Appena attivato Wireshark e selezionato l'interfaccia eth0, viene mostrato lo scambio dati che avviene tra le macchine (Figura 6).

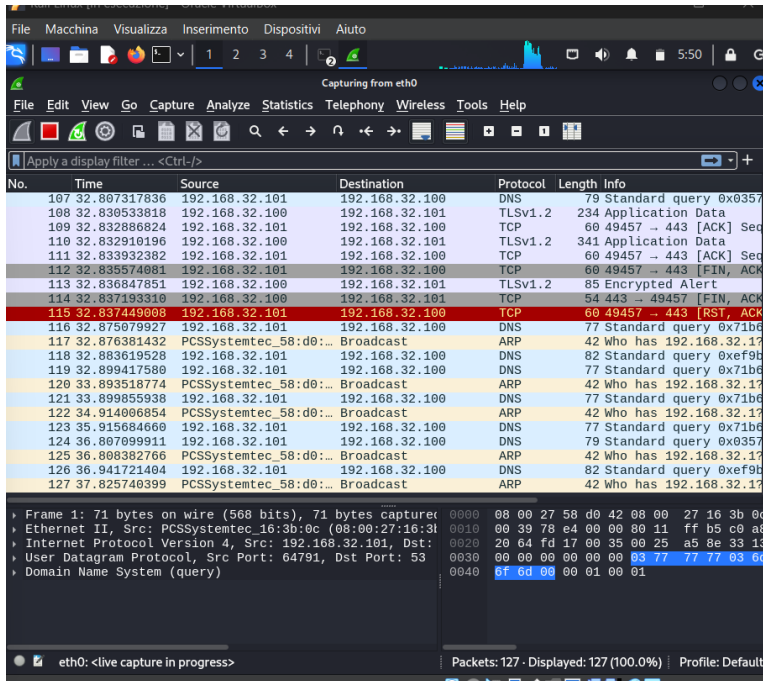


Figura 6

Dopo aver fatto comunicare le 2 macchine, ho impostato il filtro (tcp.port==443) per poter visualizzare solamente il traffico https. Selezionando un pacchetto tra quelli filtrati si viene a determinare l'indirizzo MAC sorgente e di destinazione, indirizzi IP sorgenti e destinazione insieme al contenuto della richiesta. (Figura 7)

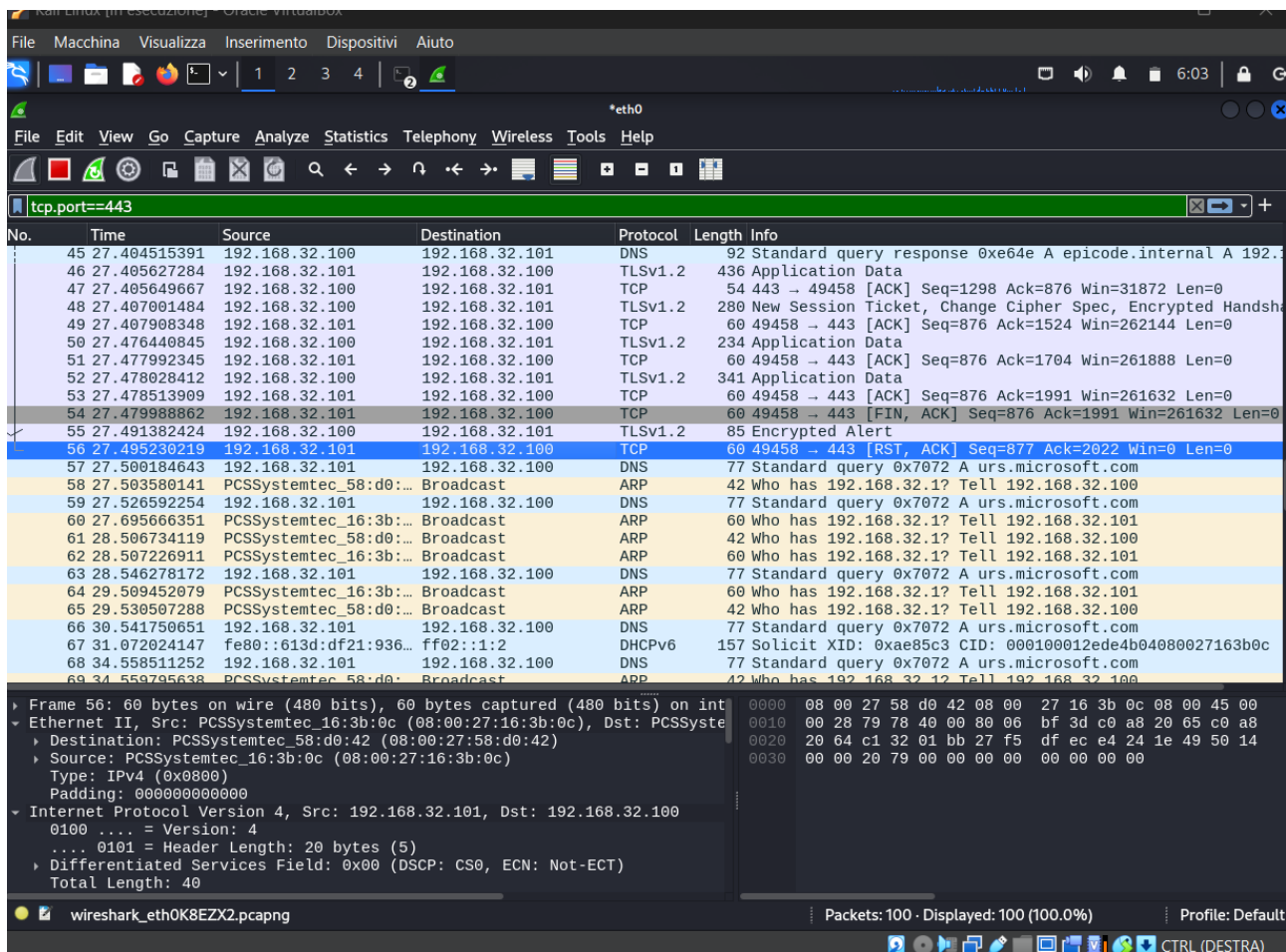


Figura 7

Intercettare, mediante l'utilizzo di Wireshark, il traffico HTTP tra Kali e Windows con l'URL `epicode.internal`.

Come per l'esercizio precedente dovrò catturare il traffico tra le 2 VM, tramite l'utilizzo di Wireshark, ma questa volta dovrò catturare il traffico HTTP.

Per fare ciò sulla VM Kali andrò ad aprire il file di configurazione del servizio `inetsim` ed andrò ad attivare il servizio `http`, disattivando il servizio `https` mediante l'apposizione di un `#` (Figura 8).

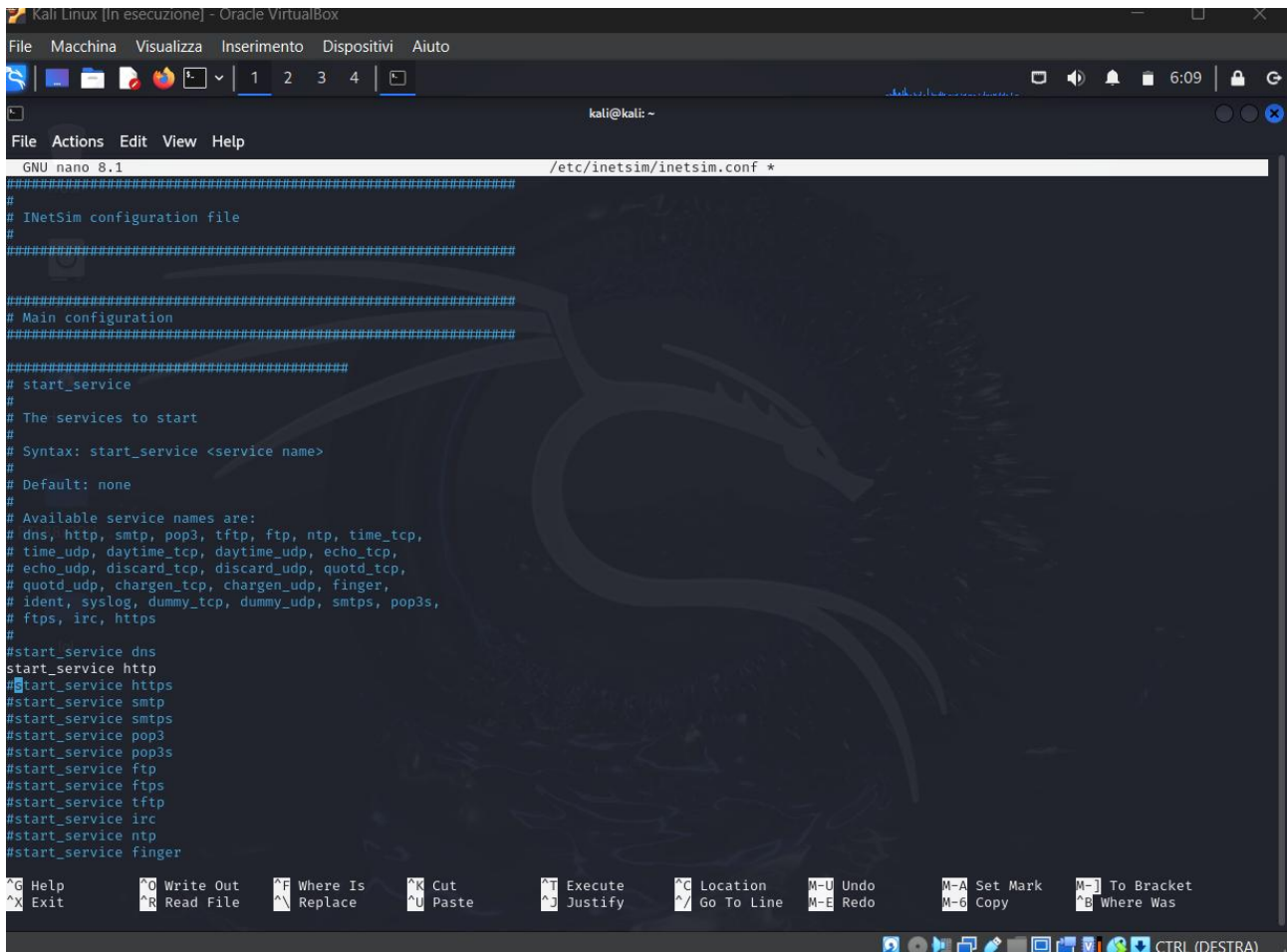


Figura 8

A questo punto avvio inetsim con il comando `sudo inetsim` ed attendo l'avvio del servizio. (Figura 9)

```
(kali@kali)-[~]
$ sudo inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 29490) ==
Session ID: 29490
Listening on: 192.168.32.100
Real Date/Time: 2024-12-02 06:11:30
Fake Date/Time: 2024-12-02 06:11:30 (Delta: 0 seconds)
Forking services...
* http_80_tcp - started (PID 29500)
done.
Simulation running
```

Figura 9

Adesso avvio nuovamente dnscchef con il comando precedente

`dnscchef --fakedomains epicode.internal --fakeip 192.168.32.100 -i 192.168.32.100`

ed attendo l'avvio del servizio. Una volta fatto ciò mi connetto sulla VM Windows e, sul browser Microsoft Edge, inserisco l'URL <http://epicode.internal> ottenendo la pagina http del server (Figura 10)

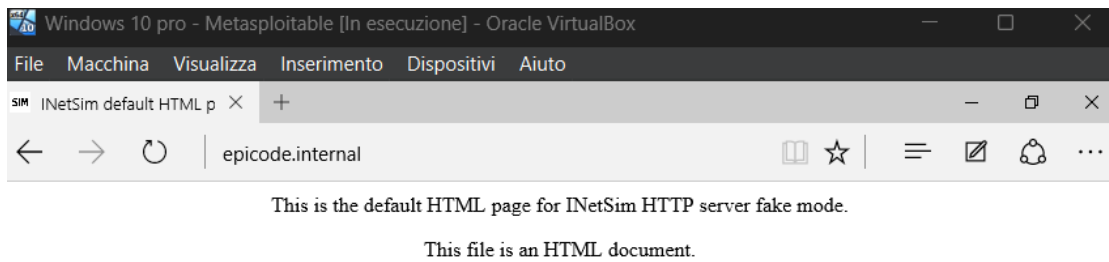


Figura 10

Apro Wireshark e catturo il traffico sull'interfaccia eth0 tra le 2 VM. Una volta catturati i pacchetti, utilizzo il filtro `tcp.port==80` per isolare il traffico HTTP (Figura 11).



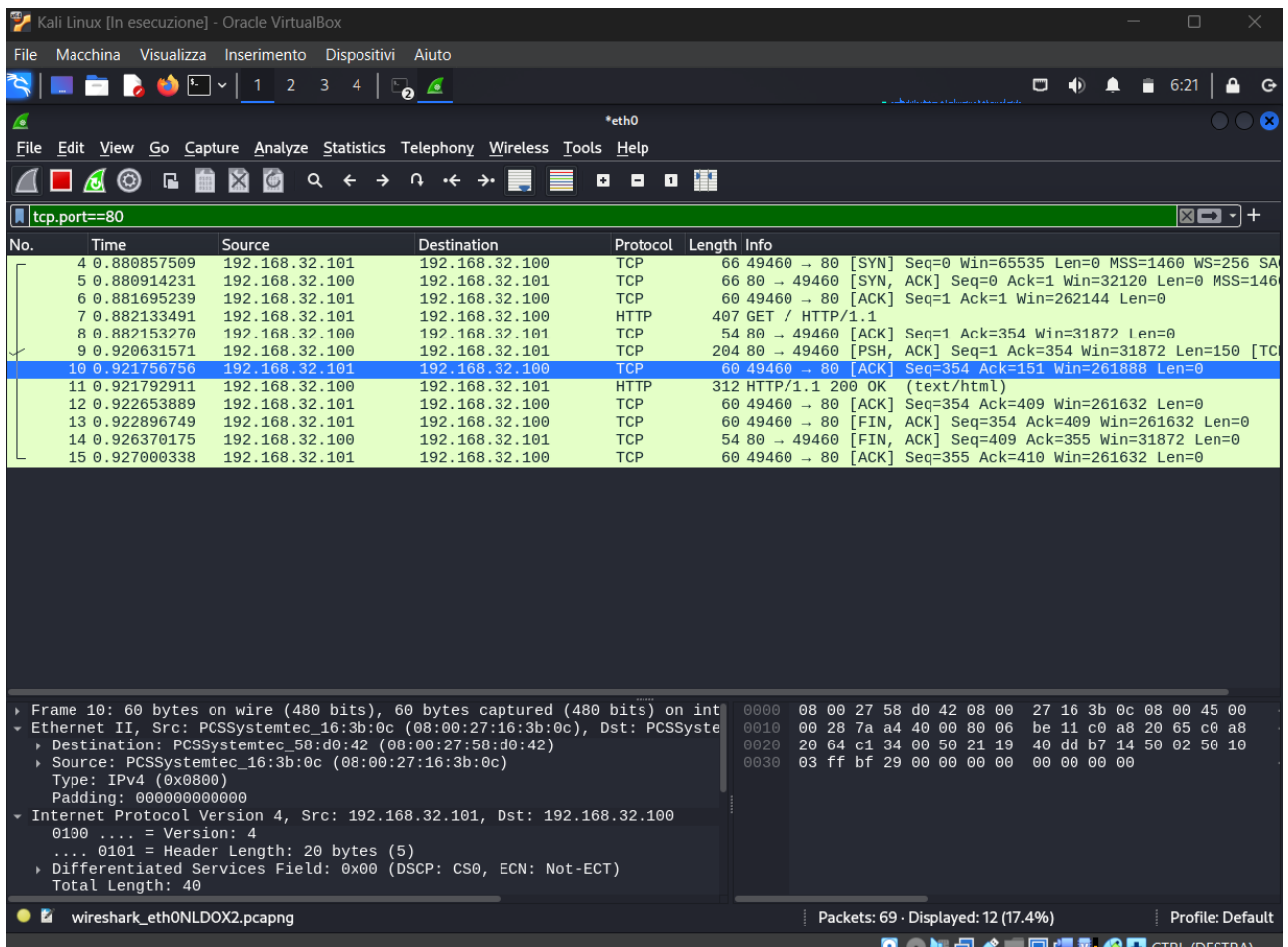


Figura 11

Anche in questo caso si riesce a determinare il MAC sorgente e di destinazione oltre agli indirizzi IP.

Le principali differenze rilevate innanzitutto il numero di porta in quanto http usa la porta 80 mentre il servizio HTTPS utilizza la porta 443. Altra differenza è come tutto sia in chiaro utilizzando il servizio http. Si riesce a rilevare la sequenza SYN, SYN ACK e ACK con le richieste GET (Figura 12 e 13).

Clicca sul tab del traffico GET si riesce a risalire alla richiesta della pagina web in chiaro.

Per quanto riguarda il servizio HTTPS anche qui è presente il 3 way handshake con le richieste SYN, SYN ACK e ACK ma abbiamo anche una richiesta TSL per la verifica del certificato di sicurezza per poter crittografare la connessione e renderla sicura (Figura 14).

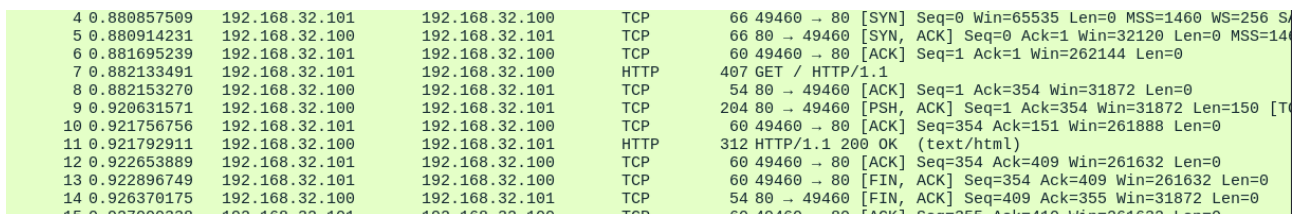


Figura 12



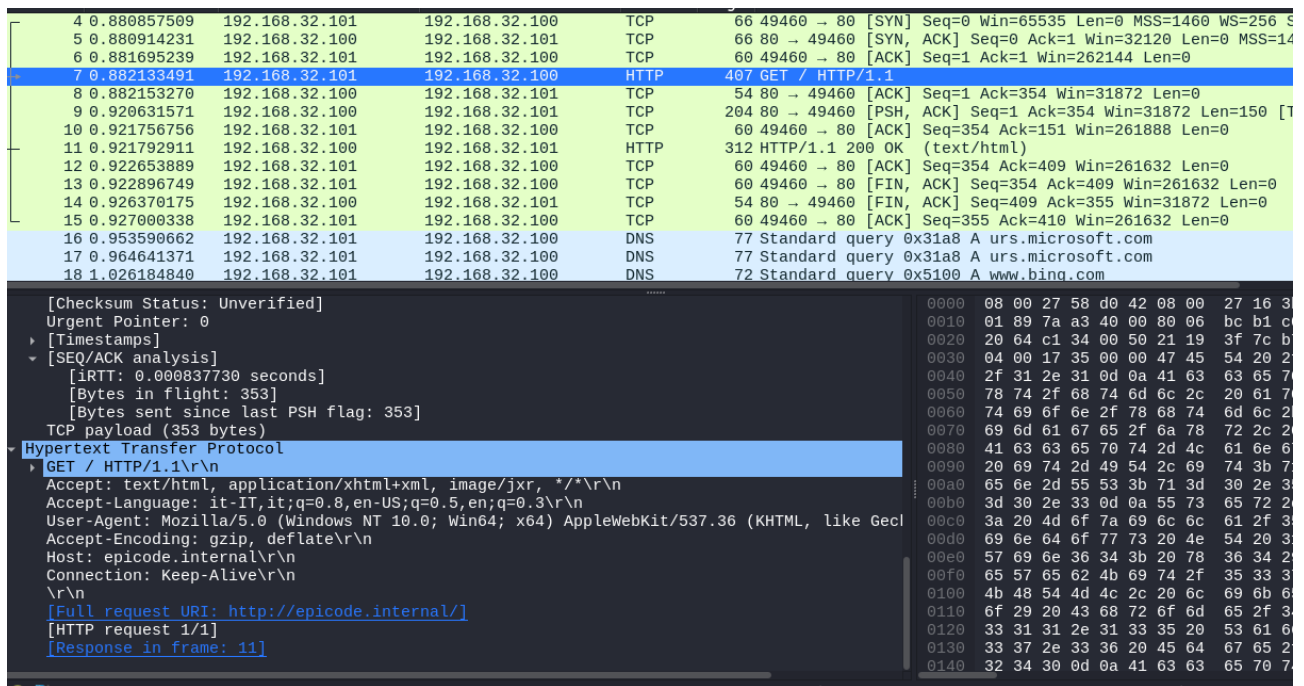


Figura 13

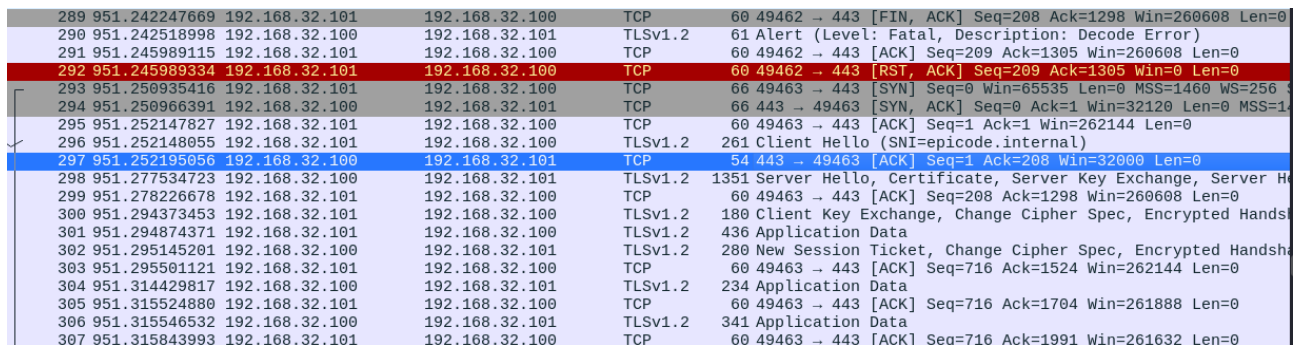


Figura 14

Progetto di Roberto Bella

2/12/2024

Progetto Finale M1 CSPT0524