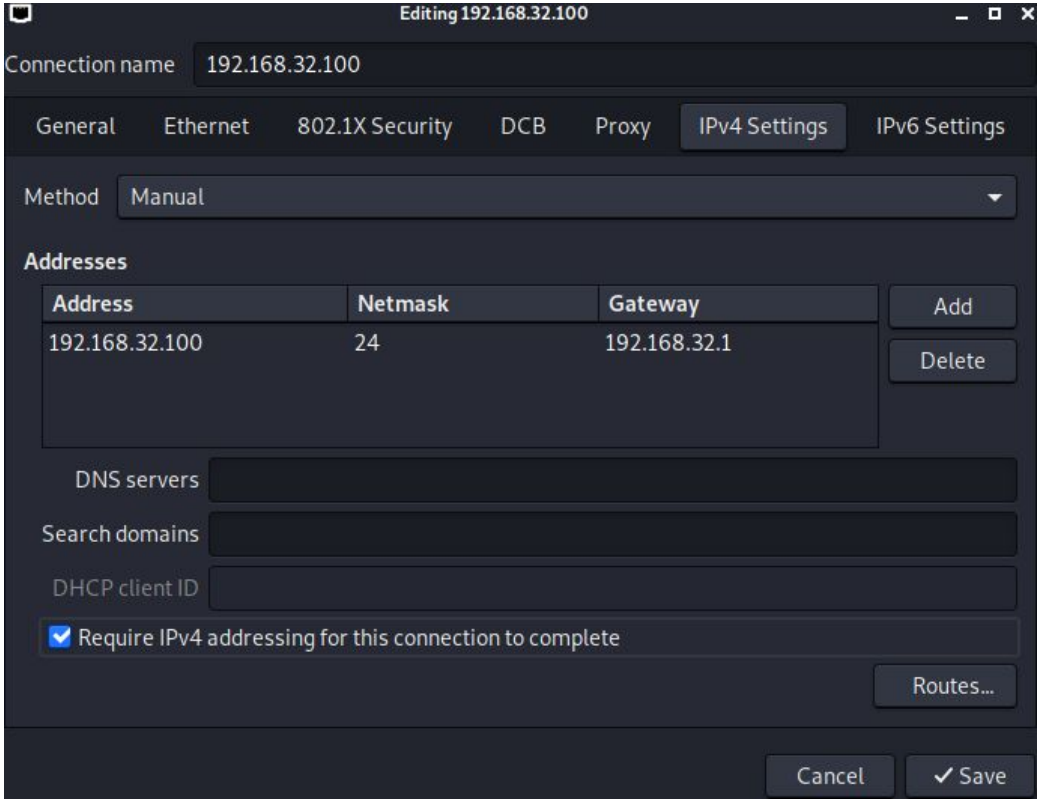
# W4D4 - Progetto

Simulazione protocolli DNS, HTTP e HTTPS su INetSim

# Impostazione indirizzo IP VM Kali Linux

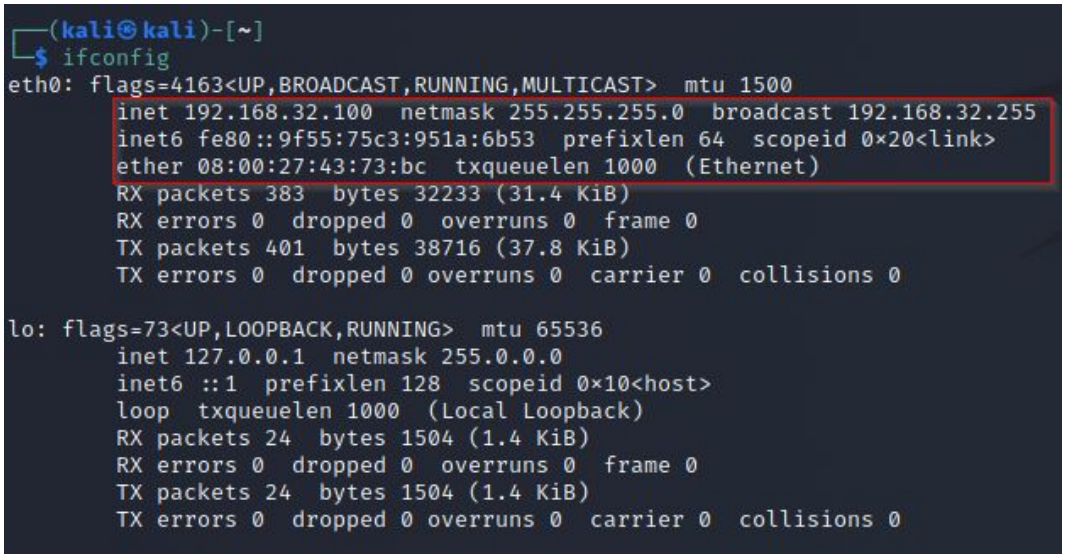



L'indirizzo IP della macchina Kali è stato impostato tramite il NetworkManager.

Indirizzo IPv4: 192.168.32.100/24

La macchina virtuale Kali Linux, grazie alla simulazione dei servizi con il tool INetSim, fungerà da server che risponderà alla richiesta di un client di accedere alla risorsa 'epicode.internal'.

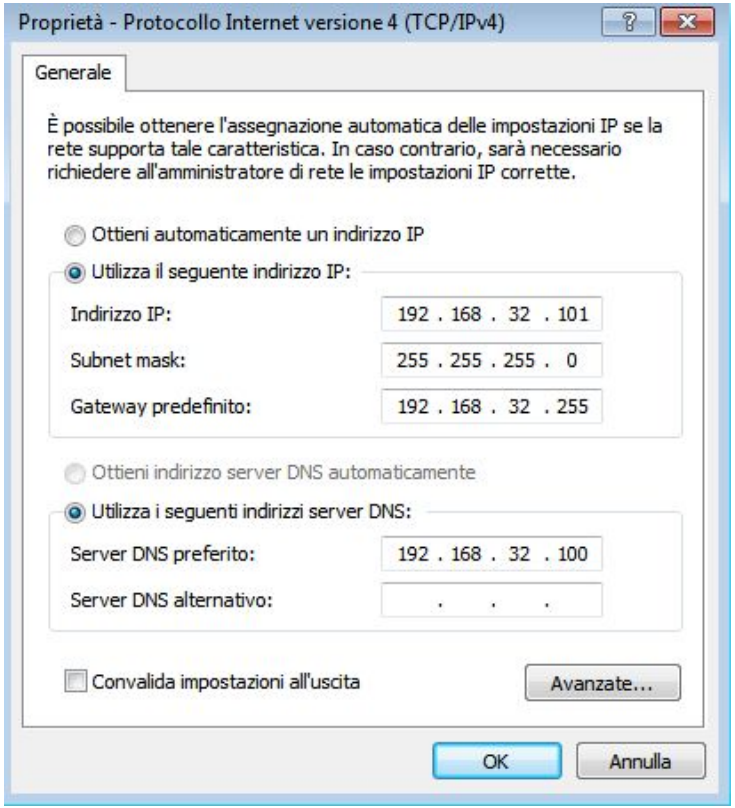
# Impostazione indirizzo IP VM Kali Linux




Verifica corretta configurazione di rete con comando ‘ifconfig’ su macchina Kali Linux. Questo comando permette di visualizzare sia lo stato della rete che la sua configurazione, evidenziando indirizzo IP, subnet mask, indirizzo di broadcast e MAC address della macchina virtuale.

Il MAC address di Kali Linux, come da screenshot al lato è: **08:00:27:43:73:bc**

# Impostazione indirizzo IP VM Windows 7

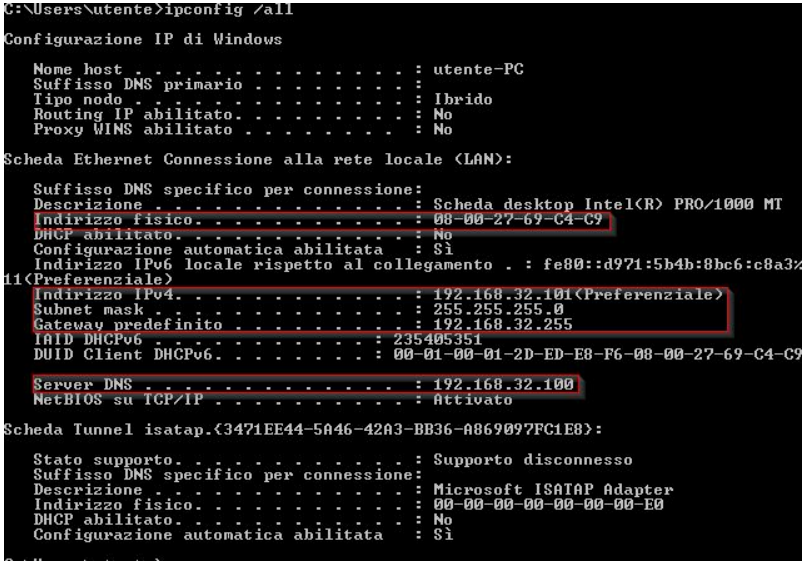



```
C:\Users\utente>ipconfig

Configurazione IP di Windows


Scheda Ethernet Connessione alla rete locale (LAN):

   Suffisso DNS specifico per connessione:
   Indirizzo IPv6 locale rispetto al collegamento . : fe80::d971:5b4b:8bc6:c8a3%
11
   Indirizzo IPv4. . . . . . . . . . . . : 192.168.32.101
   Subnet mask . . . . . . . . . . . . . : 255.255.255.0
   Gateway predefinito . . . . . . . . . : 192.168.32.255

Scheda Tunnel isatap.{3471EE44-5A46-42A3-BB36-A869097FC1E8}:

   Stato supporto. . . . . . . . . . . . : Supporto disconnesso
   Suffisso DNS specifico per connessione:
```

Indirizzo IPv4: 192.168.32.101/24

Server DNS: Indirizzo IPv4 Kali Linux 192.168.32.100

# Impostazione indirizzo IP VM Windows 7



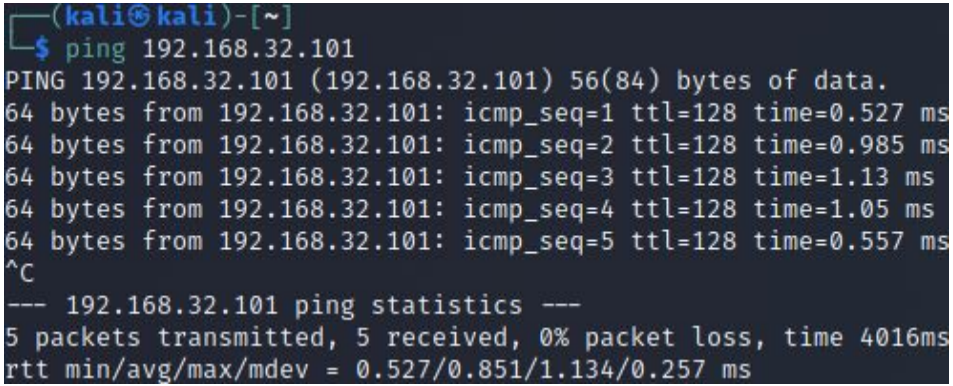

Per verificare la corretta configurazione di rete della macchina virtuale (client) Windows 7, è utile il comando ‘ipconfig /all’, che mostrerà sia l’indirizzo fisico MAC associato alla scheda di rete virtuale della macchina Windows, che l’indirizzo IPv4, la subnet mask e il DNS.

Nel caso della macchina Windows 7, il MAC address corrisponde a: **08-00-27-69-C4-C9**.

Questo tornerà utile nel successivo tracciamento dei pacchetti e comunicazioni tra client e server con WireShark.

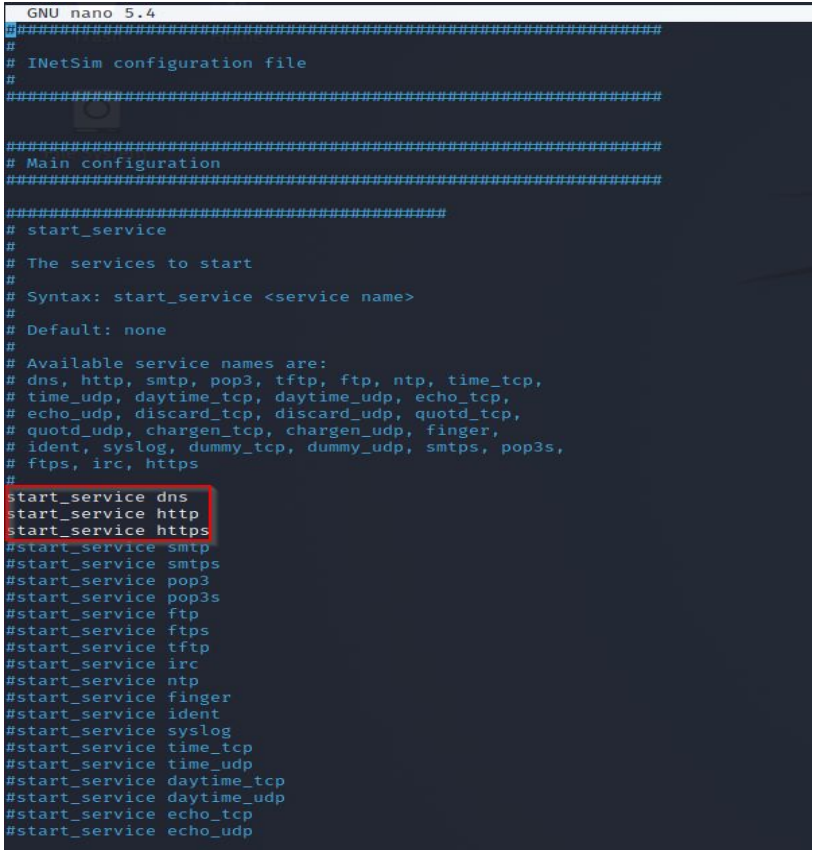
# Verifica comunicazione tra le due macchine


Ping da Client Windows 7 - Indirizzo IPv4: 192.168.32.101/24 al Server Kali Linux - Indirizzo IPv4: 192.168.32.100/24




Ping da Server Kali Linux - Indirizzo IPv4: 192.168.32.100 a Client Windows 7 - Indirizzo IPv4: 192.168.32.101/24

# Configurazione tool INetSim - attivazione servizi




Il file di configurazione del tool INetSim è salvato nella directory /ect/inetsim/inetsim.conf. Da terminale, si possono apportare modifiche al file di configurazione con privilegi amministratore accedendo al file tramite editor di testo nano.

Nello screenshot al lato si può osservare che i servizi utili allo svolgimento della traccia sono attivi.

# Configurazione tool INetSim - ascolto su porte Kali

```
##########################################
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
#service_bind_address 10.10.10.1
service_bind_address 192.168.32.100
```

```
##########################################
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
#service_bind_address 10.10.10.1
service_bind_address 0.0.0.0
```

INetSim è stato impostato per ascoltare sulle porte della macchina Kali.

In alternativa, si può lanciare il tool INetSim da terminale con il comando --bind-address=<IP address>, impostando sempre il tool per ascoltare sull'indirizzo IPv4 di Kali.

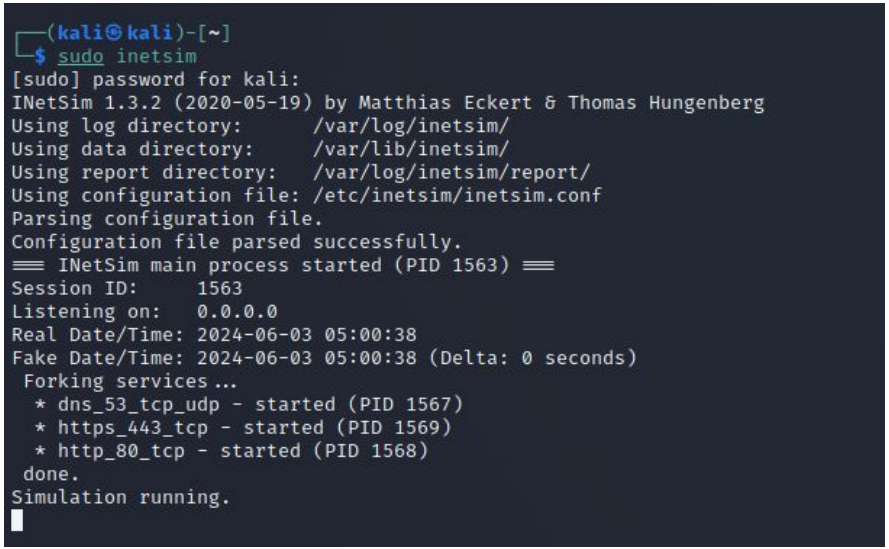
Se impostato su indirizzo IPv4 0.0.0.0, il tool ascolterà su tutti gli indirizzi IPv4 dei client che possono stabilire una connessione con il server.

# Configurazione tool INetSim - protocollo DNS

```
##########################################
# 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
```

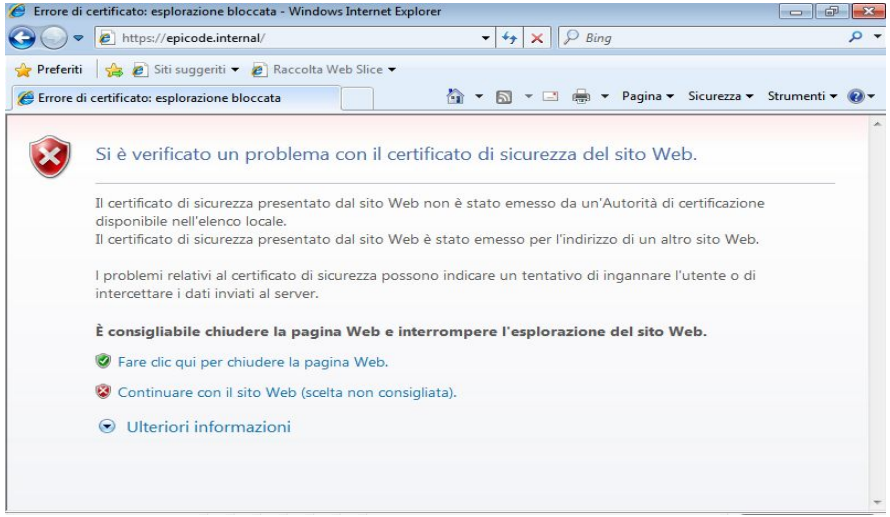
Il protocollo DNS risolverà la richiesta di una risorsa all'hostname 'epicode.internal' da parte del client in un indirizzo IPv4 - in questo caso quello della macchina virtuale Kali (server).

# Lancio tool INetSim




Per simulare i servizi di rete è innanzitutto necessario lanciare il tool INetSim tramite comando ‘sudo inetsim’.

# Richiesta risorsa web da client - HTTPS

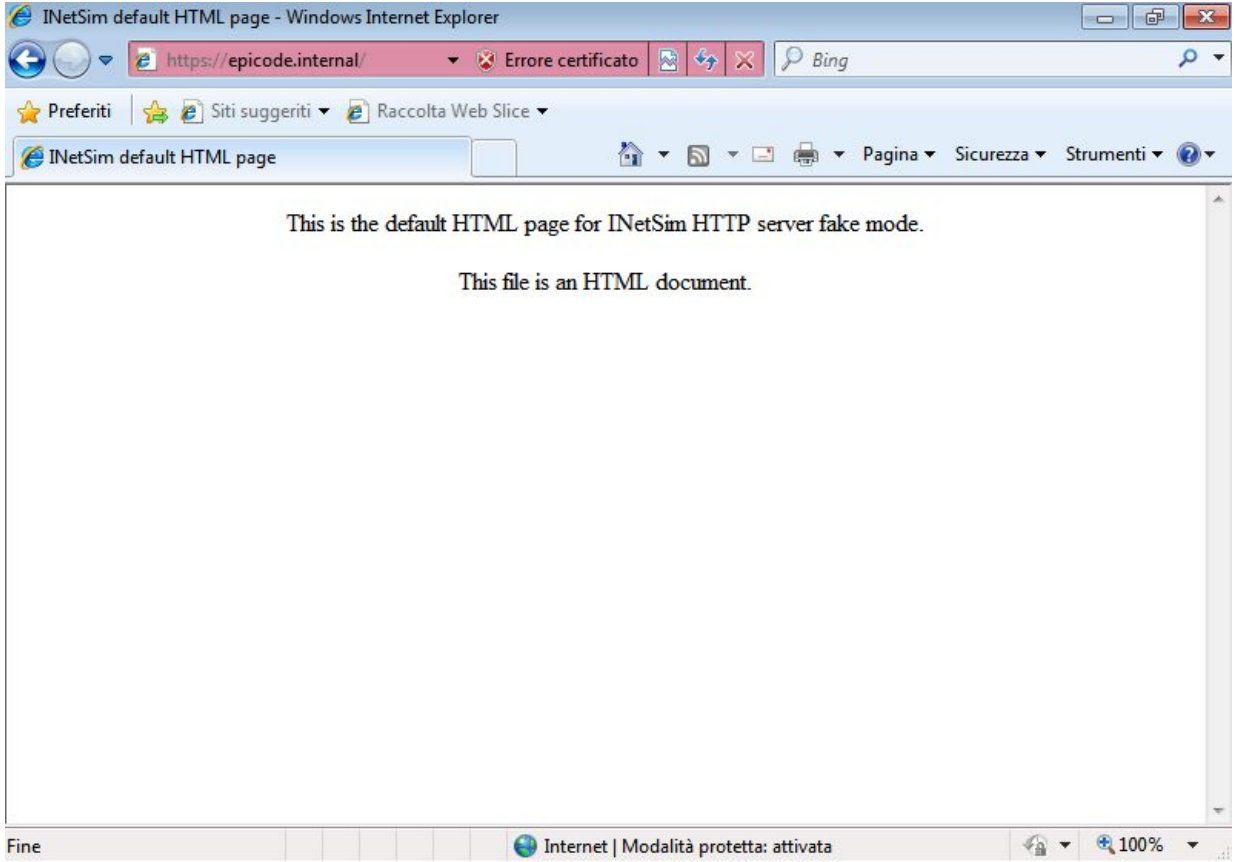



Una volta lanciato INetSim, si può procedere a richiedere una risorsa web da client Windows 7 tramite servizio HTTPS. Da web browser, digitando ‘https://epicode.internal/, Windows Explorer ci avvisa di un problema con il certificato sicurezza del sito web richiesto, che non risulta emesso da un’Autorità di certificazione riconosciuta per quella particolare risorsa web.

Facendo click sulla voce ‘Sicurezza’ > ‘Rapporto sulla sicurezza’, si potranno visualizzare i dettagli del certificato.

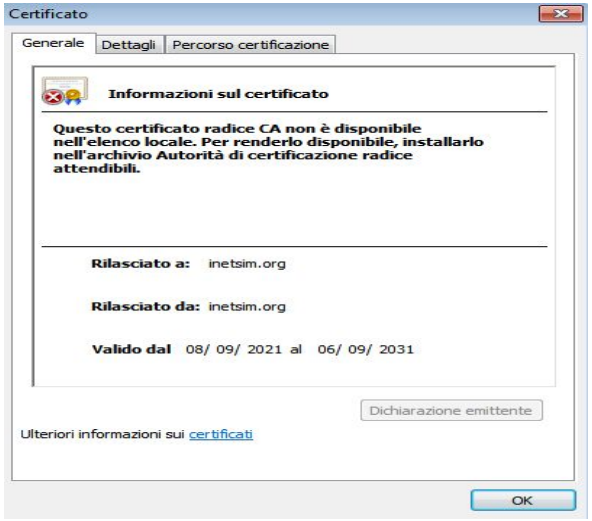
Facendo clic su ‘Continuare con il sito Web’ e ignorando l’avviso di sicurezza di Windows, si potrà invece accedere alla risorsa.

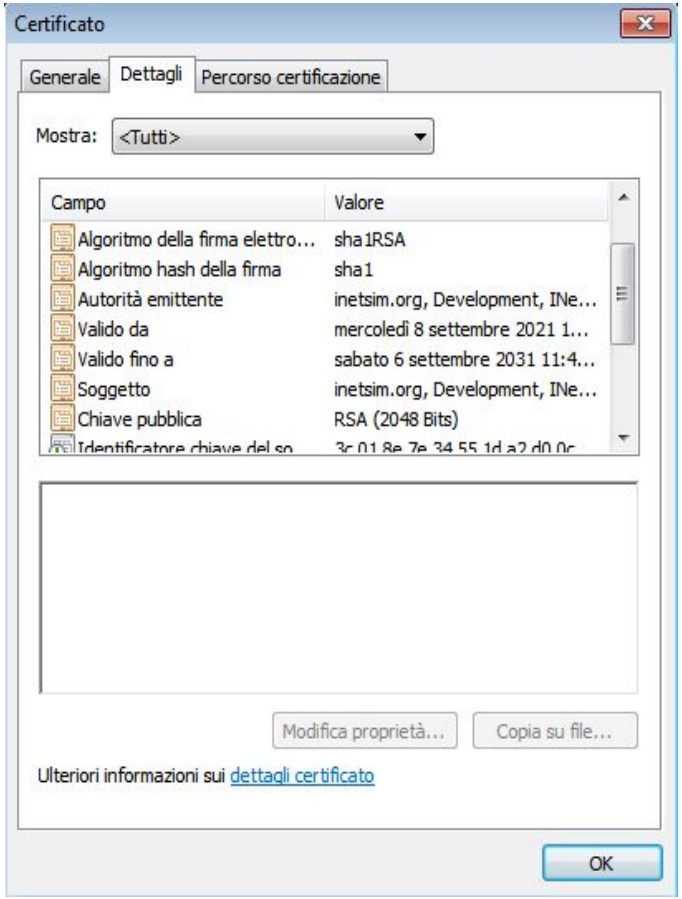
# Intercettazione comunicazione con WireShark - HTTPS




Una volta cliccato su ‘Continuare con il sito Web’, è possibile ignorare l’alert e accedere alla risorsa.

# Richiesta risorsa web da client - HTTPS



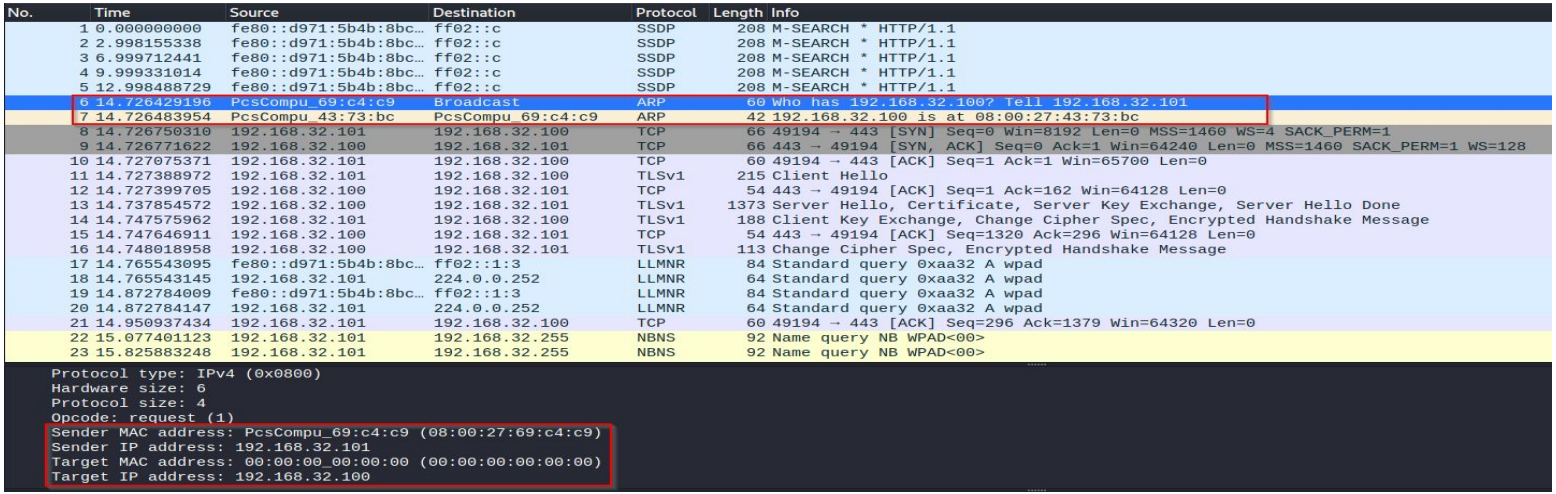

I dettagli del certificato per 'https://epicode.internal/ sono come in figura al lato. Il certificato è stato rilasciato a inetsim.org da inetsim.org.

Alla voce 'Dettagli' è possibile visualizzare la versione del certificato, il numero di serie, nonché l'algoritmo della firma elettronica e l'algoritmo hash della firma, oltre alla chiave pubblica del certificato digitale.

Dunque l'alert di Windows 7 è dovuto probabilmente al fatto che il tool INetSim include certificati SSL generici che vengono utilizzati per simulare i servizi e il certificato per la particolare risorsa web accessibile sull'hostname epicode.internal non è stato generato per quel dominio e non è stato generato da un'Autorità riconosciuta per il rilascio di certificati digitali (come ad esempio Amazon.com).

Ciò significa che non è possibile verificare l'autenticità del sito web epicode.internal.

# Intercettazione comunicazione con WireShark - HTTPS

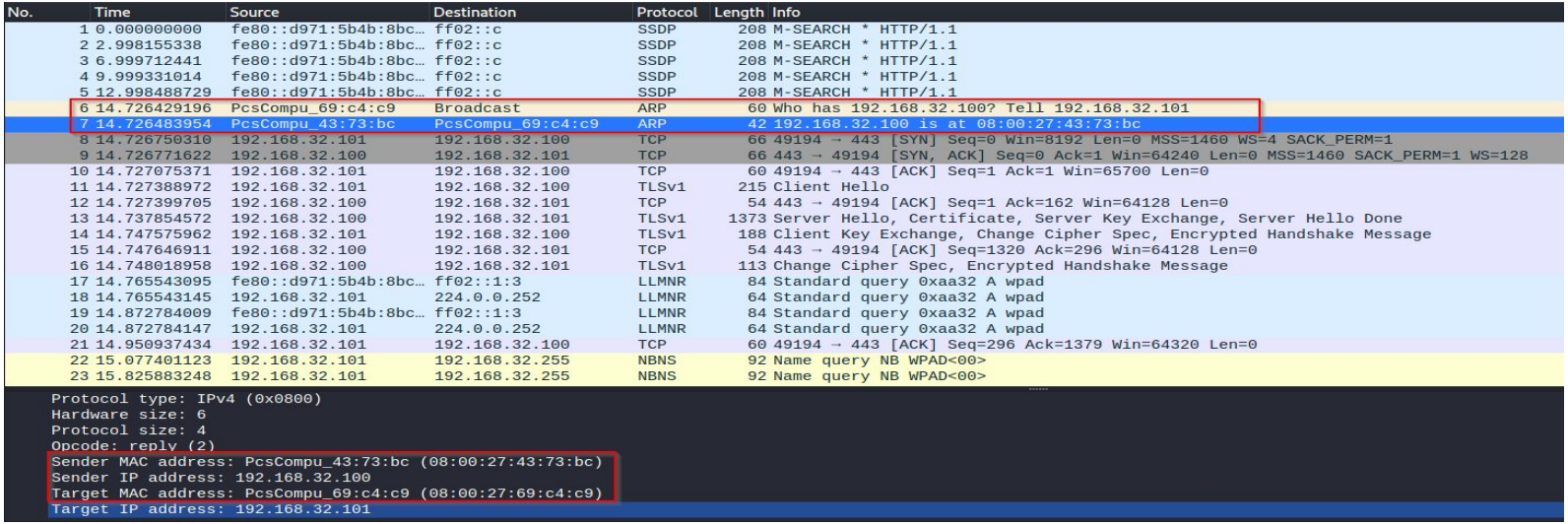


| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000000 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 2 | 2.998155338 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 3 | 6.999712441 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 4 | 9.999331014 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 5 | 12.998488729 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 6 | 14.726429196 | PcsCompu_69:c4:c9 | Broadcast | ARP | 60 | Who has 192.168.32.100? Tell 192.168.32.101 |
| 7 | 14.726483954 | PcsCompu_43:73:bc | PcsCompu_69:c4:c9 | ARP | 42 | 192.168.32.100 is at 08:00:27:43:73:bc |
| 8 | 14.726750310 | 192.168.32.101 | 192.168.32.100 | TCP | 66 | 49194 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 |
| 9 | 14.726771622 | 192.168.32.100 | 192.168.32.101 | TCP | 66 | 443 → 49194 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128 |
| 10 | 14.727075371 | 192.168.32.101 | 192.168.32.100 | TCP | 60 | 49194 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0 |
| 11 | 14.727388972 | 192.168.32.101 | 192.168.32.100 | TLSv1 | 215 | Client Hello |
| 12 | 14.727399705 | 192.168.32.100 | 192.168.32.101 | TCP | 54 | 443 → 49194 [ACK] Seq=1 Ack=162 Win=64128 Len=0 |
| 13 | 14.737854572 | 192.168.32.100 | 192.168.32.101 | TLSv1 | 1373 | Server Hello, Certificate, Server Key Exchange, Server Hello Done |
| 14 | 14.747575962 | 192.168.32.101 | 192.168.32.100 | TLSv1 | 188 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 15 | 14.747646911 | 192.168.32.100 | 192.168.32.101 | TCP | 54 | 443 → 49194 [ACK] Seq=1320 Ack=296 Win=64128 Len=0 |
| 16 | 14.748018958 | 192.168.32.100 | 192.168.32.101 | TLSv1 | 113 | Change Cipher Spec, Encrypted Handshake Message |
| 17 | 14.765543095 | fe80::d971:5b4b:8bc… | ff02::1:3 | LLMNR | 84 | Standard query 0xaa32 A wpad |
| 18 | 14.765543145 | 192.168.32.101 | 224.0.0.252 | LLMNR | 64 | Standard query 0xaa32 A wpad |
| 19 | 14.872784009 | fe80::d971:5b4b:8bc… | ff02::1:3 | LLMNR | 84 | Standard query 0xaa32 A wpad |
| 20 | 14.872784147 | 192.168.32.101 | 224.0.0.252 | LLMNR | 64 | Standard query 0xaa32 A wpad |
| 21 | 14.950937434 | 192.168.32.101 | 192.168.32.100 | TCP | 60 | 49194 → 443 [ACK] Seq=296 Ack=1379 Win=64320 Len=0 |
| 22 | 15.077401123 | 192.168.32.101 | 192.168.32.255 | NBNS | 92 | Name query NB WPAD<00> |
| 23 | 15.825883248 | 192.168.32.101 | 192.168.32.255 | NBNS | 92 | Name query NB WPAD<00> |

```
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: PcsCompu_69:c4:c9 (08:00:27:69:c4:c9)
Sender IP address: 192.168.32.101
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.32.100
```

Il client e il server sono sulla stessa rete. Il client deve richiedere una risorsa web al server 192.168.32.100, ma ne conosce solo l'indirizzo IP e non il MAC address. Dunque, per associare l'indirizzo IP al MAC invia una richiesta ARP (Address Resolution Protocol) a tutti i dispositivi sullo stesso dominio di broadcast, chiedendo: 'Chi ha l'indirizzo IP 192.168.32.100?'. Come si può vedere dal dettaglio mostrato in basso a sinistra, il MAC address del client corrisponde a quello della macchina Windows 7.

# Intercettazione comunicazione con WireShark - HTTPS

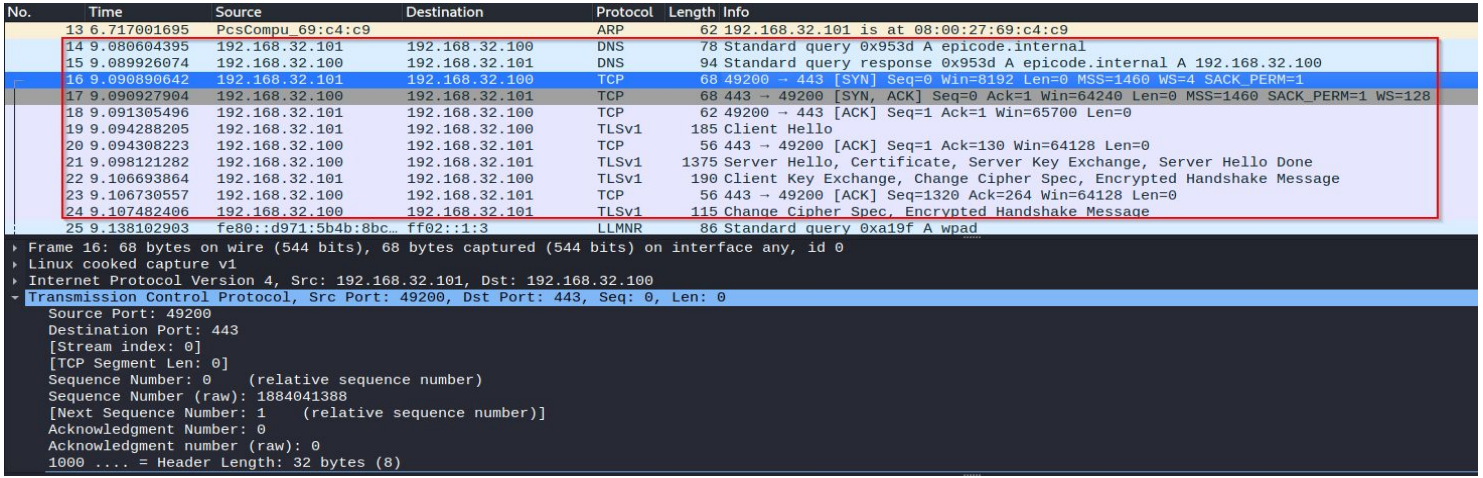


| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000000 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 2 | 2.998155338 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 3 | 6.999712441 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 4 | 9.999331014 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 5 | 12.998488729 | fe80::d971:5b4b:8bc… | ff02::c | SSDP | 208 | M-SEARCH * HTTP/1.1 |
| 6 | 14.726429196 | PcsCompu_69:c4:c9 | Broadcast | ARP | 60 | Who has 192.168.32.100? Tell 192.168.32.101 |
| 7 | 14.726483954 | PcsCompu_43:73:bc | PcsCompu_69:c4:c9 | ARP | 42 | 192.168.32.100 is at 08:00:27:43:73:bc |
| 8 | 14.726750310 | 192.168.32.101 | 192.168.32.100 | TCP | 66 | 49194 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1 |
| 9 | 14.726771622 | 192.168.32.100 | 192.168.32.101 | TCP | 66 | 443 → 49194 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128 |
| 10 | 14.727075371 | 192.168.32.101 | 192.168.32.100 | TCP | 60 | 49194 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0 |
| 11 | 14.727388972 | 192.168.32.101 | 192.168.32.100 | TLSv1 | 215 | Client Hello |
| 12 | 14.727399705 | 192.168.32.100 | 192.168.32.101 | TCP | 54 | 443 → 49194 [ACK] Seq=1 Ack=162 Win=64128 Len=0 |
| 13 | 14.737854572 | 192.168.32.100 | 192.168.32.101 | TLSv1 | 1373 | Server Hello, Certificate, Server Key Exchange, Server Hello Done |
| 14 | 14.747575962 | 192.168.32.101 | 192.168.32.100 | TLSv1 | 188 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 15 | 14.747646911 | 192.168.32.100 | 192.168.32.101 | TCP | 54 | 443 → 49194 [ACK] Seq=1320 Ack=296 Win=64128 Len=0 |
| 16 | 14.748018958 | 192.168.32.100 | 192.168.32.101 | TLSv1 | 113 | Change Cipher Spec, Encrypted Handshake Message |
| 17 | 14.765543095 | fe80::d971:5b4b:8bc… | ff02::1:3 | LLMNR | 84 | Standard query 0xaa32 A wpad |
| 18 | 14.765543145 | 192.168.32.101 | 224.0.0.252 | LLMNR | 64 | Standard query 0xaa32 A wpad |
| 19 | 14.872784009 | fe80::d971:5b4b:8bc… | ff02::1:3 | LLMNR | 84 | Standard query 0xaa32 A wpad |
| 20 | 14.872784147 | 192.168.32.101 | 224.0.0.252 | LLMNR | 64 | Standard query 0xaa32 A wpad |
| 21 | 14.950937434 | 192.168.32.101 | 192.168.32.100 | TCP | 60 | 49194 → 443 [ACK] Seq=296 Ack=1379 Win=64320 Len=0 |
| 22 | 15.077401123 | 192.168.32.101 | 192.168.32.255 | NBNS | 92 | Name query NB WPAD<00> |
| 23 | 15.825883248 | 192.168.32.101 | 192.168.32.255 | NBNS | 92 | Name query NB WPAD<00> |

```
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: PcsCompu_43:73:bc (08:00:27:43:73:bc)
Sender IP address: 192.168.32.100
Target MAC address: PcsCompu_69:c4:c9 (08:00:27:69:c4:c9)
Target IP address: 192.168.32.101
```

Il server Kali, che ha l'indirizzo IP incluso nella richiesta ARP, risponderà inviando il proprio MAC address al MAC address della macchina Windows 7.

# Intercettazione comunicazione con WireShark - HTTPS

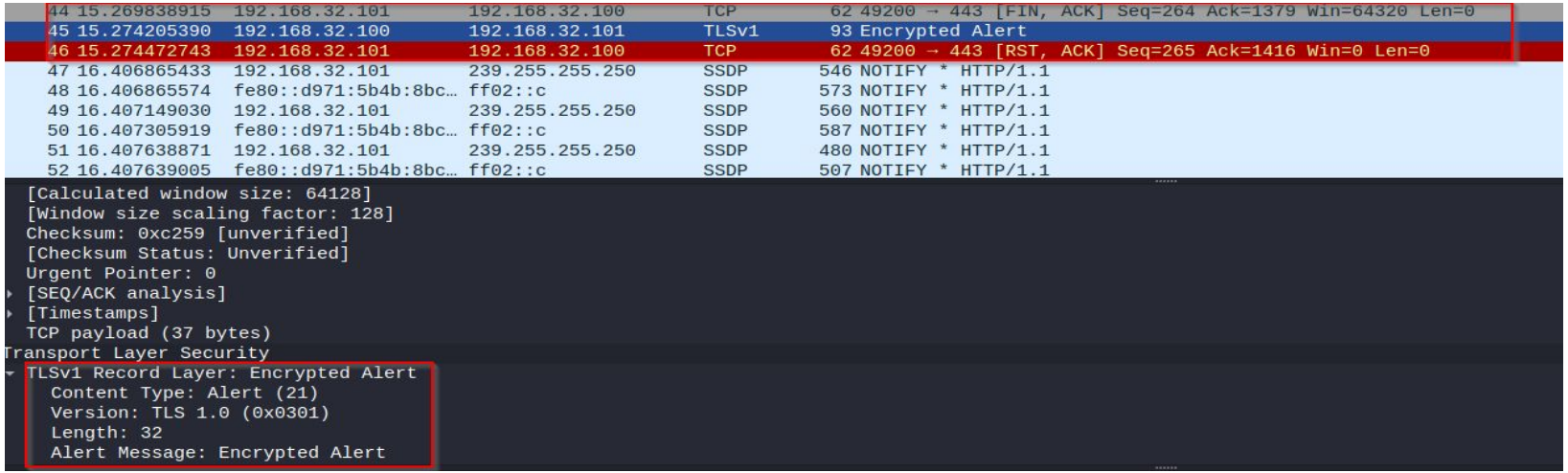



WireShark permette di avere una visuale dettagliata del ‘Three-way handshake’ tipico del protocollo TCP, che inizia a scambiare pacchetti di dati solo dopo aver stabilito una connessione tra hosts o client/server. Come da screenshot, la comunicazione viene iniziata dal client, che invia al server un pacchetto con la flag ‘SYN’ attiva e un numero di sequenza casuale. Il server risponde, inviando al client un pacchetto con i flag SYN e ACK abilitati, e un altro numero di sequenza casuale. L’ACK sarà uguale al precedente numero Seq +1. Il client completa la sincronizzazione inviando un pacchetto ACK e inviando i numeri Seq, e ACK come fatto dal server. La richiesta HTTPS avviene sulla porta 443, che è quella riservata a questo protocollo.

Una volta finito il processo di stretta di mano in tre fasi, inizia la comunicazione con il TLSv1 (Transport Layer Security version 1), protocollo di trasporto che permette una comunicazione sicura end-to-end e fornisce autenticazione, integrità dei dati e confidenzialità.

In questo frangente avviene lo scambio di chiavi tra client e server con cifratura asimmetrica. Nel protocollo HTTPS, un client può accedere alla chiave pubblica tramite il certificato digitale e possiede una chiave privata univoca. I messaggi nel protocollo HTTPS sono criptati, come è possibile verificare alla riga 22 e 24.

# Intercettazione comunicazione con WireShark - HTTPS



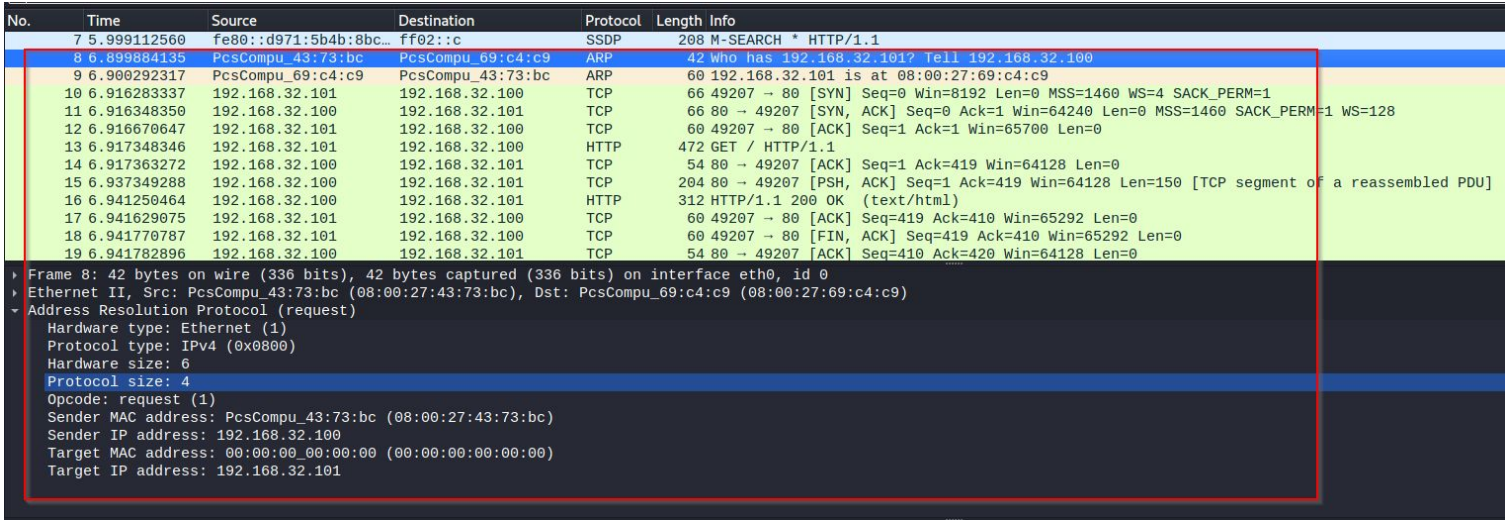

Come è possibile vedere alla riga 45, WireShark tiene traccia anche dell'alert ricevuto riguardo il certificato di 'epicode.internal'. L'alert 21 è un alert del protocollo TLS e, come si può vedere, anche questo è criptato.

La flag FIN alla riga 44 indica la fine della connessione tra client e server.

La flag RST denota invece un 'abortion' della connessione, solitamente viene inviata quando uno dei due peer (hosts o client/server) ha ragione di credere che la connessione non dovrebbe esistere.

Ad esempio, in caso di attacchi, errori, o eventi sospetti.
In questo caso, può essere dovuto all'errore nel certificato di 'epicode.internal'.

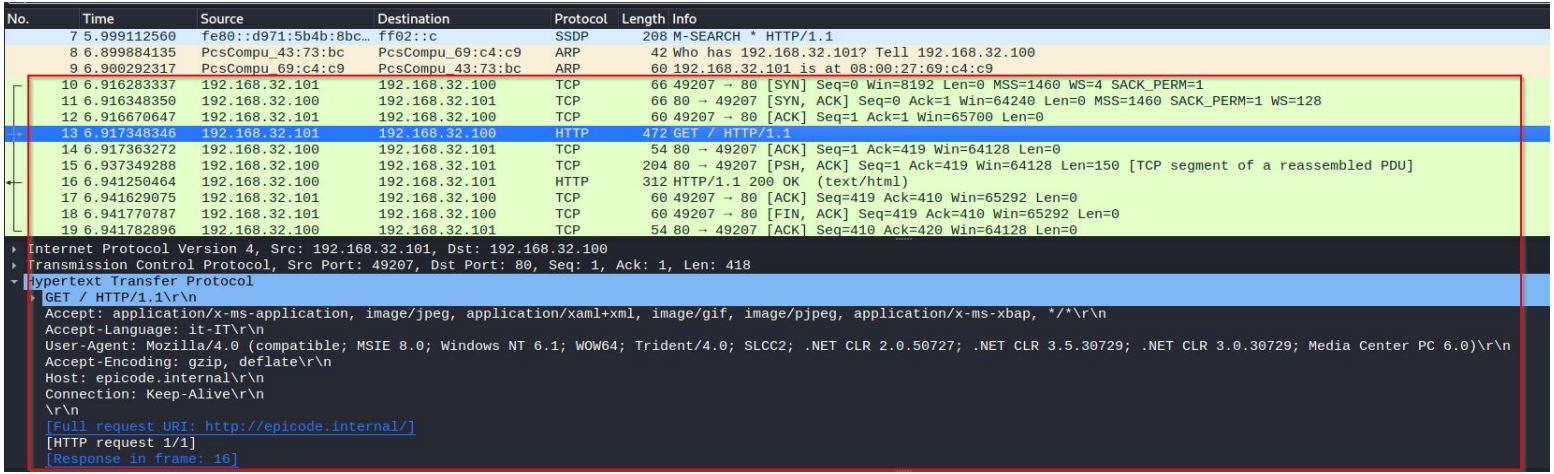
# Intercettazione comunicazione con WireShark - HTTP




Effettuando una richiesta http per ‘epicode.internal’ da web browser di Windows 7, si può osservare quali siano le differenze tra HTTP e HTTPS e come avviene la comunicazione.

Inannzitutto, anche nel caso del servizio HTTP viene lanciata una richiesta ARP dal client per conoscere il MAC address dell’indirizzo IP del server.

# Intercettazione comunicazione con WireShark - HTTP



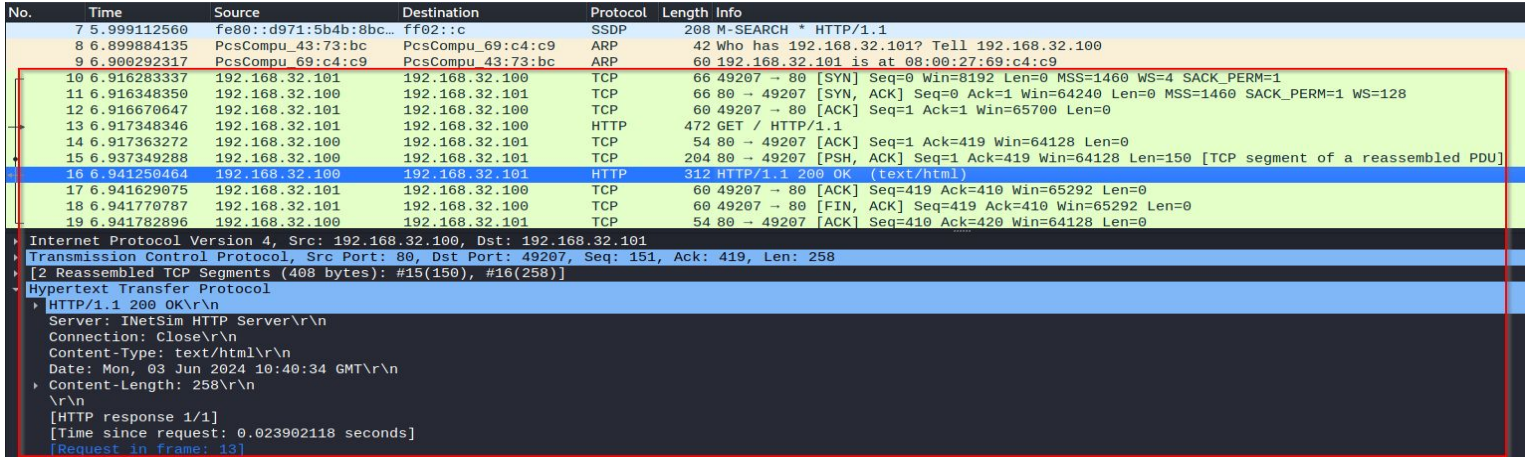

Come si può vedere alla riga 10, la richiesta HTTP avviene sulla porta 80, che è quella riservata a questo protocollo. Viene stabilita una connessione tra client e server con il three-way-handshake. Oltre alle 'Synchronization' e 'Acknowledgement' flag, alla riga 15 si osserva la flag 'PSH', che comunica al server di inviare l'intero contenuto immediatamente.

Alla riga 13, si può vedere una richiesta con metodo 'GET', che è quello utilizzato quando si richiede una risorsa web. Il campo host specifica la risorsa richiesta, in questo caso epicode.internal. Il campo user-agent dà indicazione al server del programma lato client dal quale sta partendo la richiesta.

Il campo Accept indica che tipo di documento ci si aspetta come risposta. Il parametro 'Keep-Alive' alla voce Connection farà in modo che la connessione potrà essere riutilizzata nelle successive interazioni client-server.

Come si può osservare, le richieste HTTP a differenza di quelle HTTPS non sono criptate.

# Intercettazione comunicazione con WireShark - HTTP




Quando un server riceve la chiamata, la elabora e poi spedisce la risposta al client.

Alla riga 16 si leggono i dettagli della risposta del server. La riga di stato ‘200 OK’ indica che la risorsa è stata trovata e i dettagli della risposta e della risorsa trovata sono leggibili nel pannello in basso. Ad esempio, la data in cui è stato generato il messaggio, le informazioni sul server che ha generato la richiesta (in questo caso INetSim HTTP Server) e il formato del contenuto della risorsa, ossia html.

I caratteri \r (ritorno a capo) e \n (nuova linea) sono utilizzati per terminare le righe HTTP.

# Conclusioni

Le differenze principali tra il traffico catturato in HTTP e quello catturato in HTTPS sono le seguenti:

1. La comunicazione nelle richieste HTTPS è criptata. In HTTP, client e server si scambiano dati in chiaro.
2. Nelle risposte HTTP, il server espone anche informazioni sul tipo di versione utilizzata (disclosure, può essere sfruttata per attacchi).
3. Nelle richieste HTTP non c'è l'extra layer del certificato di sicurezza SSL/TLS che consente la verifica dell'autenticità di un sito web.