

Progetto Finale: W4D4

Indice:

1) CONFIGURAZIONE LABORATORIO VIRTUALE

- 1.1) *Configurazione scheda di rete Kali.*
- 1.2) *Configurazione richiesta Https con Apache:
Installazione e aggiornamento Apache;
Abilitazione modulo SSL;
Creazione certificato autofirmato.*
- 1.3) *Configurazione Apache:
Abilitazione sito ssl;
Riavvio di Apache.*
- 1.4) *Servizio DNS:
Configurazione server DNS Kali.*
- 1.5) *Configurazione Windows e impostazione servizio DNS.*

2) INTERCETTAZIONE TRAFFICO CON SERVIZIO HTTPS

- 2.1 *Intercettazione MAC Address;*
- 2.2 *Intercettazione del traffico Https;*

3) SOSTITUZIONE SERVIZIO HTTPS con HTTP E INTERCETTAZIONE TRAFFICO

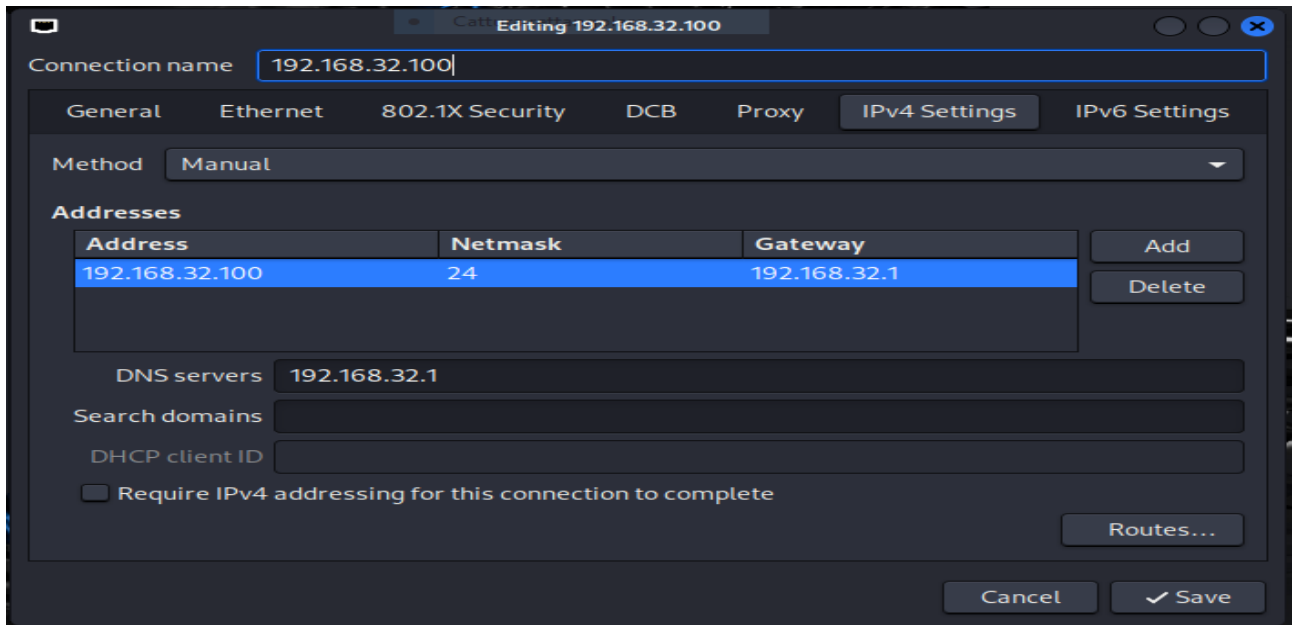
4) CONCLUSIONE: Differenze tra i servizi e considerazioni in merito.

SVOLGIMENTO.

1) CONFIGURAZIONE DEL LABORATORIO VIRTUALE

1.1) Configurazione scheda di rete Kali Linux:

Inizio con la configurazione della scheda di rete, come mostrato nell'immagine.

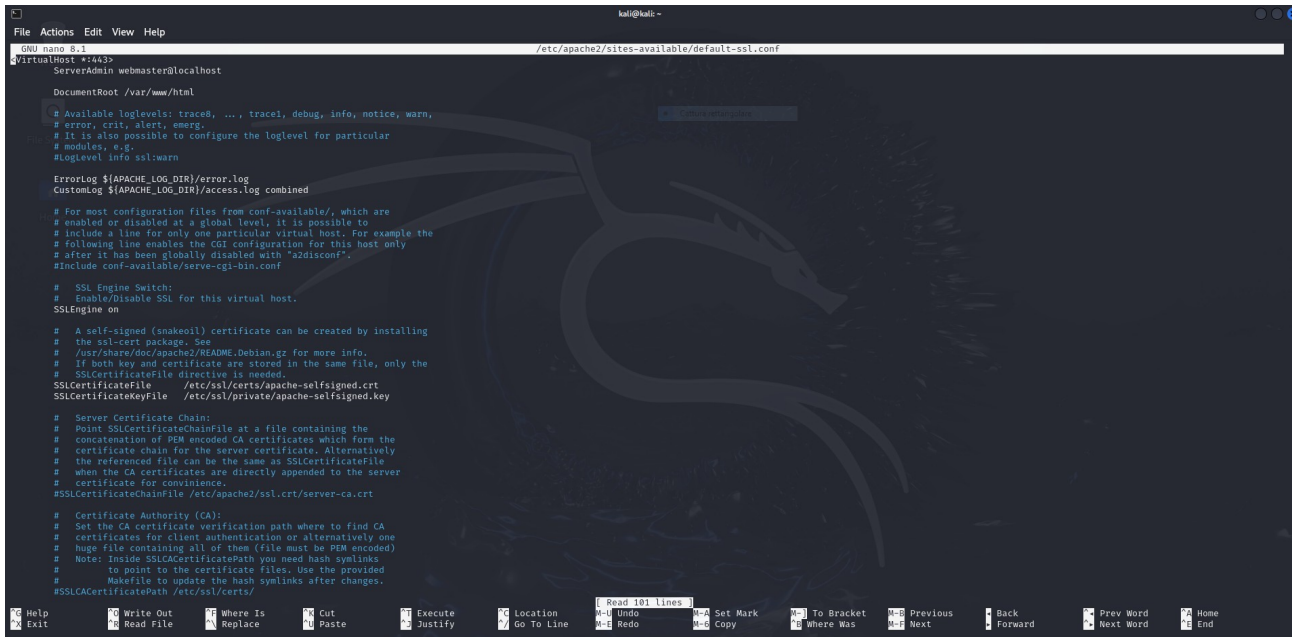


1.2 Configurazione richiesta Https (Apache)

- Verifico disponibilità update eseguendo *-sudo apt update*
- Procedo ad installare Apache eseguendo *-install apache2*
- Abilito modulo ssl eseguendo *-sudo a2enmod ssl*
- Creo Certificato autofirmato eseguendo *-sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt*

1.3 Configurazione Apache per richiesta Https

Eseguo comando **-sudo nano /etc/apache2/sites-available/default-ssl.conf**



```
GNU nano 8.1 /etc/apache2/sites-available/default-ssl.conf
ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCertificatePath you need hash symlinks
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/ssl/certs/
```

In questa fase dovrò prestare attenzione che siano presenti le seguenti voci:

- <VirtualHost default_:443>
- DocumentRoot /var/www/html
- SSLEngine on
- SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
- SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.ke

Al fine di abilitare ora il sito SSL e riavviare Apache eseguirò i comandi:

-sudo a2ensite default-ssl (Abilito ssl)

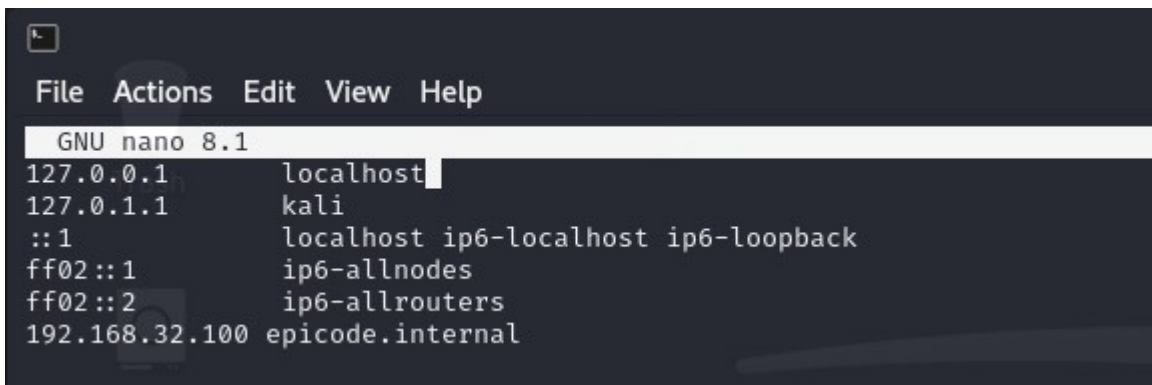
-sudo systemctl restart apache2 (Riavvio Apache)

1.4 Configurazione DNS Kali

Per assicurarmi che il servizio DNS possa risolvere `epicode.internal`, è necessario aggiungere al file `/etc/hosts` "`epicode.internal`"

-Eseguo il comando `-sudo nano /etc/hosts`

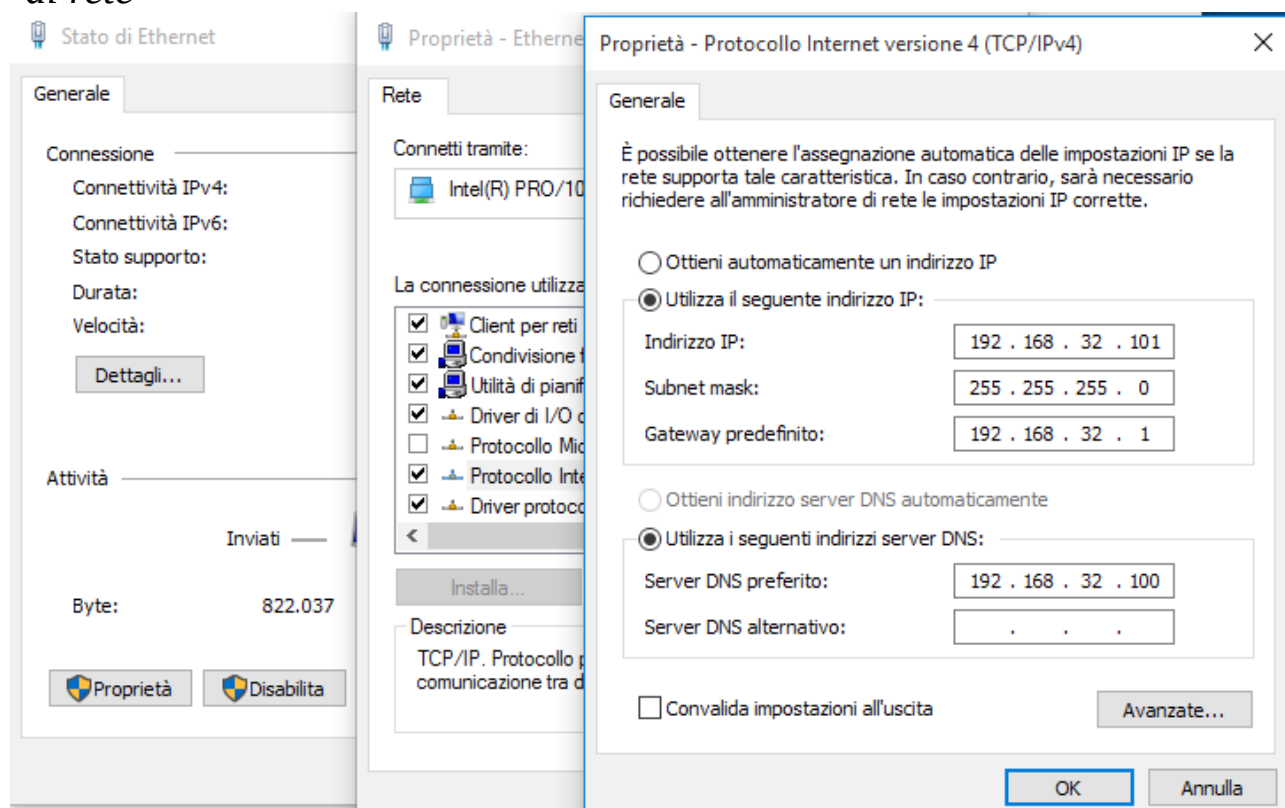
-Scorro fino alla fine e aggiungo la riga "`192.168.32.100 epicode.internal`", salvo le modifiche e termino.



```
GNU nano 8.1
File Actions Edit View Help
127.0.0.1    localhost
127.0.1.1    kali
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
192.168.32.100 epicode.internal
```

1.5 Configurazione Windows e impostazione servizio DNS

Avvio ora la VM Windows e inizio con l'impostare correttamente la scheda di rete



Oltre ad assegnare l'indirizzo IP, la Subnet mask ed il Gateway predefinito, inserisco l'indirizzo IP "192.168.32.100" (corrispondente all'IP del servizio DNS precedentemente impostato su Kali) nel riquadro : "Server DNS preferito".

*Apro il blocco note come amministratore, clicco su file, apro, dunque seguo il seguente percorso: **C:\Windows\System32\drivers\etc**.*

Mi assicuro di selezionare "tutti i file" nel menu a tendina in basso a destra per poter visualizzare il file "hosts", dunque seleziono e clicco su apri.

*A questo punto, anche qui, inserisco la riga di testo : "**192.168.32.100** **epicode.internal**." salvo le modifiche e chiudo il blocco note.*

Verifico ora la configurazione avviando un ping sia da Kali che da Windows, eseguo infine <https://epicode.internal> direttamente dal browser.

Di seguito riporto le verifiche effettuate per essere sicuro che il servizio DNS funzionasse correttamente.

```

(kali㉿kali)-[~]
$ ping epicode.internal
PING epicode.internal (192.168.32.100) 56(84) bytes of data.
 64 bytes from epicode.internal (192.168.32.100): icmp_seq=1 ttl=64 time=0.015 ms
 64 bytes from epicode.internal (192.168.32.100): icmp_seq=2 ttl=64 time=0.023 ms
 64 bytes from epicode.internal (192.168.32.100): icmp_seq=3 ttl=64 time=0.060 ms
 64 bytes from epicode.internal (192.168.32.100): icmp_seq=4 ttl=64 time=0.023 ms
 64 bytes from epicode.internal (192.168.32.100): icmp_seq=5 ttl=64 time=0.027 ms
 64 bytes from epicode.internal (192.168.32.100): icmp_seq=6 ttl=64 time=0.027 ms
 64 bytes from epicode.internal (192.168.32.100): icmp_seq=7 ttl=64 time=0.018 ms
^C
— epicode.internal ping statistics —
 7 packets transmitted, 7 received, 0% packet loss, time 6132ms
 rtt min/avg/max/mdev = 0.015/0.027/0.060/0.013 ms

```

```

Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user>ping epicode.internal

Esecuzione di Ping epicode.internal [192.168.32.100] con 32 byte di dati:
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
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\user>

```

Windows 10 pro - Metasploitable 1 [In esecuzione] - Oracle VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

Cestino debug

Icecast2 Win32 Malware

tomcat Programmi per Malwa...

Google Chrome

Apache2 Debian Default Page

Non sicuro epicode.internal/

Apache2 Debian Default Page

debian

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

Ricerca in Windows e nel Web

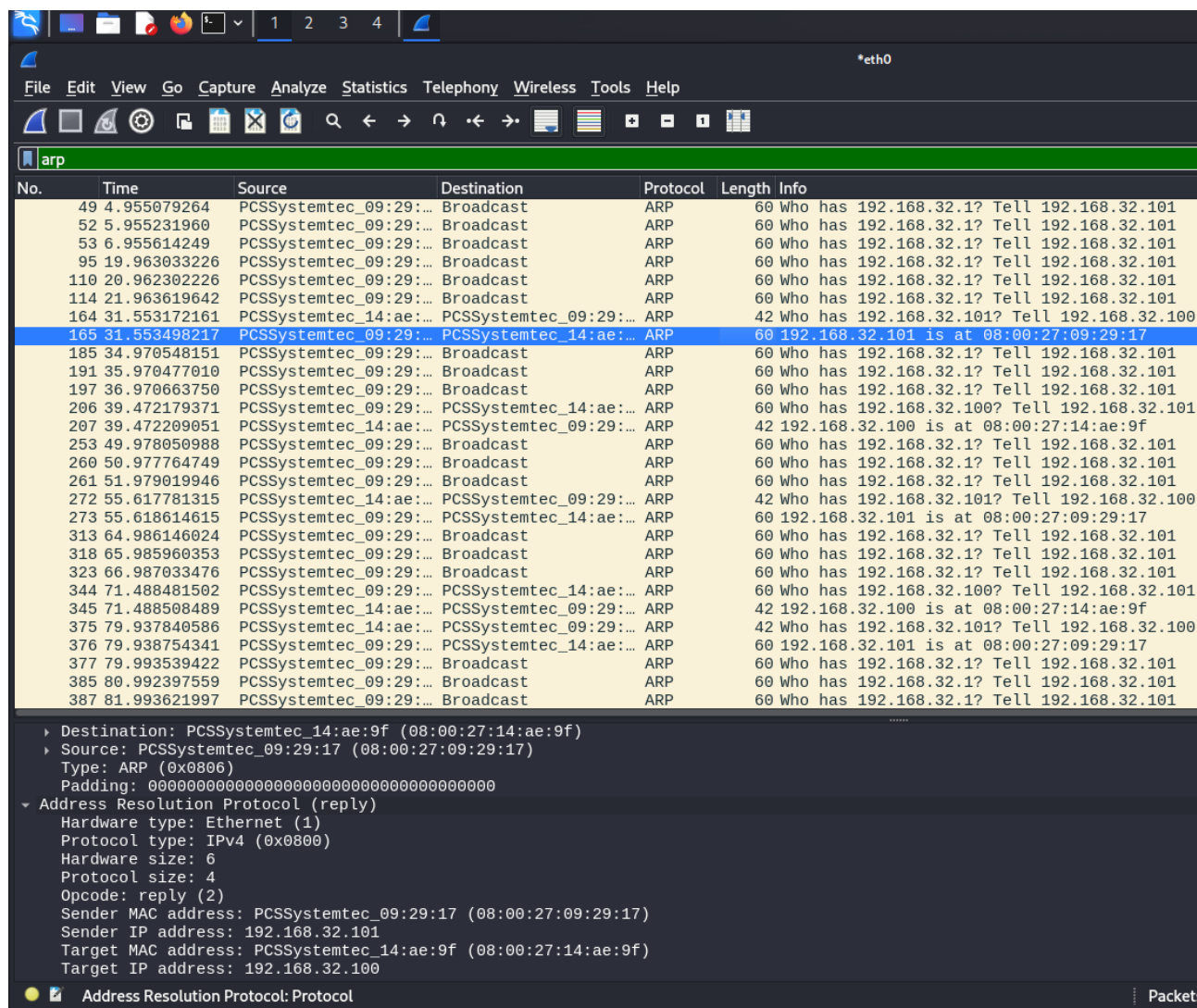
ITA 17:23 02/12/2024

2) INTERCETTAZIONE DEL TRAFFICO HTTPS

- Apro Wireshark;
- Seleziono l'interfaccia di rete "eth0";
- Apro ora dal web con la VM Windows la pagina <https://epicode.internal/>;
- Visualizzerò il traffico dati tramite Wireshark.

2.1 Indirizzi MAC:

I pacchetti ARP contengono direttamente le informazioni sugli indirizzi MAC, poiché vengono utilizzati per determinare l'indirizzo MAC associato a un determinato indirizzo IP.



The image shows a Wireshark network traffic capture on the 'eth0' interface, filtered for 'arp'. The packet list shows a series of ARP requests (No. 49 to 387) from PCSSystemtec_09:29:17 to various destinations, including Broadcast and specific IP addresses. The selected packet (No. 165) is an ARP request from PCSSystemtec_09:29:17 to PCSSystemtec_14:ae:9f. The packet details pane shows the following information:

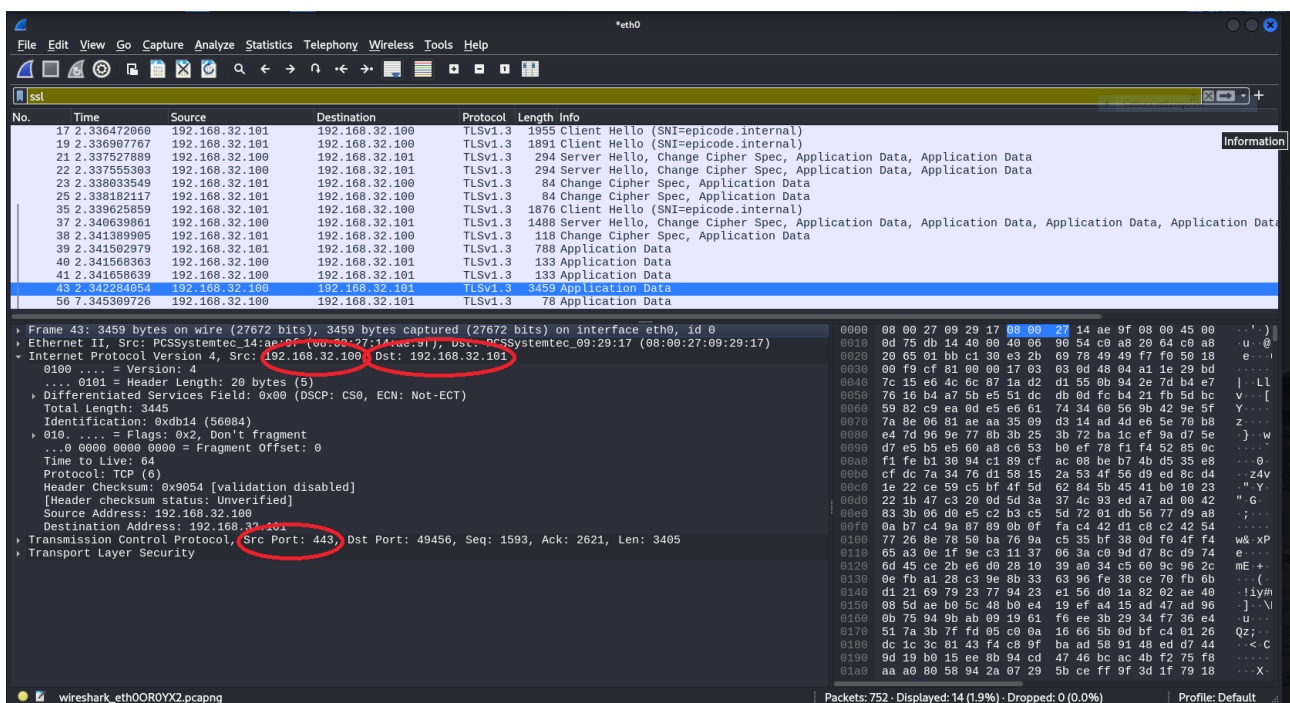
- Destination: PCSSystemtec_14:ae:9f (08:00:27:14:ae:9f)
- Source: PCSSystemtec_09:29:17 (08:00:27:09:29:17)
- Type: ARP (0x0806)
- Padding: 00000000000000000000000000000000
- Address Resolution Protocol (reply)
 - Hardware type: Ethernet (1)
 - Protocol type: IPv4 (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: reply (2)
 - Sender MAC address: PCSSystemtec_09:29:17 (08:00:27:09:29:17)
 - Sender IP address: 192.168.32.101
 - Target MAC address: PCSSystemtec_14:ae:9f (08:00:27:14:ae:9f)
 - Target IP address: 192.168.32.100

The packet bytes pane shows the raw data of the ARP request and reply.

2.2 Intercettazione traffico Https

Anche se non riesco visualizzare i dati reali del traffico HTTPS (poiché sono cifrati), posso comunque raccogliere informazioni utili dai pacchetti SSL/TLS, come:

- Gli indirizzi IP e le porte di origine e destinazione;
- Le informazioni sull'Handshake SSL/TLS;
- Il tipo di algoritmo di cifratura utilizzato.



(Si potrebbe procedere a catturare il traffico Https e TLS con decrittazione, così da avere informazioni chiare riguardo il traffico dati).

3) SOSTITUZIONE SERVIZIO HTTPS con HTTP E INTERCETTAZIONE TRAFFICO.

Per configurare il server per http, su Kali, modifico la configurazione di Apache per abilitare HTTP e assicurarsi che lo stesso sia attivo sulla porta 80, il tutto con i seguenti comandi:

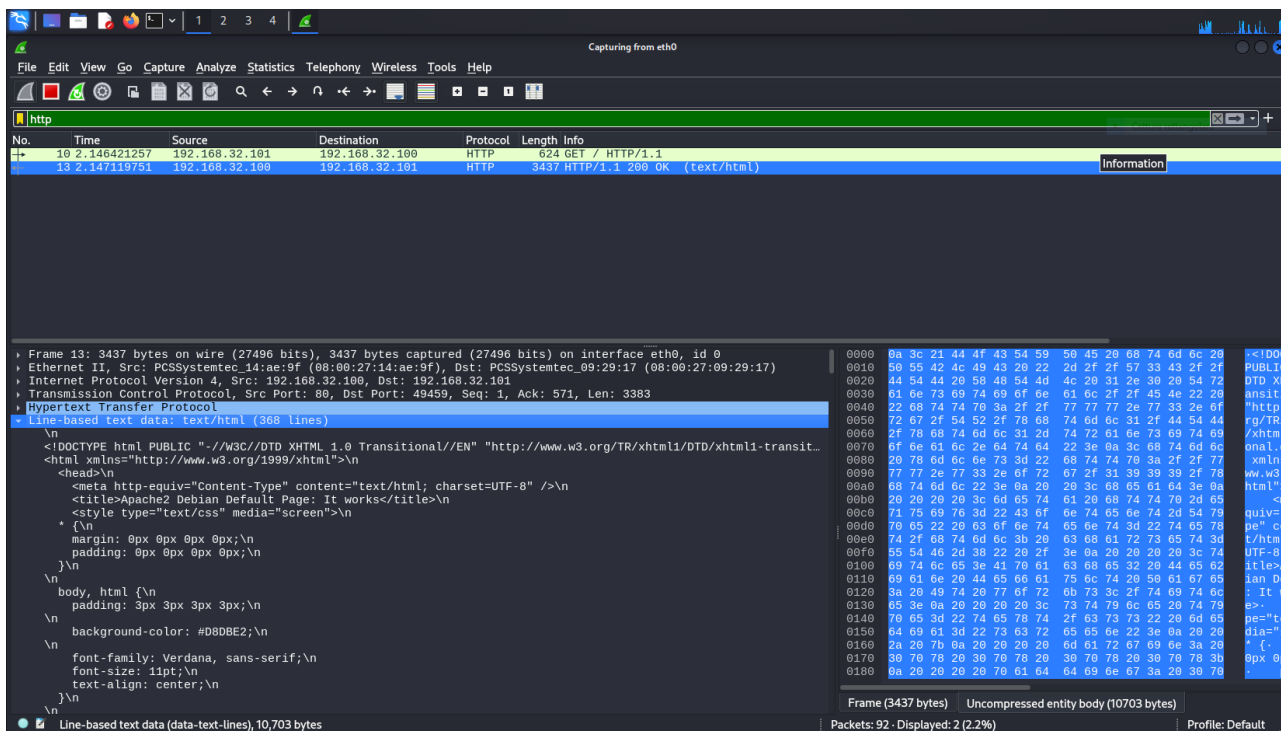
*-sudo a2dissite default-ssl;
-sudo a2ensite 000-default;
-sudo systemctl restart apache.*

Inizio ora una nuova cattura su Wireshark.

Effettuo una richiesta HTTP dal browser di windows come fatto prima questa volta inserendo <http://epicode.internal> .

Analizzo ora il traffico HTTP in Wireshark.

Inserisco nel filtro il termine “http” per visualizzare solo i pacchetti relativi ad esso e dunque osservarne i dettagli.



4) CONCLUSIONE: Differenze tra i servizi e considerazioni in merito

Dalle rilevazioni effettuate tramite Wireshark, è possibile notare quali siano le differenze tra i due tipi di traffico, che si differenziano per la presenza della crittografia, la visibilità dei dati e la porta utilizzata (443 per le comunicazioni https, crittate; 80 per le comunicazioni http, visibili e poco sicure)

Durante le rilevazioni traffico Https non è stato possibile visualizzare i dati reali del traffico HTTPS (non avendo effettuato peraltro un'operazione di decrittazione) mentre in quelle http è stato possibile.

Posso quindi affermare che HTTPS offre un livello di sicurezza e protezione dei dati grazie alla crittografia mentre HTTP espone le informazioni a potenziali intercettazioni e attacchi.

È fondamentale utilizzare HTTPS per garantire la sicurezza delle comunicazioni online, specialmente quando si trattano dati sensibili.