

Nell'esercizio di oggi metteremo insieme le competenze acquisite finora. Lo studente verrà valutato sulla base della risoluzione al problema seguente.

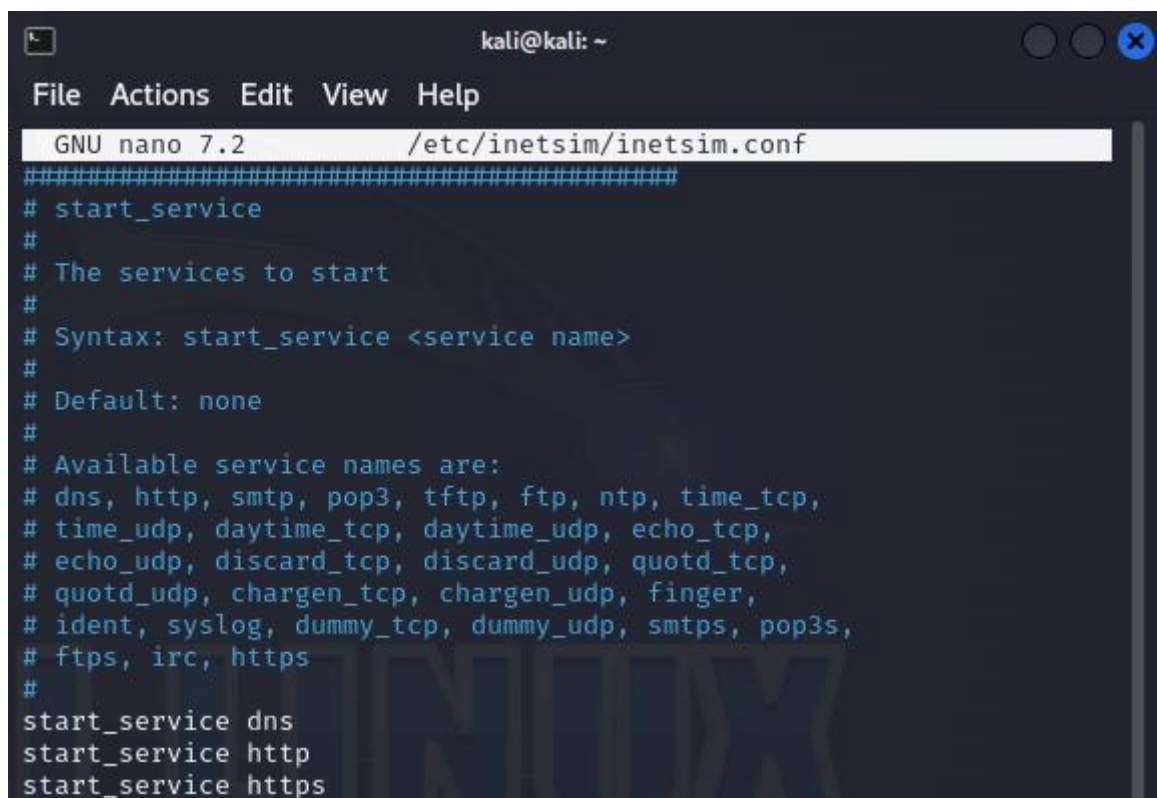
Requisiti e servizi:

- **Kali Linux - > IP 192.168.32.100**
- **Windows 7 -> IP 192.168.32.101**
- **HTTPS server: attivo**
- **Servizio DNS per risoluzione nomi di dominio: attivo**

Traccia:

1. Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows 7) richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100 (Kali).
2. Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.
3. 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.

Configurazione inetsim:



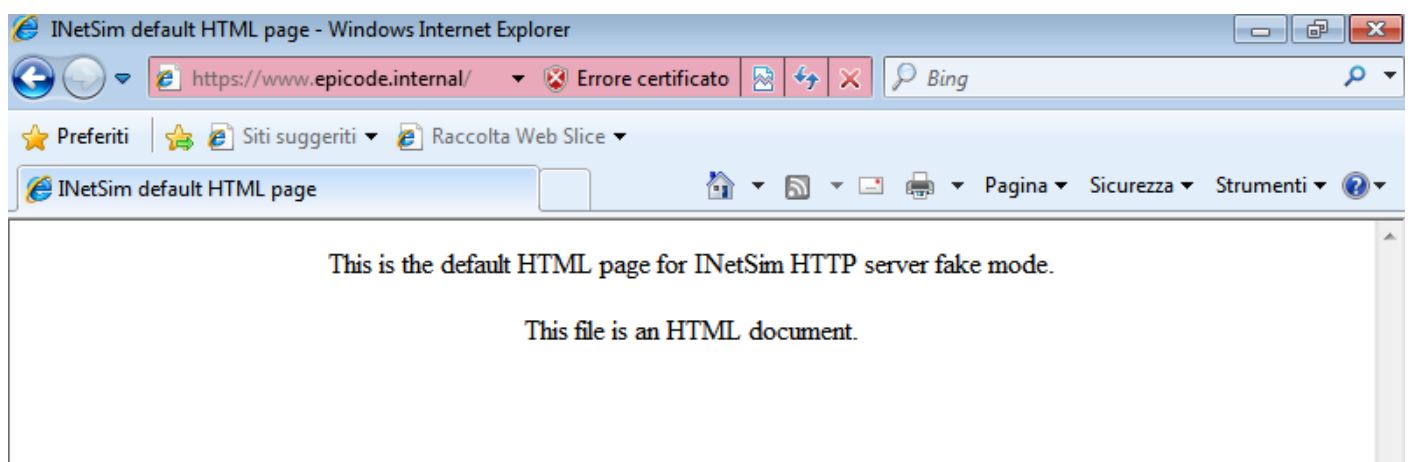
```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf  
#####  
# start_service  
#  
# The services to start  
#  
# Syntax: start_service <service name>  
#  
# Default: none  
#  
# Available service names are:  
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,  
# 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
```

```
#####  
# 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 www.epicode.internal 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 192.168.32.100
```

```
#####  
# dns_default_ip  
#  
# Default IP address to return with DNS replies  
#  
# Syntax: dns_default_ip <IP address>  
#  
# Default: 127.0.0.1  
#  
dns_default_ip 192.168.32.100
```

Connessione da browser Windows 7 su epicode.internal:



Differenze su Wireshark:

The screenshot shows a Wireshark capture of network traffic on interface eth0. The packet list displays several packets, with packet 57 (187 bytes) selected, showing a TLSv1 Record Layer: Handshake Protocol: Client Hello. The packet details pane shows the following information:

- Frame 57: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits) on interface eth0.
- Ethernet II, Src: PcsCompu_de:54:cc (08:00:27:de:54:cc), Dst: PcsCompu_cb:7e:f5 (08:00:27:de:54:cc).
- Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100.
- Transmission Control Protocol, Src Port: 49176, Dst Port: 443, Seq: 1, Ack: 1, Len: 133.
- Transport Layer Security
 - TLSv1 Record Layer: Handshake Protocol: Client Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 128
 - Handshake Protocol: Client Hello

The screenshot shows a Wireshark capture of network traffic on interface eth0. The packet list displays several packets, with packet 12 (340 bytes) selected, showing an HTTP GET request. The packet details pane shows the following information:

- Frame 12: 340 bytes on wire (2720 bits), 340 bytes captured (2720 bits) on interface eth0.
- Ethernet II, Src: PcsCompu_de:54:cc (08:00:27:de:54:cc), Dst: PcsCompu_cb:7e:f5 (08:00:27:de:54:cc).
- Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100.
- Transmission Control Protocol, Src Port: 49245, Dst Port: 80, Seq: 1, Ack: 1, Len: 286.
- Hypertext Transfer Protocol

La differenza principale tra HTTP e HTTPS sta nella sicurezza e nell'aspetto crittografico della comunicazione tra il tuo browser e il server web che stai visitando.

- **HTTP (Hypertext Transfer Protocol):** È il protocollo di comunicazione utilizzato per trasmettere e visualizzare pagine web. Tuttavia, è un protocollo non sicuro, il che significa che i dati trasmessi tramite HTTP non sono crittografati. Questo rende vulnerabili le informazioni scambiate tra il tuo browser e il server web a potenziali intercettazioni da parte di terzi. I dati inviati tramite HTTP sono in chiaro e possono essere facilmente letti o manipolati.
- **HTTPS (Hypertext Transfer Protocol Secure):** È la versione sicura di HTTP. Utilizza un protocollo aggiuntivo chiamato SSL/TLS (Secure Sockets Layer/Transport Layer Security) per crittografare i dati scambiati tra il browser e il server. Questo livello di crittografia rende molto più difficile per gli attaccanti intercettare o manipolare le informazioni trasmesse. Puoi riconoscere un sito web sicuro HTTPS dalla presenza di un lucchetto nella barra degli indirizzi del browser, insieme alla dicitura "https://" prima dell'URL del sito.

In sintesi, mentre entrambi HTTP e HTTPS sono protocolli per la trasmissione di dati su internet, HTTPS offre un livello aggiuntivo di sicurezza grazie alla crittografia dei dati, rendendo più sicure le comunicazioni tra il browser e il server web.