

Differenze tra HTTP e HTTPS



Esercizio
Traccia e requisiti

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:

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).

Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

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.

Esecuzione:

Come richiesto dalla traccia sopra riportata sono andato ad impostare gli IP statici su entrambe le macchine virtuali: tramite interfaccia grafica in Windows e invece tramite terminale in Kali Linux usando il comando: "sudo nano /etc/network/interfaces". ([Screenshot 1](#))

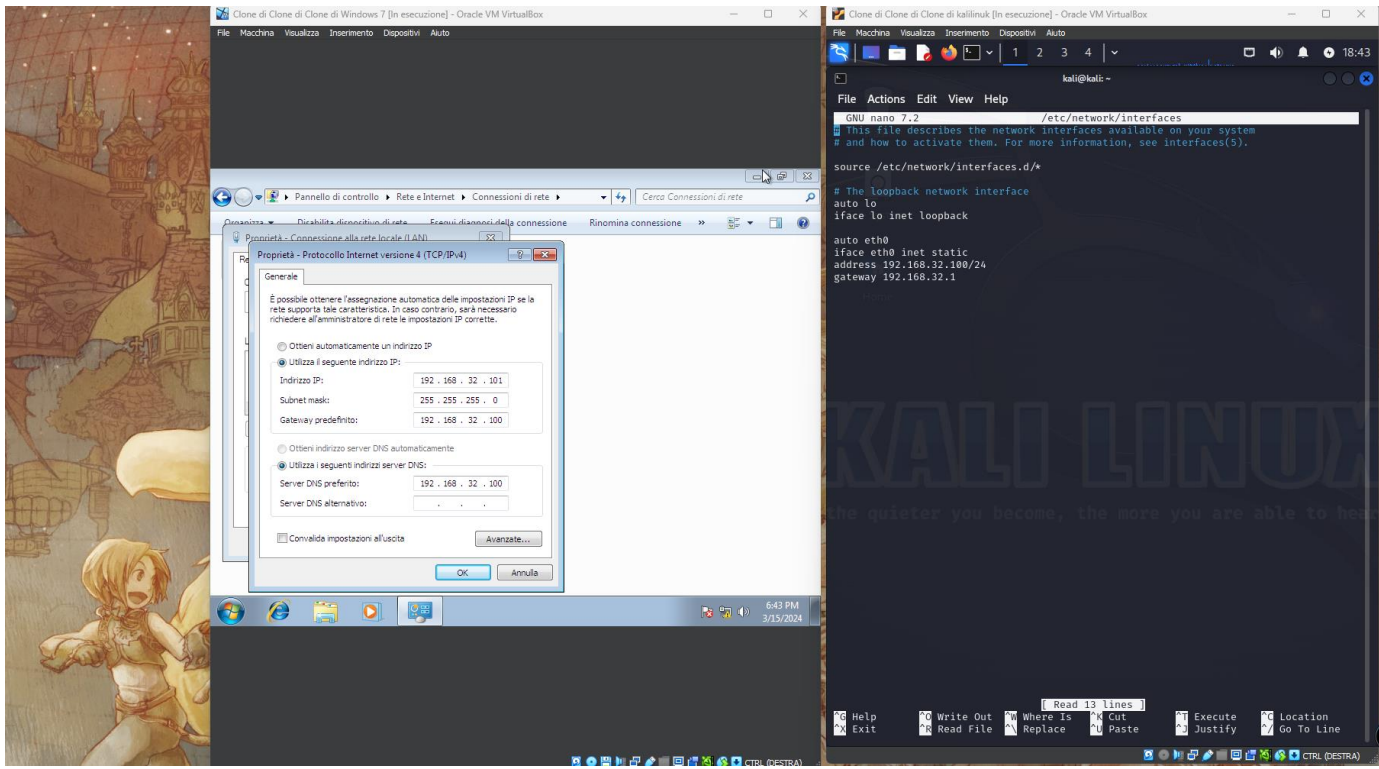
Fatto ciò tramite il comando "sudo nano /etc/inetsim/inetsim.conf" sono andato ad attivare e impostare il servizio DNS su Kali Linux e i servizi HTTP e HTTPS, per ultimo punto sono tornato in Windows e ho impostato nell'interfaccia di rete tramite pannello di controllo il DNS di Kali Linux per poter "puntare" correttamente e risolvere il dominio con successo. ([Screenshot 2](#))

A questo punto ero pronto alla cattura dei pacchetti tramite il programma Wireshark presente in Kali Linux. Dopo aver avviato Wireshark sono tornato sulla macchina di windows 7 e tramite internet explorer mi sono diretto sul sito: epicode.internal e ho analizzato i pacchetti ([Screenshot 3](#))

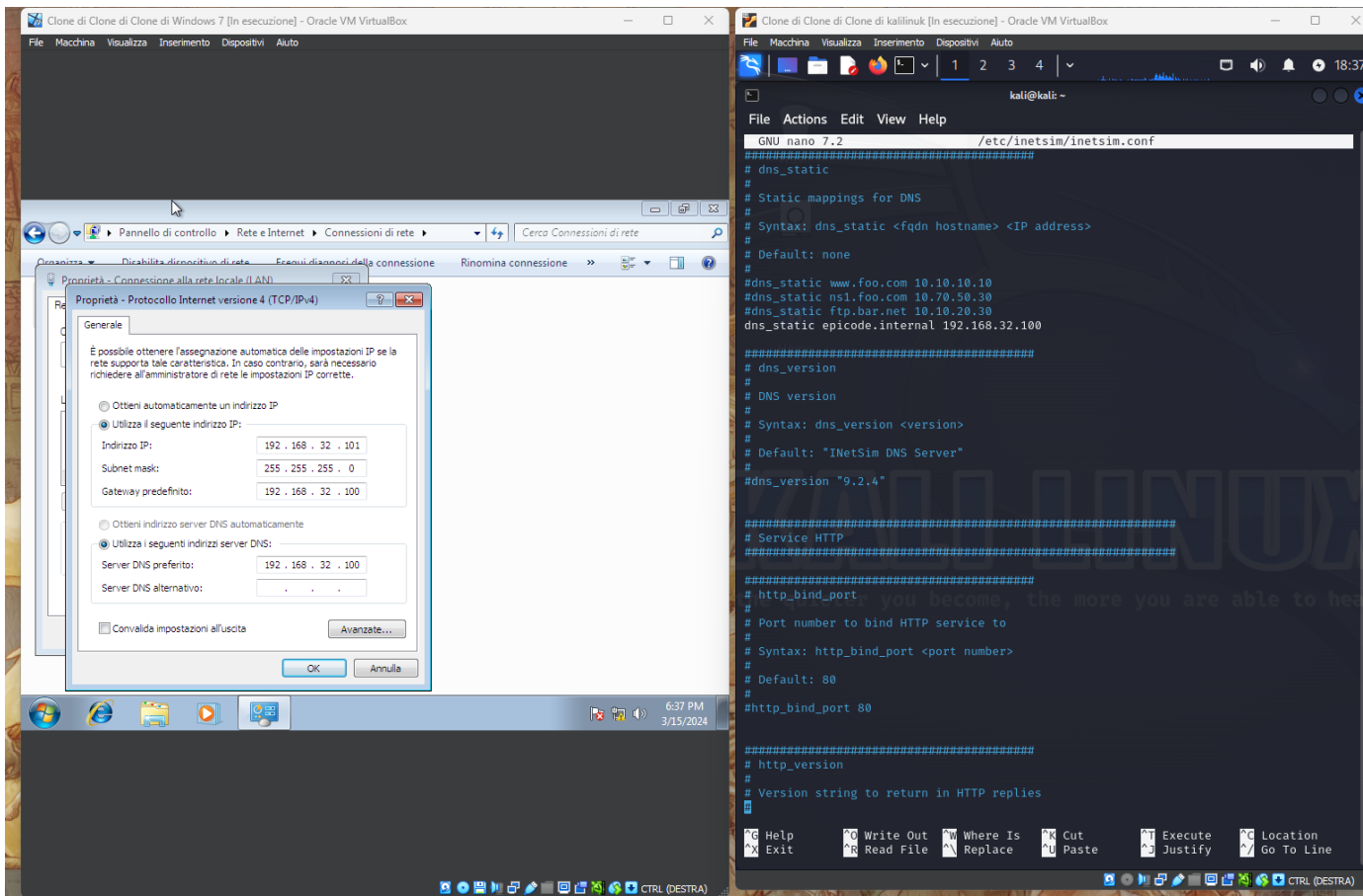
Dopo una prima fase di cattura di pacchetti del protocollo HTTP ho ricercato tramite la barra degli indirizzi la versione del sito in HTTPS e analizzato i pacchetti ([Screenshot 4 e 5](#))

Dopo gli screenshot riporterò le principali differenze riscontrate tra HTTP e HTTPS

Screenshot 1



Screenshot 2



Screenshot 3

This is the INetSim real-mode test page...

the Internet

Frame 6: 333 bytes on wire (2664 bits), 333 bytes captured (2664 bits) on Ethernet II, Src: PCSSystemtec_ab:ai:2d (08:00:27:ab:ai:2d), Dst: PCSSystemtec_ab:ai:2d (08:00:27:ab:ai:2d), Seq=1, Ack=1, Len=0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PCSSystemtec_ab:ai:2d	Broadcast	ARP	60	who has 192.168.32.100? Tell me
2	0.000013867	PCSSystemtec_62:5b:8e	PCSSystemtec_ab:ai:2d	ARP	42	192.168.32.100 is at 08:00:27:ab:ai:2d
3	0.000397794	192.168.32.101	192.168.32.100	TCP	66	49164 → 80 [SYN] Seq=0 Win=8192 Len=0
4	0.000416794	192.168.32.100	192.168.32.101	TCP	66	80 → 49164 [SYN, ACK] Seq=0 A
5	0.000615851	192.168.32.101	192.168.32.100	TCP	66	49164 → 80 [ACK] Seq=1 Ack=1
6	0.00111615	192.168.32.101	192.168.32.100	HTTP	333	GET /internet.gif HTTP/1.1
7	0.001120689	192.168.32.100	192.168.32.101	TCP	54	80 → 49164 [ACK] Seq=1 Ack=280
8	0.012315921	192.168.32.100	192.168.32.101	TCP	204	80 → 49164 [PSH, ACK] Seq=1 Ack=280
9	0.014459819	192.168.32.100	192.168.32.101	HTTP	243	HTTP/1.1 200 OK (text/html)
10	0.014802023	192.168.32.101	192.168.32.100	TCP	66	49164 → 80 [ACK] Seq=280 Ack=280
11	0.015047044	192.168.32.101	192.168.32.100	TCP	66	49164 → 80 [FIN, ACK] Seq=280
12	0.015058450	192.168.32.100	192.168.32.101	TCP	54	80 → 49164 [ACK] Seq=341 Ack=1
13	0.015718371	192.168.32.101	192.168.32.100	TCP	66	49165 → 80 [SYN] Seq=0 Win=8192
14	0.015731428	192.168.32.100	192.168.32.101	TCP	66	80 → 49165 [ACK] Seq=1 Ack=1
15	0.016140449	192.168.32.101	192.168.32.100	TCP	66	49165 → 80 [ACK] Seq=1 Ack=1
16	0.023057990	192.168.32.101	192.168.32.100	HTTP	380	GET /internet.gif HTTP/1.1
17	0.023078409	192.168.32.100	192.168.32.101	TCP	54	80 → 49165 [ACK] Seq=1 Ack=327
18	0.031738045	192.168.32.100	192.168.32.101	TCP	200	80 → 49165 [PSH, ACK] Seq=1 Ack=327
19	0.031931259	192.168.32.100	192.168.32.101	TCP	7354	80 → 49165 [PSH, ACK] Seq=153
20	0.032085555	192.168.32.100	192.168.32.101	TCP	5894	80 → 49165 [PSH, ACK] Seq=7453

Screenshot 4

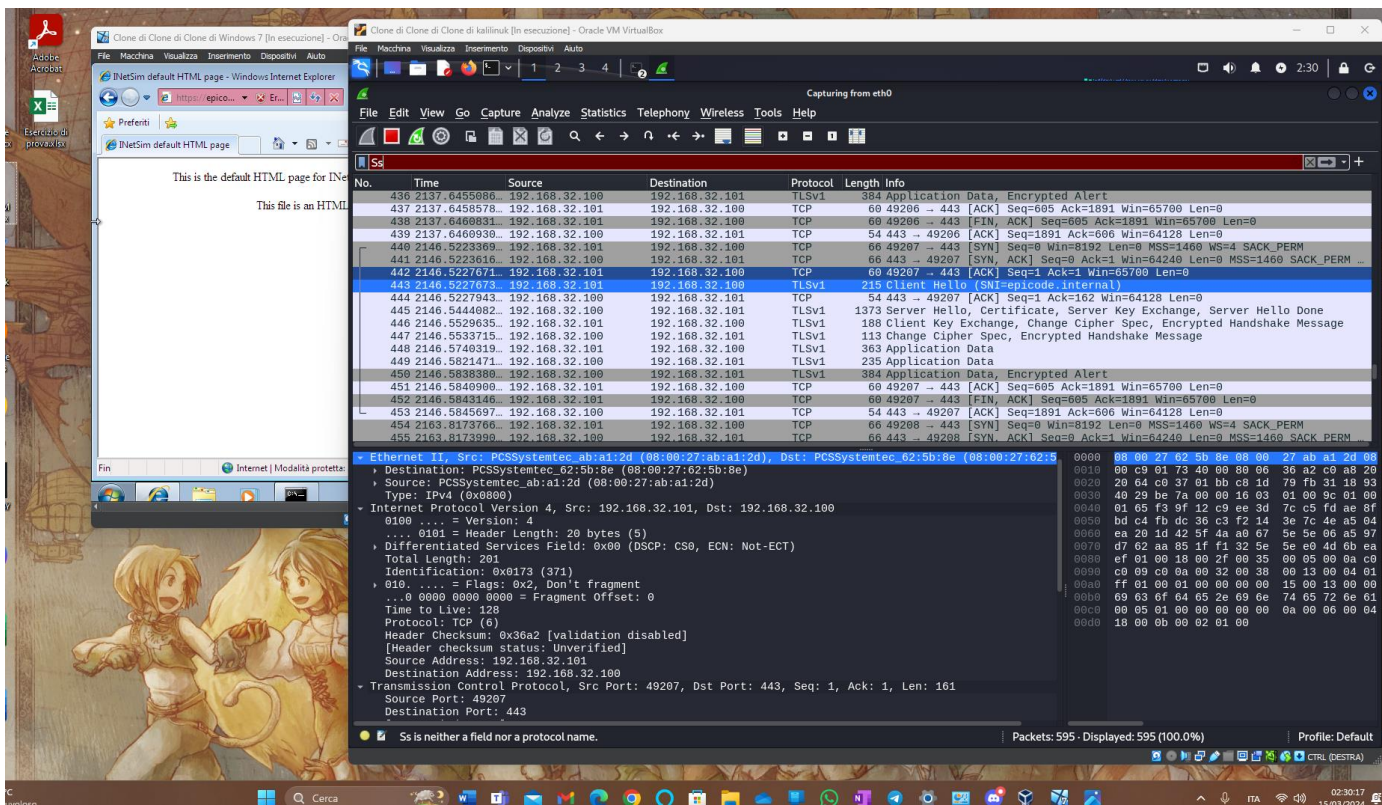
This is the default HTML page for INetSim

This file is an HTML document

Frame 6: 333 bytes on wire (2664 bits), 333 bytes captured (2664 bits) on Ethernet II, Src: PCSSystemtec_ab:ai:2d (08:00:27:ab:ai:2d), Dst: PCSSystemtec_ab:ai:2d (08:00:27:ab:ai:2d), Seq=1, Ack=1, Len=0

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000416794	192.168.32.100	192.168.32.101	TCP	66	80 → 49164 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
5	0.000615851	192.168.32.101	192.168.32.100	TCP	66	49164 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
6	0.00111615	192.168.32.101	192.168.32.100	HTTP	333	GET /internet.gif HTTP/1.1
7	0.001120689	192.168.32.100	192.168.32.101	TCP	54	80 → 49164 [ACK] Seq=1 Ack=280 Win=8192 Len=0
8	0.012315921	192.168.32.100	192.168.32.101	TCP	204	80 → 49164 [PSH, ACK] Seq=1 Ack=280 Win=8192 Len=0
9	0.014459819	192.168.32.100	192.168.32.101	HTTP	243	HTTP/1.1 200 OK (text/html)
10	0.014802023	192.168.32.101	192.168.32.100	TCP	66	49164 → 80 [ACK] Seq=280 Ack=280 Win=8192 Len=0
11	0.015047044	192.168.32.101	192.168.32.100	TCP	66	49164 → 80 [FIN, ACK] Seq=280 Ack=280 Win=8192 Len=0
12	0.015058450	192.168.32.100	192.168.32.101	TCP	54	80 → 49164 [ACK] Seq=341 Ack=281 Win=8192 Len=0
13	0.015718371	192.168.32.101	192.168.32.100	TCP	66	49165 → 80 [SYN] Seq=0 Win=8192 Len=0
14	0.015731428	192.168.32.100	192.168.32.101	TCP	66	80 → 49165 [ACK] Seq=1 Ack=1 Win=8192 Len=0
15	0.016140449	192.168.32.101	192.168.32.100	TCP	66	49165 → 80 [ACK] Seq=1 Ack=1 Win=8192 Len=0
16	0.023057990	192.168.32.101	192.168.32.100	HTTP	380	GET /internet.gif HTTP/1.1
17	0.023078409	192.168.32.100	192.168.32.101	TCP	54	80 → 49165 [ACK] Seq=1 Ack=327 Win=8192 Len=0
18	0.031738045	192.168.32.100	192.168.32.101	TCP	200	80 → 49165 [PSH, ACK] Seq=1 Ack=327 Win=8192 Len=0
19	0.031931259	192.168.32.100	192.168.32.101	TCP	7354	80 → 49165 [PSH, ACK] Seq=153 Ack=327 Win=8192 Len=0
20	0.032085555	192.168.32.100	192.168.32.101	TCP	5894	80 → 49165 [PSH, ACK] Seq=7453 Ack=327 Win=8192 Len=0
21	0.032335408	192.168.32.101	192.168.32.100	TCP	66	49165 → 80 [ACK] Seq=327 Ack=327 Win=8192 Len=0
22	0.032380431	192.168.32.100	192.168.32.101	TCP	3298	80 → 49165 [PSH, ACK] Seq=13293 Ack=327 Win=8192 Len=0
23	0.032597382	192.168.32.100	192.168.32.101	TCP	68	TCP Window Update: 49165 → 80 [ACK] Seq=327 Ack=13293 Win=65536 Len=0

Screenshot 5



Sono andato ad esplorare le informazioni sui pacchetti catturati in maniera più approfondita nei [Screenshot 4 e 5](#) ed ecco le mie conclusioni:

Prima di tutto la quantità delle informazioni catturate e visibile in chiaro è diversa, questo ci porta in evidenza la prima grande differenza tra HTTP e HTTPS, infatti il traffico HTTPS è crittografato e non sono in grado di vedere il contenuto della richiesta e della risposta in chiaro invece visibile nel http

MAC address Destination: (08:00:27:62:5b:8e)

MAC address Source: (08:00:27:ab:a1:2d)

La seconda differenza sta appunto nei protocolli utilizzati mentre per la trasmissione in chiaro vengono utilizzati HTTP e TCP nella trasmissione crittografata possiamo notare anche il protocollo TLSv1 un protocollo che si usa appunto per trasmissione di dati crittografata

La terza delle principali differenze che ho notato sta nell'uso delle porte, come si può vedere in fondo agli [Screenshot 4 e 5](#) Infatti HTTP usa la porta 80 per la comunicazione invece HTTPS usa la porta 443

In conclusione si può affermare che HTTPS fornisce un livello di sicurezza aggiuntivo, proteggendo i dati dalle intercettazioni.

Grazie per l'attenzione

Simone Cisbaglia