

Esercizio W4-D4 Garzoni Michele

Traccia esercizio:

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.

Iniziamo con impostare i corretti settaggi IP nelle due VM come scritto nella traccia, quindi:

Kali

Digito `sudo nano /etc/network/interfaces` e imposto l'ip 192.168.32.100 alla macchina Kali e 192.168.32.1 come Gateway con subnet 255.255.255.0 (24)

Digito `ifconfig` sul terminale e controllo i settaggi appena impostati.

```
(kali㉿kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.32.100 netmask 255.255.255.0 broadcast 192.168.32.255
```

Windows

Da Windows entro nelle connessioni di rete, Protocollo internet versione 4 (TCP/IP) e inserisco i valori corretti:

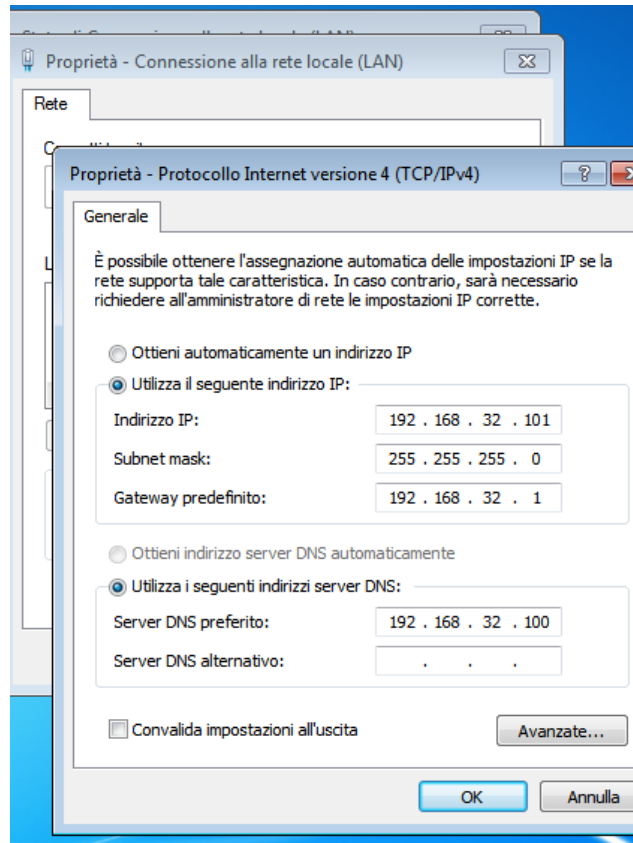
IP: 192.168.32.101

Subnet: 255.255.255.0

Gateway: 192.168.32.1

Impostiamo anche il server DNS sulla macchina Kali per dopo: 192.168.32.100

Premo Ok per confermare:



Faccio una prova di Ping, per confermare la corretta comunicazione in rete delle due VM:

```
C:\Users\uboxuser>ping 192.168.32.100

Esecuzione di Ping 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
```

```

(kali㉿kali)-[~]
$ ping 192.168.32.101
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data.
64 bytes from 192.168.32.101: icmp_seq=1 ttl=128 time=0.154 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.153 ms
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.166 ms
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.148 ms
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.155 ms
64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=0.173 ms
^C
— 192.168.32.101 ping statistics —
6 packets transmitted, 6 received, 0% packet loss, time 5113ms
rtt min/avg/max/mdev = 0.148/0.158/0.173/0.008 ms

(kali㉿kali)-[~]
$ 

```

Allego screen anche delle relative config dove si può vedere anche i MAC delle due schede di rete:

Ipconfig /all in Windows (Indirizzo fisico è il Mac Address)

```

C:\Windows\system32\cmd.exe
Scheda Ethernet Connessione alla rete locale (LAN):

Suffisso DNS specifico per connessione:
Descrizione . . . . . : Scheda desktop Intel(R) PRO/1000 MT
Indirizzo fisico. . . . . : 08-00-27-5A-DB-31
DHCP abilitato. . . . . : No
Configurazione automatica abilitata : Sì
Indirizzo IPv6 locale rispetto al collegamento . : fe80::3855:e071:1e2d:7861%11(Preferenziale)
Indirizzo IPv4. . . . . : 192.168.32.101(Preferenziale)
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . : 192.168.32.1
IAID DHCPv6 . . . . . : 235405351
DUID Client DHCPv6. . . . . : 00-01-00-01-2D-6C-25-47-08-00-27-5A-DB-31

Server DNS . . . . . : 192.168.32.100
NetBIOS su TCP/IP . . . . . : Attivato

Scheda Tunnel isatap.{06786796-DDAF-4E84-89CA-BD2F5B888D12}:

Stato supporto. . . . . : Supporto disconnesso
Suffisso DNS specifico per connessione:
Descrizione . . . . . : Microsoft ISATAP Adapter #2
Indirizzo fisico. . . . . : 00-00-00-00-00-00-E0
DHCP abilitato. . . . . : No
Configurazione automatica abilitata : No

```

ifconfig di Kali Linux (Ether è il Mac Address)

```

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.32.100 netmask 255.255.255.0 broadcast 192.168.32.255
    inet6 fe80::a00:27ff:fe21:b1d0 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)
    RX packets 434 bytes 47754 (46.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 237 bytes 121324 (118.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Proseguiamo su Kali aprendo la configurazione di INetSim, che fungerà da simulatore di una rete Internet.

Su Kali Terminale digitiamo "Sudo Nano /etc/inetsim/inetsim.conf con password "kali"

```

(kali@kali)-[~]
$ sudo nano /etc/inetsim/inetsim.conf
[sudo] password for kali:

```

Siamo nella configurazione del programma INetSim, dove andremo a cambiare i parametri per riflettere i nostri IP di rete e servizi necessari:

Modifichiamo il "Service bind address" inserendo l'IP della nostra macchina Kali:

```

#####
# service_bind_address
#
# IP address to bind services to
#er.pm line 399.
#Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address 192.168.32.100

```

Modifichiamo i settaggi DNS per risolvere correttamente il testo “epicode.internal” se digitato dal web browser Windows

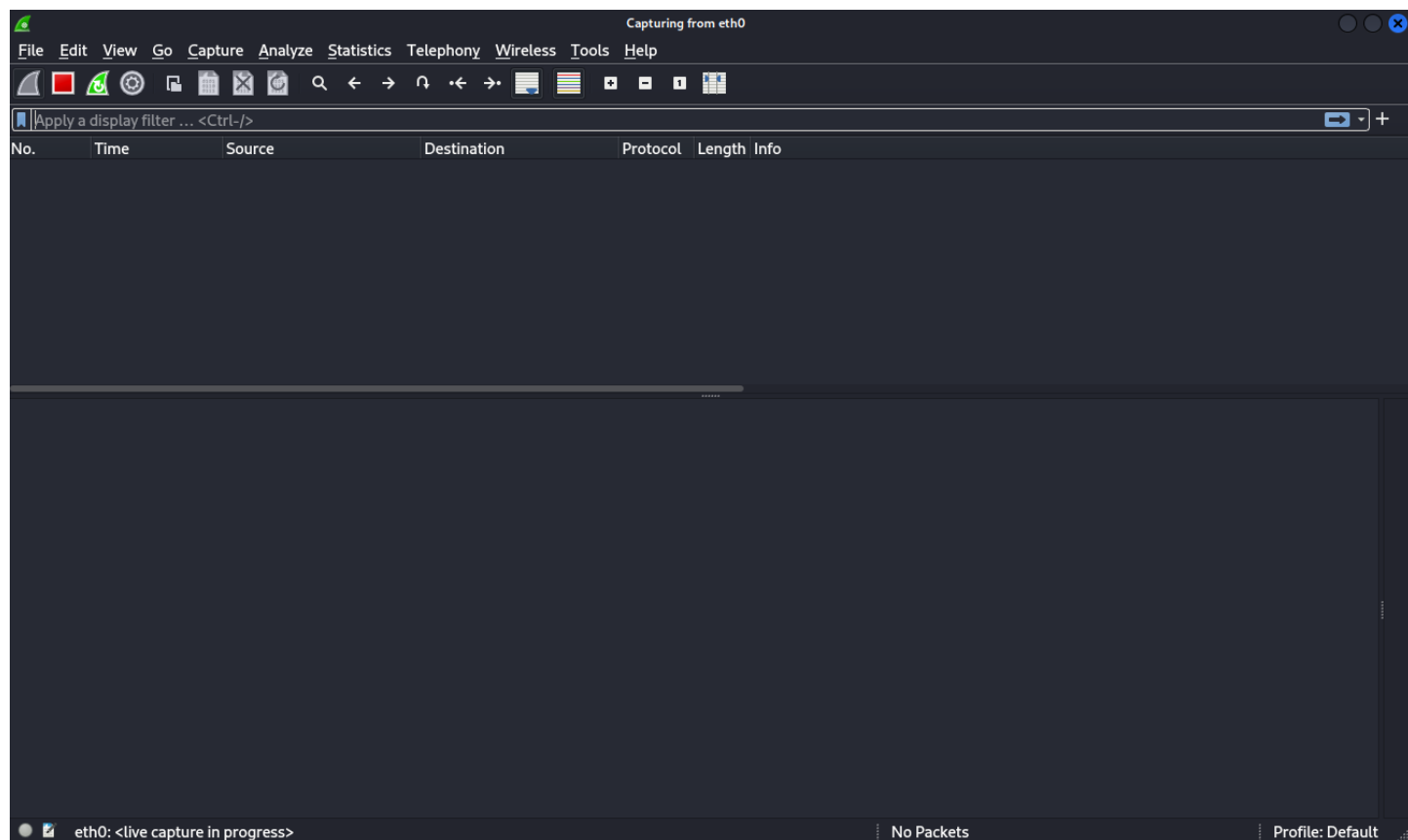
```
#####  
# dns_static  
#  
# Static mappings for DNS  
#  
# Syntax: dns_static <fqdn hostname> <IP address>  
#  
# Default: none  
#  
dns_static epicode.internal 192.168.32.100  
#dns_static ns1.foo.com 10.70.50.30  
#dns_static ftp.bar.net 10.10.20.30
```

Usciamo dal nano salvando le modifiche con le combinazioni di tasti “Ctrl + O” e “Ctrl + X”

Avvio Inetsim (Sudo INetSim) da terminale Kali

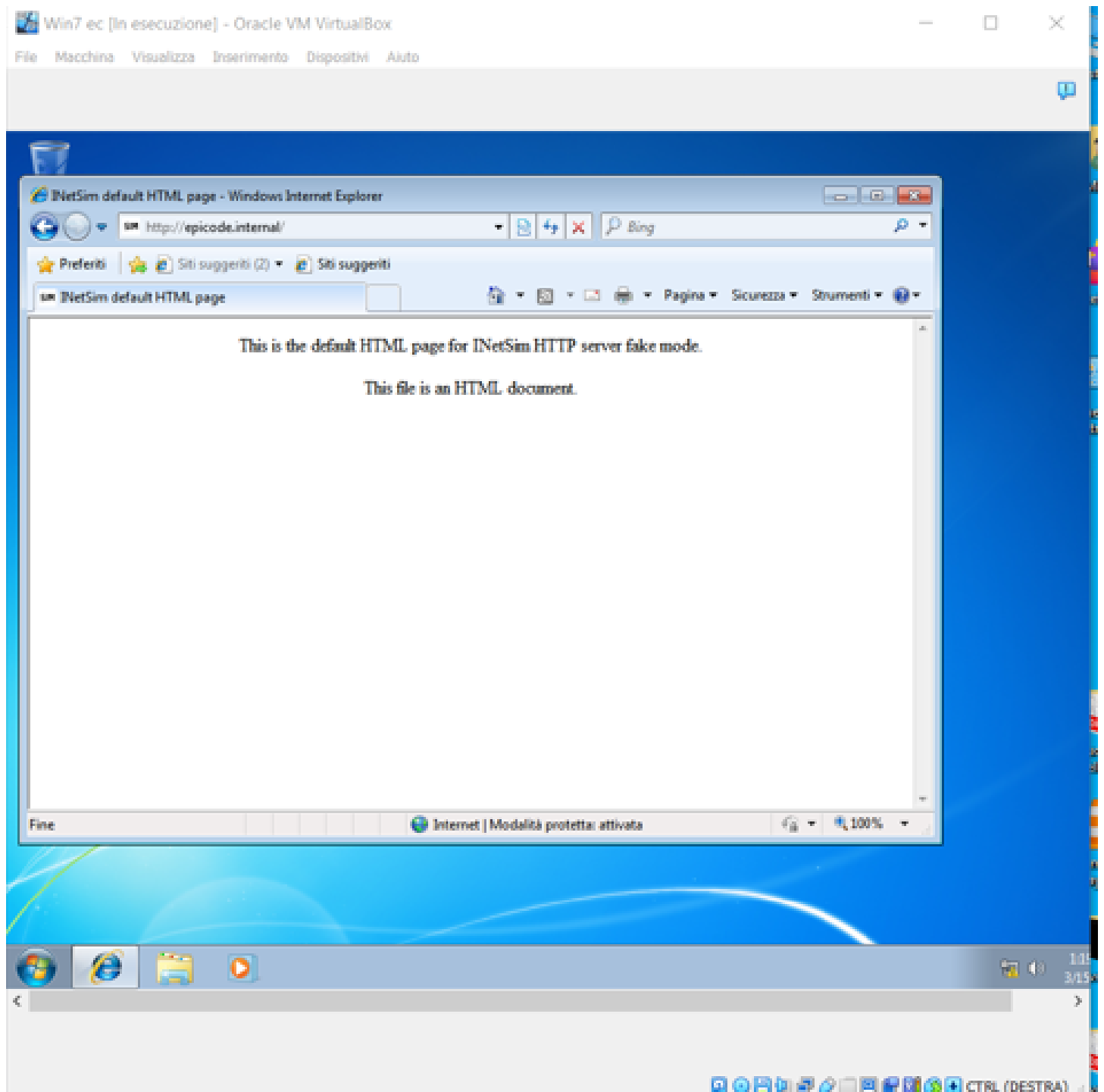
```
l-$ sudo inetsim  
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg  
Using log directory: /var/log/inetsim/  
Using data directory: /var/lib/inetsim/  
Using report directory: /var/log/inetsim/report/  
Using configuration file: /etc/inetsim/inetsim.conf  
Parsing configuration file.  
Configuration file parsed successfully.  
== INetSim main process started (PID 10182) ==  
Session ID: 10182  
Listening on: 192.168.32.100  
Real Date/Time: 2024-03-15 08:32:35  
Fake Date/Time: 2024-03-15 08:32:35 (Delta: 0 seconds)  
Forking services...  
* dns_53_tcp_udp - started (PID 10192)  
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm l  
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm l  
* irc_6667_tcp - started (PID 10202)  
* smtps_465_tcp - started (PID 10196)  
* pop3s_995_tcp - started (PID 10198)  
* ntp_123_udp - started (PID 10203)  
* echo_7_udp - started (PID 10212)  
* finger_79_tcp - started (PID 10204)  
* echo_7_tcp - started (PID 10211)  
* ident_113_tcp - started (PID 10205)  
* daytime_13_tcp - started (PID 10209)  
* syslog_514_udp - started (PID 10206)  
* discard_9_udp - started (PID 10216)  
* discard_9_tcp - started (PID 10215)  
* daytime_13_udp - started (PID 10210)  
* time_37_tcp - started (PID 10207)  
* time_37_udp - started (PID 10208)  
* pop3_110_tcp - started (PID 10197)  
* http_80_tcp - started (PID 10193)  
* ftps_990_tcp - started (PID 10200)  
* tftp_69_udp - started (PID 10201)  
* quotd_17_tcp - started (PID 10217)  
* chargen_19_tcp - started (PID 10219)  
* dummy_1_tcp - started (PID 10221)  
* dummy_1_udp - started (PID 10222)  
* chargen_19_udp - started (PID 10220)  
* quotd_17_udp - started (PID 10218)  
* ftp_21_tcp - started (PID 10199)  
* smtp_25_tcp - started (PID 10195)  
* https_443_tcp - started (PID 10194)  
done.  
Simulation running.  
[]
```

Con la simulazione attiva e funzionante, avvio Wireshark, seleziono “Eth0” e faccio partire lo sniffing dei pacchetti su Eth0.



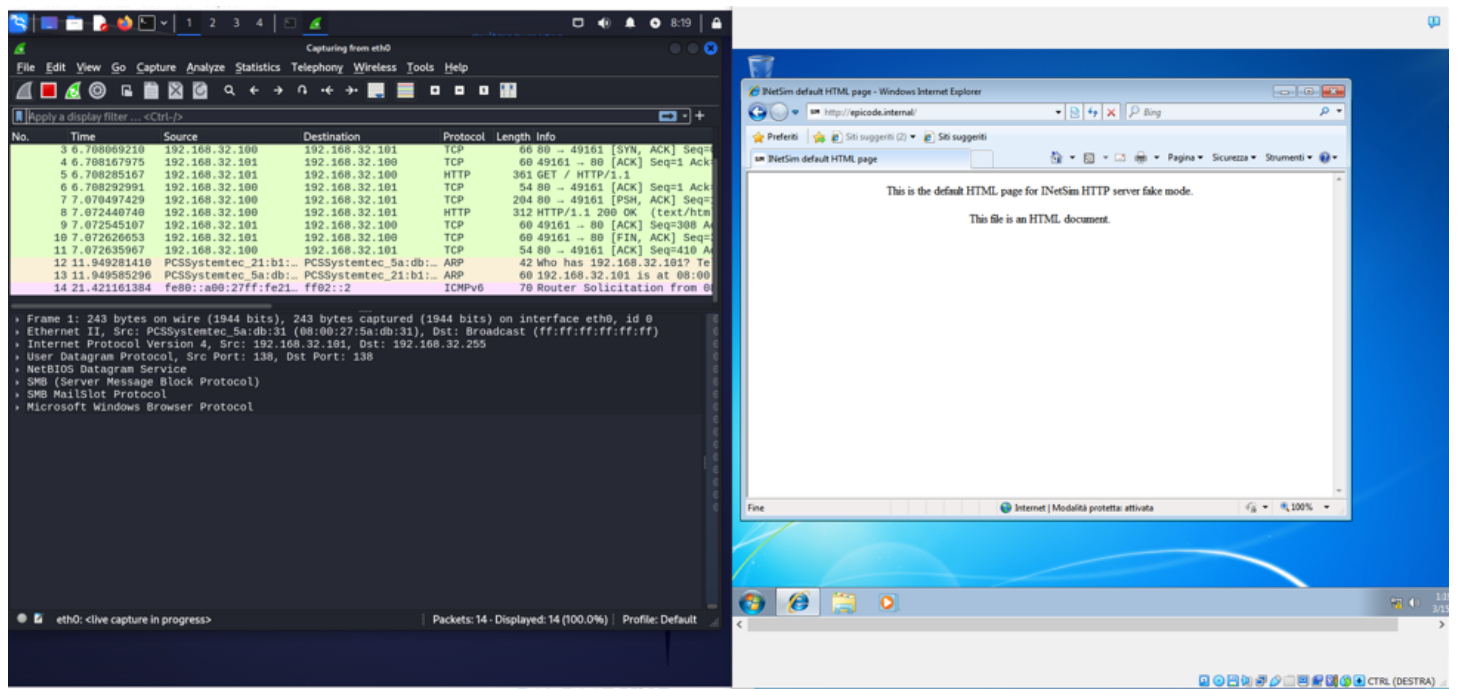
VERSIONE HTTP

Dalla macchina Windows apriamo il browser predefinito (Internet Explorer) e nella barra indirizzo digitiamo il nostro sito web simulato con INetSim



Come da screen, Il DNS risolve correttamente la pagina web convertendo l'indirizzo nell'IP della macchina Kali dove è attivo INetSim.

Nello stesso momento sulla Macchina Kali, Wireshark avrà sniffato i pacchetti che ha generato la richiesta fatta dalla macchina Windows:



Analizzando in dettaglio i pacchetti, si vedono i pacchetti (SYN,ACK) che è il processo di handshake a tre vie che serve a stabilire una connessione tra un client e un server. Il client avvia l'handshake a tre vie per stabilire una connessione TCP con il server.

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
3	6.708069210	192.168.32.100	192.168.32.101	TCP	66	80 → 49161 [SYN, ACK] Seq=
4	6.708167975	192.168.32.101	192.168.32.100	TCP	60	49161 → 80 [ACK] Seq=1 Ack=
5	6.708285167	192.168.32.101	192.168.32.100	HTTP	361	GET / HTTP/1.1
6	6.708292991	192.168.32.100	192.168.32.101	TCP	54	80 → 49161 [ACK] Seq=1 Ack=
7	7.070497429	192.168.32.100	192.168.32.101	TCP	204	80 → 49161 [PSH, ACK] Seq=
8	7.072440740	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
9	7.072545107	192.168.32.101	192.168.32.100	TCP	60	49161 → 80 [ACK] Seq=308 A
10	7.072626653	192.168.32.101	192.168.32.100	TCP	60	49161 → 80 [FIN, ACK] Seq=
11	7.072635967	192.168.32.100	192.168.32.101	TCP	54	80 → 49161 [ACK] Seq=410 A
12	11.949281410	PCSSystemtec_21:b1:...	PCSSystemtec_5a:db:...	ARP	42	Who has 192.168.32.101? Te
13	11.949585296	PCSSystemtec_5a:db:...	PCSSystemtec_21:b1:...	ARP	60	192.168.32.101 is at 08:00
14	21.421161384	fe80::a00:27ff:fe21...	ff02::2	ICMPv6	70	Router Solicitation from 0

▶ Frame 1: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits) on interface eth0, id 0
 ▶ Ethernet II, Src: PCSSystemtec_5a:db:31 (08:00:27:5a:db:31), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 ▶ Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.255
 ▶ User Datagram Protocol, Src Port: 138, Dst Port: 138
 ▶ NetBIOS Datagram Service
 ▶ SMB (Server Message Block Protocol)
 ▶ SMB MailSlot Protocol
 ▶ Microsoft Windows Browser Protocol

eth0: <live capture in progress> | Packets: 14 - Displayed: 14 (100.0%) | Profile: Default

Nel dettaglio, il pacchetto che ci interessa per l'esercizio è il pacchetto con protocollo HTTP:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PCSSystemtec_5a:db:31	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000016749	PCSSystemtec_21:b1:d0	PCSSystemtec_5a:db:31	ARP	42	192.168.32.100 is at 08:00:27:21:b1:d0
3	0.000122557	192.168.32.101	192.168.32.100	DNS	76	Standard query 0x93a9 A epicode.internal
4	0.017840933	192.168.32.100	192.168.32.101	DNS	92	Standard query response 0x93a9 A epicode.internal A 192.168.32.100
5	0.018616459	192.168.32.101	192.168.32.100	TCP	66	49187 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_
6	0.018644507	192.168.32.100	192.168.32.101	TCP	66	80 → 49187 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
7	0.018759231	192.168.32.101	192.168.32.100	TCP	60	49187 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
8	0.018893254	192.168.32.101	192.168.32.100	HTTP	361	GET / HTTP/1.1
9	0.018901324	192.168.32.100	192.168.32.101	TCP	54	80 → 49187 [ACK] Seq=1 Ack=308 Win=64128 Len=0
10	0.035518605	192.168.32.100	192.168.32.101	TCP	204	80 → 49187 [PSH, ACK] Seq=1 Ack=308 Win=64128 Len=150 [TC
11	0.037591068	192.168.32.100	192.168.32.101	HTTP	243	HTTP/1.1 200 OK (text/html)
12	0.037730631	192.168.32.101	192.168.32.100	TCP	60	49187 → 80 [ACK] Seq=308 Ack=341 Win=65360 Len=0
13	0.037839246	192.168.32.101	192.168.32.100	TCP	60	49187 → 80 [FIN, ACK] Seq=308 Ack=341 Win=65360 Len=0

Frame 8: 361 bytes on wire (2888 bits), 361 bytes captured (2888 bits) on interface eth0, id 0
 Ethernet II, Src: PCSSystemtec_5a:db:31 (08:00:27:5a:db:31), Dst: PCSSystemtec_21:b1:d0 (08:00:27:21:b1:d0)
 Destination: PCSSystemtec_21:b1:d0 (08:00:27:21:b1:d0)
 Source: PCSSystemtec_5a:db:31 (08:00:27:5a:db:31)
 Type: IPv4 (0x0800)
 Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 347
 Identification: 0x0130 (304)
 010. = Flags: 0x2, Don't fragment
 ...0 0000 0000 0000 = Fragment Offset: 0
 Time to Live: 128
 Protocol: TCP (6)
 Header Checksum: 0x3653 [validation disabled]
 [Header checksum status: Unverified]
 Source Address: 192.168.32.101
 Destination Address: 192.168.32.100
 Transmission Control Protocol, Src Port: 49187, Dst Port: 80, Seq: 1, Ack: 1, Len: 307
 Hypertext Transfer Protocol

Dove su Wireshark possiamo trovare le seguenti informazioni:

Mac adress delle due macchine virtuali

Source (**08:00:27:5a:db:31**) corrispondente alla VM Windows7

Desination (**08:00:27:21:b1:d0**) corrispondente alla macchina Kali [Ether]

C:\Windows\system32\cmd.exe	
Scheda Ethernet Connessione alla rete locale (LAN):	
Suffisso DNS specifico per connessione:	
Descrizione:	Scheda desktop Intel(R) PRO/1000 MT
Indirizzo fisico:	08-00-27-5A-DB-31
DHCP abilitato:	No
Configurazione automatica abilitata:	Sì
Indirizzo IPv6 locale rispetto al collegamento:	fe80::3855:e071:1e2d:7861
11<Preferenziale>	
Indirizzo IPv4:	192.168.32.101<Preferenziale>
Subnet mask:	255.255.255.0
Gateway predefinito:	192.168.32.1
IAID DHCPv6:	235405351
DUID Client DHCPv6:	00-01-00-01-2D-6C-25-47-08-00-27-5A-DB-3
Server DNS:	192.168.32.100
NetBIOS su TCP/IP:	Attivato
Scheda Tunnel isatap.{06786796-DDAF-4E84-89CA-BD2F5B88D12}:	
Stato supporto:	Supporto disconnesso
Suffisso DNS specifico per connessione:	
Descrizione:	Microsoft ISATAP Adapter #2
Indirizzo fisico:	00-00-00-00-00-00-00-E0
DHCP abilitato:	No
Configurazione automatica abilitata:	Sì

```

$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.32.100 netmask 255.255.255.0 broadcast 192.168.32.255
    inet6 fe80::a00:27ff:fe21:b1d0 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)
    RX packets 434 bytes 47754 (46.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 237 bytes 121324 (118.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 54 bytes 4336 (4.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 54 bytes 4336 (4.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Catturato sul ETH0 (Kali)

```

Frame 13: 472 bytes on wire (3776 bits), 472 bytes captured (3776 bits)
  Section number: 1
  ▶ Interface id: 0 (eth0)
    Encapsulation type: Ethernet (1)

```

IP delle macchine:

Si può notare nel pacchetto sniffato l'indirizzo IP delle due macchine virtuali.

```

[Header checksum status: Unverified]
Source Address: 192.168.32.101
Destination Address: 192.168.32.100

```

Tipo di protocollo:

Si nota che il Pacchetto è con protocollo HTTP, E che è stata usata la porta 80

```

[Protocols in frame: eth:ethertype:ip:tcp:http]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]

```

Contenuto del pacchetto:

Si nota che il contenuto sniffato di risposta alla domanda del pacchetto “get” è una stringa testuale in chiaro e un’immagine in .gif. Non è presente incryptazione essendo l’HTTP un protocollo che non la prevede quindi Header e Payload sono totalmente in chiaro.

[Request URI: http://epicode.internal/]
File Data: 189 bytes
Line-based text data: text/html (13 lines)
<html>\n
 <head>\n
 <title>\n
 INetSim test page\n
 </title>\n
 </head>\n
 <body>\n
 \n
 <h3>This is the INetSim real-mode test page...</h3>\n
 \n
 \n
 </body>\n
</html>\n

00a0 68 65 61 64 3e 0a 20 20 20 20 3c 74 69 74 6c 65 head>.<title
00b0 3e 0a 20 20 20 20 20 20 49 4e 65 74 53 69 6d 20 >.< INetSim
00c0 74 65 73 74 20 70 61 67 65 0a 20 20 20 20 3c 2f test pag e.</
00d0 74 69 74 6c 65 3e 0a 20 20 3c 2f 68 65 61 64 3e title>.</head>
00e0 0a 20 20 3c 62 6f 64 79 3e 0a 0a 20 20 20 20 3c .<body >.<
00f0 68 33 3e 54 68 69 73 20 69 73 20 74 68 65 20 49 h3>This is the I
0100 4e 65 74 53 69 6d 20 72 65 61 6c 2d 6d 6f 64 65 NetSim r eal-mode
0110 20 74 65 73 74 20 70 61 67 65 2e 2e 2e 3c 2f 68 test pa ge...</h
0120 33 3e 0a 0a 20 20 20 20 3c 69 6d 67 20 73 72 63 3>.<img src
0130 3d 22 69 6e 74 65 72 6e 65 74 2e 67 69 66 22 3e ="intern et.gif">
0140 0a 20 20 3c 2f 62 6f 64 79 3e 0a 3c 2f 68 74 6d .</bod y>.</htm
0150 6c 3e 0a l>.

Frame (243 bytes) Reassembled TCP (339 bytes)

VERSIONE HTTPS

Dalla macchina Windows apriamo il browser predefinito (Internet Explorer) e nella barra indirizzo digitiamo il nostro sito web simulato con INetSim specificando il protocollo HTTPS
HTTPS://Epicode.internal

No. Time Source Destination Protocol Length Info

1 0.000000000 PCSSystemtec_5a:db:... Broadcast ARP 60 Who has 192.168.32.100? Tell 192.168.32.101

2 0.000013135 PCSSystemtec_21:b1:... PCSSystemtec_5a:db:... ARP 42 192.168.32.100 is at 08:00:27:21:b1:d0

3 0.000107748 192.168.32.101 192.168.32.100 TCP 66 49207 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM

4 0.000126546 192.168.32.100 192.168.32.101 TCP 66 443 → 49207 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128

5 0.000205082 192.168.32.101 192.168.32.100 TCP 60 49207 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0

6 0.002983699 192.168.32.101 192.168.32.100 TLSv1 183 Client Hello (SNI=epicode.internal)

7 0.002992873 192.168.32.100 192.168.32.101 TCP 54 443 → 49207 [ACK] Seq=1 Ack=130 Win=64128 Len=0

8 0.436457765 192.168.32.100 192.168.32.101 TLSv1 1373 Server Hello, Certificate, Server Key Exchange, Server Hello Done

9 0.440865566 192.168.32.101 192.168.32.100 TLSv1 188 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

10 0.440895410 192.168.32.100 192.168.32.101 TCP 54 443 → 49207 [ACK] Seq=1320 Ack=264 Win=64128 Len=0

11 0.441418119 192.168.32.100 192.168.32.101 TLSv1 113 Change Cipher Spec, Encrypted Handshake Message

12 0.455386768 PCSSystemtec_5a:db:... Broadcast ARP 60 Who has 192.168.32.1? Tell 192.168.32.101

13 0.640429399 192.168.32.101 192.168.32.100 TCP 60 49207 → 443 [ACK] Seq=264 Ack=1379 Win=64320 Len=0

14 1.422152279 PCSSystemtec_5a:db:... Broadcast ARP 60 Who has 192.168.32.1? Tell 192.168.32.101

15 2.422730733 PCSSystemtec_5a:db:... Broadcast ARP 60 Who has 192.168.32.1? Tell 192.168.32.101

16 3.599457442 fe80::3855:e071:1e2... ff02::1:3 LLMNR 84 Standard query 0x25b8 A wpad

17 3.599498991 192.168.32.101 224.0.0.252 LLMNR 64 Standard query 0x25b8 A wpad

18 3.705279029 fe80::3855:e071:1e2... ff02::1:3 LLMNR 84 Standard query 0x25b8 A wpad

19 3.705279369 192.168.32.101 224.0.0.252 LLMNR 64 Standard query 0x25b8 A wpad

20 3.908005294 192.168.32.101 192.168.32.255 NBNS 92 Name query NB WPAD<00>

21 4.658660508 192.168.32.101 192.168.32.255 NBNS 92 Name query NB WPAD<00>

22 5.408229700 192.168.32.101 192.168.32.255 NBNS 92 Name query NB WPAD<00>

23 6.161726847 PCSSystemtec_5a:db:... Broadcast ARP 60 Who has 192.168.32.1? Tell 192.168.32.101

24 6.925116630 PCSSystemtec_5a:db:... Broadcast ARP 60 Who has 192.168.32.1? Tell 192.168.32.101

25 7.926042377 PCSSystemtec_5a:db:... Broadcast ARP 60 Who has 192.168.32.1? Tell 192.168.32.101

26 9.304140703 fe80::3855:e071:1e2... ff02::1:3 LLMNR 84 Standard query 0xf115 A wpad

27 9.304183746 192.168.32.101 224.0.0.252 LLMNR 64 Standard query 0xf115 A wpad

28 9.410189770 fe80::3855:e071:1e2... ff02::1:3 LLMNR 84 Standard query 0xf115 A wpad

29 9.410189980 192.168.32.101 224.0.0.252 LLMNR 64 Standard query 0xf115 A wpad

30 9.614529164 192.168.32.101 192.168.32.255 NBNS 92 Name query NB WPAD<00>

31 10.364144442 192.168.32.101 192.168.32.255 NBNS 92 Name query NB WPAD<00>

32 11.114241799 192.168.32.101 192.168.32.255 NBNS 92 Name query NB WPAD<00>

33 12.883067922 192.168.32.101 192.168.32.100 TCP 60 49207 → 443 [FIN, ACK] Seq=264 Ack=1379 Win=64320 Len=0

34 12.883159793 192.168.32.100 192.168.32.101 TLSv1 91 Encrypted Alert

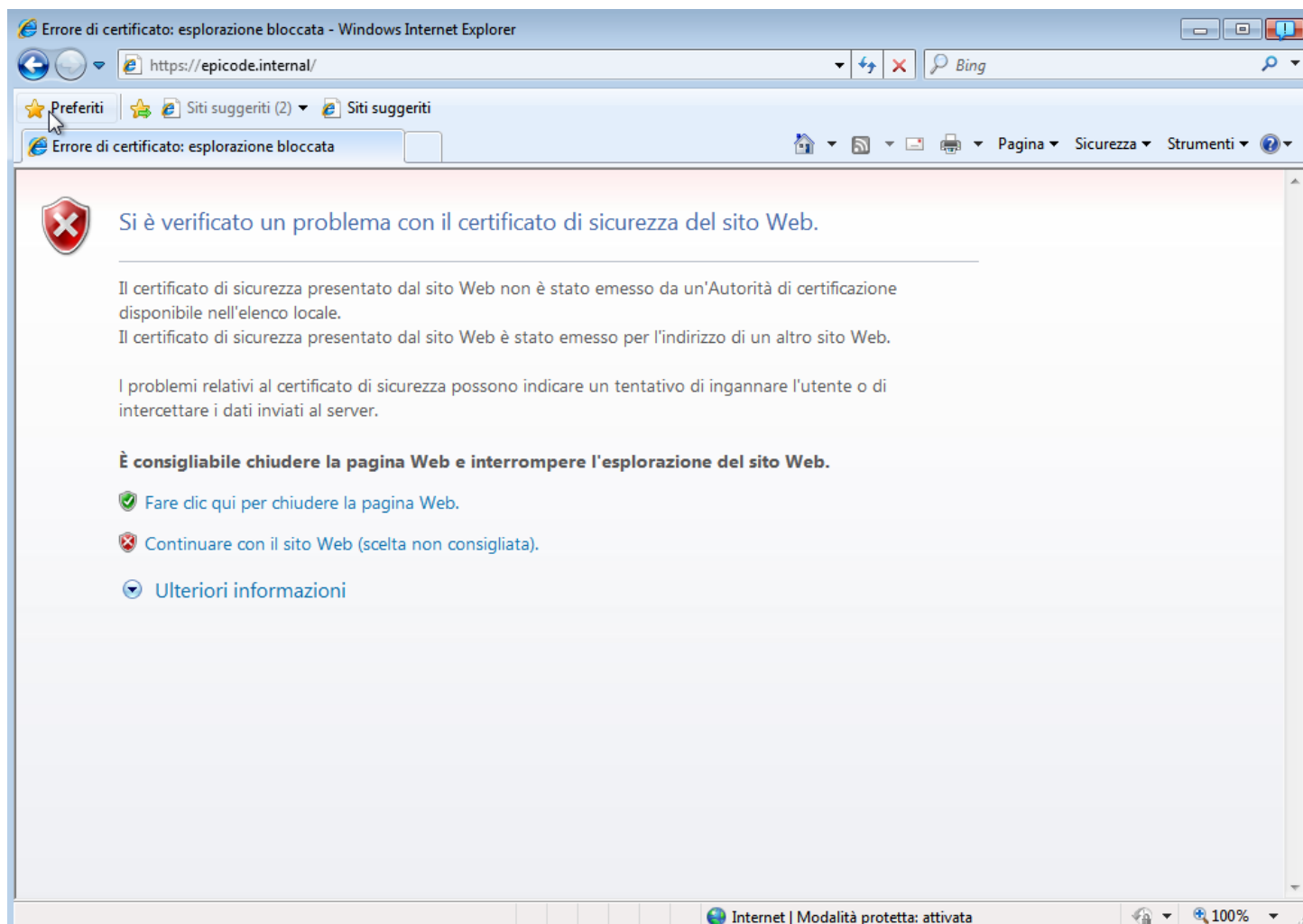
35 12.883282134 192.168.32.101 192.168.32.100 TCP 60 49207 → 443 [RST, ACK] Seq=265 Ack=1416 Win=0 Len=0

36 18.618469286 fe80::3855:e071:1e2... ff02::1:2 DHCPv6 148 Solicit XID: 0xacf1f2 CID: 000100012d6c25470800275adb31

▼ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface
Section number: 1
Interface id: 0 (eth0)
Encapsulation type: Ethernet (1)
Arrival Time: Mar 15, 2024 14:19:59.631394769 FDT

0000 ff ff ff ff ff ff 08 00 27 5a db 31 08 06 00 01 'Z 1.....
0010 08 00 06 04 00 01 08 00 27 5a db 31 c0 a8 20 65 'Z 1... e
0020 00 00 00 00 00 00 c0 a8 20 64 00 00 00 00 00 's d.....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Succederà che il browser di Windows7 blocca la renderizzazione della pagina a causa di un certificato non valido. Già questo controllo fa notare delle differenze sostanziali tra le due tipologie di protocolli (HTTP vs HTTPS)



Forziamo comunque l'apertura della nostra fake pagina web premendo su "Continuare con il sito Web" e cambiamo la visuale sulla macchina Kali

Frame 46: 215 bytes on wire (1720 bits), 215 bytes captured (1720 bits) on interface eth0, id 0

Section number: 1

Interface id: 0 (eth0)

Encapsulation type: Ethernet (1)

Arrival Time: Mar 15, 2024 14:26:28.139842682 EDT

UTC Arrival Time: Mar 15, 2024 18:26:28.139842682 UTC

Epoch Arrival Time: 1710527188.139842682

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.000179141 seconds]

[Time delta from previous displayed frame: 0.000179141 seconds]

[Time since reference or first frame: 12.466647234 seconds]

Frame Number: 46

Frame Length: 215 bytes (1720 bits)

Capture Length: 215 bytes (1720 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ethertype:ip:tcp:tls]

[Coloring Rule Name: TCP]

[Coloring Rule String: tcp]

Ethernet II, Src: PCSSystemtec_5a:db:31 (08:00:27:5a:db:31), Dst: PCSSystemtec_21:b1:d0 (08:00:27:21:b1:d0)

Destination: PCSSystemtec_21:b1:d0 (08:00:27:21:b1:d0)

Interface id (frame.interface_id)

Packets: 60 - Displayed: 60 (100.0%) - Dropped: 0 (0.0%)

Risulteranno dei pacchetti sniffati. Notiamo subito la differenza di porta in 443 (HTTPS) e il protocollo TLSv1 (Trasport Layer Security) l'evoluzione del protocollo SSL, che sono protocolli di incryptazione.

Dettagli sul pacchetto sniffato HTTPS

Mac address e IP delle due macchine virtuali

Come in HTTP, troviamo i due Mac Address:

Source (**08:00:27:5a:db:31**) corrispondente alla VM Windows7

Desination (**08:00:27:21:b1:d0**) corrispondente alla macchina Kali

E i loro relativi IP

```
▶ Frame 46: 215 bytes on wire (1720 bits), 215 bytes captured (1720 bits) on interface eth0, id 0
▼ Ethernet II, Src: PCSSystemtec_5a:db:31 (08:00:27:5a:db:31), Dst: PCSSystemtec_21:b1:d0 (08:00:27:21:b1:d0)
  ▶ Destination: PCSSystemtec_21:b1:d0 (08:00:27:21:b1:d0)
  ▼ Source: PCSSystemtec_5a:db:31 (08:00:27:5a:db:31)
    Address: PCSSystemtec_5a:db:31 (08:00:27:5a:db:31)
    .... ..0. .... = LG bit: Globally unique address (factory default)
    .... ..0. .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
  ▼ Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 201
    Identification: 0x0202 (514)
    ▶ 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x3613 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.32.101
```

Internet Protocol Version 4 (ip), 20 bytes

Tipo di protocollo:

Si nota a differenza dell'HTTP, l'utilizzo del TLSv1 che provvede a criptare i dati del pacchetto. La porta utilizzata è la default 443

```
Destination Address: 192.168.32.100
Transmission Control Protocol, Src Port: 49211, Dst Port: 443, Seq: 1, Ack: 1, Len: 161
Source Port: 49211
Destination Port: 443
[Stream index: 1]
▶ [Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 161]
Sequence Number: 1 (relative sequence number)
```

Contenuto del pacchetto:

Come si nota, il pacchetto risulta tutto cifrato

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PCSSystemtec_5a:db:...	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000011205	PCSSystemtec_21:b1:...	PCSSystemtec_5a:db:...	ARP	42	192.168.32.100 is at 08:00:27:21:b1:d0
3	0.000093357	192.168.32.101	192.168.32.100	DNS	76	Standard query 0x1828 A epicode.internal
4	0.212163755	192.168.32.100	192.168.32.101	DNS	92	Standard query response 0x1828 A epicode.internal A 192.168.32.100
5	0.212678136	192.168.32.101	192.168.32.100	TCP	66	49212 -> 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
6	0.212697955	192.168.32.100	192.168.32.101	TCP	66	443 -> 49212 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
7	0.212790536	192.168.32.101	192.168.32.100	TCP	66	49212 -> 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
8	0.212991733	192.168.32.101	192.168.32.100	TLSv1	215	Client Hello (SSL epoch=0, epicode.internal)
9	0.213002051	192.168.32.100	192.168.32.101	TCP	54	443 -> 49212 [ACK] Seq=1 Ack=162 Win=64128 Len=0
10	0.336704962	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exchange, Server Hello Done
11	0.341175628	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12	0.341193666	192.168.32.100	192.168.32.101	TCP	54	443 -> 49212 [ACK] Seq=1320 Ack=296 Win=64128 Len=0
13	0.341617906	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
14	0.345948123	PCSSystemtec_5a:db:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
15	0.545333875	192.168.32.101	192.168.32.100	TCP	60	49212 -> 443 [ACK] Seq=296 Ack=1379 Win=64320 Len=0
16	1.201695518	PCSSystemtec_5a:db:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
17	2.202495186	PCSSystemtec_5a:db:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
18	3.489741000	fe80::3855:e071:1e2...	ff02::1:3	LLMNR	84	Standard query 0x438f A wpad
19	3.489791933	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0x438f A wpad
20	3.593448668	fe80::3855:e071:1e2...	ff02::1:3	LLMNR	84	Standard query 0x438f A wpad
21	3.593449048	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0x438f A wpad
22	3.797345213	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
23	4.546683526	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
24	5.297160248	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
25	6.051633316	PCSSystemtec_5a:db:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
26	6.704361249	PCSSystemtec_5a:db:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
27	7.705198786	PCSSystemtec_5a:db:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101

Sequence Number: 1 (relative sequence number)

Sequence Number (raw): 1795131447

[Next Sequence Number: 162 (relative sequence number)]

Acknowledgment Number: 1 (relative ack number)

Acknowledgment number (raw): 3658153580

0101 = Header Length: 20 bytes (5)

Flags: 0x010 (PSH, ACK)

Window: 16425

[Calculated window size: 65700]

[Window size scaling factor: 4]

Checksum: 0x3b55 [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

[Timestamps]

[SEQ/ACK analysis]

TCP payload (161 bytes)

0000 08 00 27 21 b1 d0 08 00 27 5a db 31 08 00 45 00 ...!....'Z 1...E

0010 00 c9 02 1c 40 00 80 06 35 f9 c0 a8 20 65 c0 a8 ...@... 6...e

0020 20 64 c0 3c 01 bb 6a ff 88 37 da 0a fe 6c 50 18 d<...j...7...LP

0030 40 29 3b 55 00 00 16 03 01 00 9c 01 00 00 98 03 ...@... ..F...BA

0040 01 65 f4 ab 7e 38 4d 8b f2 5d cd da 69 7c c5 eb ...g'... kI l S...

0050 01 0b d7 06 d1 b2 61 0a 73 0d 5f ce fe 7f e8 b8 ...z 0...XG3m...

0060 06 20 72 b9 01 72 94 40 02 bd d7 a1 35 58 ef 39 ...-n7... ON V'[{

0070 bd 54 a1 47 31 58 b8 f3 01 fb f9 48 ad c6 3e aa ...A3g[... /...;

0080 b4 5c 00 18 00 2f 00 35 00 05 00 0a c0 13 c0 14 ...C... ..0...

0090 c0 09 c0 0a 00 32 00 38 00 13 00 04 01 00 00 37 ...SV... X { ...1

00a0 ff 01 00 01 00 00 00 00 15 00 13 00 00 10 65 70 ...Q... A... u...

00b0 69 63 6f 64 65 2e 69 6e 74 65 72 6e 61 6c 00 05 ...-/1... o }

00c0 00 05 01 00 00 00 00 00 0a 00 06 00 04 00 17 00

00d0 18 00 0b 00 02 01 00

e come si vede il pacchetto continene il TLS che previene la visualizzazione del contenuto del pacchetto in chiaro, ma risultano solo stringe di testo “casuali”

Urgent Pointer: 0

[Timestamps]

[SEQ/ACK analysis]

TCP payload (134 bytes)

Transport Layer Security

↳ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange

Content Type: Handshake (22)

Version: TLS 1.0 (0x0301)

Length: 70

↳ Handshake Protocol: Client Key Exchange

↳ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec

Content Type: Change Cipher Spec (20)

Version: TLS 1.0 (0x0301)

Length: 1

Change Cipher Spec Message

↳ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message

0000 08 00 27 21 b1 d0 08 00 27 5a db 31 08 00 45 00 ...!....'Z 1...E

0010 00 ae 02 1d 40 00 80 06 36 13 c0 a8 20 65 c0 a8 ...@... 6...e

0020 20 64 c0 3c 01 bb 6a ff 88 d8 da 0b 03 93 50 18 d<...j...7...LP

0030 3e df 7a f6 00 00 16 03 01 00 46 10 00 00 42 41 >z... ..F...BA

0040 04 67 27 60 af be eb 0d 6b 49 c9 6c e2 53 ec ed ...g'... kI l S...

0050 bb d3 f3 d1 d5 7a 01 4f 2c a4 58 47 33 6d 0d 83 ...z 0...XG3m...

0060 b2 0e 2b 2d 0e 37 96 5d 4f 4e 93 56 22 7b 97 da ...-n7... ON V'[{

0070 d6 5e 33 67 7b 96 f7 bc 2f 0b a2 ef 1f b4 3b c9 ...A3g[... /...;

0080 43 14 03 01 00 01 01 16 03 01 00 30 9a a7 cf 99 ...C... ..0...

0090 8a 24 56 b6 e0 dd 07 58 b2 7b 9e 1d 9f dc 31 ec ...SV... X { ...1

00a0 1d b7 51 ca f2 cd 0e 82 41 b2 99 89 da 75 e7 ce ...Q... A... u...

00b0 06 ca 8a b1 2f 31 bc 01 f5 6f 9f 7d ...-/1... o }

Packets: 43 · Displayed: 43 (100.0%) · Dropped: 0 (0.0%)

CONCLUSIONI:

HTTP (Hypertext Transfer Protocol) è il protocollo di comunicazione utilizzato per trasferire dati tra un browser web e un sito web. È standard, ma non crittografa i dati, rendendoli vulnerabili agli attacchi.

Un esempio è quando un utente visita una pagina web tramite il proprio browser. Il browser invia una richiesta HTTP al server web per ottenere la pagina richiesta, e il server risponde inviando il contenuto della pagina. Questo avviene senza crittografia, quindi i dati sono inviati in chiaro e potrebbero essere intercettati da terze parti.

HTTPS (HTTP Secure) aggiunge un livello di crittografia SSL o TLS, garantendo una comunicazione sicura e proteggendo i dati degli utenti durante la navigazione online.

Un esempio è quando un utente si connette al proprio home banking. Durante questa connessione, il browser e il server della banca negoziano un protocollo TLS per crittografare tutti i dati scambiati, come informazioni di accesso e transazioni finanziarie, proteggendo così le informazioni sensibili degli utenti durante il trasferimento.