

# Progetto S1/L5

Richiesta di una risorsa tramite DNS, Differenze tra HTTP e HTTPS



```
kali@kali: ~  
GNU nano 7.2 /etc/network/interfaces  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet static  
address 192.168.32.100/24  
gateway 192.168.32.1
```

Help Write Out Where Is Cut Execute  
Exit Read File Replace Paste Justify

Per iniziare, ho cambiato gli indirizzi IP delle due macchine virtuali, riavviato Kali e inserito il suo IP in “server DNS preferito” all’interno del pannello di configurazione di Windows

Proprietà - Protocollo Internet versione 4 (TCP/IPv4)

Generale

È possibile ottenere l'assegnazione automatica delle impostazioni IP se la rete supporta tale caratteristica. In caso contrario, sarà necessario richiedere all'amministratore di rete le impostazioni IP corrette.

☐ Ottieni automaticamente un indirizzo IP

☒ Utilizza il seguente indirizzo IP:

Indirizzo IP: 192 . 168 . 32 . 101

Subnet mask: 255 . 255 . 255 . 0

Gateway predefinito: 192 . 168 . 32 . 1

☐ Ottieni indirizzo server DNS automaticamente

☒ Utilizza i seguenti indirizzi server DNS:

Server DNS preferito: 192 . 168 . 32 . 100

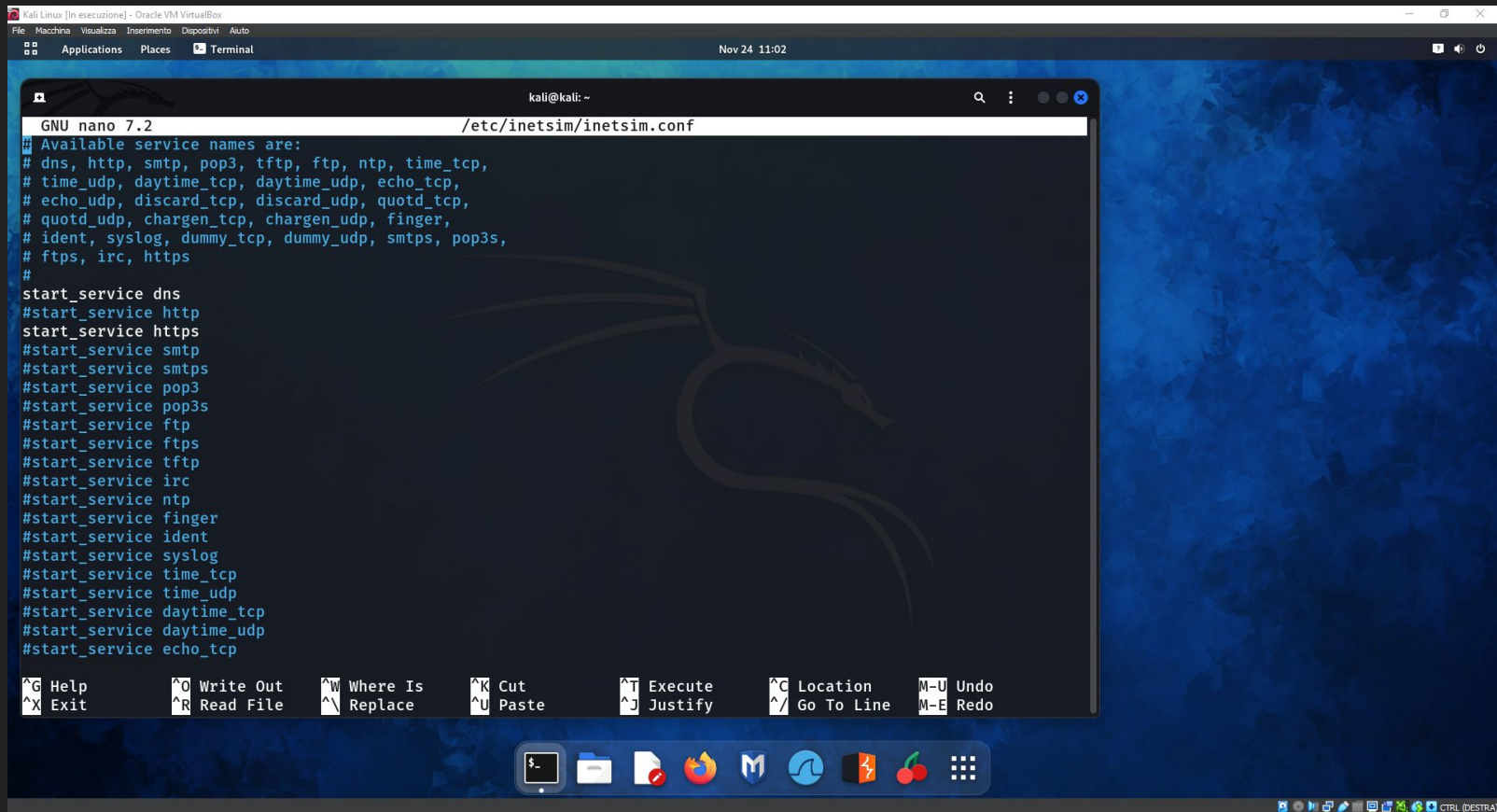
Server DNS alternativo: . . .

☐ Convalida impostazioni all'uscita

Avanzate...

OK Annulla

Come da consegna ho modificato i file di InetSim per attivare i servizi di DNS e HTTPS



```
Kali Linux [in esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Auto
Applications Places Terminal
Nov 24 11:02

kali@kali: ~
GNU nano 7.2 /etc/inetsim/inetsim.conf
# 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
start_service smtp
start_service smtps
start_service pop3
start_service pop3s
start_service ftp
start_service ftps
start_service tftp
start_service irc
start_service ntp
start_service finger
start_service ident
start_service syslog
start_service time_tcp
start_service time_udp
start_service daytime_tcp
start_service daytime_udp
start_service echo_tcp

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo

$ _
```

Kali Linux [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

Applications Places Terminal

Nov 24 11:07

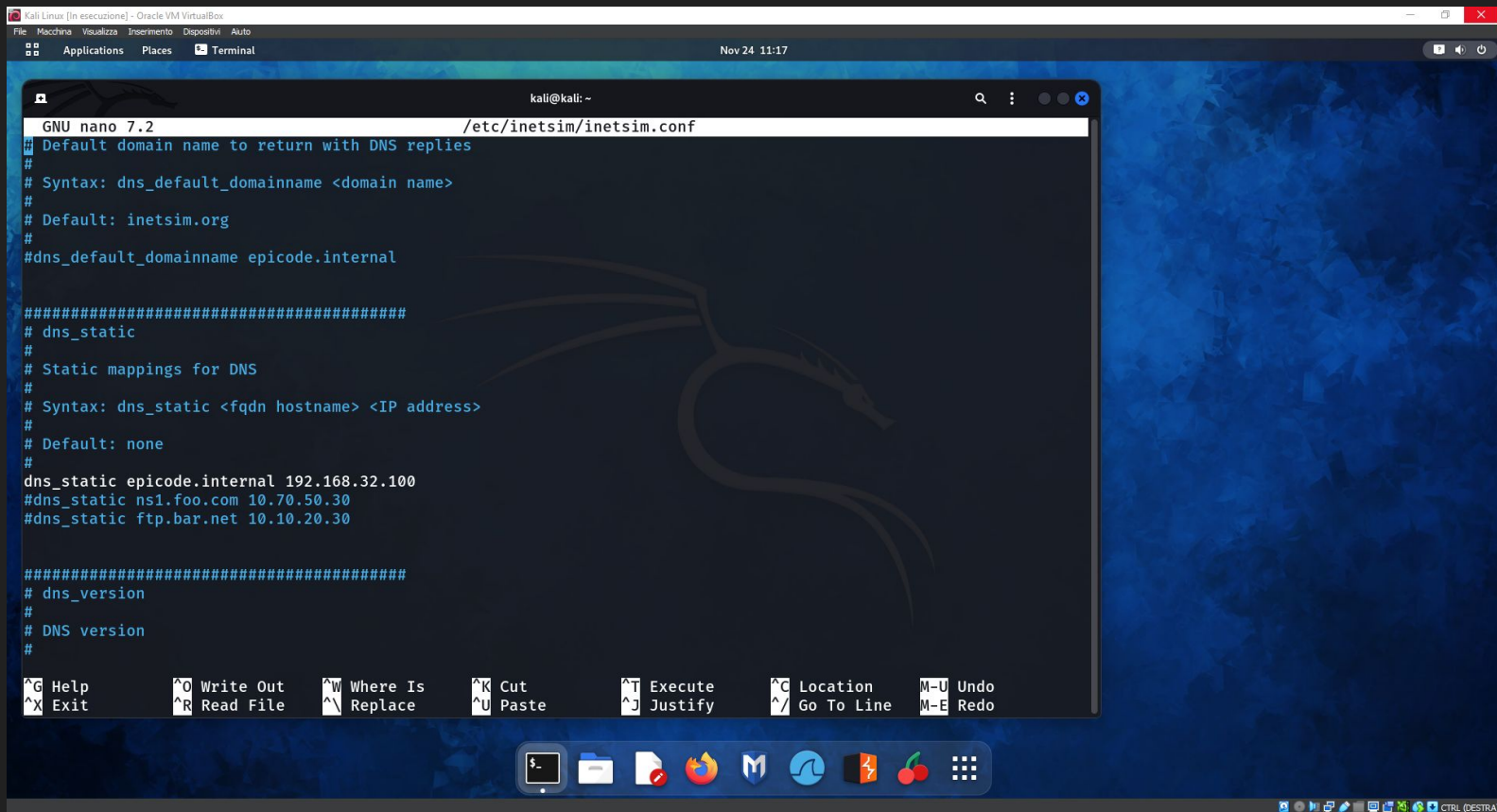
```
kali@kali: ~  
GNU nano 7.2 /etc/inetsim/inetsim.conf  
#start_service echo_udp  
#start_service discard_tcp  
#start_service discard_udp  
#start_service quotd_tcp  
#start_service quotd_udp  
#start_service chargen_tcp  
#start_service chargen_udp  
#start_service dummy_tcp  
#start_service dummy_udp  
  
#####  
# service_bind_address  
#  
# IP address to bind services to  
#  
# Syntax: service_bind_address <IP address>  
#  
# Default: 192.168.32.100  
service_bind_address 192.168.32.100  
  
#####  
# service_run_as_user  
#  
# User to run services  
#  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo  
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_/ Go To Line  M-E Redo
```

Nov 24 11:07

CTRL (DESTRA)

Modificando il campo “service bind address” ho reso l’ip della macchina Kali lo stesso del servizio DNS

Dopodiché ho  
modificato il  
campo “dns  
static”  
inserendo  
l’hostname  
dell’indirizzo ip  
della vm Kali



The screenshot shows a Kali Linux virtual machine window titled "Kali Linux [in esecuzione] - Oracle VM VirtualBox". The desktop background is blue with a dragon logo. A terminal window is open, displaying the nano 7.2 editor editing the file /etc/inetsim/inetsim.conf. The file content is as follows:

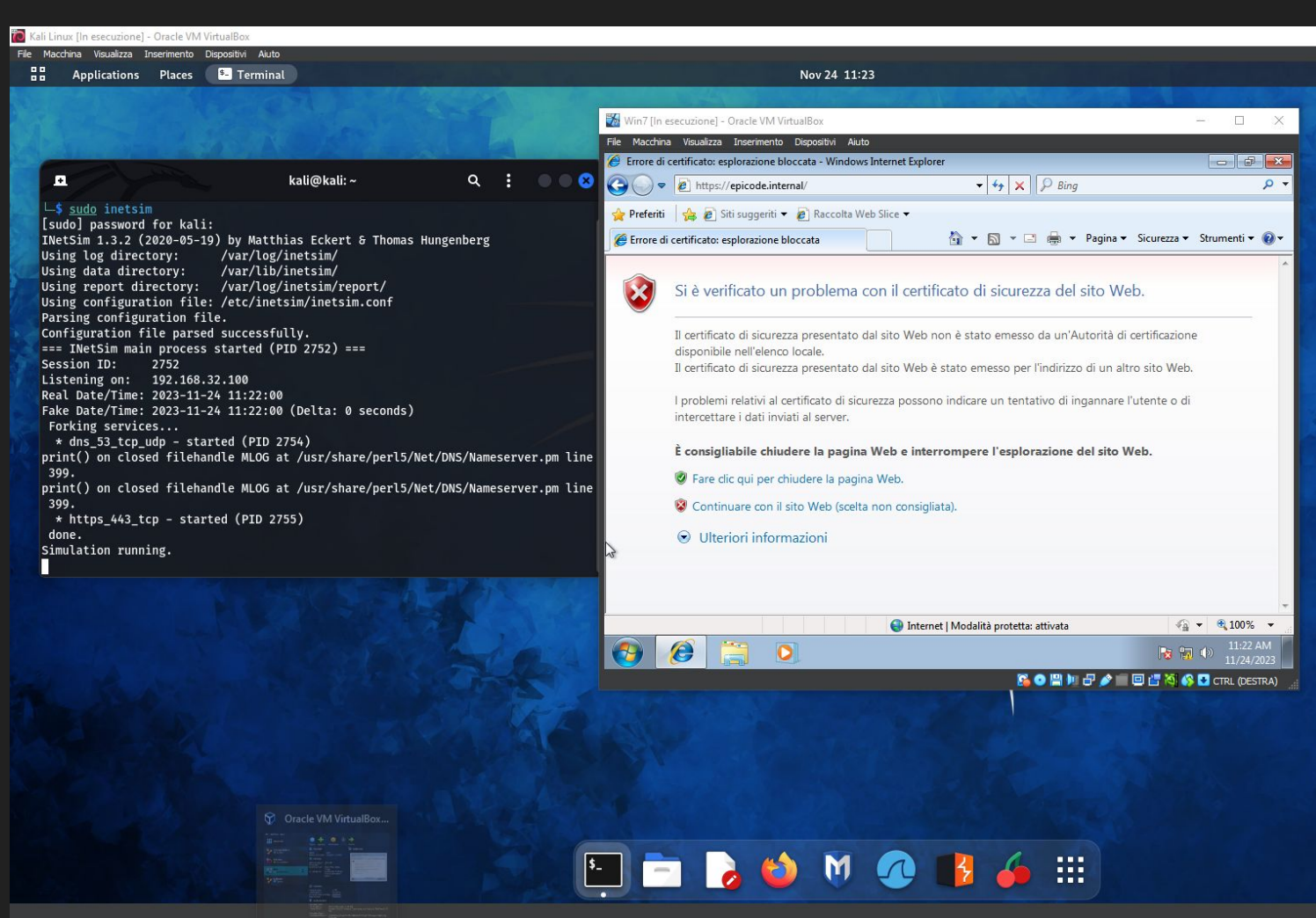
```
GNU nano 7.2 /etc/inetsim/inetsim.conf
# Default domain name to return with DNS replies
#
# Syntax: dns_default_domainname <domain name>
#
# Default: inetsim.org
#
#dns_default_domainname epicode.internal

#####
# 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

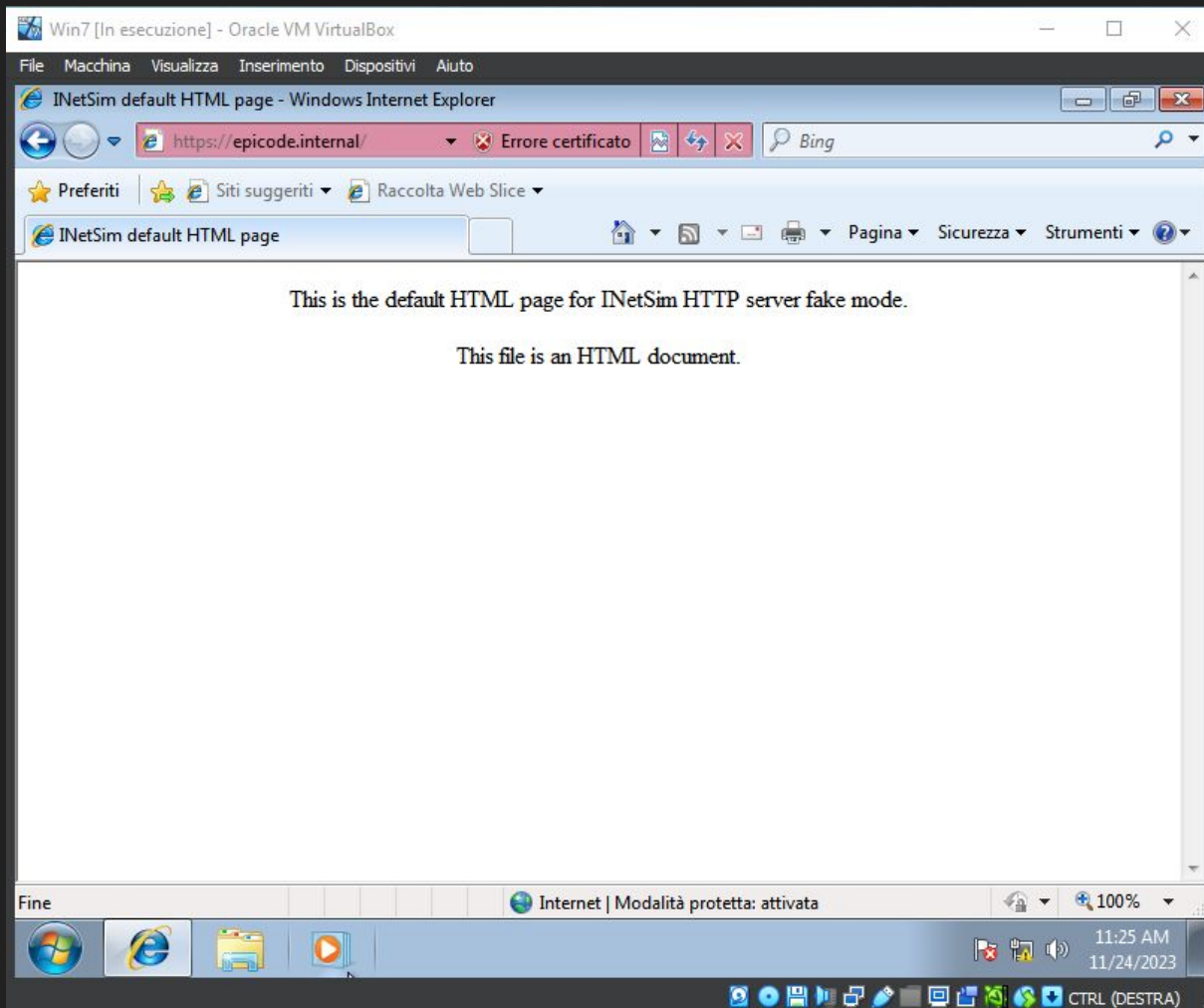
#####
# dns_version
#
# DNS version
#
```

At the bottom of the terminal window, there is a menu bar with the following options: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, M-U Undo, ^X Exit, ^R Read File, ^\_ Replace, ^U Paste, ^J Justify, ^\_/ Go To Line, M-E Redo. The bottom of the screen shows the Kali Linux desktop environment with various application icons in the taskbar and system tray.





Avviando InetSim e tentando il collegamento a epicode.internal riceviamo un messaggio di errore relativo alla sicurezza della pagina, dato che essa non ha un certificato https valido, nonostante ciò continuando la pagina si presenta correttamente, come si può vedere nella pagina seguente



Kali Linux [in esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

Applications Places wireshark

Nov 24 11:34

\*eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http || tcp/

No.	Time	Source	Destination	Protocol	Len	Info
49	68.879644030	192.168.32.100	192.168.32.101	TLSv1	1	Change Cipher Spec, Encrypted Handshake Message
51	69.085724025	192.168.32.100	192.168.32.101	TCP	1	[TCP Retransmission] 443 -> 49207 [PSH, ACK] Seq=1315 Ack=291 Win=64128 Len=59
52	69.086213989	192.168.32.101	192.168.32.101	TCP	66	49207 -> 443 [ACK] Seq=291 Ack=1374 Win=64324 Len=0 SLE=1315 SRE=1374
74	84.490458725	192.168.32.101	192.168.32.101	TCP	60	49207 -> 443 [FIN, ACK] Seq=291 Ack=1374 Win=64324 Len=0
75	84.490686242	192.168.32.100	192.168.32.101	TLSv1	91	Encrypted Alert
76	84.491434585	192.168.32.101	192.168.32.101	TCP	60	49207 -> 443 [RST, ACK] Seq=292 Ack=1411 Win=0 Len=0
77	84.492467913	192.168.32.101	192.168.32.101	TCP	66	49209 -> 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
78	84.492497489	192.168.32.100	192.168.32.101	TCP	66	443 -> 49209 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
79	84.493381063	192.168.32.101	192.168.32.101	TCP	60	49209 -> 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
80	84.493381063	192.168.32.101	192.168.32.101	TLSv1	2	Client Hello
81	84.493422363	192.168.32.100	192.168.32.101	TCP	54	443 -> 49209 [ACK] Seq=1 Ack=157 Win=64128 Len=0
82	84.535670689	192.168.32.100	192.168.32.101	TLSv1	1	Server Hello, Certificate, Server Key Exchange, Server Hello Done
83	84.542754187	192.168.32.101	192.168.32.101	TLSv1	1	Client Key Exchange, change Cipher Spec, Encrypted Handshake Message
84	84.543387624	192.168.32.100	192.168.32.101	TLSv1	1	Change Cipher Spec, Encrypted Handshake Message
85	84.550929858	192.168.32.101	192.168.32.101	TLSv1	5	Application Data
86	84.565349072	192.168.32.100	192.168.32.101	TLSv1	2	Application Data
87	84.567719243	192.168.32.100	192.168.32.101	TLSv1	3	Application Data, Encrypted Alert
88	84.568423732	192.168.32.101	192.168.32.101	TCP	60	49209 -> 443 [ACK] Seq=744 Ack=1886 Win=65700 Len=0
89	84.568810682	192.168.32.101	192.168.32.101	TCP	60	49209 -> 443 [FIN, ACK] Seq=744 Ack=1886 Win=65700 Len=0
90	84.568826692	192.168.32.101	192.168.32.101	TCP	54	443 -> 49209 [ACK] Seq=1886 Ack=745 Win=64128 Len=0
91	84.621413291	192.168.32.101	192.168.32.101	TCP	66	49210 -> 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
92	84.621443641	192.168.32.100	192.168.32.101	TCP	66	443 -> 49210 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
93	84.621966152	192.168.32.101	192.168.32.101	TCP	60	49210 -> 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
94	84.624377563	192.168.32.101	192.168.32.101	TLSv1	1	Client Hello
95	84.624395253	192.168.32.100	192.168.32.101	TCP	54	443 -> 49210 [ACK] Seq=1 Ack=125 Win=64128 Len=0
96	84.661949671	192.168.32.100	192.168.32.101	TLSv1	1	Server Hello, Certificate, Server Key Exchange, Server Hello Done
97	84.668489623	192.168.32.101	192.168.32.101	TLSv1	1	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
98	84.669082810	192.168.32.100	192.168.32.101	TLSv1	1	Change Cipher Spec, Encrypted Handshake Message
100	84.873749015	192.168.32.100	192.168.32.101	TCP	1	[TCP Retransmission] 443 -> 49210 [PSH, ACK] Seq=1315 Ack=259 Win=64128 Len=59
101	84.873904398	192.168.32.101	192.168.32.101	TCP	60	49210 -> 443 [ACK] Seq=259 Ack=1374 Win=64324 Len=0
102	84.877739190	192.168.32.101	192.168.32.101	TCP	66	[TCP Dup ACK 101#1] 49210 -> 443 [ACK] Seq=259 Ack=1374 Win=64324 Len=0 SLE=1315 SRE=1374
122	99.320443578	192.168.32.101	192.168.32.101	TCP	60	49210 -> 443 [FIN, ACK] Seq=259 Ack=1374 Win=64324 Len=0
123	99.320663868	192.168.32.100	192.168.32.101	TLSv1	91	Encrypted Alert
124	99.321212408	192.168.32.101	192.168.32.101	TCP	60	49210 -> 443 [RST, ACK] Seq=260 Ack=1411 Win=0 Len=0

Frame 80: 210 bytes on wire (1680 bits), 210 bytes captured (1680 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu\_f4:03:d5 (08:00:27:f4:03:d5), Dst: PcsCompu\_9d:d4:64 (08:00:27:9d:d4:64)

Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100

Transmission Control Protocol, Src Port: 49209, Dst Port: 443, Seq: 1, Ack: 1, Len: 156

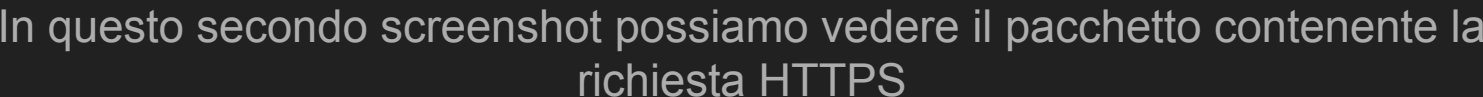
Transport Layer Security

0000 08 00 27 9d d4 64 08 00 27 f4 03 d5 08 00 45 00 .....d.....E  
0010 00 c4 02 bb 40 00 80 06 35 5f c0 a8 20 65 c0 a8 .....@...5...e...  
0020 20 64 c0 39 01 bb 1f 39 09 9a f2 95 3f 23 50 18 .....d9...9...?HP  
0030 40 29 37 5e 00 00 16 03 01 00 97 01 00 00 93 03 .....@)7A.....  
0040 01 65 60 7a a3 11 2f 4c ff ec cf 5a 2e ff 60 76 .....e'z../L...Z...v  
0050 03 c6 0e 11 31 4c 73 01 ba 1f ff c2 91 c2 c8 1e .....lls.....  
0060 0a 20 ea 44 44 67 ec 6d e8 09 1a 1a 01 80 29 d2 .....DDg m.....)

Unexpected end of filter expression. Packets: 131 · Displayed: 55 (42.0%) Profile: Default

Nell'immagine soprastante si trova l'intercettazione delle comunicazioni su wireshark, con evidenziati in basso a destra gli indirizzi mac delle due macchine

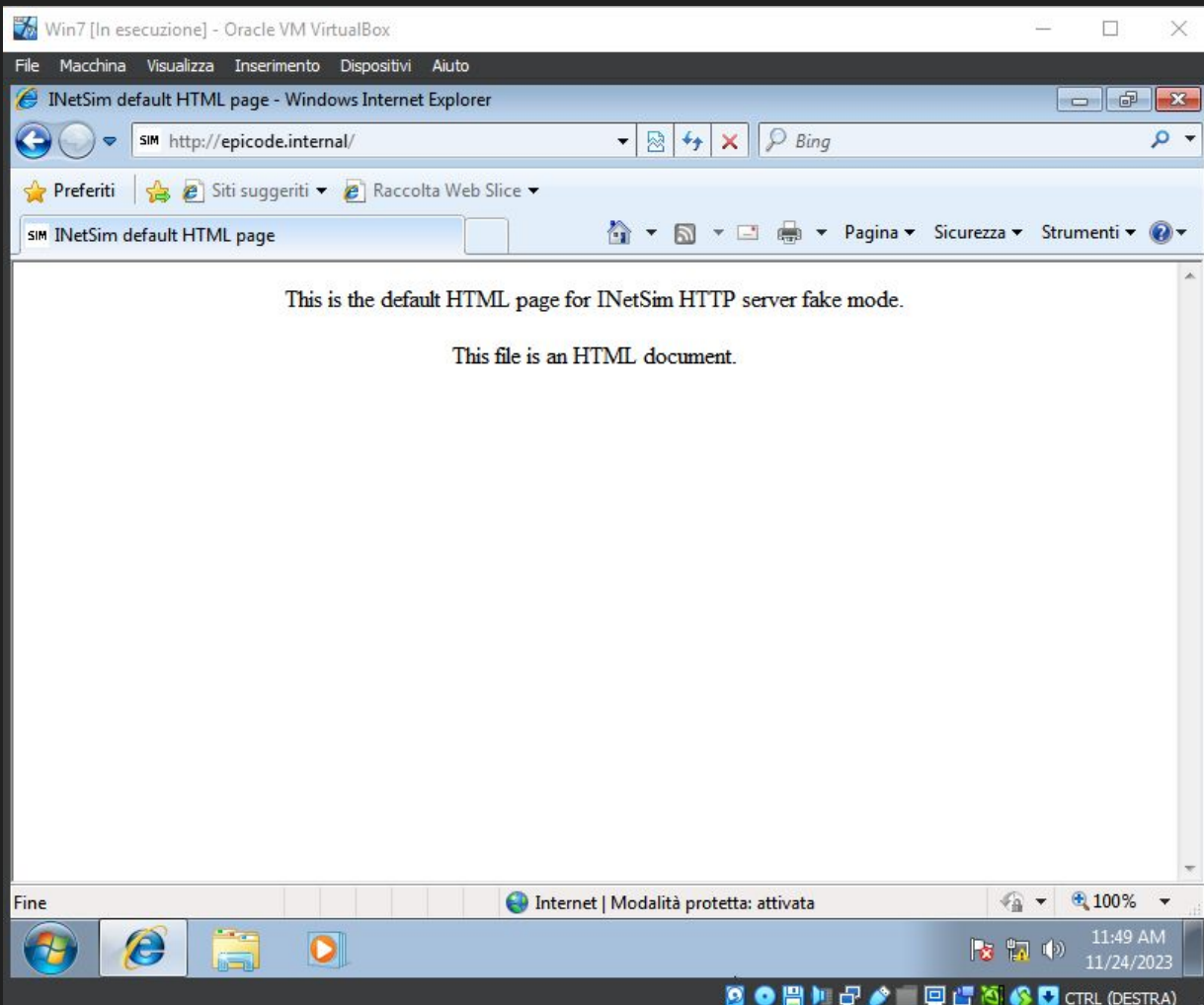




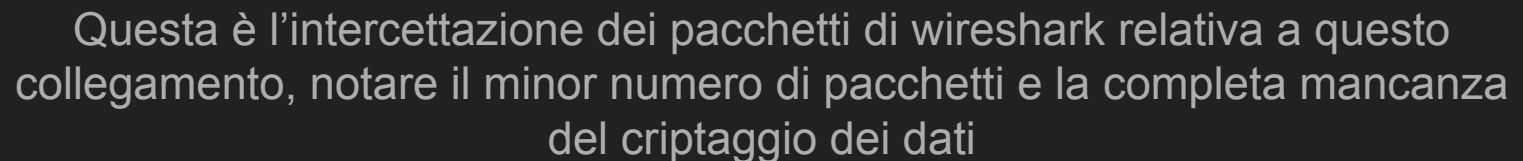
In questo secondo screenshot possiamo vedere il pacchetto contenente la richiesta HTTPS

Per la seconda parte della consegna ho nuovamente modificato inetsim.conf, questa volta attivando il protocollo HTTP

```
kali@kali: ~  
GNU nano 7.2 /etc/inetsim/inetsim.conf *  
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,  
# ftps, irc, https  
#  
start_service dns  
start_service http  
#start_service https  
#start_service smtp  
#start_service smtps  
#start_service pop3  
#start_service pop3s  
#start_service ftp  
#start_service ftps  
#start_service tftp  
#start_service irc  
#start_service ntp  
#start_service finger  
#start_service ident  
#start_service syslog  
#start_service time_tcp  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```



La pagina si apre correttamente, stavolta senza nessun tipo di errore





# CONCLUSIONI

Possiamo rilevare 3 differenze sostanziali tra i protocolli http e https:

## Sicurezza

Il protocollo http è totalmente privo di qualsiasi tipo di cifratura, rendendo i dati totalmente leggibili da un qualsiasi programma di sniffing. Https è invece cifrato, rendendo i dati illeggibili se non si possiede la chiave di cifratura.

## Dimensioni

La mole dei pacchetti inviata con il protocollo https è maggiore

## Velocità

Inversamente proporzionale alle dimensioni, il protocollo https risulta più lento dell'http, per via del tempo impiegato nella creazione della chiave di cifratura e la quantità maggiore di pacchetti inviati. Tuttavia in Computer e Browser moderni questo rallentamento non dovrebbe risultare percepibile.