

Per la consegna dell'esercizio è stata eseguita la configurazione su INetSim di un DNS risolutivo per `epicode.internal`. Inoltre, si è fatto in modo che tramite browser si riuscisse ad accedere sia a `http://epicode.internal` che a `https://epicode.internal`. L'accesso è stato reso possibile anche attraverso il browser di Windows 7.

Inizialmente, con la versione 2024.1 di Kali Linux, non sono riuscito ad attivare la porta 53 del DNS di INetSim, in quanto la versione attuale di Perl di Kali non è compatibile con INetSim. Ho quindi cercato una versione precedente di Kali, ossia una versione dello stesso periodo di INetSim, e ho provato con la versione 2021.1. Tuttavia, questa versione non aveva un certificato SSL valido né una chiave attendibile per Internet Explorer su Windows 7.

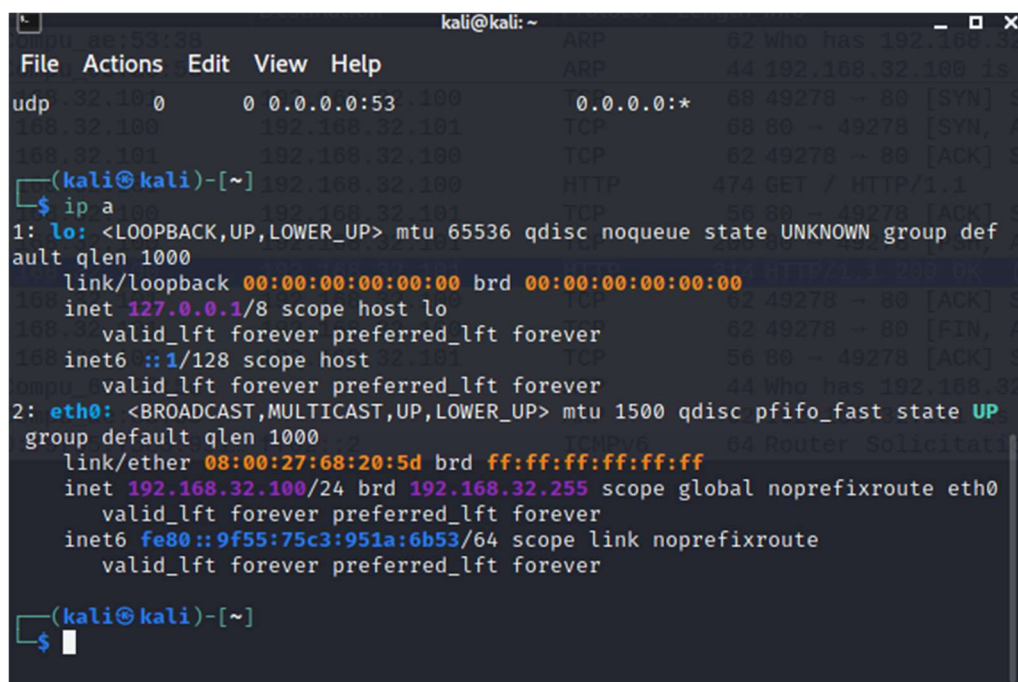
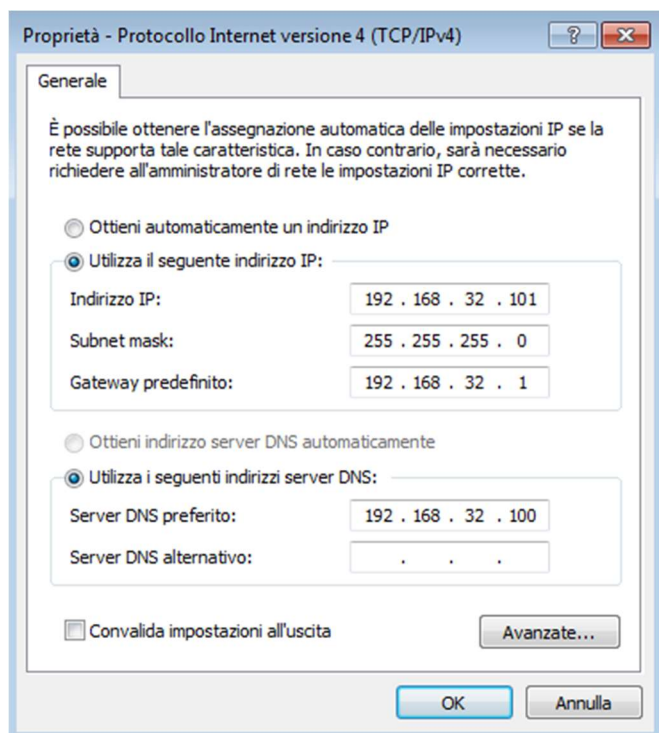
Ho provato a generare una chiave con il comando "openssl", ma, essendo ancora inesperto con tali competenze e avendo poco tempo per la consegna, ho cercato un'altra versione di Kali, in questo caso la 2021.3, dove INetSim era completo di chiave e certificato SSL che non fosse rigettato da Explorer.

I passaggi per impostare tale compito sono stati:

1. Assegnare un IP statico alle due macchine come richiesto dalla traccia, assegnando inoltre a Win7 un server DNS che corrispondesse a quello generato su INetSim.
2. Eseguire sulla Bash di Kali il comando ``sudo nano -c /etc/inetsim/inetsim.conf``. Aperto il file di configurazione, si è proceduto all'aggiunta della riga ``dns_static epicode.internal 192.168.32.100``.
3. Avviare INetSim con il comando ``sudo inetsim --bind-address 0.0.0.0``. Impostando il bind address su 0.0.0.0, INetSim ascolterà su tutte le interfacce di rete della macchina. Questo è utile in ambienti di testing dove non si vuole limitare il servizio a una singola interfaccia di rete o indirizzo IP. Tuttavia, è importante considerare le implicazioni di sicurezza di questa configurazione. Ascoltando su tutte le interfacce di rete, si potrebbe esporre INetSim a reti non sicure o non fidate.
4. Una volta avviato INetSim, tramite browser si può richiedere la visualizzazione con risoluzione di `epicode.internal` sia in http (porta 80) che in https (porta 443). Controllando con Wireshark, si evince che quando si instaura una connessione http, il codice HTML della pagina web è in chiaro, mentre quando è in https, tale messaggio non è in chiaro.

Per dimostrare questi passaggi con delle immagini:

1. **Assegnazione IP Statico**



2. **Configurazione di INetSim**

```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 5.4 /etc/inetsim/inetsim.conf  
#####  
# dns_default_domainname some.domain  
# Default domain name to return with DNS replies  
# Syntax: dns_default_domainname <domain name>  
# Default: inetsim.org  
# dns_default_domainname some.domain  
#####  
# dns_static 192.168.32.101 192.168.32.100  
# 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 epicode.internal 192.168.32.100  
#####  
[ line 222/1999 (11%), col 1/42 (2%), char 4659/41768 (11%) ]  
^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
```

3. **Comando di avvio di INetSim**

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ sudo inetsim --bind-address 0.0.0.0  
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 1657) ===  
Session ID: 1657  
Listening on: 0.0.0.0  
Real Date/Time: 2024-06-03 04:38:46  
Fake Date/Time: 2024-06-03 04:38:46 (Delta: 0 seconds)  
Forking services ...  
* dns_53_tcp_udp - started (PID 1661)  
* http_80_tcp - started (PID 1662)  
* https_443_tcp - started (PID 1663)  
done.  
Simulation running.
```

4. **Controllo dei servizi(porte) in ascolto**

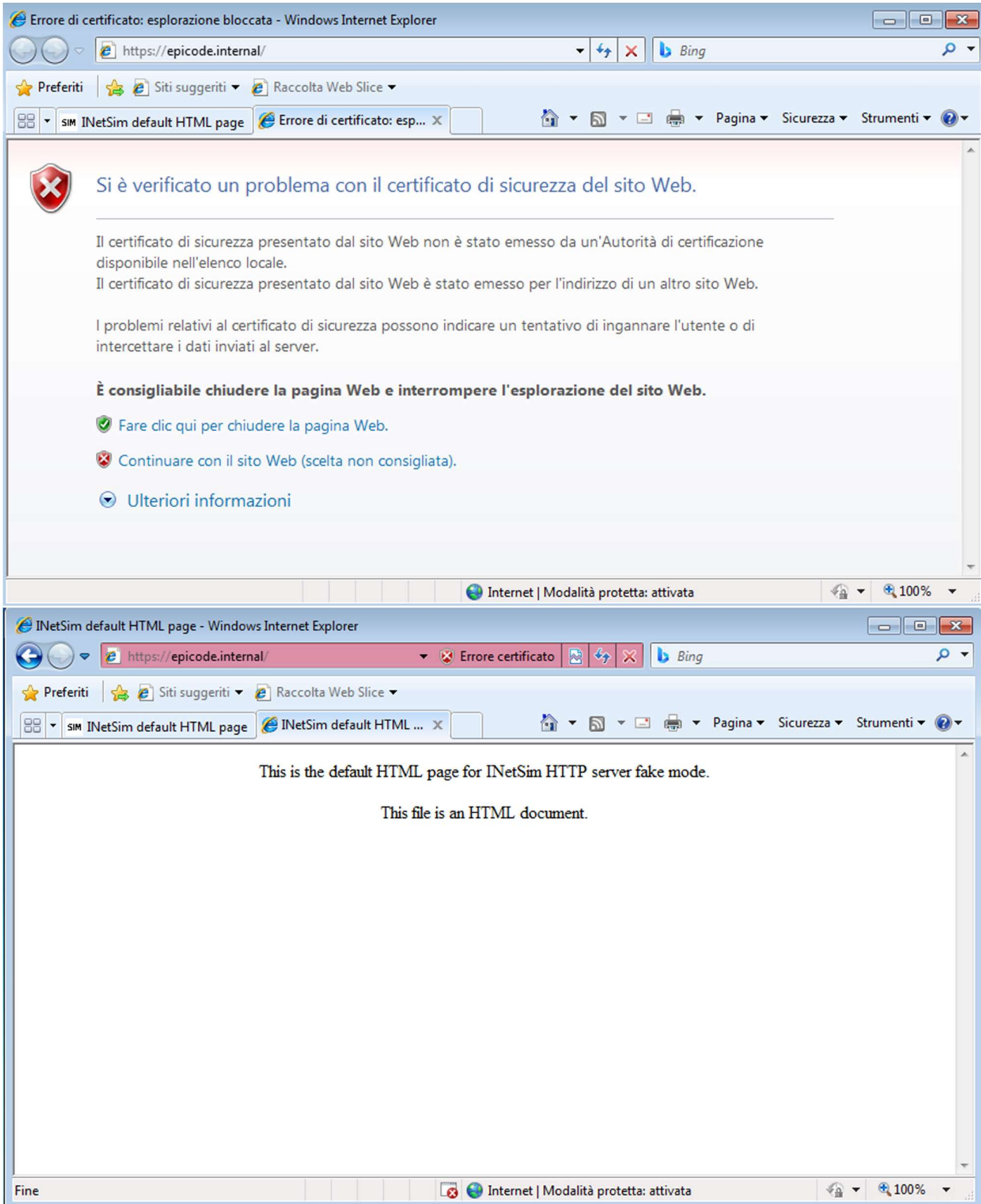
```
(kali@kali)-[~]  
$ sudo netstat -tuln  
Active Internet connections (only servers) 130 x  
Proto Recv-Q Send-Q Local Address Foreign Address State  
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN  
tcp 0 0 0.0.0.0:53 0.0.0.0:* LISTEN  
tcp 0 0 0.0.0.0:443 0.0.0.0:* LISTEN  
udp 0 0 0.0.0.0:53 0.0.0.0:*
```

5. **Connessione Browser da Win 7 e controllo connessione con Wireshark(http)**

The image displays two windows from a Windows 7 system. The top window is a Windows Internet Explorer browser showing a default HTML page from INetSim. The address bar shows the URL `http://epicode.internal/`. The page content reads: "This is the default HTML page for INetSim HTTP server fake mode. This file is an HTML document."

The bottom window is Wireshark, showing a network packet capture. The packet list on the left shows a sequence of packets, including ARP requests, TCP SYN, ACK, and HTTP GET/200 OK. The selected packet (No. 9) is an HTTP 200 OK response. The packet details pane on the right shows the structure of the HTTP response, including the status line `HTTP/1.1 200 OK` and the content type `text/html`. The packet bytes pane at the bottom shows the raw data of the response, including the HTML header and body.

6. ** Connessione Browser da Win 7 e controllo connessione con Wireshark (HTTPS)**



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_ae:53:38		ARP	62	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000011104	PcsCompu_68:20:5d		ARP	44	192.168.32.100 is at 08:00:27:68:20:5d
3	0.000135502	192.168.32.101	192.168.32.100	TCP	68	49270 → 443 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=4 SACK_PERM=1
4	0.000150784	192.168.32.100	192.168.32.101	TCP	68	443 → 49270 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
5	0.000240479	192.168.32.101	192.168.32.100	TCP	62	49270 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.000477001	192.168.32.101	192.168.32.100	TLShv1	217	Client Hello
7	0.000498520	192.168.32.100	192.168.32.101	TCP	56	443 → 49270 [ACK] Seq=1 Ack=162 Win=64128 Len=0
8	0.003141173	192.168.32.100	192.168.32.101	TLShv1	1375	Server Hello, Certificate, Server Key Exchange, Server Hello Done
9	0.007094520	192.168.32.101	192.168.32.100	TLShv1	190	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.007104215	192.168.32.100	192.168.32.101	TCP	56	443 → 49270 [ACK] Seq=1320 Ack=290 Win=64128 Len=0
11	0.007419627	192.168.32.100	192.168.32.101	TLShv1	115	Change Cipher Spec, Encrypted Handshake Message
12	0.018093081	PcsCompu_ae:53:38		ARP	62	Who has 192.168.32.1? Tell 192.168.32.101
13	0.197987422	192.168.32.101	192.168.32.100	TCP	62	49270 → 443 [ACK] Seq=296 Ack=1379 Win=64320 Len=0
14	0.869353028	PcsCompu_ae:53:38		ARP	62	Who has 192.168.32.1? Tell 192.168.32.101
15	1.069319520	PcsCompu_ae:53:38		ARP	62	Who has 192.168.32.1? Tell 192.168.32.101
16	3.447017666	192.168.32.101	192.168.32.255	NBNS	94	Name query NB WPAD<00>
17	4.196852343	192.168.32.101	192.168.32.255	NBNS	94	Name query NB WPAD<00>
18	4.940405952	192.168.32.101	192.168.32.255	NBNS	94	Name query NB WPAD<00>
19	5.237035407	PcsCompu_68:20:5d		ARP	44	Who has 192.168.32.101? Tell 192.168.32.100
20	5.238102559	PcsCompu_ae:53:38		ARP	62	192.168.32.101 is at 08:00:27:ae:53:38
21	5.697962532	PcsCompu_ae:53:38		ARP	62	Who has 192.168.32.1? Tell 192.168.32.101
22	6.366059037	PcsCompu_ae:53:38		ARP	62	Who has 192.168.32.1? Tell 192.168.32.101
23	7.366371679	PcsCompu_ae:53:38		ARP	62	Who has 192.168.32.1? Tell 192.168.32.101
24	9.147848644	192.168.32.101	192.168.32.255	NBNS	94	Name query NB WPAD<00>
25	9.890564041	192.168.32.101	192.168.32.255	NBNS	94	Name query NB WPAD<00>
26	10.645796121	192.168.32.101	192.168.32.255	NBNS	94	Name query NB WPAD<00>
27	12.417133314	192.168.32.101	192.168.32.100	TLShv1	509	Application Data
28	12.41735173	192.168.32.100	192.168.32.101	TCP	56	443 → 49270 [ACK] Seq=1379 Ack=749 Win=64128 Len=0
29	12.425581405	192.168.32.100	192.168.32.101	TLShv1	237	Application Data
30	12.427047181	192.168.32.100	192.168.32.101	TLShv1	386	Application Data, Encrypted Alert
31	12.427130815	192.168.32.101	192.168.32.100	TCP	62	49270 → 443 [ACK] Seq=749 Ack=1091 Win=65700 Len=0
32	12.427203643	192.168.32.101	192.168.32.100	TCP	62	49270 → 443 [FIN, ACK] Seq=749 Ack=1091 Win=65700 Len=0
33	12.427211419	192.168.32.100	192.168.32.101	TCP	56	443 → 49270 [ACK] Seq=1891 Ack=750 Win=64128 Len=0
Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface any, id 0						
Linux cooked capture v1						
0000	00 01 00 01 00 06 00 00	27 ae 53 38 00 00 00 06S8..			
0010	00 01 00 06 04 00 01 00	00 00 27 ae 53 38 c9 a8S8..			
0020	00 05 00 00 00 00 00 00	c0 a8 20 64 00 00 00 00	e.....d...			
0030	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			