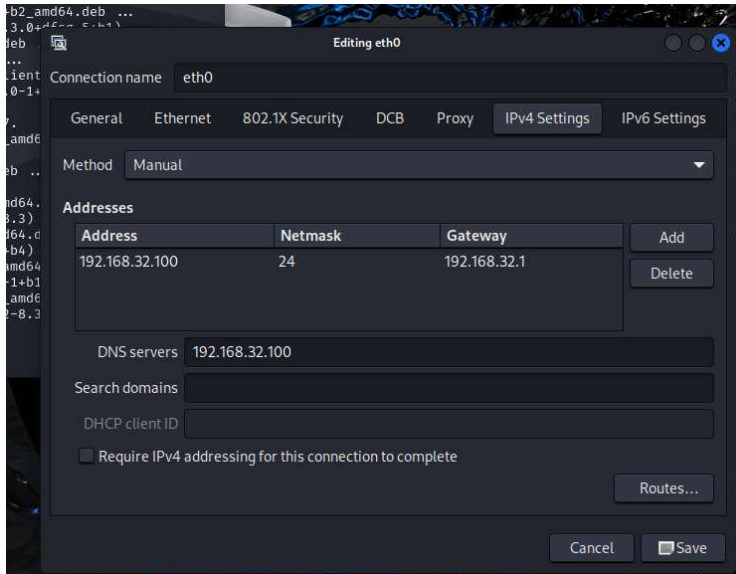


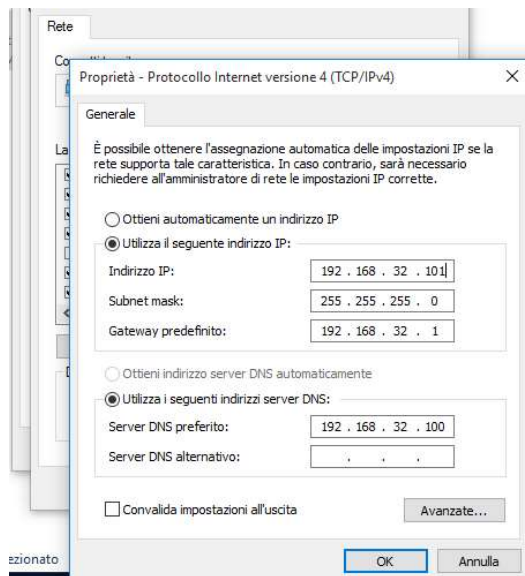
Richiesta progetto finale: mediante Wireshark intercettare il traffico HTTPS e HTTP tra Kali Linux e Windows 10 con epicode.internal

Configurazione ambiente Kali Linux IP 192.168.32.10



```
kali@kali: ~  
File Actions Edit View Help  
inet6 fe80::96d7:410a:68c0:6769/64 scope link noprefixroute  
    valid_lft forever preferred_lft forever  
  
[kali@kali]~  
$  
  
[kali@kali]~  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def  
    ault qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g  
    roup default qlen 1000  
    link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.32.100/24 brd 192.168.32.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
    inet6 fd00::786a:9751:8e8b:90c0/64 scope global dynamic noprefixroute  
        valid_lft 86396sec preferred_lft 14396sec  
    inet6 fe80::935f:9ed3:f1e4:5936/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
  
[kali@kali]~  
$
```

Configurazione ambiente Windows 10 IP 192.168.32.101



```

Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user>ipconfig

Configurazione IP di Windows

Scheda Ethernet Ethernet:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::2c60:754f:65cd:300a%4
    Indirizzo IPv4. . . . . : 192.168.32.101
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.32.1

Scheda Tunnel isatap.{92D61F82-1D19-45C9-B7CF-2E5AF2D63627}:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

C:\Users\user>

```

Eseguo un ping di test tra le due macchine inserendole in rete interna

```

kali@kali: ~
File Actions Edit View Help

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
    valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.32.100/24 brd 192.168.32.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fd00::786a:9751:8e8b:90c0/64 scope global dynamic noprefixroute
        valid_lft 86396sec preferred_lft 14396sec
    inet6 fe80::935f:9ed3:f1e4:5936/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(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=0.329 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.559 ms
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.296 ms
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.661 ms
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.449 ms
64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=0.619 ms
64 bytes from 192.168.32.101: icmp_seq=7 ttl=128 time=0.432 ms
64 bytes from 192.168.32.101: icmp_seq=8 ttl=128 time=0.338 ms

```

```
C:\ Prompt dei comandi

Scheda Ethernet Ethernet:

Suffisso DNS specifico per connessione:
Indirizzo IPv6 locale rispetto al collegamento . : fe80::2c60:754f:65cd:300a%4
Indirizzo IPv4. . . . . : 192.168.32.101
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . : 192.168.32.1

Scheda Tunnel isatap.{92D61F82-1D19-45C9-B7CF-2E5AF2D63627}:

Stato supporto. . . . . : Supporto disconnesso
Suffisso DNS specifico per connessione:

C:\Users\user>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=3ms 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 = 3ms, Medio = 1ms

C:\Users\user>
```

L'applicativo intesim permette di simulare il traffico HTTPS ma anche http, affinché possa essere in grado di eseguire tale compito è necessario attivare i servizi necessari modificandone la configurazione.

Disattivo inserendo il # davanti a tutti i servizi non necessari.

Infine, attivo il servizio come da screen.

```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.2 /etc/inetsim/inetsim.conf *
# 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
#start_service smtp
#start_service smtps
#start_service pop3
#start_service pop3s
#start_service ftp
#start_service ftps
^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify
```

```
kali@kali: ~  
File Actions Edit View Help  
L-$ sudo inetsim  
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg  
Main logfile '/var/log/inetsim/main.log' does not exist. Trying to create it.  
..  
Main logfile '/var/log/inetsim/main.log' successfully created.  
Sub logfile '/var/log/inetsim/service.log' does not exist. Trying to create i  
t ...  
Sub logfile '/var/log/inetsim/service.log' successfully created.  
Debug logfile '/var/log/inetsim/debug.log' does not exist. Trying to create i  
t ...  
Debug logfile '/var/log/inetsim/debug.log' successfully created.  
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 96590) ==  
Session ID: 96590  
Listening on: 127.0.0.1  
Real Date/Time: 2024-12-05 14:34:21  
Fake Date/Time: 2024-12-05 14:34:21 (Delta: 0 seconds)  
Forking services ...  
* https_443_tcp - started (PID 96600)  
done.  
Simulation running.  
█
```

L'attivazione del server DNS viene eseguito tramite il dnsmasq precedentemente installato.

```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.2 /etc/dnsmasq.conf  
# Log lots of extra information about DHCP transactions.  
#log-dhcp  
  
# Include another lot of configuration options.  
#conf-file=/etc/dnsmasq.more.conf  
#conf-dir=/etc/dnsmasq.d  
  
# Include all the files in a directory except those ending in .bak  
#conf-dir=/etc/dnsmasq.d,.bak  
  
# Include all files in a directory which end in .conf  
#conf-dir=/etc/dnsmasq.d/*.conf  
  
# If a DHCP client claims that its name is "wpad", ignore that.  
# This fixes a security hole. see CERT Vulnerability VU#598349  
#dhcp-name-match=set:wpad-ignore,wpad  
#dhcp-ignore-names=tag:wpad-ignore  
interface-eth0  
listen-address=192.168.32.100  
bind-interfaces  
address=/epicode.internal/192.168.32.100  
█  
  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Prova ad eseguire ping su epicode.internal

```
C:\Users\user>ping epicode.internal

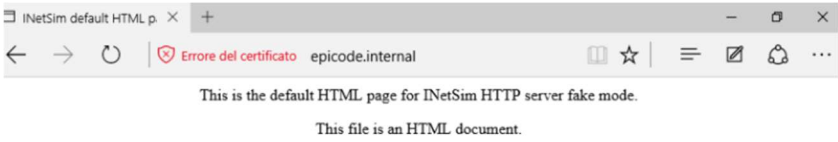
Esecuzione di Ping epicode.internal [192.168.32.100] con 32 byte di dati:
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64

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

C:\Users\user>
Ping: min/avg/max/mdev = 0.027/0.048/0.131/0.024 ms

(kali@kali)-[~]
└─$ ping epicode.internal
PING epicode.internal (192.168.32.100) 56(84) bytes of data.
64 bytes from 192.168.32.100: icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from 192.168.32.100: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 192.168.32.100: icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from 192.168.32.100: icmp_seq=4 ttl=64 time=0.031 ms
64 bytes from 192.168.32.100: icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from 192.168.32.100: icmp_seq=6 ttl=64 time=0.031 ms
64 bytes from 192.168.32.100: icmp_seq=7 ttl=64 time=0.038 ms
64 bytes from 192.168.32.100: icmp_seq=8 ttl=64 time=0.032 ms
64 bytes from 192.168.32.100: icmp_seq=9 ttl=64 time=0.030 ms
64 bytes from 192.168.32.100: icmp_seq=10 ttl=64 time=0.040 ms
64 bytes from 192.168.32.100: icmp_seq=11 ttl=64 time=0.044 ms
```

Provando a cercare sul browser predefinito in Windows <http://epicode.internal> viene visualizzata questa pagina HTML



Ora apro su Kali Linux il programma Wireshark per il monitoraggio del traffico in questo caso HTTPS tra le due macchine.

Tra le interfacce disponibili seleziono eth0 ed attivo il filtro sulla porta 443 (HTTPS).

No.	Time	Source	Destination	Protocol	Length	Info
45	27.404515391	192.168.32.100	192.168.32.101	DNS	92	Standard query response 0xe64e A epicode.internal A 192.168.32.101
46	27.405627284	192.168.32.101	192.168.32.100	TLSv1.2	436	Application Data
47	27.405649667	192.168.32.100	192.168.32.101	TCP	54	443 → 49458 [ACK] Seq=1298 Ack=876 Win=31872 Len=0
48	27.407901484	192.168.32.100	192.168.32.101	TLSv1.2	280	New Session Ticket, Change Cipher Spec, Encrypted Handshake
49	27.407908348	192.168.32.101	192.168.32.100	TCP	60	49458 → 443 [ACK] Seq=876 Ack=1524 Win=262144 Len=0
50	27.476440845	192.168.32.100	192.168.32.101	TLSv1.2	234	Application Data
51	27.477992345	192.168.32.101	192.168.32.100	TCP	60	49458 → 443 [ACK] Seq=876 Ack=1704 Win=261888 Len=0
52	27.478028412	192.168.32.100	192.168.32.101	TLSv1.2	341	Application Data
53	27.478513909	192.168.32.101	192.168.32.100	TCP	60	49458 → 443 [ACK] Seq=876 Ack=1991 Win=261632 Len=0
54	27.479988862	192.168.32.101	192.168.32.100	TCP	60	49458 → 443 [FIN, ACK] Seq=876 Ack=1991 Win=261632 Len=0
55	27.491382424	192.168.32.100	192.168.32.101	TLSv1.2	85	Encrypted Alert
56	27.495230219	192.168.32.101	192.168.32.100	TCP	60	49458 → 443 [RST, ACK] Seq=877 Ack=2022 Win=0 Len=0
57	27.500184643	192.168.32.101	192.168.32.100	DNS	77	Standard query 0x7072 A urs.microsoft.com
58	27.503580141	PCSSystemtec_58:d0:...	Broadcast	ARP	42	Who has 192.168.32.1? Tell 192.168.32.100
59	27.526592254	192.168.32.101	192.168.32.100	DNS	77	Standard query 0x7072 A urs.microsoft.com
60	27.695666351	PCSSystemtec_16:3b:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
61	28.506734119	PCSSystemtec_58:d0:...	Broadcast	ARP	42	Who has 192.168.32.1? Tell 192.168.32.100
62	28.507226911	PCSSystemtec_16:3b:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
63	28.546278172	192.168.32.101	192.168.32.100	DNS	77	Standard query 0x7072 A urs.microsoft.com
64	29.509452079	PCSSystemtec_16:3b:...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
65	29.530507288	PCSSystemtec_58:d0:...	Broadcast	ARP	42	Who has 192.168.32.1? Tell 192.168.32.100
66	30.541750651	192.168.32.101	192.168.32.100	DNS	77	Standard query 0x7072 A urs.microsoft.com
67	31.072024147	fe80::613d:df21:936...	ff02::1:2	DHCPv6	157	Solicit XID: 0xae85c3 CID: 000100012ede4b04080027163b0c
68	34.558511252	192.168.32.101	192.168.32.100	DNS	77	Standard query 0x7072 A urs.microsoft.com
69	34.559256038	PCSSystemtec_58:d0:...	Broadcast	ARP	42	Who has 192.168.32.1? Tell 192.168.32.100

Frame 56: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0	0000 00 00 27 58 d0 42 08 00 27 16 3b 0c 08 00 45 00
Ethernet II, Src: PCSSystemtec_16:3b:0c (08:00:27:16:3b:0c), Dst: PCSSystemtec_58:d0:42 (08:00:27:58:d0:42)	0010 00 28 79 78 40 00 00 06 bf 3d c0 a8 20 65 c0 a8
Destination: PCSSystemtec_58:d0:42 (08:00:27:58:d0:42)	0020 20 64 c1 32 01 bb 27 f5 df ec e4 24 1e 49 50 14
Source: PCSSystemtec_16:3b:0c (08:00:27:16:3b:0c)	0030 00 00 20 79 00 00 00 00 00 00 00 00 00 00 00
Type: IPv4 (0x0800)	
Padding: 000000000000	
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: 40	

Per eseguire la seconda parte dell'esercizio sarà necessario andare a disattivare da inetsim il servizio HTTPS e attivare HTTP

```

kali@kali: ~
File Actions Edit View Help
GNU nano 8.2 /etc/inetsim/inetsim.conf *
# 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
#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
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify

```

Il procedimento a seguire è il medesimo a differenza di Wireshark, quindi aprendo l'applicativo seleziono eth0 e questa volta filtro per la porta 80 (HTTP)

tcp.port==80						
No.	Time	Source	Destination	Protocol	Length	Info
4	0.880857509	192.168.32.101	192.168.32.100	TCP	66	49460 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256
5	0.880914231	192.168.32.100	192.168.32.101	TCP	66	80 → 49460 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1460
6	0.881695239	192.168.32.101	192.168.32.100	TCP	66	49460 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
7	0.882133491	192.168.32.101	192.168.32.100	HTTP	497	GET / HTTP/1.1
8	0.882153270	192.168.32.100	192.168.32.101	TCP	54	80 → 49460 [ACK] Seq=1 Ack=354 Win=31872 Len=0
9	0.920631571	192.168.32.100	192.168.32.101	TCP	294	80 → 49460 [PSH, ACK] Seq=1 Ack=354 Win=31872 Len=150 [T
10	0.921750756	192.168.32.101	192.168.32.100	TCP	66	49460 → 80 [ACK] Seq=354 Ack=151 Win=261888 Len=0
11	0.921792911	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
12	0.922653889	192.168.32.101	192.168.32.100	TCP	66	49460 → 80 [ACK] Seq=354 Ack=409 Win=261632 Len=0
13	0.922896749	192.168.32.101	192.168.32.100	TCP	66	49460 → 80 [FIN, ACK] Seq=354 Ack=409 Win=261632 Len=0
14	0.926370175	192.168.32.100	192.168.32.101	TCP	54	80 → 49460 [FIN, ACK] Seq=409 Ack=355 Win=31872 Len=0
15	0.927009338	192.168.32.101	192.168.32.100	TCP	66	49460 → 80 [ACK] Seq=355 Ack=410 Win=261632 Len=0

Frame 10: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

Ethernet II, Src: PCSSystemtec_16:3b:0c (08:00:27:16:3b:0c), Dst: PCSSystemtec_5b:d0:42 (08:00:27:16:5b:d0:42)

Destination: PCSSystemtec_5b:d0:42 (08:00:27:16:5b:d0:42)

Source: PCSSystemtec_16:3b:0c (08:00:27:16:3b:0c)

Type: IPv4 (0x0800)

Padding: 000000000000

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: 40

0000

08 00 27 5b d0 42 08 00 27 16 3b 0c 08 00 45 00

0010

00 28 7a a4 40 00 00 00 be 11 c0 a8 20 05 c0 a8

0020

20 64 c1 34 00 00 21 19 40 dd b7 14 50 02 50 10

0030

03 ff bf 29 00 00 00 00 00 00 00 00