

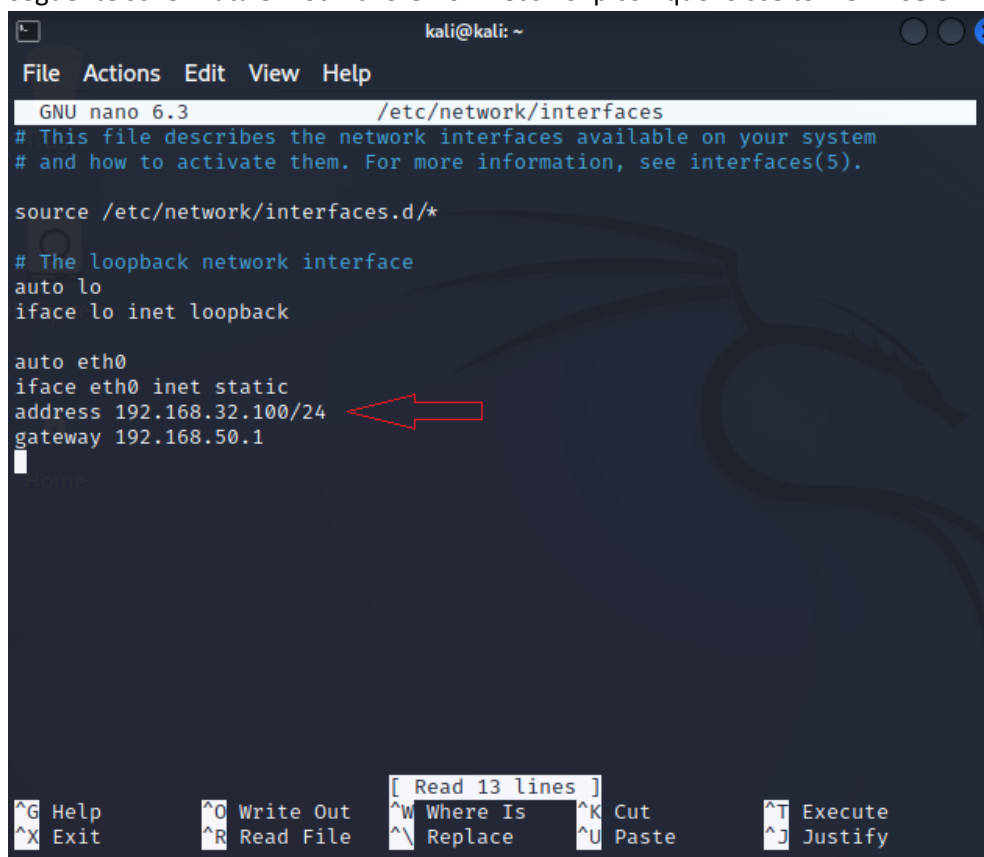
Simulazione e intercettazioni dati

Requisiti

- Kali Linux Ip: 192.168.32.100
- Windows 7 Ip: 192.168.32.101
- HTTPS Server: Attivo
- Servizio DNS per risoluzione di dominio: Attivo

Modifica Ip Kali Linux :

Accedendo al Terminale di Kali Linux scriveremo: **sudo nano /etc/network/interfaces** , dove ci apparirà la seguente schermata e modificheremo il vecchio Ip con quello scelto: **192.168.32.100**



```
kali@kali: ~
File Actions Edit View Help
GNU nano 6.3 /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

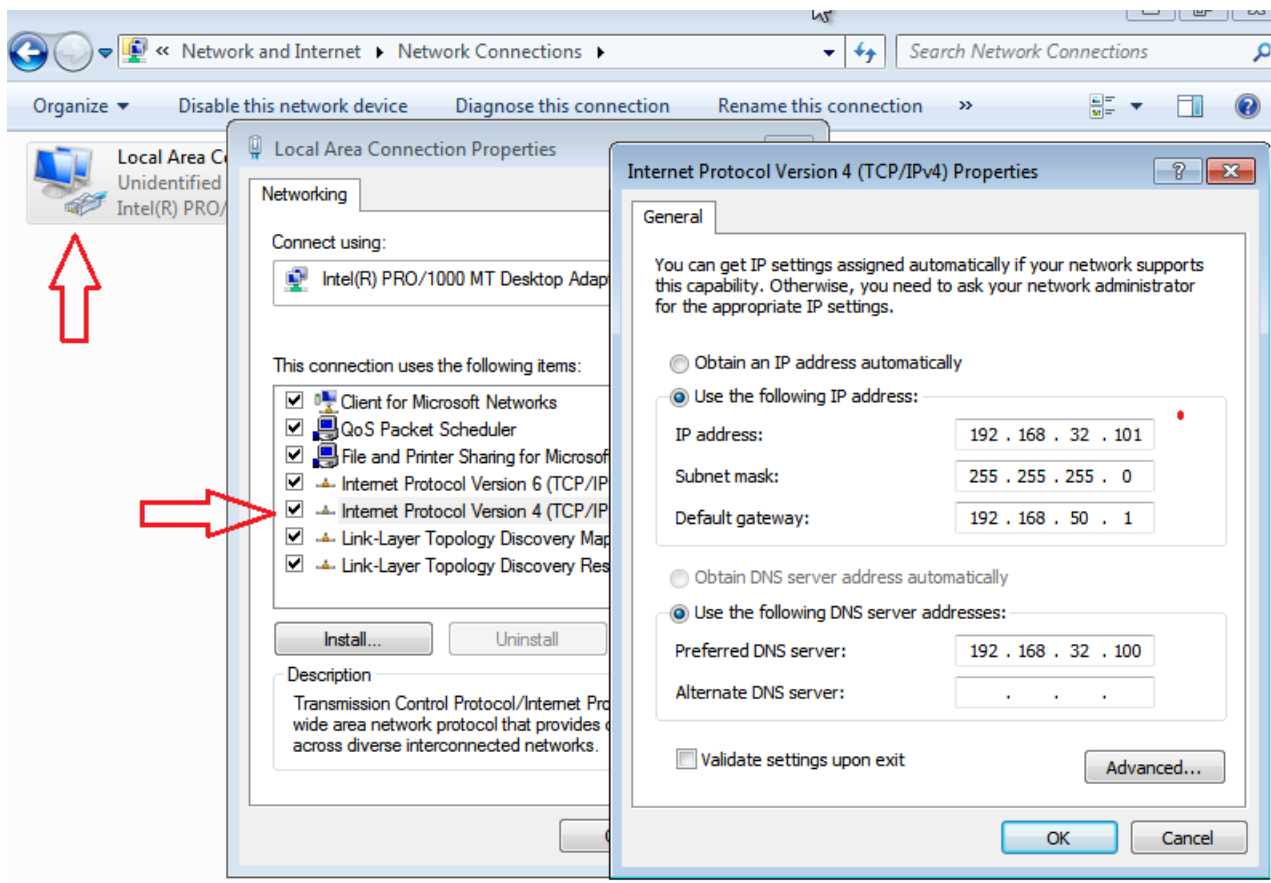
auto eth0
iface eth0 inet static
address 192.168.32.100/24
gateway 192.168.50.1
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
[ Read 13 lines ]
```

Una volta cambiato Ip, tramite il comando **CTRL+O**, salveremo la modifica e tramite il comando

Sudo /etc/init.d/networking restart verrà riavviata la scheda di rete.

Modifica Ip Windows 7

Dal pannello di controllo di **Windows -> Pannello di Controllo -> Network and Internet -> Network -> Change adapter Setting** ; ci ritroveremo a seguire i passaggi nell'immagine qui sotto per cambiare il nostro Ip in: **192.168.32.101**



Configurazione INETSIM

Dal terminale di Kali avvieremo il comando: **sudo nano /etc/inetsim/inetsim.conf** andando ad aggiungere le seguenti righe per poi Salvare e chiudere.

```
#####
# dns_static
#
# 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
```

```
#####
# service_bind_address
#
# IP address to bind services to
# Syntax: service_bind_address <IP address>
# Default: 127.0.0.1
#
#service_bind_address 10.10.10.1
#service_bind_address 192.168.32.100
#####
```

Avviamo il Terminale insieme a : **sudo inetsim** per avviare il processo e confermare l'avvenuta operazione ci basterà andare su **Windows 7** e ricercare nel Browser "**Epicode.Internal**"

```
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 3069) ==
Session ID: 3069
Listening on: 192.168.32.100
Real Date/Time: 2022-10-28 05:20:24
Fake Date/Time: 2022-10-28 05:20:24 (Delta: 0 seconds)
Forking services ...
```

MAC ADDRESS

```
Connection-specific DNS Suffix . : 
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-10-A7-69
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::a8c9:c5a4:b4f8:5041%11(Preferred)
IPv4 Address. . . . . : 192.168.32.101(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.50.1
DHCPv6 IAID . . . . . : 235405351
DHCPv6 Client DUID. . . . . : 00-01-00-01-2A-E8-4E-0F-00-00-27-10-A7-69

DNS Servers . . . . . : 192.168.32.100
NetBIOS over Tcpip. . . . . : Enabled
```

Mac Address Windows 7

```
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:fe22:464f prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:22:46:4f txqueuelen 1000 (Ethernet)
    RX packets 262 bytes 19000 (18.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 104 bytes 12287 (11.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Mac Address Kali

Wireshark

Dopo aver configurato il nostro ambiente di lavoro apriremo Wireshark da Kali e **Epicode.Internal** da Windows, procedendo al catturare il traffico di pacchetti.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_10:a7:69	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000017312	PcsCompu_22:46:4f	PcsCompu_10:a7:69	ARP	42	192.168.32.100 is at 08:00:27:22:46:4f
3	0.000287660	192.168.32.101	192.168.32.100	TCP	66	49189 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.000311568	192.168.32.100	192.168.32.101	TCP	66	80 → 49189 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
5	0.000530161	192.168.32.101	192.168.32.100	TCP	60	49189 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
6	0.000718423	192.168.32.101	192.168.32.100	HTTP	305	GET / HTTP/1.1
7	0.000725781	192.168.32.100	192.168.32.101	TCP	54	80 → 49189 [ACK] Seq=1 Ack=252 Win=64128 Len=0
8	0.012272011	192.168.32.100	192.168.32.101	TCP	204	80 → 49189 [PSH, ACK] Seq=1 Ack=252 Win=64128 Len=150 [TCP segment of a reassembled PDU]
9	0.012649953	192.168.32.101	192.168.32.100	TCP	60	49189 → 80 [ACK] Seq=252 Ack=151 Win=65536 Len=0
10	0.012659606	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
11	0.012900243	192.168.32.101	192.168.32.100	TCP	60	49189 → 80 [ACK] Seq=252 Ack=409 Win=65280 Len=0
12	0.013732238	192.168.32.101	192.168.32.100	TCP	60	49189 → 80 [FIN, ACK] Seq=252 Ack=409 Win=65280 Len=0
13	0.013900065	192.168.32.100	192.168.32.101	TCP	54	80 → 49189 [FIN, ACK] Seq=409 Ack=253 Win=64128 Len=0
14	0.014223746	192.168.32.101	192.168.32.100	TCP	60	49189 → 80 [ACK] Seq=253 Ack=410 Win=65280 Len=0

Frame 6: 305 bytes on wire (2440 bits), 305 bytes captured (2440 bits) on interface eth0, id 0
Ethernet II, Src: PcsCompu_10:a7:69 (08:00:27:10:a7:69), Dst: PcsCompu_22:46:4f (08:00:27:22:46:4f)
Destination: PcsCompu_22:46:4f (08:00:27:22:46:4f)
Source: PcsCompu_10:a7:69 (08:00:27:10:a7:69)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100
Transmission Control Protocol, Src Port: 49189, Dst Port: 80, Seq: 1, Ack: 1, Len: 251
Hypertext Transfer Protocol

HTTP

Possiamo vedere come viene intercettata la comunicazione analizzando i **MAC ADDRESS** dei rispettivi dispositivi. Seguendo il numero progressivo dei pacchetti (**No.**) si può notare nel **No.3** la richiesta di comunicazione (**SYN**) dal client verso il Server con un numero di sequenza casuale (**Es. SEQ = 4444**) , nel **No.4** il Server riceverà il pacchetto rispondendo con un nuovo numero casuale (**SEQ = 7454**) ed includerà un numero di riconoscimento, ovvero la sequenza + 1 (**ACK = 4445**), nel **No.5** stabilisce la comunicazione rimandando un pacchetto contenente la Flag + 1 (**ACK = 7455**) questa è chiamato **Three-Way-Handshake**

HTTPS

No.	Time	Source	Destination	Protocol	Length	Info
10	12.204465127	192.168.32.101	192.168.32.100	TCP	62	49195 → 443 [SYN] Seq=0 Win=8192 L
11	12.204496747	192.168.32.100	192.168.32.101	TCP	62	443 → 49195 [SYN, ACK] Seq=0 Ack=1
12	12.204764856	192.168.32.101	192.168.32.100	TCP	60	49195 → 443 [ACK] Seq=1 Ack=1 Win=
13	12.206610561	192.168.32.101	192.168.32.100	TLSv1.2	271	Client Hello
14	12.206625279	192.168.32.100	192.168.32.101	TCP	54	443 → 49195 [ACK] Seq=1 Ack=218 Wi
15	12.214995127	192.168.32.100	192.168.32.101	TLSv1.2	1821	Server Hello, Certificate, Server
16	12.215443640	192.168.32.101	192.168.32.100	TCP	60	49195 → 443 [ACK] Seq=218 Ack=1768
17	12.232602771	192.168.32.101	192.168.32.100	TLSv1.2	372	Client Key Exchange, Change Cipher
18	12.232622940	192.168.32.100	192.168.32.101	TCP	54	443 → 49195 [ACK] Seq=1768 Ack=536
19	12.235321254	192.168.32.100	192.168.32.101	TLSv1.2	105	Change Cipher Spec, Encrypted Hand
20	12.235631483	192.168.32.101	192.168.32.100	TCP	60	49195 → 443 [ACK] Seq=536 Ack=1819
21	12.238895441	PcsCompu_10:a7:69	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168

Frame 13: 271 bytes on wire (2168 bits), 271 bytes captured (2168 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu_10:a7:69 (08:00:27:10:a7:69), Dst: PcsCompu_22:46:4f (08:00:27:22:46:4f)

- Destination: PcsCompu_22:46:4f (08:00:27:22:46:4f)
- Source: PcsCompu_10:a7:69 (08:00:27:10:a7:69)
- Type: IPv4 (0x0800)

Qui possiamo vedere, a differenza di HTTP, un protocollo diverso TLS (Transport Layer Security) progettato per proteggere le comunicazioni di rete.

Transport Layer Security
TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message

Quindi utilizzando degli algoritmi di crittografia.

Fabio De Rosa