

## Progetto Finale

---

Abbiamo predisposto un laboratorio virtuale configurando una macchina Kali Linux con l'indirizzo IP statico 192.168.50.100. Successivamente, abbiamo modificato il file di configurazione inetsim.conf per attivare i servizi HTTP e HTTPS offerti da INetSim.

```
File  Actions  Edit  View  Help

(kali@kali)-[/etc/inetsim]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group de
fault 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
group default qlen 1000
    link/ether 08:00:27:c9:0e:da brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.100/24 brd 192.168.50.255 scope global noprefixroute eth
0
        valid_lft forever preferred_lft forever
    inet6 fe80::2a93:2dec:ede9:660a/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(kali@kali)-[/etc/inetsim]
$
```

File di configurazione inetsim

## File di configurazione inetsim

```
File  Actions  Edit  View  Help
GNU nano 8.4                                inetsim.conf
# 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

^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Con wireshark intercettiamo il traffico dei pacchetti e vedremo le richieste con mac address e anche il contenuto anche se l'https è cifrato quindi il contenuto non si può vedere.

FileEditViewGoCaptureAnalyzeStatisticsTelephonyWirelessToolsHelp

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

No.	Time	Source	Destination	Protocol	Length	Info
69	1.307202925	192.168.50.100	192.168.50.102	TLSv1.3	1497	Server Hello, Change Cipher Spec, Applica
70	1.308069949	192.168.50.102	192.168.50.100	TLSv1.3	84	Change Cipher Spec, Application Data
71	1.308070225	192.168.50.102	192.168.50.100	TCP	60	49506 → 443 [FIN, ACK] Seq=2067 Ack=1444
72	1.308098791	192.168.50.100	192.168.50.102	TCP	54	443 → 49506 [ACK] Seq=1444 Ack=2067 Win=6
73	1.310105498	192.168.50.102	192.168.50.100	TCP	66	49507 → 443 [SYN] Seq=0 Win=8192 Len=0 MS
74	1.310140253	192.168.50.100	192.168.50.102	TCP	66	443 → 49507 [SYN, ACK] Seq=0 Ack=1 Win=64
75	1.310752644	192.168.50.102	192.168.50.100	TCP	60	49507 → 443 [ACK] Seq=1 Ack=1 Win=65536 L
76	1.313085569	192.168.50.102	192.168.50.100	TLSv1.3	2090	Client Hello
77	1.313106920	192.168.50.100	192.168.50.102	TCP	54	443 → 49507 [ACK] Seq=1 Ack=2037 Win=6310
78	1.336455081	192.168.50.100	192.168.50.102	TCP	54	443 → 49506 [FIN, ACK] Seq=1444 Ack=2068
79	1.336848227	192.168.50.102	192.168.50.100	TCP	60	49506 → 443 [ACK] Seq=2068 Ack=1445 Win=6
80	1.405242658	192.168.50.100	192.168.50.102	TLSv1.3	1497	Server Hello, Change Cipher Spec, Applica
81	1.406524721	192.168.50.102	192.168.50.100	TLSv1.3	134	Change Cipher Spec, Application Data
82	1.406553471	192.168.50.100	192.168.50.102	TCP	54	443 → 49507 [ACK] Seq=1444 Ack=2117 Win=6
83	1.406823592	192.168.50.102	192.168.50.100	TLSv1.3	687	Application Data
84	1.406830776	192.168.50.100	192.168.50.102	TCP	54	443 → 49507 [ACK] Seq=1444 Ack=2750 Win=6

▶ Source: PCSSystemtec\_a3:2f:ab (08:00:27:a3:2f:ab)  
 Type: IPv4 (0x0800)  
 [Stream index: 0]  
 Padding: 000000000000

Internet Protocol Version 4, Src: 192.168.50.102, Dst: 192.168.50.100  
 0100 .... = Version: 4  
 .... 0101 = Header Length: 20 bytes (5)

0000 08 00 27 c9 0e da 08 00  
 0010 00 28 2a b9 40 00 80 06  
 0020 32 64 c1 63 01 bb 5b 99  
 0030 00 fe 84 11 00 00 00 00