

TASK

Nell'esercizio di oggi metteremo insieme le competenze acquisite finora. Lo studente verrà valutato sulla base della risoluzione al problema seguente.

Requisiti e servizi:

- Kali Linux □ IP 192.168.32.100
- Windows 7 □ IP 192.168.32.101
- HTTPS server: attivo
- Servizio DNS per risoluzione nomi di dominio: attivo

Traccia:

Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100.

Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

Ripetere l'esercizio, sostituendo il server HTTPS, con un server HTTP. Si intercetti nuovamente il traffico, evidenziando le differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS. Spiegare, motivandole, le principali differenze.

NUOVA CONFIGURAZIONE IP LINUX E WINDOWS

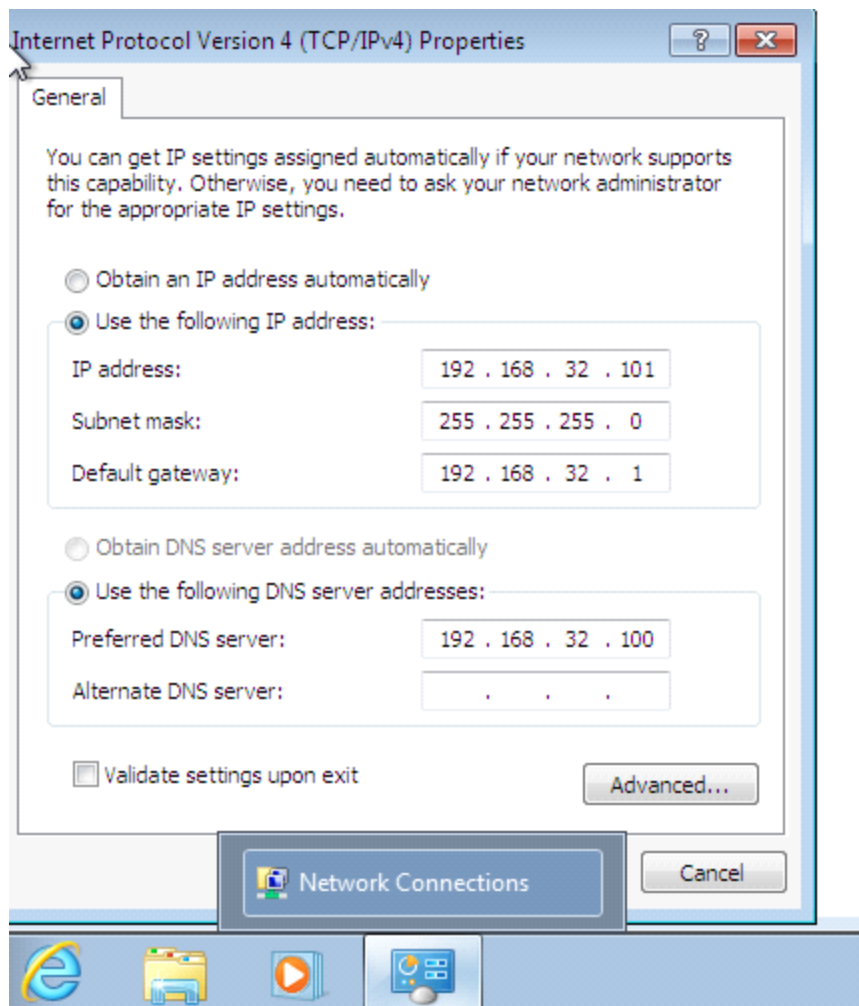
Come richiesto dalla traccia uno dei passi che abbiamo effettuato per la risoluzione di questa task, è quello di assegnare dei nuovi codici IP alle due macchine virtuali precedentemente create.

Linux : Per la procedura di configurazione del nuovo IP per la macchina Linux ci siamo recati nella shell accedendo come super user (sudo su), e con il comando nano /etc/network/interfaces siamo riusciti a cambiare l'indirizzo IP assegnandone uno nuovo con i seguenti valori:

```
(kali@kali)-[~]
└─$ sudo su
[sudo] password for kali:
(kali@kali)-[/home/kali]
└─# nano /etc/network/interfaces
```

```
# The loopback network interface
auto eth0
iface eth0 inet static
address 192.168.32.100
gateway 192.168.56.1
```

Windows: Per il cambio IP della macchina Windows è stato sufficiente cambiare l'ip tramite le opzioni del sistema operativo. Una volta impostato l'IP assegnatoci dalla traccia è stato necessario anche impostare il DNS che ci servirà nelle fasi successive della task



Di seguito riportata la fase di test ping per le due macchine :

```
C:\Users\Andrea Cuore>ping 192.168.32.100

Pinging 192.168.32.100 with 32 bytes of data:
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.32.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milliseconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Andrea Cuore>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=0.348 ms
 64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.897 ms
 64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.946 ms
 64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.224 ms
 64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.433 ms
 64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=0.858 ms
```

CONFIGURAZIONE SERVER CON INETSIM

Per simulare un server, e successivamente una richiesta HTTP e HTTPS ci siamo avvalsi del tool usato in precedenza INETSIM.

La configurazione di tale tool è avvenuta in 4 fasi

- 1. Disabilitazione dei servizi non richiesti

- 2. Impostare come IP di default il nostro IP macchina
- 3. Assegnare il DNS IP che avevamo utilizzato su windows (Indirizzo IP di kali ndr.)
- 4. Assegnare un hostname per la chiamata DNS

E' stato possibile modificare tutte queste parti, accedendo alla shell di linux come super user e modificare il file .conf , richiamando il comando nano /etc/inetsim/inetsim.conf

Di seguito riportate le immagini di configurazione:

```

# start_service
#
# 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

#####
# 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

#####
# service_run_as_user
#
# User to run services
#
# Syntax: service_run_as_user <username>
#
# Default: inetd
#
#service_run_as_user nobody

#####
# service_max_childs
#
# Maximum number of child processes (parallel connections)
# for each service
#

```

```

dns_default_ip 192.168.32.100
#####
# dns_default_hostname
#
# Default hostname to return with DNS replies
#
# Syntax: dns_default_hostname <hostname>
#
# Default: www
#
#dns_default_hostname somehost

#####
# dns_default_domainname
#
# Default domain name to return with DNS replies
#
# Syntax: dns_default_domainname <domain name>
#
# Default: inetsim.org
#
#dns_default_domainname some.domain

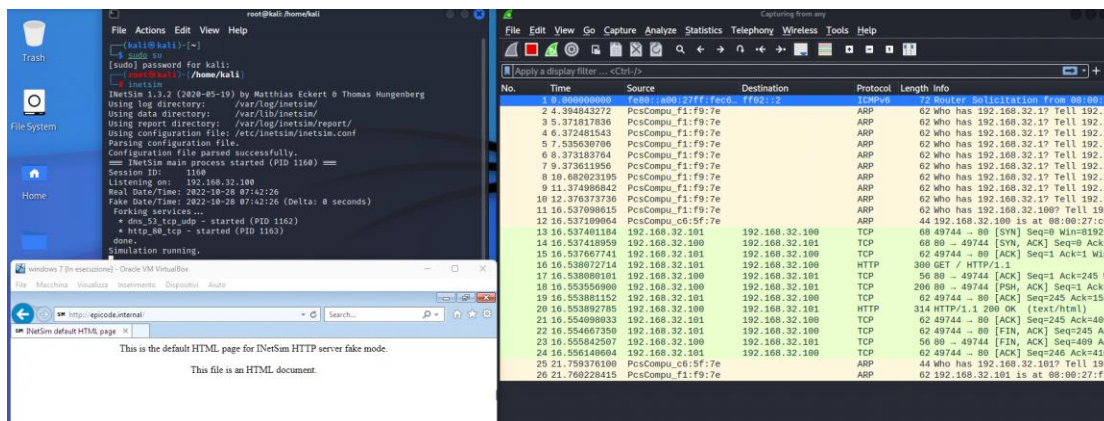
#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
dns_static www.epicode.internal 192.168.32.100
#####
# dns_version

```

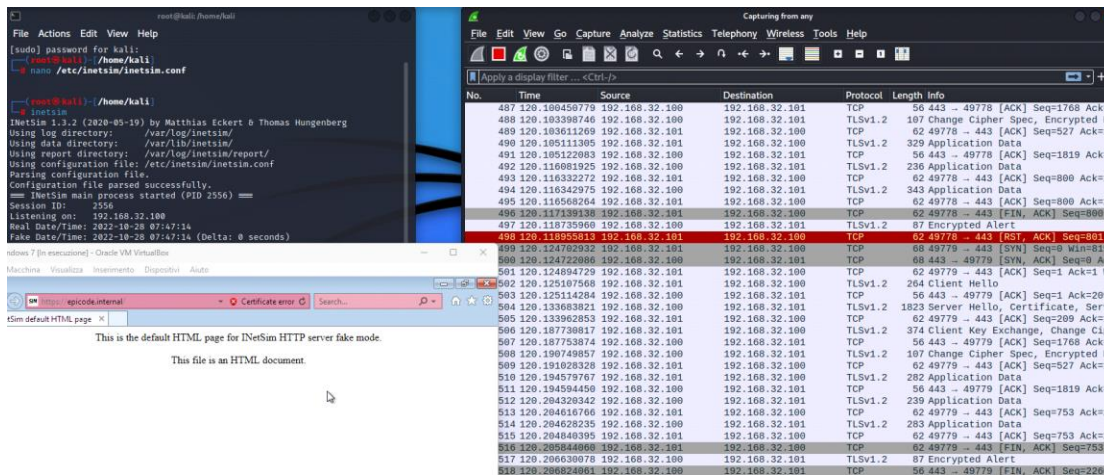
TEST SU WINDOWS E SNIFFING DI PACCHETTI:

Ultimata la configurazione del nostro file .conf è stato necessario richiamare il programma nella shell di comando per avviare il server e collegarci attraverso Windows 7.

SNIFFING PACCHETTI HTTP:



SNIFFING PACCHETTI HTTPS:



DIFFERENZA TRA HTTP E HTTPS

Le differenze che ho riscontrato nella fase di collegamento HTTP e HTTPS sono principalmente due, la prima riguarda il Flusso di pacchetti, infatti possiamo notare come la struttura di chiamata web cambia, Nella connessione in HTTP vediamo un three-way HandShake con GET diretto alla pagina HTTP, mentre se analizziamo lo sniffing di pacchetti HTTPS, possiamo notare anche i pacchetti di cifratura del protocollo TLS, con la relativa creazione della chiave di sessione, e relativo crypt del messaggio che possiamo analizzare più nel dettaglio nello screen qui sotto:

210	31.605204001	192.168.32.101	192.168.32.100	TCP	62	49173 -> 443 [RST, ACK] Seq=275 Ack=120 Win=0 Len=0
219	31.605421151	PcsCompu_f1:f9:7e	192.168.32.101	ARP	62	who has 192.168.32.17 Tell 192.168.32.101
220	32.293622718	PcsCompu_f1:f9:7e	192.168.32.101	ARP	62	who has 192.168.32.17 Tell 192.168.32.101
221	33.203910213	PcsCompu_f1:f9:7e	192.168.32.101	ARP	62	who has 192.168.32.17 Tell 192.168.32.101
222	34.830181629	192.168.32.101	192.168.32.100	TCP	60	49183 -> 443 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
223	34.830295078	192.168.32.100	192.168.32.101	TCP	60	443 -> 49183 [ACK] Seq=1383 Win=0 Len=0 MSS=1460 SACK_PERM=1 WS=128
224	34.830656418	192.168.32.101	192.168.32.100	TCP	62	49183 -> 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0
225	34.830962455	192.168.32.101	192.168.32.100	TLV1	195	Client Hello
226	34.830973227	192.168.32.100	192.168.32.101	TCP	56	443 -> 49183 [ACK] Seq=1 Ack=1 Win=64128 Len=0
227	34.83085621	192.168.32.100	192.168.32.101	TLV1	1379	Server Hello, Certificate, Server Key Exchange, Server Hello Done
228	34.847726172	192.168.32.101	192.168.32.100	TLV1	190	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
229	34.847744519	192.168.32.100	192.168.32.101	TCP	56	443 -> 49183 [ACK] Seq=1383 Win=64128 Len=0
230	34.848157855	192.168.32.100	192.168.32.101	TLV1	115	Change Cipher Spec, Encrypted Handshake Message
231	34.848494994	192.168.32.101	192.168.32.100	TCP	62	49183 -> 443 [FIN, ACK] Seq=274 Ack=1383 Win=64256 Len=0
232	34.850637774	192.168.32.100	192.168.32.101	TLV1	93	Encrypted Alert

Altra differenza che possiamo notare è legata alla "Non autenticità del certificato TLS", ovvero il nostro certificato TLS non disponendo di una firma autentica viene riconosciuto dal nostro

Browser come "Sito non sicuro", e per tale ragione il browser cercherà di bloccarci nell'accesso.

