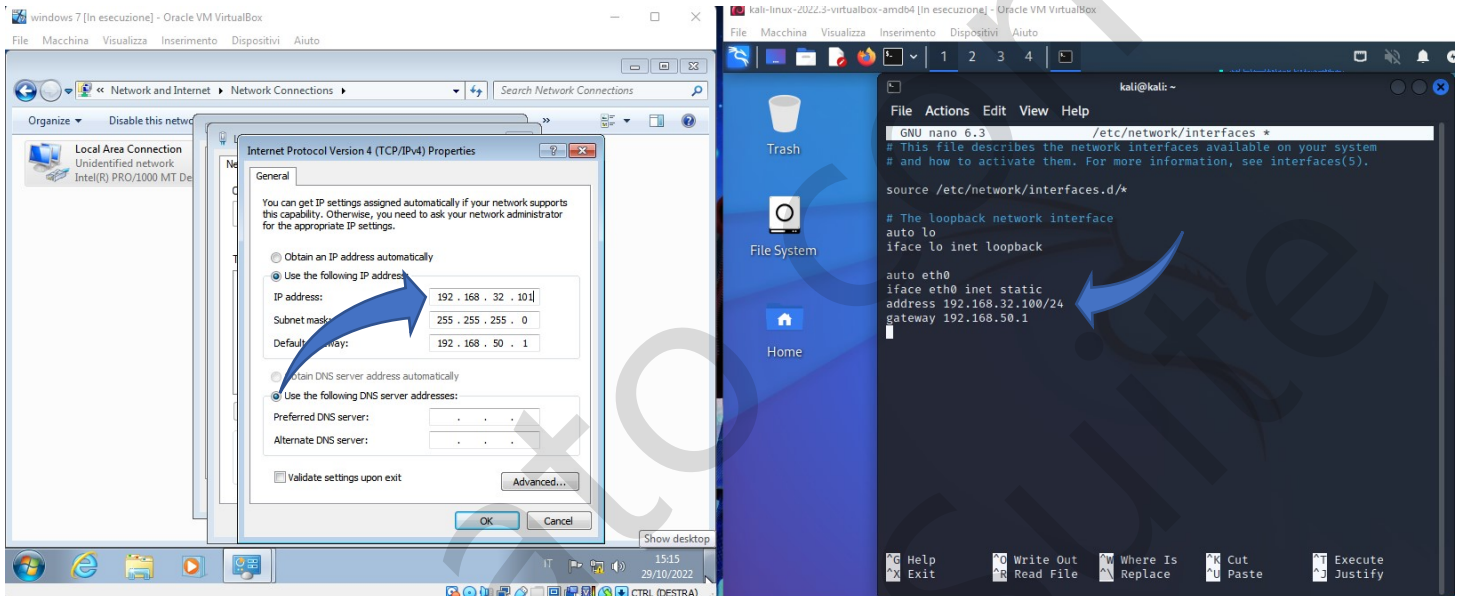


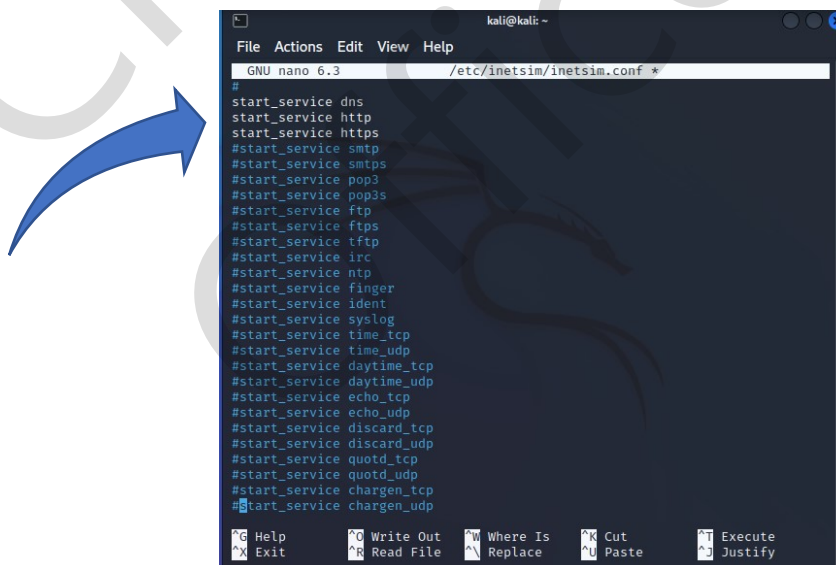
Claudio De Cicco

29/10/2022

Report del Venerdì



Come prima richiesta andiamo a modificare i nuovi IP di windows 7 e di linux. (rispettivamente 192.168.32.101 e 192.168.32.100)



Successivamente da linux andiamo ad aprire la configurazione di Inetsim in modo da lasciare attivi solo i servizi di cui abbiamo bisogno. Sempre nella configurazione andiamo ad assegnare al "dns bind address" l'ip di linux e al "dns static", epicode.internal associandolo all'ip di linux (192.168.32.100)

```
#####
# 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 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
```

Ora , dopo aver avviato la simulazione con inetsim , possiamo vedere come il sito venga raggiunto correttamente e come dal rapporto di Wireshark si riesca a risalire ai MAC.

The screenshot shows a virtual machine environment. On the left, a web browser displays the default HTML page for the INetSim HTTP server. On the right, Wireshark is capturing network traffic. The packet list shows several ARP and TCP packets. The packet details pane for Frame 10 is expanded, showing the link-layer address type as Ethernet (1) and the source MAC address as 08:00:27:22:46:4f. A blue arrow points from the Wireshark packet list to the packet details pane.

MAC linux 08:00:27:b9:0d:44
MAC Win 08:00:27:22:46:4f

Successivamente ho registrato anche la comunicazione sostituendo il server in HTTPS. La differenza che possiamo notare è che ora abbiamo registrato una comunicazione che sfrutta il protocollo TLS che permette di crittografare lo scambio di dati conferendogli quindi maggiore sicurezza e privacy cosa che con server HTTP non avviene.

Apply a display filter ... <Ctrl-/>					
No.	Time	Source	Destination	Protocol	Length Info
7	9.585259770	PcsCompu_b9:0d:...		ARP	62 Who has 192.168.32.100? Tell
8	9.585272137	PcsCompu_22:46:...		ARP	44 192.168.32.100 is at 08:00:
9	9.585543727	192.168.32.101	192.168.32...	TCP	68 49452 → 443 [SYN] Seq=0 Win
10	9.585561355	192.168.32.100	192.168.32...	TCP	68 443 → 49452 [SYN, ACK] Seq=
11	9.585867672	192.168.32.101	192.168.32...	TCP	62 49452 → 443 [ACK] Seq=1 Ack
12	9.586131683	192.168.32.101	192.168.32...	TLSv1.2	273 Client Hello
13	9.586140801	192.168.32.100	192.168.32...	TCP	56 443 → 49452 [ACK] Seq=1 Ack
14	9.599990730	192.168.32.100	192.168.32...	TLSv1.2	1823 Server Hello, Certificate,
15	9.600445578	192.168.32.101	192.168.32...	TCP	62 49452 → 443 [ACK] Seq=218 A
16	9.626619672	192.168.32.101	192.168.32...	TLSv1.2	374 Client Key Exchange, Change
17	9.626658235	192.168.32.100	192.168.32...	TCP	56 443 → 49452 [ACK] Seq=1768
18	9.631423083	192.168.32.100	192.168.32...	TLSv1.2	107 Change Cipher Spec, Encrypt
19	9.631690904	192.168.32.101	192.168.32...	TCP	62 49452 → 443 [ACK] Seq=536 A
20	9.642361498	PcsCompu_b9:0d:...		ARP	62 Who has 192.168.50.1? Tell
21	10.448867839	PcsCompu_b9:0d:...		ARP	62 Who has 192.168.50.1? Tell
22	11.450577769	PcsCompu_b9:0d:...		ARP	62 Who has 192.168.50.1? Tell

▶ Frame 19: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface any, id 0
 ▾ Linux cooked capture v1
 Packet type: Unicast to us (0)
 Link-layer address type: Ethernet (1)
 Link-layer address length: 6
 Source: PcsCompu_b9:0d:44 (08:00:27:b9:0d:44)