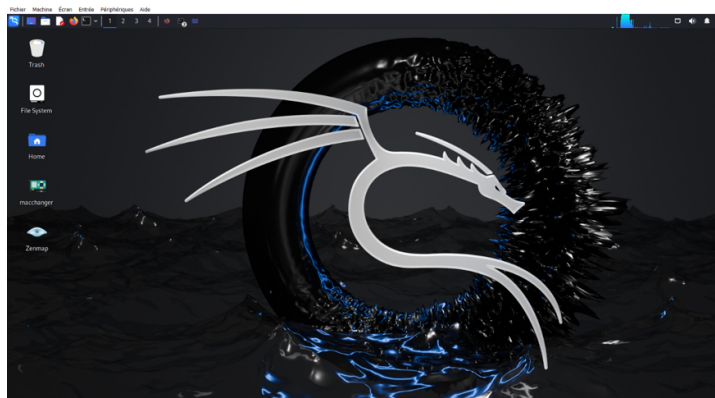


TP Découverte Kali Linux – Bloc 3 – JOBARD Guillaume – 2023/2024 – Mewo

Objectif : Découvrir l'intérêt de sécuriser correctement une machine sous Windows ou Linux et savoir se protéger en se mettant à la place de l'attaquant via Kali Linux.



Interface de kali linux

Introduction

Ce TP vise à nous initier à la sécurité des systèmes informatiques en utilisant Kali Linux. Nous allons apprendre à utiliser des outils pour tester et sécuriser des machines sous Windows et Linux.

En se mettant dans la peau d'un attaquant, nous découvrirons comment les vulnérabilités peuvent être exploitées et quelles mesures prendre pour s'en protéger. Nous verrons notamment comment changer une adresse MAC et utiliser des outils de scan réseau.

L'objectif est de mieux comprendre les enjeux de la sécurité informatique et d'apprendre à protéger efficacement nos systèmes. (Des parties extrait de goole)

1. Configuration des Machines Virtuelles (VM)

Kali Linux VM

1. Identifier l'adresse IP et l'adresse MAC

Préparation des Machines Virtuelles

1. Vérification de la machine virtuelle Kali Linux

- Adresse IP et MAC :
 - Commande pour afficher l'adresse IP : `ip addr show`
 - Commande pour afficher l'adresse MAC : `ip link show`

```
File Actions Edit View Help
└─$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 08:00:27:97:34:22 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.172/24 brd 192.168.1.255 scope global dynamic noprefixroute
        valid_lft 86394sec preferred_lft 86394sec
    inet6 2a02:8429:1:5301:a00:27ff:fe97:3422/64 scope global temporary dynamic
        valid_lft 744sec preferred_lft 744sec
    inet6 2a02:8429:1:5301:a00:27ff:fe97:3422/64 scope global dynamic mngtmpa
        valid_lft 744sec preferred_lft 744sec
    inet6 fe80::a00:27ff:fe97:3422/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:69:6b:38:a2 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever

(magatteseye@kali)-[~]
└─$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:97:34:22 brd ff:ff:ff:ff:ff:ff
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default
    link/ether 02:42:69:6b:38:a2 brd ff:ff:ff:ff:ff:ff

(magatteseye@kali)-[~]
└─$
```

2. Vérification de la machine virtuelle Windows

- Adresse IP et MAC :
 - Commande pour afficher l'adresse IP : `ipconfig`
 - Commande pour afficher l'adresse MAC : `getmac`

```

C:\Users\seve>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . : 
    IPv6 Address. . . . . : 2a02:8429:1:5301:4947:c9bf:e028:2180
    Temporary IPv6 Address. . . . . : 2a02:8429:1:5301:65da:2aca:64eb:e422
    Link-local IPv6 Address . . . . . : fe80::4947:c9bf:e028:2180%4
    IPv4 Address. . . . . : 192.168.1.248
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::1%4
                                192.168.1.1

C:\Users\seve>getmac

Physical Address      Transport Name
=====
08-00-27-E6-39-9C     \Device\Tcpip_{1C2212F6-E8F5-4BE2-B8BE-15C88F9A42D0}

C:\Users\seve>_

```

3. Vérification de la machine virtuelle Linux

- Adresse IP et MAC :
 - Commande pour afficher l'adresse IP : `ip addr`
 - Commande pour afficher l'adresse MAC : `ip link show`

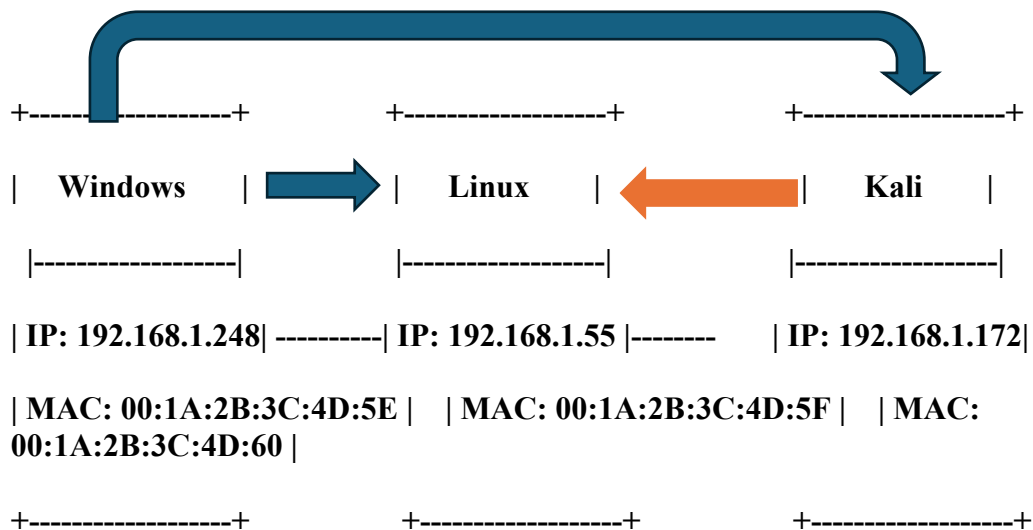
```

debian@debian:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:bd:d5:b0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.55/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 86368sec preferred_lft 86368sec
    inet6 2a02:8429:1:5301:5d0b:8949:ec51:40f4/64 scope global temporary dynamic
        valid_lft 720sec preferred_lft 720sec
    inet6 2a02:8429:1:5301:a00:27ff:febd:d5b0/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 720sec preferred_lft 720sec
    inet6 fe80::a00:27ff:febd:d5b0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
debian@debian:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:bd:d5:b0 brd ff:ff:ff:ff:ff:ff
debian@debian:~$ █

```

4. Vérification de la communication entre les machines

Une petite shema



ping 192.168.56.102 # Depuis Kali vers Linux (flèche orange)

ping 192.168.56.100 # Depuis Windows/Linux vers Kali (flèche bleue)

- **Ping** : Commande pour tester la connectivité :

ping 192.168.1.55 # Depuis Kali vers Linux

```
(magatteseye@kali)-[~]
$ ping 192.168.1.55
PING 192.168.1.55 (192.168.1.55) 56(84) bytes of data.
64 bytes from 192.168.1.55: icmp_seq=1 ttl=64 time=1.79 ms
64 bytes from 192.168.1.55: icmp_seq=2 ttl=64 time=2.30 ms
64 bytes from 192.168.1.55: icmp_seq=3 ttl=64 time=1.33 ms
64 bytes from 192.168.1.55: icmp_seq=4 ttl=64 time=2.46 ms
64 bytes from 192.168.1.55: icmp_seq=5 ttl=64 time=1.51 ms
64 bytes from 192.168.1.55: icmp_seq=6 ttl=64 time=1.54 ms
64 bytes from 192.168.1.55: icmp_seq=7 ttl=64 time=0.886 ms
64 bytes from 192.168.1.55: icmp_seq=8 ttl=64 time=1.77 ms
64 bytes from 192.168.1.55: icmp_seq=9 ttl=64 time=1.01 ms
64 bytes from 192.168.1.55: icmp_seq=10 ttl=64 time=0.860 ms
^C
— 192.168.1.55 ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9021ms
rtt min/avg/max/mdev = 0.860/1.544/2.464/0.526 ms
```

ping 192.168.1.172 # Depuis Windows vers kali

```

C:\Users\seye>ping 192.168.1.172

Pinging 192.168.1.172 with 32 bytes of data:
Reply from 192.168.1.172: bytes=32 time=1ms TTL=64
Reply from 192.168.1.172: bytes=32 time=1ms TTL=64
Reply from 192.168.1.172: bytes=32 time=5ms TTL=64
Reply from 192.168.1.172: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.172:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 5ms, Average = 2ms

C:\Users\seye>

```

ping 192.168.1.172 # Depuis linux vers Kali

```

lebian@debian:~$ ping 192.168.1.172
PING 192.168.1.172 (192.168.1.172) 56(84) bytes of data.
54 bytes from 192.168.1.172: icmp_seq=1 ttl=64 time=1.67 ms
54 bytes from 192.168.1.172: icmp_seq=2 ttl=64 time=0.657 ms
54 bytes from 192.168.1.172: icmp_seq=3 ttl=64 time=1.22 ms
54 bytes from 192.168.1.172: icmp_seq=4 ttl=64 time=0.476 ms
^C
--- 192.168.1.172 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3037ms
rtt min/avg/max/mdev = 0.476/1.003/1.666/0.469 ms
lebian@debian:~$ █

```

1.
 - **Traceroute** : Commande pour tracer la route :

traceroute 192.168.1.55 # Linux

```

debian@debian:~$ traceroute 192.168.1.248
traceroute to 192.168.1.248 (192.168.1.248), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
debian@debian:~$ █

```

tracert 192.168.1.248 # Windows

```

C:\Users\seye>tracert 192.168.1.172

Tracing route to 192.168.1.172 over a maximum of 30 hops
  1    <1 ms    <1 ms    <1 ms    192.168.1.172

Trace complete.

C:\Users\seye>

```

- **NSLookup** : Commande pour résoudre les noms de domaine :

nslookup 192.168.1.172 windows

```
C:\Users\seye>nslookup 192.168.1.172
server: vip-dns-gp-primary.dns.sfr.net
address: 109.0.66.10

*** vip-dns-gp-primary.dns.sfr.net can't find 192.168.1.172: Non-existent domain

C:\Users\seye>nslookup 192.168.1.55
server: vip-dns-gp-primary.dns.sfr.net
address: 109.0.66.10

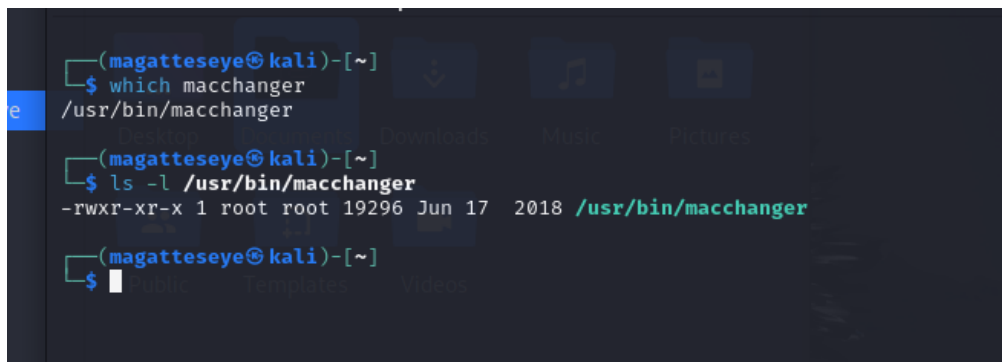
*** vip-dns-gp-primary.dns.sfr.net can't find 192.168.1.55: Non-existent domain

C:\Users\seye>
```

Expérimentation avec macchanger

1. J'ouvre un terminal sur votre machine Kali Linux.
2. Je tape la commande suivante pour trouver l'emplacement de `macchanger` :

```
which macchanger
```



```
(magatteseye@kali)-[~]
$ which macchanger
/usr/bin/macchanger

(magatteseye@kali)-[~]
$ ls -l /usr/bin/macchanger
-rwxr-xr-x 1 root root 19296 Jun 17 2018 /usr/bin/macchanger

(magatteseye@kali)-[~]
$
```

`/usr/bin/macchanger` : Cela indique que `macchanger` se trouve dans le répertoire `/usr/bin`.

Ensuite J'affiche la documentation de `macchanger` pour comprendre son utilisation :

```
man macchanger
```

```
File Actions Edit View Help
$ macchanger -h
GNU MAC Changer
Usage: macchanger [options] device

-h, --help                Print this help
-V, --version             Print version and exit
-s, --show                Print the MAC address and exit
-e, --ending              Don't change the vendor bytes
-a, --another             Set random vendor MAC of the same kind
-A,                       Set random vendor MAC of any kind
-p, --permanent          Reset to original, permanent hardware MAC
-r, --random              Set fully random MAC
-l, --list[=keyword]      Print known vendors
-b, --bia                 Pretend to be a burned-in-address
-m, --mac=XX:XX:XX:XX:XX:XX
  --mac XX:XX:XX:XX:XX:XX Set the MAC XX:XX:XX:XX:XX:XX

Report bugs to https://github.com/alobbs/macchanger/issues

(magatteseye@kali)-[~/Desktop]
$ which macchanger
/usr/bin/macchanger

(magatteseye@kali)-[~/Desktop]
$ man macchanger

(magatteseye@kali)-[~/Desktop]
```

Changer l'adresse MAC :

- Pour changer l'adresse MAC de manière aléatoire, j'utilise la commande suivante :

```
sudo macchanger -r eth0
```

- sudo : Permet d'exécuter la commande avec les privilèges administrateur.
- macchanger : Le nom de l'application.
- -r : Option pour générer une adresse MAC aléatoire.
- eth0 : Le nom de l'interface réseau que je souhaite modifier (à adapter selon mon configuration, j'utilise `ip addr` pour identifier l'interface correcte).

```
(magatteseye@kali)-[~/Desktop]
$ sudo macchanger -r eth0
[sudo] password for magatteseye:
Current MAC: 08:00:27:97:34:22 (CADMUS COMPUTER SYSTEMS)
Permanent MAC: 08:00:27:97:34:22 (CADMUS COMPUTER SYSTEMS)
New MAC: 22:74:18:59:8d:0b (unknown)

(magatteseye@kali)-[~/Desktop]
$
```

1. Après expérimentations du documents
 - **Temps nécessaire** : Moins d'une minute.
 - **Enjeux/Dangers** :
 - **Usurpation d'identité** : Pour changer l'adresse MAC peut permettre de contourner des filtres d'adresse MAC sur un réseau.
 - **Difficulté de traçage** : Les administrateurs réseau peuvent avoir du mal à suivre les activités d'une machine spécifique.

- **Protection :**
 - **Surveillance des logs réseau :** Pour détecter les changements d'adresse MAC.
 - **Utilisation de systèmes de détection d'intrusion (IDS) :** Pour alerter en cas de changement suspect.

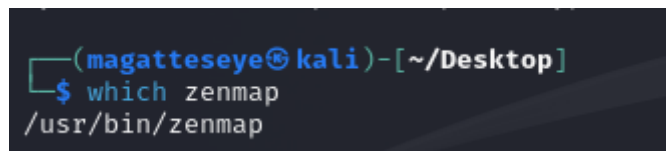
Expérimentation avec zenmap-kbx

Ce logiciel n'était pas disponible sur kali je l'ai installer grâce à un tuto sur YouTube:
Voici le lien tuto : https://youtu.be/EGxzu_IycCM

1. Localisation de zenmap-kbx

- Pour localiser zenmap ,je tape :

```
which zenmap
```



/usr/bin/zenmap : Cela indique que zenmap se trouve dans le répertoire /usr/bin.

▪

2. Utilisation de zenmap-kbx

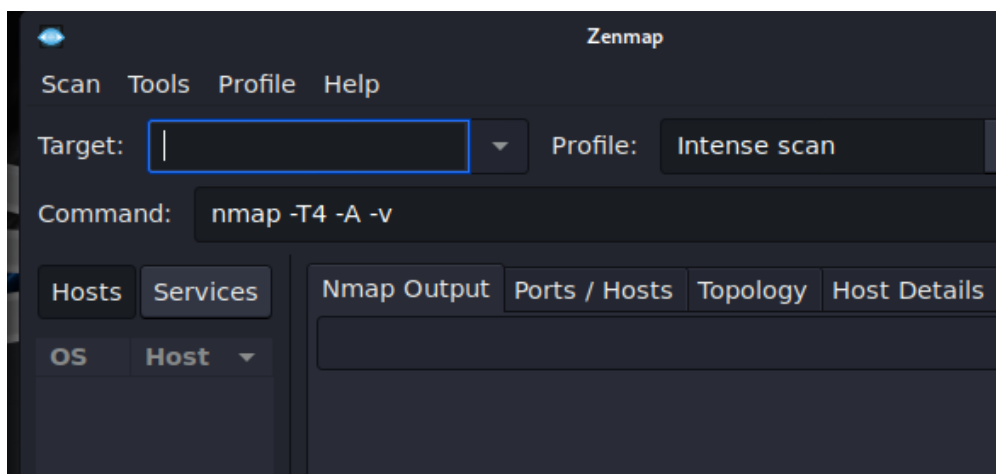
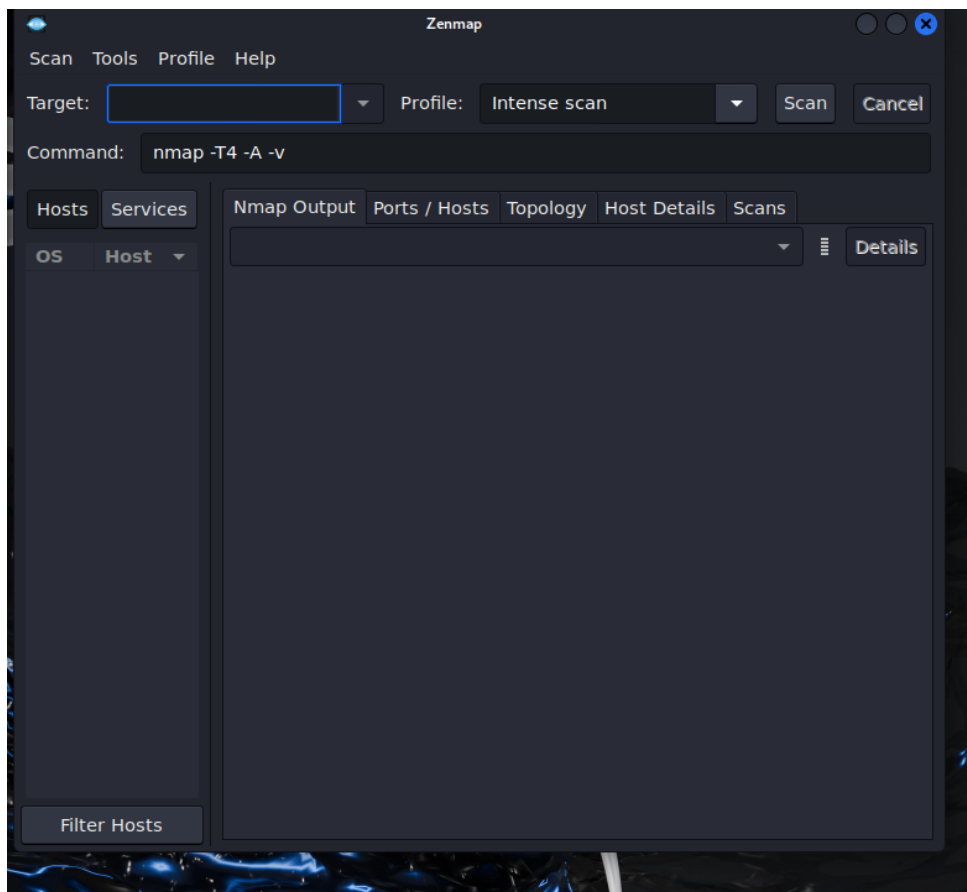
- J'ouvre Zenmap , je tape :

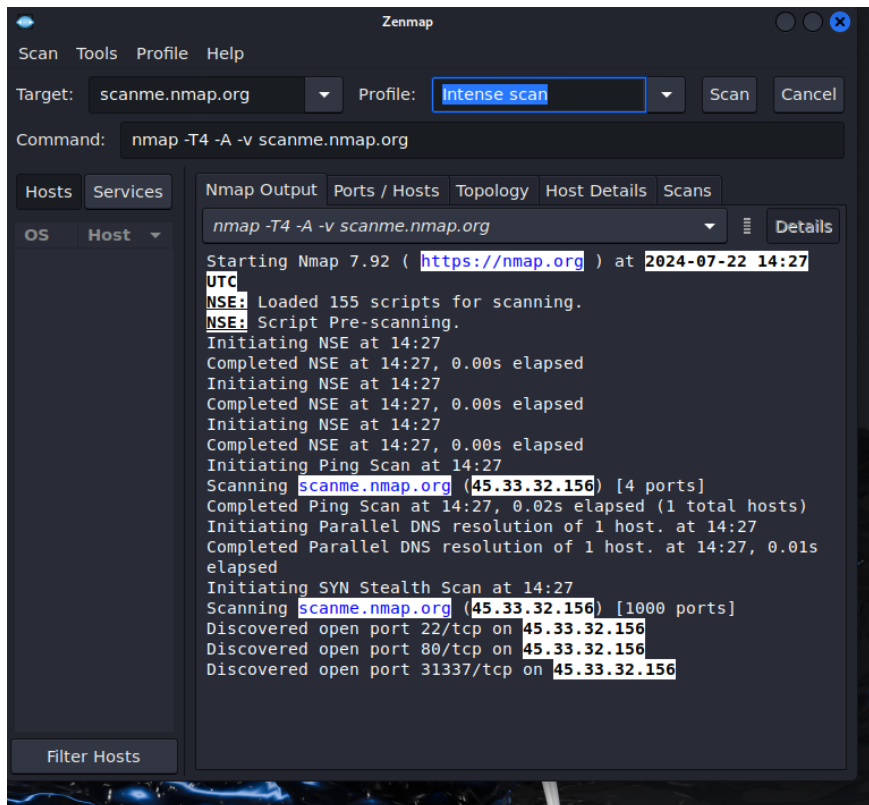
```
sudo zenmap
```

- **Documentation :**
 - J'affiche la documentation, je tape:

```
man zenmap
```

- **Expérimentation avec scanme.nmap.org :**
 - J'entre `scanme.nmap.org` dans le champ "Target" et sélectionne un profil de scan.
- **Application derrière zenmap-kbx :** Zenmap est l'interface graphique pour Nmap, un outil de scan de réseau.





- **Fonctionnalités :**
 - **Identification des systèmes d'exploitation :** Pour déterminer quel OS tourne sur les machines.
 - **Détection de vulnérabilités :** Pour identifier les failles potentielles dans le réseau.
- **Enjeux/Dangers :**
 - **Utilisation pour des attaques réseau :** Les attaquants peuvent utiliser ces informations pour cibler des vulnérabilités spécifiques.
 - **Perturbations réseau :** Les scans peuvent parfois causer des ralentissements ou des déconnexions.
- **Protection :**
 - **Filtrage des ports et des adresses IP :** Pour configurer les pare-feu pour limiter les scans de ports.
 - **Surveillance et analyse des logs :** Pour vérifier régulièrement les logs pour repérer les activités suspectes.

Conclusion

Ce TP nous a permis de comprendre l'importance de sécuriser nos systèmes en utilisant des outils de Kali Linux pour simuler des attaques. Nous avons exploré des applications comme macchanger et zenmap, et vérifié la connectivité entre différentes machines virtuelles. Cette expérience nous a montré comment identifier et résoudre des problèmes de sécurité, tout en soulignant la nécessité de mettre en place des mesures de protection robustes pour défendre nos réseaux contre les menaces.

