

Torii Security



19/06/2024

RAPPORT DE STAGE DE LICENCE INFORMATIQUE

Réalisation d'une box de pentest

Réalisé par : Massinissa BRAHIMI

Tuteur responsable : Damien BOLUS

Encadré par : Said ABDEDDAIM

Année Universitaire : 2023/2024

Tables des matières

Glossaire	3
Remerciement	5
1. Introduction	6
2. Présentation de l'entreprise	7
3. Contexte du stage	8
3.1. Présentation du projet :	8
3.2. Contexte du projet :	8
3.3. Méthodologie utilisée:	8
3.4. Planning de mon stage :	9
4. Réalisation du projet.....	10
4.1. Exploration du projet et mise en place de l'environnement de travail	12
4.1.1. Diagnostic VPN.....	13
4.2. Automatisation des services	15
4.3. Chiffrement d'une partition de disque.....	17
4.4. Automatisation de l'installation de l'OS	18
4.5. Problèmes rencontrés:.....	20
1) Erreur d'installation de GRUB:	20
2) Non-détection du fichier preseed:	20
3) Problème de partitionnement:	24
4) Paquets de base manquants :	24
4.6. Mesures de sécurité mises en place	26
4.7. Automatisation de post installation avec Ansible.....	29
4.8. Tests et validation.....	33
6. Suivis du projet :	35
7. Mission de pentest :	36
7.1. Enseignements tirés de la mission :	36
7.2. Perspectives personnelles :	37
8. Bilan:	38
8.1. Compétences techniques :	38
8.2. Compétences organisationnelles :	38
9. Conclusion de stage:	40

Tables de figures :

Figure 1 : Planning	9
Figure 2: Planning de la semaine du 27 au 31 Mai.....	10
Figure 3: Schéma illustrant mon environnement de travail	13
Figure 4: connexion ssh établie.....	14
Figure 5 : Activation de SSH	15
Figure 6: Démarrage du service SSH.....	15
Figure 7: Activation du service Vpn	16
Figure 8: démarrage et vérification du service	16
Figure 9: Exemple d'utilisation de "dd"	20
Figure 10: Installation de mkisofs	21
Figure 11: création de répertoire temporaire	21
Figure 12: Montage de l'ISO	21
Figure 13: copie de fichiers vers le répertoire temporaire	21
Figure 14: Modification du "grub.cfg"	22
Figure 15: Checksum du grub.cfg	22
Figure 16: Mise à jour dans le "md5sum.txt"	22
Figure 17: modification du chemin du fichier preseed.....	22
Figure 18: menu.cfg	23
Figure 19: Création de l'ISO	23
Figure 20: Rendre l'ISO bootable	23
Figure 21: Rendre l'image ISO isohybrid.....	23
Figure 22 : recette de partitionnement.....	24
Figure 23: Sélection des paquets	25
Figure 24: règles autorisant le flux ssh	27
Figure 25: mot de passe configuré	28
Figure 26: garder que le disque comme option de démarrage	29
Figure 27: exemple de tâches ansible	30
Figure 28: activation de ssh avec ansible	30
Figure 29: Activation du pare-feu	30
Figure 30: Autorisé le trafic ssh entrant	31
Figure 31: Fichier inventaire.....	31
Figure 32: Commande d'exécution du fichier Yaml sur les hosts spécifiés	32
Figure 33: Cahier de test.....	33
Figure 34: solution apportée pour le problème de dépôt	34

Glossaire

Pour les termes techniques que j'ai utilisés, veuillez consulter les références fournies ici. Si un terme spécifique n'est pas mentionné, je vous recommande vivement de consulter la documentation appropriée et/ou de rechercher sur Internet pour obtenir plus d'informations

Tests de pénétration (Pentest):

évaluations de sécurité proactives effectuées sur un système informatique, une infrastructure réseau ou une application pour identifier et exploiter activement les vulnérabilités potentielles.

Clé USB bootable :

Une clé USB configurée pour démarrer un système d'exploitation.

Image ISO :

Un fichier qui contient une copie exacte d'un disque optique.

Chiffrement :

C'est un procédé de cryptographie grâce auquel on souhaite rendre la compréhension d'un document impossible à toute personne qui n'a pas la clé de chiffrement.

SSH :

Protocole permettant d'établir une connexion sécurisée à distance.

VPN :

Réseau privé virtuel pour sécuriser la navigation internet permettant ainsi d'assurer la confidentialité des données qui transitent et de sécuriser les communications avec d'autres réseaux ou appareils distants.

UNetbootin :

Outil pour créer des clés USB bootables à partir d'images ISO.

Package :

Ensembles de fichiers nécessaires pour installer des logiciels sur un système.

systemd :

Gestionnaire de systèmes et de services pour Linux.

cryptsetup :

Outil de gestion du chiffrement des disques sur Linux.

Système de fichiers :

Structure qui permet de stocker et organiser les fichiers sur un disque.

Intégrité :

Assurance que les données n'ont pas été altérées.

CTF :

Exercices de cybersécurité dans lesquels les participants, individuellement ou en équipe, sont mis au défi de trouver et d'exploiter des vulnérabilités dans un système pour capturer un "drapeau"

Rolling release :

Modèle de distribution logiciel où les mises à jour sont continues.

Dépôt en ligne :

Un serveur distant sur Internet qui héberge les paquets logiciels officiels et les mises à jour validées par les développeurs, pour installation (ex: dépôt Kali Linux).

Dépôt local:

C'est un emplacement sur le réseau local ou sur le système de fichier local qui stocke des paquets logiciels personnalisés, des mises à jour spécifiques ou des versions modifiées des logiciels pour une utilisation interne ou un déploiement localisé.

Open-source: (source ouverte)

Les logiciels dont le code source est accessible publiquement, permettant ainsi à quiconque de l'étudier, de le modifier et de le distribuer librement.

Fichier de service :

C'est un fichier de configuration utilisé dans les systèmes Linux pour définir et contrôler le comportement d'un service ou d'une application. Il contient des instructions sur la manière dont le service doit être démarré, arrêté, et configuré,

Remerciement

Avant tout propos, je tiens à exprimer ma profonde gratitude à mes professeurs de la faculté, qui m'ont transmis les connaissances nécessaires pour réaliser ce stage. Leurs enseignements ont été fondamentaux pour aborder cette expérience professionnelle avec assurance et compétence. Je remercie également M. Jean Gabriel Luque, mon responsable de formation, pour sa disponibilité constante et ses réponses à toutes mes questions, ainsi que M. ABDEDDAIM Said pour son encadrement tout au long de cette période.

Mes remerciements les plus sincères vont à M. Damien BOLUS, fondateur de l'entreprise et mon tuteur de stage, pour m'avoir offert l'opportunité de travailler au sein de son entreprise et pour son encadrement exceptionnel. Grâce à lui, j'ai pu améliorer ma réflexion, ma créativité et acquérir un avant-goût précieux de mon futur métier de pentester. Son soutien m'a permis de développer un solide bagage pratique dans ce domaine.

Je tiens aussi à remercier toute l'équipe de *TORII SECURITY* pour leur accueil chaleureux et leur soutien, je leur suis très reconnaissant pour leur accompagnement et leurs conseils qu'ils m'ont accordée tout au long de cette période.

Pour finir, je dois beaucoup de gratitude à mes parents et ma famille pour leur investissement dans les actions que j'entreprends.

1. Introduction

Dans ce rapport de stage, je souhaite partager mon expérience chez *Torii Security*, un cabinet de conseil et d'expertise en cybersécurité. Mon stage, qui s'est déroulé du **23 avril au 19 juin 2024** sous la direction de Damien BOLUS, m'a permis de plonger dans le monde professionnel au-delà de mes connaissances techniques acquises précédemment, offrant des perspectives inédites par rapport à mes études.

Je me suis investi dans le domaine de la cybersécurité au sein de *Torii Security*, travaillant sur le développement d'une box de pentest. Mon rôle consistait à concevoir et réaliser cette box, une solution visant à faciliter les missions de tests de pénétration à distance et à améliorer la qualité de service de l'entreprise. En plus de cette mission principale, j'ai également participé à des missions de pentest, mes toutes premières expériences dans le monde réel de la cybersécurité.

Le plan de mon rapport s'articule principalement en 5 parties qui sont :

- Présentation du stage
- Réalisation du projet
- Test et validation
- Suivi du projet
- Bilan

2. Présentation de l'entreprise

Torii Security est un cabinet de conseil et d'expertise en cybersécurité créé en 2016, lorsque Damien BOLUS, un hacker éthique et consultant sénior en sécurité des systèmes d'information, a décidé de lancer une entreprise spécialisée dans la protection du patrimoine numérique des entreprises. Sa vision est d'améliorer durablement le niveau de protection et de résilience de ses clients, qu'il s'agisse d'industries, de PME ou de grands comptes.



Son local est établi à Sotteville-lès-Rouen, en région Normandie.

Torii Security se distingue avec son expertise en cybersécurité, labellisée France Cybersecurity, l'entreprise se spécialise dans les audits et les tests d'intrusion qui constitue l'activité principale de l'entreprise et la formation en sécurité informatique. Sa mission est de protéger les actifs numériques de ses clients en identifiant les risques, en anticipant les menaces et en proposant des solutions adaptées. En évaluant les failles de sécurité, en concevant des stratégies de défense et en formant les équipes, *elle* assure une protection fiable et proactive pour ses clients.

3. Contexte du stage

3.1. Présentation du projet :

Mon projet consiste à développer une solution pratique dans le domaine de la cybersécurité, une box de pentest. Cette solution vise à faciliter la réalisation des tests de pénétration à distance pour évaluer la sécurité des systèmes d'informations des entreprises.

Concrètement, cet implant est une machine qui utilise le système d'exploitation Kali Linux et que nous allons placer chez des clients afin de réaliser des opérations de tests de pénétration (pentest) à distance, cette méthode présente plusieurs avantages. Elle permet de réduire les frais de déplacement et d'optimiser l'efficacité opérationnelle en évitant le besoin constant d'être physiquement présent sur site.

De plus, dans certains cas, il est crucial d'être discret et de mener un pentest sans informer certains membres de l'entreprise.

Ne pas se déplacer garantit également une réponse rapide aux besoins de sécurité des clients et permet d'intervenir immédiatement en cas de découverte de failles critiques.

Grâce à cette solution, nous pouvons contrôler et gérer l'implant depuis notre local, offrant ainsi un service flexible et efficace pour effectuer des évaluations en temps réel de la sécurité des systèmes des clients et identifier les vulnérabilités potentielles.

3.2. Contexte du projet :

Le travail demandé s'inscrit dans un projet global visant à améliorer les services de pentest de *Torii Security*, la réalisation de ce projet a rencontré plusieurs contraintes spécifiques liées à sa nature notamment :

- **Sécurité de l'implant:** il est essentiel qu'il soit hautement sécurisé pour éviter tout accès non autorisé ou compromission potentielle.
- **Fiabilité :** l'implant doit garantir un fonctionnement fiable chez le client, minimisant ainsi les risques de pannes ou d'interruptions.
- **Protection des données :** la protection des données du client est primordiale, exigeant que toutes les informations collectées lors des tests doivent être sécurisées et traitées de manière confidentielle.

Je travaille seul sur ce projet, tout en bénéficiant de la supervision et des conseils de mon tuteur.

3.3. Méthodologie utilisée:

Pour garantir une bonne organisation dans le développement de mon projet, j'adopte une approche personnalisée, adaptée à ma situation de travail en solo. Mon tuteur,

qui me supervise, me laisse une certaine liberté dans l'organisation de mes tâches et de leur réalisation. Cependant, pour assurer une progression efficace, il intervient en m'expliquant les objectifs à atteindre à la fin de chaque étape. De plus, à la fin de chaque journée de travail, nous faisons le point sur l'avancement du projet, en discutant des réalisations accomplies et des éventuels obstacles rencontrés.

Concernant la planification à plus long terme, nous avons établi une routine hebdomadaire où nous faisons un bilan exhaustif de ce qui a été réalisé au cours de la semaine précédente, ainsi que des prochaines étapes à entreprendre et d'ajuster si nécessaire notre plan d'action en fonction des circonstances.

3.4. Planning de mon stage :

J'ai élaboré un plan pour offrir une visualisation claire des principales étapes sur lesquelles j'ai travaillé pendant une partie de mon stage qui montre les différentes phases de mon implication dans le projet, en soulignant les principaux objectifs atteints à chaque étape. Il illustre également la continuité de mon travail et la progression logique de mes responsabilités au fil du temps.

Tâches	Début	Durée(jours)	Finalisée
Prise en main du projet	23/04/2024	5	30/04/2024
automatisation de la connexion VPN	01/05/2024	2	03/05/2024
Chiffrement d'une Partition de Disque	06/05/2024	4	10/05/2024
Participation à une mission de Pentest	13/04/2024	2	14/05/2024
Sécurisation des ports physiques et logiques	15/05/2024	3	17/05/2024
Automatisation de l'Installation de Kali Linux avec un Fichier Preseed	20/05/2024	10	31/05/2024
Automatisation des Tâches de Post-installation avec Ansible	03/06/2024	3	06/06/2024
Sécurisation de la box	07/06/2024	1	07/06/2024
Phases de tests	10/06/2024	2	11/06/2024
Élaboration de la Documentation Projet	12/06/2024	5	19/06/2024

Figure1: plan de travail et progression des phases du projet

Pour organiser et suivre ma progression, j'ai utilisé un outil simple qui est le calendrier Outlook

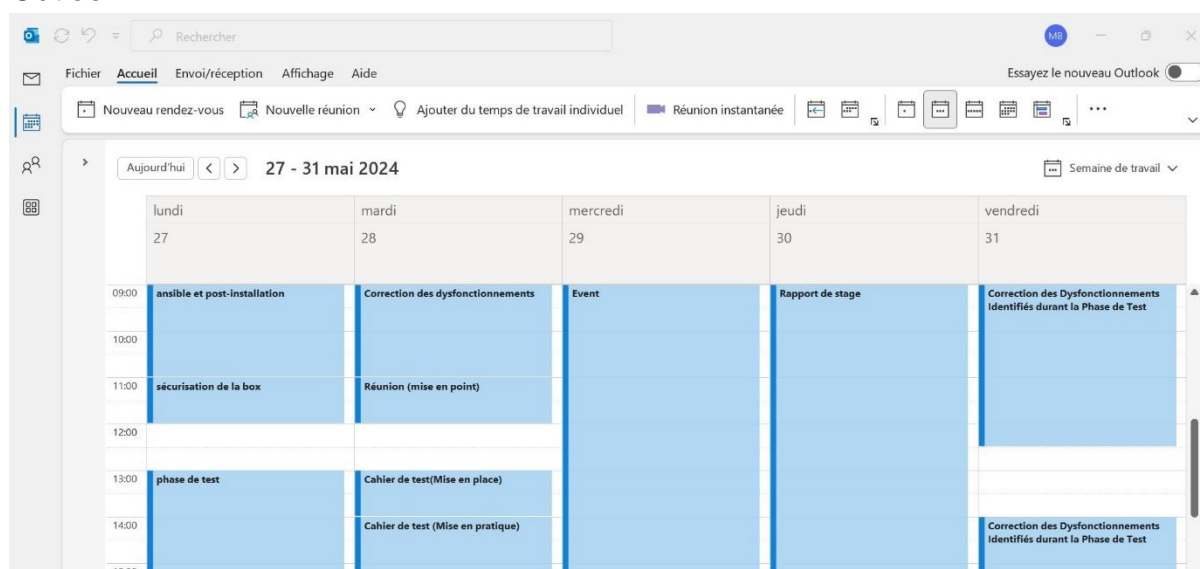


Figure 2: Planning de la semaine du 27 au 31 Mai

4. Réalisation du projet

Dans cette section, je vais vous conduire à travers le processus complet de développement de la « Box de Pentest ». Guidé par ma propre expérience et méthodologie, ce parcours détaillera les différentes étapes que j'ai suivies, les défis rencontrés, ainsi que les solutions que j'ai apportées pour les surmonter. En suivant cette progression chronologique, vous aurez un aperçu complet du travail accompli pour concevoir et mettre en œuvre cette solution innovante en cybersécurité, tout en gardant à l'esprit qu'il existe différentes approches possibles pour atteindre ces objectifs.

Voici le plan détaillé des étapes suivies dans ce projet :

1. Exploration du projet et mise en place de l'environnement de travail

Cette première étape consiste à comprendre les exigences du projet et à préparer l'environnement de travail nécessaire, incluant la création d'une clé USB bootable et l'installation de Kali Linux sur l'implant.

2. Automatisation des services

Dans cette étape, l'objectif est d'automatiser les services essentiels tels que SSH et OpenVPN, afin de garantir une connexion sécurisée et stable

3. Chiffrement d'une partition de disque

Je procède ici au chiffrement d'une partition de disque à l'aide de LUKS et cryptsetup, pour assurer la confidentialité des données sensibles stockées sur la box de pentest.

4. Automatisation de l'installation de l'OS

L'étape consiste à automatiser l'installation de Kali Linux en utilisant un fichier preseed. Cela permet de standardiser et de simplifier le processus d'installation sur plusieurs machines.

5. Mesures de sécurité mises en place

Cette étape se focalise sur la mise en place de diverses mesures de sécurité pour protéger la box de pentest, notamment la sécurisation des ports USB et des ports logiques , ainsi que la protection du BIOS.

6. Automatisation de la post-installation avec Ansible

Une fois l'OS installé, dans cette étape on utilise Ansible pour automatiser les configurations et installations postérieures, garantissant une configuration uniforme et sécurisée.

7. Tests et validation

Enfin, cette étape implique la réalisation de tests rigoureux pour valider le bon fonctionnement de la box de pentest, ainsi que la vérification de la sécurité et de l'efficacité des configurations mises en place.

Résultats des tests et ajustements de configuration :

L'ensemble des tests réalisés pour évaluer le fonctionnement global de la box de pentest et identifier d'éventuels besoins d'ajustement.

Suivis du projet :

À la fin du stage, qui reprend le projet? quels sont les documents remis pour la maintenance et l'évolution du projet?

Maintenant que le plan est établi, je peux passer à l'explication de chaque étape afin que vous puissiez comprendre le processus suivi dans la réalisation de ce projet.

4.1. Exploration du projet et mise en place de l'environnement de travail

Au début de cette phase initiale du projet, j'ai entrepris une exploration approfondie afin de bien cerner les exigences et les contours de la création de la station de pentest. Cette étape est cruciale pour définir clairement les objectifs à atteindre, identifier les contraintes techniques et opérationnelles, et esquisser le résultat final attendu.

Pour commencer, j'ai posé des questions pour comprendre le concept de la box de pentest :

- Q1: Concrètement c'est quoi une box de pentest ?
- Q2: Comment s'effectue une opération de pentest à distance?

Concernant la première question, en effectuant des recherches, j'ai découvert qu'une station de pentest est une machine Linux. Mon tuteur m'a indiqué que nous utiliserons la distribution Kali Linux, réputée dans le domaine de la sécurité offensive. Dans ce rapport, je vais donc travailler dans l'environnement Kali Linux.

Ensuite, j'ai entrepris de me renseigner sur la création d'une clé USB bootable à partir de l'ISO de Kali Linux et sur la manière de le faire sur une machine Kali (car auparavant, je ne l'avais fait que sur Windows avec Rufus (application portable gratuite et open-source pour Microsoft Windows qui permet de formater et de créer des clés USB bootables ou des Live USB.)).

Pour simplifier cette tâche, j'ai choisi d'utiliser le logiciel Unetbootin, qui offre une interface conviviale pour créer la clé USB bootable.

Une fois la clé USB préparée, j'ai procédé à l'installation de Kali Linux sur la machine destinée à devenir la box de pentest.

Quant à la deuxième question, pour effectuer une opération de pentest à distance, il est crucial de prendre d'abord le contrôle de la box que nous allons déposer chez le client. Pour cela, il est nécessaire de créer une connexion VPN, ce qui permet de créer un tunnel chiffré pour protéger la transmission des données. Ensuite, il faut utiliser le protocole SSH pour se connecter à distance et prendre le contrôle de la box.

Pour cette tâche, j'ai utilisé OpenVPN. J'ai installé un client OpenVPN fourni par mon tuteur sur la box et la machine principale pour faire une simulation de la configuration finale. Une fois les clients VPN installés, j'ai activé le service OpenVPN ainsi que le service SSH, qui est préinstallé sur Kali Linux, j'ai alors essayé de connecter ma machine principale à l'implant.

Cependant, j'ai rencontré un problème : la connexion ne s'effectuait pas.

4.1.1. Diagnostic VPN

Pour diagnostiquer ce problème, j'ai suivi plusieurs étapes :

- **Cartographie du réseau :** J'ai dessiné un plan partiel de l'environnement pour comprendre les serveurs et les plages réseaux existants. J'ai découvert qu'il y avait deux serveurs VPN et non pas un seul.

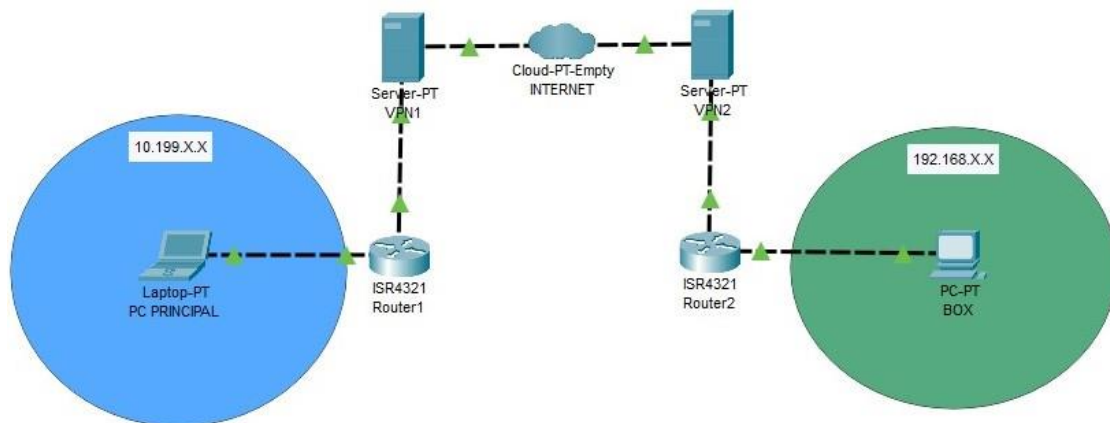


Figure 3: Schéma illustrant mon environnement de travail

- **Vérification du Routage :** J'ai vérifié s'il s'agissait d'un problème de routage en utilisant les commandes suivantes :

`$ ip route` sur la box

(montre des routes pour atteindre le réseau 10.X.X.X/24 via Serveur VPN 2)

`$route print` sur ma machine principale Windows.

(montre des routes pour atteindre le réseau 192.X.X.X/24 via Serveur VPN 1,)

(Ces deux commandes nous permettent de voir les routes configurées pour permettre la communication entre les deux machines dans ce cas et permettent de visualiser et de manipuler les tables de routage IP du système pour gérer le trafic réseau dans le cas général.)


J'ai pensé que le serveur vpn2 était éteint car j'ai effectué des tests de ping sur les deux serveurs vpn afin de vérifier leur connectivité j'ai découvert que le serveur vpn2 ne répondait pas.

J'ai donc signalé ce problème à mon tuteur en expliquant que le serveur vpn2 semblait être injoignable, il a vérifié l'état du serveur vpn2 et a constaté que tout était censé marcher il a alors redémarrer le serveur, ce qui a résolu le problème de connectivité.

Remarque :

Il est important de noter qu'un test de ping n'est pas toujours fiable quant à la disponibilité d'un serveur. Si le ping fonctionne, le serveur est disponible, mais si le ping échoue, cela ne prouve pas nécessairement que le serveur soit indisponible, car le service ping peut être bloqué pour éviter les spams par exemple.

À la suite de cela, j'ai pu établir une connexion SSH à travers le tunnel VPN précédemment monté entre les deux machines, ce qui a finalement fonctionné.



```
(kali@kali)-[~/Desktop]
$ ssh torii@192.168.254.3
torii@192.168.254.3's password:
Permission denied, please try again.
torii@192.168.254.3's password:
Linux kali 6.6.9-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.6.9-1kali1 (2024-01-08)
```

Figure 4: connexion ssh établie

4.2. Automatisation des services

Maintenant, que j'ai pu me connecter à la box, à chaque fois, je me pose une question cruciale : si je place cette box chez le client, est-ce que tout fonctionne correctement? Après une réflexion simple et avec l'aide de mon encadrant qui a orienté ma réflexion, j'en suis arrivé à la conclusion qu'il est nécessaire d'automatiser chaque service rencontré jusqu'à présent, à savoir le service OpenVPN et le SSH dans un premier temps.

Automatisation du service SSH:

Pour automatiser le démarrage du service SSH, il suffit d'exécuter les 2 commandes suivantes :

```
(kali㉿kali)-[~]  
$ sudo systemctl enable ssh  
[sudo] password for kali:  
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
```

Figure 5 : Activation de SSH

```
(kali㉿kali)-[~]  
$ sudo systemctl start ssh
```

Figure 6: Démarrage du service SSH

Ces commandes permettent de s'assurer que le service SSH démarre automatiquement à chaque démarrage de la machine.

Automatisation du service openVPN:

L'automatisation du service openVPN est un peu plus complexe et nécessite la création d'un fichier de service dans systemd. Voici les étapes que j'ai suivies pour automatiser la connexion au VPN :

- **Création du fichier de configuration OpenVPN:** J'ai placé le fichier de configuration OpenVPN fourni par mon tuteur dans le répertoire `/etc/openvpn/`
- **Création du fichier de service systemd:** J'ai créé un fichier de service pour OpenVPN dans le répertoire `/etc/systemd/system/`

- **Activation du service:** Ensuite, j'ai activé le service openVPN avec les commandes suivantes:

```
(torii@kali)-[~]
$ sudo systemctl enable openvpn-autoconnect.service
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-autoconnect.service → /etc/systemd/system/openvpn-autoconnect.service.

(torii@kali)-[~]
$ sudo systemctl daemon-reload
```

Figure 7: Activation du service Vpn

- **Démarrage et vérification du service:** j'ai démarré le service et vérifié qu'il fonctionne correctement avec les commandes suivantes:

```
(torii@kali)-[~]
$ sudo systemctl restart openvpn-autoconnect.service

(torii@kali)-[~]
$ sudo systemctl status openvpn
● openvpn.service - OpenVPN service
   Loaded: loaded (/usr/lib/systemd/system/openvpn.service; enabled; preset: >
   Active: active (exited) since Wed 2024-06-12 12:07:51 CEST; 24min ago
     Docs: man:openvpn(8)
    Main PID: 804 (code=exited, status=0/SUCCESS)
      CPU: 7ms

Jun 12 12:07:51 kali systemd[1]: Starting openvpn.service - OpenVPN service ...
```

Figure 8: démarrage et vérification du service

Une fois l'automatisation de ces deux services effectués, une question posée par mon encadrant m'a fait réaliser une autre contrainte importante.

Supposons que le pentest est terminé et les données rassemblées, ces dernières ne doivent pas traîner n'importe où et en clair. Cela signifie qu'il faut chercher un moyen de les chiffrer, mais sans chiffrer tout le disque, sinon le système demanderait un mot de passe au démarrage de la machine, ce qui ne fonctionnerait pas puisque la box sera chez le client et que nous n'est pas là pour le saisir à chaque redémarrage qu'on a besoin de faire.

Cette réflexion m'a conduit à l'étape suivante : le chiffrement d'une partition de disque.

4.3. Chiffrement d'une partition de disque

Pour chiffrer un disque, après des recherches sur le sujet, j'ai découvert qu'il fallait utiliser le standard associé au noyau Linux pour le chiffrement de disque, connu sous l'acronyme **LUKS** (Linux Unified Key Setup), ainsi que l'outil **cryptsetup** qui est un utilitaire open-source utilisé pour configurer de manière pratique le chiffrement de disque basé sur le module noyau dm-crypt.

Problème rencontré:

Lors de l'installation initiale, je n'avais pas laissé d'espace pour une partition dédiée au chiffrement.

Solution apportée:

Réinstaller entièrement le système d'exploitation sur la box, cette fois en prévoyant une partition spécifique pour le chiffrement avec **LUKS**.

Une fois le système réinstallé, j'ai procédé au chiffrement de cette partition.

Pour tester le bon fonctionnement du chiffrement, j'ai créé un fichier test contenant des données aléatoires et je l'ai placé dans la partition chiffrée. Ensuite, j'ai fermé et démonter la partition chiffrée et lors de la réouverture, il m'a demandé la passphrase, ce qui a confirmé que le chiffrement fonctionnait correctement.

Après avoir effectué ces démarches, notre box devrait normalement répondre aux attentes, bien que certaines nuances restent à prendre en compte, notamment en termes de sécurité, sujet que j'aborderai ultérieurement dans ce rapport. Cependant, en me posant à nouveau la question essentielle : sera-t-elle fonctionnelle chez le client?

La réponse, malgré quelques ajustements nécessaires, demeure positive.

Toutefois, réaliser ces étapes pour chaque nouveau client serait fastidieux, et cela ne correspond pas à nos objectifs.

Cette réflexion m'a amené à une phase plus complexe, nécessitant davantage de temps : l'automatisation du processus d'installation du système d'exploitation.

4.4. Automatisation de l'installation de l'OS

Étant donné que Kali Linux est une distribution basée sur Debian, j'ai opté pour l'utilisation du fichier **preseed** qui est un fichier de configuration utilisé par Debian pour automatiser l'installation du système d'exploitation plus concrètement, son principe est de contenir les réponses posées en temps normal par l'installateur Debian. Celui-ci ouvre normalement tout un tas de boîtes de dialogues selon un scénario bien précis.

L'intérêt du fichier preseed est de répondre à ces questions en amont. Si une boîte de dialogue dispose déjà d'une réponse, elle ne sera pas présentée à l'utilisateur.

c'est un simple fichier texte qui contient des chaînes de configuration qui prennent la forme suivante:

d-i module/paramètre type_de_données contenu_de_la_réponse.

Champ	Signification
d-i indique	qu'on s'adresse à l'installateur Debian (d-i)
module	indique quel est le module de l'installateur Debian concerné. Car l'installateur Debian fonctionne avec des modules. Par exemple un module s'occupe de la configuration réseau (netcfg), un autre du partitionnement (partman), etc.
paramètre	indique quelle est la variable qu'on souhaite renseigner.
type_de_données	indique quel est le type de la variable concernée (ex: string pour une chaîne de caractère, toggle pour cocher une case, etc.).
contenu_de_la_réponse	contient la valeur affectée à la variable.

Voici les étapes que j'ai suivies pour créer et éditer mon propre fichier preseed.

- **Création du fichier preseed:** J'ai créé un fichier preseed personnalisé selon nos besoins sur ma machine principale
- **Configuration des options de base:** J'ai commencé par configurer les options de base telles que la langue, le pays, le fuseau horaire, le clavier, les interfaces réseaux, la création d'utilisateurs, et les miroirs de dépôts.
- **Configuration du partitionnement automatique :** J'ai spécifié les options de partitionnement automatique
- **Configuration des packages :** J'ai spécifié les packages à installer automatiquement pendant le processus d'installation.
- **Automatisation de l'installation de Grub :** J'ai configuré l'installation automatique du chargeur d'amorçage GRUB.
- **Utilisation du fichier preseed pendant l'installation:** pour cela je l'ai placé à la racine de la clé USB bootable

4.5. Problèmes rencontrés:

Après avoir lancé le processus d'installation en utilisant le fichier `preseed`, j'ai rencontré plusieurs erreurs de configuration, qui sont apparues au fil de plusieurs tests. Chaque tentative d'installation a mis en lumière une nouvelle erreur, nécessitant des ajustements pour résoudre les problèmes rencontrés.

1) Erreur d'installation de GRUB:

L'installation automatique du chargeur d'amorçage GRUB a échoué, ce qui a empêché le démarrage du système après l'installation.

Pour résoudre cette erreur, après un processus de débogage, j'ai identifié que le problème survenait lors de l'installation de l'ISO. Il s'est avéré que l'installateur ne parvenait pas à reconnaître que la clé USB était bootable, ce qui affichait un message d'erreur `"aucun système trouvé"` lors du démarrage.

Pour remédier à cela, j'ai opté pour une autre méthode de gravure de l'ISO sur la clé USB en utilisant la commande `dd`, comme suit :



```
(kali㉿kali)-[~/Downloads]
$ sudo dd if=output.iso of=/dev/sdb bs=4M status=progress
```

Figure 9: Exemple d'utilisation de "dd"

Cette commande permet de copier l'image ISO de Kali Linux sur la clé USB (/dev/sdX étant le périphérique de la clé USB) en utilisant un bloc de 4 Mo, garantissant ainsi que la clé soit correctement bootable et l'option `status=progress` me permet de suivre l'état d'avancement de la copie.

2) Non-détection du fichier preseed:

L'installateur n'a pas détecté le fichier `preseed` spécifié à la racine de la clé USB bootable, ce qui a entraîné une installation manuelle standard au lieu d'une installation automatisée.

Donc j'ai opté pour la modification directe de l'image ISO de Kali Linux. Cette démarche consistait à intégrer le fichier `preseed` à la racine de l'image ISO et à ajuster le chemin du fichier `preseed` dans le fichier de configuration du GRUB (`grub.cfg`) ainsi que tous les autres fichiers nécessaires à l'installation.

Il était également nécessaire de mettre à jour les valeurs `md5sum` pour tous les fichiers modifiés afin de garantir l'intégrité des données et que la lecture de ces fichiers au niveau de l'installation se fasse correctement.

Voici le processus détaillé :

1. Installation de mkisofs :

Tout d'abord, j'ai installé l'outil mkisofs qui est nécessaire pour créer une nouvelle image ISO modifiée, vous pouvez l'installer avec la commande suivante:

```
(kali㉿kali)-[~]  
$ sudo apt-get install mkisofs  
[sudo] password for kali:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Note, selecting 'genisoimage' instead of 'mkisofs'  
genisoimage is already the newest version (9:1.1.11-3.5).  
The following packages were automatically installed and are no longer required:  
cgroupfs-mount containerd libintl-perl libintl-xs-perl libmodule-find-perl libmodule-scandeps-perl
```

Figure 10: Installation de mkisofs

2. Edition de l'image ISO :

Création d'un répertoire temporaire pour travailler avec les fichiers de l'image ISO.

```
(kali㉿kali)-[~/Downloads]  
$ mkdir /tmp/custom_iso
```

Figure 11: création de répertoire temporaire

3. Montage de l'image ISO d'origine :

```
(kali㉿kali)-[~]  
$ sudo mount -t iso9660 -o loop /home/kali/Downloads/kali-linux-2024.1-installer-amd64.iso /mnt/  
mount: /mnt: WARNING: source write-protected, mounted read-only.
```

Figure 12: Montage de l'ISO

4. Copie des fichiers de l'image montée vers le répertoire temporaire :

```
(kali㉿kali)-[/mnt]  
$ tar cf - . | (cd /tmp/custom_iso; tar xfp -)
```

Figure 13: copie de fichiers vers le répertoire temporaire

Cette étape est nécessaire car le montage est en lecture seule, donc nous devons copier les fichiers pour pouvoir les modifier.

5. Modification des fichiers :

Maintenant que les fichiers de l'image ISO sont copiés dans le répertoire temporaire (/tmp/custom_iso), j'ai ajouté le fichier **preseed** (préparé au préalable) à la racine de ma clé USB, ensuite, j'ai procédé à des modifications sur des fichiers critiques pour l'installation, notamment le fichier **grub.cfg**, (que vous allez trouver dans le dossier /boot/) ce fichier est indispensable pour le démarrage du système car il détermine les options de démarrage et les paramètres du noyau Linux.

Mes modifications visaient à inclure une référence au fichier **preseed**, garantissant ainsi que le processus d'installation démarre avec les configurations appropriées. Vous pouvez voir un exemple de ces modifications dans l'image ci-dessous :

```
linux /install.amd/vmlinuz net.ifnames=0 preseed/file=/cdrom/preseed.cfg preseed/file/
checksum=97968f21c24146c4eb717e62ceabf5cb simple-cdd/profiles=kali,offline desktop=xfce auto=true priority=critical
vga=788 — quiet
initrd /install.amd/gtk/initrd.gz
}
}
```

Figure 14: Modification du "grub.cfg"

J'ai précisé le chemin correct vers le fichier **preseed** sur la clé USB et j'ai également mis à jour son checksum pour préserver son intégrité et aussi pour que l'installateur puisse le reconnaître au niveau de l'installation:

```
(root@kali)-[/tmp/custom_iso/boot/grub]
# md5sum grub.cfg
608ad97b819da66b1499aa689ccaa15c grub.cfg
```

Figure 15: Checksum du grub.cfg

```
14 352b97579c5739aac18951bdaf0089fb ./boot/grub/efi.img
15 fc07119a612a66809d996a5104fc008e ./boot/grub/font.pf2
16 608ad97b819da66b1499aa689ccaa15c ./boot/grub/grub.cfg
17 815c0ae016fccc06294a49f3e41935fa ./boot/grub/i386-efi/grub.cfg
```

Figure 16: Mise à jour dans le "md5sum.txt"

Cette procédure a été réalisée pour tous les fichiers nécessaires à l'installation, notamment **txt.cfg**, **adgtk.cfg**

```
1 label install
2     menu label ^Install
3     kernel /install.amd/vmlinuz
4     append net.ifnames=0 preseed/file=/cdrom/preseed.cfg preseed/file/checksum=97968f21c24146c4eb717e62ceabf5cb|
simple-cdd/profiles=kali,offline desktop=xfce vga=788 initrd=/install.amd/initrd.gz — quiet
5
```

Figure 17: modification du chemin du fichier preseed

Et éventuellement `menu.cfg` (ce fichier n'est responsable que de la partie graphique de l'installateur, je l'ai modifié juste pour ne laisser que les options qui m'intéressent) :


```
1 menu hshift 4
2 menu width 70
3
4 menu title  Kali iso installer menu (BIOS mode)
5 include stdmenu.cfg
6 menu begin advanced
7     menu label ^automated options
8         menu title automated options
9             include stdmenu.cfg
10         label mainmenu
11             menu label ^Back..
12             menu exit
13         include adgtk.cfg
```

Figure 18: `menu.cfg`

J'ai également veillé à mettre à jour le fichier `md5sum.txt` pour refléter les modifications apportées. (c'est un fichier qui contient les valeurs de hachage MD5 associées à chaque fichier présent dans l'image ISO de Kali Linux. Ces valeurs permettent de vérifier l'intégrité des fichiers téléchargés en comparant les hachages calculés localement avec ceux répertoriés dans le fichier `md5sum.txt`. Cela garantit que les fichiers n'ont pas été altérés ou corrompus lors du téléchargement).

6. Création d'une nouvelle image ISO :

Une fois les modifications terminées, je suis retourné dans le répertoire `/mnt` et j'ai exécuté la commande suivante pour créer une nouvelle image ISO :

```
(kali㉿kali)-[/mnt]
$ sudo mkisofs -o output.iso -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -allow-limited-size -J -R -V "Kali Installer" .
```

Figure 19: Création de l'ISO

7. Rendre l'image ISO bootable avec isohybrid :

Pour rendre l'image ISO bootable, j'ai installé l'utilitaire `isohybrid` (permettant ainsi une utilisation flexible tant pour la création de supports bootables sur CD/DVD que sur clé USB) et j'ai exécuté la commande suivante:

```
(kali㉿kali)-[/mnt]
$ sudo isohybrid output.iso
```

Figure 21: Rendre l'image ISO isohybrid

```
(kali㉿kali)-[/mnt]
$ sudo dd if=output.iso of=/dev/sdb bs=1M status=progress
```

Figure 20: Rendre l'ISO bootable

3) Problème de partitionnement:

La configuration du partitionnement que j'ai précisé n'a pas fonctionné comme prévu, ce qui a entraîné une disposition incorrecte des partitions sur le disque dur.

Pour résoudre ce problème, après avoir fait des recherches dans la documentation Debian sur le sujet j'ai trouvé qu'utiliser la méthode "keep" permet d'indiquer que la partition doit être conservée telle quelle (sans formatage). Cette approche implique de spécifier explicitement dans le fichier **preseed** que je vais garder cette partition telle qu'elle afin de l'utiliser pour stocker les données des clients plus tard.

Par exemple, voici un extrait de configuration utilisant la méthode "keep" pour conserver une partition existante et la laisser libre pour une utilisation ultérieure:

```
# Spécifier la recette de partitionnement expert
d-i partman-auto/expert_recipe string
    my-recipe ::
        10 10 10 ext3
            $primary{ }
            method{ keep }
        .
        100 10000 10000000000 ext3
            method{ format } format{ }
            use_filesystem{ } filesystem{ ext3 }
            mountpoint{ / }
        .
        64 512 300% linux-swaps
            method{ swap } format{ }
        .

# Spécifier la recette choisie
d-i partman-auto/choose_recipe select my-recipe
```

Figure 22 : recette de partitionnement

4) Paquets de base manquants :

Certains paquets de base nécessaires pour un environnement Kali Linux standard n'ont pas été inclus dans le fichier preseed, ce qui a abouti à une configuration incomplète du système après l'installation.

Donc, j'ai simplement ajouté les packages appropriés dans la partie d'installation des paquets du fichier **preseed**. Le package nécessaire était "**kali-linux-default**", qui contient les packages de base essentielle pour un environnement Kali Linux standard.

De plus, j'ai également inclus le package "**xfce-desktop**" qui fournit l'environnement de bureau XFCE, offrant ainsi une interface graphique conviviale pour les utilisateurs.

```
75 ### Package selection
76 taskel taskel/first multiselect standard ssh-server desktop
77 d-i pkgsel/include string build-essential sudo kali-desktop-xfce kali-linux-
  default
78
```

Figure 23: Sélection des paquets

Etant donné que la machine une fois installée sera située chez le client, il est crucial de réfléchir aux méthodes de sécurisation, tant logiques que physiques, ainsi qu'au durcissement de la machine. Ces mesures sont nécessaires pour protéger la machine contre tout accès non autorisé ou toute manipulation malveillante, qui pourraient compromettre la sécurité des données et des opérations de tests de pénétration.

Toute personne externe, peut représenter un risque de sécurité involontaire ou intentionnel. Pour cette raison, j'ai privilégié la sécurisation des ports, des partitions et le BIOS plutôt que d'envisager d'autres solutions plus complexes et coûteuses tels que l'utilisation de matériel spécialisé avec des fonctionnalités de sécurité avancées. Ces mesures permettent de renforcer la protection de la machine de manière efficace tout en restant simple à déployer et à gérer.

Ce qui nous amène à l'étape suivante : le durcissement de la machine.

4.6. Mesures de sécurité mises en place

Pour commencer, j'ai réfléchi à la manière de sécuriser les ports USB. Initialement, je me suis penché vers une solution complexe impliquant la gestion des autorisations directement à partir de `udev`.

`udev` est le gestionnaire de périphériques de Linux, qui gère les nœuds de périphériques dans le système de fichiers. Avec `udev`, il est possible de créer des règles spécifiques pour autoriser ou interdire certains périphériques USB en fonction de divers critères comme les IDs de fabricant et de produit.

Cependant, cette approche est complexe et sujette à des erreurs, ce qui pourrait compromettre la disponibilité de la box si une configuration incorrecte est déployée chez le client.

Après une mise au point avec mon encadrant, il m'a conseillé de chercher une solution plus simple. L'objectif étant de minimiser la configuration nécessaire chez le client, tout en assurant que la box reste fonctionnelle et disponible.

En suivant ce conseil, j'ai trouvé une solution alternative avec `usbguard`.

`usbguard` est un logiciel conçu pour contrôler l'accès aux périphériques USB au niveau du système d'exploitation. Il fournit une interface simple pour autoriser ou bloquer les périphériques USB basés sur des règles définies par l'utilisateur.

Pour simplifier, j'ai installé `usbguard` et je l'ai activé ce qui bloque tous les périphériques USB à l'exception de ceux déjà connectés.

Pour l'ajout ou le retrait d'un périphérique en cas de besoin, j'ajoute une règle spécifique en utilisant l'ID du fabricant et l'ID du produit, ce qui permet de contrôler précisément les accès USB et d'assurer une protection contre les périphériques non autorisés.

Après avoir sécurisé les ports physiques, j'ai entrepris de sécuriser les ports logiques.

Voici les étapes que j'ai suivies :

Première Réflexion : utiliser **iptables**

iptables est un utilitaire de ligne de commande permettant de configurer les règles de filtrage de paquets dans le noyau Linux.

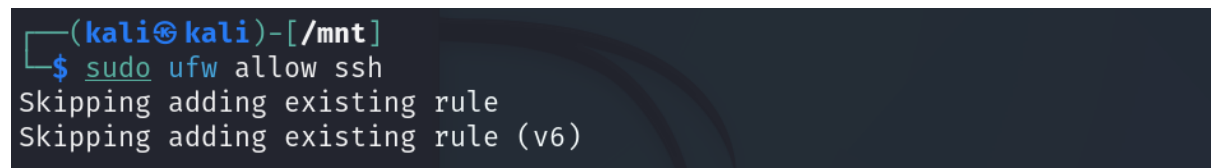
Cependant, comme avec **udev**, il est facile de commettre des erreurs de configuration avec **iptables**, ce qui peut compromettre la sécurité et/ou la disponibilité de la box.

Solution alternative : utiliser **ufw**

ufw (Uncomplicated Firewall) est un outil destiné à simplifier la gestion des pare-feux sur les systèmes basés sur Debian. Il fournit une interface plus conviviale pour gérer les règles de pare-feu par rapport à **iptables** il est conçu pour être simple à utiliser tout en offrant une protection robuste.

Par défaut, il bloque tout le trafic entrant et permet tout le trafic sortant, offrant ainsi une bonne base de sécurité.

En sachant cela, j'ai installé et activé **ufw**, j'ai ensuite ajouté une règle spécifique pour autoriser les connexions entrantes **ssh**, comme mentionné précédemment, il est nécessaire de pouvoir se connecter en **ssh** pour prendre le contrôle à distance de la box.



```
(kali㉿kali)-[/mnt]
└─$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
```

Figure 24: règles autorisant le flux ssh

Après avoir sécurisé les ports physiques et logiques de la box, j'ai entrepris de sécuriser le BIOS et le système de démarrage pour prévenir tout accès non autorisé.

L'objectif principal est de bloquer toute tentative de démarrage à partir de supports autres que le disque dur principal, empêchant ainsi un attaquant potentiel de lancer le système avec son propre dispositif de démarrage, ce qui pourrait compromettre la sécurité globale du système.

Exemple:

Un tel scénario pourrait se produire si un attaquant démarrait le système à partir de son propre support, puis arrivait d'une manière à se connecter au VPN. À distance, nous pourrions croire que nous sommes connectés à notre propre machine, alors qu'en réalité, c'est l'attaquant qui en a le contrôle et si nous transmettons des informations confidentielles en pensant être en sécurité, l'attaquant pourrait les intercepter.

Afin de réaliser cela j'ai accédé au menu de démarrage du BIOS, dans l'onglet "Security" puis cherché l'option "Set Supervisor Password" pour configurer le mot de passe administrateur.

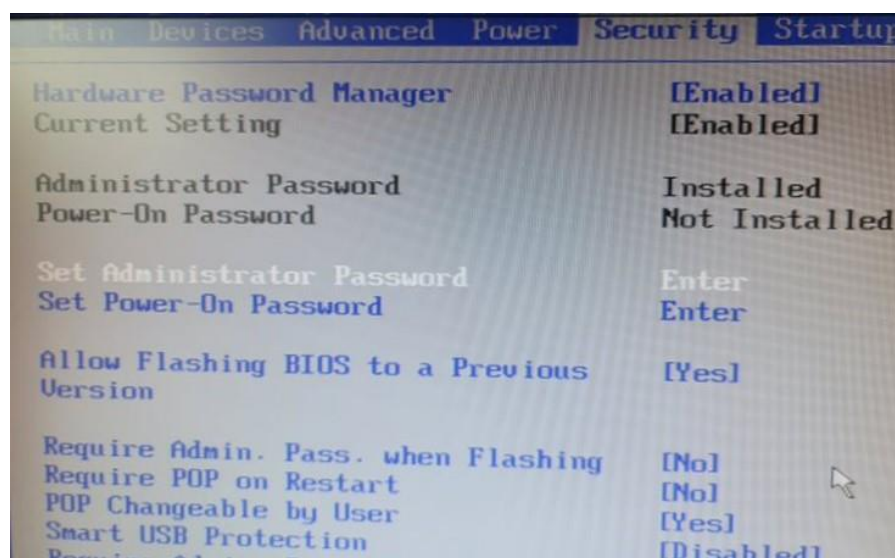


Figure 25: mot de passe configuré

Cette étape empêche toute modification non autorisée des paramètres du BIOS car à partir du moment où le mot de passe est configuré chaque accès au BIOS nécessite une authentification.

Et j'ai ensuite configuré les options de démarrage de sorte que j'ai désactivé toutes les options de démarrage autres que le disque dur principal. Cela inclut les options de démarrage à partir d'une clé USB, d'un CD/DVD, ou d'un réseau.

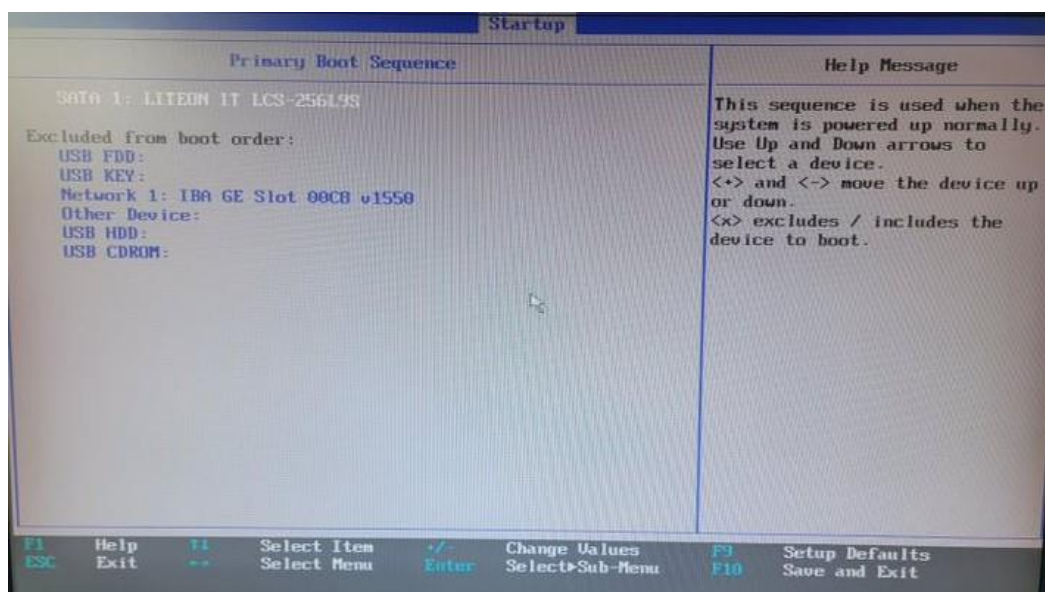


Figure 26: garder que le disque comme option de démarrage

4.7. Automatisation de post installation avec Ansible

Afin d'automatiser les tâches de post-installation sur la box, j'ai utilisé [Ansible](#), qui est un outil de gestion de configuration et d'automatisation permettant de déployer des applications et configurations de manière simple et efficace.

Pour utiliser [Ansible](#), j'ai préparé un fichier YAML (qui est un format de fichier de sérialisation de données lisible par l'homme, souvent utilisé pour la configuration, qui utilise une indentation par espaces pour indiquer les niveaux de structure), contenant les instructions à exécuter ainsi qu'un fichier d'inventaire qui répertorie les machines cibles avec leurs adresses IP et les informations d'accès (pour la connexion en ssh). Il permet de regrouper et gérer ces hôtes de manière centralisée pour les configurations automatisées.

En exécutant Ansible, une connexion en SSH est établie entre le serveur Ansible (la machine qui l'exécute) et les machines cibles (hôtes). Une fois connecté, Ansible déploie les configurations en réalisant les tâches définies dans les fichiers YAML sur les machines cibles, en envoyant des modules.

Ces modules sont traités séquentiellement et peuvent comporter des actions telles que l'installation de logiciels, la modification de fichiers de configuration, etc...

Voici comment j'ai utilisé Ansible pour automatiser les tâches de post-installation sur la box

Installation des Logiciels :

J'ai installé une sélection de logiciels essentiels pour les opérations de pentest ainsi que certains paquets indispensables.

Voici un extrait du fichier YAML qui montre comment j'ai rédigé les instructions pour installer ces logiciels avec Ansible :

```
23 - name: installing softwares
24   ansible.builtin.apt:
25     name:
26       - gufw
27       - joplin
28       - usbguard
29       - keepassxc
30       - docker.io
31     state: latest
```

Figure 27: exemple de tâches ansible

Activation des services au démarrage :

J'ai configuré Ansible pour activer et démarrer automatiquement des services comme **ssh** et **docker**.

Cela garantit que les services sont disponibles dès que l'utilisateur se connecte à la machine.

```
59
60 - name: Activer et démarrer le service SSH
61   ansible.builtin.systemd:
62     name: ssh
63     enabled: yes
64     state: started
65
```

Figure 28:activation de ssh avec ansible

Activation d'UFW :

Après l'installation, j'active UFW pour qu'il soit opérationnel, voici un exemple de tâche Ansible pour activer le service :

```
70
71 - name: Activer le pare-feu
72   ansible.builtin.ufw:
73     state: enabled
```

Figure 29: Activation du pare-feu

Définition des règles de filtrage :

Exemple de tâches ansible, pour autoriser spécifiquement le trafic nécessaire comme le SSH

```
66 - name: Activer le pare-feu et autoriser le trafic SSH entrant
67   ansible.builtin.ufw:
68     rule: allow
69     name: SSH
70
```

Figure 30: Autorisé le trafic ssh entrant

Maintenant que le fichier YAML est bien édité, je suis passé au fichier inventaire.

Voici un exemple de ce à quoi ressemble mon fichier inventaire:

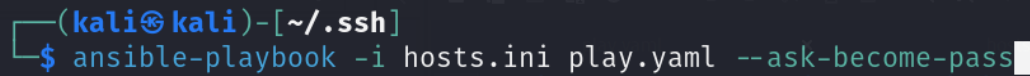
```
1 [myhosts]
2 kali ansible_host=192.168.254.2 ansible_user=torii ansible_ssh_private_key_file=/
  home/kali/.ssh/id_ed25519
3
```

Figure 31: Fichier inventaire

Explication :

Champ	Signification
myhosts	C'est une étiquette que nous utilisons pour grouper les machines à configurer (une seule dans notre cas) cette étiquette sera ensuite incluse dans l'en-tête du fichier YAML
kali	C'est le nom de la machine
ansible_host	spécifie l'adresse IP de la machine cible (celle fournie avec le VPN qu'on a configuré auparavant),
ansible_user	spécifie l'utilisateur à utiliser pour la connexion.
ansible_ssh_private_key_file	spécifie le chemin vers la clé privée SSH utilisée lors de la connexion aux hôtes cibles

Une fois ces fichiers préparés, la commande suivante permet d'exécuter le fichier YAML (ou playbook) sur les machines spécifiées :



```
(kali㉿kali)-[~/ssh]
$ ansible-playbook -i hosts.ini play.yaml --ask-become-pass
```

Figure 32: Commande d'exécution du fichier Yaml sur les hosts spécifiés

Afin d'éviter d'écrire le mot de passe en clair dans le fichier inventaire, j'ajoute l'option `--ask-become`, qui permet au programme de demander le mot de passe à l'exécution et de le taper directement dans le terminal. Cette option est utilisée pour les commandes nécessitant des privilèges élevés pour s'exécuter.

4.8. Tests et validation

Après avoir fait l'automatisation des tâches, il est essentiel de passer par une phase de tests pour confirmer que tout fonctionne correctement.

Pour ce faire, j'ai rédigé un cahier de test détaillant toutes les tâches réalisées jusqu'à présent et incluant divers tests pour des vérifications supplémentaires.

J'ai ensuite exécuté ces tests sur une machine vierge pour assurer la fiabilité et la cohérence de l'ensemble du processus. Le cahier de test a été réalisé sur Excel pour une meilleure organisation des tests.

1	Section	Valorisation	Description	Résultat attendu
2	Installation et Configuration Initiale	Test de l'installation automatisée avec Preseed	Vérifier que l'installation de Kali Linux se fait correctement via un fichier preseed	La machine s'installe sans intervention manuelle et démarre avec les paramètres souhaités.
3				
4				
5		SSH	Vérifier que le service SSH est en cours d'exécution sur la machine de pentest (même après redémarrage) Teste la connexion SSH à partir d'un ordinateur distant en utilisant les identifiants (situé en bas)	La connexion SSH est établie avec succès, permettant un accès distant sécurisé à la machine de pentest.
6				
7				
8		VPN		
9				
10				

Figure 33: Cahier de test

Résultats des Tests:

Les résultats des tests m'ont montré que :

- L'installation de Kali Linux est entièrement automatisée et se déroule sans intervention manuelle.
- La connexion VPN est automatiquement établie et sécurisée après chaque démarrage.
- Les partitions sont correctement chiffrées et nécessitent une authentification pour l'accès.
- Toutes les tâches de post-installation sont correctement automatisées via Ansible, garantissant une configuration uniforme.
- Les mesures de sécurité, telles que les règles USBGuard et UFW, fonctionnent comme prévu.

Cependant, certains ajustements de configuration ont été nécessaires à savoir :

Dépôt de mise à jour :

Problème rencontré :

Lors des tests, j'ai découvert que le dépôt utilisé pour les mises à jour n'était pas un dépôt en ligne mais un dépôt local. Une fois l'installation terminée, le système continuait d'utiliser le dépôt local pour les mises à jour, ce qui rend impossible à la machine de télécharger de nouveaux logiciels.

Solution apportée :

Modification du fichier `/etc/apt/source.list` pour inclure un dépôt en ligne officiel de Kali Linux.

```
(kali㉿kali)-[~]  
$ echo "deb http://http.kali.org/kali kali-rolling main contrib non-free" | sudo tee -a /etc/apt/sources.list
```

Figure 34: solution apportée pour le problème de dépôt

6. Suivis du projet :

À la fin de mon stage, le projet sera repris par les responsables techniques de l'entreprise, notamment mon encadrant. J'ai développé une première version fonctionnelle du système, mais il est important de noter que cette version peut et devrait être améliorée et adaptée en fonction des besoins évolutifs de l'entreprise et des retours des utilisateurs finaux.

Pour permettre une maintenance efficace, j'ai réalisé une documentation technique couvrant les aspects essentiels du projet.

Bien que la version actuelle soit opérationnelle, il existe plusieurs axes d'amélioration pour faire évoluer le projet à savoir :

Renforcement de la sécurité :

- Intégration de solutions supplémentaires pour la détection et la prévention des intrusions.
- Mise en place de politiques de sécurité avancées pour le pare-feu.

Amélioration de l'automatisation :

- Optimisation des scripts Ansible pour couvrir davantage de configurations et réduire les interventions manuelles.

Scalabilité et portabilité :

- Adapter les configurations pour permettre une mise à l'échelle facile à d'autres environnements et infrastructures.

7. Mission de pentest :

Durant mon stage, j'ai eu l'opportunité de participer à une mission de pentest pour un client qui est responsable d'un réseau de transport. Cette mission constituait un pentest en boîte noire (black box), ce qui signifie que nous n'avions aucune information préalable sur le système cible. Notre objectif principal était de tester la sécurité de l'ensemble du système d'information de l'entreprise, en simulant une attaque externe afin d'identifier les vulnérabilités potentielles.

Mon implication principale était concentrée sur l'application web de l'entreprise et j'ai suivis la méthodologie suivante :

Reconnaissance et collecte d'information :

- Recueil des informations sur l'application web et son infrastructure.
- Utilisation d'outils de scan (nmap) pour identifier les points d'entrée et les technologies utilisées.

Scanning et enumeration :

- Réalisation de scans de ports et d'analyses de services pour comprendre la surface d'attaque.
- Identification des pages et des fonctionnalités disponibles sur l'application web.

Exploitation :

- Tentatives d'exploitation des vulnérabilités trouvées pour obtenir un accès non autorisé.
- Utilisation de diverses techniques d'attaque et vulnérabilités spécifiques aux applications web.

Post-Exploitation :

- Analyse des impacts potentiels de l'exploitation réussie des vulnérabilités.

7.1. Enseignements tirés de la mission :

Cette mission de pentest m'a permis d'acquérir une expérience précieuse et de comprendre les différences fondamentales entre un pentest réel et un Capture The Flag (CTF) classique. Voici les leçons clés que j'ai apprises :

Complexité d'un Pentest Réel :

Contrairement à un CTF, où l'objectif final est souvent de récupérer un "drapeau" symbolique, un pentest réel implique une analyse approfondie et détaillée des vulnérabilités et des impacts potentiels sur l'entreprise cible.

Importance de l'Information :

Être un pentester de qualité nécessite une accumulation constante d'informations et de compétences. La connaissance des différentes technologies, des vulnérabilités courantes, et des techniques d'attaque est cruciale pour réussir un pentest.

Pragmatisme et Méthodologie :

Suivre une méthodologie structurée est essentiel pour assurer un pentest efficace et exhaustif. De la reconnaissance à la post-exploitation, chaque étape doit être menée avec rigueur et attention aux détails.

7.2. Perspectives personnelles :

Participer à cette mission a été une expérience très enrichissante. En tant que première mission de pentest dans un environnement réel, elle m'a offert une vision claire des défis et des responsabilités d'un pentester. J'ai particulièrement apprécié :

Application Pratique :

Mettre en pratique les compétences théoriques dans un contexte réel m'a permis de mieux comprendre les nuances et les subtilités du pentesting.

Équipe et Encadrement :

Travailler aux côtés d'un professionnels expérimenté m'a permis d'apprendre des meilleures pratiques et de recevoir des conseils précieux.

Vision Clairvoyante :

Cette expérience m'a aidé à définir les domaines sur lesquels je dois me concentrer pour améliorer mes compétences en sécurité informatique en général

8. Bilan:

Ce stage a représenté une expérience enrichissante à plusieurs égards. Il m'a offert l'opportunité de mettre en pratique mes connaissances théoriques dans un domaine technique complexe et varié. Travailler sur un projet d'une telle envergure m'a exposé à des situations concrètes et m'a permis d'approfondir ma compréhension de divers aspects technologiques et organisationnels. En travaillant au sein d'un cabinet d'expertise, par opposition à une grande entreprise multinationale, j'ai pu bénéficier d'une atmosphère plus dynamique. Cela m'a permis de prendre davantage de responsabilités et d'acquérir une expérience précieuse dans la gestion de projets de bout en bout.

8.1. Compétences techniques :

J'ai acquis une expertise en matière de mise en place et d'automatisation de VPN, ce qui inclut la configuration et la gestion des connexions sécurisées. Mon expérience m'a également permis de perfectionner mes compétences en débogage, en identifiant et en résolvant les problèmes techniques rencontrés tout au long du processus d'installation et de configuration.

De plus, j'ai des compétences avancées dans l'automatisation des tâches d'installation et de configuration à l'aide de l'outil [Ansible](#). Parallèlement à ces compétences techniques, j'ai également amélioré ma capacité à rédiger des rapports techniques clairs et structurés.

8.2. Compétences organisationnelles :

Cette expérience a renforcé mon autonomie, notamment grâce à la responsabilité que j'ai eu en étant seul sur le projet. Cela m'a poussé à développer mes compétences en gestion du temps, en planification des tâches et en respect des délais que je me suis fixés.

De plus, elle m'a offert l'opportunité d'approfondir mes compétences en gestion de projet et en prise de décisions autonomes. J'ai particulièrement apprécié la méthode de travail de mon tuteur, qui m'a encouragé à rechercher d'abord de manière autonome toutes les informations nécessaires pour acquérir une compréhension générale du sujet. Ensuite, il m'a guidé et donné des indices pour atteindre les objectifs spécifiques du projet.

En résumé, ce stage a été une expérience enrichissante à bien des égards, m'offrant l'opportunité de développer mes compétences personnelles et professionnelles tout en acquérant une vision plus claire du domaine de la sécurité informatique.

Les interactions quotidiennes avec mon encadrant ont également été précieuses, m'offrant des informations pertinentes et des conseils qui me seront certainement utiles dans ma future carrière. Durant ces deux mois, j'ai profondément compris le fonctionnement d'une entreprise, ainsi que les dynamiques de travail au sein de cette dernière.

9. Conclusion de stage:

Ce stage m'a donc non seulement permis d'appliquer mes connaissances théoriques dans un contexte professionnel, mais il m'a également donné un aperçu concret des défis et des opportunités qui m'attendent dans le domaine de la sécurité informatique.

Je tiens également à exprimer ma gratitude envers Monsieur Damien BOLUS. Son soutien constant, ses conseils avisés et son expertise ont été d'une valeur inestimable tout au long de mon stage, son encadrement attentif a grandement contribué à ma réussite et à l'élaboration de ce mémoire technique. Je souhaite également remercier chaleureusement chaque individu chez *Torii Security* pour leur précieuse aide et l'atmosphère familiale qui y règne.

Je suis également heureux d'annoncer que le projet que j'ai mis en place est désormais en production chez un client, ce qui constitue une grande satisfaction professionnelle. Pour le reste de mon stage, je vais commencer un nouveau projet qui consiste à réaliser des stations avec les technologies cloud sur du proxmox et à effectuer des migrations de CentOS 7 vers AlmaLinux, ce qui me permettra de continuer à bénéficier de l'encadrement attentif de M. Damien BOLUS et de poursuivre mon apprentissage au sein de *Torii Security*. Je suis reconnaissant de cette opportunité et j'espère pouvoir continuer à contribuer au succès de l'équipe tout en développant encore davantage mes compétences.

Je tiens à m'excuser par avance pour toute erreur de rédaction ou de contenu qui pourrait se trouver dans ce rapport. Je vous remercie pour votre compréhension et suis reconnaissant pour toute correction ou suggestion que vous pourrez apporter.