PROJET N°2 - Déploiement d'un Honeypot Cowrie et centralisation des logs avec Graylog

Technologies utilisées :

- Linux Debian
- Honeypot Cowrie
- Graylog (Docker)
- GELF
- Machine virtuelle attaquante Kali
- Virtualisation (POD/Proxmox)

SOMMAIRE

1. INTRODUCTION

1.1 Présentation personnelle et contexte

Je suis actuellement étudiante en BTS Services Informatiques aux Organisations, option SISR, en alternance au sein du service informatique de l'entreprise Alliance Environnement.

Dans le cadre de l'épreuve E6 – Administration des systèmes et réseaux, nous devons concevoir et mettre en œuvre une solution technique permettant d'illustrer leurs compétences en administration système, réseau et cybersécurité.

Le projet présenté dans ce dossier consiste à déployer un honeypot permettant de détecter et analyser les tentatives d'intrusion sur un serveur exposé.

L'objectif est de comprendre les techniques utilisées par les attaquants et de mettre en place un système permettant de collecter et analyser ces attaques en temps réel.

1.2 Contexte professionnel et problématique

Avec la transformation numérique des entreprises, les systèmes d'information sont devenus essentiels au fonctionnement des organisations. De nombreux services (serveurs, applications web, bases de données ou infrastructures réseau) sont aujourd'hui accessibles via Internet ou via des réseaux interconnectés.

Cette connectivité permanente expose les systèmes informatiques à diverses menaces de cybersécurité, telles que les scans de ports automatisés, les attaques par force brute visant les mots de passe, ou encore les tentatives d'exploitation de vulnérabilités.

Pour les administrateurs systèmes et réseaux, il est donc important de comprendre le fonctionnement de ces attaques afin de mieux protéger les infrastructures. Toutefois, analyser ces attaques directement sur un serveur réel peut représenter un risque.

C'est pourquoi il existe des outils appelés honeypots. Un honeypot est un système volontairement exposé aux attaquants afin d'observer leur comportement. Il simule un service réel et enregistre les actions réalisées lors des tentatives d'intrusion.

Dans ce projet, l'objectif est de mettre en place un honeypot Cowrie, simulant un serveur SSH. Les tentatives de connexion seront enregistrées puis envoyées vers la plateforme Graylog afin de centraliser et analyser les logs.

1.3 Objectifs du projet

Les objectifs de ce projet sont les suivants :

<u>Objectifs pédagogiques</u>

- Comprendre le fonctionnement d'un honeypot
- Mettre en place une architecture de cybersécurité
- Centraliser et analyser des logs de sécurité
- Documenter une infrastructure informatique

Objectifs techniques

- Déployer un honeypot Cowrie
- Simuler des attaques via une machine attaquante
- Centraliser les logs dans Graylog
- Analyser les tentatives d'intrusion
- Observer les comportements des attaquants

Résultats attendus :

Obtenir une infrastructure fonctionnelle comprenant un honeypot Cowrie permettant de simuler des tentatives d'attaque sur un serveur SSH, ainsi qu'une plateforme Graylog permettant de collecter, centraliser et analyser les logs générés afin d'observer les comportements des attaquants.

2. ANALYSE ET ETUDE DU BESOIN

2.1 Expression du besoin

Problématique générale : Comment mettre en place une infrastructure permettant de détecter et analyser les tentatives d'intrusion sur un serveur afin de mieux comprendre les méthodes utilisées par les attaquants ?

Besoins identifiés :

a) Simuler un serveur vulnérable
b) Enregistrer les connexions malveillantes
c) Observer les commandes exécutées par les attaquants
d) Centraliser les logs dans un système d'analyse
e) Visualiser les tentatives d'intrusion

2.2 Cahier des charges et contraintes

Contraintes techniques :

CONTRAINTE	DESCRIPTION	IMPACT
Environnement virtualisé	Le projet est réalisé à l'aide de machines virtuelles	Limitation des ressources matérielles disponibles

Systèmes d'exploitation	Utilisation de systèmes Linux pour les serveurs	Nécessité de maîtriser l'administration Linux
Solutions open-source	Les outils utilisés doivent être gratuits et open-source	Choix de Cowrie, Graylog et GELF
Architecture réseau	Les différentes machines doivent communiquer entre elles	Configuration du réseau entre les machines virtuelles
Centralisation des logs	Les logs du honeypot doivent être envoyés vers Graylog	Mise en place d'un système de collecte de logs

Contraintes fonctionnelles et organisationnelles :

CONTRAINTE	**DESCRIPTION**	**IMPACT**
Simplicité d'utilisation	L'infrastructure doit rester simple à comprendre et à expliquer	Architecture claire pour la présentation du projet
Sécurité	Le honeytpot ne doit pas compromettre les autres services	Isolation dans un environnement virtualisé
Temps de réalisation	Rédaction d'un dossier technique détaillé	Choix d'outils simples à déployer
Analyse des données	Les logs collectés doivent être exploitables et lisibles	Utilisation de Graylog pour la visualisation
Documentation	Le projet doit être entièrement documenté	Rédaction d'un dossier technique détaillé

2.3 Etude des solutions existantes

Avant de mettre en place l'infrastructure du projet, il est nécessaire d'étudier les différentes solutions existantes permettant de simuler des attaques et de collecter les informations associées. Cette étude porte à la fois sur les **outils de honeypot**, permettant de simuler un système vulnérable, ainsi que sur les **solutions de gestion et d'analyse de logs**, permettant de centraliser et d'exploiter les données générées par ces attaques.

ETUDE DES HONEYPOTS

Solution 1 : Dionaea

Dionaea est un honeypot permettant de simuler plusieurs services réseau afin de capturer les tentatives d'exploitation de vulnérabilités.

Avantages

- Permet de simuler plusieurs types de services (FTP, HTTP, SMB, etc.)
- Conçu pour capturer des malwares utilisés lors des attaques
- Permet d'observer différentes techniques d'exploitation
- Projet reconnu dans le domaine de la cybersécurité

Inconvénients

- Installation et configuration relativement complexes
- Nécessite une bonne compréhension des services réseau
- Documentation parfois difficile à exploiter pour les débutants

Solution 2 : Honeyd

Honeyd est un honeypot permettant de simuler des systèmes d'exploitation et différents services réseau.

Avantages

- Permet de simuler plusieurs machines virtuelles sur une seule machine
- Très flexible dans la configuration des services simulés
- Possibilité d'imiter différents systèmes d'exploitation
- Permet de créer des environnements réseau simulés

Inconvénients

- Configuration complexe
- Projet plus ancien avec moins de mises à jour récentes
- Prise en main difficile
- Installation parfois longue à mettre en place

Solution 3 : Cowrie (solution retenue)

Cowrie est un honeypot spécialisé dans la simulation de services SSH et Telnet. Il permet d'enregistrer les tentatives de connexion ainsi que les commandes exécutées par les attaquants.

Avantages

- Installation et configuration relativement simples
- Spécialisé dans la simulation des services SSH et Telnet, souvent ciblés par les attaques
- Enregistre les tentatives de connexion et les identifiants utilisés
- Permet d'observer les commandes exécutées par les attaquants
- Génère des logs exploitables pouvant être envoyés vers des outils d'analyse comme Graylog

Inconvénients

- Simulation limitée à certains services (SSH et Telnet)
- Ne simule pas un système complet
- Peut être détecté par certains attaquants expérimentés
- Analyse dépendante d'un outil externe pour la visualisation des logs

ETUDE DES SOLUTIONS DE GESTION DES LOGS

Solution 1 : ELK Stack (Elasticsearch, Logstash, Kibana)

La stack ELK est une solution très répandue permettant de collecter, stocker et visualiser des logs.

Avantages

- Solution très complète pour l'analyse de logs
- Visualisation avancée des données
- Très utilisée dans les infrastructures professionnelles
- Grande flexibilité dans l'analyse des logs
- Large communauté et documentation importante

Inconvénients

- Installation relativement complexe
- Configuration parfois longue
- Consommation importante de ressources
- Nécessite une bonne maîtrise des différents composants

Solution 2 : **Splunk**

Splunk est une plateforme d'analyse de logs très utilisée dans les environnements professionnels.

Avantages

- Interface très complète
- Puissants outils d'analyse et de recherche
- Visualisations avancées
- Solution très répandue en entreprise

Inconvénients

- Solution principalement commerciale
- Version gratuite limitée
- Installation et configuration complexes
- Consommation importante de ressources

Solution 3 : Graylog (solution retenue)

Graylog est une plateforme open-source permettant de centraliser et d'analyser des logs provenant de différents systèmes.

Avantages

- Permet de centraliser les logs provenant de plusieurs machines ou services
- Offre une interface web claire pour consulter et analyser les logs
- Permet de rechercher et filtrer facilement les événements enregistrés
- Compatible avec plusieurs outils de collecte de logs comme Filebeat ou Syslog

Inconvénients

- Nécessite plusieurs services pour fonctionner (MongoDB, OpenSearch)
- Consommation de ressources plus importante
- Configuration initiale nécessaire
- Certaines fonctionnalités avancées disponibles uniquement dans la version entreprise

Tableaux comparatifs des honeypots :

CRITERES	**DIONAEA**	**HONEYD**	**COWRIE**
Types de services simulés	Plusieurs services (FTP, HTTP, SMB…)	Plusieurs services et systèmes simulés	SSH / Telnet
Facilité d'installation	Moyenne	Complexe	Facile
Analyse des attaques	Capture de malwares	Simulation réseau	Enregistrement des commandes attaquantes
Documentation	Correcte	Limitée	Bonne

Tableaux comparatifs des solutions des solutions de gestion de logs :

CRITERES	**ELK STACK**	**SPLUNK**	**GRAYLOG**
Centralisation des logs	Oui	Oui	Oui
Facilité d'installation	Moyenne à complexe	Complexe	Moyenne
Interface de visualisation	Complète	Avancée	Claire et intuitive
Coût	Open-source	Payant en parti	Open-source

Solutions retenues :

Les solutions Cowrie et Graylog ont été retenues pour la réalisation de ce projet. Ils représentent un bon compromis entre simplicité de mise en œuvre, efficacité pour la simulation d'attaques et pertinence pédagogique dans le cadre d'un projet de cybersécurité en BTS SIO. Ces outils permettent de simuler des tentatives d'intrusion tout en centralisant et analysant les logs générés.

3. Conception et choix des solutions

3.1 Architecture logique retenue

L'infrastructure est composée de plusieurs machines virtuelles :

1. Honeypot : Simulation d'un serveur vulnérable
2. Graylog : Centralisation et analyse des logs
3. Attaquant : Simulation d'attaques
4. Client : Accès à l'interface Graylog

3.2 Architecture physique

L'infrastructure est composée de plusieurs machine virtuelles sur un environnement virtualisé hébergé sous Proxmox.

VM 1 : Machine virtuelle attaquante

- **Système d'exploitation** : Kali Linux
- **Rôle** : simulation des tentatives d'attaque SSH vers le honeypot
- **Fonction** : cette machine permet de générer des connexions malveillantes afin de tester le fonctionnement du honeypot et de vérifier la remontée des logs vers Graylog

VM 2 : Machine virtuelle honeypot

- **Système d'exploitation** : Debian
- **Service principal** : Cowrie
- **Adresse IP** : 10.2.132.133
- **Port exposé** : 2222
- **Rôle** : simulation d'un serveur SSH vulnérable afin d'attirer les tentatives d'intrusion
- **Fonction** : cette machine enregistre les tentatives de connexion ainsi que les commandes exécutées par les attaquants, puis transmet les logs générés vers Graylog à l'aide de GELF

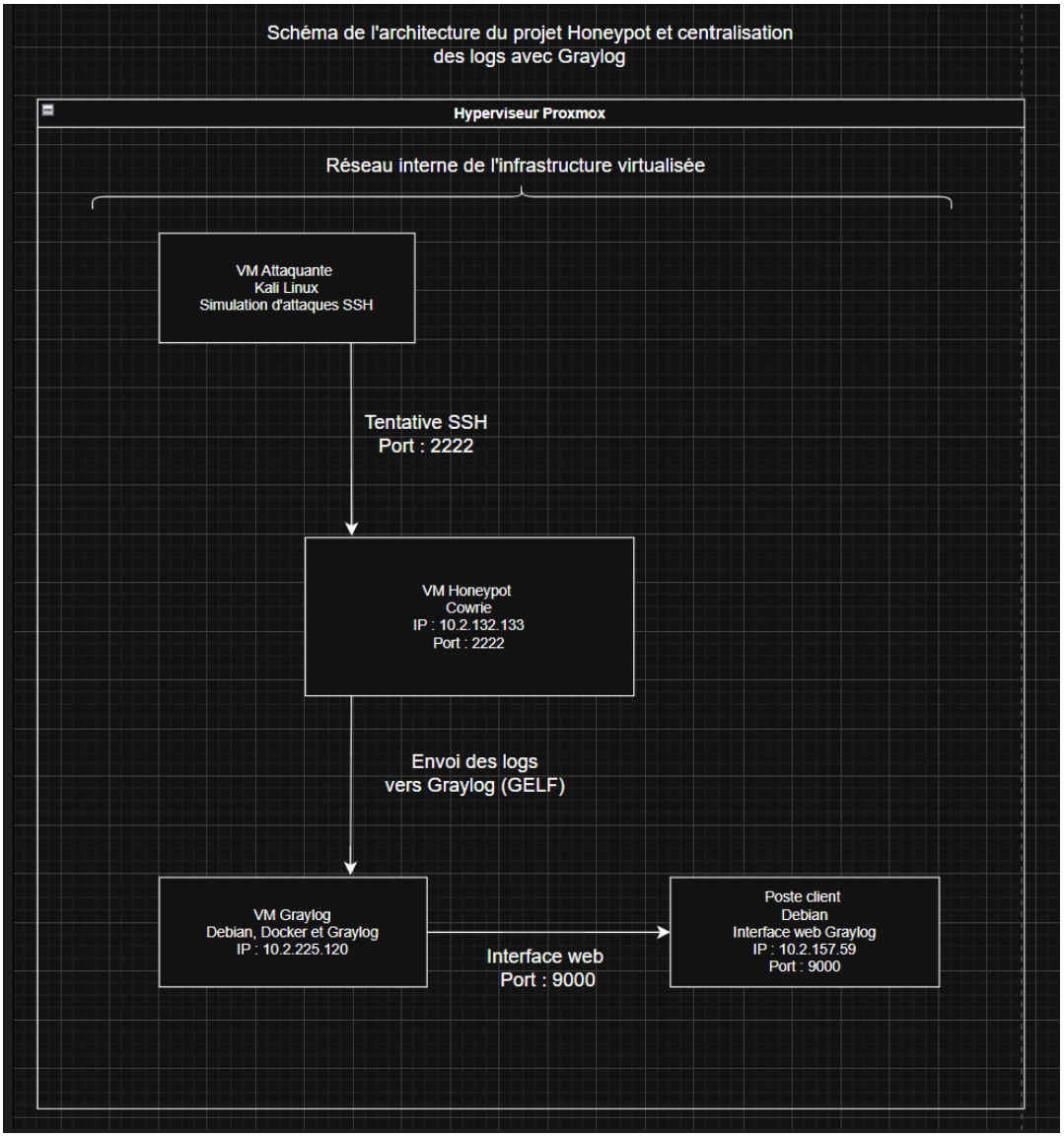
VM 3 : Machine virtuelle Graylog

- **Système d’exploitation** : Debian
- **Services installés** : Docker, Filebeat et Graylog
- **Adresse IP** : 10.2.225.120
- **Rôle** : centralisation, stockage et analyse des logs issus du honeypot
- **Fonction** : cette machine reçoit les journaux envoyés par Filebeat et permet leur consultation via une interface web

VM 4 : Poste client

- **Système d’exploitation** : Debian
- **Adresse IP** : 10.2.157.59
- **Port utilisé** : 9000
- **Rôle** : consultation de l’interface web de Graylog
- **Fonction** : cette machine permet d’accéder aux logs centralisés afin d’analyser les tentatives d’intrusion enregistrées par le honeypot

3.3 Schéma de l'architecture du projet



4. Mise en œuvre technique

4.1 Installation et configuration du honeypot Cowrie

Etape 1 : Installation du honeypot Cowrie

Dans un premier temps, il est nécessaire de récupérer le projet Cowrie depuis son dépôt officiel GitHub. Pour cela, la commande suivante est utilisée :

```
root@honeypot:~# cd ~
root@honeypot:~# git clone https://github.com/cowrie/cowrie.git_
```

Cette commande permet de télécharger le code source du honeypot Cowrie depuis le dépôt officiel et de le copier dans un dossier nommé cowrie sur la machine.

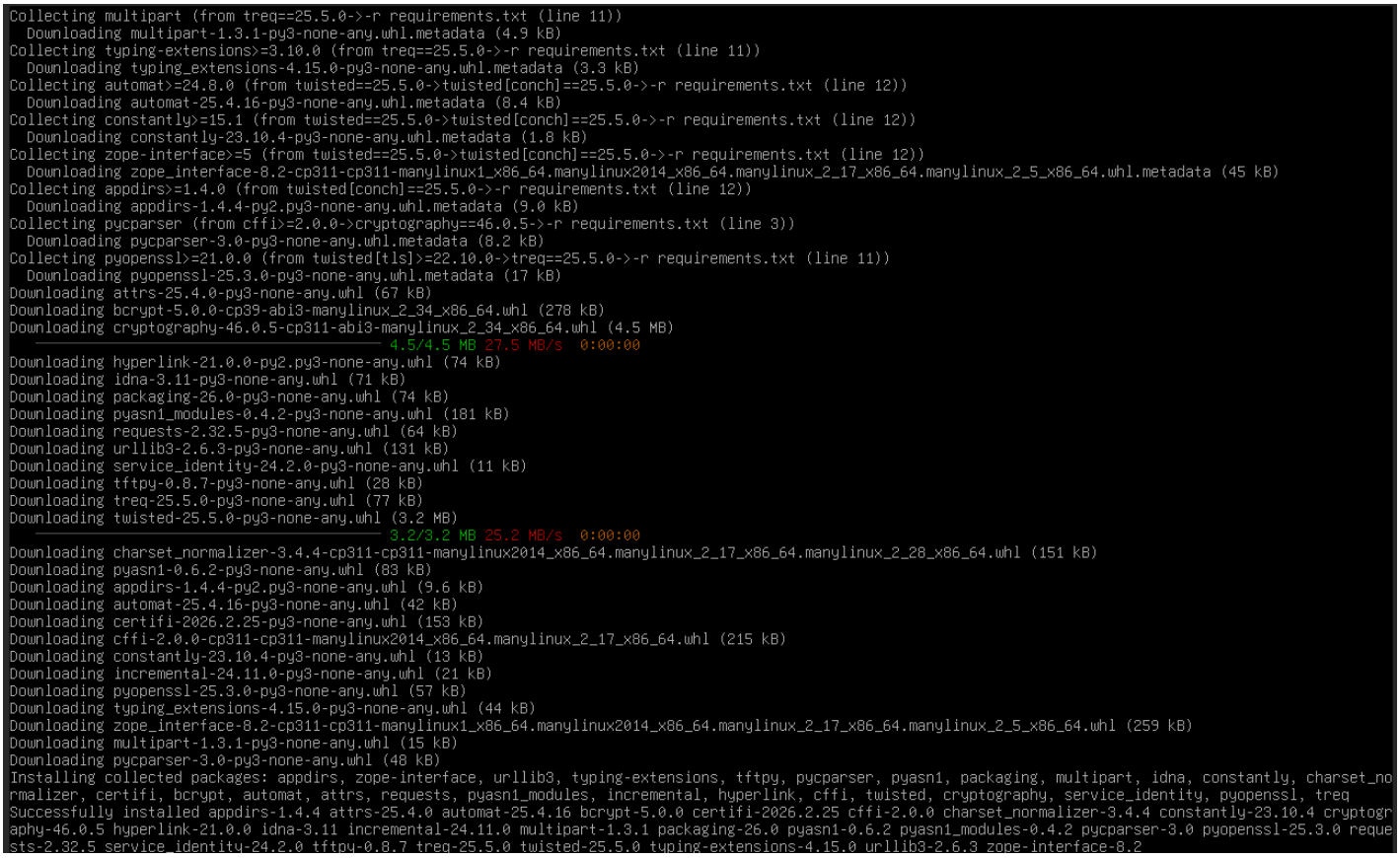
<u>Etape 2 :</u> Installation des dépendances :

Une fois le projet téléchargé, il est nécessaire d'installer les différentes dépendances nécessaires au fonctionnement du honeypot.

Lors du lancement de l'installation, plusieurs bibliothèques Python sont automatiquement téléchargées et installées, comme par exemple :

- twisted
- cryptography
- requests
- pyasn1
- service_identity

Ces bibliothèques sont nécessaires pour permettre au honeypot de fonctionner correctement et de simuler un service SSH.



<u>Etape 3 :</u> Démarrage du honeypot et vérification du fonctionnement du service

Une fois l'installation terminée, le honeypot peut être démarré afin de simuler un serveur SSH et commencer à enregistrer les tentatives de connexion.

Pour lancer le service Cowrie, la commande suivante est utilisée :

cowrie start

Cette commande permet de démarrer le honeypot. Lors de l'exécution, différents messages apparaissent dans le terminal indiquant que le service est en cours de lancement.
Afin de vérifier que le service fonctionne correctement, la commande suivante peut être utilisée :

cowrie status

Cette commande permet de vérifier l'état du service

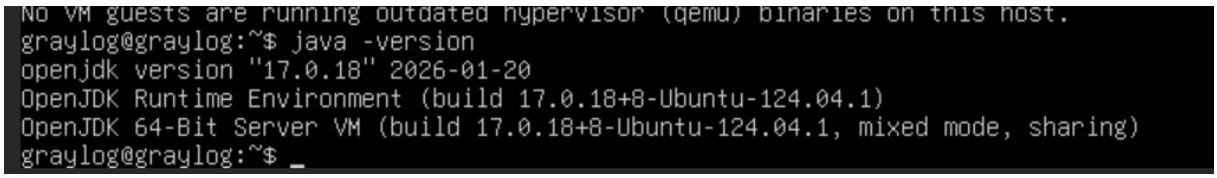
4.2 Mise en place de la plateforme Graylog

Etape 1 : Vérification de l'environnement Java

Avant l'installation de Graylog, il est nécessaire de vérifier que Java est installé sur la machine, car Graylog fonctionne sur la plateforme Java.
La commande suivante permet de vérifier la version installée :

java -version

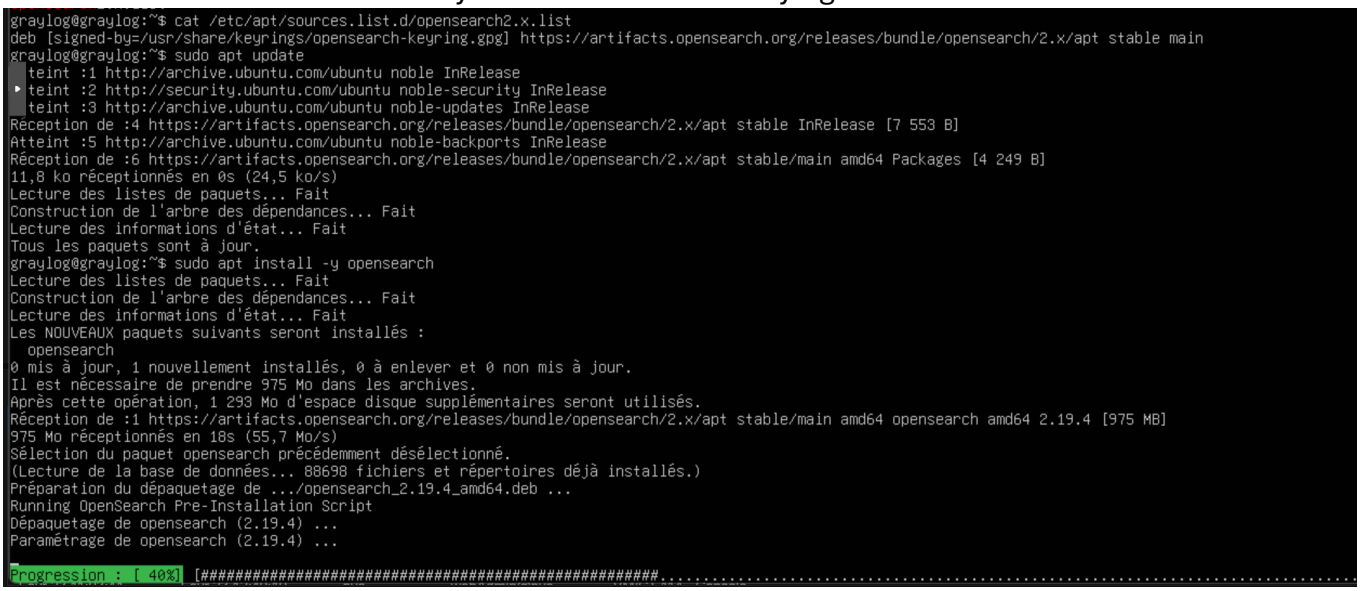
Cette vérification permet de s'assurer que l'environnement est compatible avec le fonctionnement de Graylog.



Etape 2 : Installation du moteur de stockage Opensearch

Graylog utilise un moteur de recherche afin de stocker et indexer les logs. Dans ce projet, l'outil OpenSearch est utilisé

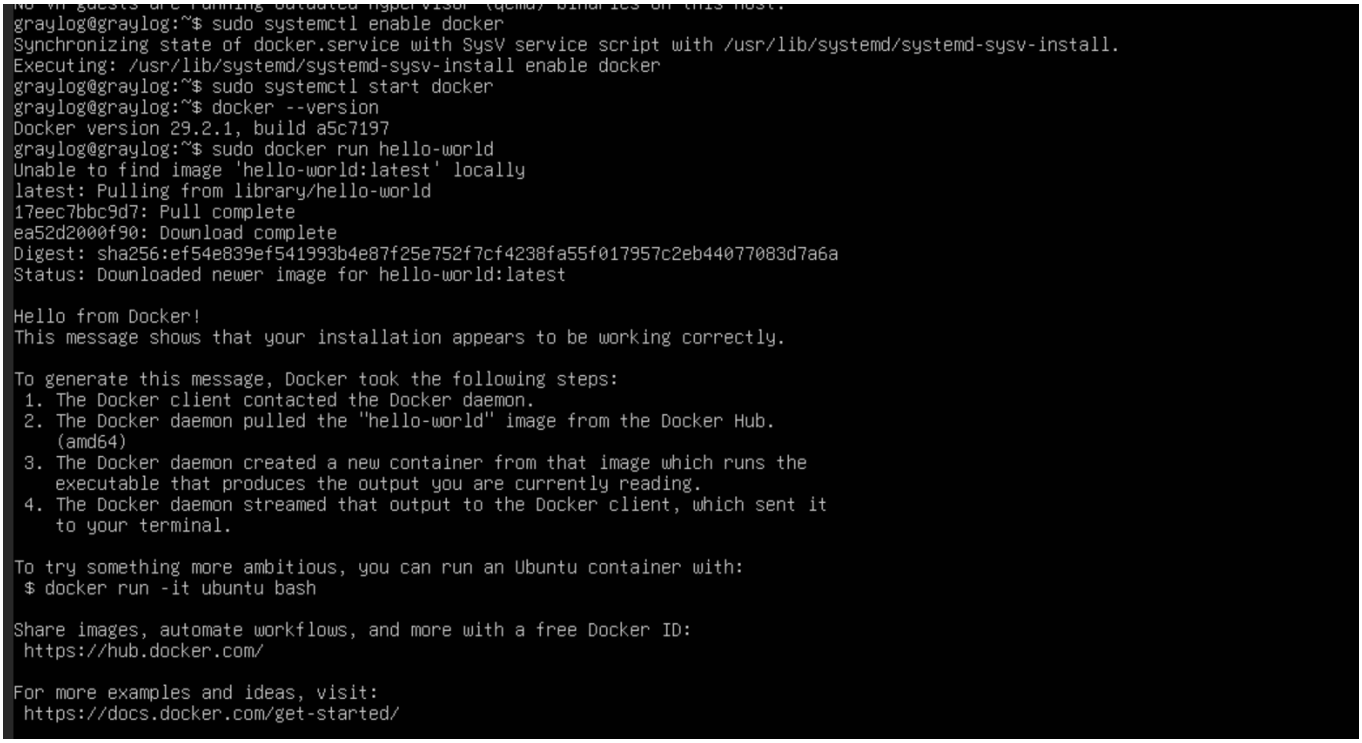
Ce composant permet de stocker les journaux collectés et de faciliter leur recherche et leur analyse dans l'interface Graylog.



<u>Etape 3 :</u> Préparation de l'environnement Docker

La plateforme Graylog est déployée à l'aide de Docker, ce qui permet de simplifier la gestion des différents services nécessaires à son fonctionnement.

Un test est réalisé afin de vérifier que Docker fonctionne correctement sur la machine.



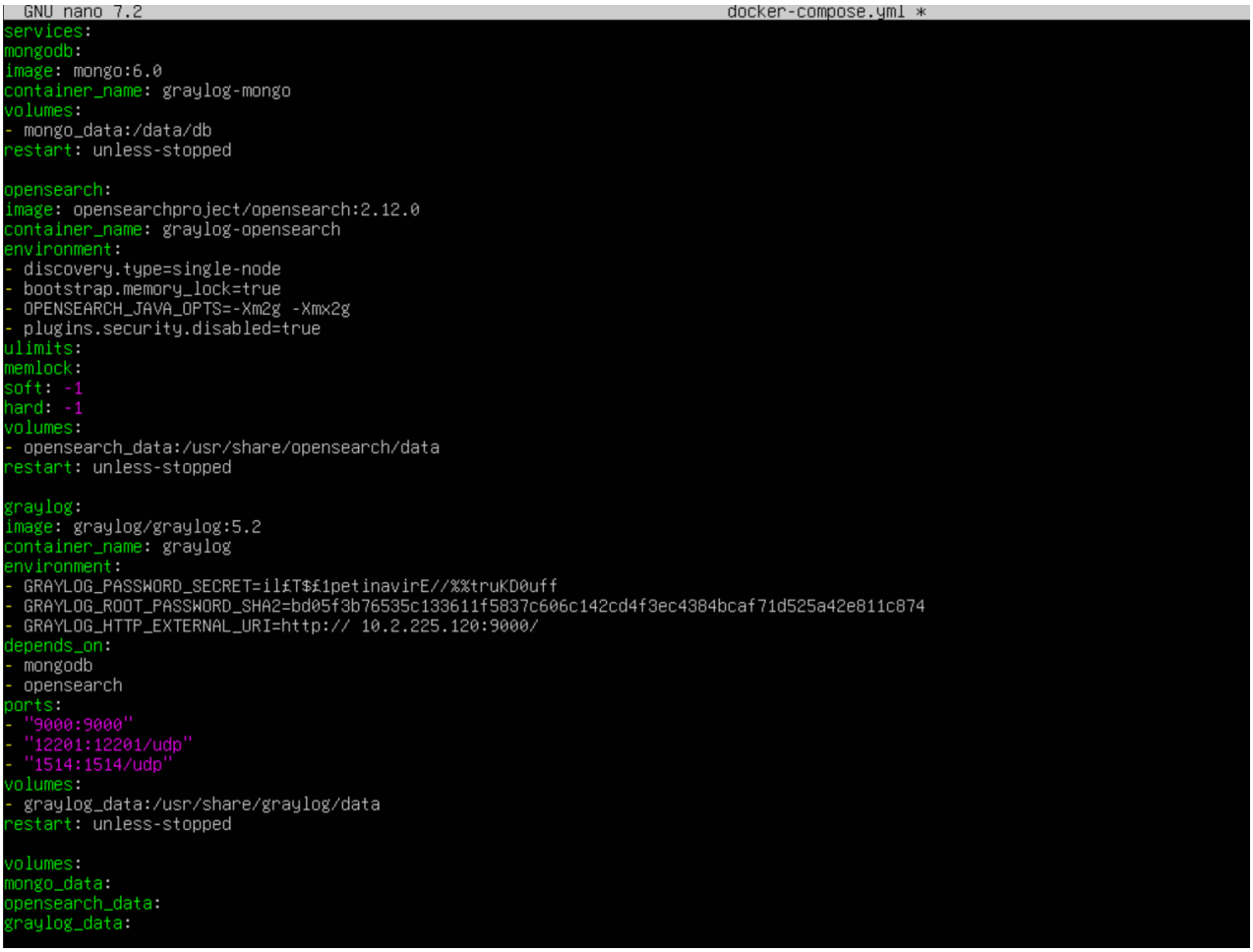
<u>Etape 4</u> : Configuration de l'infrastructure Graylog

Un fichier docker-compose.yml est créé afin de définir les différents services nécessaires au fonctionnement de Graylog.

Cette configuration inclut notamment :

- **MongoDB** pour la gestion des métadonnées
- **OpenSearch** pour l'indexation des logs
- **Graylog** pour la collecte et l'analyse des journaux

Ce fichier permet de définir les conteneurs Docker, les ports utilisés ainsi que les variables de configuration nécessaires au démarrage du service.



<u>Etape 5 :</u> Déploiement des conteneurs Graylog

Une fois la configuration terminée, les conteneurs sont lancés à l'aide de Docker Compose. Cette commande permet de démarrer automatiquement l'ensemble des services nécessaires à Graylog.

Etape 6 : Vérification du fonctionnement des services

Enfin, une vérification est réalisée afin de s'assurer que les différents conteneurs sont bien en cours d'exécution.

Cette étape permet de confirmer que MongoDB, OpenSearch et Graylog fonctionnent correctement, ce qui rend possible l'accès à l'interface web de Graylog.

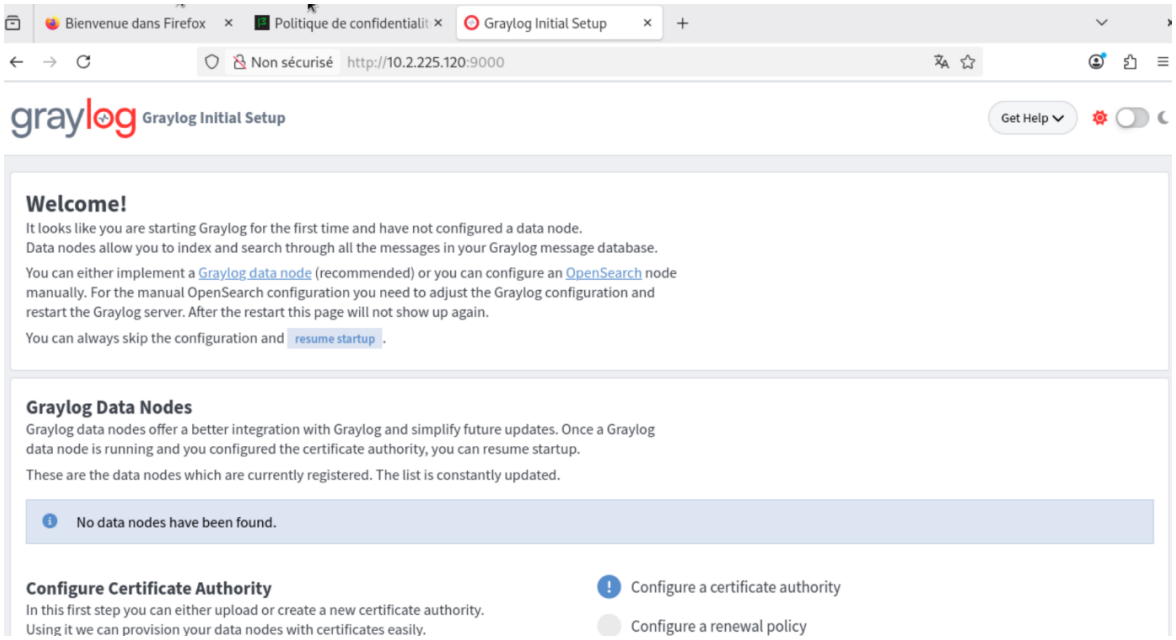
4.3 Configuration de l'interface web Graylog et réception des logs

Etape 1 : Accès à l'interface web Graylog

Une fois les conteneurs Docker démarrés, l'interface web de Graylog est accessible depuis un navigateur via l'adresse suivante :

http://10.2.225.120:9000

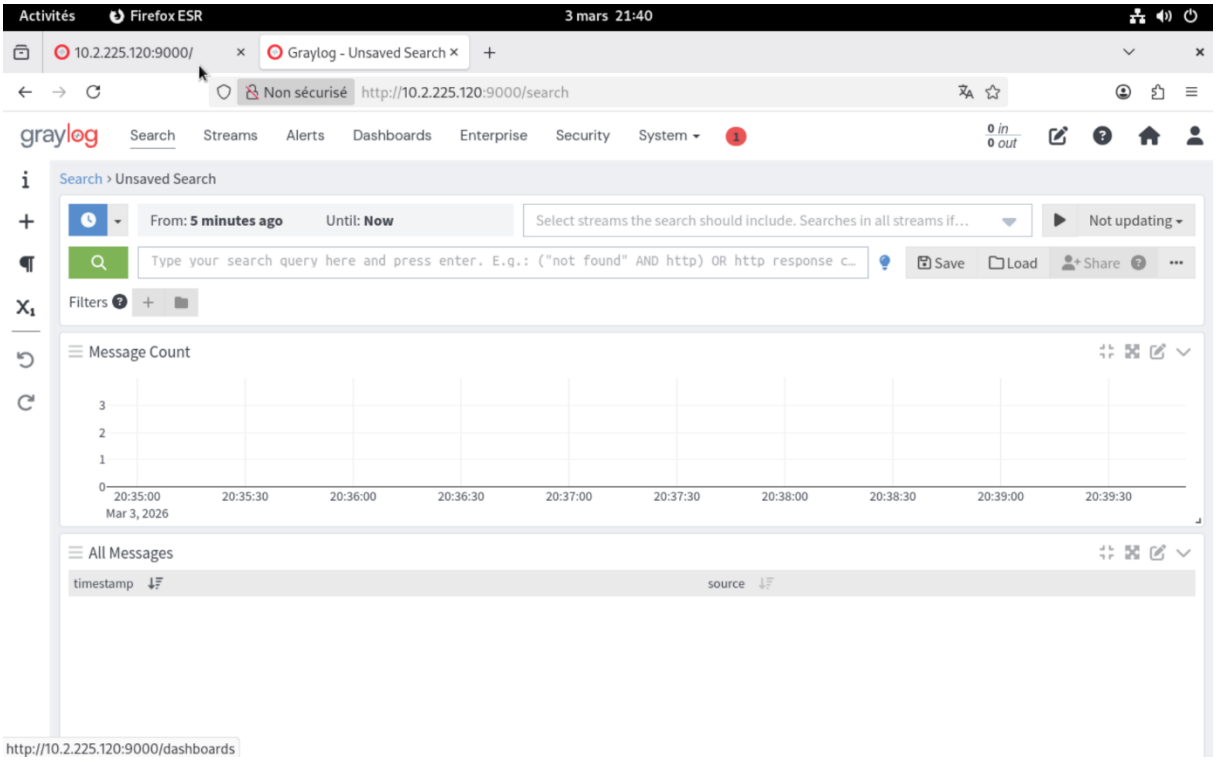
Lors du premier accès, Graylog affiche l'écran Graylog Initial Setup permettant de finaliser la configuration du serveur.



<u>Etape 2 :</u> Accès à l'interface principale

Après l'initialisation, l'utilisateur accède à l'interface principale de Graylog. Cette interface permet de :

- Consulter les logs
- Effectuer des recherches
- Créer des tableaux de bord
- Analyser les événements de sécurité

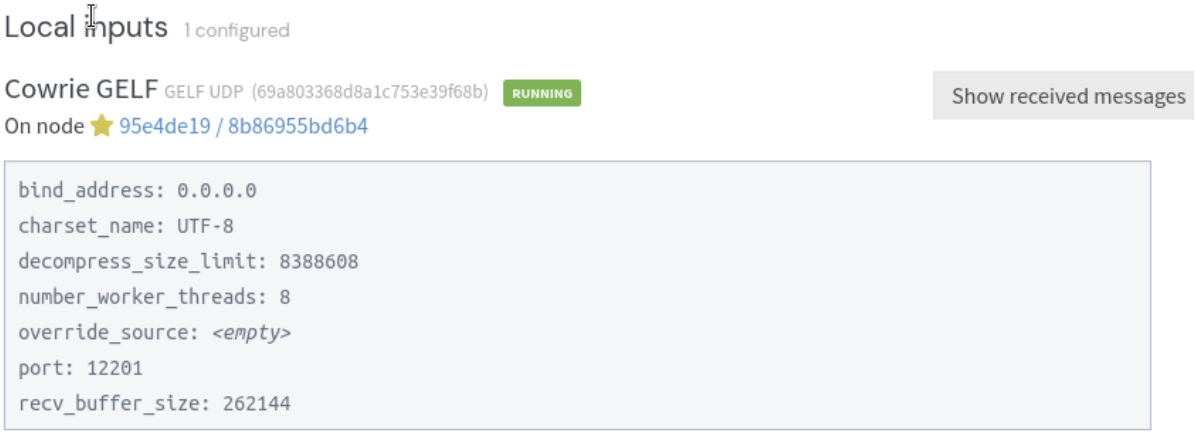


<u>Etape 3 :</u> Mise en place d'une entrée GELF

Afin de permettre à Graylog de recevoir les logs provenant du honeypot Cowrie, une entrée GELF est configurée dans l'interface.

Dans le menu System → Inputs, une nouvelle entrée GELF UDP est créée et associée au port configuré sur le serveur Graylog.

Cette entrée permet à Graylog de recevoir les messages envoyés par les différentes machines du réseau.



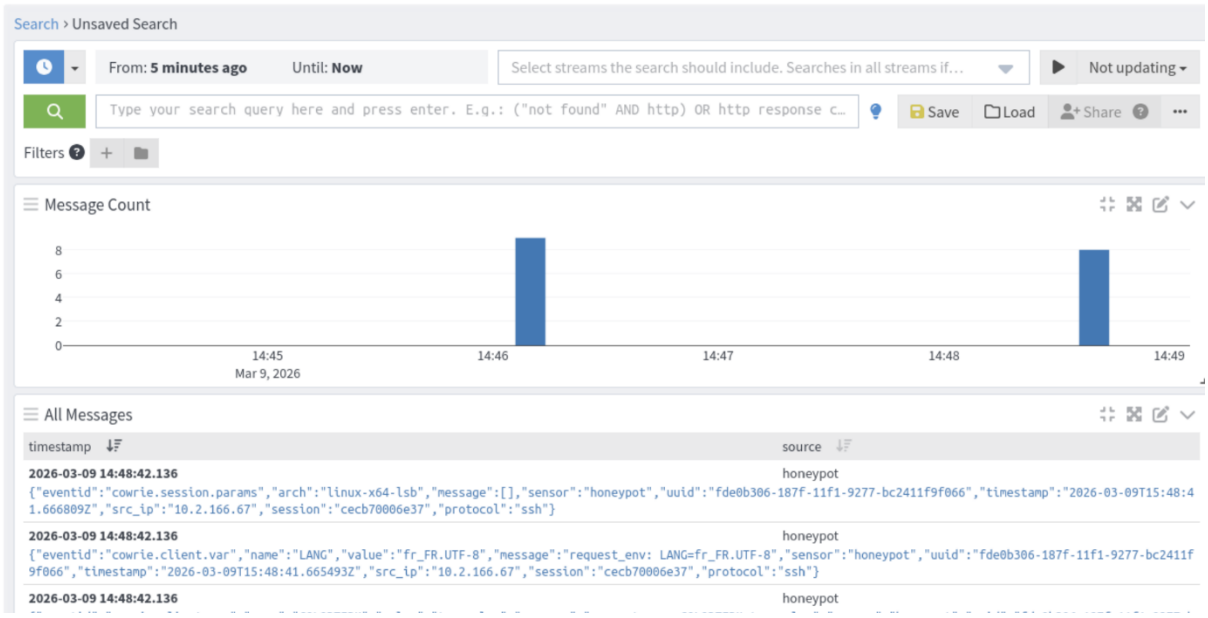
<u>Etape 4 :</u> Visualisation des logs du honeypot

Une fois l'entrée GELF active, les logs générés par le honeypot Cowrie sont automatiquement envoyés vers Graylog.

Les événements apparaissent alors dans la section **Search**, où il est possible d'observer différentes informations comme :

- L'adresse IP source de l'attaque
- Le protocole utilisé (SSH)
- La date et l'heure de la tentative
- Les actions effectuées par l'attaquant

Ces informations permettent d'analyser les tentatives d'intrusion et d'étudier le comportement des attaquants.



4.4 Configuration de la vm attaquante Kali

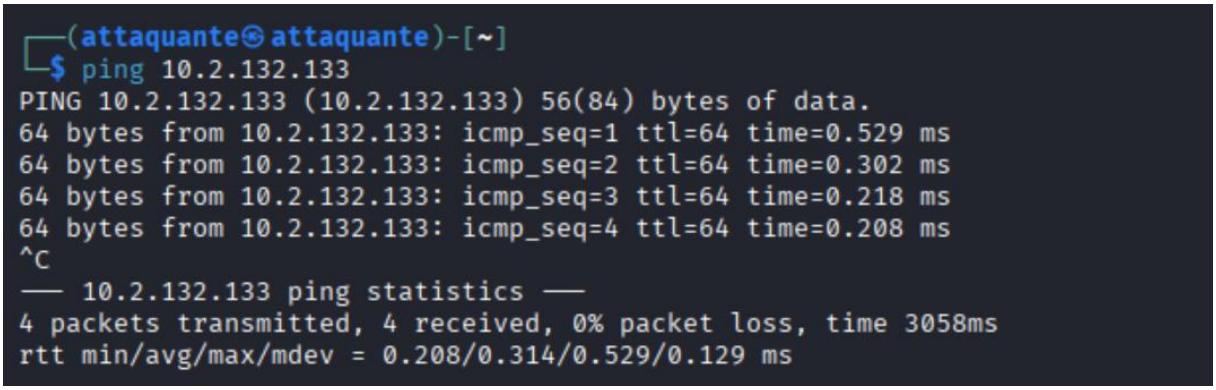
Etape 1 : Vérification de la connectivité réseau

Avant de lancer une attaque, il est nécessaire de vérifier que la machine cible est accessible.

La commande suivante est utilisée :

ping 10.2.132.133

Cette commande permet de tester la communication entre la machine attaquante Kali Linux et le honeypot.



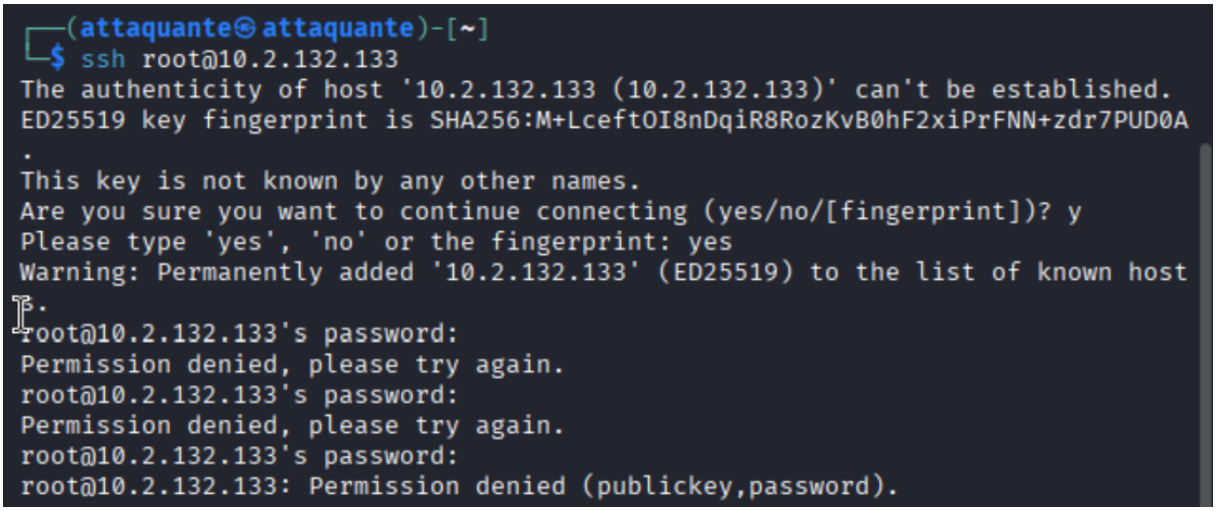
Etape 2 : Tentative de connexion SSH

Une tentative de connexion SSH est ensuite réalisée sur le honeypot.

Commande utilisée :

ssh root@10.2.132.133

Cette action permet de simuler une tentative d'accès au serveur SSH exposé par le honeypot.



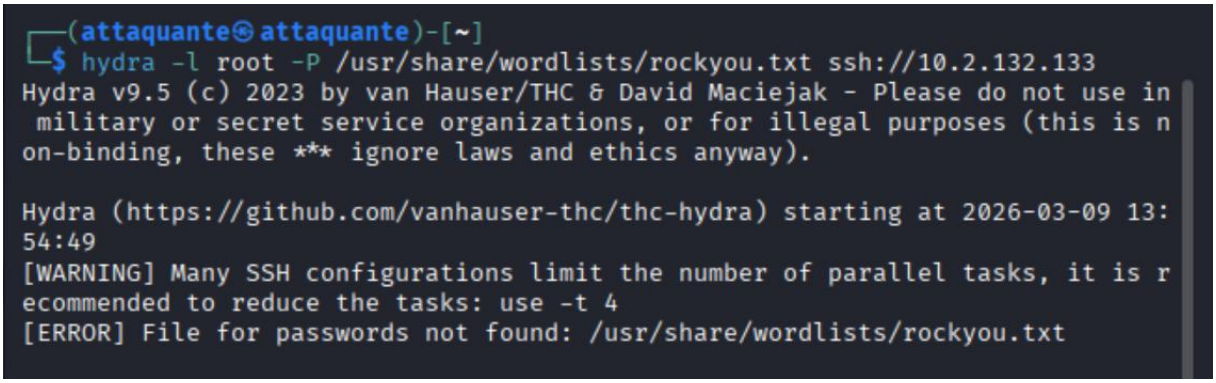
Etape 3 : Simulation d'une attaque par force brute

Pour simuler une attaque automatisée, l'outil **Hydra** est utilisé afin de tester plusieurs mots de passe.

Commande utilisée :

hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://10.2.132.133

Cette attaque permet de générer des tentatives d'authentification qui seront enregistrées par le honeypot et envoyées vers Graylog pour analyse.



5. Cybersécurité et politique de sécurité

5.1 Analyse des risques

RISQUE	IMPACT
Attaque brute force	Accès au serveur
Scan réseau	Découverte des ports
Injection de commandes	Compromission

5.2 Mesures de sécurité mises en œuvre

Plusieurs mesures de sécurité ont été appliquées afin de limiter les risques liés à l'exposition du honeypot et de garantir une meilleure surveillance des activités suspectes.

Isolation de l'environnement

Le honeypot Cowrie est déployé sur une machine virtuelle dans un environnement isolé. Cette isolation permet d'éviter qu'un attaquant puisse accéder au reste de l'infrastructure en cas de compromission du système. L'utilisation de la virtualisation permet également de tester des scénarios d'attaque sans impacter un environnement réel.

Centralisation des logs

Les logs générés par le honeypot sont envoyés vers la plateforme Graylog à l'aide du protocole GELF (Graylog Extended Log Format). Cette centralisation permet de regrouper les événements de sécurité dans une interface unique afin de faciliter leur analyse.

Grâce à Graylog, il est possible d'observer les tentatives d'attaque telles que :

- Les tentatives de connexion SSH
- Les adresses IP des attaquants
- Les actions réalisées sur le système

Cette visibilité permet d'améliorer la détection des comportements suspects.

Limitation de l'exposition réseau

Seuls les services nécessaires sont accessibles sur le réseau. Par exemple :

- Le port 2222 est utilisé pour simuler un service SSH sur le honeypot
- Le port 9000 permet d'accéder à l'interface web de Graylog

Cette limitation réduit la surface d'attaque et contribue à sécuriser l'infrastructure.

5.3 Surveillance et analyse des événements de sécurité

L'utilisation d'un honeypot permet de collecter des informations précieuses sur les méthodes utilisées par les attaquants.

Les tentatives d'intrusion réalisées depuis la machine Kali Linux sont enregistrées par Cowrie puis envoyées vers Graylog. Ces informations peuvent ensuite être analysées dans l'interface web afin d'identifier :

- Les tentatives de connexion SSH
- Les identifiants testés lors des attaques
- Les adresses IP des machines attaquantes
- Les commandes exécutées par l'attaquant

L'analyse de ces événements permet de mieux comprendre les techniques d'attaque et d'améliorer les stratégies de défense des systèmes d'information.

6 . TEST ET VALIDATION DE LA SOLUTION

6.1 Plan de tests

Test	Objectif	Méthode	Résultat attendu
Test 1	Vérifier la connectivité réseau	Ping depuis Kali vers le honeypot	Réponse du honeypot
Test 2	Vérifier le service SSH du honeypot	Connexion SSH depuis Kali	Tentative de connexion enregistrée
Test 3	Générer une attaque simulée	Utilisation de Hydra	Plusieurs tentatives enregistrées
Test 4	Vérifier l'envoi des logs	Consultation dans Graylog	Apparition des événements
Test 5	Vérifier l'ananlyse des logs	Observation dans l'interface web	Affichage des informations d'attaque

6.2 Résultats des tests

Test 1 : Vérification de la connectivité réseau

Procédure :

Depuis la machine Kali Linux, un test de connectivité est réalisé vers la machine hébergeant le honeypot.

Commande utilisée :

ping 10.2.132.133

Résultat :
Le honeypot répond correctement aux requêtes ICMP, ce qui confirme que la communication réseau entre les machines fonctionne.

Test 2 : Tentative de connexion SSH

Procédure :

Une tentative de connexion SSH est réalisée vers le honeypot afin de vérifier que le service est accessible.

Commande utilisée :

ssh root@10.2.132.133

Résultat :
Le honeypot simule un serveur SSH et enregistre les tentatives de connexion.

Test 3 : Simulation d'une attaque brute force

Procédure :

L'outil Hydra est utilisé afin de générer plusieurs tentatives d'authentification.

Commande utilisée :

hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://10.2.132.133

Résultat :
Les différentes tentatives de connexion sont enregistrées par le honeypot

Test 4 : Vérification de la réception des logs

Procédure :

Les logs générés par le honeypot sont envoyés vers Graylog via le protocole **GELF**.

Les événements sont ensuite consultés dans l'interface web Graylog.

Résultat :
Les logs apparaissent dans l'interface et contiennent plusieurs informations telles que :

- L'adresse IP de l'attaquant
- Le protocole utilisé (SSH)
- L'heure de la tentative de connexion
- Les actions réalisées

Conclusion :
L'ensemble des tests confirme que l'infrastructure mise en place fonctionne correctement. Les attaques simulées depuis la machine Kali Linux sont bien détectées

par le honeypot et les événements sont correctement centralisés et analysés dans Graylog.

7 . COMPETENCES MISES EN OEUVRE

La réalisation de ce projet a mobilisé plusieurs compétences du référentiel BTS SIO – option SISR, notamment dans les domaines de la conception d'infrastructure, du déploiement de services et de la supervision des systèmes.

a) Concevoir une solution d'infrastructure réseau

Dans ce projet, une phase d'analyse et de conception a été réalisée afin de définir l'architecture de l'infrastructure permettant de simuler des attaques et d'analyser les logs.

Les compétences mobilisées comprennent :

- Analyser un besoin exprimé et son contexte technique afin de définir les objectifs du projet.
- Élaborer un dossier de choix de solution d'infrastructure en sélectionnant les technologies adaptées telles que Cowrie, Graylog et Kali Linux.
- Choisir les éléments nécessaires pour assurer la qualité et la disponibilité d'un service, notamment en mettant en place une architecture basée sur plusieurs machines virtuelles.
- Maquetter et prototyper une solution d'infrastructure, grâce à la création d'un environnement virtualisé permettant de tester les différentes configurations.

b) Installer, tester et déployer une solution d'infrastructure réseau

La mise en œuvre technique du projet a permis de déployer les différents composants nécessaires au fonctionnement de l'infrastructure.

Les activités réalisées incluent :

- Installation et configuration des éléments d'infrastructure, notamment les machines virtuelles Linux utilisées pour le honeypot et le serveur Graylog.
- Installation et configuration des services nécessaires, comme le honeypot Cowrie et la plateforme de centralisation des logs Graylog.

- Rédaction de la documentation technique, décrivant les différentes étapes de mise en place de l'infrastructure.
- Test de l'intégration de la solution, en simulant des attaques depuis la machine Kali Linux afin de vérifier la collecte des logs.

c) Exploiter, dépanner et superviser une solution d'infrastructure réseau

Une fois l'infrastructure mise en place, des opérations de supervision et d'analyse ont été réalisées afin de vérifier le bon fonctionnement du système.

Les compétences mobilisées comprennent :

- Administration des éléments de l'infrastructure, notamment la gestion des machines virtuelles et des services déployés.
- Gestion des fichiers d'activité et des logs, grâce à l'utilisation de Graylog pour centraliser et analyser les événements de sécurité.
- Identification et analyse des incidents de sécurité, en observant les tentatives d'intrusion simulées depuis la machine Kali Linux.
- Évaluation et amélioration de la qualité du service, en vérifiant que les logs sont correctement collectés et analysés

8 CONCLUSION

8.1 Bilan du projet

Ce projet m'a permis de mettre en place une infrastructure permettant de simuler des attaques informatiques et d'analyser les événements de sécurité grâce à un honeypot et à un système de centralisation des logs.

Les difficultés rencontrées :

- Configuration et installation de Graylog avec Docker
- Problèmes de connexion entre les machines virtuelles
- Configuration de l'envoi des logs vers Graylog
- Vérification du bon fonctionnement du honeypot

Les points forts :

- Mise en place d'un environnement réaliste de cybersécurité, permettant de simuler des attaques informatiques.
- Utilisation d'outils professionnels tels que Cowrie, Graylog et Kali Linux.
- Centralisation et analyse des logs, permettant d'observer les tentatives d'intrusion et d'identifier les comportements suspects.
- Architecture virtualisée, facilitant les tests et la reproduction de scénarios d'attaque.
- Compréhension des mécanismes d'attaque et de surveillance, essentiels dans le domaine de la cybersécurité.

8.2 Apport personnels

Ce projet m'a apporté :

- Compréhension des mécanismes d'attaque sur les services SSH
- Apprentissage de l'utilisation de Graylog pour l'analyse des logs
- Amélioration des compétences en administration Linux
- Découverte des outils de cybersécurité

8.3 Perspectives d'avolution

Plusieurs améliorations pourraient être apportées à cette infrastructure afin d'améliorer la détection et l'analyse des tentatives d'intrusion :

- Ajout de tableaux de bord dans le Graylog afin de visualiser plus facilement les tentatives d'attaque.
- Mise en place d'alerte automatiques lorsqu'un événement suspect est détecté.
- Ajout d'autres types de honeypots afin de simuler différents services (http, ftp,...)

9 . CONCLUSION

La réalisation de ce projet m'a permis de mettre en place une infrastructure capable de simuler et d'observer des tentatives d'intrusion. Grâce au honeypot Cowrie, il a été possible de reproduire le comportement d'un serveur SSH afin d'attirer des attaques et d'enregistrer les actions réalisées lors des tentatives de connexion.

Les informations collectées ont ensuite été centralisées et analysées à l'aide de Graylog, ce qui permet de visualiser les événements de sécurité et de mieux comprendre les méthodes utilisées par les attaquants. Les tests réalisés depuis la machine Kali Linux ont permis de vérifier le bon fonctionnement de l'ensemble du système et de générer des logs exploitables pour l'analyse.

Ce projet m’a également permis de développer des compétences en administration Linux, en mise en place d’infrastructures virtualisées et en analyse de logs de sécurité. Il illustre l’intérêt des honeypots dans une démarche de cybersécurité, car ils permettent d’observer les attaques tout en limitant les risques pour les systèmes réels.

Enfin, ce travail m’a permis de mieux comprendre les enjeux liés à la sécurité des systèmes d’information et l’importance de mettre en place des outils permettant de détecter et d’analyser les activités suspectes.