***Luis MIRANDA***

***Khalil HACIB***

***4SI4***

**Document Technique:**

**HoneyMonit**

**Mise en place d'un honeypot monitorée par un SIEM**

* Outil conçu pour enregistrer des attaques par force brute, l'interaction du shell entre autres... effectuée par un attaquant. Observer son comportement vis-à-vis d'un autre système.

SOMMAIRE

[Introduction 3](#_Toc109303466)

[Analyse Technique du Projet 4](#_Toc109303467)

[Solutions choisies 5](#_Toc109303468)

[Mise en œuvre 7](#_Toc109303469)

[Honeypots 7](#_Toc109303470)

[Kfsensor 7](#_Toc109303471)

[Cowrie 9](#_Toc109303472)

[Wordpot 11](#_Toc109303473)

[SIEM 13](#_Toc109303474)

[App Tango Honeypot 18](#_Toc109303475)

[Outil de contrôle d'intégrité 20](#_Toc109303476)

[Attaque / Autres Vulnérabilités 21](#_Toc109303477)

[Conclusion 23](#_Toc109303478)

# Introduction

Tout d’abord, ce projet intitulé « Mise en place d'un honeypot monitorée par un SIEM» nous a directement attiré. A la différence d’autres projets qui étaient soit pas très parlant, on n’était pas sûr par où s’y prendre.

Celui-ci avait le bon mélange entre très intéressant et très profond.

Avec ce projet on a fortement amélioré nous connaissances sur tout ce qui est en lien avec les honeypots et le fonctionnement d’un SIEM. Et on sait aussi qu’on aura encore d’autres aspects à apprendre avec le temps pour nous permettre d’aller encore plus en détail dans nos analyses du comportement d’un attaquant et arriver à une analyse encore plus complète. C’est un sujet énorme qu’on verra certainement beaucoup dans notre futur professionnel et on est content d’avoir pu l’aborder a l’occasion de ce projet Annuel.

Ce projet aller premièrement nous défier dans l’aspect de prise de connaissance, en effet on connaissait très peu sur les honeypots. On connaissait de nom mais on n’avait jamais utilisé. De même Splunk on avait abordé dans un cours cette année à l’ESGI et ce projet annuel nous a permis de prendre ses connaissances les mettre en pratique et aller plus loin pour atteindre nos buts pour ce projet.

Cette partie serait longue mais très instructive, nous cours à l’ESGI, nous connaissances et tout ce qu’on apprenait souvent grâce à des tutoriels ou guides en ligne. Nous a permis d’entamer ce projet. Et avoir une bonne vision globale sur le projet afin de pouvoir vraiment commencer à réfléchir à ce qu’on va choisir comme honeypot, outil d’intégrité, SIEM etc...

# Analyse Technique du Projet

Une fois la globalité du projet cerné on a eu un certain nombre de points à aborder.

Le plus important serait de voir quelle infrastructure on allait choisir, on devait choisir une infrastructure qui nous permettra le mieux et de la façon la plus complète possible d’analyser le comportement d’un attaquant. Nous avions décidé de réaliser ce projet dans un environnement local, ce qui voudrait dire qu’on devait nous-même réaliser une attaque fictive afin de souligner les outils qu’on a mis en place. Or on aurait pu mettre notre outil sur internet sans problème hormis le prix pour héberger les serveurs.

Une fois l’infrastructure choisi, faudrait choisir les honeypots qu’on allait mettre en place, quels sont les logs qu’on peut récupérer de ces honeypot, est-ce qu’ils sont intéressants, etc...

Le SIEM, est l’outil destiné à centraliser les milliers (voire les millions) d'évènements issus de notre infrastructure. Le choix de vendeur est large entre Splunk, IBM, SolarWinds, McAfee entre autres mais Splunk reste le plus utilisé par le monde professionnel .

Un des points de ce projet était aussi l’utilisation d’un outil d’intégrité, ceci n’est pas proposé par les SIEM directement donc un autre outil doit être choisi, avec la possibilité d’importer sur splunk les données de cet outil.

Finalement, il faut réfléchir à une attaque qui est tout à fait possible par un attaquant souvent biais du brute force qui arrive à accéder à nos machines avec honeypot afin qu’on puisse analyser derrière ce qu’il a fait et essayer de retracer le plus possible ce qui s’est passé.

# Solutions choisies

Notre premier choix c’est fait rapidement. On était tous les deux d’accord qu’on devait avoir au moins un poste Linux et un poste Windows. En plus de ça on allait avoir besoin d’un poste pour le serveur SIEM(systèmes de gestion des événements et des informations de sécurité) .

Pour cela on a choisi un Debian 11 pour notre poste Linux, celui-ci imiterait une machine Web en production.

Le choix d’un Windows 10 pour le poste Windows est logique cela servirait comme un poste utilisateur, ainsi qu’un serveur de transfert de fichiers à distance.

Pour notre machine SIEM, il nous semblait logique d’avoir un Windows Server, on a pris le Windows Server 2022 cela nous permet aussi d’utiliser ce projet pour l’utiliser et tester cette nouvelle version.

Une fois l’infrastructure décidé, il fallait choisir les honeypots. Ceci est peut-être encore plus important en effet c’est ce qui va attirer les attaquants à nos serveurs, si nos serveurs ne sont pas très intéressants personne souhaitera tenter sa chance et en retour on aura très peu à analyser. (Des scan shodan par exemple si accessible depuis internet)

Pour Windows très peu de honeypots existent, on a passé un peu de temps avec Valhala et kfsensor.

Ces deux honeypots réalisent plus au moins la même chose, ils permettent d’ouvrir des ports et analyser toute tentative de connexion sur ces ports, des tentatives de scan etc... Il permet donc d’ouvrir des ports intéressants sur la machine qu’attirera un attaquant à jeter un coup d’œil.

Kfsensor détenait quand même plus de fonctionnalités et c’était donc un produit plus complet donc on l’a choisi.

Pour Linux on avait beaucoup plus de choix, mais on souhaitait rester dans le thème d’un serveur Web donc on a cherché des honeypots Web/WordPress on a trouvé un appelé Wordpot. On ne pouvait pas faire grand-chose avec mais son principal but c’était pour nous d’avoir une page web sur un des ports Web. De plus, il était possible de par exemple réaliser du brute force au niveau de la page login et analyser cela dans un fichier log, donc toute tentative de connexion au site web on pourrait analyser.

De plus, on souhaite trouver un honeypot interactif sur lequel l’attaquant pouvait vraiment avoir la sensation qu’il progressait et donc on supposerait qu’un changement de comportement aurait lieu. On pourrait vraiment analyser ce que l’attaquant chercherai à faire s’il avait le plein pouvoir sur un serveur.

Changerait-il le système, est-ce qu’il chercherait des informations confidentielles, chercher à créer un backdoor, supprimer/rendre obsolète le système.

Cela était très intéressant pour nous donc on a décidé de trouver un honeypot qui donnerai un accès shell à l’attaquant et comme c’est un honeypot ce serait une session virtuelle et qu’aucune des commandes aurait un effet sur la machine en elle-même.

On a trouvé Cowrie. Cowrie développé en Python permet de faire exactement cela, un terminal dans lequel on a l’impression d’être dans un serveur en production, pouvoir rentrer des commandes, avoir des retours réalistes comme par exemple /etc/passwd. Sans modifier quoique ce soit sur la machine en elle-même.

Pour le SIEM, le choix fut très simple on avait tous les deux déjà utilisé Splunk a un niveau très débutant, on allait cette fois ci chercher à aller plus loin, pour rendre ce projet possible.

Splunk va nous permettre de rendre les données générées par des machines compréhensibles, en identifiant des tendances de données, en donnant des outils de mesure grâce notamment aux alertes qu’on peut créer.

En effet, ces données seront celles générées par nos honeypots et d’autres audits d’events Windows ou Linux (sysmon, auditd).

On devait aussi choisir un outil d’intégrité, on a trouvé un sur Linux, finalement sur notre projet son utilisation était limitée mais si on déciderait d’aller plus loin dans ce projet un jour, cela deviendrait très intéressant comme outil à analyser.

Pour notre projet on a choisi AIDE (Advanced Intrusion Detection Environment).

AIDE nous permet grâce un crontab qu’on a configuré dans le système d’avoir tous les jours un résumé qui nous dit pour les dossiers qu’on souhaite. S’ils ont été supprimés, modifié, crée, à quelle heure, la taille des fichiers, hash des fichiers entre autres.

# Mise en œuvre

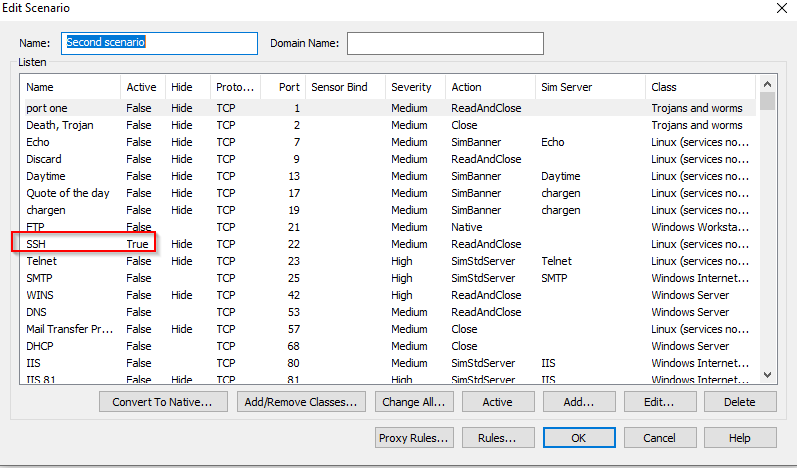
## Honeypots

### Kfsensor

On a vérifié un honeypot similaire sur Windows, kfsensor était supérieur avec plus de fonctionnalités.

Son installation est assez simple. C’est sa configuration qui est intéressant à en parler, un peut créer plusieurs scénarios, ceux-ci servent à ouvrir des ports qui sont autre fois fermée et décider ce qu’un attaquant voit comme résolution de connexion sur ce port.

Cela crée un fichier log qu’on peut récupérer avec notre SIEM pour analyse extensive et créer des alertes.



Pour ce projet on a décidé d’ouvrir le port SSH, ce qui sur un Windows n’est pas très commun et va surement attirer l’œil d’un attaquant.

Une image contenant texte

Description générée automatiquement

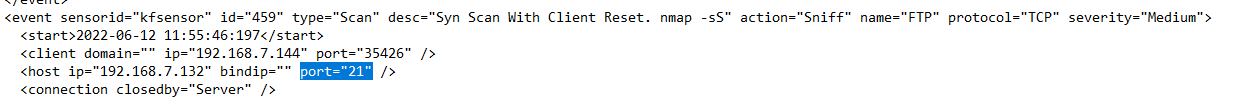
Résultat scan nmap sur notre machine Windows 10 (le point d’entrée pour l’attaquant)

Une image contenant texte

Description générée automatiquement

Image du fichier log Kfsensor qui détecte un scan nmap sur plusieurs ports dont le port 22 ouvert par ce honeypot.

Ce fichier log nous permet aussi de voir quels autres ports on subit un scan nmap même si on n’a pas configuré Kfsensor pour ouvrir ces ports

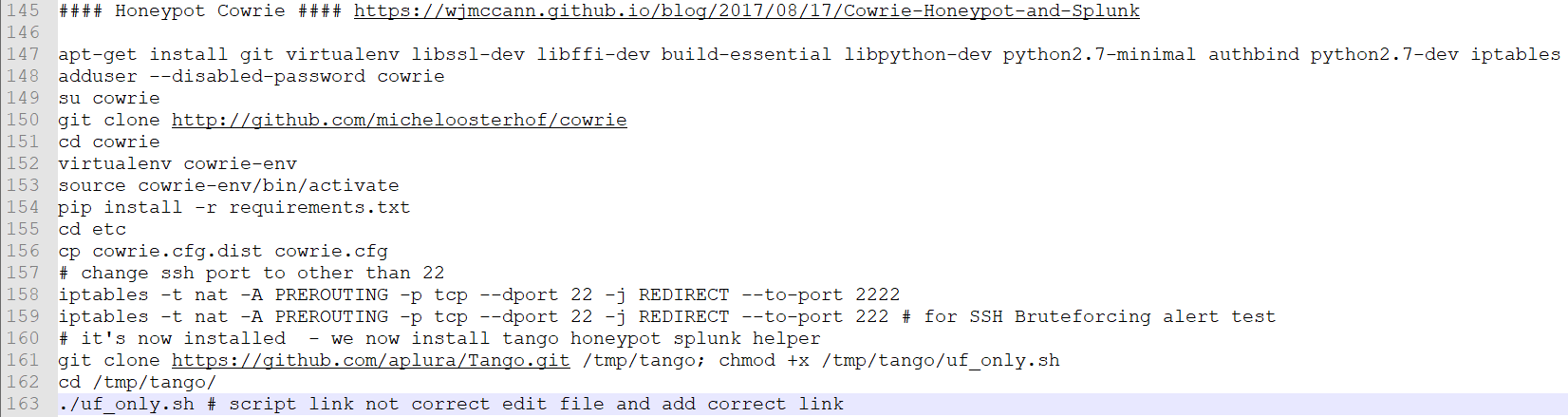


Notamment le port 21 (FTP), qu’on a configuré comme le point d’entrée sur la machine Windows pour l’attaquant.

Ce fichier log est reçu par Splunk qui nous informe de ce scan nmap.

### Cowrie

Cowrie est assez intéressant en effet, développé en python cela permet de créer un terminal virtuel auquel un attaquant se connecte, assez facilement il récupère une session en tant que root et peut commencer à taper certaines commandes.



*Installation Cowrie Linux*

L’environnement ressemble fortement à un environnement de production ce qui induit au moins pendant quelques instants ou minutes un attaquant à baisser sa garde et essayer les commandes qu’il aurait essayé ayant récupéré une machine en tant que root, quelles sont ces intentions, leur philosophie d’attaque etc.… ceci nous permet de récupérer un fichier log riche en éléments à analyser.

Une image contenant texte

Description générée automatiquement

On regarde le fichier log

Une image contenant texte

Description générée automatiquement

On retrouve les commandes lancées par l’attaquant, un ID de session, un timestamp ainsi que le source IP dans ce cas-là le source IP c’est notre machine Windows 10 ce qui veut dire quelle est déjà compromis.

Une image contenant texte

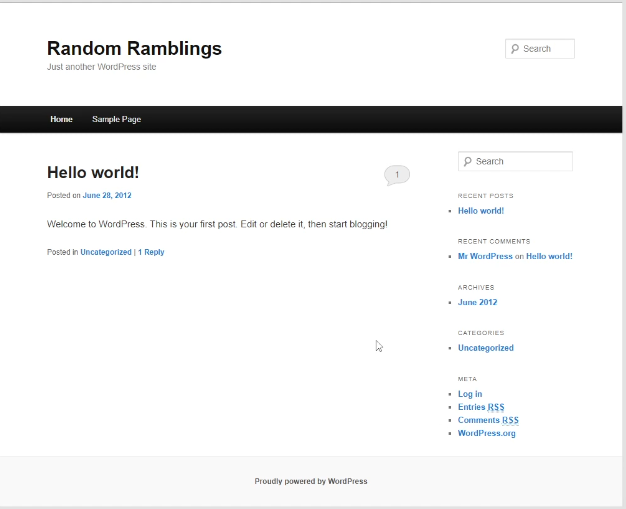
Description générée automatiquement

### Wordpot

Notre honeypot bonus sur Linux, notre Linux était considérée en tant que Web Server donc il est logique de choisir un honeypot Web pour cela. On a choisi on honeypot Wordpress aussi développé en Python assez simple à exécuter



On retrouve une page banale Wordpress avec par exemple l’option de se connecter sur cette page, une tentative de brute force ce login peut être remonté par un fichier log que on transmet à Splunk pour analyse.



Une image contenant texte

Description générée automatiquement

*Fichier log Wordpot remonté sur Splunk par le forwarder*

On remarque par exemple les mots de passe en claire essayé par l’attaquant ainsi que l’IP de l’attaquant.

Comme pour tous les autres honeypots, on rajoute un monitor à ce fichier ce qui l’ajoute à cette liste que le Splunk forwarder remonte une fois installé et configuré.

Une image contenant texte

Description générée automatiquement

## SIEM

Splunk est un outil très riche qui nous offre tout ce dont on a besoin pour créer des alertes pour note infrastructure.

Une image contenant texte, capture d’écran, moniteur, écran

Description générée automatiquement

*Screenshot du Dashboard Splunk*

Splunk permet la collète de données et nous permet de la trier grâce à des requêtes dans un barre de recherche.

Son installation est assez simple donc on va juste aborder notre configuration d’indexes, alertes et comment cela marche.

Avant tout, faut également installer l’application Splunk Forwarder sur les machines distantes sur lesquelles on veut envoyer les données.

Une image contenant texte

Description générée automatiquement

*Configuration Linux, par exemple de l’envoie de données auditd avec Splunk Forwarder*

Une image contenant texte

Description générée automatiquement

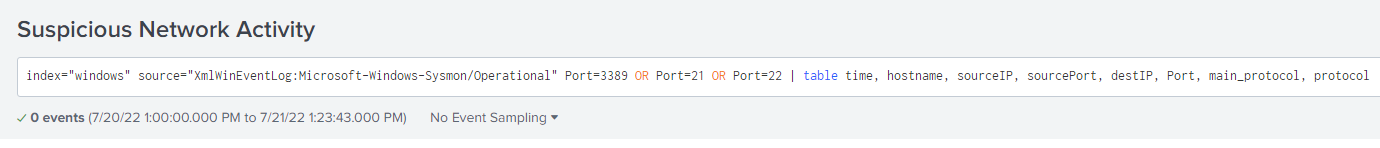
*Image contenant les principaux source types qu’on collecte avec notre serveur Splunk.*

Les évènements proviennent principalement pour notre projet de sysmon et directement des fichiers log des honeypot. Sur l’image au-dessus on remarque aussi 95% pour auditd mais pour notre projet on n’utilise pas vraiment car a aucun moment l’attaquant a la main sur la machine Linux directement, bien sûr que si un jour on décide d’aller plus loin avec ce projet, auditd serait déjà en place.

Une image contenant texte

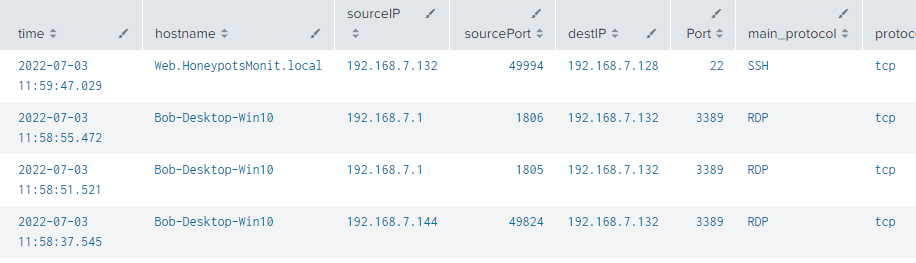
Description générée automatiquement

*Exemple Alertes*

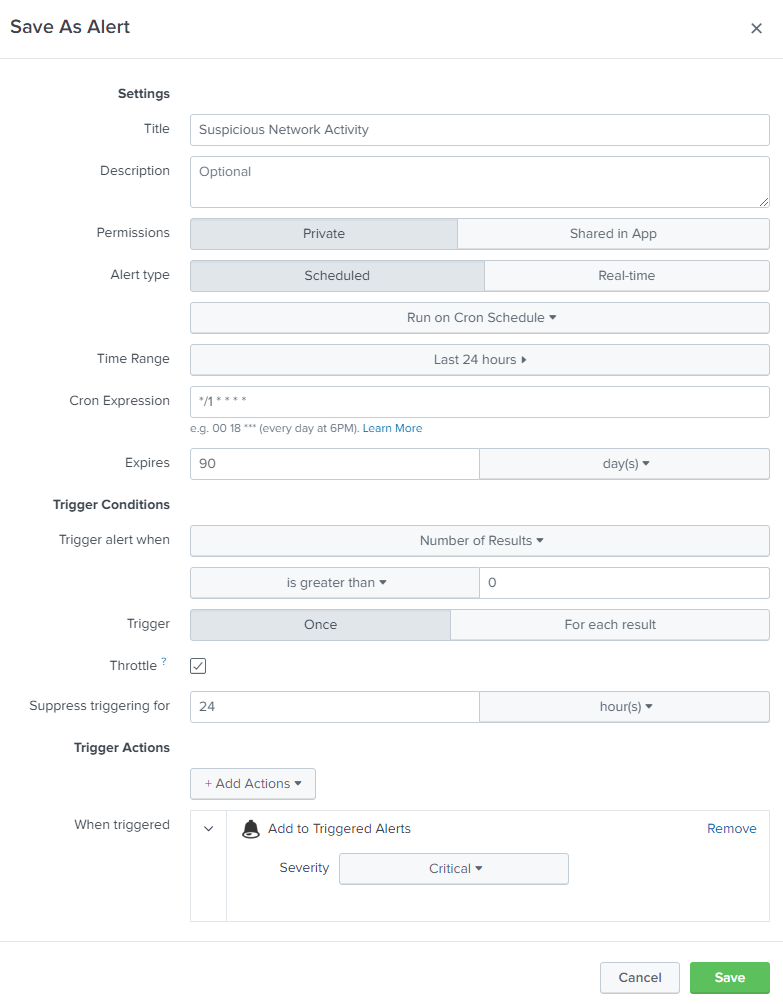
*Exemple requête d’alerte*

Prenant cette requête comme exemple sur celle-ci nous allons vérifier dans notre Sysmon des tentatives de connexion sur les ports 3389 (RDP), 21(FTP) ou 22(SSH) en effet ces 3 ports sont ouverts sur la machine Windows donc ce sont les 3 ports qui nous intéressent à surveiller.

Ensuite après le pipe on dit à Splunk que on veut le résultat de notre requête sous la forme d’une table avec seulement les champs qu’on a cité en effet pour des alertes il est important d’avoir accès aux données les plus importantes et qu’on risque d’avoir besoin directement. Si besoin on peut toujours vérifier les autres champs plus tard.

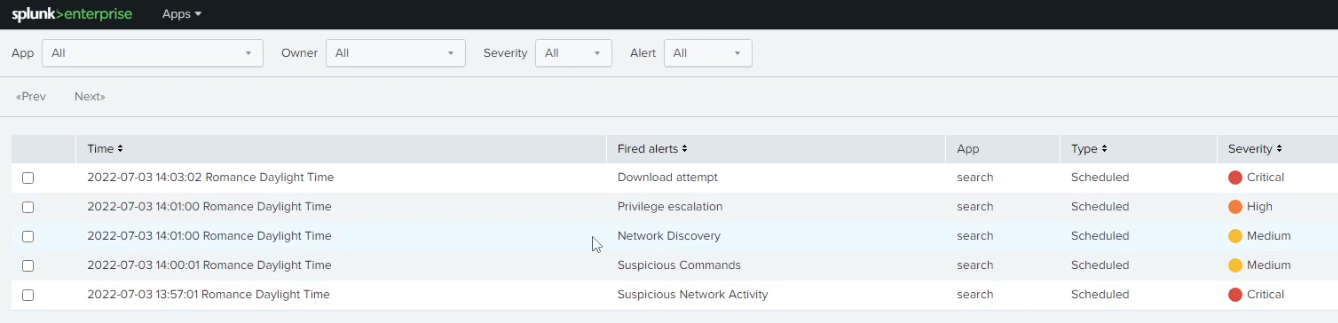


*Exemple résultat requête*



*Exemple création d’alerte*

Pour nos alertes on a pris la décision que le mieux était de les exécuter dans un Schedule cron \*/1 \* \* \* \* , ce qui équivaut à « tous les minutes », ça veut dire tous les minutes on vérifie si notre requête retourne des résultats si on ne reçoit pas de résultat tout va bien, on ne se fait pas attaquer, si on a un résultat il y a des très fortes chances qu’il s’agit d’un attaquant encore plus quand il s’agit dans un réseau honeypot sur lequel personne est normalement connecté pour trigger ses alertes. Une fois un alerte pour une requête trigger ce même alerte on le met un throttle de 24h sur lequel on est en alerte. Puis 24h plus tard s’il est trigger à nouveau on aura à nouveau droit à une nouvelle alerte pour nos prévenir que l’alerte est bien pour une nouvelle attaque pas le même de 24h avant. La raison aussi pour laquelle on prend un time range de 24h sur l’alerte.

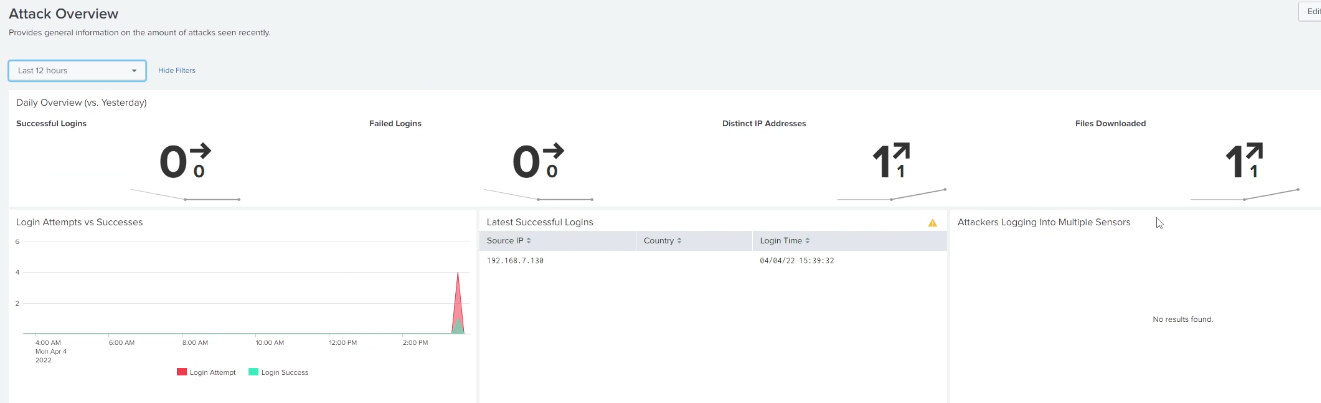
**

*Exemple alertes actionnés après une attaque sur le réseau*

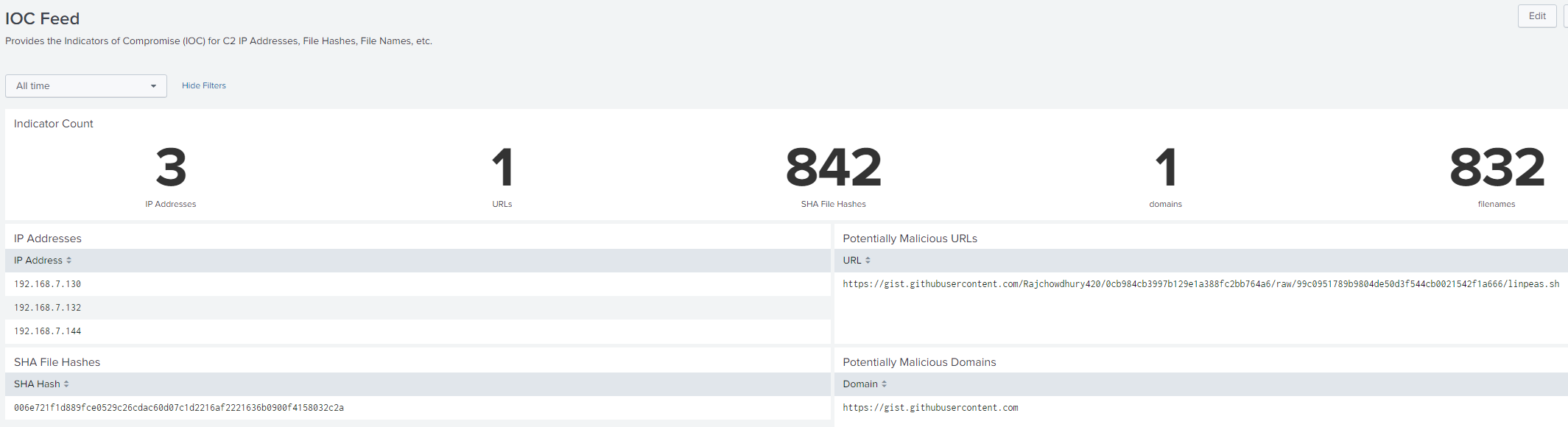
L’ordre de trigger des alertes nous aide également à comprendre comment l’attaque a eu lieu, avec le niveau de sévérité admis a une alerte dans un vrai réseau honeypot on pourrait avoir des démarches différentes a exécuter pour remédier cela, par exemple si on trouve qu’on a assez de données sur l’attaquant et qu’il est toujours connecté au réseau on pourrait l’éjecter du réseau.

## App Tango Honeypot

Sur Splunk il est aussi possible d’installer des applications afin d’aider dans l’analyse de certaines données pour Cowrie par exemple, l’app Tango Honeypot était recommandé, il permet de remontrer les IPs trouvé, vérifier des tentative d’upload de fichiers, et même les vérifier sur VirusTotal grâce à leur hash.



*Tentatives Login*

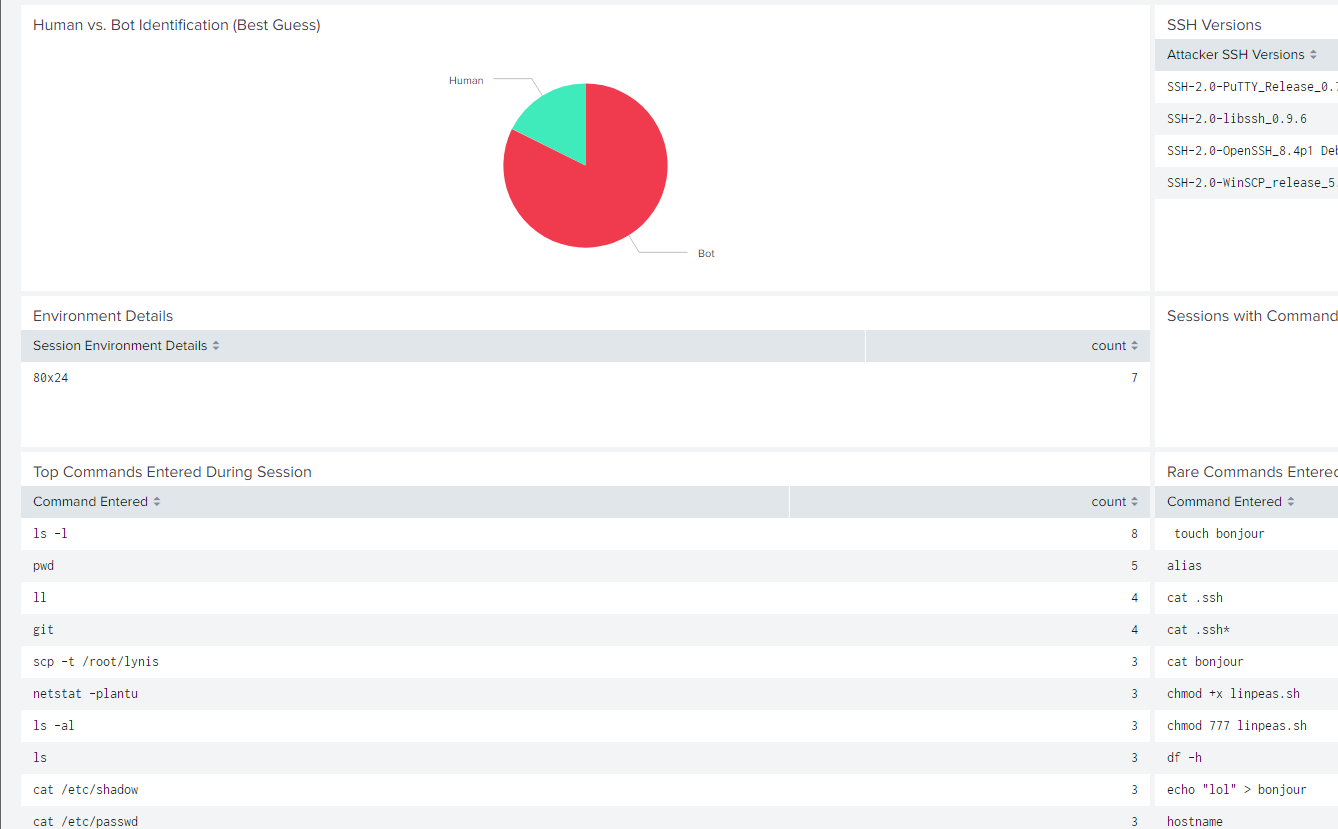


*Image IOC Feed*

Une image contenant texte

Description générée automatiquement

*Domaines*



*Analyse Session*

## Outil de contrôle d'intégrité

Pour l’outil d’intégrité (AIDE), cela nous a permis d’observer notamment le comportement dans le répertoire maison de l’honeypot Cowrie lorsque celui est lancé.

Toute tentative de download, on peut visualiser le fichier téléchargé coté linux.

Une fois Cowrie a l’arrêt se répertoire se vide.

Une image contenant texte

Description générée automatiquement

Aide tourne grâce un daily crontab qui crée un fichier aide.db.new qui compare avec le précèdent aide.db et vérifie pour les répertoires qu’on a décrit avec les règles qu’on a choisi en autres s’il y a eu des modifications, suppressions, créations, la taille du fichier, ses hash etc...

Pour notre projet on a décidé il était suffisant de créer une seule règle qu’on a appelé MYRULE, sur celle-ci on vérifie les p(permissions), n(nombre de liens), u(users), g(group), s(taille), m(mtime), c(ctime), xattrs (attributs étendus du fichier) et dernièrement les checksum md5 et sha512

Une image contenant texte

Description générée automatiquement

*Extrait d’un fichier log AIDE*

Sur celui-ci on voit bien l’évolution des répertoires qu’on a configuré AIDE pour surveiller.

Une image contenant texte

Description générée automatiquement

*Fichier log centralisé sur Splunk*

## Attaque / Autres Vulnérabilités

A travers notre présentation on présentera une vidéo d’attaque ainsi qu’ensuite une vidéo qui retrace l’attaque dans Splunk afin d’avoir une attaque réaliste on a aussi rajouté certaines vulnérabilités dans notre infrastructure plutôt intéressantes, cela rend notre attaque plus réaliste ou similaire à ce qu’on peut-être retrouver dans un capture the flag(ctf), ainsi que cela donne des actions additionnelles à devoir retracer plus tard dans notre SIEM.

Tout d’abord, on souhaitait avoir un point d’entrée dans notre machine Windows. L’honeypot qu’on a choisi ne nous permettez pas ceci, donc on a décidé de rajouter un serveur FTP sur la machine Windows accessible en mode Anonymous.

Une image contenant texte

Description générée automatiquement

Sur celui-ci on a un fichier password qui donne illusion que le mot de passe de l’utilisateur « Bob » est faible et donc serait vulnérable à une attaque par bruteforce.

Une image contenant texte

Description générée automatiquement

Cela rend plus intéressant notre nmap sur la machine où on découvre beaucoup de ports ouverts certains ouvert par l’honeypot d’autres dont celui-ci vraiment ouverts.

On a aussi ouvert l’accès RDP sur la machine pour permettre à l’attaquant de se connecter sur la machine et lui donner un ressenti d’élévation de privilèges, des outils tel que Hydra rend cela possible.

Une image contenant texte

Description générée automatiquement

C’est ce qu’on fait aussi plus tard quand on se connecte sur la machine Linux avec les éléments présents sur la machine Windows.

En donnant accès à la machine Windows cela nous permet aussi de capturer les actions de l’attaquant avec des outils comme sysmon ce qui rend notre analyse Splunk encore plus intéressant et riche en informations.

Tout cela donne l’illusion à l’attaquant que la machine auquel il a accès n’est pas un honeypot et permettra au mieux d’analyser son comportement.

# Conclusion

On conclut que ce projet fut une réussite. On avait un défi, on a su l’aborder déployer plusieurs honeypots sur une infrastructure réaliste et les analyser avec un SIEM qu’on avec lequel on avait des bases mais on se rend compte qu’on avait beaucoup à apprendre.

On devait choisir des honeypots intéressants mais aussi des honeypots qui étaient exploitable après par notre analyse Splunk, on devait aussi trouver une façon de mettre en évidence tout notre projet a travers une démo d’attaque sur l’infrastructure.

Celle-ci devait être fluide comportant des interactions avec les honeypot et les soulignant comme une des raisons que l’attaquant aurait choisi notre réseau comme le réseau a attaquer mais aussi avoir un fil connecteur dans notre cas d’autres services ouverts en parallèle pour donner à l’attaquant un ressenti de réussite, d’élévation de privilèges.

Afin qu’on puisse avoir le maximum de données possible pour notre analyse et comprendre au maximum les actions de l’attaquant.