# 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

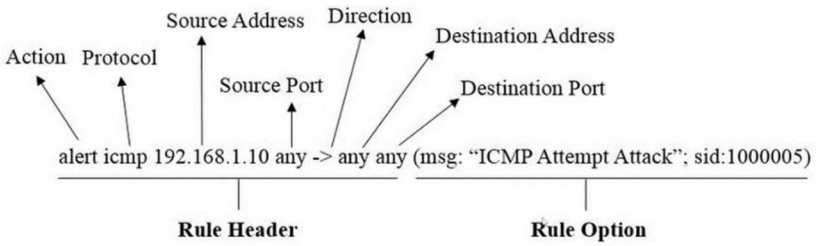
- DDos
- Backdoor
- Malware
- ARP Flooding
- Spoofing
- Attaque Web
- Attaques de BDD
- Et enfin on surveillera les tentatives d'accès à Facebook

## 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.

## 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;)
```

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 :

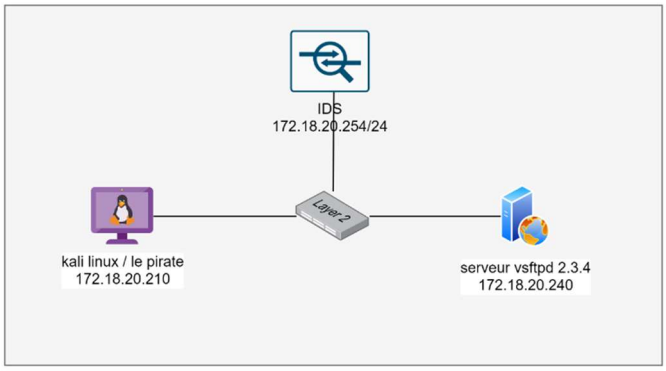
https://sourceforge.net/projects/metasploitable/files/Metasploitable2/

### *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

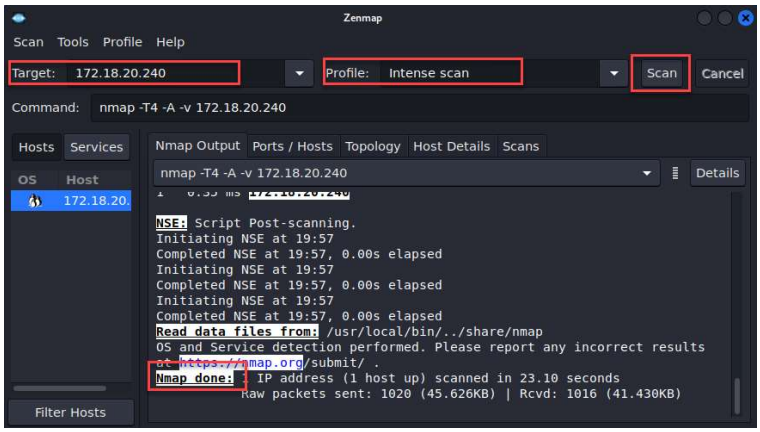
VSFTPD est un serveur FTP qui peut être trouvé dans les systèmes d'exploitation Unix comme Ubuntu. Par défaut, ce service est sécurisé, mais un incident majeur s'est produit en juillet 2011 lorsque quelqu'un a remplacé la version originale par une version contenant une porte dérobée. La porte dérobée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

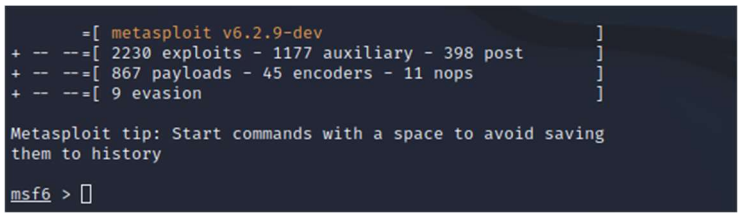
1. Allumez les trois machines et mettez-les dans le même sous-réseau. Nous installons les trois machines sur VirtualBox. L'IDS est installé sur le Pfsense qui a le rôle du pare-feu.
2. Lancez le *Zenmap* en cherchant son nom dans la menue application de Kali.
3. Entrez l'adresse IP de Metasploitable 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 :



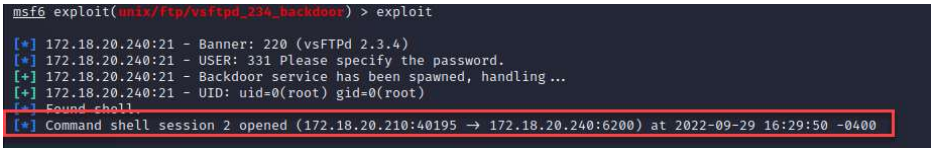

5. Cherchez le nom de l'exploit que l'on veut avec la commande *search*. Cela nous trouvera l'exploit vsftpd_234_backdoor qui correspond à notre attaque :


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

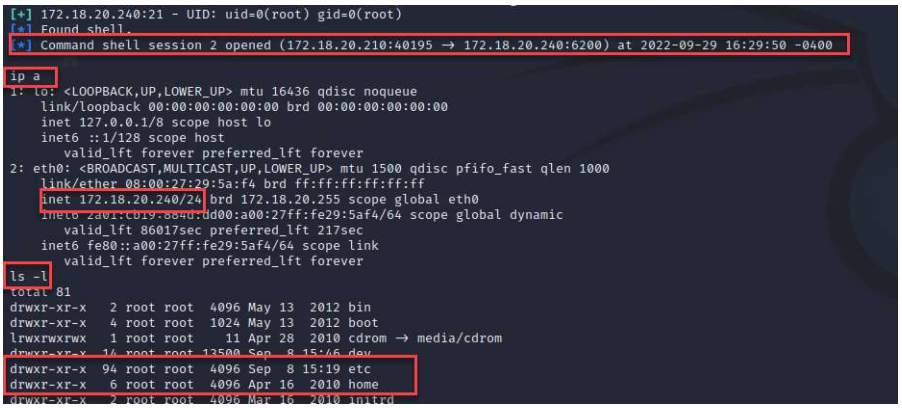

7. Une fois l'exploit sélectionné, vérifier les différentes options existantes et configurables avec la commande *show option* :

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 :

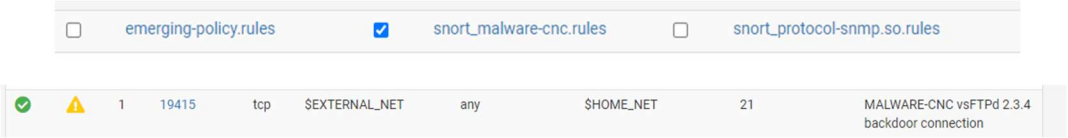



10. Vous pouvez vérifier que les différentes commande linux sont exécutables sur le serveur piraté ( *ls -l, ip a*, etc.). Donc un backdoor a été créé sur le serveur victime :

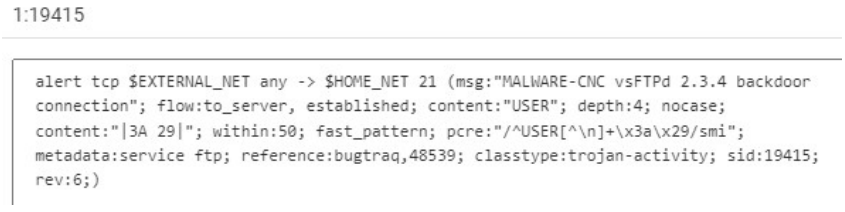



## Capturer l'attaque par Snort

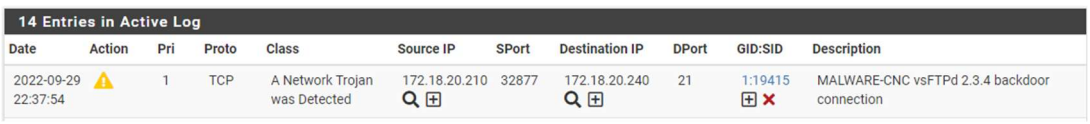
La règle par rapport à l'attaque vsftpd 2.3.4 est dans une catégorie qui s'appelle *snort_malware_cnc.rules*. Cette catégorie est déjà parmi nos catégories téléchargées et il nous suffit de la sélectionner dans l'onglet *Lan Categories* et activer la règle depuis l'onglet *Lan Rules* :



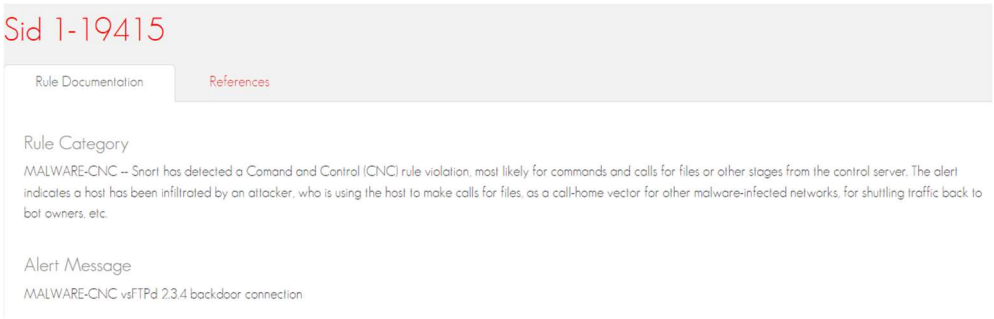

Voici la règle dans sa forme complète :




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

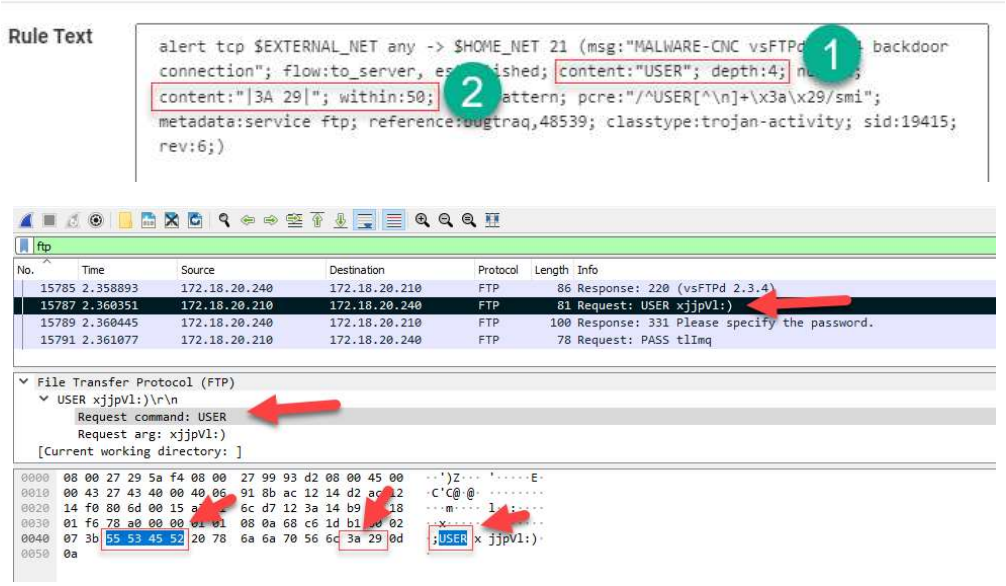



On peut vérifier les détails de cette règle en cliquant sur son numéro *SID (19415)* qui va nous diriger vers une page web sur le site *snort.org* :




## Analyse de trames par WireShark

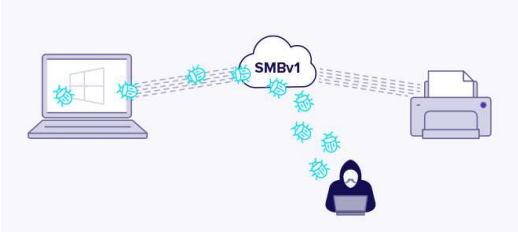
Si on essaye de traduire la règle ci-dessus, il nous dit qu'en premier, le contenu « USER » doit être trouvé dans les 4 premiers octets du paquet ftp et le contenu 3A 29 doit être trouvé parmi les 50 octets à partir du premier contenu trouvé. Et grâce à WireShark on voit bien que cela est le cas dans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

### *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

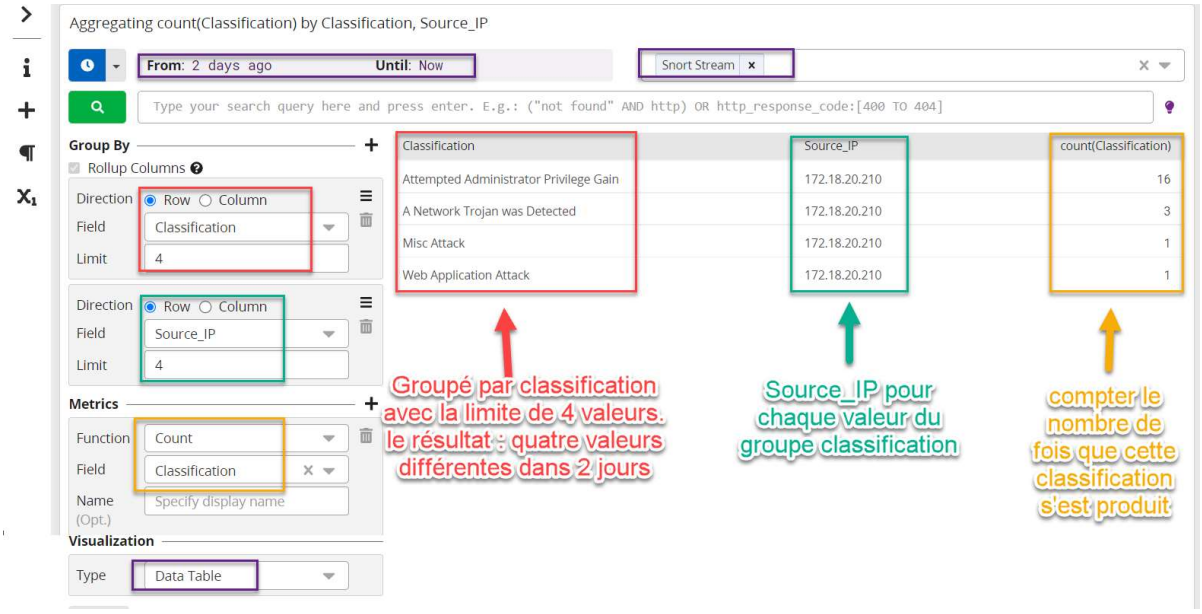
1. Accédez à la section Tableaux de bord en utilisant le lien dans la barre de menu de Graylog.
2. Appuyez sur le bouton Créer un nouveau tableau de bord pour créer un nouveau tableau de bord vide.
3. Appuyez sur le bouton Save as sur le côté droit de la barre de recherche pour enregistrer le tableau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

Maintenant on va ajouter des widgets dans notre Dashboard.

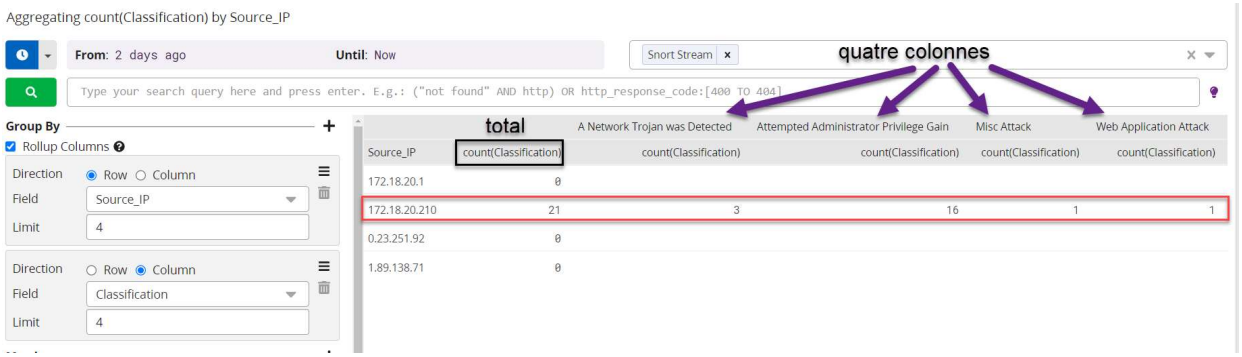
### Créer des widgets

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.

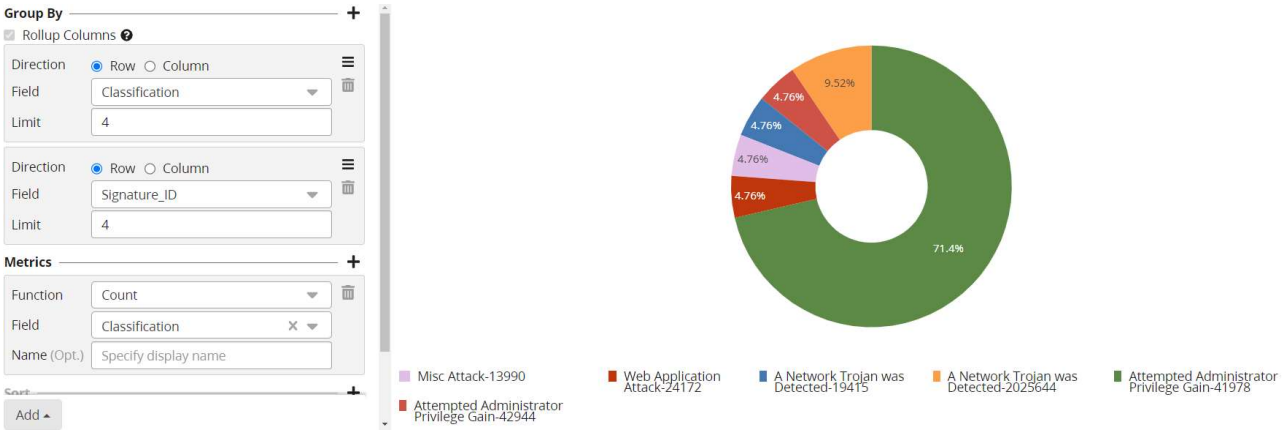
### Agrégation



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* :

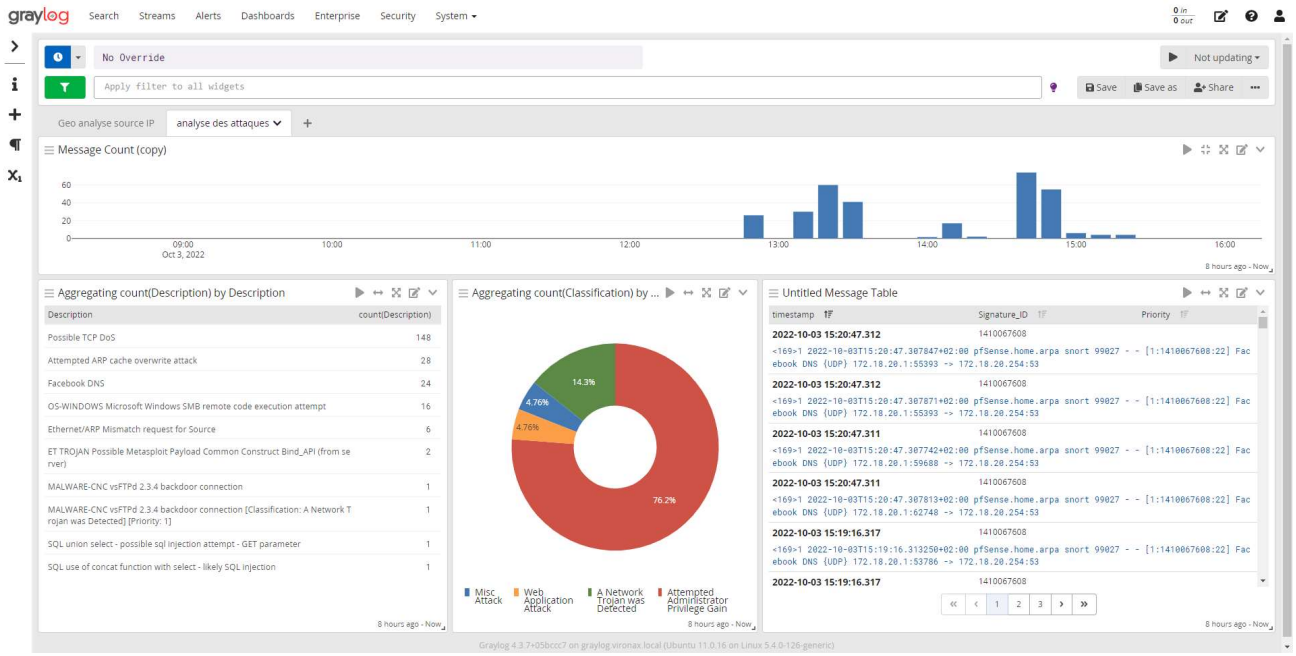



Suivez ce lien pour plus d'informations sur les widgets :
https://docs.graylog.org/docs/widgets

## Agrégation prédéfinie

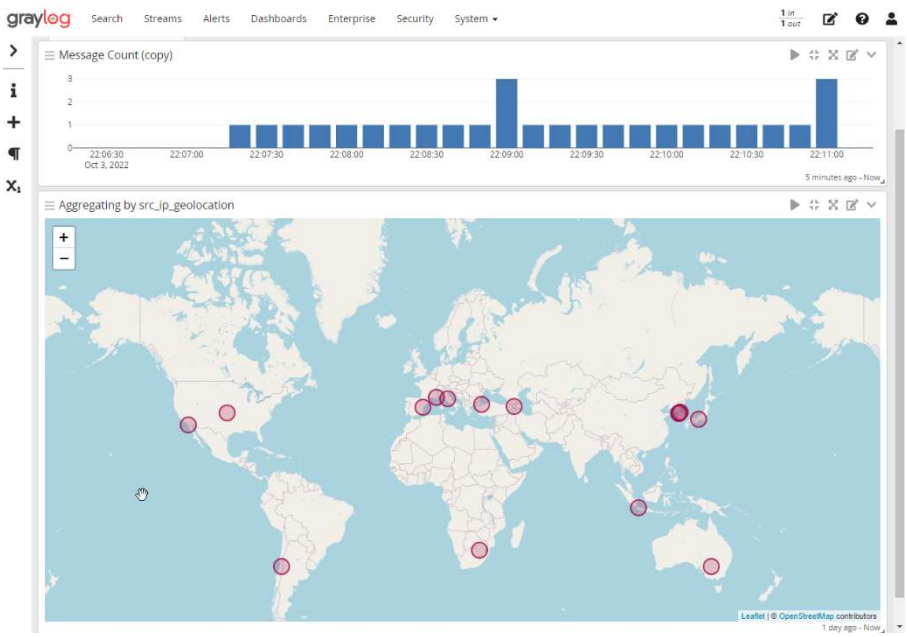
8. Le Message Table affiche les logs et leurs champs. Le Message Table peut être configuré pour afficher les champs et le log lui-même. Le log lui-même est rendu en police bleue sous les champs. Cliquer sur une ligne de message ouvre la vue détaillée d'un message avec tous ses champs.


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 :

**La version complète est disponible aussi mais protéger par un mot de passe.**

**Merci de me contacter par email :**

**Ershad.ra@gmail.com**