ADRAR FORMATION

Mise en place d'un NIPS avec SIEM

MARAVAL Liam
28/02/2025

Ce document sera décomposé en plusieurs chapitres :

1.Contexte : Définition des besoins ainsi que le choix des solutions en réponse à la demande client.
2. Configuration technique : Procédure technique de mise en place des solutions
3. Conclusion

Table des matières

1. Contexte

1.1 Demandes du client

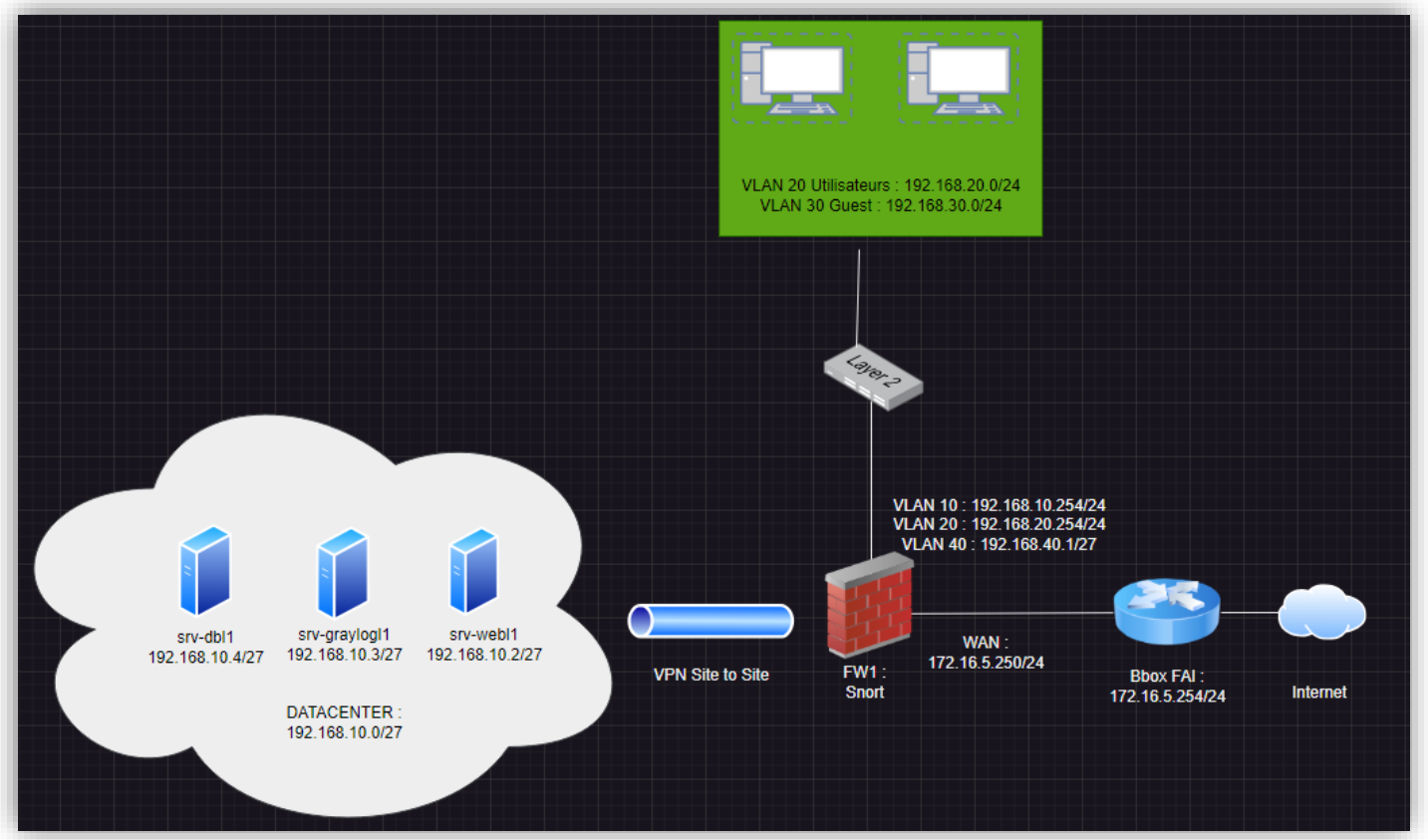
Nous sommes contactés par l'ADRAR afin de mettre en place une solution pour sécuriser le réseau de potentielles attaques type DDOS, Brut force FTP, ARP Spoofing ect..
De plus, ils nous est demandés de mettre en place un outil de centralisation des logs afin de faciliter la gestion des informations et événements lié à la sécurité.

Voici le cahier des charges défini avec l'ADRAR :

- Installation d'un service NIDS
- Configuration du service NIDS afin de protéger le réseau d'attaques Brut Force, DDoS ICMP
- Installation et configuration d'un outil de centralisation des logs (SIEM)
- Mise en place des règles communautaires de Snort pour prévenir des attaques les plus courantes

1.2 Evolution de l'infrastructure

Pour mieux comprendre la mise en place de la nouvelle infrastructure, veuillez-vous référer au nouveau schéma effectué :



Afin de répondre à la demande de l'ADRAR, nous allons donc implémenter **« Snort »** sur le Pfsense existant.
Ce services aura la charge de détecter le trafic suspect et de générer des alertes.
Nous configurerons élégamment Snort en mode NIPS afin qu'il puisse bloquer le trafic en plus de remonter les alertes, cependant cette pratique peux être inadapté car il pourrait bloquer du trafic légitime.
C'est pour cela que dès la fin du POC NIPS l'option sera désactivé, mais pourras être réactiver à l'avenir si l'ADRAR le souhaite en suivant la même procédure.

De plus, nous allons en place un serveur Graylog, celui-ci servira afin de récupérer les logs du service **Snort** ainsi que du pare feu Pfsense plus généralement afin de les centraliser dans sa console.
Voici ce qui va être mis en place :

- Intégration avec le pare-feu et avec Snort
- Configuration d'une politique de rétention des logs (Plus de détail dans la partie **1.4 Sécurisation des accès)**
- Configuration d'un stream afin de filtrer et afficher uniquement les logs du service Snort
- Configuration d'alertes afin de remonter les événements liés à la détection de trafic non légitime par Snort.

- Configuration d'envoi de mail lors du déclenchement de l'alerte.

Dans le cadre de notre POC, nous avons choisi de surveiller l'interface LAN à des fins des tests, cependant Snort est configurable sur n'importe quelle interface du pare-feu.

Une fois notre POC validé dans l'ensemble, nous mettrons en place les « **Community rules** » de Snort qui regroupe toutes les règles créées par la communauté afin de remonter les alertes des attaques les plus courantes (toutes les règles ne seront pas activées nous sélectionnerons uniquement celles voulues).

1.3 Choix de la solution

Nous avons opté pour **Snort** intégré à PFsense afin d'assurer le service NIPS, voici les raisons qui ont motivé notre choix :

- Intégration complète dans Pfsense
- Interface graphique intuitive
- Positionnement stratégique car permet des surveiller toutes les pattes de nos réseaux
- Leger et optimisé
- Beaucoup de règles communautaires disponibles gratuitement

En ce qui concerne Graylog, voici pourquoi nous l'avons choisi :

- Centralisation des logs en temps réel
- Interface web intuitive pour l'analyse et la visualisation
- Recherche avancée et filtrage puissant des logs
- Alertes et tableaux de bord personnalisables
- Compatibilité avec de nombreux formats de logs (Syslog, JSON, GELF…)
- Extensible grâce aux plugins et intégrations tierces
- Gestion efficace de grande quantité de données

Concernant les couts, nos 2 solutions sont open source.
Une version « **business** » est disponible pour **Snort** afin d'accéder à un support en priorité, un accès à des règles anticipé ect…

Une version « **entreprise** » de **Graylog** est disponible qui permet d'accéder à un support premium ainsi que des fonctionnalités supplémentaires comme des alertes via teams, discord…, une intégration avec O365, Azure, AWS, Palo Alto Networking, des modules avancés de gestion ect…

Une version « **sécurity** » est aussi disponible qui ajoute des fonctionnalités intéressantes. Voici les principales :

- Une détection des menaces avancées avec les réponses associées
- Détection d'anomalie via des événements inhabituels
- Enrichies les journaux d'événement avec une des informations complémentaires recueillis via une base de données des menaces connues
- Intégration du framework MITRE ATT&CK afin d'aider à identifier les techniques utilisées par des attaquants.

La version business de Snort ne présente pas de réels avantages à ce jour pour l'ADRAR.
Cependant, la version de Security de Graylog offre un vrai avantage à notre sens.

Voici les couts pour chaque version :

- Snort business : 399$/an et par capteur Snort
- Graylog Security : 1550$/mois

Concernant le temps de déploiement de la solution il est estimé à quatre jours.

1.4 Sécurisation des accès

Afin de sécuriser notre solution, nous allons mettre en place les configurations suivantes :

- Modification des ports par défaut de Snort et Graylog
- Compte d'administration de Graylog dédié
- Règle Snort afin de détecter de potentielle attaques DDOS ICMP Flood
- Règle Snort afin de détecter de potentielle attaque Brut Force du serveur FTP.
- Règles communautaires pour détecter les scans réseau via NMAP.

Conformément aux recommandations de la CNIL, une politique de rétention des logs sur le SIEM sera configurée en suivant ces règles :

- 30 jours de conservation des logs avant archive
- Archive conservée pendant 1 an avant suppression

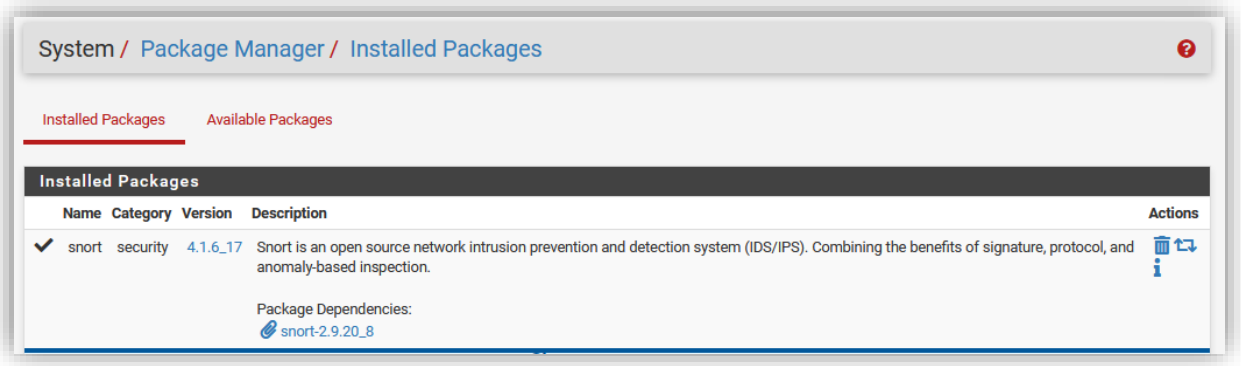
2. Configuration technique

Afin de faciliter la compréhension de cette procédure voici quelques termes à connaitre :

- **NIDS** : Système qui permet de détecter en temps réel de potentielles attaques réseau
- **NIPS** : Système qui permet de détecter et bloquer en temps réel de potentielles attaques réseau
- **SIEM** : Système qui permet de collecter, analyser et centraliser les données de sécurité pour détecter et répondre aux incidents
- **DDoS** : Une attaque qui consiste à inonder un serveur de requête (ICMP, HTTP, SYN ect…) depuis plusieurs sources afin de le rendre indisponible.
- **Brut Force** : Attaque qui consiste à essayer de deviner un mot de passe ou une clé en essayant un grand nombre de combinaisons possible automatiquement
- **Extractor** : Dans Graylog un extractor permet d'extraire et structurer des informations spécifiques à partir de logs bruts pour une meilleure analyse.

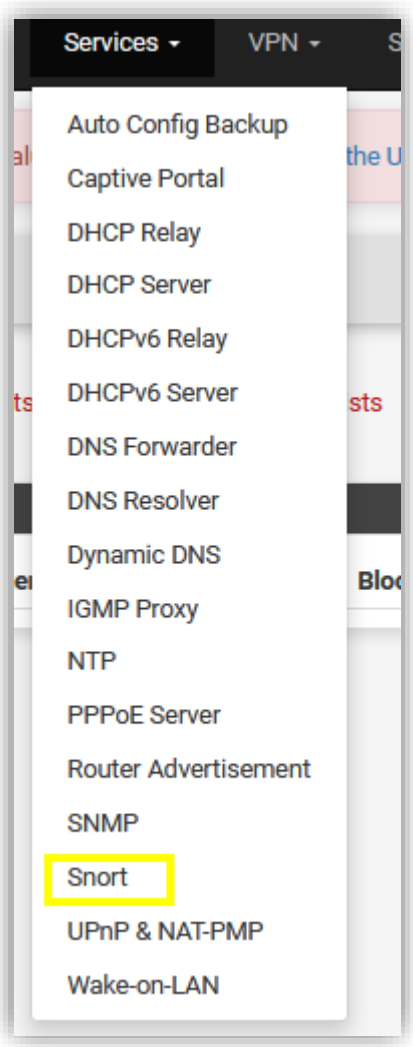
2.1 Installation et configuration de Snort

Pour l'installation du service Snort, il faut se rendre dans le package manager de Pfsense :

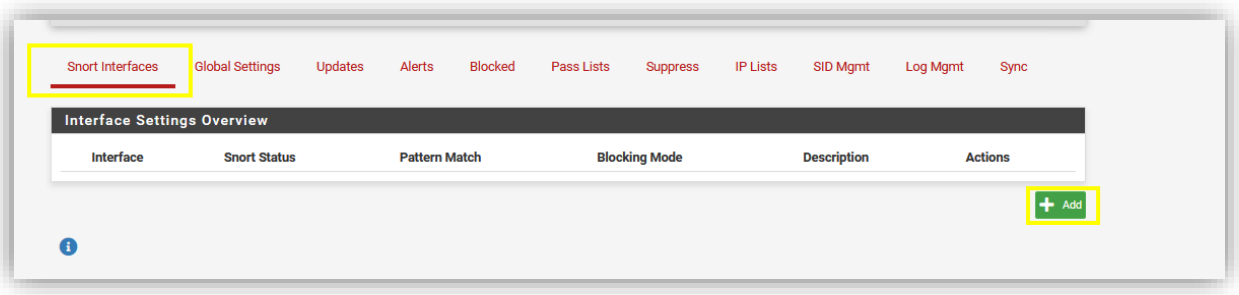


Une fois cela fait, nous allons commencer la configuration.

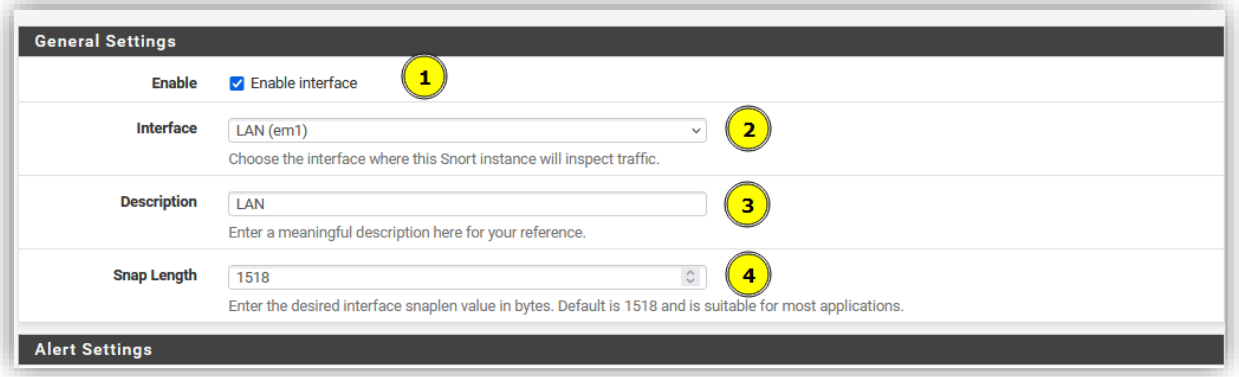
Nous allons dans le menu **« Services -> Snort »** :



Ensuite nous dans le menu **« Snort »** interface nous allons cliquer sur **« Add »** afin d'ajouter et configurer notre service sur l'interfaces de notre pare feu que nous souhaitons :

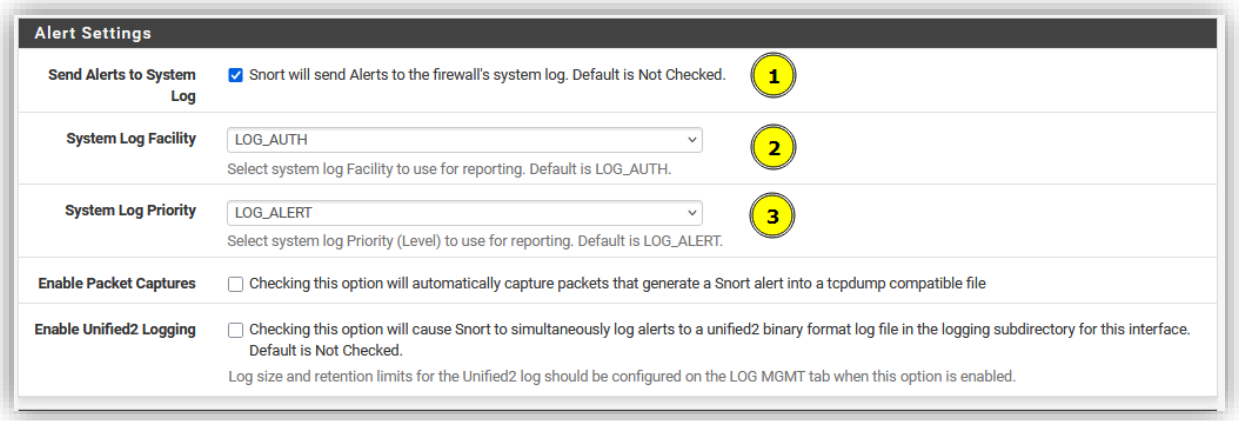


Dans **« General Settings »** nous configurons :



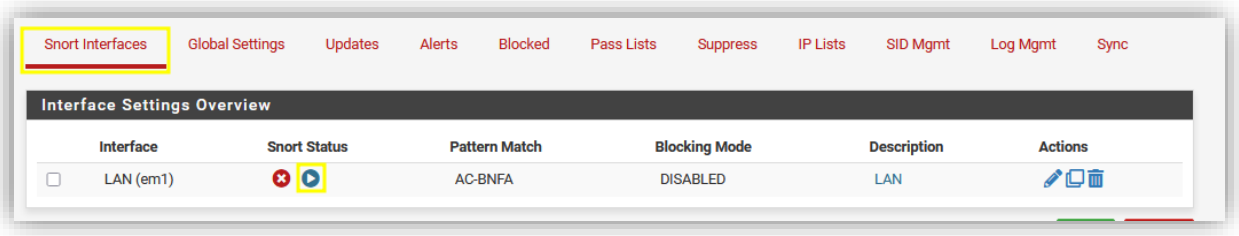
1. Active l'interface
2. Interface d'écoute
3. Description
4. Taille maximal des paquets analysés

Dans **« Alert Setting »** :

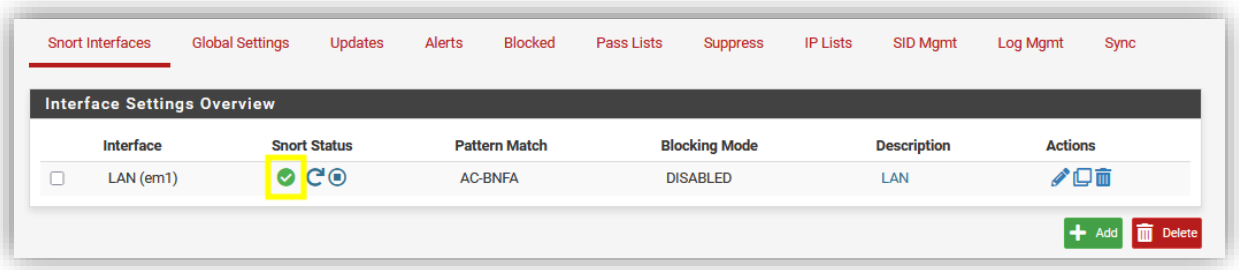


1. Envoyer les alertes dans les logs systèmes du pare feu
2. Envoie des logs de type sécurité et authentification
3. Envoie les logs de type alertes

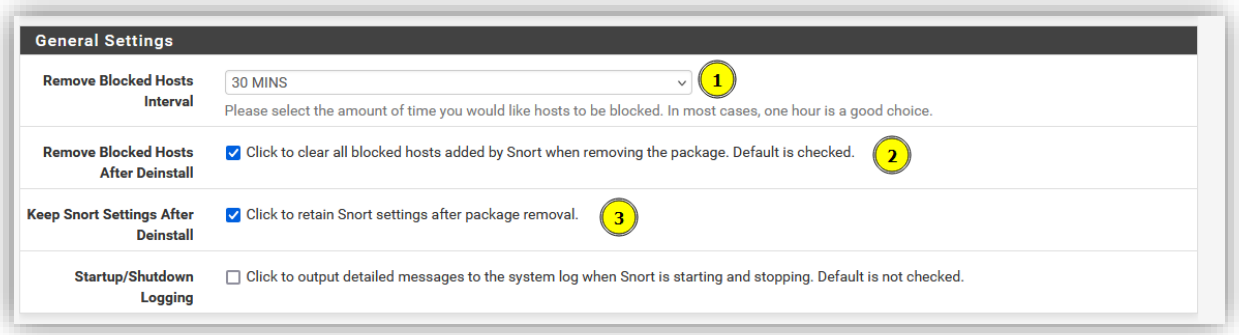
Nous sauvegardons ensuite la configuration et lançons le service :



Snort est désormais actif :



Nous allons maintenant dans le menu **« Snort -> Global Settings »** nous allons renseigner les paramètres suivants :

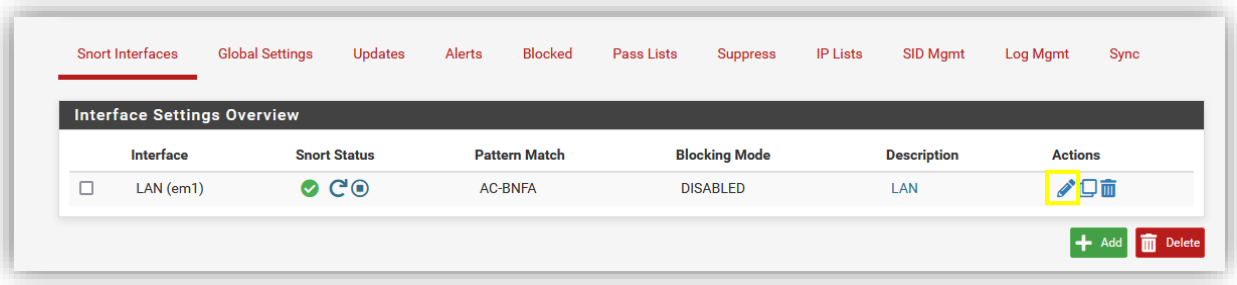


1. Vide la liste des hôtes bloqués toutes les 30mins (sera utile pour la partie NIPS que nous configurerons plus tard dans la procédure)
2. Vide la liste des hôtes bloqués après désinstallation du service
3. Conserve les paramétrages de Snort après désinstallation

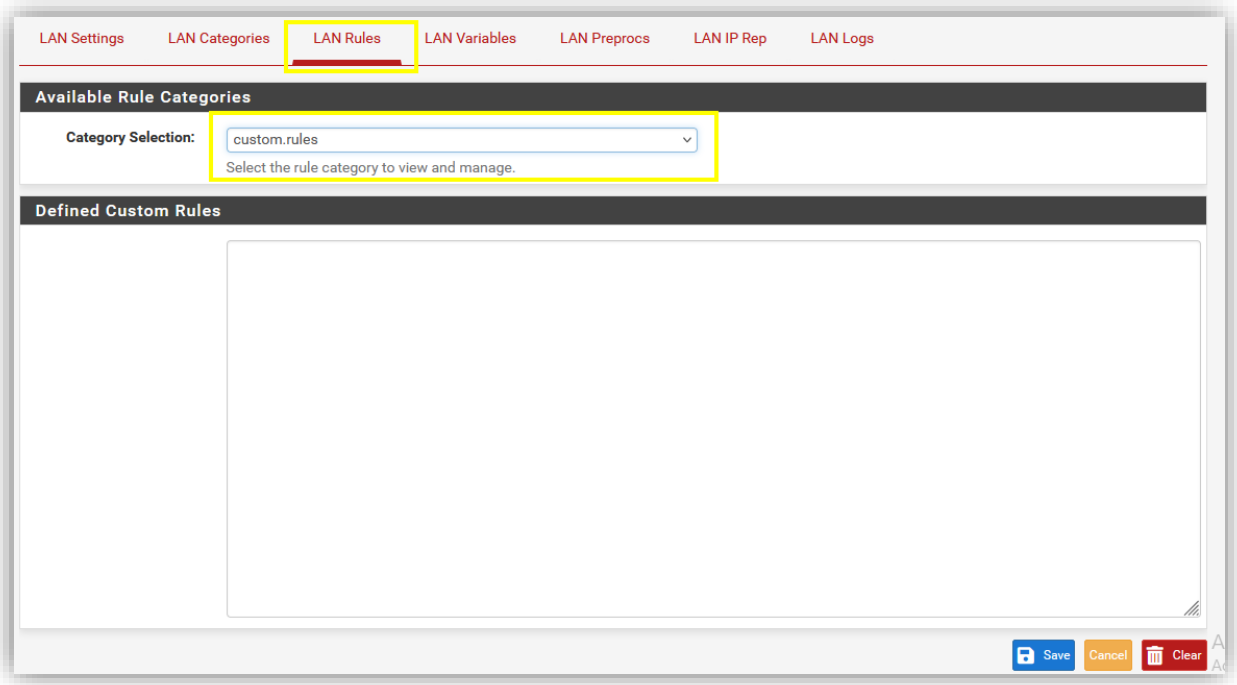
Maintenant que Snort est configuré nous allons pouvoir commencer la création de nos règles personnalisées.

2.2 Configuration des règles Snort DDoS ICMP

Pour configurer nos règles nous allons éditer notre interface créée précédemment :



Puis dans **« LAN Rules »** nous allons dans **« custom rules »** afin d'écrire nos règles personnalisées



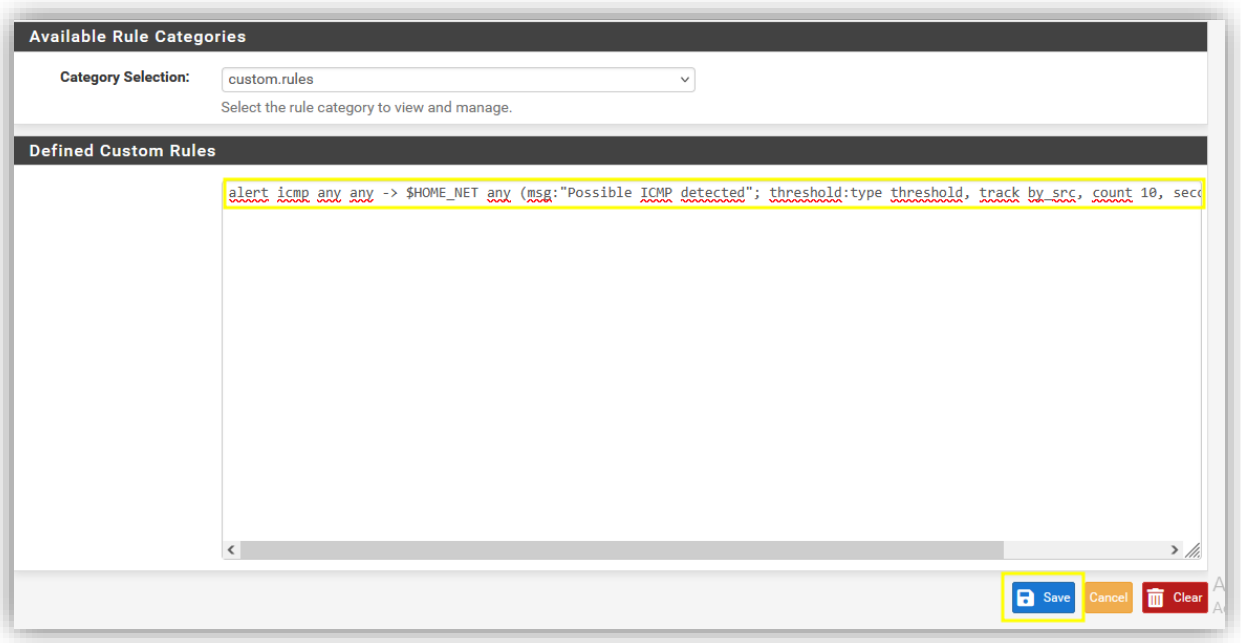
Voici la règle que nous avons mise en place :

```
alert icmp any any -> $HOME_NET any (msg:"Possible DDoS ICMP detected";
threshold:type threshold, track by_src, count 100, seconds 1; sid:1000002;)
```

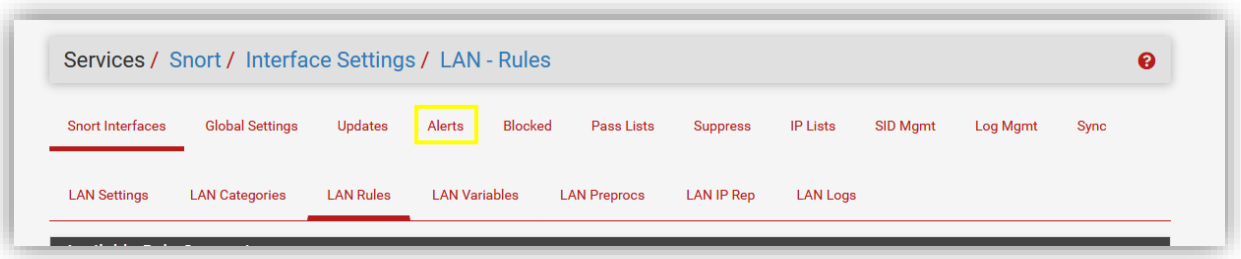
Nous allons l'analyser en détail :

- **« Alert »** : créé une alerte dès le match de la règle
- **« Icmp »** : définie le protocole ICMP
- **« Any any »**: IP et port sources
- **« -> $HOME_NET any »** : flux en direction de la variable $HOME_NET qui regroupe nos LAN vers n'importe quels ports
- **« Msg »** : Message à afficher
- **« threshold:type threshold, track by_src, count 100, seconds 1 »** : Définie le nombre de requêtes à 100 en l'intervalle de 1 second par source pour considérer le trafic comme non légitime et générer l'alerte
- **« sid:1000002 »** : Identifiant de la règle

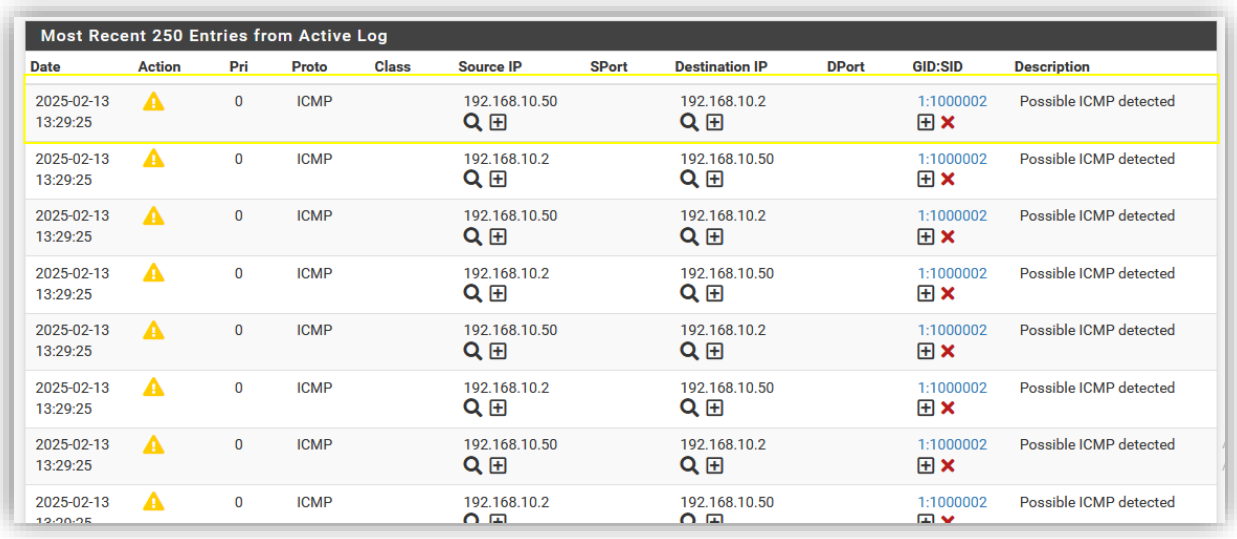
Une fois celle-ci rédigée, nous sauvegardons la configuration :



Maintenant nous allons tester notre règle via un Kali Linux en mettant en place une simulation de DDoS ICMP.
Une fois cela fait, nous allons nous rendre dans le menu **« Alerts »** pour voir les logs remonter :



Nous voyons donc plusieurs informations :



Most Recent 250 Entries from Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2025-02-13 13:29:25	⚠	0	ICMP		192.168.10.50		192.168.10.2		1:1000002	Possible ICMP detected
2025-02-13 13:29:25	⚠	0	ICMP		192.168.10.2		192.168.10.50		1:1000002	Possible ICMP detected
2025-02-13 13:29:25	⚠	0	ICMP		192.168.10.50		192.168.10.2		1:1000002	Possible ICMP detected
2025-02-13 13:29:25	⚠	0	ICMP		192.168.10.2		192.168.10.50		1:1000002	Possible ICMP detected
2025-02-13 13:29:25	⚠	0	ICMP		192.168.10.50		192.168.10.2		1:1000002	Possible ICMP detected
2025-02-13 13:29:25	⚠	0	ICMP		192.168.10.2		192.168.10.50		1:1000002	Possible ICMP detected
2025-02-13 13:29:25	⚠	0	ICMP		192.168.10.50		192.168.10.2		1:1000002	Possible ICMP detected
2025-02-13	⚠	0	ICMP		192.168.10.2		192.168.10.50		1:1000002	Possible ICMP detected

- La date et l’heure
- Le protocole
- L’IP source
- L’IP de destination
- Le SID de la règle
- Le message défini précédemment

Notre règle est donc fonctionnelle, nous allons pouvoir passer à la règle pour le Brut Force du FTP

2.3 Configuration règle Snort Brut Force FTP

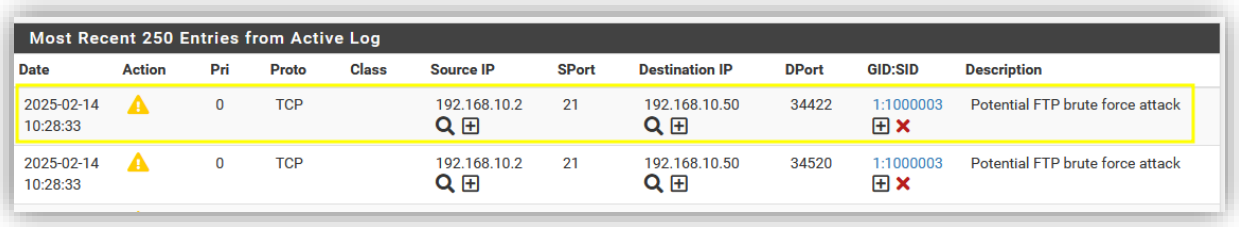
Dans la même logique que précédemment nous allons créer notre règle pour prévenir un éventuel Brut Force sur le serveur FTP.
Voici la règle que nous allons mettre en place :

```
alert tcp $HOME_NET 21 -> any any (msg:"Potential FTP brute force attack";
flow:established,from_server; content:"530 "; threshold:type threshold, track
by_src, count 5, seconds 60; sid:1000003;)
```

Le début de la règle suit la même logique que pour le DDoS ICMP cependant quelques éléments changent :

- **« flow:established,from_server; content:"530 " »** : Analyse les réponse du serveur FTP contenant le code 530 qui renvoi un message « login/password incorrect »
- **« threshold:type threshold, track by_src, count 5, seconds 60; »** : Définie le seuil à 5 tentatives de connexion échouées en 60 secondes par hôte pour générer une alerte

Afin de s’assurer du bon fonctionnement de notre règle nous allons simuler une attaque brut force par dictionnaire via notre Kali Linux et ensuite nous analysons les logs :

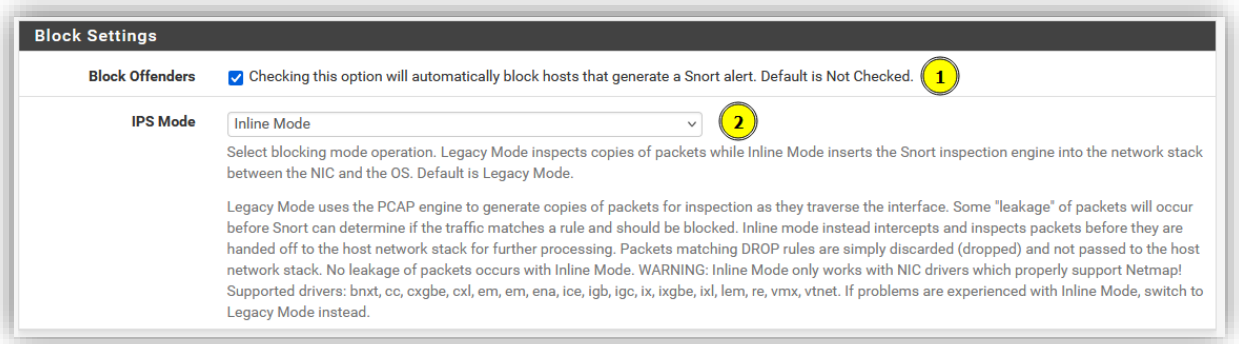


Ci-dessus nous retrouvons :

- Les IP source et destination
- Les ports sources et destination
- L'heure, le numéro de la règle et le message défini dans la règle.

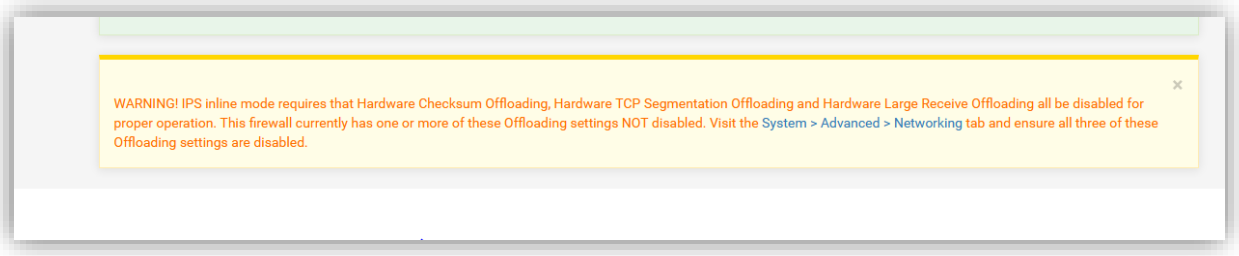
2.4 Configuration Snort en mode NIPS

Comme vu ci-dessus, il est possible de configurer Snort afin de détecter mais également bloquer le trafic suspect.
Nous allons donc nous rendre dans le menu **« Snort interfaces »** et nous éditons notre interface.
Dans le menu **« Global Settings »** nous allons appliquer les paramétrages suivants :

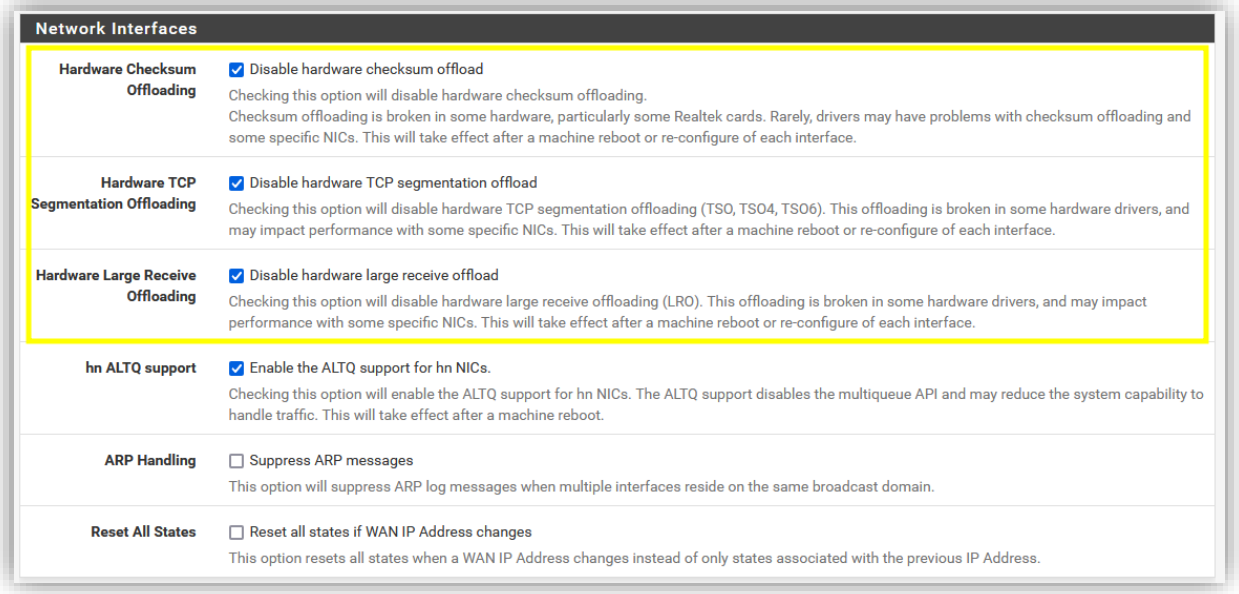


1. Cocher pour activer le mode NIPS
2. Mode d'inspection des paquets réseau en direct sans créer de copies de paquets. Contrairement au mode legacy qui créé des copies de paquets et qui peux permettre au trafic malveillant d'atteindre le réseau

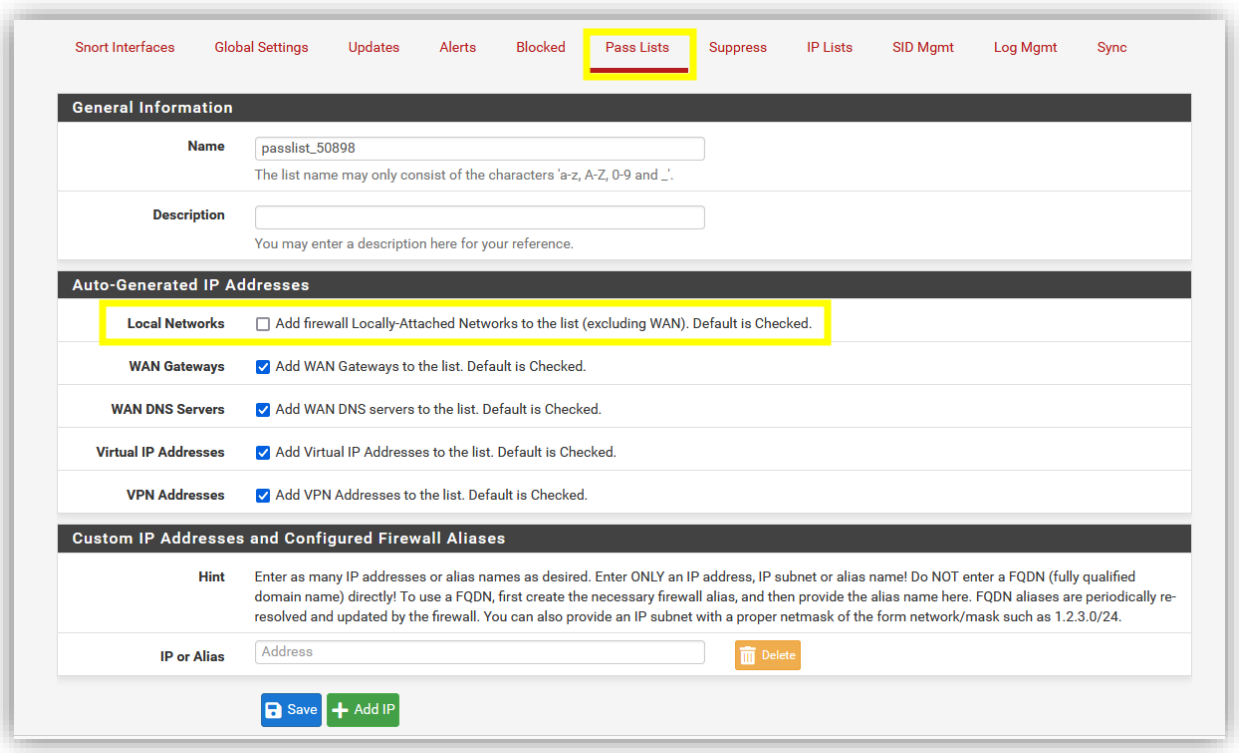
Attention : Quelques configurations sont nécessaires pour le fonctionnement en mode IPS dans notre LAB.
Dans un 1er temps nous avons ce message qui apparait :



Nous allons donc désactiver les options suivantes dans le menu **« System -> Advanced -> Networking » :**

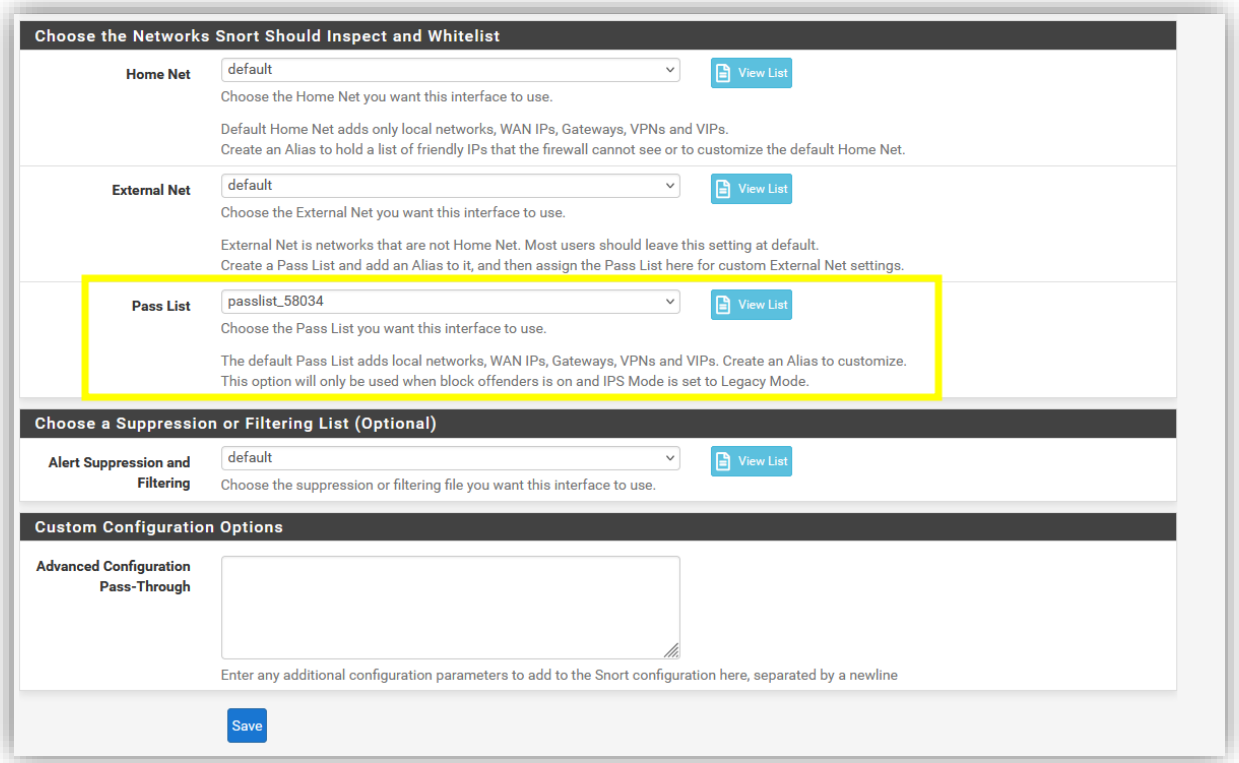


Attention : Comme nous allons tester notre NIPS depuis notre LAN avec une machine Kali Linux, il est nécessaire de créer une passlist personnalisée car celle par défaut whitelist les hôtes de nos LAN, donc notre machine générant du trafic malveillant ne seras jamais bloqué car elle en fait partie.
Pour cela il faut se rendre dans le menu **« Snort -> PassList »** et cliquer sur **« Add »** :

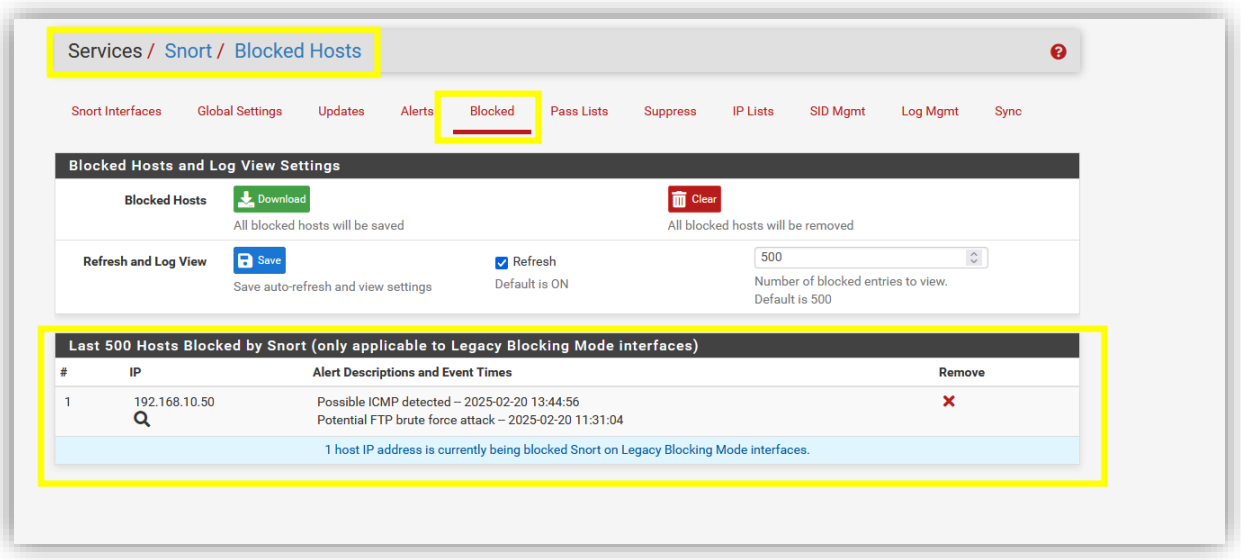


Ici nous décochons donc **« Local Networks »** pour enlever le LAN de la whitelist.

Enfin dans le menu **« Snort Interfaces -> LAN Settings »** nous sélectionnons notre nouvelle Pass List :



Maintenant nous pouvons retenter notre attaque, puis dans le menu **« Snort -> Blocked »** nous retrouvons notre hôte bloqué :

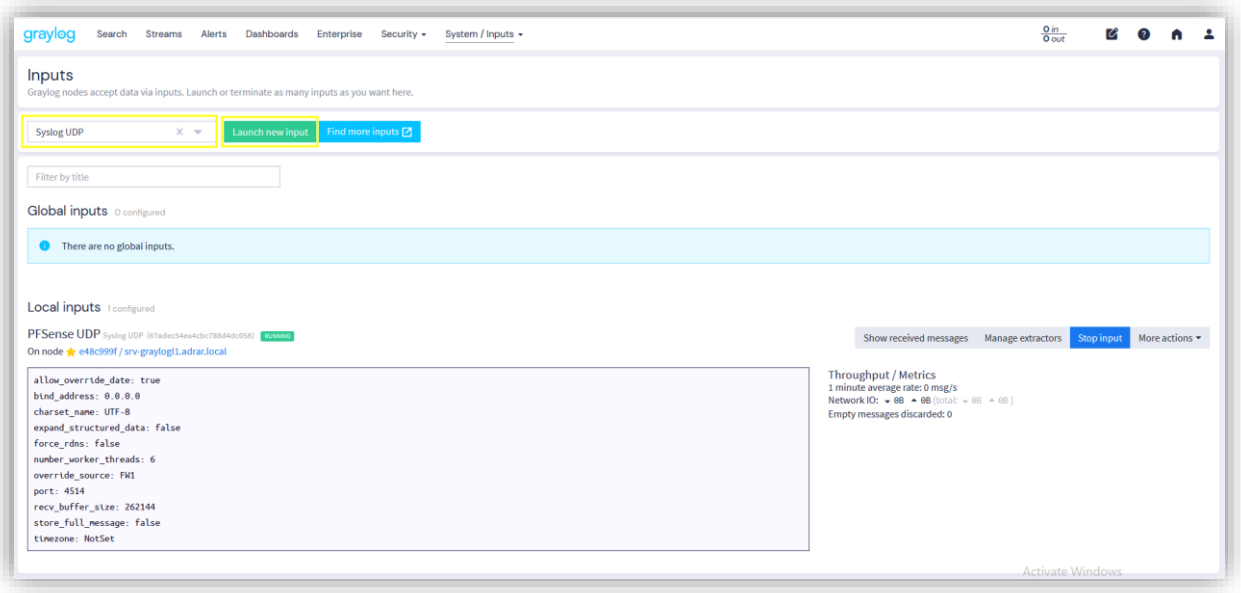


Pour rappel, ce mode peut potentiellement bloquer du trafic légitime c'est pour cela que nous avons décidé de désactiver ce mode.

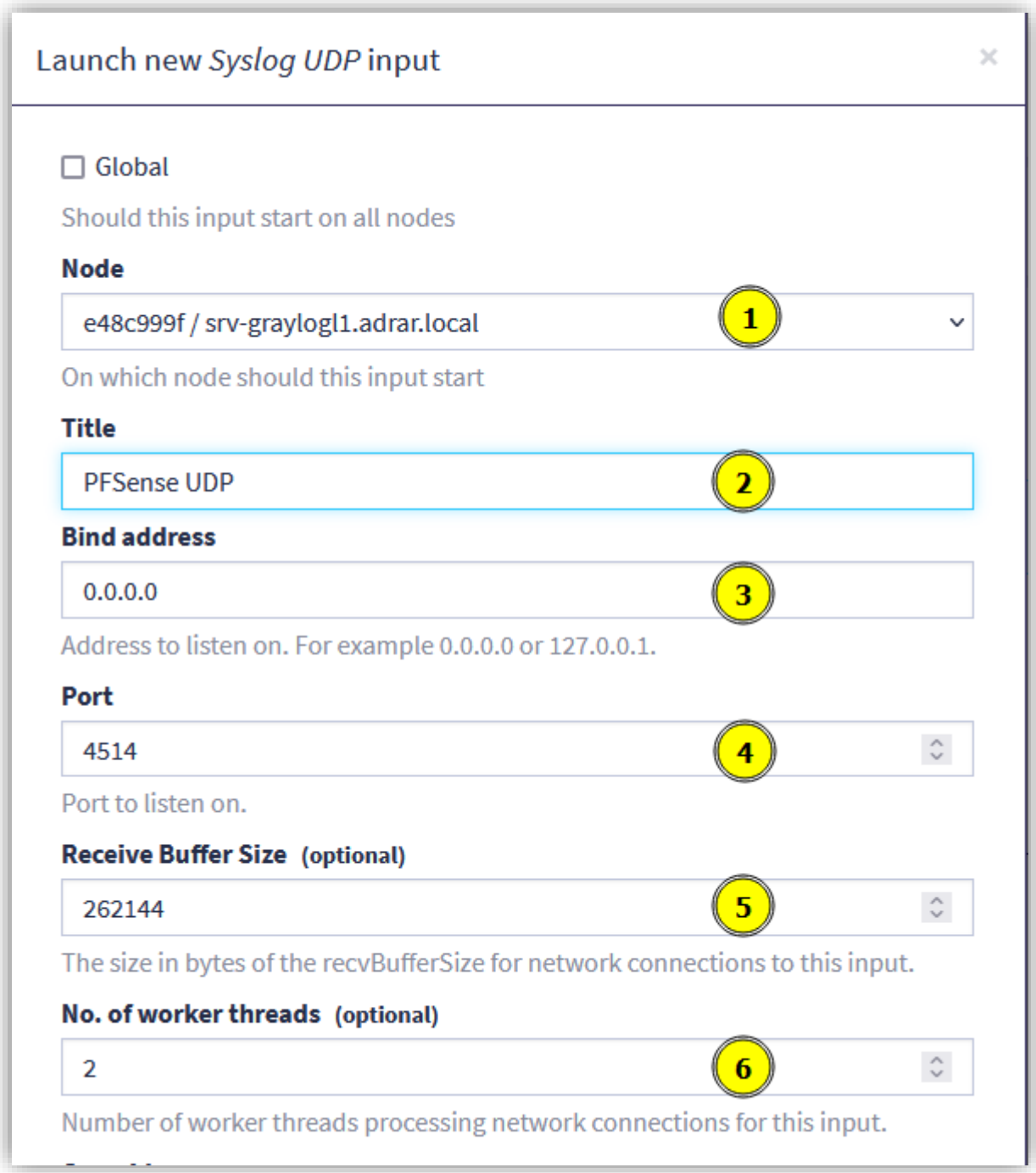
2.5 Intégration avec Graylog

Nous allons désormais intégrer nos logs dans notre solution SIEM qui est « **Graylog** ».
L'installation est considéré comme faite, la documentation officielle d'installation est disponible dans la partie Annexes de ce document.

Nous allons nous rendre sur notre serveur afin de configurer une entrée **« input »** pour que Gaylog puisse recevoir les logs venant du Pfsense.
Pour cela nous allons dans le menu **« System / Inputs -> Inputs »** et sélectionnons **« Syslog UDP »** et **« Launch New Inputs » :**

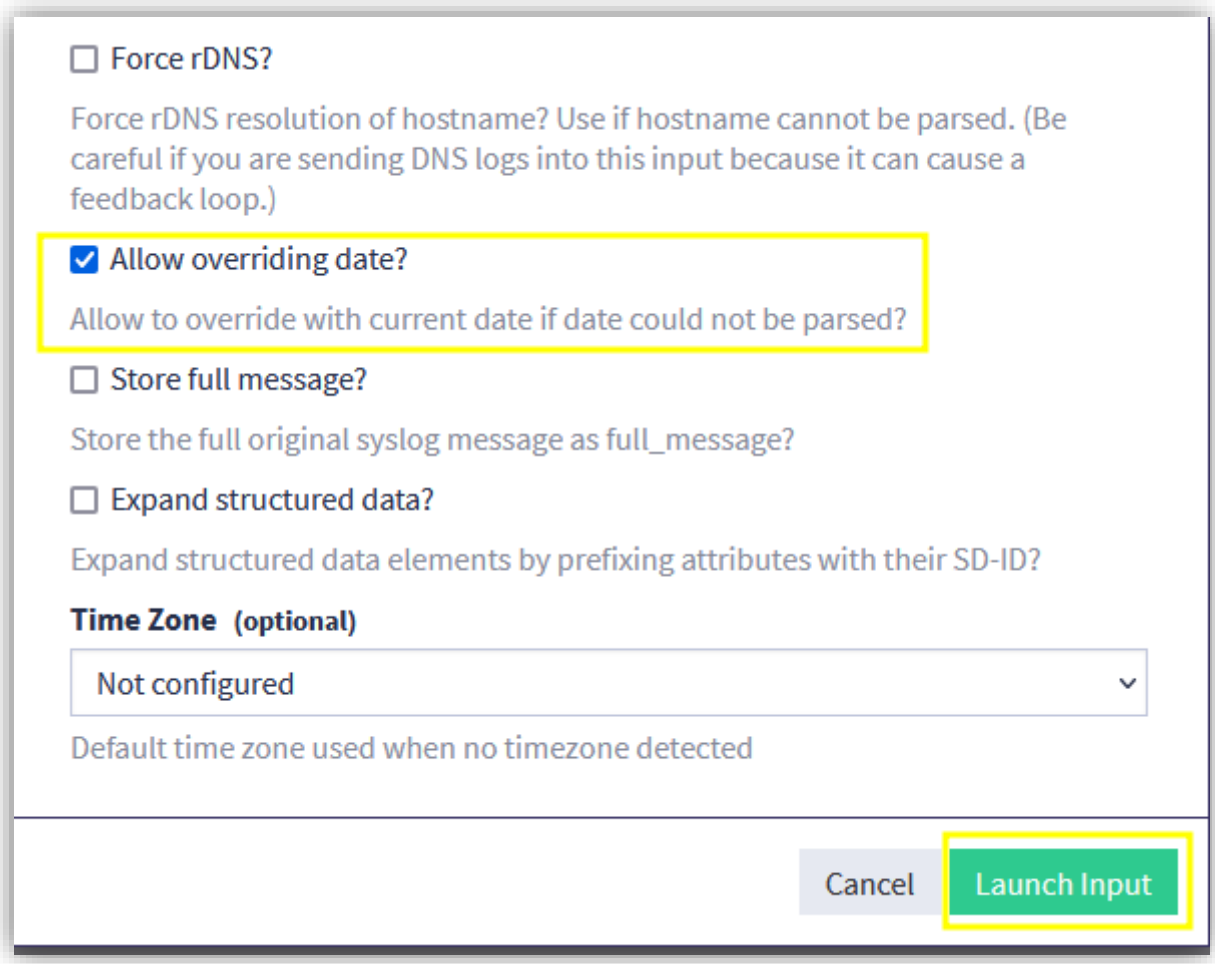


Ensuite nous allons renseigner les paramètres suivants :



1. Nom du serveur Graylog
2. Nom du connecteur
3. Adresse d'écoute
4. Port d'écoute
5. Taille du buffer de réceptions des logs
6. Nombre de worker pour traiter les logs entrants

Ensuite nous cochons **« Allow overriding date »** afin de pouvoir réécrire la date si nécessaire et nous sauvegardons notre input :

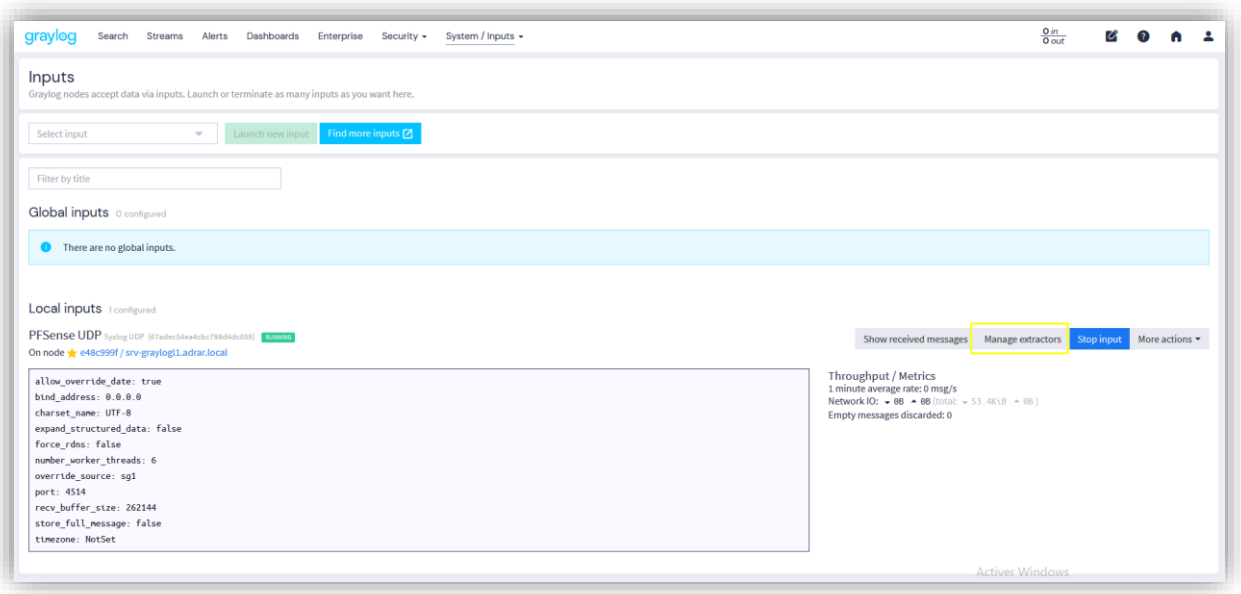


Maintenant nous allons adapter un fichier. json que nous avons trouvé sur Github (lien en annexe) qui va servir à extraire et transformer les logs envoyées par PFsense.
Celui-ci étant pour Suricata nous allons l'adapter pour Snort

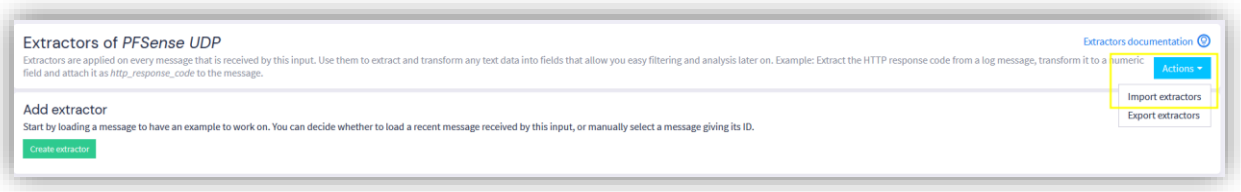
```
{
    "extractors": [
      {
        "title": "Extract Snort Application Name",
        "extractor_type": "regex",
        "converters": [
          {
            "type": "lowercase",
            "config": {}
          }
        ],
        "order": 0,
        "cursor_strategy": "copy",
        "source_field": "message",
        "target_field": "application_name",
        "extractor_config": {
```

```
          "regex_value": "\\[[0-9]:[0-9]*:[0-9]\\] (SNORT) .*"
        },
        "condition_type": "regex",
        "condition_value": "\\[[0-9]:[0-9]*:[0-9]\\] (SNORT) .*"
      },
      {
        "title": "General Application Name Extractor",
        "extractor_type": "regex",
        "converters": [],
        "order": 0,
        "cursor_strategy": "copy",
        "source_field": "message",
        "target_field": "application_name",
        "extractor_config": {
          "regex_value": "(^[a-z-]+).*"
        },
        "condition_type": "regex",
        "condition_value": "(^[a-z-]+).*"
      }
    ],
    "version": "3.2.5"
}
```

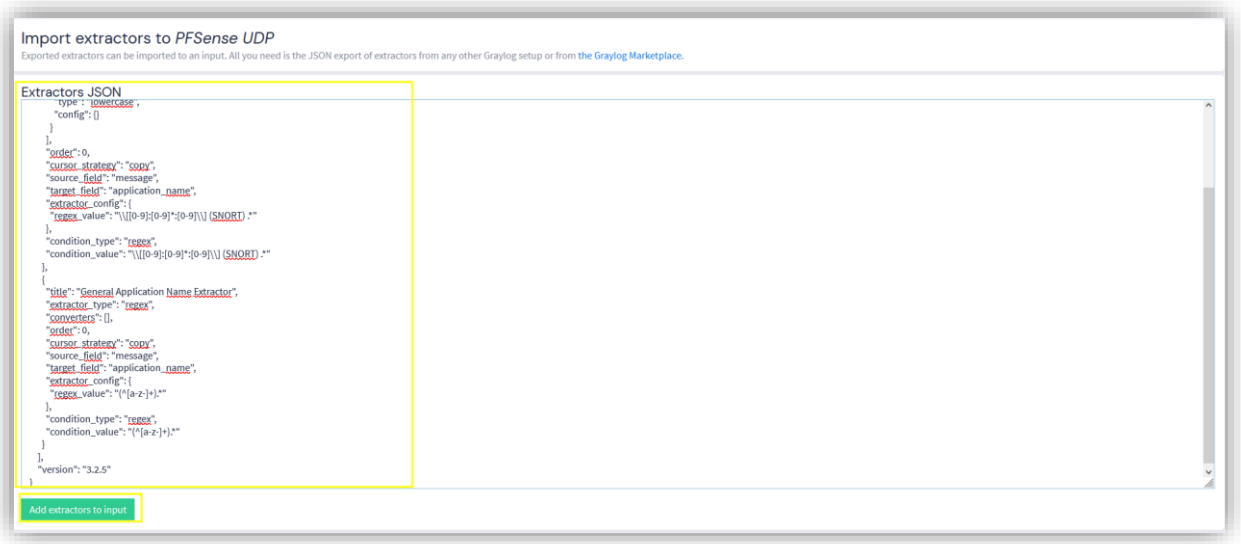

Dans le menu **« System / Inputs -> Inputs »** de Graylog nous allons donc cliquer sur **« Manage Extractor »** :



Puis **« Actions -> Import extractor »**

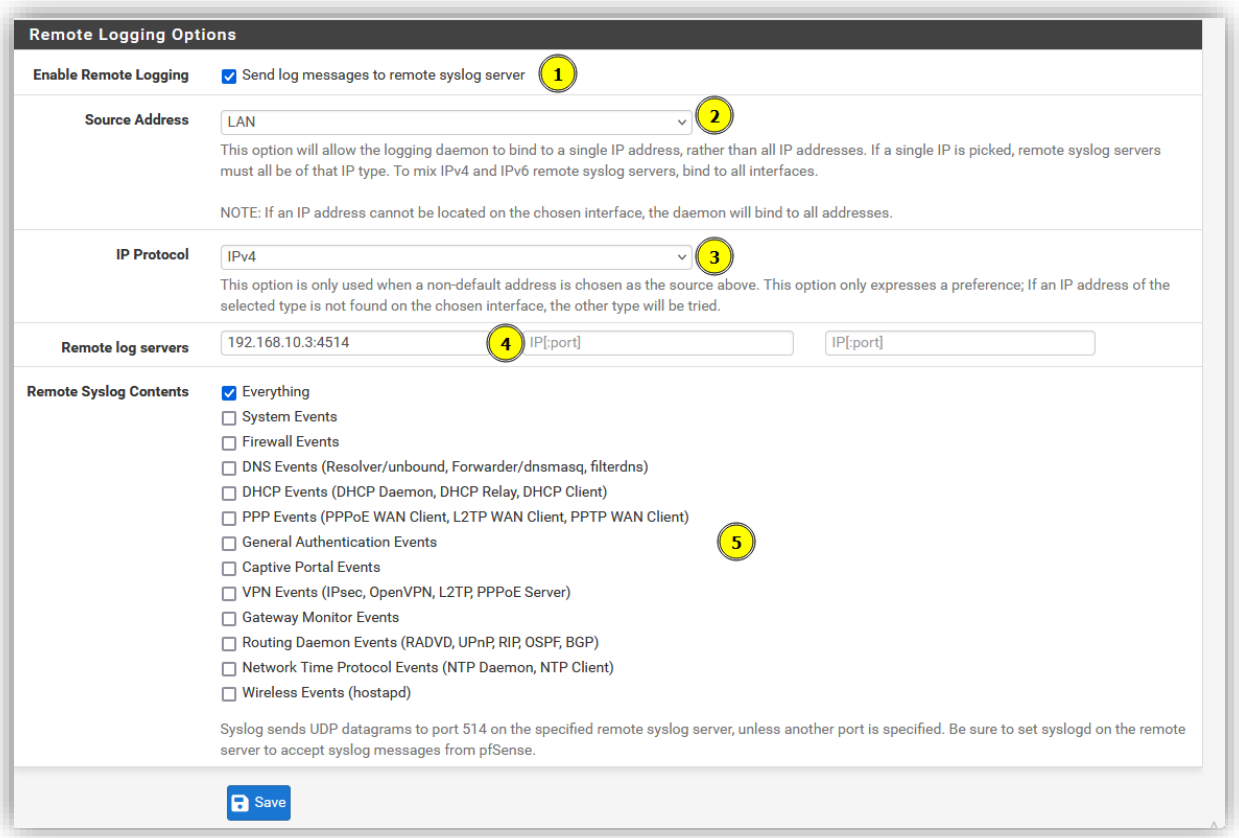


Ensuite nous collons notre code et ajoutons l'extractor à l'input :



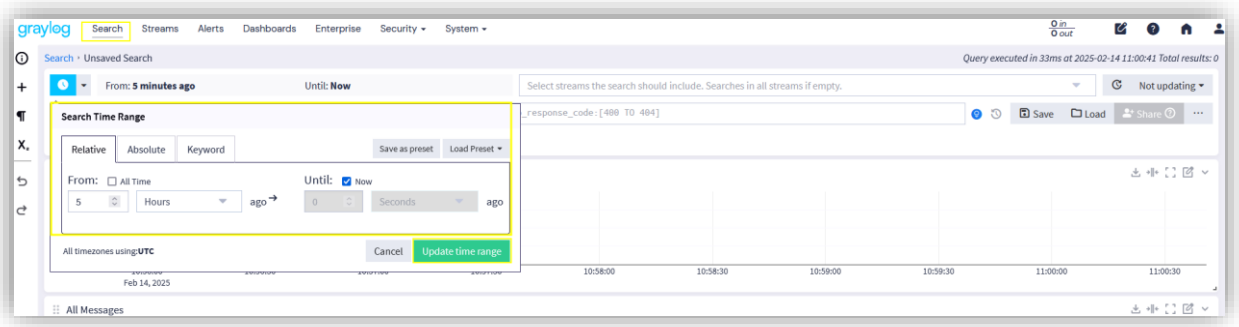
Maintenant une configuration est nécessaire sur Pfsense.
Nous allons dans le menu **« Status -> System Log -> Setting »** dans le bas de la page nous

trouvons les configurations pour envoyer les logs à un hôte distant :

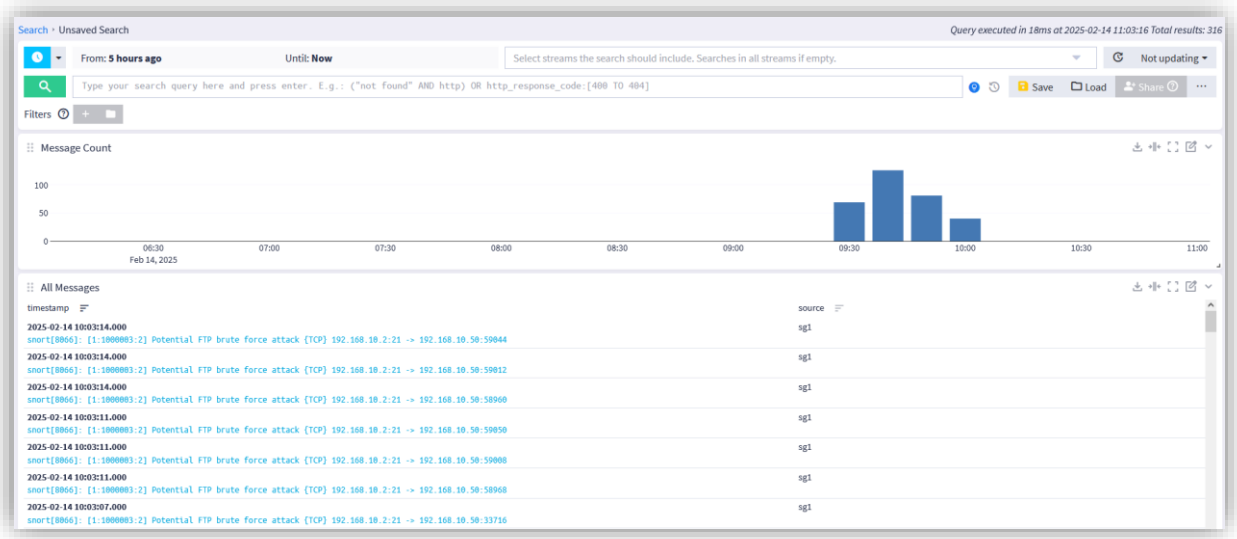


1. Active les logs distants
2. Interface utilise pour envoyer les logs
3. Protocole IPv4
4. IP et port utilisé du serveur Graylog
5. Informations envoyées au serveur Graylog (que nous pourrons affiner via ce menu plus tard ou restreindre via la création de Dashboard personnalisé dans Graylog)

Une fois cela fait dans le menu « **Search** » nous pouvons effectuer notre recherche de log en indiquant les filtres voulus :



Une fois la recherche lancée nous retrouvons les logs générés par notre Pfsense :

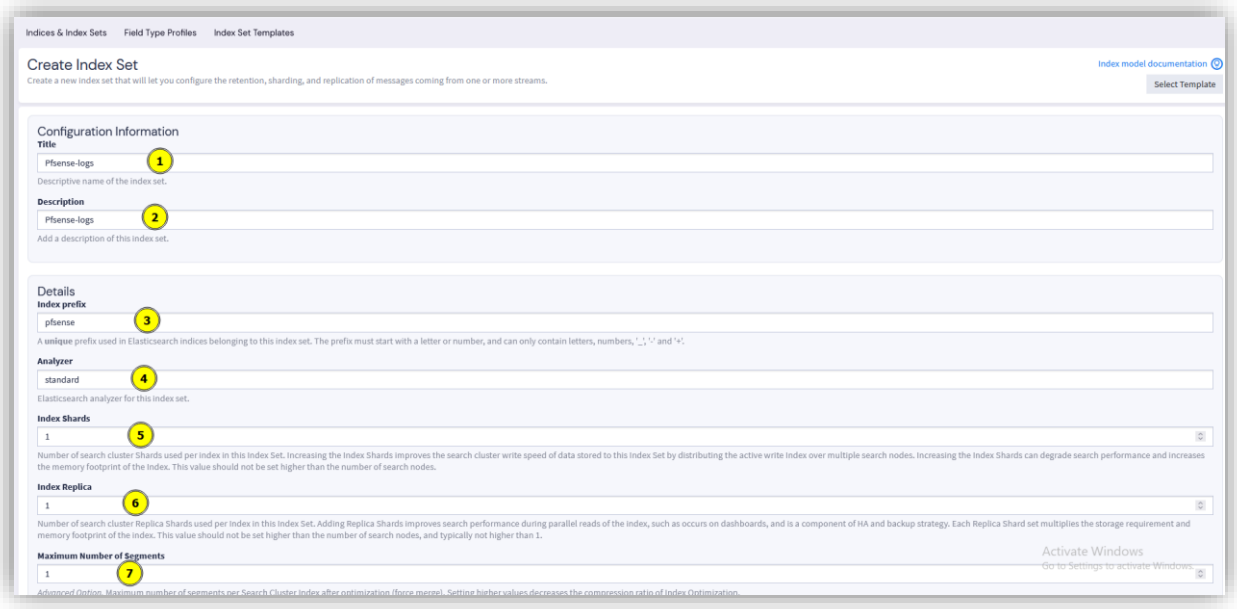


Puis en cliquant sur une ligne, dans notre cas la tentative de brut force, nous avons plus de détail :

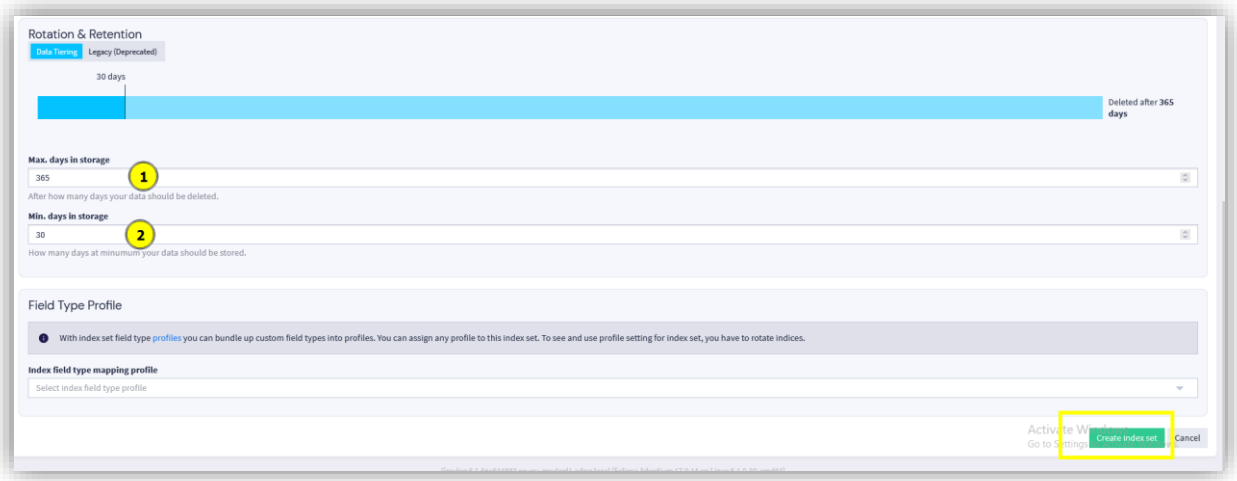
1. Heure de l'alerte
2. Input utilisé pour la réception des logs
3. Service qui envoi l'alerte
4. Message de l'application
5. Source de l'envoi des logs (notre pare feu PFsense)

Beaucoup d'options sont possibles pour les recherches par exemples des filtrages par noms d'événements, nom de service, SID d'une règle, IP sources ou destination ect…

Nous allons maintenant configurer une politique de rétention des logs.
Pour cela, nous allons dans le menu « **System -> Indices** » puis « **create index set** » :

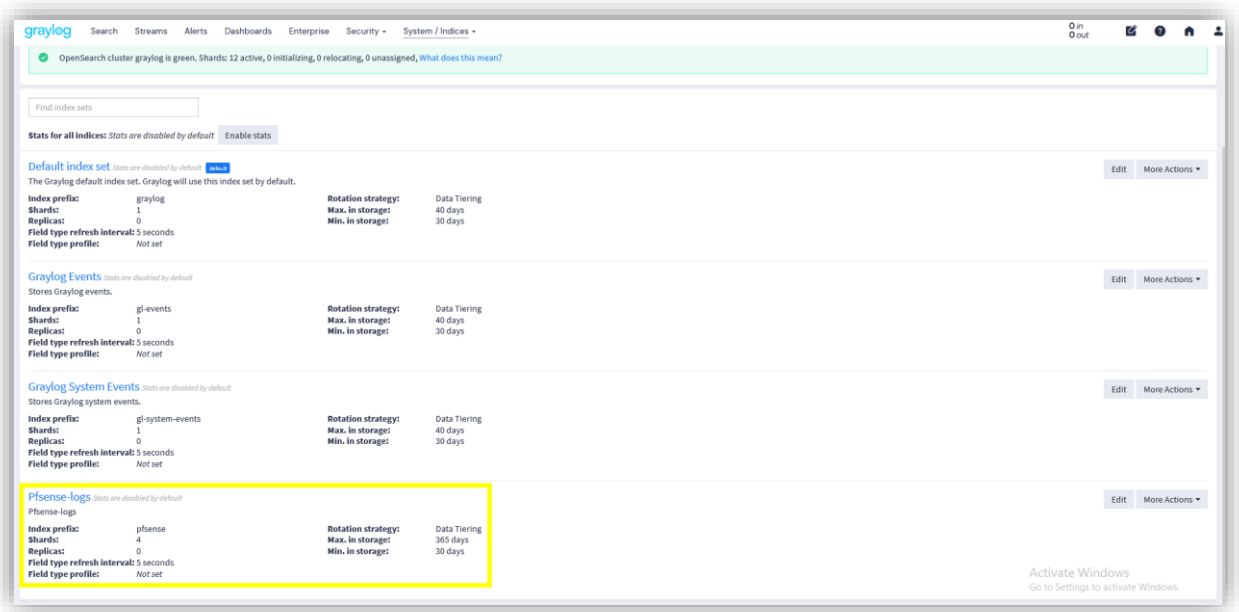


1. Titre de l'indice
2. Description
3. Préfix utilisé pour les logs
4. Nombre de fragment (Shards) défini pour diviser l'index (Dépend du nombre de nœud Graylog, ne pas mettre une valeur trop élevée au risque réduire les performances)
5. Nombre de copie de fragment (Shards) à effectuer
6. Copies des données effectuées pour en éviter la perte et accélérer la lecture.
7. Compression des index pour optimiser l'espace et les performances.

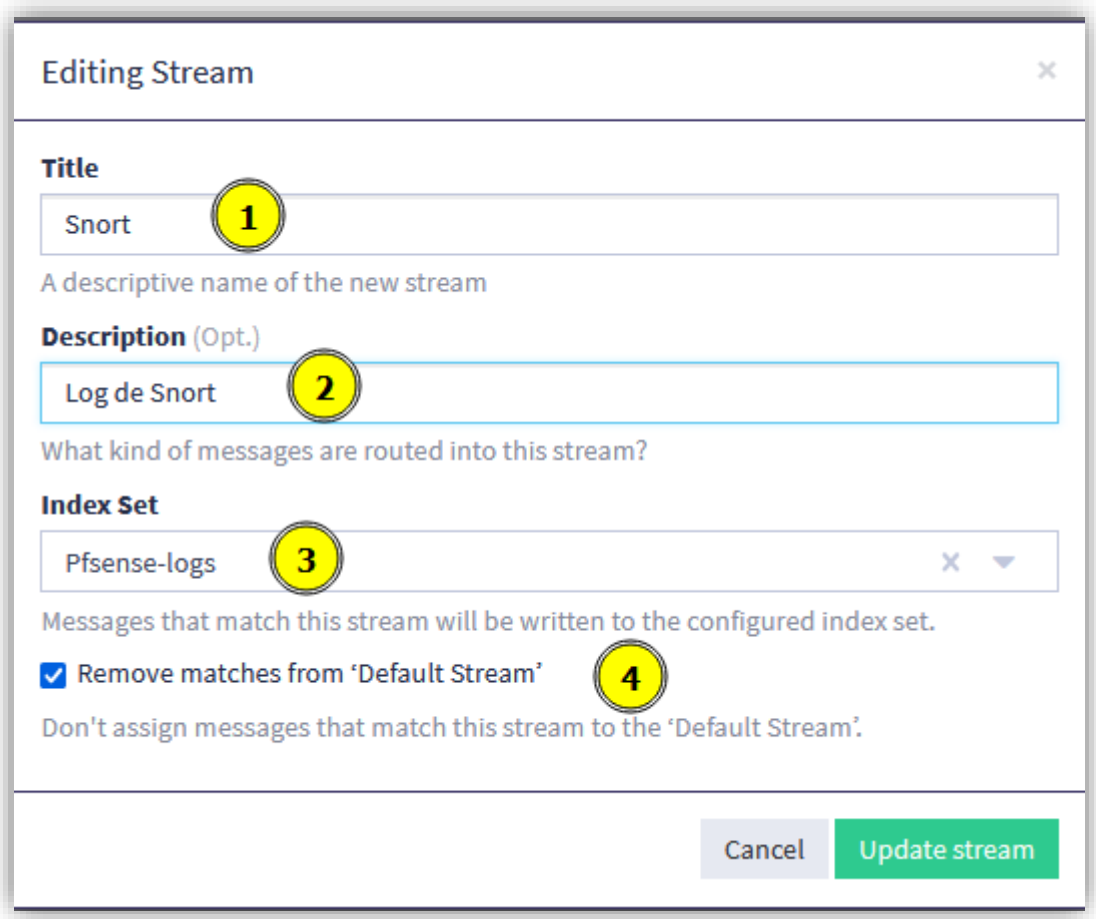


1. Durée de conservation des archives
2. Durée de conservation des logs avant archives

Nous retrouvons maintenant notre indice créé avec les paramètres définis :

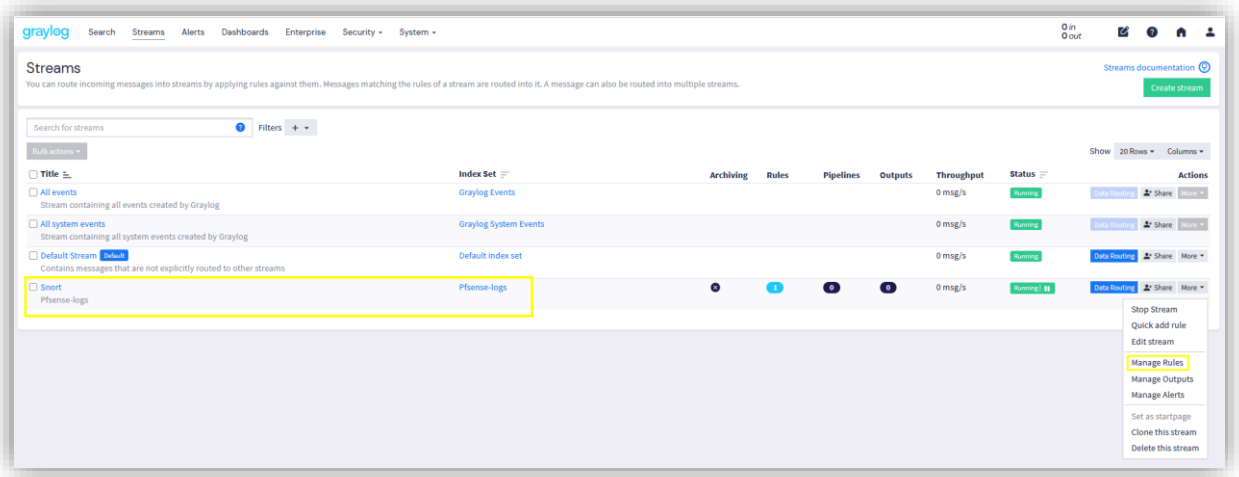


Nous pouvons maintenant affiner nos recherches en créant un stream qui permettra de filtrer les logs et afficher uniquement celle concernant Snort.
Pour cela nous allons dans le menu « **Streams** » puis « **create streams** » et renseignons les paramètres suivants :

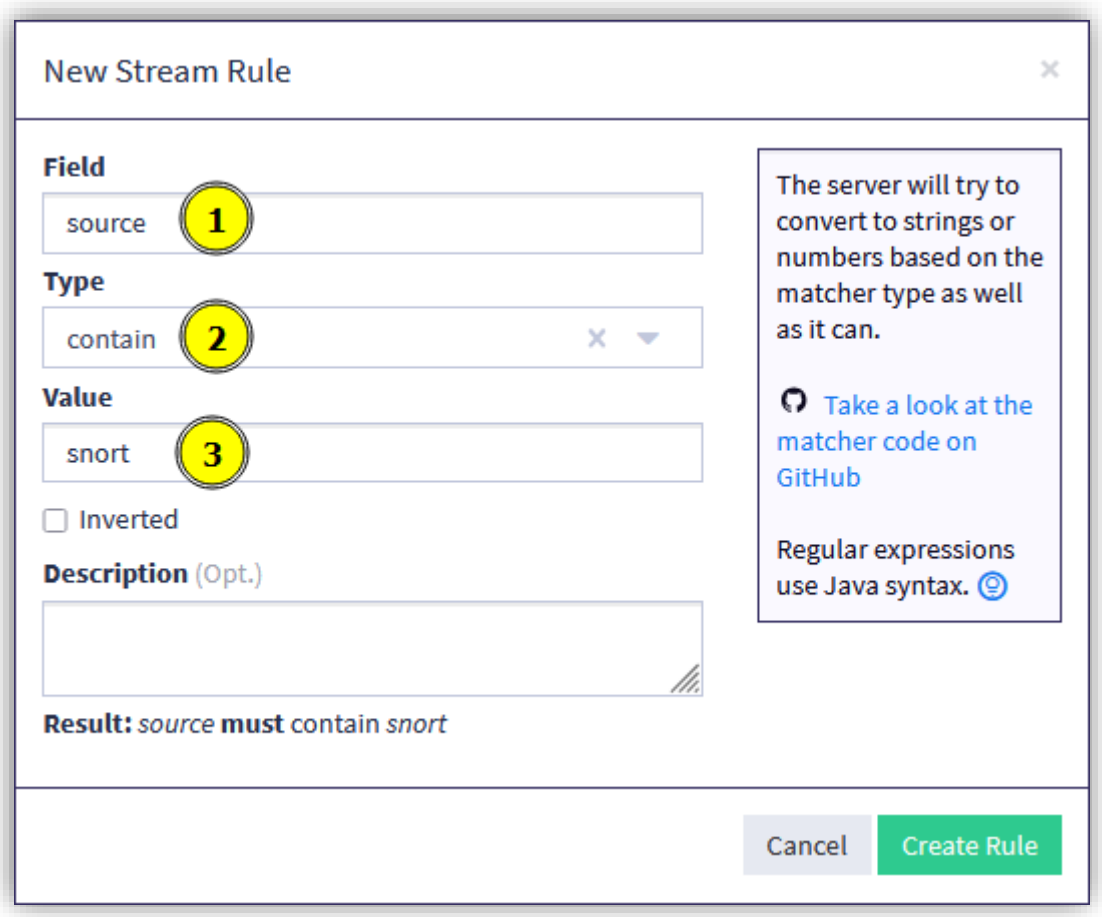


1. Titre du stream
2. Description
3. Indice utilisé pour la rétention des logs (celui créé précédemment)
4. Supprime les logs du stream par défaut afin de ne pas avoir de doublon et surcharger le stockage

Nous voyons ensuite notre stream créé, nous allons donc pouvoir créer notre règle de filtrage des logs via **« manage rules »** :

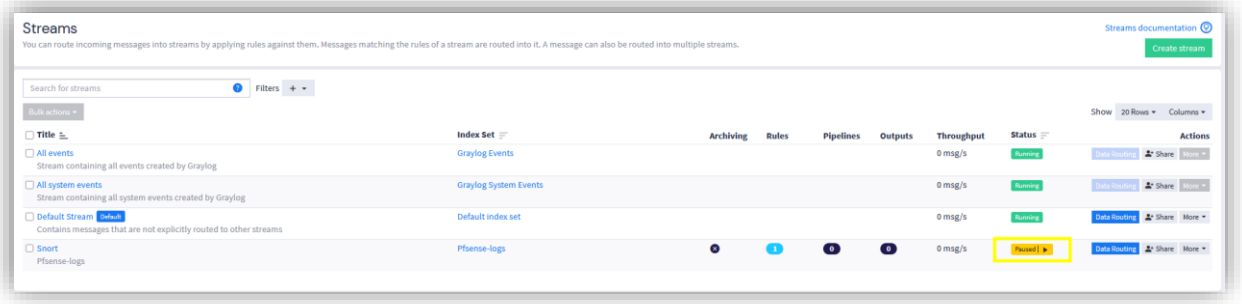


Nous renseignons les paramètres suivants :

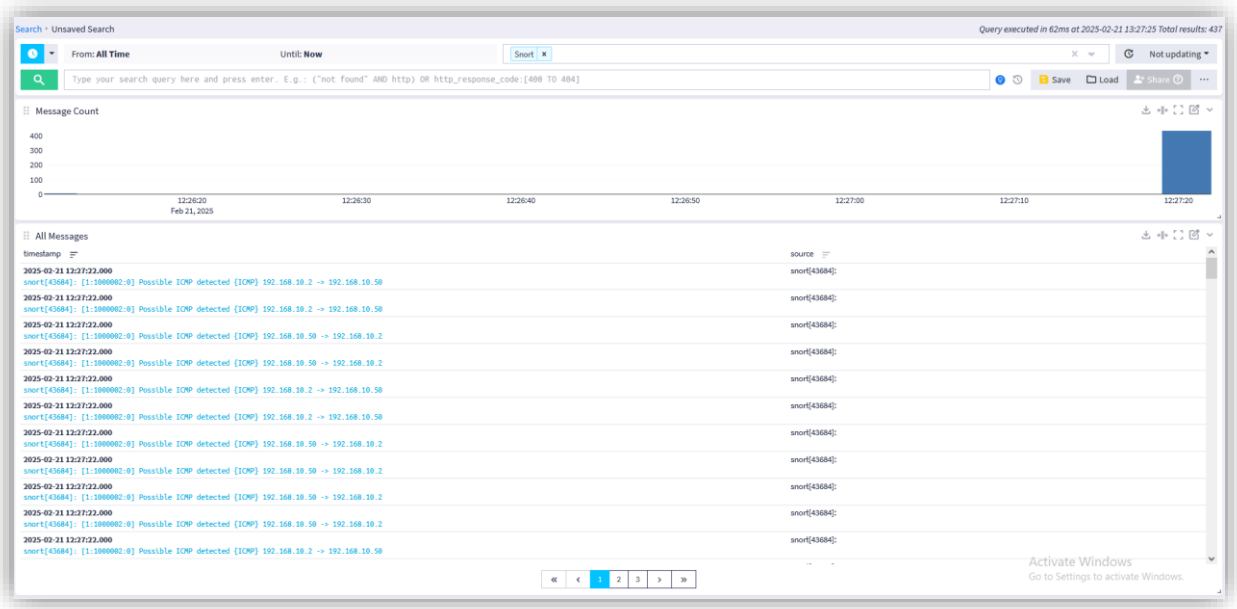


1. Filtrage des logs sur la source
2. Doit contenir la valeur dans le 3
3. Valeur à respecter pour la règle

Nous allons maintenant lancer notre stream en cliquant sur le bouton played :

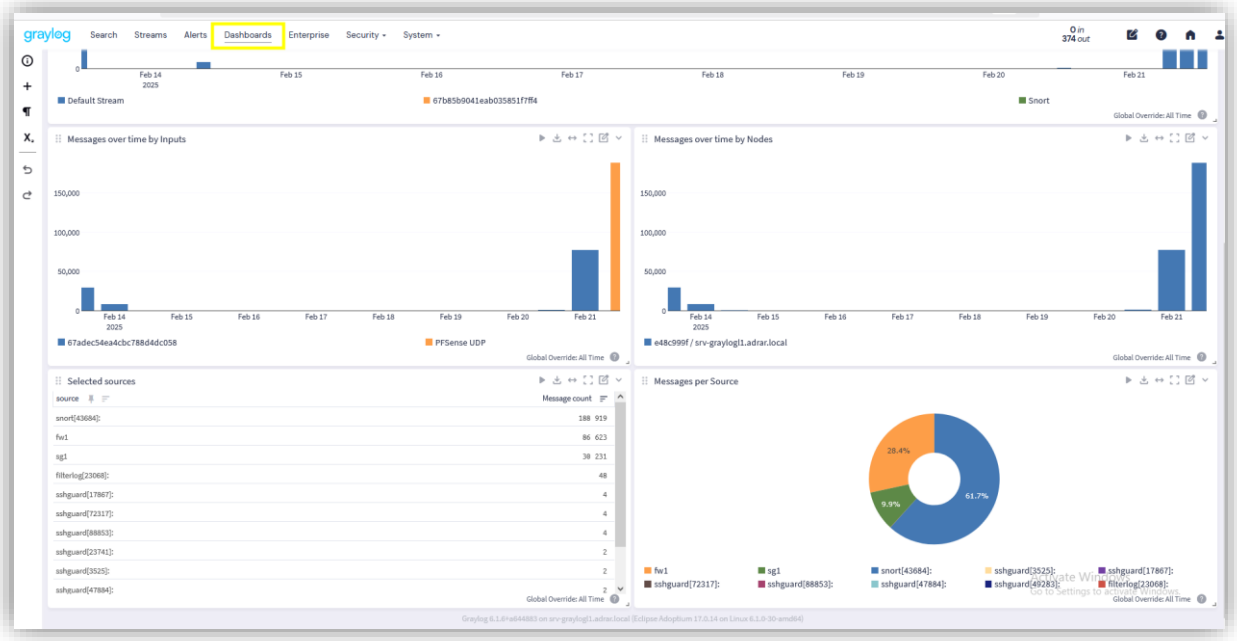


En cliquant sur notre stream nous retrouvons maintenant les logs concernant Snort uniquement :



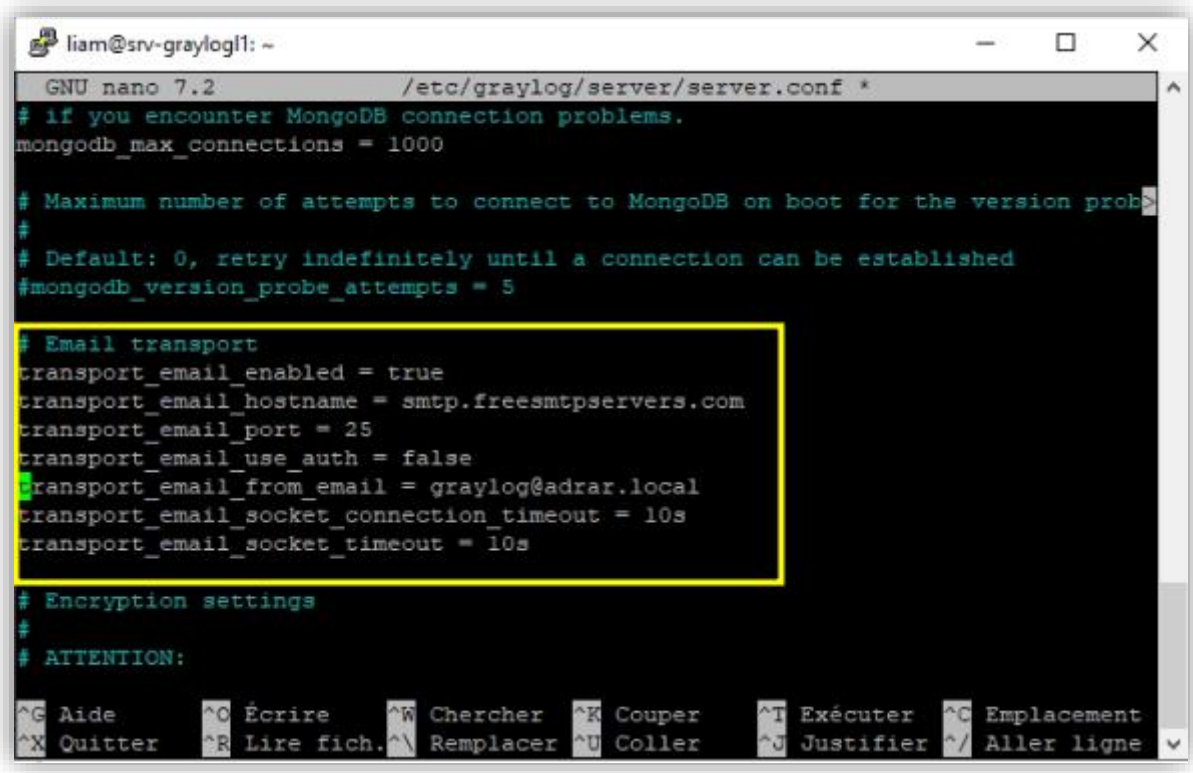
Dans l'onglet Dashboard nous pouvons aussi créer des vues personnalisées et les associer à notre stream pour avoir des statiques sur les logs traité par Graylog (nombre d'alertes par services, date des alertes, statistiques sur les attaques remontées par Snort ect…).

Voici un exemple en utilisant le Dashboard par défaut que nous retrouvons dans le menu « **Dashboard** » :



Il est possible de personnaliser ces Dashboards à souhait afin d'affiner les statistiques selon les besoins cependant un certain temps de configuration sera nécessaire afin d'extraire correctement les informations des logs via des regexs et extractors personnalisés.

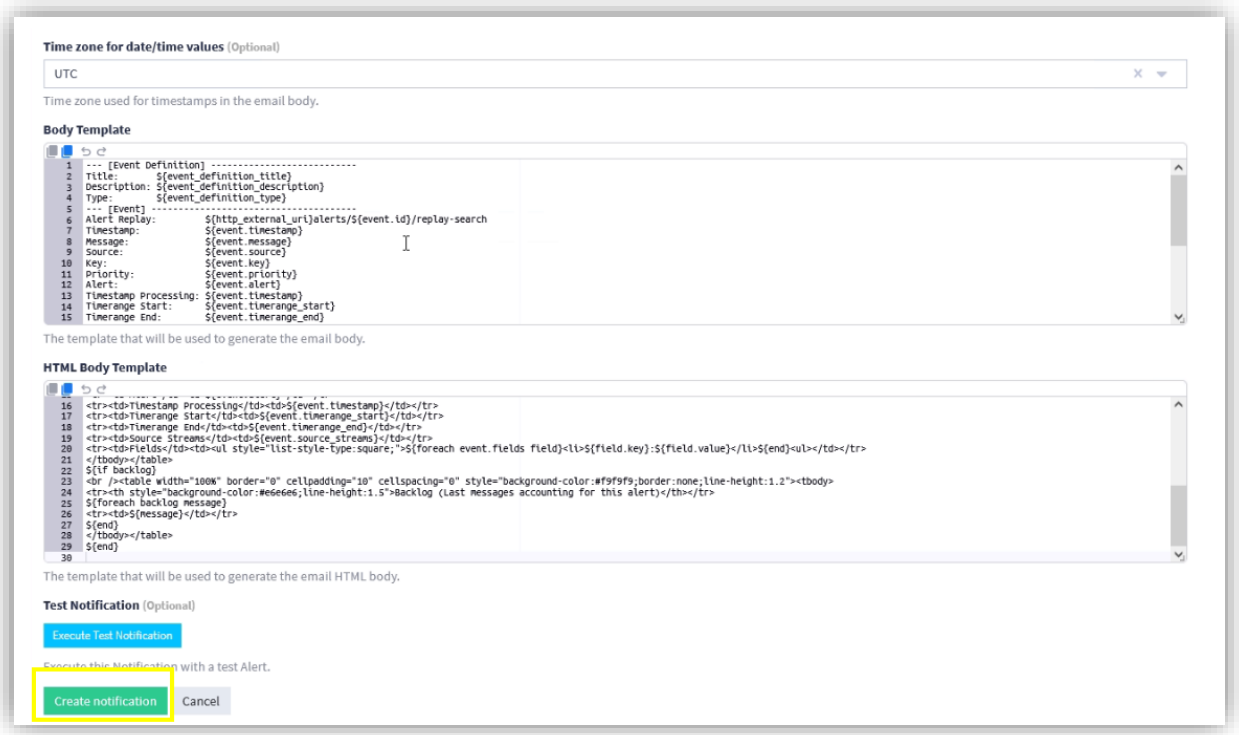
Maintenant nous allons configurer les remontées d'alertes par mail.
Nous commençons par configurer le serveur SMTP dans le fichier
« /etc/graylog/server/server.conf »



Maintenant nous allons dans le menu **« Alerts –> Notifications »** puis **« create Notifications »**

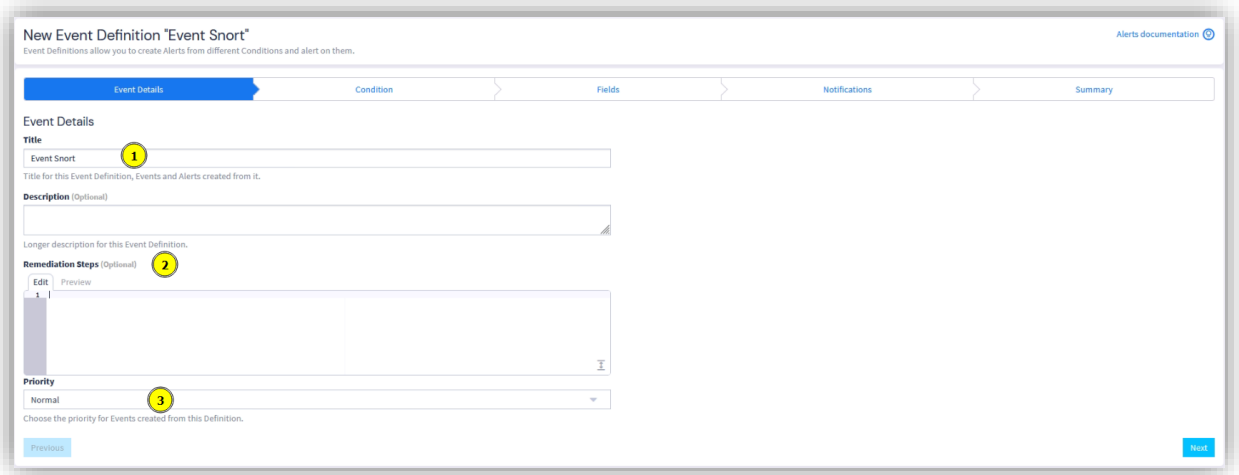
1. Titre de la notification
2. Type de notification
3. Objet du mail
4. Utilisateur pour l'envoi du mail

Puis nous cliquons sur **« Create notification »**

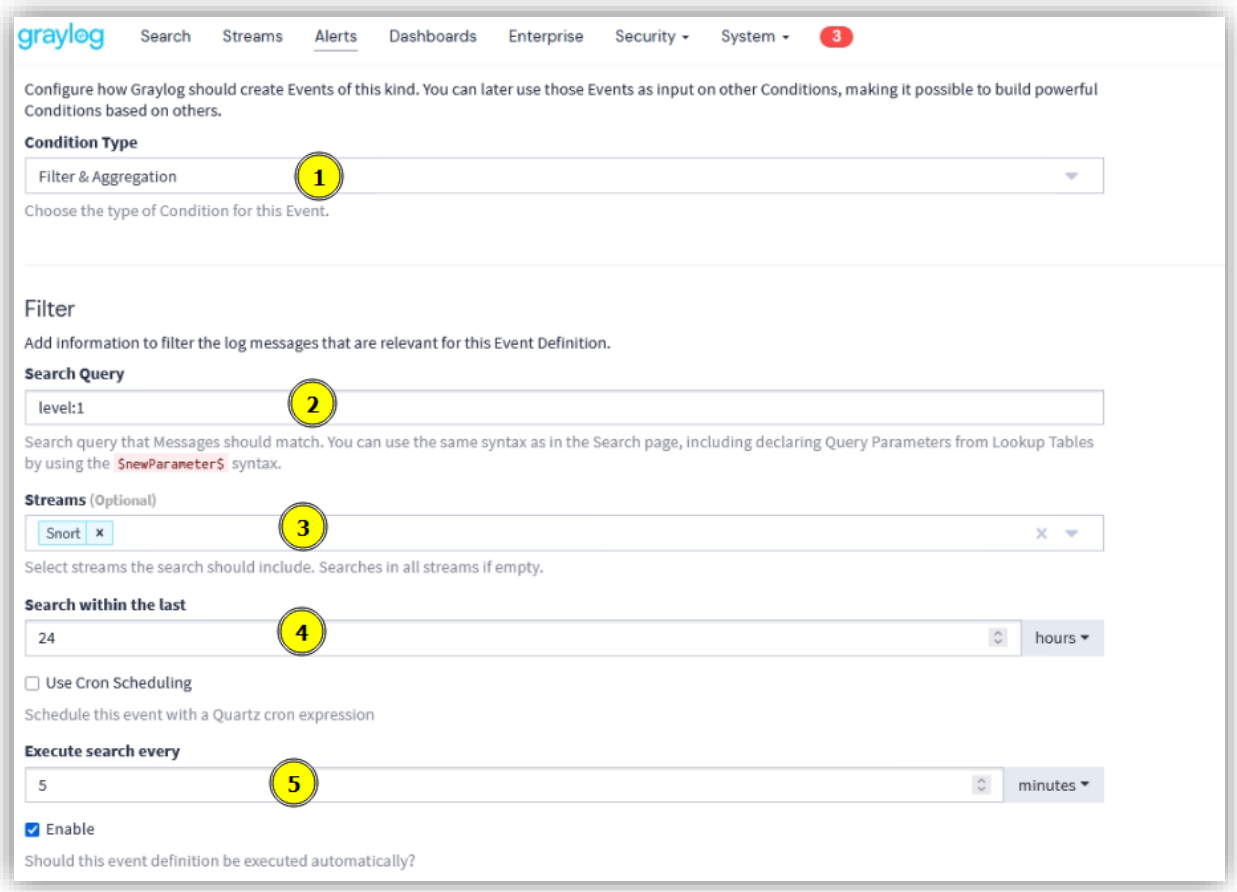


Nous voyons également ci-dessus le **« Body template** » qui est 100% personnalisable afin d'envoyer les informations voulues concernant l'événement.

Maintenant nous allons dans le menu « **Alerts -> Events definition** » puis nous allons cliquer sur « **create event definition** »

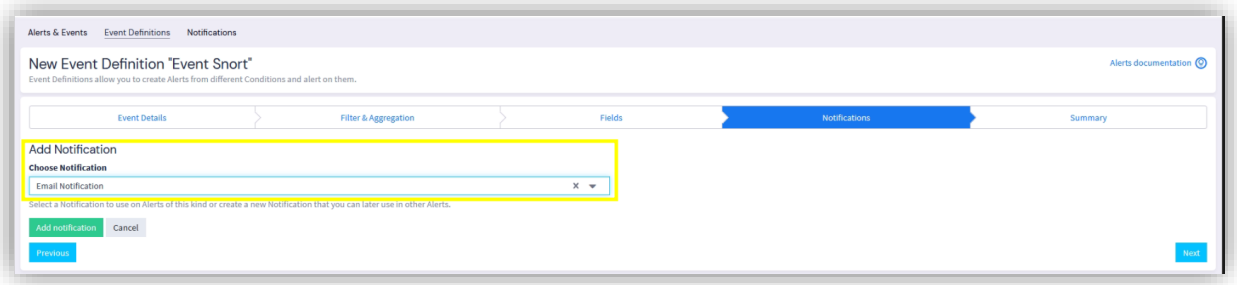


1. Titre de l'event
2. Possibilité de rajouter des étapes de remédiation
3. Priorité de l'event

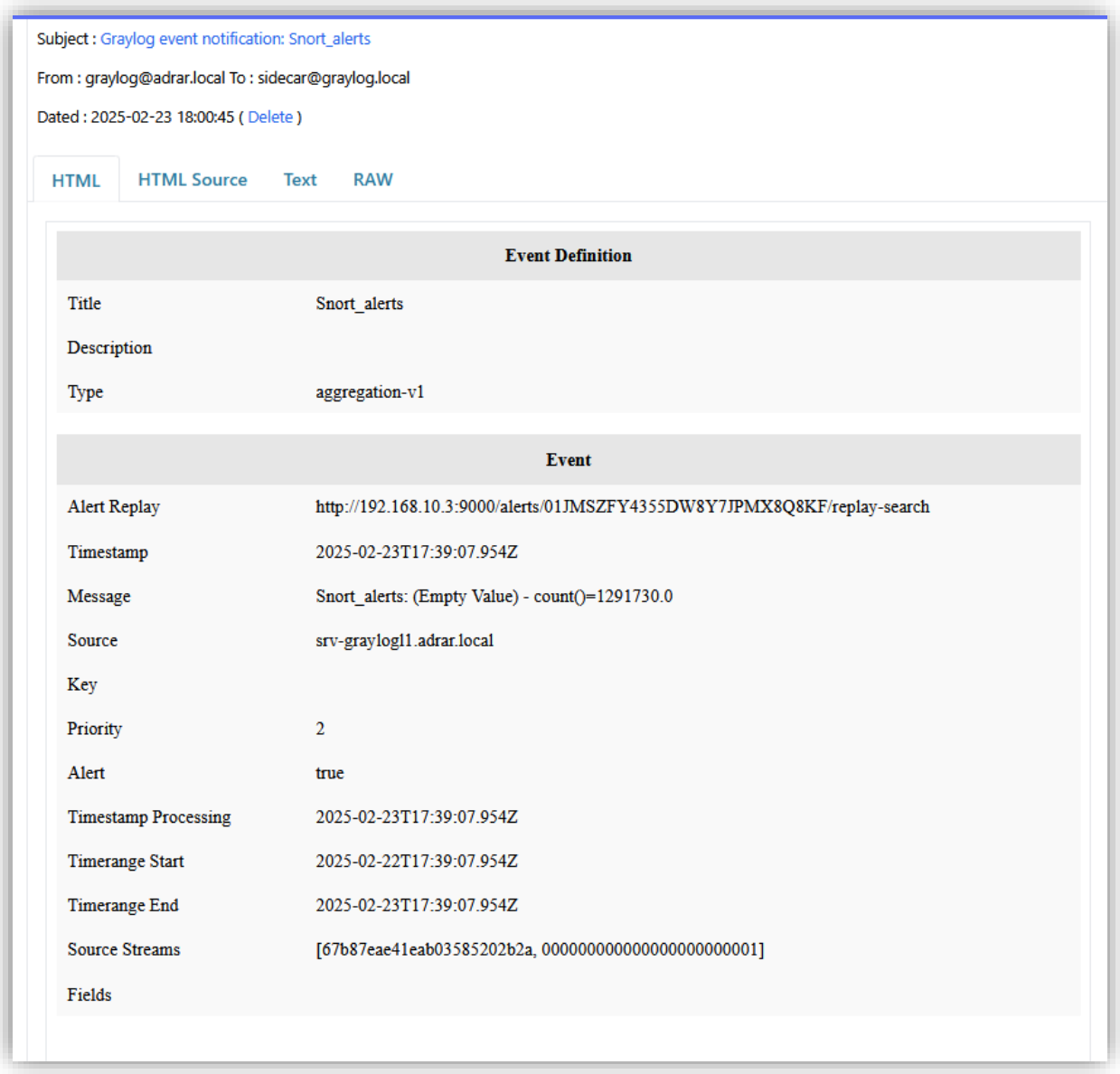


1. Paramètre de déclenchement de l'alerte
2. Filtre sur les alertes de niveau 1 minimum (pour nos tests, le niveau d'alerte est à ajuster au besoin)
3. Filtre sur le stream créé précédemment pour les alertes Snort
4. Cherche dans les logs générés depuis les dernières 24 heures
5. Exécute la recherche toutes les 5 mins

Puis nous définissons la notification créée précédemment :



Ensuite nous effectuons un test de notification de l'alerte :



Subject : Graylog event notification: Snort_alerts

From : graylog@adrar.local To : sidecar@graylog.local

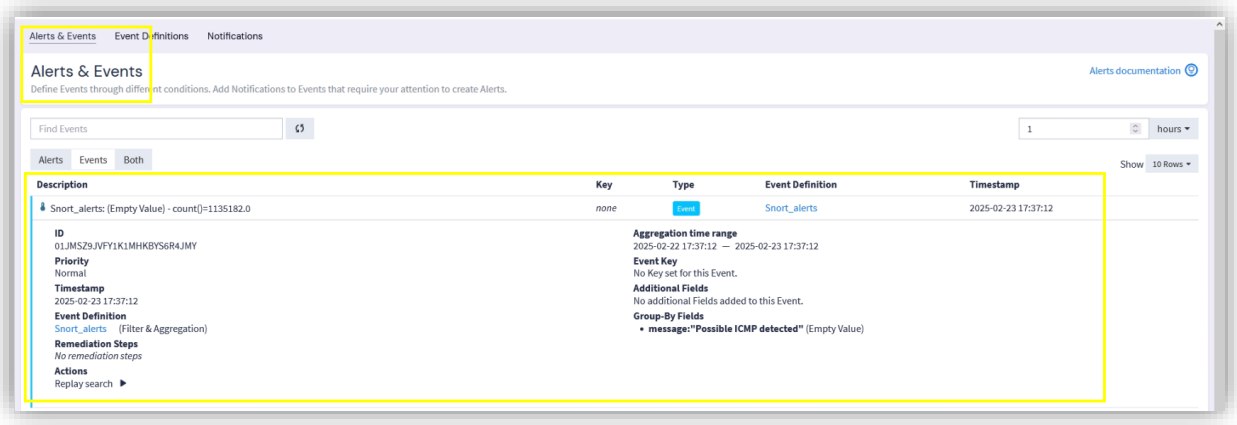
Dated : 2025-02-23 18:00:45 (Delete)

HTML | HTML Source | Text | RAW

Event Definition	
Title	Snort_alerts
Description	
Type	aggregation-v1

Event	
Alert Replay	http://192.168.10.3:9000/alerts/01JMSZFY4355DW8Y7JPMX8Q8KF/replay-search
Timestamp	2025-02-23T17:39:07.954Z
Message	Snort_alerts: (Empty Value) - count()=1291730.0
Source	srv-graylog11.adrar.local
Key	
Priority	2
Alert	true
Timestamp Processing	2025-02-23T17:39:07.954Z
Timerange Start	2025-02-22T17:39:07.954Z
Timerange End	2025-02-23T17:39:07.954Z
Source Streams	[67b87eae41eab03585202b2a, 000000000000000000000001]
Fields	

Les alertes sont également disponibles dans le menu « **Alerts et events** » ou plus de détail sont disponible :



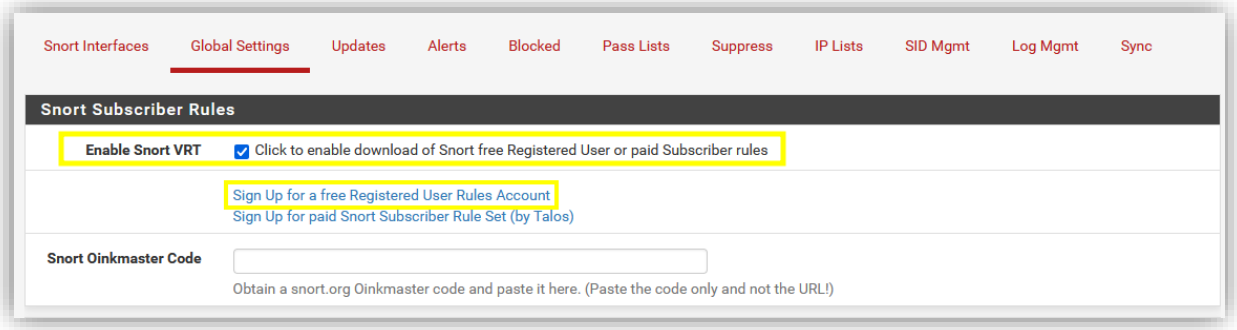
Il est aussi possible de rejouer l'alerte afin de remonter tous les événements associés à cette alerte.

Graylog étant un outil très complet, il est possible de modifier toutes les alertes, stream, notifications, dashboard… à souhait afin d'avoir les informations les plus précises possible. Cependant un certain temps de configuration sera nécessaire afin d'arriver au résultat les plus optimisés possible car cela implique la création de nouveau extractor (que l'on peut télécharger via la market place Graylog ou les créer manuellement).

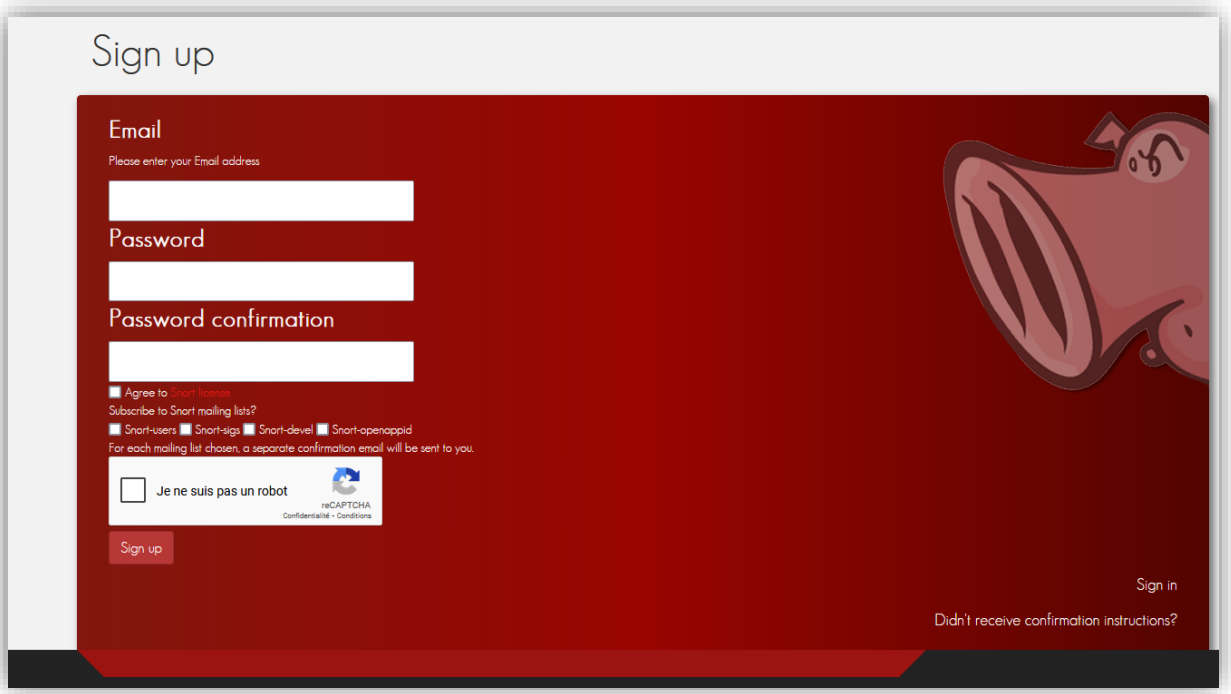
2.6 Configuration des règles « Snort Community »

Maintenant que nous avons qualifié notre POC nous allons pouvoir mettre en place les règles Snort Community qui regroupent énormément de règles créées par la communauté pour les attaques les plus courantes (cela évite la création manuelle de toutes les règles qui demande un certain temps).

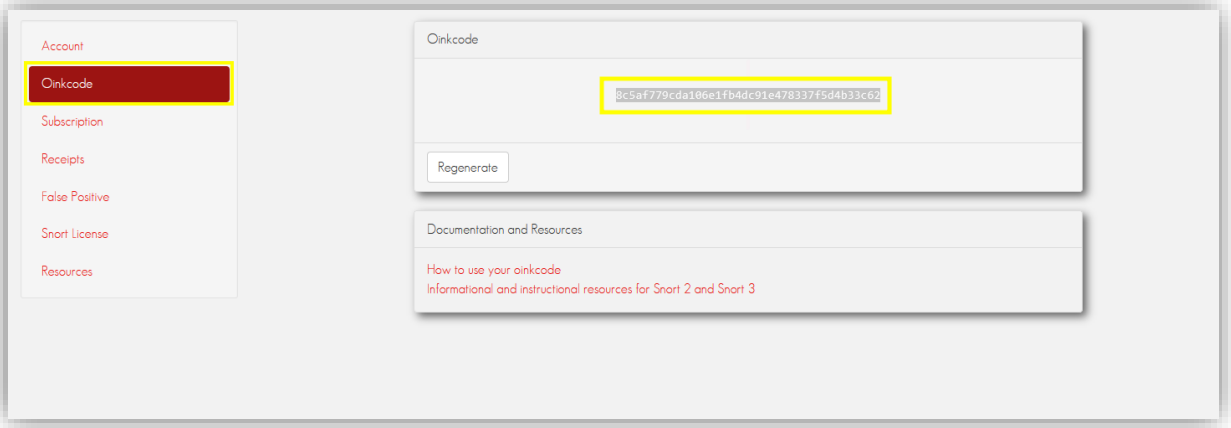
Nous éditons donc notre interface puis nous allons dans le menu **« Snort -> Global Settings »** pour configurer les options suivantes :

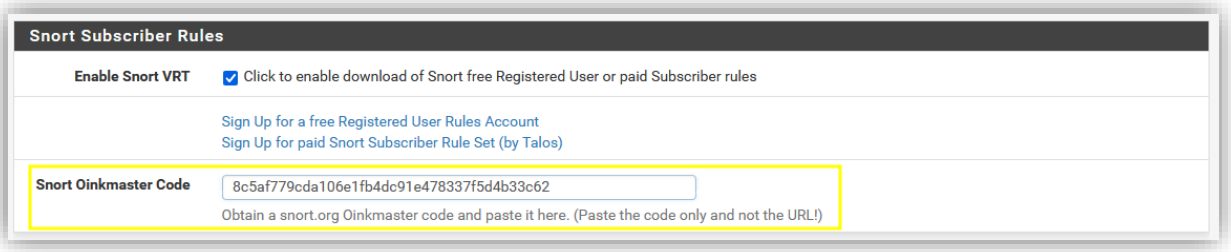


En cliquant sur **« Sign up a free … »** nous somme redirigées vers le site de Snort afin de récupérer un token pour télécharger les règles communautaires.

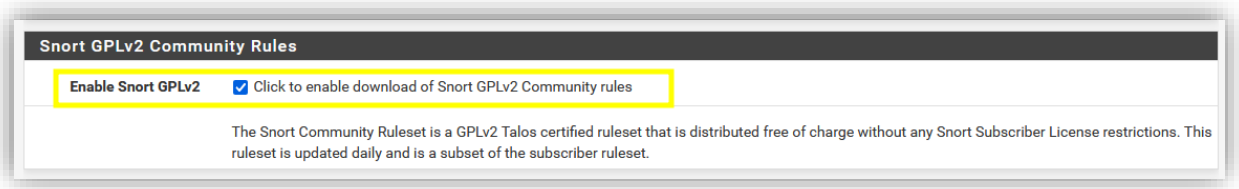


Une fois notre compte créé dans le menu **« Account -> Oinkcode »** nous retrouvons notre token à renseigner dans Pfsense :

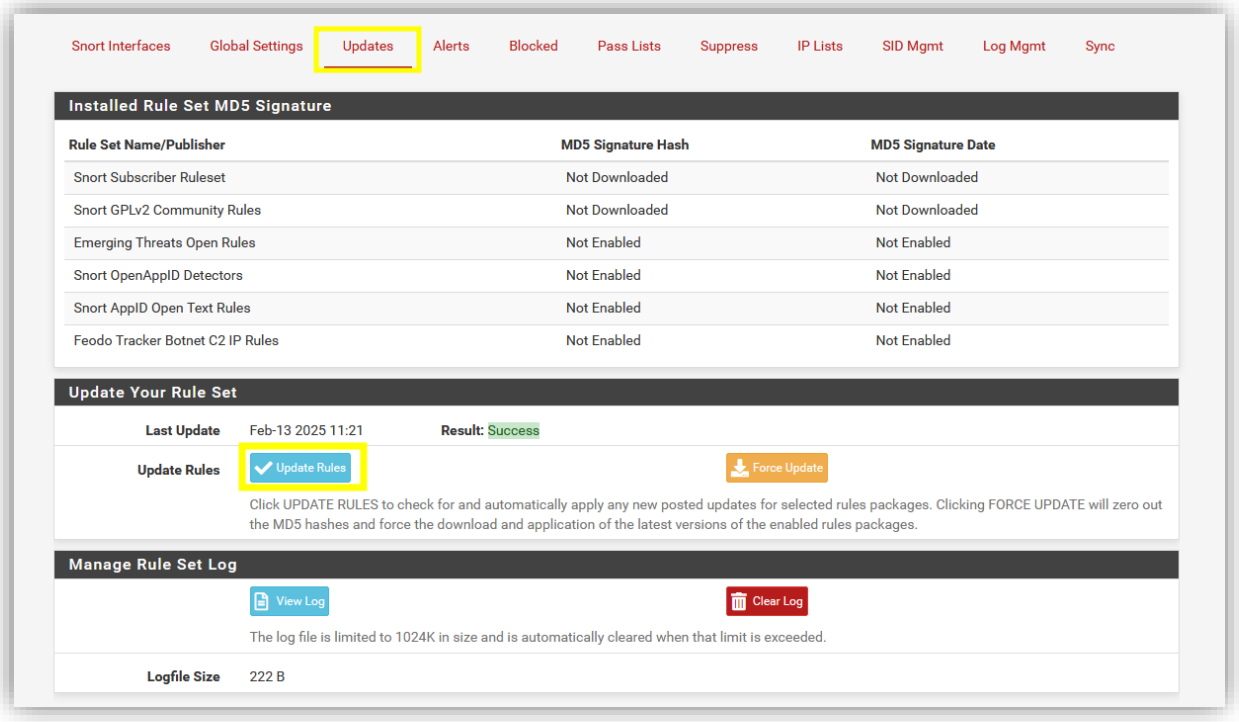




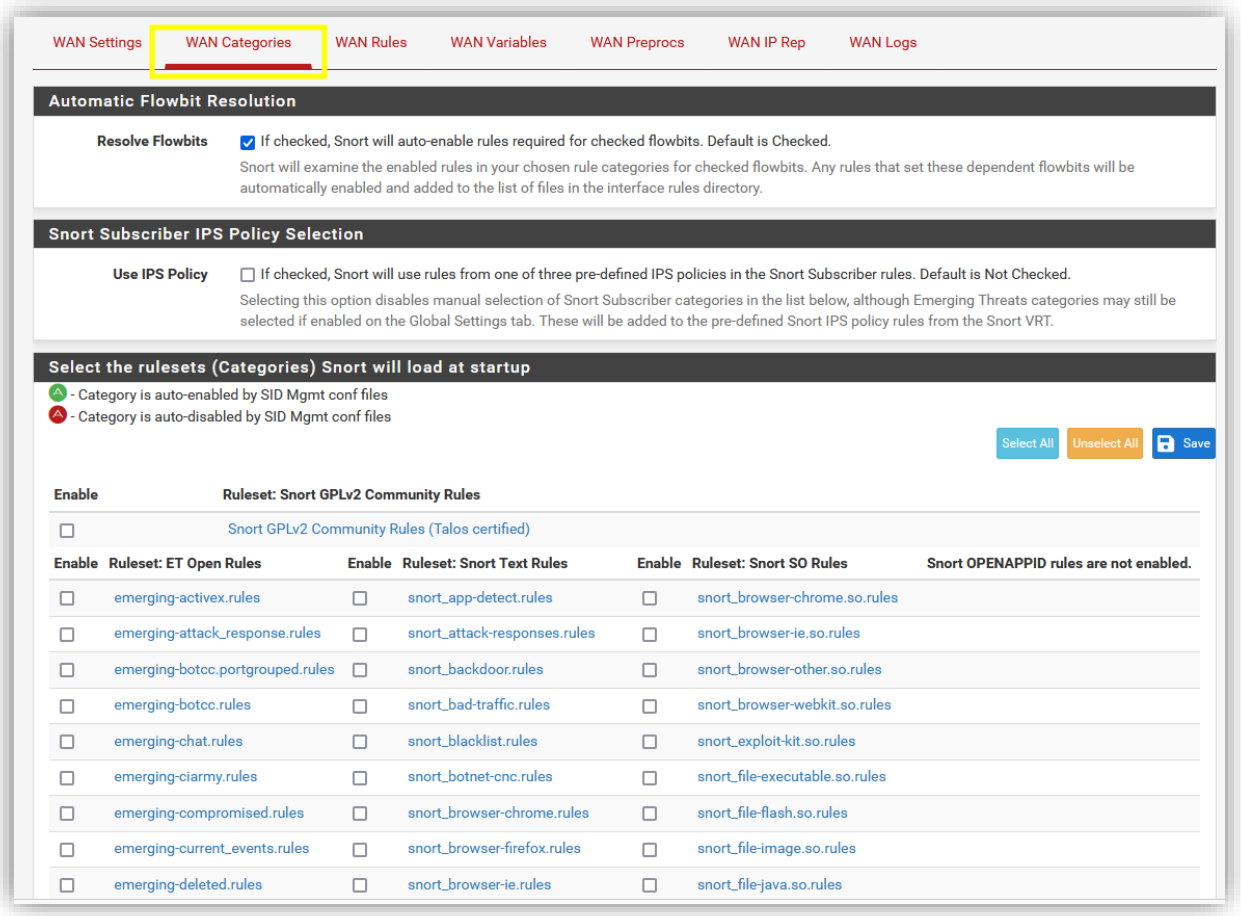
Nous activons ensuite les règles communautaires :



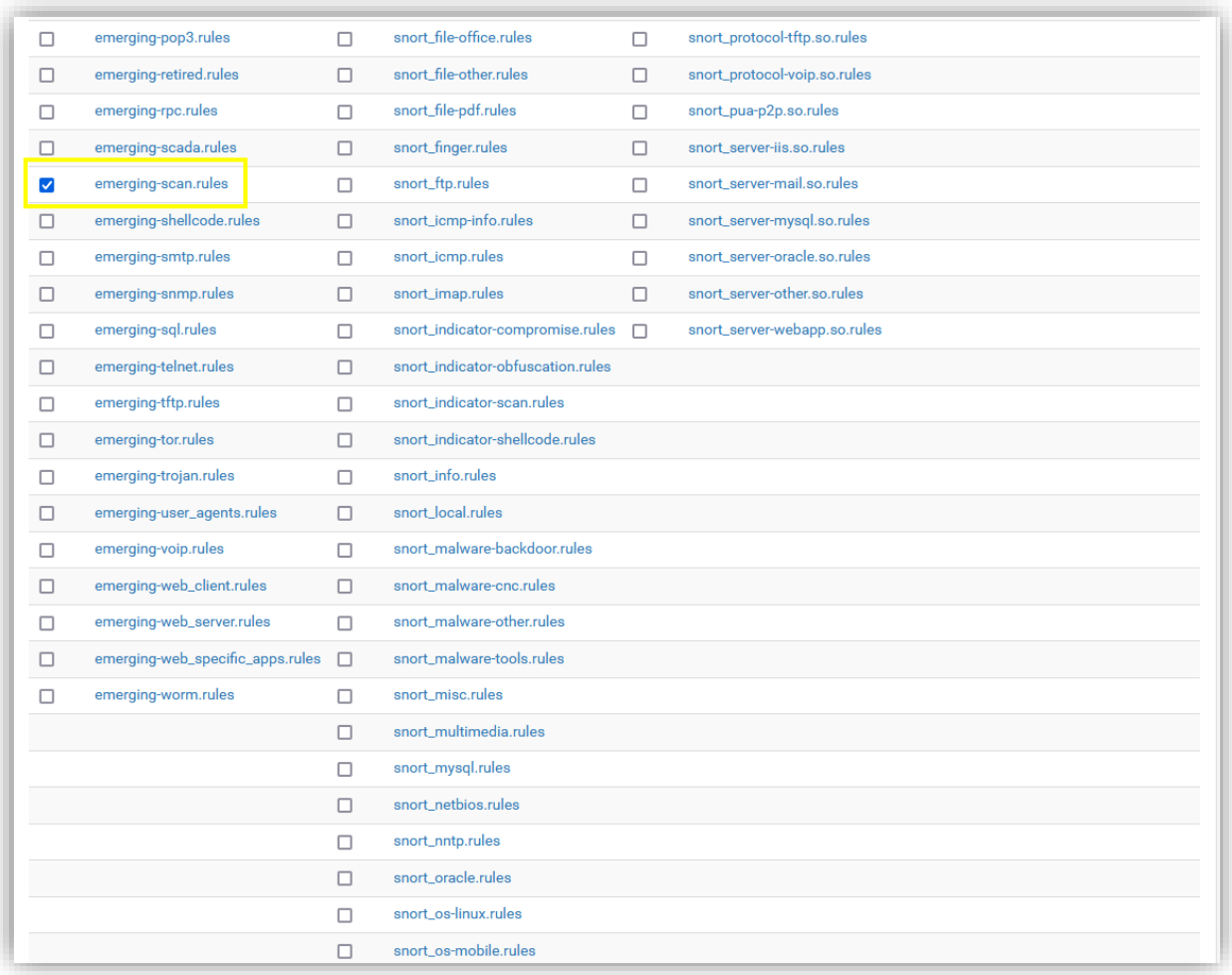
Nous mettons à jour les règles via le menu **« Updates »** :



Dans le menu de l'interface puis **« WAN Categories »** nous retrouvons toutes les règles à activer au besoin :

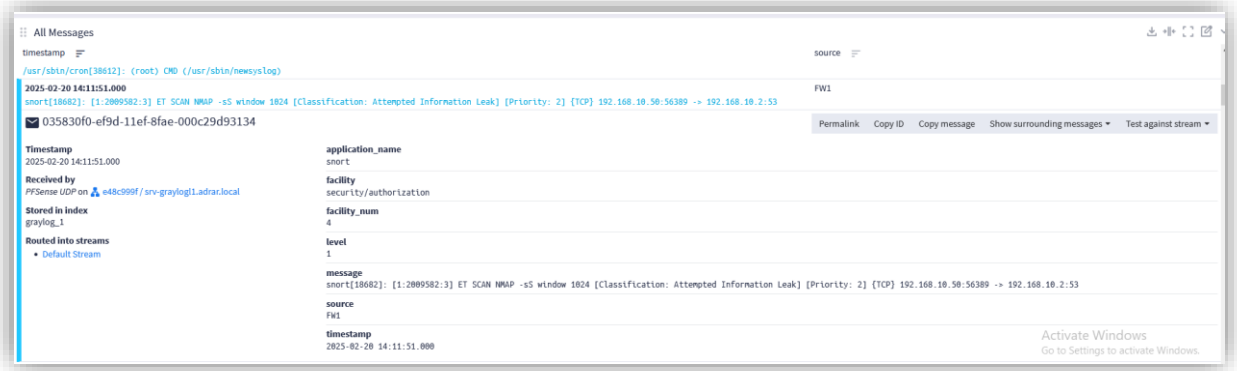


Pour notre exemple, nous allons activer **« emerging-scan rules »** afin d'alerter en cas de scan de port ouvert sur le réseau par exemple :



emerging-pop3.rules	snort_file-office.rules	snort_protocol-tftp.so.rules
emerging-retired.rules	snort_file-other.rules	snort_protocol-voip.so.rules
emerging-rpc.rules	snort_file-pdf.rules	snort_pua-p2p.so.rules
emerging-scada.rules	snort_finger.rules	snort_server-iis.so.rules
emerging-scan.rules	snort_ftp.rules	snort_server-mail.so.rules
emerging-shellcode.rules	snort_icmp-info.rules	snort_server-mysql.so.rules
emerging-smtp.rules	snort_icmp.rules	snort_server-oracle.so.rules
emerging-snmp.rules	snort_imap.rules	snort_server-other.so.rules
emerging-sql.rules	snort_indicator-compromise.rules	snort_server-webapp.so.rules
emerging-telnet.rules	snort_indicator-obfuscation.rules	
emerging-tftp.rules	snort_indicator-scan.rules	
emerging-tor.rules	snort_indicator-shellcode.rules	
emerging-trojan.rules	snort_info.rules	
emerging-user_agents.rules	snort_local.rules	
emerging-voip.rules	snort_malware-backdoor.rules	
emerging-web_client.rules	snort_malware-cnc.rules	
emerging-web_server.rules	snort_malware-other.rules	
emerging-web_specific_apps.rules	snort_malware-tools.rules	
emerging-worm.rules	snort_misc.rules	
	snort_multimedia.rules	
	snort_mysql.rules	
	snort_netbios.rules	
	snort_nntp.rules	
	snort_oracle.rules	
	snort_os-linux.rules	
	snort_os-mobile.rules	

Puis sur notre Kali nous allons faire un NMAP et constater sur Graylog l'alerte remontée.



Attention : Beaucoup de règles sont disponibles, il est nécessaire de faire un tri et d'activer celle qui nous semble les plus pertinentes afin de ne pas surcharger notre SIEM de logs qui ne sont pas pertinents.

3. Conclusion

Dans le cadre de ce projet, nous avons mis en place une solution de sécurisation du réseau pour l’ADRAR en intégrant un service NIDS avec Snort sur le pare-feu Pfsense.
Cette solution permet de détecter les attaques de type Brute Force FTP, DDoS ICMP et ARP Spoofing mais aussi les scans réseau… tout en générant des alertes pour une surveillance proactive.

Afin d’améliorer la gestion des événements de sécurité, nous avons également déployé un serveur Graylog pour la centralisation et l’analyse des logs.
Grâce à son interface intuitive et ses capacités de filtrage avancées, l’ADRAR pourra mieux visualiser les tentatives d’intrusion et réagir efficacement en cas d’incident.

À terme, cette infrastructure sécurisée et évolutive permettra à l’ADRAR de mieux protéger son réseau contre les menaces et d’assurer une supervision efficace des événements de sécurité.

4. Annexes

Installation et configuration de Snort sur Pfsense : https://iritt.medium.com/setting-up-snort-for-network-monitoring-on-pfsence-for-your-cybersecurity-lab-f724e48d6221

Installation et configuration de Graylog : https://www.it-connect.fr/tuto-graylog-sur-debian-centraliser-et-analyser-logs/

Créer des règles Snort personalisées : https://cyvatar.ai/write-configure-snort-rules/

Marketplace Graylog : https://community.graylog.org/c/marketplace/31

Recommandation ANSSI SIEM https://cyber.gouv.fr/sites/default/files/2022/01/anssi-guide-recommandations_securite_architecture_systeme_journalisation.pdf