

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 | 2 |
| 1.1 | Demandes du client..... | 2 |
| 1.2 | Evolution de l'infrastructure | 3 |
| 1.3 | Choix de la solution | 4 |
| 1.4 | Sécurisation des accès..... | 5 |
| 2. | Configuration technique..... | 5 |
| 2.1 | Installation et configuration de Snort | 5 |
| 2.2 | Configuration des règles Snort DDoS ICMP..... | 8 |
| 2.3 | Configuration règle Snort Brut Force FTP..... | 11 |
| 2.4 | Configuration Snort en mode NIDS..... | 12 |
| 2.5 | Intégration avec Graylog | 15 |
| 2.6 | Configuration des règles « Snort Community » | 32 |
| 3. | Conclusion..... | 36 |
| 4. | Annexes | 38 |

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..

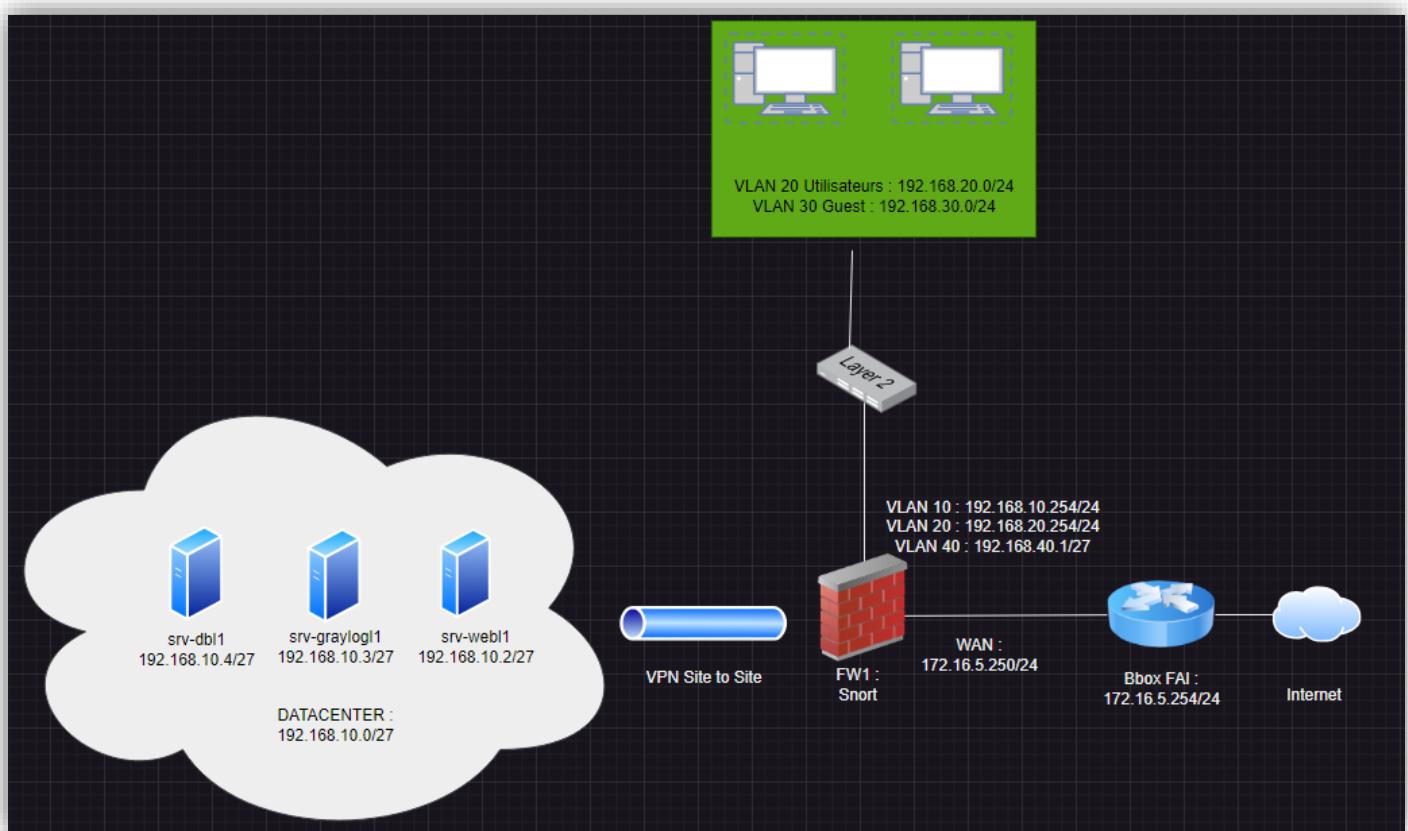
De plus, ils nous sont demandés de mettre en place un outil de centralisation des logs afin de faciliter la gestion des informations et événements liés à la sécurité.

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 service 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 peut être inadaptée car il pourrait bloquer du trafic légitime.

C'est pour cela que dès la fin du POC NIPS l'option sera désactivé, mais pourra 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 de surveiller toutes les pattes de nos réseaux
- Léger 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 anticipées 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 « **sécurité** » 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 potentielles attaques DDOS ICMP Flood
- Règle Snort afin de détecter de potentielles attaques 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 connaître :

- **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 etc...) 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 possibles 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 :

The screenshot shows the 'Installed Packages' tab selected in the 'System / Package Manager / Installed Packages' interface. A single package, 'snort', is listed in the table:

| Name | Category | Version | Description | Actions |
|---------|----------|----------|---|---------|
| ✓ snort | security | 4.1.6_17 | Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection. | |

Below the table, it says 'Package Dependencies:' followed by 'snort-2.9.20_8'.

Une fois cela fait, nous allons commencer la configuration.

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

The screenshot shows the 'Services' menu with various services listed. The 'Snort' service is highlighted with a yellow box.

- Auto Config Backup
- Captive Portal
- DHCP Relay
- DHCP Server
- DHCPv6 Relay
- DHCPv6 Server
- DNS Forwarder
- DNS Resolver
- Dynamic DNS
- IGMP Proxy
- NTP
- PPPoE Server
- Router Advertisement
- SNMP
- Snort**
- UPnP & NAT-PMP
- Wake-on-LAN

Ensute 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 :

The screenshot shows the 'Snort Interfaces' tab selected in the top navigation bar. Below it is a table titled 'Interface Settings Overview' with columns: Interface, Snort Status, Pattern Match, Blocking Mode, Description, and Actions. A green 'Add' button is located in the bottom right corner of the table area.

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

The screenshot shows the 'General Settings' configuration page. It includes fields for Enable (checkbox checked, circled 1), Interface (set to LAN (eth0), circled 2), Description (set to LAN, circled 3), and Snap Length (set to 1518, circled 4). Below this is an 'Alert Settings' section.

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

Dans « **Alert Setting** » :

The screenshot shows the 'Alert Settings' configuration page. It includes fields for Send Alerts to System Log (checkbox checked, circled 1), System Log Facility (set to LOG_AUTH, circled 2), System Log Priority (set to LOG_ALERT, circled 3), Enable Packet Captures (checkbox unchecked), and Enable Unified2 Logging (checkbox unchecked).

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 :

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

Snort est désormais actif :

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

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

| General Settings | |
|--|---|
| Remove Blocked Hosts Interval | 30 MINS |
| Please select the amount of time you would like hosts to be blocked. In most cases, one hour is a good choice. | |
| Remove Blocked Hosts After Deinstall | <input checked="" type="checkbox"/> Click to clear all blocked hosts added by Snort when removing the package. Default is checked. |
| Keep Snort Settings After Deinstall | <input checked="" type="checkbox"/> Click to retain Snort settings after package removal. |
| Startup/Shutdown Logging | <input type="checkbox"/> Click to output detailed messages to the system log when Snort is starting and stopping. Default is not checked. |

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 :

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Interface Settings Overview

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

+ Add Delete

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

LAN Settings LAN Categories **LAN Rules** LAN Variables LAN Preprocs LAN IP Rep LAN Logs

Available Rule Categories

Category Selection: **custom.rules**
Select the rule category to view and manage.

Defined Custom Rules

Save Cancel Clear

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** » : définit 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 :

The screenshot shows the 'Available Rule Categories' section with 'custom.rules' selected. In the 'Defined Custom Rules' section, a single rule is listed:

```
alert icmp any any -> $HOME_NET any (msg:"Possible ICMP detected"; threshold:type threshold, track by_src, count 10, sec
```

At the bottom right, there are 'Save', 'Cancel', and 'Clear' buttons, with 'Save' highlighted.

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 :

The screenshot shows the 'Services / Snort / Interface Settings / LAN - Rules' interface. The 'Alerts' tab is highlighted with a yellow box. Below it, the 'LAN Rules' tab is also highlighted with a red box.

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 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 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 |

- 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 réponses du serveur FTP contenant le code 530 qui renvoie un message « login/password incorrect »
 - « **threshold:type threshold, track by_src, count 5, seconds 60;** » : Définit 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 :

| Most Recent 250 Entries from Active Log | | | | | | | | | | |
|---|--------|-----|-------|-------|--------------|-------|----------------|-------|------------------|----------------------------------|
| Date | Action | Pri | Proto | Class | Source IP | SPort | Destination IP | DPort | GID:SID | Description |
| 2025-02-14 10:28:33 | ⚠️ | 0 | TCP | | 192.168.10.2 | 21 | 192.168.10.50 | 34422 | 1:1000003 + ✘ | Potential FTP brute force attack |
| 2025-02-14 10:28:33 | ⚠️ | 0 | TCP | | 192.168.10.2 | 21 | 192.168.10.50 | 34520 | 1:1000003 + ✘ | Potential FTP brute force attack |

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 :

Block Settings

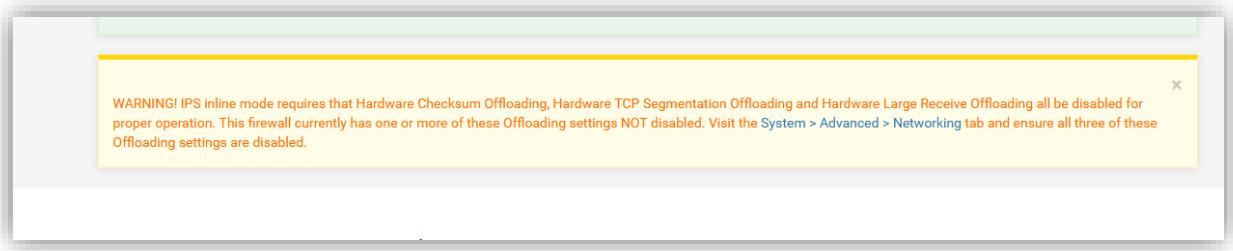
Block Offenders Checking this option will automatically block hosts that generate a Snort alert. Default is Not Checked. (1)

IPS Mode (2)
Select blocking mode operation. Legacy Mode inspects copies of packets while Inline Mode inserts the Snort inspection engine into the network stack between the NIC and the OS. Default is Legacy Mode.
Legacy Mode uses the PCAP engine to generate copies of packets for inspection as they traverse the interface. Some "leakage" of packets will occur before Snort can determine if the traffic matches a rule and should be blocked. Inline mode instead intercepts and inspects packets before they are handed off to the host network stack for further processing. Packets matching DROP rules are simply discarded (dropped) and not passed to the host network stack. No leakage of packets occurs with Inline Mode. WARNING: Inline Mode only works with NIC drivers which properly support Netmap! Supported drivers: bnxt, cc, cxgbe, cxl, em, em, ena, ice, igb, igc, ix, ixgbe, ixl, lem, re, vmx, vtnet. If problems are experienced with Inline Mode, switch to Legacy Mode instead.

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 crée des copies de paquets et qui peut permettre au trafic malveillant d'atteindre le réseau

Attention : Quelques configurations sont nécessaires pour le fonctionnement en mode IPS dans notre LAB.

Dans un 1^{er} temps nous avons ce message qui apparait :



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

| Network Interfaces | |
|---|---|
| Hardware Checksum Offloading | <input checked="" type="checkbox"/> Disable hardware checksum offload Checking this option will disable hardware checksum offloading. Checksum offloading is broken in some hardware, particularly some Realtek cards. Rarely, drivers may have problems with checksum offloading and some specific NICs. This will take effect after a machine reboot or re-configure of each interface. |
| Hardware TCP Segmentation Offloading | <input checked="" type="checkbox"/> Disable hardware TCP segmentation offload Checking this option will disable hardware TCP segmentation offloading (TSO, TS04, TS06). This offloading is broken in some hardware drivers, and may impact performance with some specific NICs. This will take effect after a machine reboot or re-configure of each interface. |
| Hardware Large Receive Offloading | <input checked="" type="checkbox"/> Disable hardware large receive offload Checking this option will disable hardware large receive offloading (LRO). This offloading is broken in some hardware drivers, and may impact performance with some specific NICs. This will take effect after a machine reboot or re-configure of each interface. |
| hn ALTQ support | <input checked="" type="checkbox"/> Enable the ALTQ support for hn NICs. Checking this option will enable the ALTQ support for hn NICs. The ALTQ support disables the multiqueue API and may reduce the system capability to handle traffic. This will take effect after a machine reboot. |
| ARP Handling | <input type="checkbox"/> Suppress ARP messages This option will suppress ARP log messages when multiple interfaces reside on the same broadcast domain. |
| Reset All States | <input type="checkbox"/> Reset all states if WAN IP Address changes This option resets all states when a WAN IP Address changes instead of only states associated with the previous IP Address. |

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 sera 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 :

The screenshot shows the 'Services / Snort / Blocked Hosts' page. The 'Blocked' tab is selected. The main section displays 'Blocked Hosts and Log View Settings'. It includes a 'Save' button, a 'Clear' button, and refresh options. Below this is a table titled 'Last 500 Hosts Blocked by Snort (only applicable to Legacy Blocking Mode interfaces)'. The table has columns for '#', 'IP', and 'Alert Descriptions and Event Times'. One entry is shown: '# 1 IP 192.168.10.50 Alert Descriptions and Event Times Possible ICMP detected – 2025-02-20 13:44:56 Potential FTP brute force attack – 2025-02-20 11:31:04'. A red 'Remove' button is next to the IP address. A note at the bottom says '1 host IP address is currently being blocked Snort on Legacy Blocking Mode interfaces.'

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 considérée 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 Graylog 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** » :

Syslog UDP

PFSense UDP (87adec54ea4cbbc78dd4dc058) [Edit](#)

On node ★ 48C99F / srv-graylog1.adar.local

```

allow_override_date: true
bind_address: 0.0.0.0
charset_name: UTF-8
expand_structured_data: false
force_rdns: false
number_worker_threads: 6
override_source: FW1
port: 4514
recv_buffer_size: 262144
store_full_message: false
timezone: NotSet
  
```

Throughput / Metrics
1 minute average rate: 0 msg/s
Network IO: ▲ 0B ▲ 0B (total: ▲ 0B ▲ 0B)
Empty messages discarded: 0

Ensuite nous allons renseigner les paramètres suivants :

Launch new *Syslog UDP* input ×

Global
Should this input start on all nodes

Node
e48c999f / srv-graylog1.adrar.local 1 ▼
On which node should this input start

Title
PFSense UDP 2

Bind address
0.0.0.0 3
Address to listen on. For example 0.0.0.0 or 127.0.0.1.

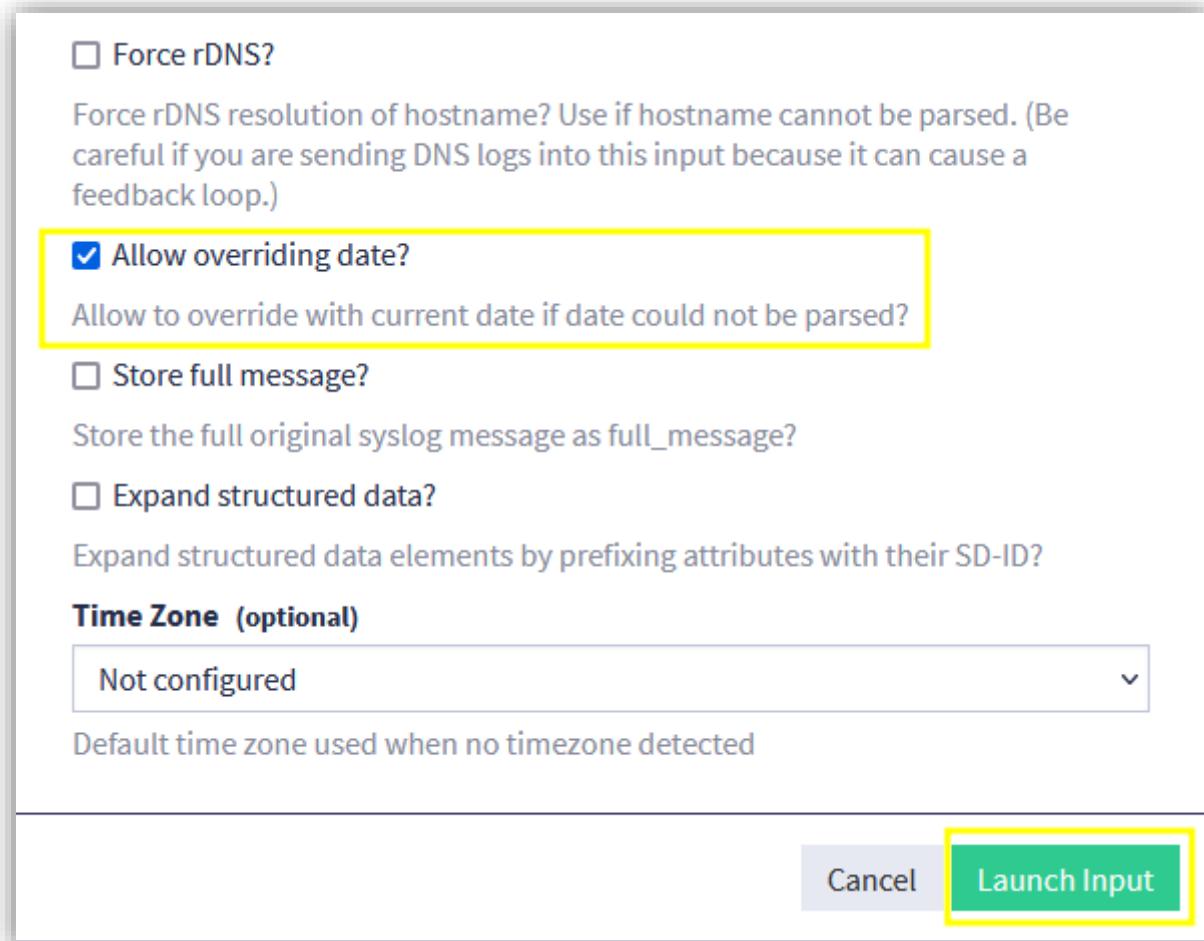
Port
4514 4 ^
Port to listen on.

Receive Buffer Size (optional)
262144 5 ^
The size in bytes of the recvBufferSize for network connections to this input.

No. of worker threads (optional)
2 6 ^
Number of worker threads processing network connections for this input.

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": {
        "fields": [
          {
            "name": "application_name"
          }
        ]
      }
    }
  ]
}
```

```

    "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-zA-Z-]+).*"
    },
    "condition_type": "regex",
    "condition_value": "(^[-a-zA-Z-]+).*"
}
],
"version": "3.2.5"
}

```

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

The screenshot shows the Graylog interface with the 'System / Inputs' menu selected. The 'Inputs' section is displayed, showing a list of configured inputs. A global input for 'PFSense UDP' is listed, and a local input for a node with IP 'e48c99f' is selected. The configuration details for this local input are shown in a modal or expanded view, including fields like 'allow_override_date', 'bind_address', 'charset_name', etc. The top right of the interface shows network metrics: 0 in and 0 out.

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

Extractors of PFsense UDP
Extractors are applied on every message that is received by this input. Use them to extract and transform any text data into fields that allow you easy filtering and analysis later on. Example: Extract the HTTP response code from a log message, transform it to a numeric field and attach it as `http_response_code` to the message.

Add extractor
Start by loading a message to have an example to work on. You can decide whether to load a recent message received by this input, or manually select a message giving its ID.

Actions Import extractors Export extractors

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

Import extractors to PFsense UDP
Exported extractors can be imported to an input. All you need is the JSON export of extractors from any other Graylog setup or from the [Graylog Marketplace](#).

```
Extractors JSON
{
  "type": "lowercase",
  "config": []
},
{
  "order": 0,
  "cursor_strategy": "copy",
  "source_field": "message",
  "target_field": "application_name",
  "extractor_config": [
    {
      "rexes": [
        "^(\\d{0,3}\\d{0,3}\\d{0,3}\\d{0,3})$"
      ],
      "condition_type": "rexes",
      "condition_value": "(\\d{0,3}\\d{0,3}\\d{0,3}\\d{0,3})$"
    }
  ],
  "title": "General Application Name Extractor",
  "extractor_type": "rexes",
  "converters": [],
  "order": 0,
  "cursor_strategy": "copy",
  "source_field": "message",
  "target_field": "application_name",
  "extractor_config": [
    {
      "rexes": [
        "[a-zA-Z]+"
      ],
      "condition_type": "rexes",
      "condition_value": "[a-zA-Z]+"
    }
  ],
  "version": "3.2.5"
}
```

Add extractors to 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 :

Remote Logging Options

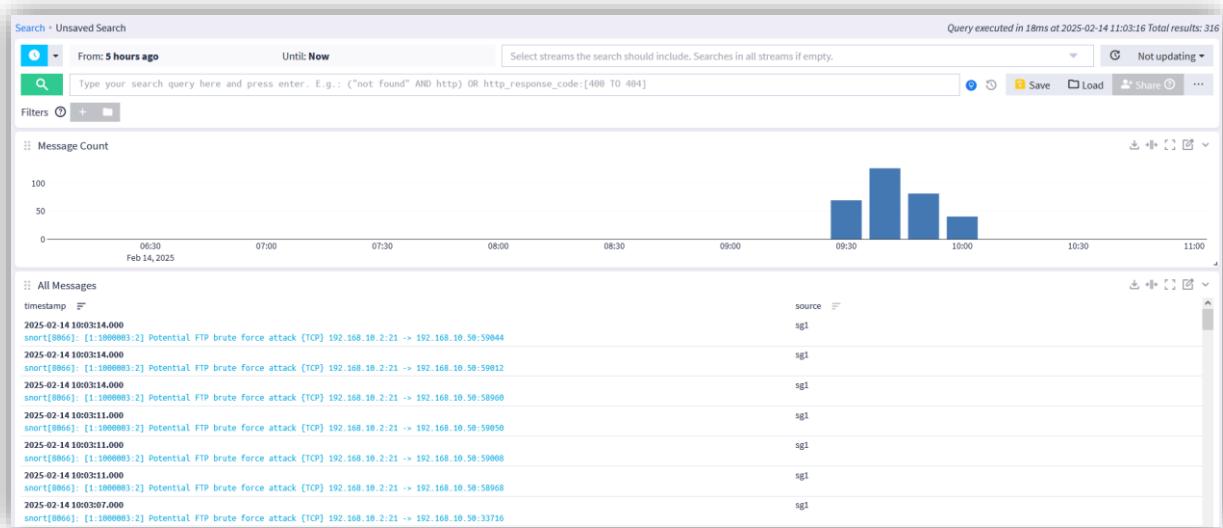
| | | |
|--|---|-----|
| Enable Remote Logging | <input checked="" type="checkbox"/> Send log messages to remote syslog server | (1) |
| Source Address | LAN | (2) |
| This option will allow the logging daemon to bind to a single IP address, rather than all IP addresses. If a single IP is picked, remote syslog servers must all be of that IP type. To mix IPv4 and IPv6 remote syslog servers, bind to all interfaces. | | |
| NOTE: If an IP address cannot be located on the chosen interface, the daemon will bind to all addresses. | | |
| IP Protocol | IPv4 | (3) |
| This option is only used when a non-default address is chosen as the source above. This option only expresses a preference; if an IP address of the selected type is not found on the chosen interface, the other type will be tried. | | |
| Remote log servers | 192.168.10.3:4514 | (4) |
| <input checked="" type="checkbox"/> Everything <input type="checkbox"/> System Events <input type="checkbox"/> Firewall Events <input type="checkbox"/> DNS Events (Resolver/unbound, Forwarder/dnsmasq, filterdns) <input type="checkbox"/> DHCP Events (DHCP Daemon, DHCP Relay, DHCP Client) <input type="checkbox"/> PPP Events (PPPoE WAN Client, L2TP WAN Client, PPTP WAN Client) <input type="checkbox"/> General Authentication Events <input type="checkbox"/> Captive Portal Events <input type="checkbox"/> VPN Events (IPsec, OpenVPN, L2TP, PPPoE Server) <input type="checkbox"/> Gateway Monitor Events <input type="checkbox"/> Routing Daemon Events (RADVD, UPnP, RIP, OSPF, BGP) <input type="checkbox"/> Network Time Protocol Events (NTP Daemon, NTP Client) <input type="checkbox"/> Wireless Events (hostapd) | | |
| Syslog sends UDP datagrams to port 514 on the specified remote syslog server, unless another port is specified. Be sure to set syslogd on the remote server to accept syslog messages from pfSense. | | |
| <input type="button" value="Save"/> | | |

1. Active les logs distants
 2. Interface utilisée 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 :

The screenshot shows the Graylog search interface. The top navigation bar includes links for Streams, Alerts, Dashboards, Enterprise, Security, and System. The search bar is highlighted in yellow and contains the query "response_code:[400 TO 404]". Below the search bar, the search results are displayed, showing a single event with the timestamp "Feb 14, 2025 10:58:00". The event details show a "response_code": 404 and a "request": "GET / HTTP/1.1". The bottom of the screen shows a timeline from 10:58:00 to 11:30:00 with several other events listed.

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** » :

Configuration Information

Title pfSense-logs 1
Descriptive name of the index set.

Description pfSense-logs 2
Add a description of this index set.

Details

Index prefix pfSense 3
A unique prefix used in Elasticsearch indices belonging to this index set. The prefix must start with a letter or number, and can only contain letters, numbers, "-", "\\", and "*".

Analyzer standard 4
Elasticsearch analyzer for this index set.

Index Shards 1 5
Number of search cluster Shards used per index in this Index Set. Increasing the Index Shards improves the search cluster write speed of data stored to this Index Set by distributing the active write index over multiple search nodes. Increasing the Index Shards can degrade search performance and increase the memory footprint of the index. This value should not be set higher than the number of search nodes.

Index Replica 1 6
Number of search cluster Replica Shards used per index in this Index Set. Adding Replica Shards improves search performance during parallel reads of the index, such as occurs on dashboards, and is a component of HA and backup strategy. Each Replica Shard set multiplies the storage requirement and memory footprint of the index. This value should not be set higher than the number of search nodes, and typically not higher than 1.

Maximum Number of Segments 1 7
Advanced Option. Maximum number of segments per Search Cluster Index after optimization (force merge). Setting higher values decreases the compression ratio of Index Optimization.

Activate Windows
Go to Settings to activate Windows 8

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.

Rotation & Retention

Data-Tiering Legacy (Deprecated)

30 days | Deleted after 365 days

Max. days in storage 365 1
After how many days your data should be deleted.

Min. days in storage 30 2
How many days at minimum your data should be stored.

Field Type Profile

With index set field type profiles you can bundle up custom field types into profiles. You can assign any profile to this index set. To see and use profile setting for index set, you have to rotate indices.

Index field type mapping profile Select index field type profile

Create index set Create index set Cancel

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

Nous retrouvons maintenant notre indice créé avec les paramètres définis :

The screenshot shows the Graylog interface for managing index sets. At the top, there's a navigation bar with links like Search, Streams, Alerts, Dashboards, Enterprise, Security, System / Indices, and a user icon. Below the navigation, a green banner indicates that the OpenSearch cluster is green with 12 active shards, 0 initializing, 0 relocating, and 0 unassigned.

The main area displays several index sets:

- Default index set**: Stats are disabled by default. Configuration: Index prefix: graylog, Shards: 1, Replicas: 0, Field type refresh interval: 5 seconds, Field type profile: Not set. Rotation strategy: Data Tiering, Max. in storage: 40 days, Min. in storage: 30 days.
- Graylog Events**: Stores Graylog events. Configuration: Index prefix: gl-events, Shards: 1, Replicas: 0, Field type refresh interval: 5 seconds, Field type profile: Not set. Rotation strategy: Data Tiering, Max. in storage: 40 days, Min. in storage: 30 days.
- Graylog System Events**: Stores Graylog system events. Configuration: Index prefix: gl-system-events, Shards: 1, Replicas: 0, Field type refresh interval: 5 seconds, Field type profile: Not set. Rotation strategy: Data Tiering, Max. in storage: 40 days, Min. in storage: 30 days.
- PfSense-logs**: Stats are disabled by default. Configuration: Index prefix: pfSense, Shards: 4, Replicas: 0, Field type refresh interval: 5 seconds, Field type profile: Not set. Rotation strategy: Data Tiering, Max. in storage: 365 days, Min. in storage: 30 days.

A yellow box highlights the PfSense-logs index set. In the bottom right corner, there's a message: "Activate Windows. Go to Settings to activate Windows."

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 :

Editing Stream

Title

Snort 1

A descriptive name of the new stream

Description (Opt.)

Log de Snort 2

What kind of messages are routed into this stream?

Index Set

Pfsense-logs 3

Messages that match this stream will be written to the configured index set.

Remove matches from 'Default Stream' 4

Don't assign messages that match this stream to the 'Default Stream'.

Cancel Update stream

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** » :

The screenshot shows the Graylog Streams interface. At the top, there's a navigation bar with links for Search, Streams, Alerts, Dashboards, Enterprise, Security, and System. Below the navigation is a search bar labeled "Search for streams" and a "Filters" button. The main area is titled "Streams" and contains a table of streams. The table has columns for "Title", "Index Set", "Archiving", "Rules", "Pipelines", "Outputs", "Throughput", "Status", and "Actions". There are 20 rows visible. One row, "Snort", is highlighted with a yellow box and has its "Index Set" value "Pfsense-logs" also highlighted. In the "Actions" column for the "Snort" row, the "Manage Rules" option is highlighted with a yellow box. Other options like "Stop Stream", "Edit stream", "Manage Outputs", "Manage Alerts", "Set as startpage", "Clone this stream", and "Delete this stream" are also listed.

Nous renseignons les paramètres suivants :

New Stream Rule

Field

source **1**

Type

contain **2**

Value

snort **3**

Inverted

Description (Opt.)

Result: source must contain snort

The server will try to convert to strings or numbers based on the matcher type as well as it can.

Take a look at the matcher code on GitHub

Regular expressions use Java syntax.

Cancel Create Rule

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 :

Streams

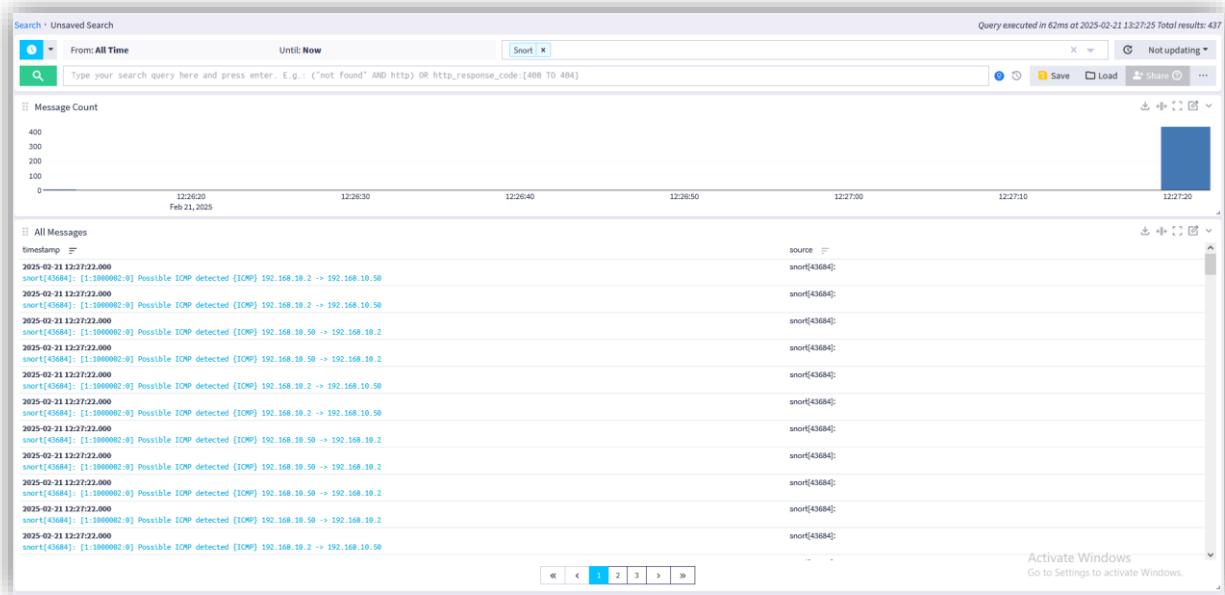
You can route incoming messages into streams by applying rules against them. Messages matching the rules of a stream are routed into it. A message can also be routed into multiple streams.

Create stream

Search for streams Filters Bulk actions

| | Index Set | Archiving | Rules | Pipelines | Outputs | Throughput | Status | Actions |
|--|--|-----------|-------|-----------|---------|------------|---------|-------------------------|
| <input type="checkbox"/> Title | Graylog Events | | | | | 0 msg/s | Running | Data Routing Share More |
| <input type="checkbox"/> All events | Stream containing all events created by Graylog | | | | | 0 msg/s | Running | Data Routing Share More |
| <input type="checkbox"/> All system events | Stream containing all system events created by Graylog | | | | | 0 msg/s | Running | Data Routing Share More |
| <input type="checkbox"/> Default Stream | Default index set | | | | | 0 msg/s | Running | Data Routing Share More |
| <input type="checkbox"/> Snort | Pfsense-logs | | | | | 0 msg/s | Paused | Data Routing Share More |

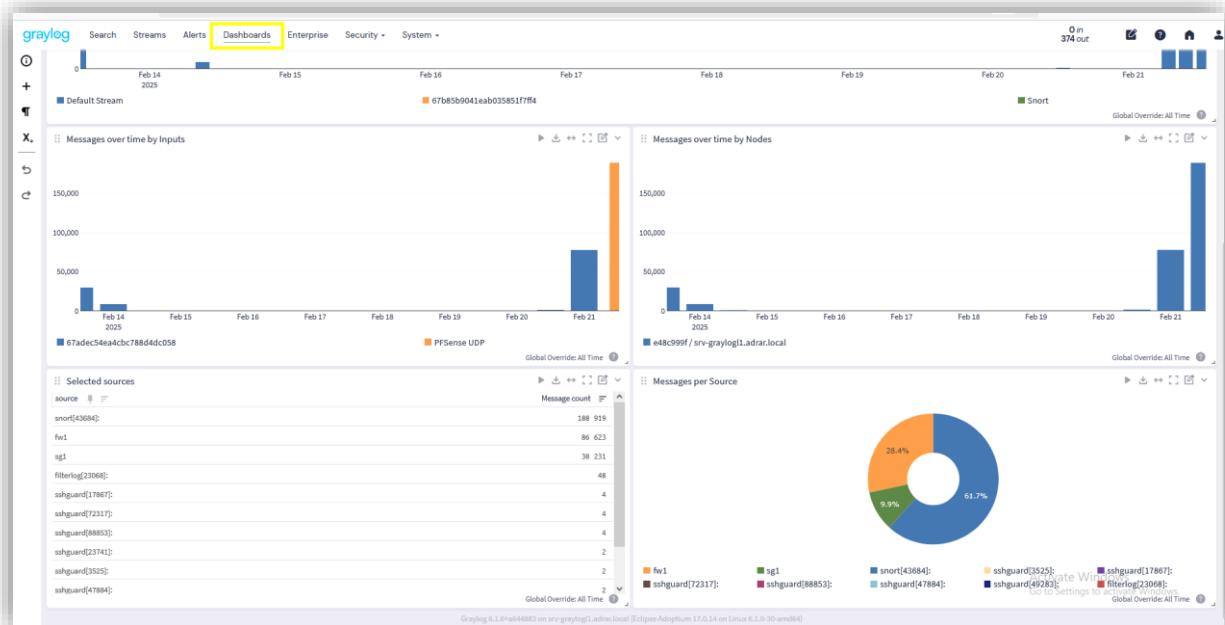
En cliquant sur notre stream nous retrouvons maintenant les logs concernant Snort uniquement :



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 » :

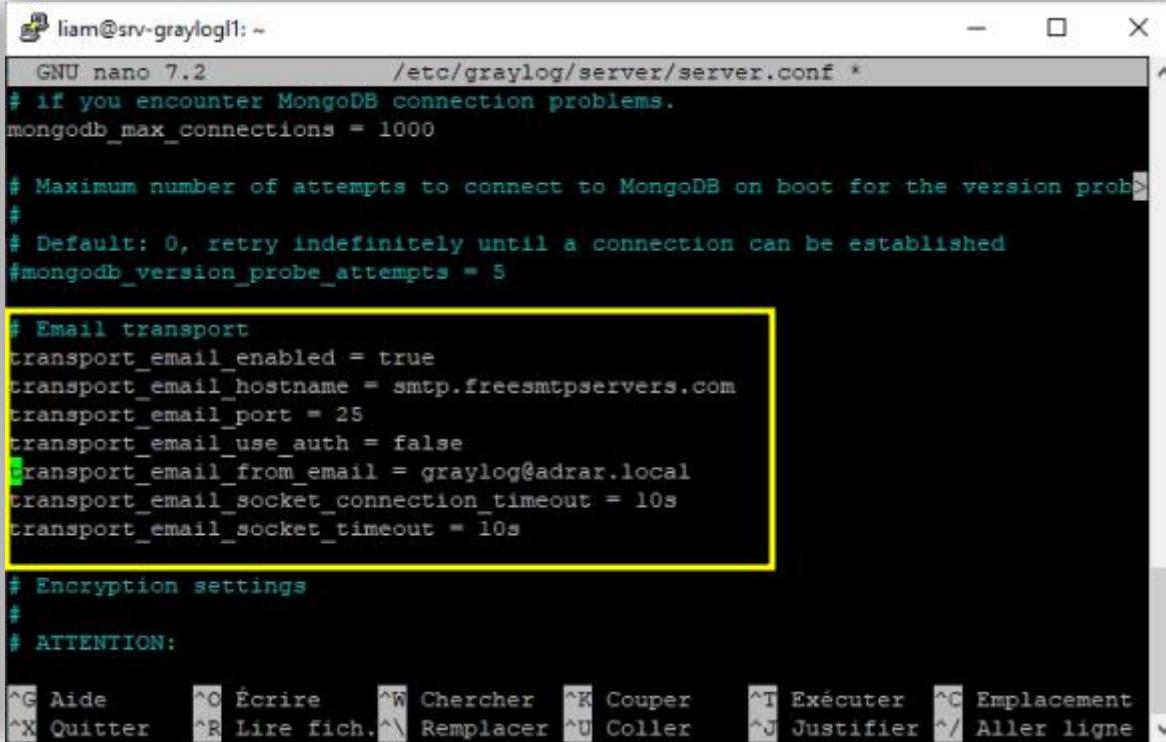


Il est possible de personnaliser ces Dashboards à souhait afin d'affiner les statistiques selon les besoins cependant un certain temps de configuration sera nécessaire afin d'extraire correctement les informations des logs via des regexs et extractors personnalisés.

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 »



```

liam@srv-graylog1: ~
GNU nano 7.2          /etc/graylog/server.conf *
# if you encounter MongoDB connection problems.
mongodb_max_connections = 1000

# Maximum number of attempts to connect to MongoDB on boot for the version probe
#
# Default: 0, retry indefinitely until a connection can be established
#mongodb_version_probe_attempts = 5

# Email transport
transport_email_enabled = true
transport_email_hostname = smtp.freesmtpservers.com
transport_email_port = 25
transport_email_use_auth = false
transport_email_from_email = graylog@adrar.local
transport_email_socket_connection_timeout = 10s
transport_email_socket_timeout = 10s

# Encryption settings
#
# ATTENTION:

^G Aide      ^O Écrire     ^W Chercher   ^K Couper      ^T Exécuter   ^C Emplacement
^X Quitter    ^R Lire fich. ^\ Remplacer   ^U Coller      ^J Justifier  ^/ Aller ligne

```

Maintenant nous allons dans le menu « Alerts → Notifications » puis « create Notifications »



Title
Email Notification 1

Description (Optional)

Notification Type
Email Notification 2

Subject
Graylog event notification: \${event_definition_title} 3

Reply-To (Optional)

The email address that recipients should use for replies.
 Use lookup table for Reply To email

Sender (Optional)

The email address that should be used as the notification sender. Leave it empty to use the default sender address.
 Use lookup table for Sender email

User recipient(s) (Optional)
graylog-sidecar (Sidecar System User (built-in)) 4

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 »

The template that will be used to generate the email body.

The template that will be used to generate the email HTML body.

Test Notification (Optional)

Execute Test Notification

Create notification Cancel

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** »

New Event Definition "Event Snort"

Event Definitions allow you to create Alerts from different Conditions and alert on them.

Event Details

Title: Event Snort 1

Description (Optional):

Remediation Steps (Optional) 2

Priority: Normal 3

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

Configure how Graylog should create Events of this kind. You can later use those Events as input on other Conditions, making it possible to build powerful Conditions based on others.

Condition Type

Choose the type of Condition for this Event.

Filter

Add information to filter the log messages that are relevant for this Event Definition.

Search Query

Search query that Messages should match. You can use the same syntax as in the Search page, including declaring Query Parameters from Lookup Tables by using the `$newParameters$` syntax.

Streams (Optional)

Select streams the search should include. Searches in all streams if empty.

Search within the last

Use Cron Scheduling

Schedule this event with a Quartz cron expression

Execute search every

Enable

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 :

New Event Definition "Event Snort"

Event Definitions allow you to create Alerts from different Conditions and alert on them.

Notifications

Add Notification

Choose Notification

Email Notification

Select a Notification to use on Alerts of this kind or create a new Notification that you can later use in other Alerts.

Add notification Cancel

Next

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, 0000000000000000000000000001] |
| Fields | |

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

| Description | Key | Type | Event Definition | Timestamp |
|--|------|-------|------------------|---------------------|
| Snort_alerts: (Empty Value) - count[] = 1135182.0 ID 01JMSZ9JVFY1K1MHKBYSR4JMY Priority Normal Timestamp 2025-02-23 17:37:12 Event Definition Snort_alerts (Filter & Aggregation) Remediation Steps No remediation steps Actions Replay search ► | none | Event | Snort_alerts | 2025-02-23 17:37:12 |

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 sommes redirigées vers le site de Snort afin de récupérer un token pour télécharger les règles communautaires.

Sign up

Email
Please enter your Email address

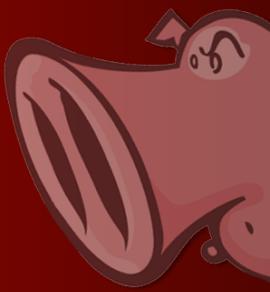
Password

Password confirmation

Agree to [Snort license](#)
 Subscribe to Snort mailing lists?
 Snort-users Snort-sigs Snort-devel Snort-openappid
 For each mailing list chosen, a separate confirmation email will be sent to you.

Je ne suis pas un robot 
reCAPTCHA
Confidentialité + Conditions

[Sign up](#) [Sign in](#) [Didn't receive confirmation instructions?](#)



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

- Account
- Oinkcode**
- Subscription
- Receipts
- False Positive
- Snort License
- Resources

Oinkcode

8c5af779cda106e1fb4dc91e478337f5d4b33c62

[Regenerate](#)

Documentation and Resources

[How to use your oinkcode](#)
Informational and instructional resources for Snort 2 and Snort 3

Snort Subscriber Rules

Enable Snort VRT Click to enable download of Snort free Registered User or paid Subscriber rules

[Sign Up for a free Registered User Rules Account](#)
[Sign Up for paid Snort Subscriber Rule Set \(by Talos\)](#)

Snort Oinkmaster Code

Obtain a snort.org Oinkmaster code and paste it here. (Paste the code only and not the URL!)

Nous activons ensuite les règles communautaires :

Snort GPLv2 Community Rules

Click to enable download of Snort GPLv2 Community rules

The Snort Community Ruleset is a GPLv2 Talos certified ruleset that is distributed free of charge without any Snort Subscriber License restrictions. This ruleset is updated daily and is a subset of the subscriber ruleset.

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

Snort Interfaces Global Settings **Updates** Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Installed Rule Set MD5 Signature

| Rule Set Name/Publisher | MD5 Signature Hash | MD5 Signature Date |
|----------------------------------|--------------------|--------------------|
| Snort Subscriber Ruleset | Not Downloaded | Not Downloaded |
| Snort GPLv2 Community Rules | Not Downloaded | Not Downloaded |
| Emerging Threats Open Rules | Not Enabled | Not Enabled |
| Snort OpenAppID Detectors | Not Enabled | Not Enabled |
| Snort AppID Open Text Rules | Not Enabled | Not Enabled |
| Feodo Tracker Botnet C2 IP Rules | Not Enabled | Not Enabled |

Update Your Rule Set

| Last Update | Result: |
|-------------------|---------|
| Feb-13 2025 11:21 | Success |

Update Rules

Click UPDATE RULES to check for and automatically apply any new posted updates for selected rules packages. Clicking FORCE UPDATE will zero out the MD5 hashes and force the download and application of the latest versions of the enabled rules packages.

Manage Rule Set Log

| View Log | Clear Log |
|--|-----------|
| The log file is limited to 1024K in size and is automatically cleared when that limit is exceeded. | |
| Logfile Size | 222 B |

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

Automatic Flowbit Resolution

Resolve Flowbits If checked, Snort will auto-enable rules required for checked flowbits. Default is Checked.
Snort will examine the enabled rules in your chosen rule categories for checked flowbits. Any rules that set these dependent flowbits will be automatically enabled and added to the list of files in the interface rules directory.

Snort Subscriber IPS Policy Selection

Use IPS Policy If checked, Snort will use rules from one of three pre-defined IPS policies in the Snort Subscriber rules. Default is Not Checked.
Selecting this option disables manual selection of Snort Subscriber categories in the list below, although Emerging Threats categories may still be selected if enabled on the Global Settings tab. These will be added to the pre-defined Snort IPS policy rules from the Snort VRT.

Select the rulesets (Categories) Snort will load at startup

| Enable | Ruleset: Snort GPLv2 Community Rules | Enable | Ruleset: Snort Text Rules | Enable | Ruleset: Snort SO Rules | Snort OPENAPPID rules are not enabled. |
|--------------------------|---|--------------------------|------------------------------|--------------------------|--------------------------------|--|
| <input type="checkbox"/> | Snort GPLv2 Community Rules (Talos certified) | <input type="checkbox"/> | snort_app-detect.rules | <input type="checkbox"/> | snort_browser-chrome.so.rules | |
| <input type="checkbox"/> | emerging-activex.rules | <input type="checkbox"/> | snort_attack-responses.rules | <input type="checkbox"/> | snort_browser-ie.so.rules | |
| <input type="checkbox"/> | emerging-attack_response.rules | <input type="checkbox"/> | snort_backdoor.rules | <input type="checkbox"/> | snort_browser-other.so.rules | |
| <input type="checkbox"/> | emerging-botcc.portgrouped.rules | <input type="checkbox"/> | snort_bad-traffic.rules | <input type="checkbox"/> | snort_browser-webkit.so.rules | |
| <input type="checkbox"/> | emerging-botcc.rules | <input type="checkbox"/> | snort_blacklist.rules | <input type="checkbox"/> | snort_exploit-kit.so.rules | |
| <input type="checkbox"/> | emerging-chat.rules | <input type="checkbox"/> | snort_botnet-cnc.rules | <input type="checkbox"/> | snort_file-executable.so.rules | |
| <input type="checkbox"/> | emerging-ciarmy.rules | <input type="checkbox"/> | snort_browser-chrome.rules | <input type="checkbox"/> | snort_file-flash.so.rules | |
| <input type="checkbox"/> | emerging-compromised.rules | <input type="checkbox"/> | snort_browser-firefox.rules | <input type="checkbox"/> | snort_file-image.so.rules | |
| <input type="checkbox"/> | emerging-current_events.rules | <input type="checkbox"/> | snort_browser-ie.rules | <input type="checkbox"/> | snort_file-java.so.rules | |
| <input type="checkbox"/> | emerging-deleted.rules | | | | | |

Buttons: Select All, Unselect All, Save

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

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

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

The screenshot shows a log entry from the 'All Messages' search results. The log details an event where Snort detected an attempt to scan port 1824 on host 192.168.10.50, which was classified as an 'Attempted Information Leak'. The event was triggered by an incoming UDP packet on port 18682 from host 192.168.10.2. The log includes metadata such as timestamp, source (FW1), application name (snort), facility (security/authorization), facility number (4), level (1), and message content. The message content is identical to the log entry above. The interface also shows standard Graylog controls like Permalink, Copy ID, Copy message, Show surrounding messages, and Test against stream.

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-pfsense-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 personnalisées : <https://cavatari.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