Sommaire

1. Introduction

Dans un contexte numérique en constante évolution, les menaces informatiques sont de plus en plus sophistiquées et fréquentes. Les entreprises et les organisations doivent mettre en œuvre des solutions de sécurité fiables pour protéger leurs infrastructures réseaux, leurs données sensibles et assurer la continuité de leurs activités. C'est dans cette optique que notre projet intitulé « Sécurisation Réseau avec IPFire » s'inscrit. Il vise à concevoir, déployer et tester un système de protection réseau basé sur le pare-feu IPFire, capable de détecter et bloquer les activités malveillantes, tout en assurant une gestion fine du trafic réseau.

Le projet consiste à mettre en place une architecture réseau sécurisée autour de la distribution IPFire. Cette solution open source joue le rôle de pare-feu principal de l'infrastructure, intégrant à la fois des fonctionnalités de routage, de filtrage de paquets, de détection et de prévention d'intrusions (IDS/IPS) via Suricata. Le projet a été conçu dans un environnement virtualisé , simulant un réseau d'entreprise avec plusieurs zones de sécurité : une DMZ, un LAN interne, kali machine pour l'analyse comportementale des attaques, et une passerelle IPFire centrale pour contrôler et surveiller le trafic.

Le projet vise principalement à :

1. Mettre en place une solution de pare-feu robuste et évolutive grâce à IPFire.
2. Segmenter le réseau en différentes zones de sécurité pour limiter la propagation des menaces.
3. Intégrer un système de détection et de prévention d'intrusions (IDS/IPS) avec Suricata pour analyser en temps réel le trafic réseau.
4. Enregistrer les activités suspectes et générer des alertes via des logs détaillés.
5. Simuler des attaques simples (ex. : scan Nmap) pour tester l'efficacité des règles de filtrage et de détection.
6. Fournir une interface d'administration conviviale permettant un suivi et une gestion simplifiés.

2. Présentation de l'Architecture Réseau

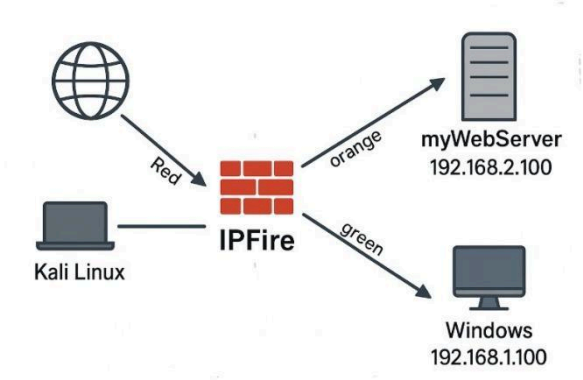
2.1 Présentation de l'Architecture Réseau

Dans le cadre de ce projet de sécurisation réseau, l'architecture déployée repose sur une segmentation classique en zones de sécurité, facilitée par IPFire. Cette séparation permet de mieux contrôler les flux, limiter les risques et surveiller efficacement les interactions entre les machines.

1. Description des zones

- Zone RED (réseau non sécurisé)
  Représente l'extérieur, c'est-à-dire Internet ou un réseau simulé d'attaque. Toute communication venant de cette zone est considérée comme non fiable.
  → Exemple : Kali Linux (machine attaquante) est placée dans la zone RED.
- Zone GREEN (réseau interne sécurisé)
  Représente le réseau interne de confiance. Les machines dans cette zone ont un accès complet aux ressources internes.
  → Exemple : Une machine utilisateur interne ou poste administratif.
- Zone ORANGE (DMZ - zone démilitarisée)
  Sert à héberger des services accessibles depuis l'extérieur, comme un serveur Web, tout en le gardant isolé du réseau interne (GREEN).
  → Exemple : Serveur Web Apache.

2. Schéma reseau



3. Plan d'adressage IP détaillé

Zone	Interface IPFire	IP de la machine associée	Rôle
RED	192.168.19.165	192.168.19.x	Kali Linux (attaquant)
ORANGE	192.168.2.1	192.168.2.100	Serveur Web Apache
GREEN	192.168.1.1	192.168.1.100	Poste utilisateur interne

4. Rôle des machines

- Kali Linux (RED) : utilisée pour simuler des attaques (scan Nmap, requêtes malveillantes, etc.) et tester la capacité d'IPFire à détecter et bloquer les comportements suspects.
- Serveur Web (ORANGE) : héberge un service HTTP (Apache) accessible depuis RED. Sert de cible dans les scénarios de test d'attaque.
- Poste interne (GREEN) : machine utilisateur placée en zone de confiance, non exposée aux attaques externes.

3. Installation de IPFire et et des autres machines

Cette section décrit les étapes de création de la machine virtuelle IPFire, l'installation du système, ainsi que la configuration réseau initiale.

Note : Certaines étapes basiques comme le choix de la langue et du clavier ont été personnalisées selon la langue du système et peuvent être ajustées à vos préférences.

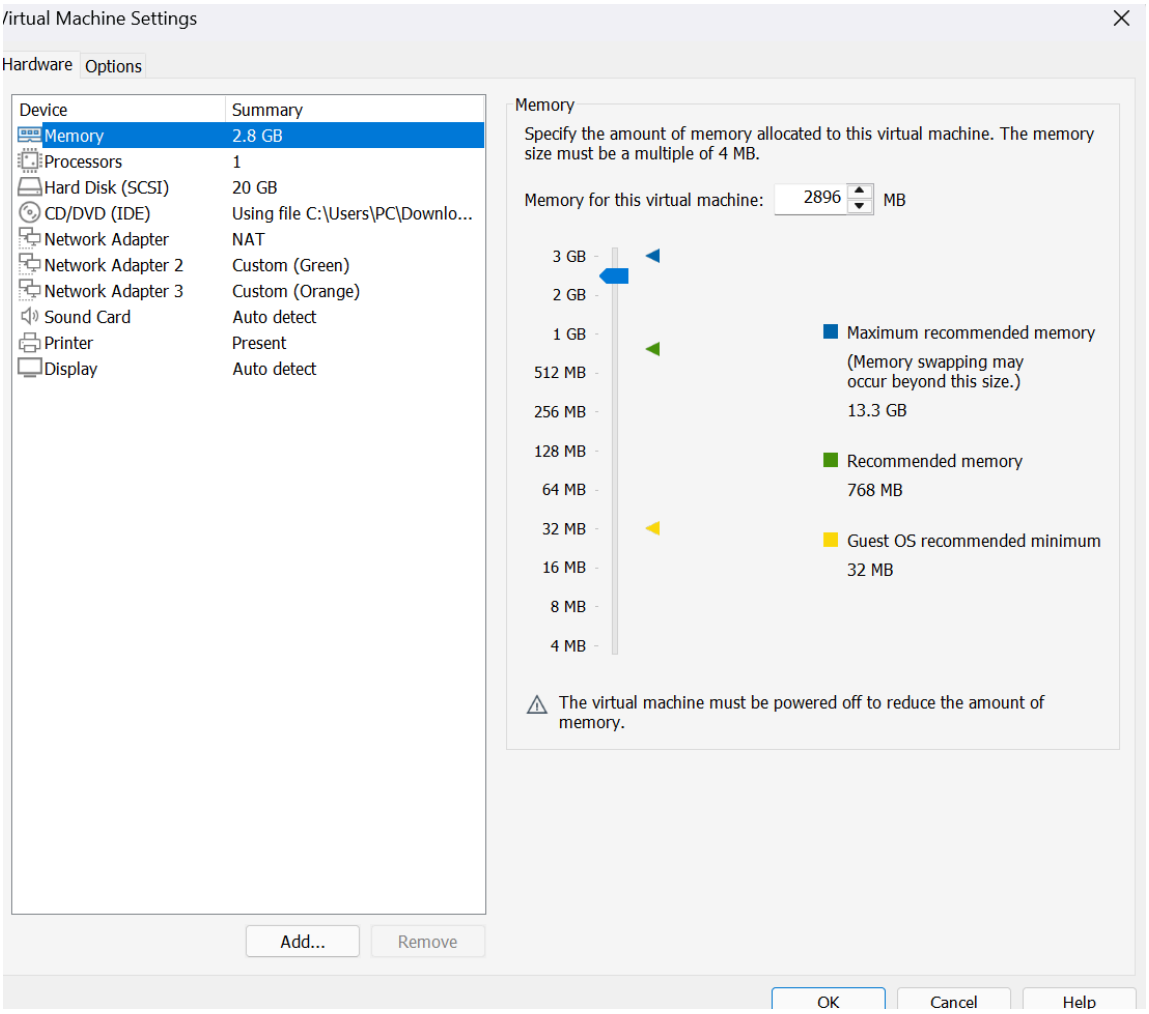
3.1 Création de la machine virtuelle (VM IPFIRE)

Avant l'installation d'IPFire, il est nécessaire de créer une machine virtuelle adaptée à notre architecture.

Étapes de création avec Vmware :

- Choisir l'assistant personnalisé (Custom).
- Laisser la compatibilité matérielle par défaut.
- Sélectionner l'option « J'installerai le système d'exploitation plus tard ».
- Choisir comme système invité : Linux / Debian (64-bit).
- Nommer la machine virtuelle (ex. : ipfire-vm).

- Laisser le nombre de processeurs proposé par défaut.
- Attribuer au moins 512 Mo de mémoire vive (selon les capacités de la machine physique).
- Créer un disque dur virtuel (VDI) d'au moins 10 Go, en allocation dynamique.
- Ajouter le fichier ISO d'IPFire comme lecteur optique virtuel.
- Configurer trois adaptateurs réseau :
    - RED : en mode NAT ou Bridge (pour accès Internet)
    - ORANGE : en réseau interne (intnet1, par exemple)
    - GREEN : en réseau interne (intnet2)

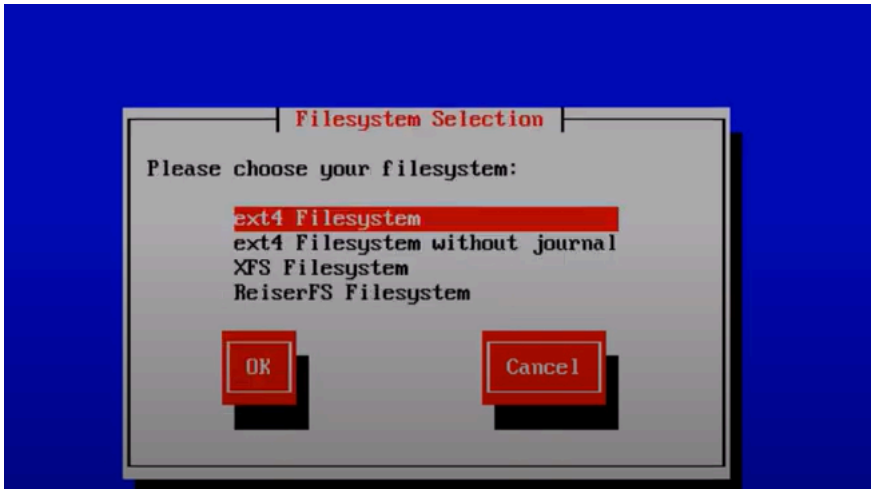


3.2 Installation du système IPFire

Une fois la machine virtuelle créée et configurée, on peut procéder à l'installation d'IPFire :

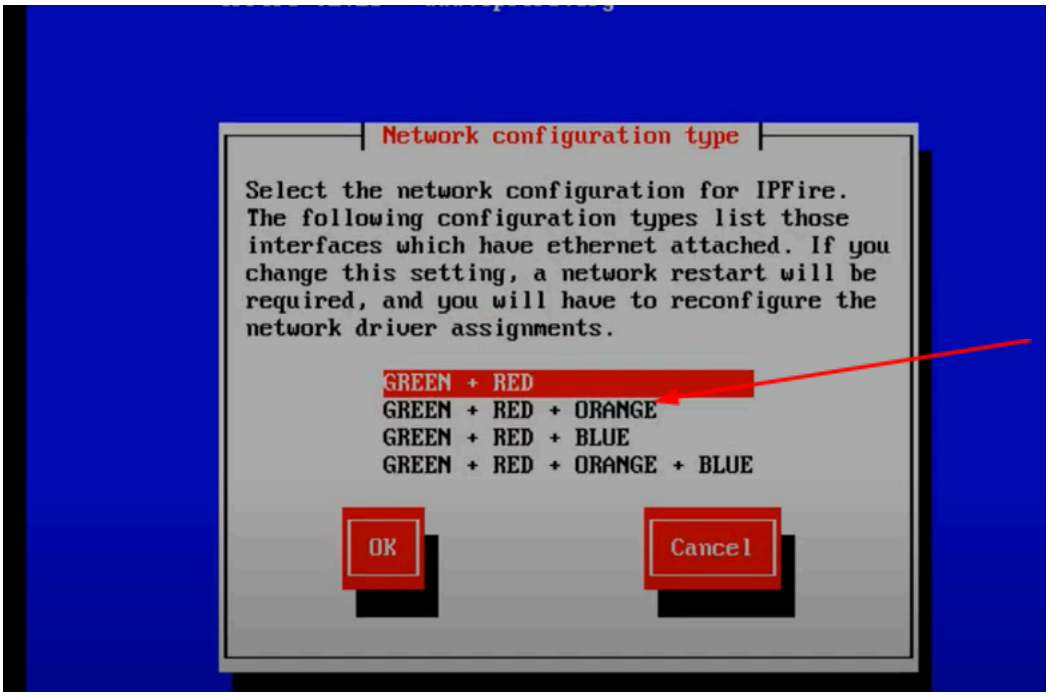
- Démarrer la machine avec l'ISO inséré.
- Appuyer sur Entrée pour lancer l'installation.

- Valider les messages d'avertissement (partitionnement).
- Accepter le système de fichiers proposé.

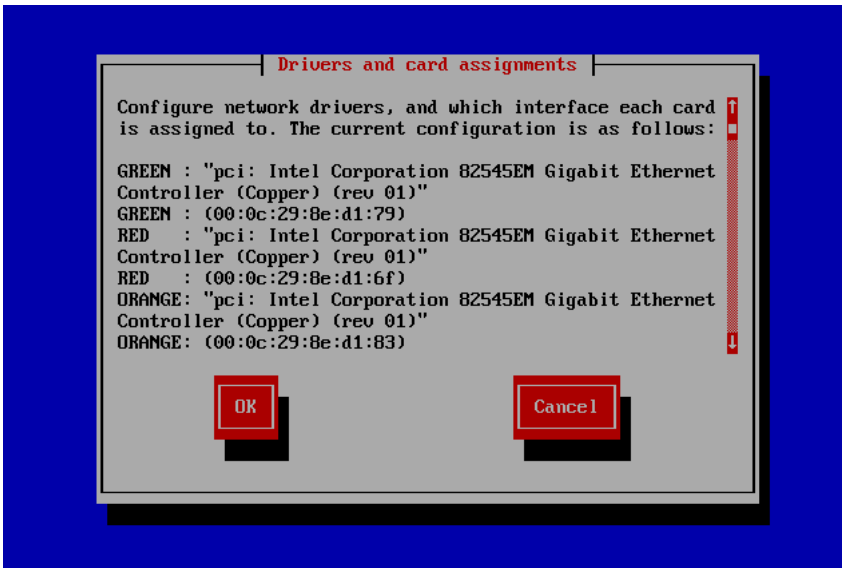


- Créer les mots de passe pour les deux comptes :
    - root : pour l'accès terminal
    - admin : pour l'accès via l'interface web

- Choisir la topologie réseau : RED + ORANGE + GREEN.



- Attribuer les interfaces physiques aux zones :
  - eth0 → GREEN
  - eth1 → ORANGE
  - eth2 → RED

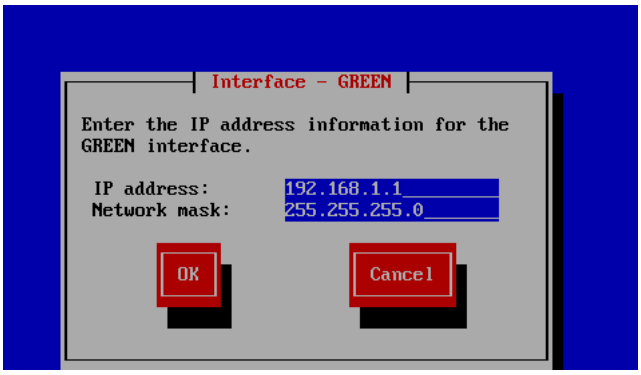


3.3 Configuration initiale réseau

Configuration des adresses IP pour chaque interface :

- RED : configurée en DHCP (selon la configuration NAT/Bridge du réseau)

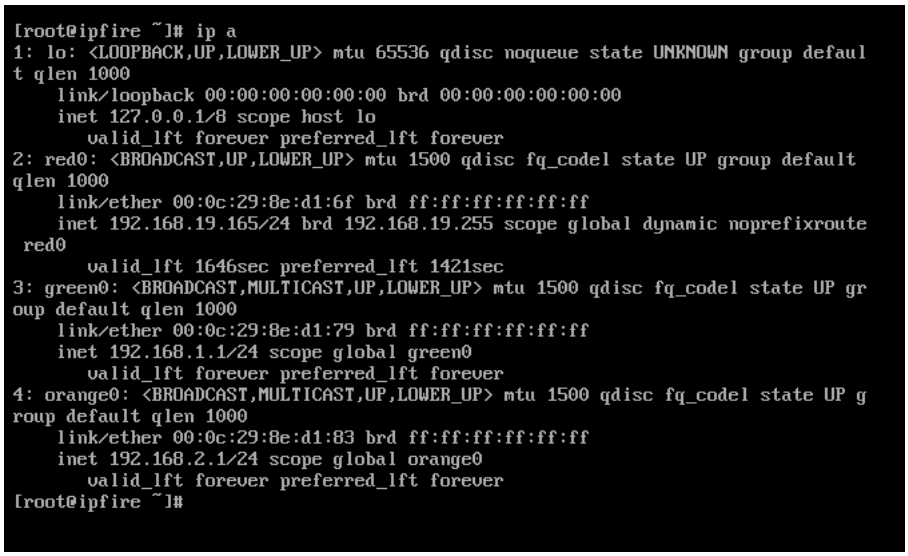
- GREEN : IP statique (ex. : 192.168.1.1/24)



- ORANGE : IP statique (ex. : 192.166.2.1/24)

Test de connectivité :

- Vérifier la configuration avec la commande ifconfig ou ip a

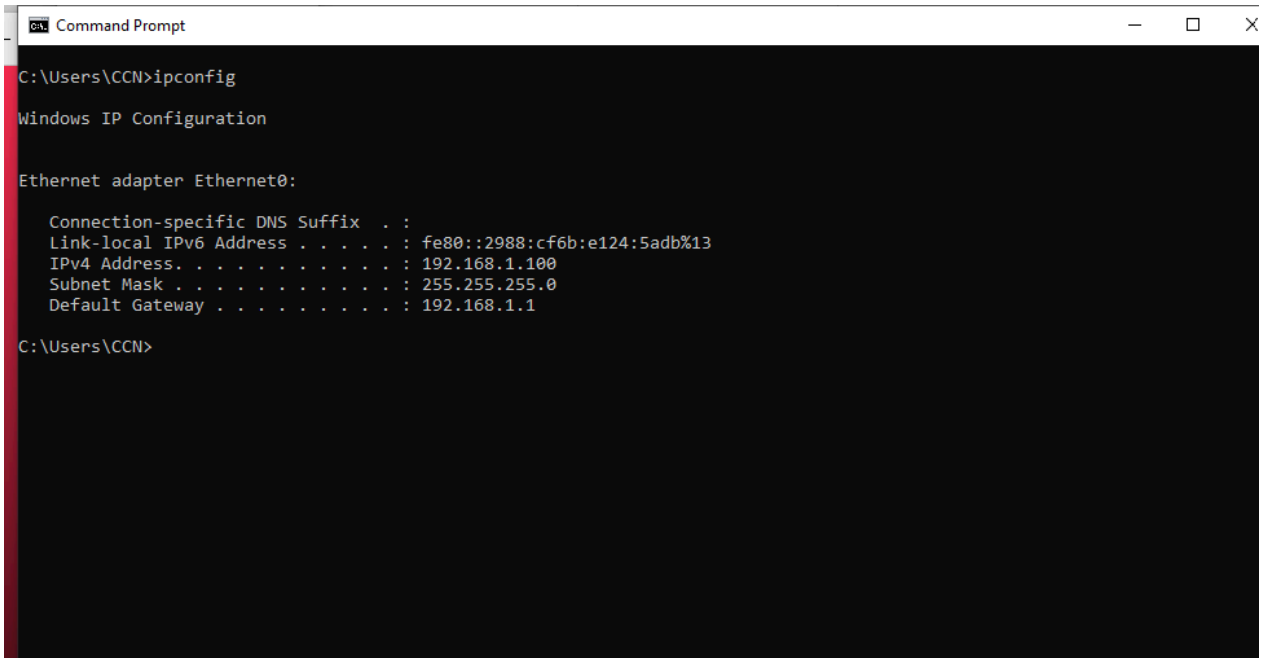


- Accéder à l'interface web via l'IP GREEN (ex. : http://192.168.1.1:444)

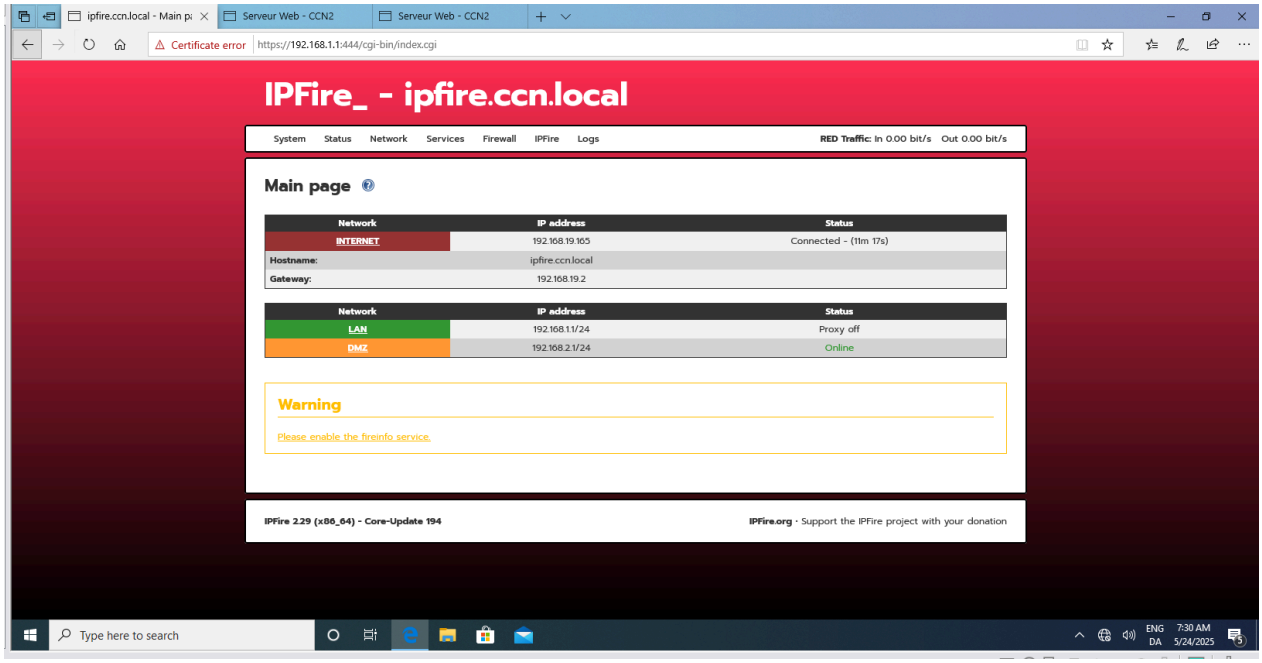
3.3 Création des autres machines virtuelles

3.4.1 Machine Windows (zone GREEN)

- Création d'une VM Windows 10
- Interface réseau : réseau interne « intnet2 » (zone GREEN)
- IP statique : 192.168.1.100/24
- Passerelle : 192.168.1.1 (IPFire)



- But : tester l'accès utilisateur, accéder à Internet via IPFire, accéder à l'interface Web d'administration



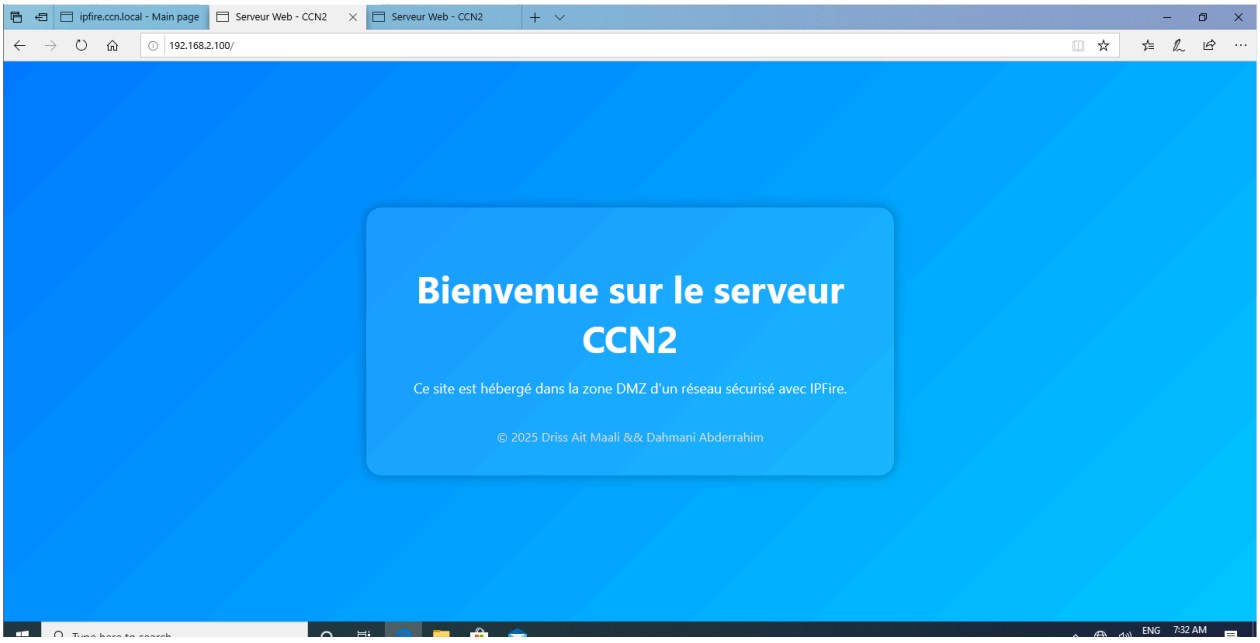
3.4.2 Serveur Linux – DMZ (zone ORANGE)

- Création d'une VM Linux (ex. : Ubuntu Server)

- Interface réseau : réseau interne « intnet1 » (zone ORANGE)
- IP statique : 192.168.2.100/24
- Passerelle : 192.168.2.1 (IPFire)

```
ccn@ccn:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:7b:db:aa brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.2.100/24 brd 192.168.2.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe7b:dbaa/64 scope link
       valid_lft forever preferred_lft forever
ccn@ccn:~$ _
```

- But : servir de serveur web/FTP exposé (future DMZ)



4. Mise en place de Suricata sur IPFire

Dans cette section, nous allons configurer le système de détection/prévention d'intrusions (IDS/IPS) intégré à IPFire, basé sur Suricata. Ce système permet d'analyser le trafic réseau en profondeur et de détecter, voire bloquer, les attaques et comportements suspects.

4.1 Présentation de Suricata

Suricata est un moteur d'analyse de trafic réseau open source qui agit en tant qu'IDS (Intrusion Detection System) et IPS (Intrusion Prevention System). Il inspecte les paquets en temps réel à la recherche de signatures correspondant à des menaces connues.

Fonctionnalités principales :

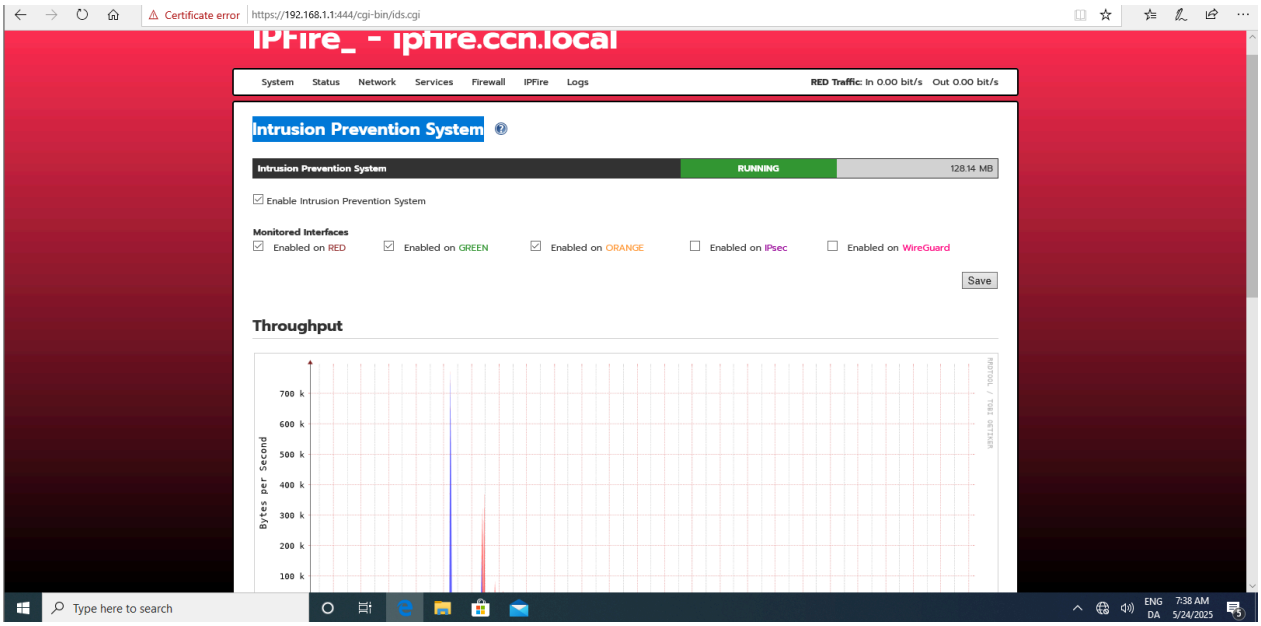
- Inspection profonde des paquets (DPI)
- Détection d'intrusions basée sur des signatures
- Blocage de paquets en mode IPS
- Support multi-thread pour hautes performances

Deux options principales s'offrent à nous :

- Mode "Monitor traffic only" : le trafic est analysé et enregistré, mais non bloqué. Idéal pour le débogage ou les premières phases d'observation.
- Mode "Drop" : Suricata bloque activement les paquets suspects.

Pour activer Suricata sur IPFire :

1. Aller dans l'interface Web : IPFire **firewall → (IPS – Intrusion Prevention System)**
2. Cochez « Activer l'IPS »
3. Sélectionnez la ou les zones à surveiller (ex. : RED, ORANGE)

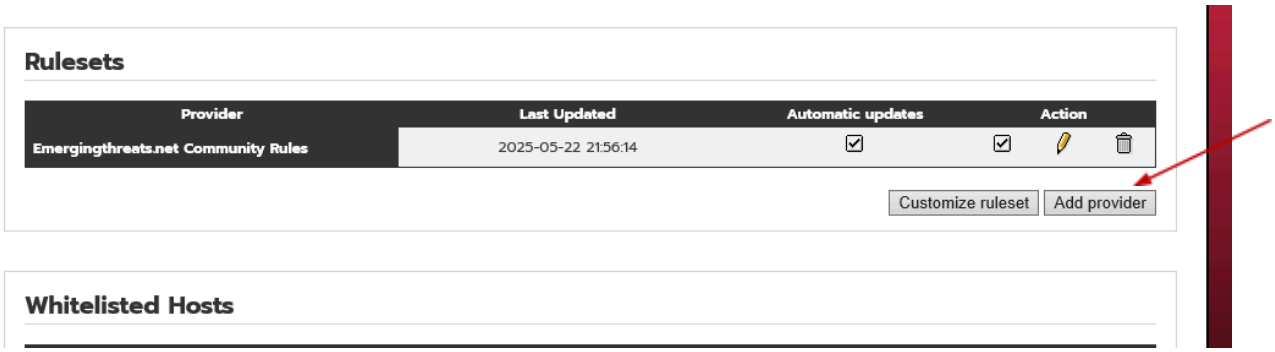


Remarque : Seul le trafic entrant ou sortant d'une zone cochée sera analysé par l'IPS. Le trafic des zones non cochées ne sera pas filtré.

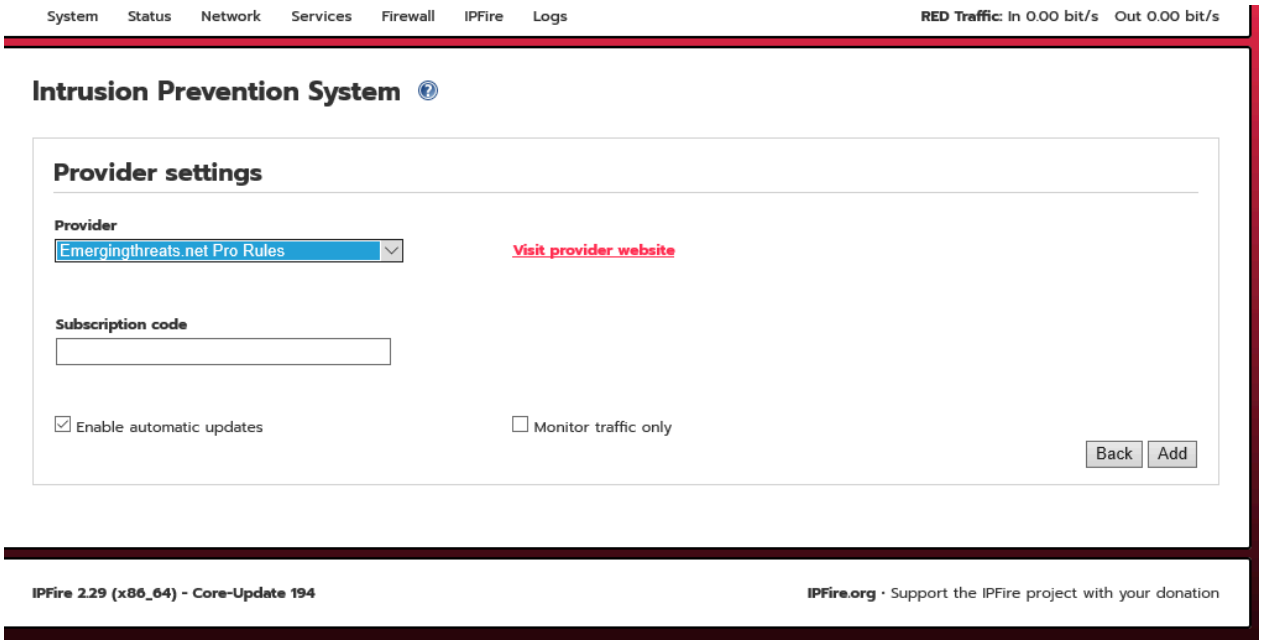
4.2 Configuration des règles

Sur la page de configuration des règles :

1. Appuyez sur « Ajouter un fournisseur »



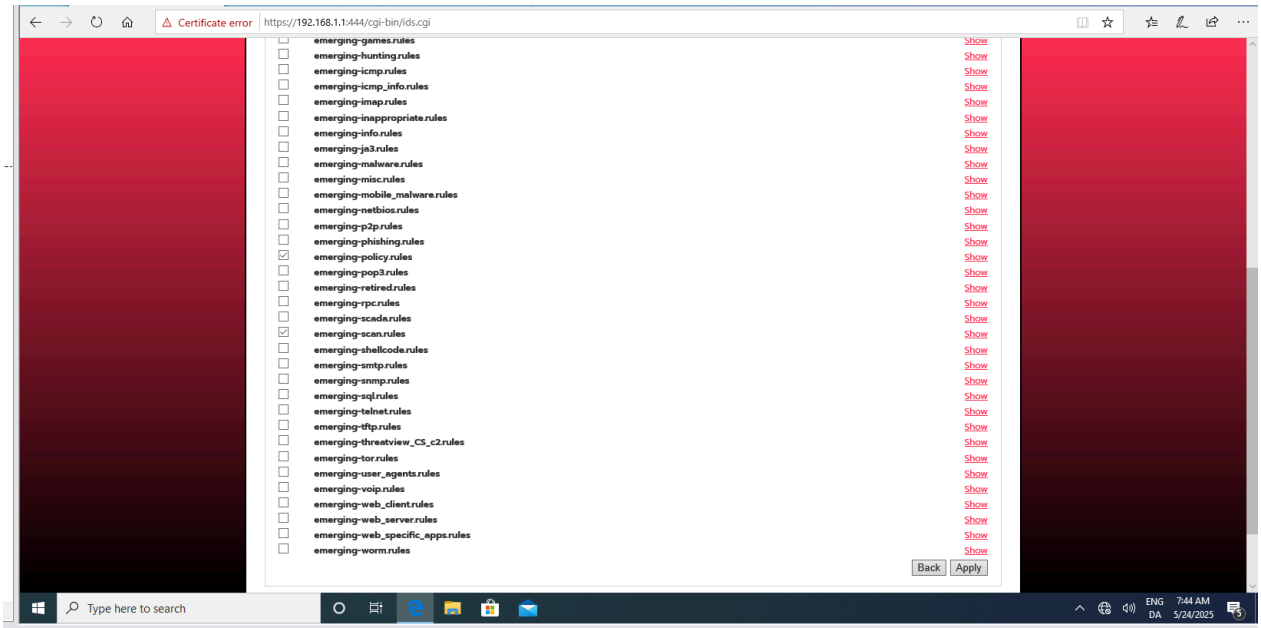
2. Sélectionnez « Emerging Threats » comme fournisseur
3. Activez la mise à jour automatique
4. Cochez ou non « Analyse seulement » (voir plus bas)
5. Cliquez sur « Ajouter »



Cliquez sur « Personnaliser les règles » pour accéder à toutes les signatures disponibles. Cochez les catégories pertinentes selon vos besoins :

Exemples :

- emerging-malware.rules
- emerging-web_client.rules
- emerging-scan.rules
- emerging-policy.rules



5. Configuration des Zones et du Pare-feu

IPFire repose sur une architecture réseau en zones de sécurité distinctes : GREEN (réseau interne sûr), RED (Internet), ORANGE (DMZ), BLUE (Wi-Fi). Le pare-feu joue un rôle central dans la régulation du trafic entre ces zones.

6.1 Création des règles de firewall

Depuis l'interface Web d'IPFire :

- Accéder à « Firewall » → « Règles »
- Cliquer sur « Nouvelle règle »

Pour chaque règle, on peut définir :

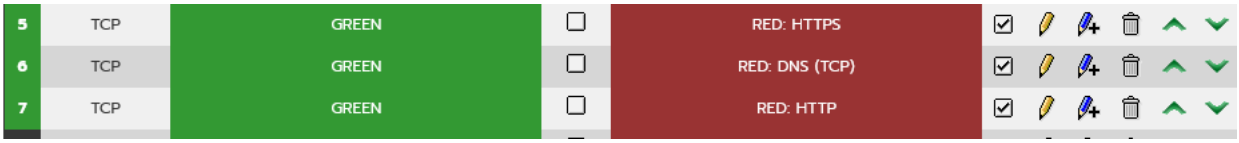
- Source (ex. : GREEN, IP spécifique)
- Destination (ex. : RED, ORANGE)
- Protocole (TCP, UDP, ICMP)
- Port (ex. : 80 pour HTTP)
- Action (Autoriser ou Bloquer)

6.2 Exemple : Autoriser le trafic HTTP de GREEN vers Internet

Objectif : permettre aux utilisateurs internes (zone GREEN) d'accéder au web via le port 80.

Détails :

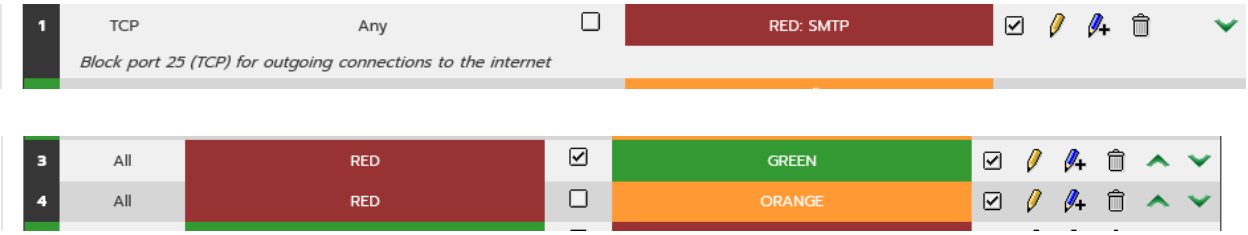
- Source : GREEN
- Destination : RED
- Protocole : TCP
- Port destination : 80 (HTTP)
- Action : Autoriser



6.3 Blocage de comportements suspects

Exemples de protections :

- Blocage des ports non utilisés
- Blocage des connexions entrantes depuis ORANGE vers GREEN
- Blocage des connexions sortantes anormales



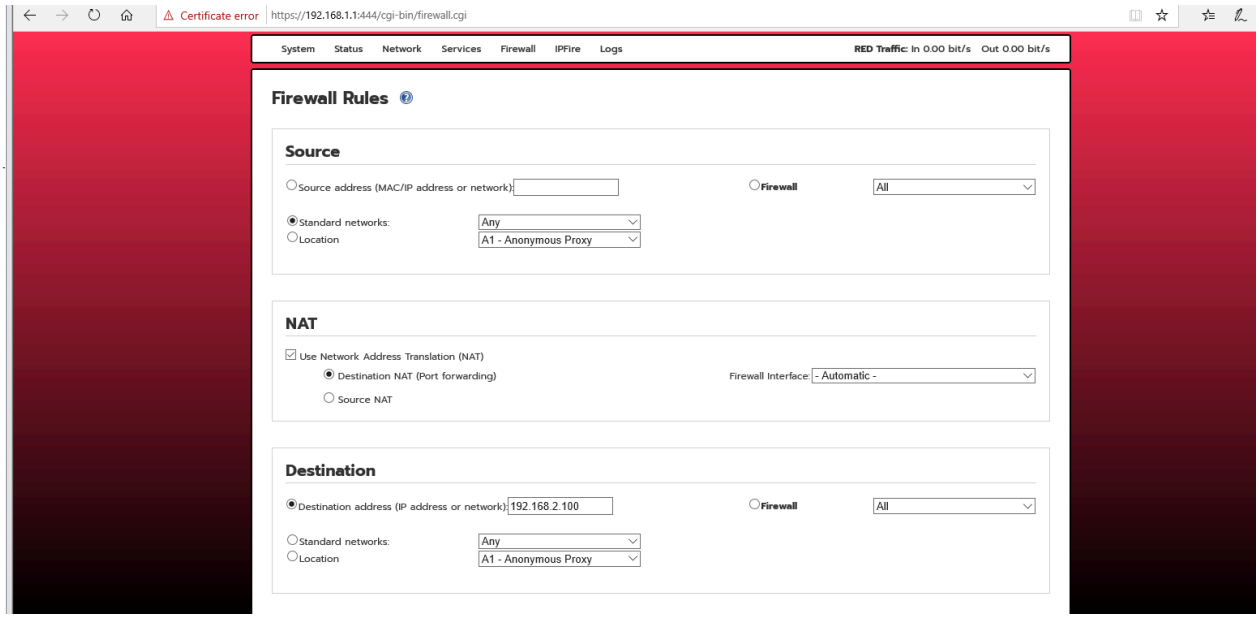
6.5 Redirection de port pour le serveur Web dans la DMZ

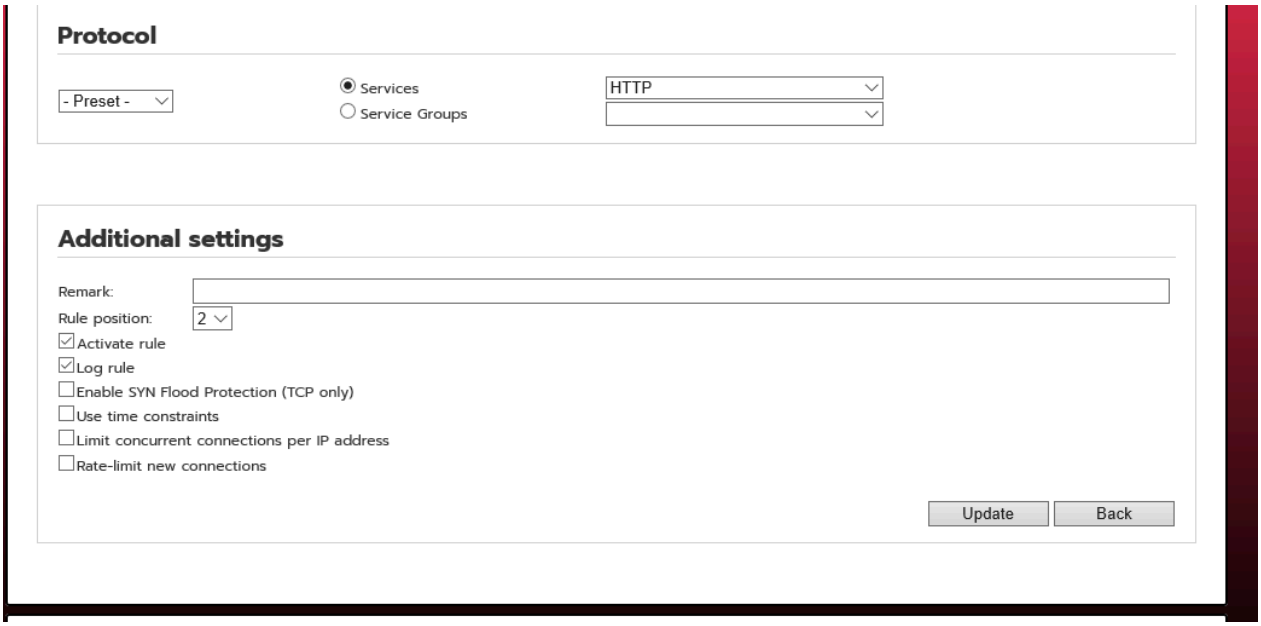
Objectif : rendre le serveur Web (dans la zone ORANGE) accessible depuis l'extérieur (zone RED, ici via la machine Kali).

Exemple : rediriger les requêtes HTTP (port 80) reçues sur l'interface RED vers le serveur Web hébergé dans la zone ORANGE (par exemple 192.168.2.100).

Étapes :

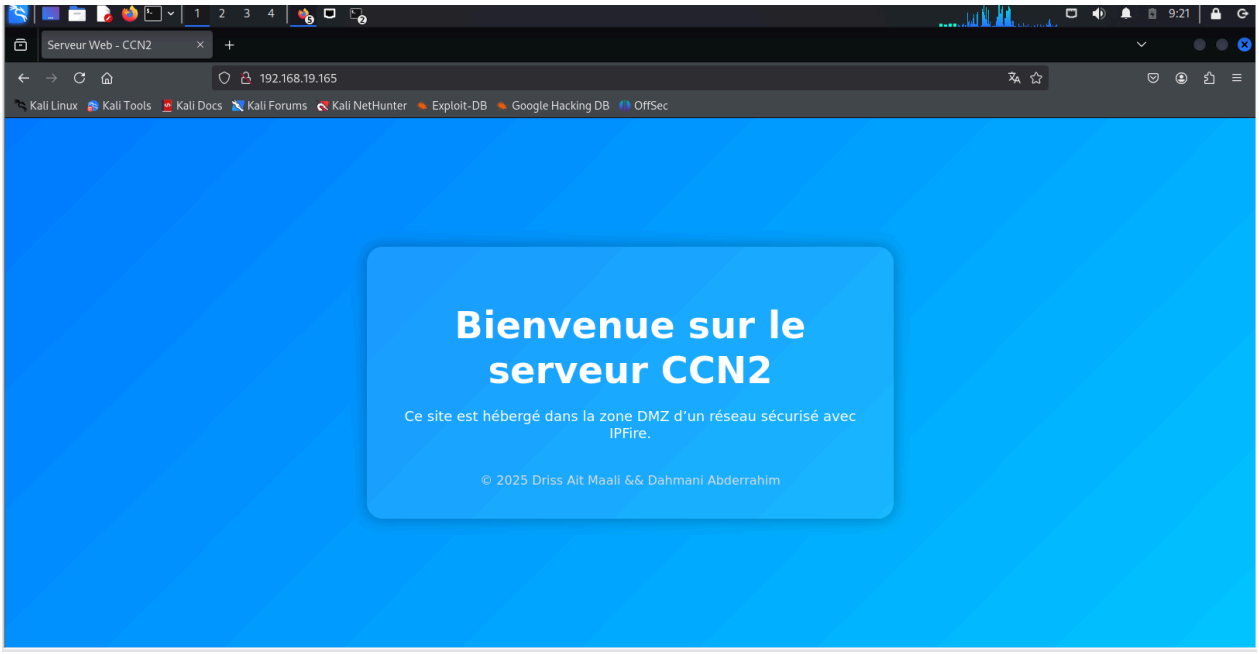
- Interface Web IPFire → Firewall → Port Forwarding
- Créer une nouvelle règle de redirection
    - Source : RED (ou Any si besoin)
    - Port externe : 80
    - Destination IP : 192.168.2.100
    - Port interne : 80
    - Protocole : TCP
- Cocher « Créer automatiquement une règle de pare-feu »





Validation :

- Depuis Kali (placé dans la zone RED), accéder à l'adresse IP publique ou RED d'IPFire (ex. http://10.0.2.1)
- Résultat : affichage de la page Web hébergée sur la machine dans la zone ORANGE



7. Tests de Détection d'Intrusion

7.1 Cas de test – Scan réseau avec Nmap

Nous avons lancé un scan Nmap avec options agressives depuis la machine Kali (RED) vers le serveur Web situé en zone ORANGE (192.168.2.100).

```
└─$ nmap -A -T4 192.168.19.165
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-24 09:26 +01
Nmap scan report for 192.168.19.165
Host is up (0.0021s latency).
Not shown: 999 filtered tcp ports (no-response)
```

Interface IPFire → > Logs → IPS

02:29:51 suricata: fast output device (regular) initialized: fast.log

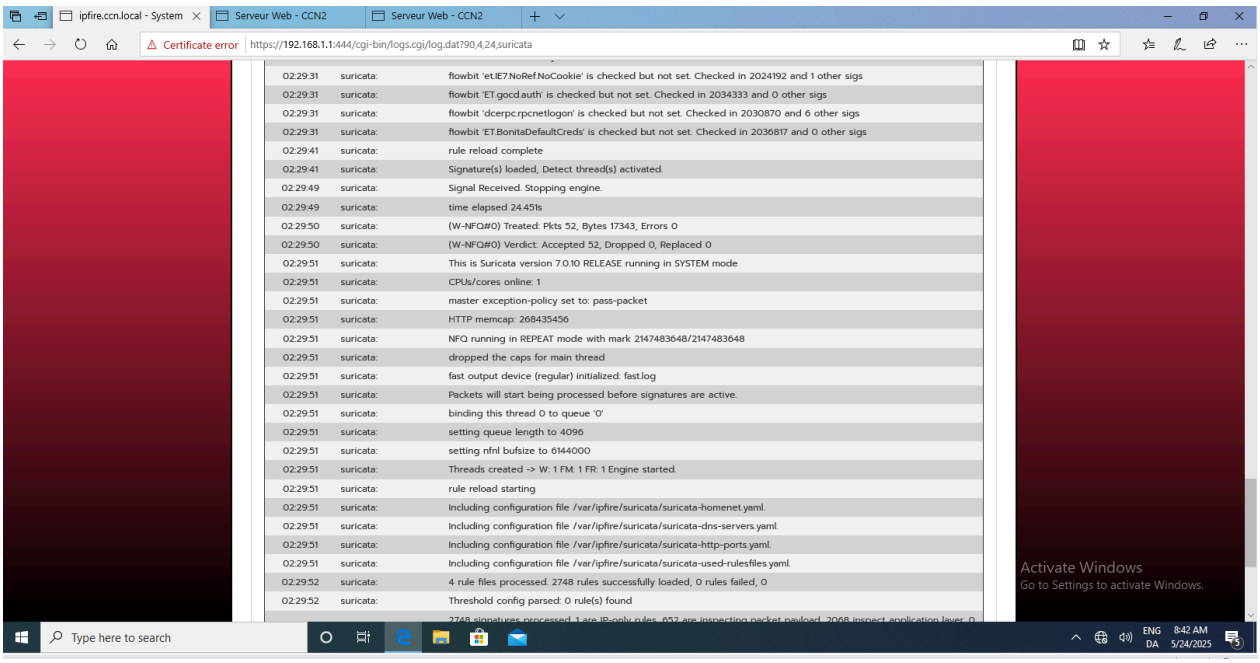
02:29:52 suricata: 2748 signatures processed.

02:29:58 suricata: rule reload complete

02:29:58 suricata: Signature(s) loaded, Detect thread(s) activated.

Ce journal confirme que :

- Les signatures ont bien été chargées (2748 règles).
- Le fichier fast.log est prêt pour enregistrer les alertes.
- Les threads de détection sont actifs et fonctionnels.

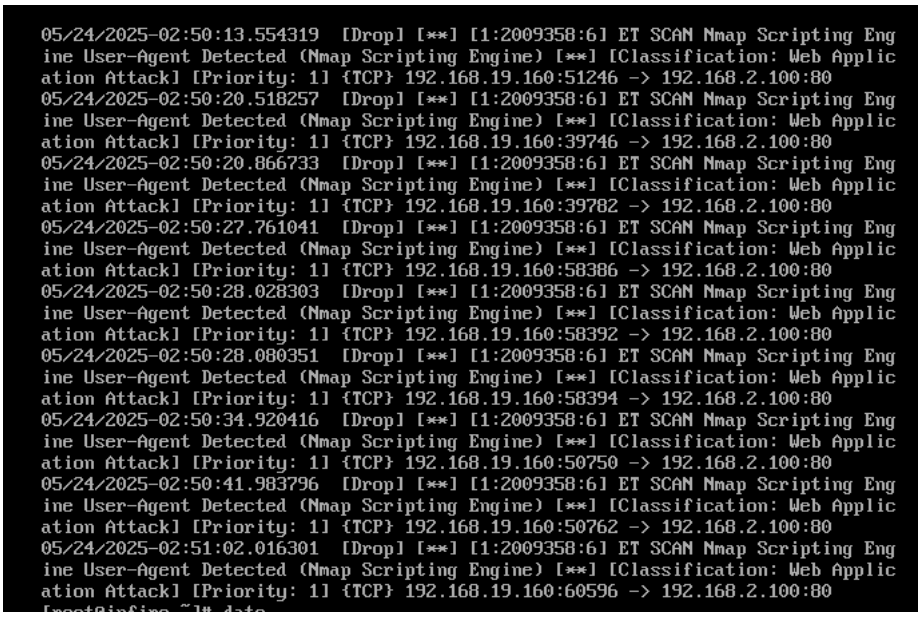


Sur IPFire, Suricata a détecté et bloqué le scan grâce aux signatures intégrées. Voici un extrait du fichier /var/log/suricata/fast.log :

05/24/2025-02:50:13.554319 [Drop] [**] [1:2009358:6] ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine) [**] [Classification: Web Application Attack] [Priority: 1] (TCP) 192.168.19.160:51246 -> 192.168.2.100:80

...

05/24/2025-02:50:02.016301 [Drop] [**] [1:2009358:6] ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine) [**] [Classification: Web Application Attack] [Priority: 1] (TCP) 192.168.19.160:60596 -> 192.168.2.100:80



8. Limitations et Propositions d'Amélioration

8.1 Limitations rencontrées avec IPFire

1. **Performance et ressources**
   IPFire consomme peu de ressources mais Suricata peut devenir exigeant en cas de trafic élevé. L'utilisation sur de petits équipements peut limiter l'efficacité de l'IDS.

2. **Précision des alertes**
   Suricata génère parfois des faux positifs (ex : simples scans détectés comme attaques critiques), ce qui peut surcharger l'analyse.

3. **Visualisation des logs**
   L'interface d'IPFire reste basique : elle n'offre ni tableau de bord dynamique ni corrélation automatisée des événements.

4. **Fonctionnalités SOAR manquantes**
   IPFire ne permet pas de réponses automatiques aux incidents. Il faut analyser et réagir manuellement aux alertes.

8.2 Propositions d'amélioration

1. **Utilisation conjointe avec ELK ou Wazuh**

- Intégrer Suricata avec la stack ELK (Elasticsearch, Logstash, Kibana) pour analyser visuellement les alertes.
- Ajouter Wazuh pour une supervision complète avec gestion des hôtes, détection de malwares, et alertes enrichies.

2. **Mise en place d'un système de corrélation des événements**
   L'intégration de Suricata avec une plateforme comme TheHive permettrait de centraliser, classer et traiter les alertes de manière plus structurée. Cela facilite le travail d'analyse et améliore la réactivité face aux menaces.

3. **Automatisation de la réponse aux incidents**
   En connectant IPFire/Suricata à un moteur d'automatisation comme Shuffle ou Cortex, certaines actions peuvent être déclenchées automatiquement (ex. : blocage d'adresse IP malveillante), réduisant ainsi le temps de réaction et la charge opérationnelle.

Conclusion

La mise en place d'un système de détection d'intrusion avec IPFire et Suricata a permis de démontrer l'efficacité d'une solution open source pour surveiller et sécuriser un réseau en temps réel. Les différents scénarios de tests (scans, attaques web, reverse shell) ont généré des alertes pertinentes, confirmant le bon fonctionnement du système.

Néanmoins, certaines limites comme les faux positifs, la consommation de ressources ou la gestion des logs doivent être prises en compte. Pour aller plus loin, il serait judicieux d'envisager l'intégration d'outils de visualisation ou d'analyse avancée des événements, tout en assurant une mise à jour régulière des règles de détection.

Cette expérience constitue une base concrète pour renforcer la sécurité réseau dans un environnement professionnel.