BTS SIO – SERVICES INFORMATIQUES AUX ORGANISATIONS

OPTION SISR – SOLUTIONS D'INFRASTRUCTURE, SYSTÈMES ET RÉSEAUX

Session : 2024-2026

DOSSIER TECHNIQUE E6 - Projet 2

Déploiement d'un NIDS (Network Intrusion Detection System) sur l'infrastructure Vénus

GOVINDIN Devendra

EDN-CCI ILE DE LA REUNION

TABLES DES MATIÈRES

PROJET 2 - Déploiement d'un NIDS (Network Intrusion Detection System)

I] Présentation du contexte et de l'architecture

Ce projet s'inscrit dans la sécurisation de l'infrastructure Vénus. Il consiste à déployer une sonde de détection d'intrusions réseau (*NIDS - Network Intrusion Detection System*) basée sur Suricata, hébergée sur une machine virtuelle Kali Linux. L'objectif est d'analyser le trafic réseau de l'entreprise afin d'identifier et d'alerter sur d'éventuels comportements malveillants, tout en garantissant l'intégrité de la sonde elle-même.

L'infrastructure réseau est segmentée en plusieurs VLANs virtuels gérés par un hyperviseur Proxmox :

- VLAN 99 *(Management)* : Réseau d'administration sécurisé.
- VLAN 10, 20 et 30 : Réseaux de production *(Administration, Direction, Wi-Fi)* ciblés par la surveillance.

Pour assurer ce rôle, la sonde Kali (*ID Proxmox 104*) est équipée de cinq interfaces réseau virtuelles : une interface d'administration (`eth0`) raccordée au VLAN 99, et quatre interfaces d'écoute (`eth1`, `eth2`, `eth3`, `eth4`) dédiées à la capture du trafic sur les réseaux de l'infrastructure (VLAN 10,20,30 et 99).

II] Phase 1 : Préparation de l'infrastructure et de la sonde

A) Configuration de l'hyperviseur Proxmox (Port Mirroring)

Afin que la sonde puisse analyser le trafic des réseaux de production sans perturber leur fonctionnement, l'hyperviseur Proxmox a été configuré pour dupliquer les paquets transitant sur les commutateurs virtuels (*Port Mirroring*). Les interfaces virtuelles de la sonde ont été raccordées aux différents bridges de l'hyperviseur (`vmbr0`, `vmbr2`, `vmbr3`, `vmbr4`).

B) Configuration réseau des sondes (Kali Linux)

Le bon fonctionnement d'un NIDS nécessite que ses interfaces de capture (`eth1`, `eth2`, `eth3`, `eth4`) opèrent selon un double principe d'écoute globale et d'invisibilité réseau.
Dans un premier temps, les interfaces ont été configurées en mode *`Promiscuous`*. Ce mode matériel force les cartes réseau de la sonde à accepter et lire l'intégralité des trames Ethernet répliquées par l'hyperviseur, y compris celles qui ne leur sont pas destinées.
Dans un second temps, pour que cette écoute s'effectue de manière strictement passive, une architecture réseau furtive a été mise en place via l'outil `NetworkManager(nmcli)`. Les protocoles IPv4 et IPv6 ont été totalement désactivés sur les trois interfaces d'écoute, les privant ainsi de toute adresse IP.

Cette absence d'adressage IP répond à deux objectifs majeurs de conception réseau:

1. **Garantir la non-ingérence** : Les interfaces sondes n'émettent aucun paquet, ne répondent pas aux requêtes ARP et ne créent aucun conflit d'adressage dans les VLANs de production supervisés.
2. **Assurer la sécurité de l'infrastructure de détection** : L'absence d'adresse IP rend la machine Kali virtuellement invisible et inattaquable depuis les réseaux surveillés. Son administration reste exclusivement possible via l'interface `eth0` (`192.168.99.204`), isolée et sécurisée au sein du VLAN 99 (Management).

III] Phase 2 : Déploiement et configuration du NIDS (Suricata)

Suricata est un moteur de détection d'intrusions open-source à hautes performances. Il a été choisi pour sa capacité à analyser le trafic répliqué sur nos interfaces sondes et à journaliser les événements de sécurité.

A) Paramétrage du moteur d'analyse

Après l'installation du service via les dépôts officiels, la configuration principale (`suricata.yaml`) a été adaptée à la topologie de l'infrastructure Vénus :

1. **Définition du périmètre** (`HOME_NET`) : Les sous-réseaux légitimes de la PME *(VLAN 10, 20, 30 et 99)* ont été déclarés comme réseau local à protéger. Tout trafic provenant de l'extérieur est considéré comme `EXTERNAL_NET`.
2. **Assignation des interfaces** (`af-packet`) : Les cinq interfaces réseau de la VM ont été déclarées en mode écoute. Les interfaces sondes (`eth1` à `eth4`) ont explicitement été configurées avec le paramètre `promisc: yes` afin d'appliquer l'écoute globale.

B) Création de règles de détection personnalisées

Bien que Suricata intègre nativement des règles communautaires, il est essentiel de concevoir des règles spécifiques aux contraintes de notre infrastructure. L'ensemble de nos règles métier a été consigné dans un fichier dédié `custom.rules`.

Exemple de règle déployée :

```
alert icmp 192.168.10.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE VENUS : Reconnaissance PING depuis VLAN 10 vers Management !"; itype:8; sid:1000101; rev:1;)
```

Justification : Dans notre infrastructure, un poste de l'administration n'a aucune raison légitime d'envoyer des requêtes ICMP vers les serveurs de production. Ce comportement révèle une potentielle phase de reconnaissance réseau justifiant une remontée d'alerte immédiate.

IV] Phase 3 : Centralisation et visualisation des alertes (Stack ELK)

A) Enjeux et choix d'architecture

Afin de garantir une supervision réactive et exploitable face au volume d'événements réseau, nous avons déployé une stack ELK *(Elasticsearch, Logstash/Filebeat, Kibana)* directement sur la sonde d'analyse.

Cette architecture repose sur trois composants travaillant en synergie :

- Filebeat : Agent léger chargé de la collecte. Il lit en temps réel le fichier `eve.json` généré par Suricata et pré-formate les données grâce à un module natif dédié.
- Elasticsearch : Moteur de base de données et d'indexation. Il réceptionne les événements envoyés par Filebeat et permet des recherches textuelles quasi-instantanées.
- Kibana : Interface de visualisation web connectée à Elasticsearch. Elle permet à l'administrateur de corréler les alertes via des tableaux de bord dynamiques.

B) Déploiement et adaptation à l'environnement de test

Dans le cadre de notre maquette *Single-Node*, la fonction xpack.security d'Elasticsearch a été désactivée afin d'alléger la configuration de test *(les données ne transitant que sur l'interface locale)*. De plus, le service Kibana a été configuré pour écouter exclusivement sur l'adresse IP de l'interface d'administration de la sonde (`192.168.99.204`), garantissant que le tableau de bord n'est exposé que sur le réseau de management sécurisé.

C) Intégration de Suricata et exploitation visuelle

L'activation du module spécifique Suricata au sein de Filebeat a permis l'analyse automatique du format complexe des journaux. Les structures d'indexation et les tableaux de bord ont été importés dans Kibana. L'administrateur peut désormais consulter deux tableaux de bord majeurs :

- **[Filebeat Suricata] Events overview** : Une synthèse globale du trafic réseau capturé, incluant les requêtes DNS et les connexions réseau standards.
- **[Filebeat Suricata] Alert overview :** Un tableau de bord critique affichant exclusivement les alertes de sécurité déclenchées par nos règles personnalisées ou par les règles communautaires.

L'infrastructure de détection et de supervision étant désormais opérationnelle, elle est prête à être soumise à des tests d'intrusion (Phase 4) afin de valider la pertinence de nos règles.

V] Phase 4 : Tests d’intrusion sur le réseau

A) Scénario d'attaque

Dans ce scénario, nous simulons la compromission d’un poste utilisateur interne appartenant au réseau Administration (VLAN 10). L’objectif de l’attaquant est d’effectuer un mouvement latéral vers le cœur de l’infrastructure, en ciblant le serveur critique hébergeant les services essentiels au fonctionnement de l'infrastructure (VLAN 99). La difficulté réside dans l’architecture sécurisée de Vénus : les réseaux sont segmentés et le routage inter-VLANs est théoriquement bloqué par le pare-feu.

B) Présentation de l'environnement offensif

Afin de mener à bien ces simulations sans utiliser les postes de production, une machine virtuelle d'attaque dédiée a été déployée sur l'hyperviseur Proxmox :

- Machine attaquante : Distribution Kali Linux (`VLAN 10, IP : 192.168.10.10`)
- Cible : Windows Server 2025 hébergeant l'Active Directory, le DNS et le DHCP (`VLAN 99, IP : 192.168.99.201`)
- Outil de détection : NIDS Suricata & Interface Kibana

C) Arsenal offensif et outils de validation

Pour évaluer la robustesse du réseau et la réactivité de notre IDS, l'attaquant va dérouler une méthodologie de piratage classique (la Cyber Kill Chain) :

1. Reconnaissance basique (Ping).
2. Scan furtif de vulnérabilités (TCP SYN).
3. Tentative de prise de contrôle à distance (SSH et RDP).
4. Recherche de protocoles alternatifs en cas d'échec (Balayage UDP massif).

D) Validation des règles et visualisation dans Kibana

Test 1 : Ping inter-VLANs (Reconnaissance active)

- Le scénario : L'attaquant cherche à vérifier si le routage réseau lui permet d'atteindre la cible via une requête ICMP Echo.

- La commande exécutée : `ping -c 4 192.168.99.201`

```
┌──(root㉿kaliAttack)-[~]
└─# ping -c 4 192.168.99.201
PING 192.168.99.201 (192.168.99.201) 56(84) bytes of data.

── 192.168.99.201 ping statistics ──
4 packets transmitted, 0 received, 100% packet loss, time 3071ms
```

- Résultat de l'attaque : Comme l'illustre la capture de l'attaquant, la requête échoue totalement (`100% packet loss`). Le pare-fu bloque nativement le trafic inter-VLANs.

- Résultat dans Kibana :

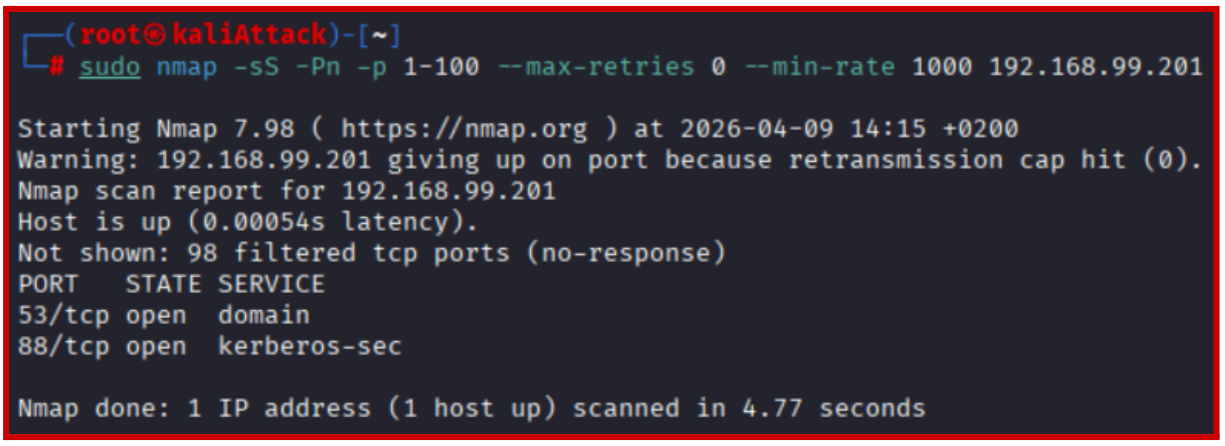
suricata.eve.alert.signature	source.ip	destination.ip	source.port	destination.port	network.transport
ALERTE VENUS : Reconnaissance PING depuis VLAN 10 vers …	192.168.10.10	192.168.99.201	-	-	icmp
ALERTE VENUS : Reconnaissance PING depuis VLAN 10 vers …	192.168.10.10	192.168.99.201	-	-	icmp
ALERTE VENUS : Reconnaissance PING depuis VLAN 10 vers …	192.168.10.10	192.168.99.201	-	-	icmp
ALERTE VENUS : Reconnaissance PING depuis VLAN 10 vers …	192.168.10.10	192.168.99.201	-	-	icmp
ALERTE VENUS : Reconnaissance PING	192.168.10.10	192.168.99.201	-	-	icmp

- Analyse et pertinence métier : La règle s'est déclenchée avec succès. Même si le paquet a été rejeté par la politique de segmentation, Suricata a intercepté son émission. Cette alerte confirme que le cloisonnement fonctionne, mais avertit immédiatement qu'une machine du VLAN 10 tente de sonder le réseau de Management.

Test 2 : Détection de Scans Furtifs (TCP SYN)

- Le scénario : L'attaquant suppose qu'un pare-feu bloque l'ICMP. Il passe à un scan de ports TCP furtif (Half-open) pour découvrir d'éventuels services autorisés à traverser le pare-feu.

- La commande exécutée : `nmap -sS -Pn -p 1-100 --max-retries 0 --min-rate 1000 192.168.99.201`



- Résultat de l'attaque : Nmap indique que 98 ports sont filtrés (bloqués). En revanche, il découvre que les ports 53 (DNS) et 88 (Kerberos) sont ouverts. L'attaquant sait désormais que le pare-feu autorise le trafic vers ces services.

- Résultat dans Kibana :

suricata.eve.alert.signature	source.ip	destination.ip	source.port	destination.port	network.transport
ALERTE VENUS : Scan de ports furtif (SYN) depuis VLAN 10 !	192.168.10.10	192.168.99.201	51579	13	tcp
ALERTE VENUS : Scan de ports furtif (SYN) depuis VLAN 10 !	192.168.10.10	192.168.99.201	51579	81	tcp
ALERTE VENUS : Scan de ports furtif (SYN) depuis VLAN 10 !	192.168.10.10	192.168.99.201	51579	84	tcp
ALERTE VENUS : Scan de ports furtif (SYN) depuis VLAN 10 !	192.168.10.10	192.168.99.201	51579	2	tcp
ALERTE VENUS : Scan de ports furtif (SYN) depuis	192.168.10.10	192.168.99.201	51579	68	tcp

- Analyse et pertinence métier : La règle a fonctionné comme prévu. Même si le pare-feu laisse légitimement passer les requêtes vers le DNS et Kerberos pour le fonctionnement du domaine, Suricata a détecté l'anomalie globale du comportement. Grâce à notre compteur (`threshold: 20 paquets en 60 secondes`), l'alerte offre au SOC une détection fiable d'une phase de

reconnaissance active, avant même que l'attaquant n'exploite les ports découverts.

Test 3.1 : Tentative de connexion SSH (Port 22)

- Le scénario : Cherchant à s'introduire sur le serveur, l'attaquant sonde le port 22 (SSH). N'ayant pas d'identifiants valides, il envoie un paquet SYN pour déterminer si le service d'administration est joignable.

- La commande exécutée : `nmap -sS -Pn -p 22 192.168.99.201`

```
┌──(root㉿kaliAttack)-[~]
└─# nmap -sS -Pn -p 22 192.168.99.201
Starting Nmap 7.98 ( https://nmap.org ) at 2026-04-10 14:11 +0200
Nmap scan report for VENUS-WinServ.venus.local (192.168.99.201)
Host is up.

PORT   STATE    SERVICE
22/tcp filtered ssh

Nmap done: 1 IP address (1 host up) scanned in 2.09 seconds
```

- Résultat de l'attaque : L'outil indique que le port est `filtered`. Le pare-feu bloque effectivement les tentatives de connexion directe depuis le VLAN 10 vers le service d'administration.

- Résultat dans Kibana :

suricata.eve.alert.signature	source.ip	destination.ip	source.port	destination.port	network.transport
ALERTE CRITIQUE : Tentative d'accès SSH vers Management depuis …	192.168.10.10	192.168.99.201	62164	22	tcp
ALERTE CRITIQUE : Tentative d'accès SSH vers Management depuis …	192.168.10.10	192.168.99.201	62162	22	tcp
ALERTE CRITIQUE : Tentative d'accès SSH vers Management depuis …	192.168.10.10	192.168.99.201	50267	22	tcp
ALERTE CRITIQUE : Tentative d'accès SSH vers Management depuis …	192.168.10.10	192.168.99.201	50265	22	tcp
ALERTE CRITIQUE : Tentative d'accès SSH	192.168.10.10	192.168.99.201	48359	22	tcp

- Analyse et pertinence métier : L'alerte CRITIQUE s'est déclenchée instantanément. Une simple tentative d'initialisation TCP vers ce port d'administration depuis un VLAN non autorisé est un marqueur d'intention très fort. Cette alerte permet au SOC de bloquer le poste compromis avant une attaque par force brute.

Test 3.2 : Tentative de connexion RDP (Port 3389)

- Le scénario : N'ayant pas accès au SSH, l'attaquant tente de s'authentifier via le service d'administration Windows (Remote Desktop Protocol).

- La commande exécutée : `nmap -sS -Pn -p 3389 192.168.99.201`

```
┌──(root㉿kaliAttack)-[~]
└─# nmap -sS -Pn -p 3389 192.168.99.201
Starting Nmap 7.98 ( https://nmap.org ) at 2026-04-10 07:12 +0200
Nmap scan report for 192.168.99.201
Host is up.

PORT     STATE    SERVICE
3389/tcp filtered ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 6.58 seconds
```

- Résultat de l'attaque : Le port est identifié comme `filtered`.

- Résultat dans Kibana :

suricata.eve.alert.signature	source.ip	destination.ip	source.port	destination.port	network.transport
ALERTE CRITIQUE : Tentative de prise en main RDP vers …	192.168.10.10	192.168.99.201	44355	3389	tcp
ALERTE CRITIQUE : Tentative de prise en main RDP vers …	192.168.10.10	192.168.99.201	44353	3389	tcp

- Analyse et pertinence métier : La règle identifie formellement une tentative d'accès RDP illégitime, confirmant l'isolement du serveur Windows tout en conservant une trace détaillée pour la remédiation.

Test 4 : Balayage ciblé et massif UDP

- Le scénario : Face à l'échec de ses requêtes TCP, l'attaquant change de stratégie. Il cible spécifiquement les ports UDP des services d'infrastructure vitaux liés à l'Active Directory, espérant que l'un d'eux soit mal sécurisé.

- La commande exécutée : `sudo nmap -sU -Pn -p 53,67,88,123,389 --min-rate 200 192.168.99.201`

```
┌──(root㉿kaliAttack)-[~]
└─# sudo nmap -sU -Pn -p 53,67,88,123,389 --min-rate 200 192.168.99.201
Starting Nmap 7.98 ( https://nmap.org ) at 2026-04-09 13:39 +0200
Nmap scan report for 192.168.99.201
Host is up (0.00064s latency).

PORT    STATE         SERVICE
53/udp  open|filtered domain
67/udp  open|filtered dhcps
88/udp  open          kerberos-sec
123/udp open|filtered ntp
389/udp open          ldap

Nmap done: 1 IP address (1 host up) scanned in 4.86 seconds
```

- Résultat de l'attaque : Contrairement au TCP, Nmap parvient ici à identifier les services UDP du domaine comme `open|filtered`.

- Résultat dans Kibana :

suricata.eve.alert.signature	source.ip	destination.ip	source.port	destination.port	network.transport
ALERTE VENUS : Scan de ports UDP massif depuis VLAN 10 !	192.168.10.10	192.168.99.201	43072	62	udp
ALERTE VENUS : Scan de ports UDP massif depuis VLAN 10 !	192.168.10.10	192.168.99.201	43070	27	udp
ALERTE VENUS : Scan de ports UDP massif depuis VLAN 10 !	192.168.10.10	192.168.99.201	43070	1	udp
ALERTE VENUS : Scan de ports UDP massif depuis VLAN 10 !	192.168.10.10	192.168.99.201	43072	6	udp

- Analyse et pertinence métier : La détection d'un scan UDP massif est complexe car le protocole ne possède pas de mécanisme de connexion. L'attaquant est obligé d'envoyer des paquets "à l'aveugle" rapidement. Le seuil paramétré a permis d'intercepter cette rafale. L'identification de ces ports ouverts offre à l'attaquant des opportunités d'attaques avancées (ex :DHCP Starvation) et ainsi causer un déni de service (attaque DDOS).

VI] Bilan et Conclusion

Bilan du projet

Le déploiement de la sonde NIDS Suricata couplée à la suite de journalisation ELK répond avec succès aux exigences de sécurisation de l'infrastructure Vénus. La phase de tests d'intrusion a démontré que la solution était capable de surveiller l'ensemble des réseaux de l'infrastructure de manière furtive (port mirroring, interfaces sans IP) et d'alerter en temps réel sur des comportements malveillants complexes (scans furtifs TCP SYN, tentatives de mouvement latéral SSH/RDP, balayage massif UDP).

Ce projet m'a permis d'acquérir une vision globale de la cybersécurité défensive : de l'ingénierie réseau (segmentation, routage) à la création de règles de détection métier, jusqu'à la centralisation des journaux (SOC).

ANNEXES

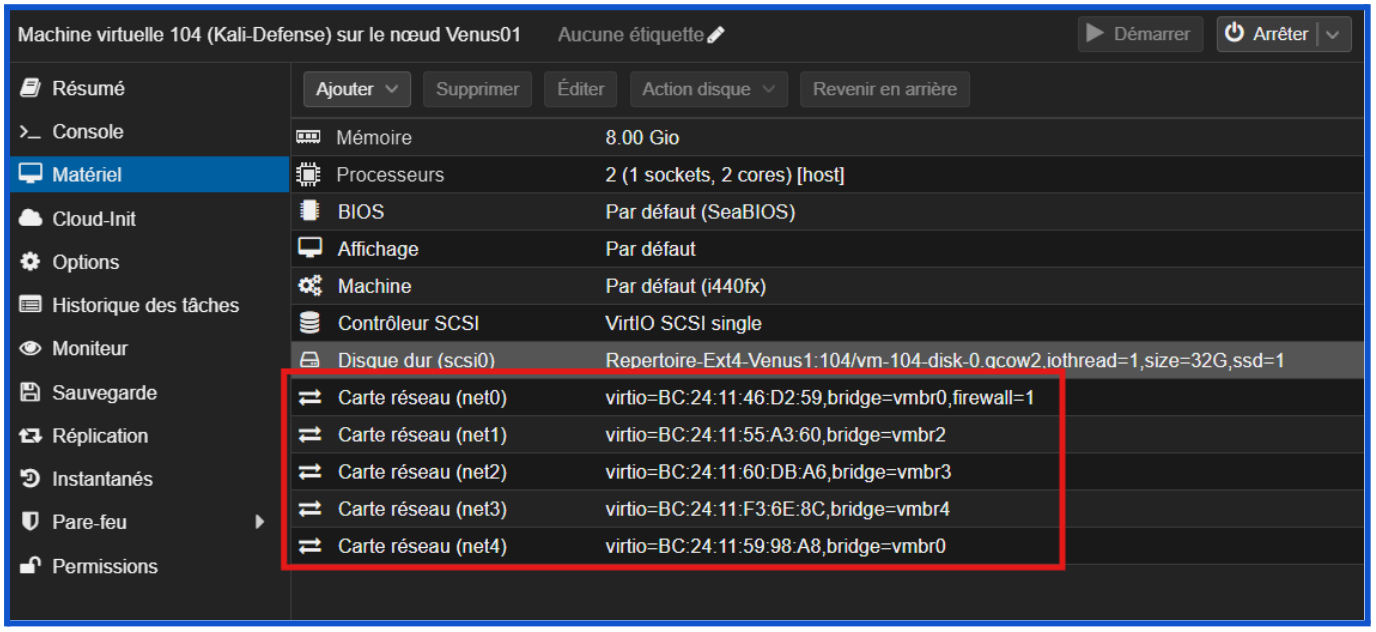
Annexe 1 - Configuration du port Mirroring Proxmox

Pour que la sonde Kali puisse analyser le réseau, l'hyperviseur doit dupliquer les paquets vers elle (Port Mirroring).

1. Ajout des cartes réseaux à la machine virtuelle

Avant de configurer la VM, il faut s'assurer qu'elle possède une interface connectée à chaque VLAN de l'infrastructure.

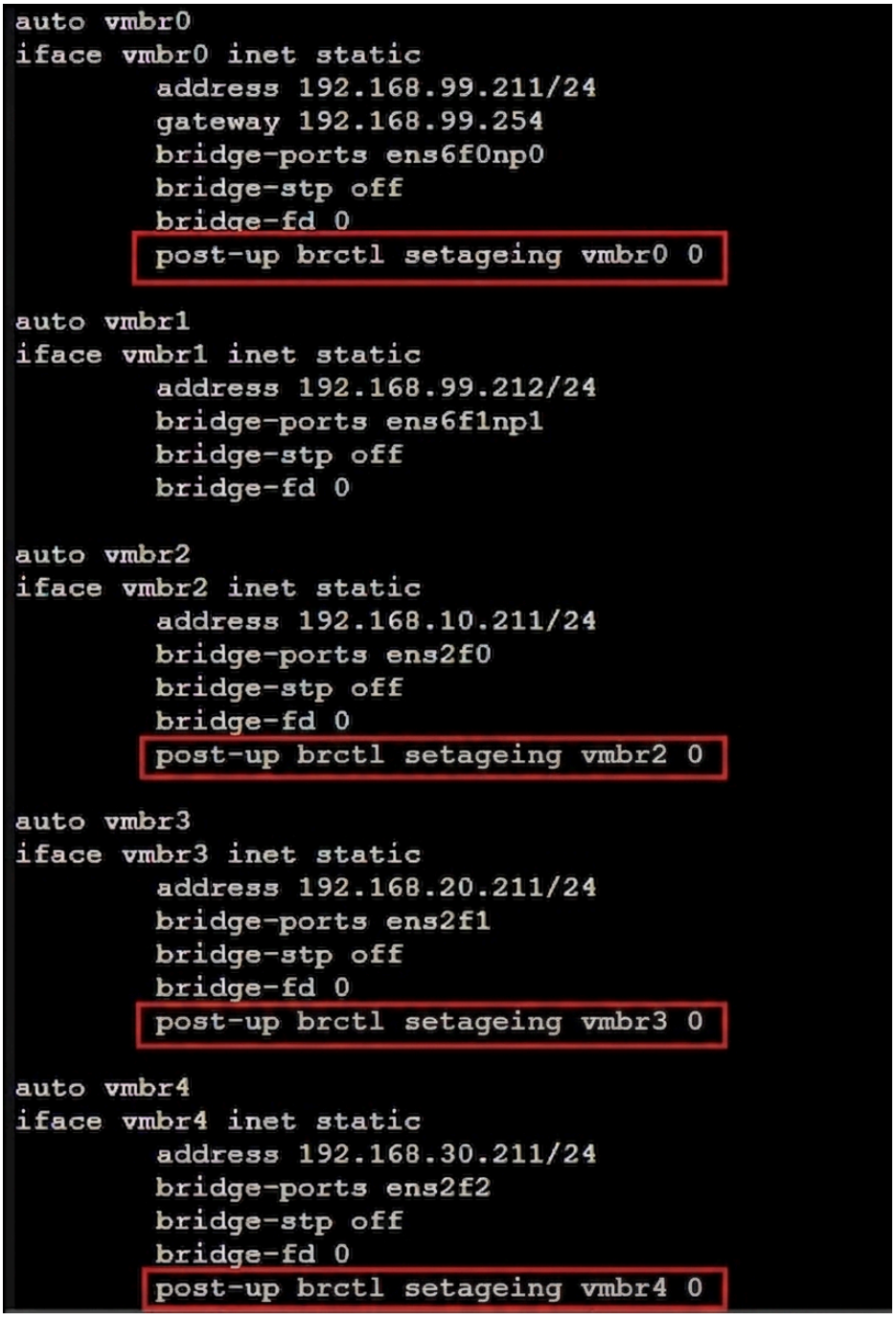
- Connectez-vous à l'interface Web de Proxmox *(ici `192.168.99.211`)*
- Sélectionnez la VM Kali *(ici `ID 104`)* dans le menu de gauche et cliquez sur Hardware *(Matériel)*.
- Cliquez sur Add *(Ajouter)* > Network Device *(Carte Réseau)*.
- Ajoutez les cartes dans cet ordre précis:
    - Carte 1 (`eth0`) : Sélectionnez le bridge `vmbr0` *(Management)*.
      `Note: carte principale de la VM, a sélectionné lors de l'installation de Kali`
    - Carte 2 (`eth1`) : Sélectionnez le bridge `vmbr2` *(Sonde VLAN 10)*.
    - Carte 3 (`eth3`) : Sélectionnez le bridge `vmbr3` *(Sonde VLAN 20)*.
    - Carte 4 (`eth3`) : Sélectionnez le bridge `vmbr4` *(Sonde VLAN 30)*.
    - Carte 5 (`eth4`) : Sélectionnez le bridge `vmbr0` *(Sonde VLAN 99)*.



2. Activation de l'écoute sur les commutateurs virtuels

- Éditez la configuration réseau de l'hôte Proxmox:
  `nano /etc/network/interfaces`

- Ajoutez la ligne `post-up` à la fin de chaque bridge surveillé (`vmbr2`, `vmbr3`, `vmbr4`) :



- Sauvegardez et quittez.

La redirection du trafic vers la sonde s'effectue directement sur le shell de l'hôte Proxmox via l'outil `Traffic Control (tc)`.

Exemple de commande pour rediriger le trafic du VLAN 10 (vmbr2) vers l'interface de la sonde (tap104i1) :

```
tc qdisc add dev vmbr2 ingress
tc filter add dev vmbr2 parent ffff: protocol all u32 match
u32 0 0 action mirred egress mirror dev tap104i1
```

(Cette opération doit être répétée pour chaque bridge surveillé)

Annexe 2 - Configuration des interfaces réseau nmcli

Les cartes d'écoute `(eth1, eth2, eth3,eth4)` doivent être invisibles sur le réseau *(sans adresse IP)* pour ne pas perturber la PME. Toute la configuration réseau sera gérée par l'outil moderne `NetworkManager(nmcli)`.

1.Vérification du fichier d'interfaces historique

Pour garantir que NetworkManager a le contrôle exclusif des cartes réseau, il faut s'assurer que le fichier historique de Debian ne gère aucune interface physique.

- Vérifiez la configuration de base :
  ```
  sudo nano /etc/network/interfaces
  ```

- Assurez-vous que le fichier ne contient que ces deux lignes exactes concernant la boucle locale *(supprimez toute autre référence à `ethX` si elle existe) :*
  ```
  auto lo
  iface lo inet loopback
  ```

- Sauvegardez et quittez. Si vous avez fait une modification, redémarrez la VM avec `sudo reboot`.

2.Configuration des interfaces (nmcli)

Appliquez les profils réseau suivants :

- Interface d'administration (`eth0` - IP fixe) :
  ```
  sudo nmcli connection add type ethernet ifname eth0
  con-name "Management" ipv4.method manual ipv4.addresses
  192.168.99.204/24 ipv4.gateway 192.168.99.254 ipv4.dns
  192.168.99.201 connection.autoconnect yes
  ```

  Note: 192.168.99.204 : Ip de Kali // 192.168.99.254 : Gateway // 192.168.99.201 : Ip

Windows Server (pour DNS)

- Interfaces Sondes (`eth1,eth2,eth3,eth4` - mode silencieux) :
  (Ces commandes activent les interfaces en désactivant totalement les protocoles IP pour qu'elles n'interfèrent pas avec le réseau de la PME).

- Interface `eth1` :
  ```
  sudo nmcli connection add type ethernet ifname eth1
  con-name "Sonde-VLAN10" ipv4.method disabled ipv6.method
  ignore connection.autoconnect yes
  ```

  Interface `eth2` :
  ```
  sudo nmcli connection add type ethernet ifname eth2
  con-name "Sonde-VLAN20" ipv4.method disabled ipv6.method
  ignore connection.autoconnect yes
  ```

  Interface `eth3` :
  ```
  sudo nmcli connection add type ethernet ifname eth3
  con-name "Sonde-VLAN30" ipv4.method disabled ipv6.method
  ignore connection.autoconnect yes
  ```

  Interface `eth4` :
  ```
  sudo nmcli connection add type ethernet ifname eth4
  con-name "Sonde-VLAN99" ipv4.method disabled ipv6.method
  ignore connection.autoconnect yes
  ```

Vérification: Depuis le terminal tapez `ip a`. *Seule `eth0` doit afficher une adresse IP, les autres interfaces sont `UP` sans IP.*

Annexe 3 -Configuration de Suricata

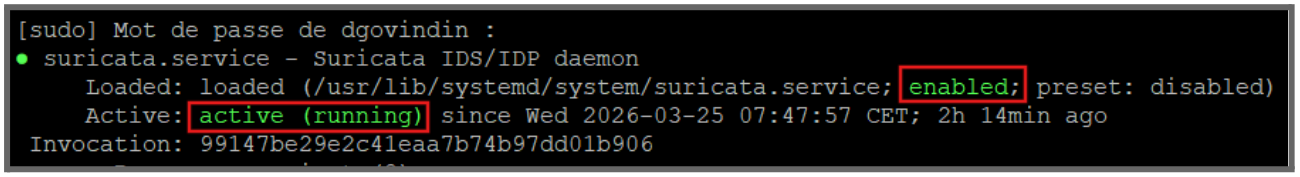
1.Installation et activation du service

- Mettez à jour les dépôts et installez Suricata :
  ```
  #Mise à jour des paquets
  sudo apt update && sudo apt upgrade -y
  #Installation depuis les dépôts officiels
  sudo apt install suricata -y
  #Vérifier la version installée
  suricata --version
  ```

#Activation et vérification de l'état du service

```
sudo systemctl start suricata
sudo systemctl enable suricata
sudo systemctl status suricata
```

Résultat attendu :



2.Configuration du réseau des interfaces d'écoute

Il est indispensable d'indiquer à Suricata quel est le périmètre réseau de la PME à protéger (`HOME_NET`), puis de lui assigner les interfaces physiques virtuelles à écouter. Cela inclut les interfaces de port mirroring (`eth1`, `eth2`, `eth3`) ainsi que l'interface de management principale (`eth0`) pour détecter d'éventuels mouvements latéraux vers les serveurs.

- Éditez le fichier de configuration principal :
  `sudo nano /etc/suricata/suricata.yaml`

- Définition du réseau local protégé (HOME_NET) :

```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.10.0/24,192.168.20.0/24,192.168.30.0/24,192.168.99.0/24]"

    EXTERNAL_NET: "!$HOME_NET"
```

- Configuration des interfaces *(Sondes et Interface VLAN99)* :
  Ajout explicite du paramètre `promisc: yes` pour les 4 interfaces sondes.

```
af-packet:
  - interface: eth0
    cluster-id: 99
    cluster-type: cluster_flow
    defrag: yes
    promisc: no

  - interface: eth1
    cluster-id: 98
    cluster-type: cluster_flow
    defrag: yes
    promisc: yes
```

Configuration identique pour `eth2`, `eth3` et `eth4`

- Sauvegardez et quittez

Annexe 4 - Règles personnalisées Suricata

- Créez et éditez le fichier des règles personnalisées de Suricata :
  `sudo nano etc/suricata/custom.rules`

- Création des règles : Voir *Règles Suricata**

- Sauvegardez et quittez

- Inclusion du fichier dans la configuration principale :
  Il faut maintenant indiquer à Suricata de charger ce nouveau fichier au démarrage: `sudo nano /etc/suricata/suricata.yaml`

- Localisez la directive `default-rule-path` et la section `rule-files:`
  Assurez-vous d'y ajouter votre fichier `custom.rules` :

```
default-rule-path: /var/etc/suricata/suricata.rules

rule-files:
  - suricata.rules
  - /etc/suricata/custom.rules
```

- Sauvegardez et quittez

- Mettez à jour la base de données pour compiler les règles communautaires avec la nouvelle configuration :
  `sudo suricata-update`

- Redémarrez le moteur NIDS pour prendre en compte les nouvelles interfaces, les réseaux déclarés et les règles personnalisées :
  `sudo systemctl restart suricata`

*RÈGLES SURICATA UTILISÉES DANS L'INFRASTRUCTURE

```
# ========================================================================
# FICHIER DE RÈGLES SURICATA - PROJET VENUS
# ========================================================================

# ------------------------------------------------------------------------
# 1. DETECTION DES PHASES DE RECONNAISSANCE (SCANS PING)
# ------------------------------------------------------------------------

alert icmp 192.168.10.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Reconnaissance PING depuis VLAN 10 vers Management !";
itype:8; sid:1000101; rev:1;)

alert icmp 192.168.20.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Reconnaissance PING depuis VLAN 20 vers Management !";
itype:8; sid:1000102; rev:1;)

alert icmp 192.168.30.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Reconnaissance PING depuis VLAN 30 vers Management !";
itype:8; sid:1000103; rev:1;)

# ------------------------------------------------------------------------
# 2. DETECTION DES SCANS FURTIFS DE PORTS (NMAP SYN)
# ------------------------------------------------------------------------

alert tcp 192.168.10.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Scan de ports furtif (SYN) depuis VLAN 10 !"; flags:S;
window:1024; threshold: type threshold, track by_src, count 20,
seconds 60; sid:1000201; rev:1;)

alert tcp 192.168.20.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Scan de ports furtif (SYN) depuis VLAN 20 !"; flags:S;
window:1024; threshold: type threshold, track by_src, count 20,
seconds 60; sid:1000202; rev:1;)

alert tcp 192.168.30.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Scan de ports furtif (SYN) depuis VLAN 30 !"; flags:S;
window:1024; threshold: type threshold, track by_src, count 20,
seconds 60; sid:1000203; rev:1;)
```

```
# ------------------------------------------------------------------------------------------
# 3. DETECTION DES MOUVEMENTS LATERAUX INTER-VLANS (SSH & RDP)
# ------------------------------------------------------------------------------------------
# 3.1 SSH
alert tcp 192.168.10.0/24 any -> 192.168.99.0/24 22 (msg:"ALERTE
CRITIQUE : Tentative d'accès SSH vers Management depuis VLAN 10
!"; flags:S; sid:1000301; rev:1;)

alert tcp 192.168.20.0/24 any -> 192.168.99.0/24 22 (msg:"ALERTE
CRITIQUE : Tentative d'accès SSH vers Management depuis VLAN 20
!"; flags:S; sid:1000302; rev:1;)

alert tcp 192.168.30.0/24 any -> 192.168.99.0/24 22 (msg:"ALERTE
CRITIQUE : Tentative d'accès SSH vers Management depuis VLAN 30
!"; flags:S; sid:1000303; rev:1;)

# 3.2 RDP
alert tcp 192.168.10.0/24 any -> 192.168.99.0/24 3389 (msg:"ALERTE
CRITIQUE : Tentative de prise en main RDP vers Management depuis
VLAN 10 !"; flags:S; sid:1000401; rev:1;)

alert tcp 192.168.20.0/24 any -> 192.168.99.0/24 3389 (msg:"ALERTE
CRITIQUE : Tentative de prise en main RDP vers Management depuis
VLAN 20 !"; flags:S; sid:1000402; rev:1;)

alert tcp 192.168.30.0/24 any -> 192.168.99.0/24 3389 (msg:"ALERTE
CRITIQUE : Tentative de prise en main RDP vers Management depuis
VLAN 30 !"; flags:S; sid:1000403; rev:1;)

# ------------------------------------------------------------
# 4. DETECTION DES SCANS UDP MASSIFS
# ------------------------------------------------------------
alert udp 192.168.10.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Scan de ports UDP massif depuis VLAN 10 !"; dsize:0;
threshold: type threshold, track by_src, count 20, seconds 60;
sid:1000501; rev:1;)

alert udp 192.168.20.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Scan de ports UDP massif depuis VLAN 20 !"; dsize:0;
threshold: type threshold, track by_src, count 20, seconds 60;
sid:1000502; rev:1;)

alert udp 192.168.30.0/24 any -> 192.168.99.0/24 any (msg:"ALERTE
VENUS : Scan de ports UDP massif depuis VLAN 30 !"; dsize:0;
threshold: type threshold, track by_src, count 20, seconds 60;
sid:1000503; rev:1;)
```

Annexe 5 - Déploiement Stack ELK

1.Configuration d'Elasticsearch (Moteur de base de données)

- Éditez la configuration principale d'Elasticsearch :
  ```
  sudo nano /etc/elasticsearch/elasticsearch.yml
  ```

- Recherchez et modifiez le composant `xpack.security` pour le désactiver totalement :
  ```
  xpack.security.enabled: false

  xpack.security.http.ssl:
    enabled: false
  ```

- Sauvegardez le fichier et activez le service au démarrage
  ```
  sudo systemctl enable --now elasticsearch
  ```

3.Configuration de Kibana (Data Visualisation)

- Éditez la configuration de l'interface Web :
  ```
  sudo nano /etc/kibana/kibana.yml
  ```

- Décommentez *(retirez le #)* et modifiez les deux directives suivantes :
  ```
  server.host: "192.168.99.204"
  elasticsearch.hosts: ["http://localhost:9200"]
  ```

- Sauvegardez et lancez le service :
  ```
  sudo systemctl enable --now kibana
  ```

4.Intégration de Filebeat (Liaison avec Suricata)

- Activez le module natif Suricata dans Filebeat :
  ```
  sudo filebeat modules enable suricata
  ```
- Éditez la configuration du module pour lui indiquer précisément l'emplacement du fichier journal:
  ```
  sudo nano /etc/filebeat/modules.d/suricata.yml

  - module: suricata
    eve:
      enabled: true
      var.paths: ["/var/log/suricata/eve.json"]
  ```

  Sauvegardez et quittez

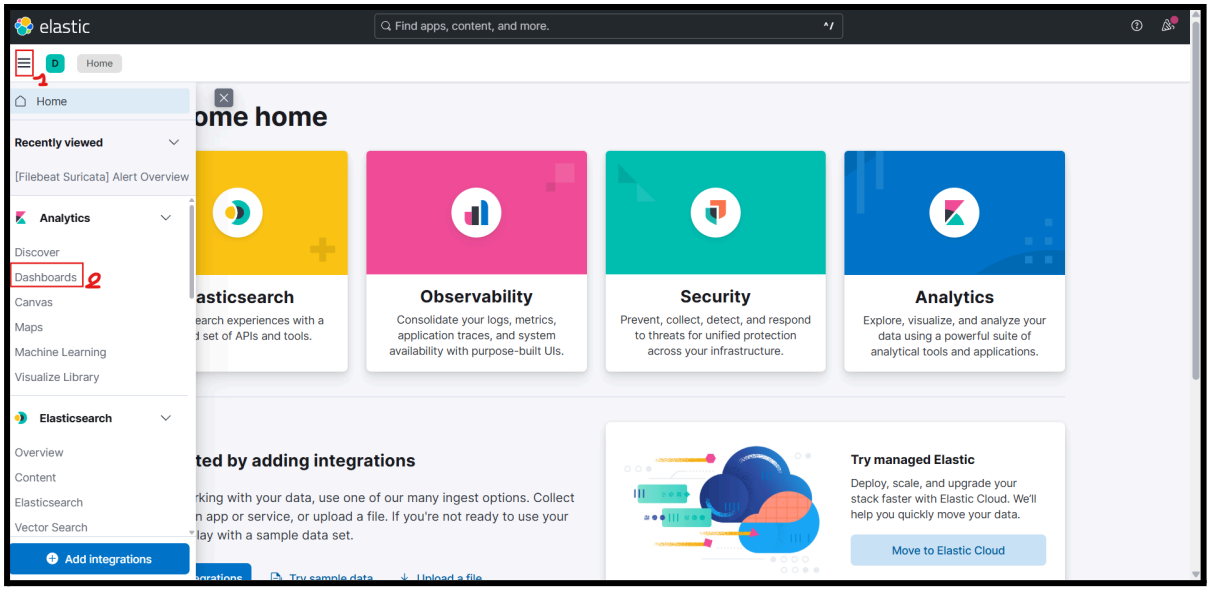
- Chargement des Tableaux de bord : Demandez à Filebeat de se connecter à Elasticsearch pour y créer automatiquement les structures d'indexation et importer les tableaux de bord "Suricata" prêts à l'emploi dans Kibana :
  `sudo filebeat setup`

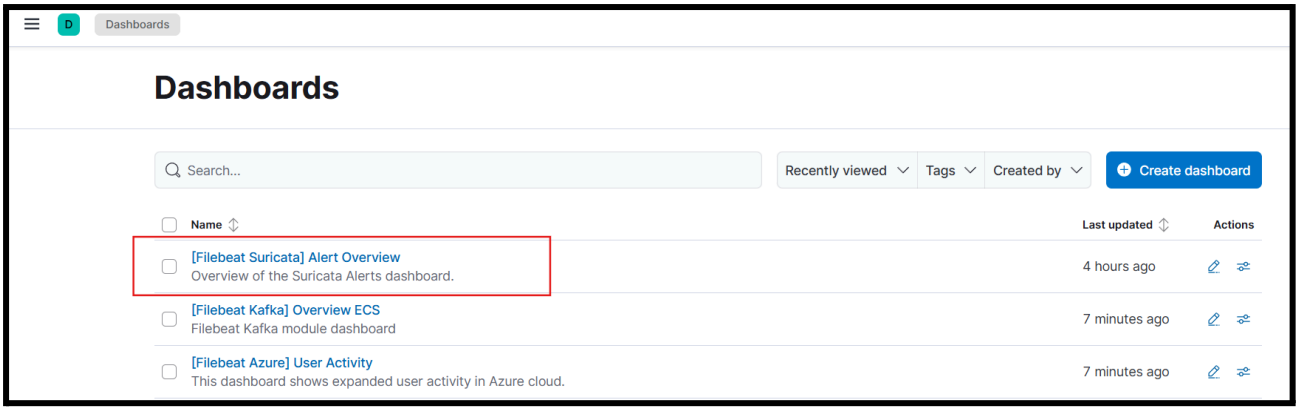
- Lancez le service d'expédition de logs en temps réel :
  `sudo systemctl enable --now filebeat`

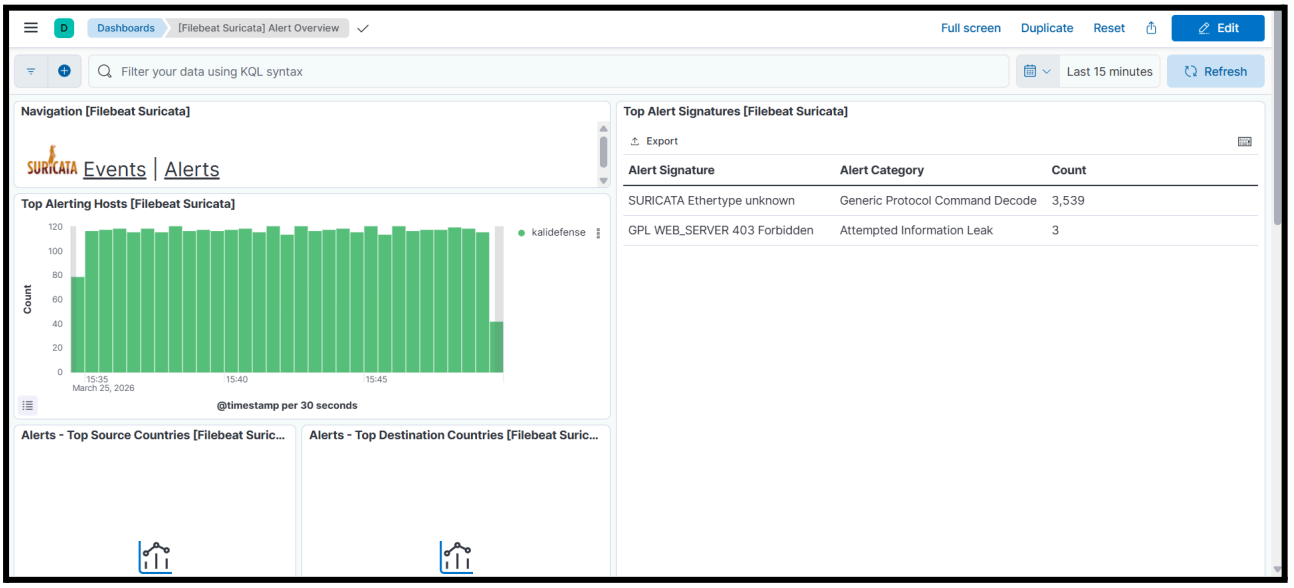
5.Vérification de l'interface graphique

L'infrastructure de collecte, de journalisation et de corrélation visuelle est désormais intégralement opérationnelle et en attente d'événements.

- Depuis un poste physique *(connecté au réseau de Management VENUS)*, ouvrez un navigateur Web.
- Accédez à l'interface via l'adresse IP de management (`eth0`) de la VM Kali : `http://192.168.99.204:5601`
- Sur l'écran d'accueil de bienvenue d'Elastic, validez en cliquant sur le bouton "`Explore on my own`".
- Déployez le menu principal (icône en haut à gauche) et naviguez vers la section `Analytics > Dashboard`.
- Saisissez "`Suricata`" dans la barre de recherche pour trouver les tableaux de bord importés, notamment :
  - `[Filebeat Suricata] Events overview` : Synthèse globale du trafic, requêtes DNS, connexions réseau.
  - `[Filebeat Suricata] Alert overview` : Tableau de bord affichant exclusivement les alertes critiques (issues des règles `custom.rules` et Emerging Threats).







6.Configuration du tableau de bord - Vénus Alert

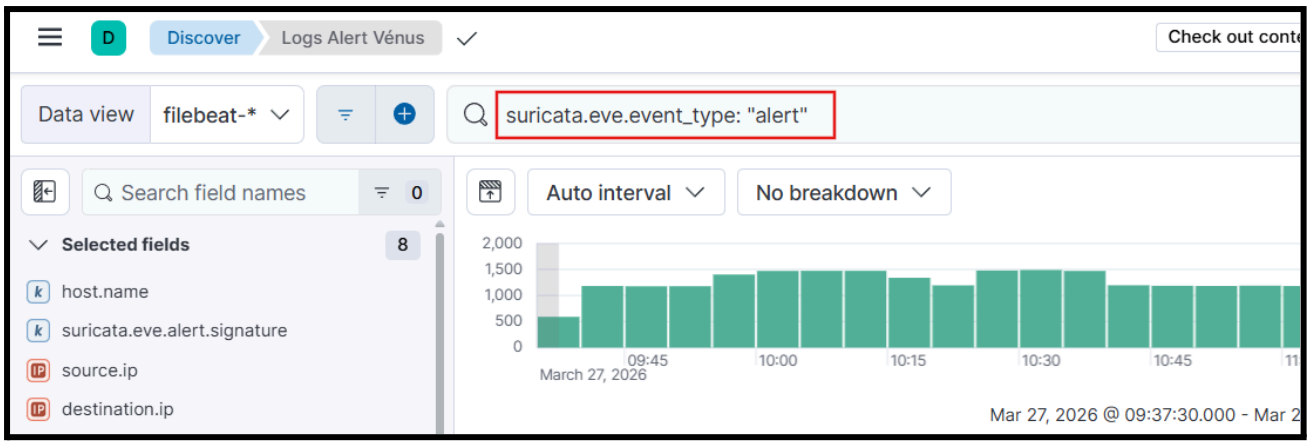
CRÉATION DU DASHBOARD "ALERTES VENUS"

Cette section explique comment créer un tableau de bord permanent en temps réel pour visualiser exclusivement nos attaques personnalisées Vénus, avec les adresses IP et la gravité.

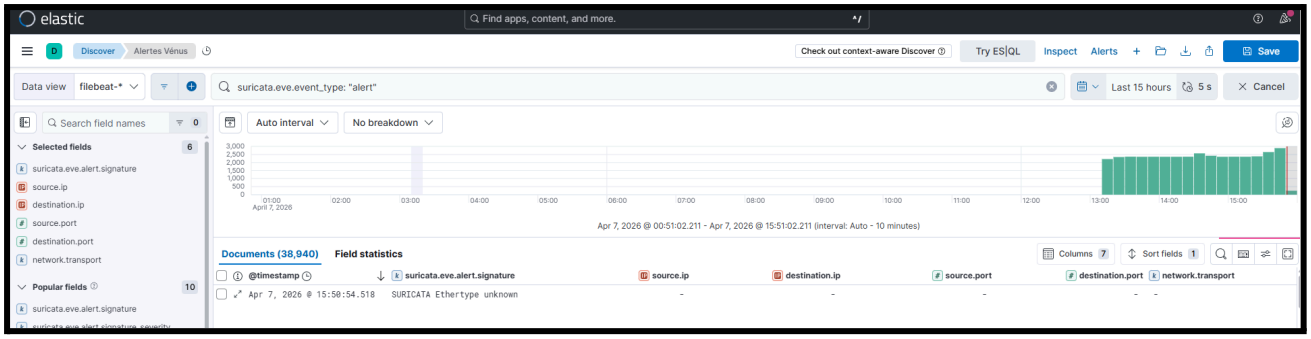
1. Préparation de la vue dans Discover

C'est l'étape la plus importante : on va choisir les bonnes colonnes avant de créer le tableau.

- Allez dans le menu principal (☰ *en haut à gauche)* > Analytics > Discover.
- Dans la barre de recherche en haut, tapez exactement cette requête et faites *Entrée* : suricata.eve.event_type: "alert"



- Déployez le panneau de gauche (s'il est caché) en cliquant sur l'icône de liste.
- Dans la section Available fields, cherchez les champs suivants avec la barre de recherche et cliquez sur le petit "+" pour les ajouter à votre tableau :
    - `signature` (Le texte de notre alerte, ex: "ALERTE VENUS")
    - `source.ip` (L'IP de l'attaquant)
    - `destination.ip` (L'IP de la cible)
    - `source.port` (Port de l'attaquant)
    - `destination.port` (Port de la cible)
    - `network.transport` (Le protocole : icmp, tcp, udp)

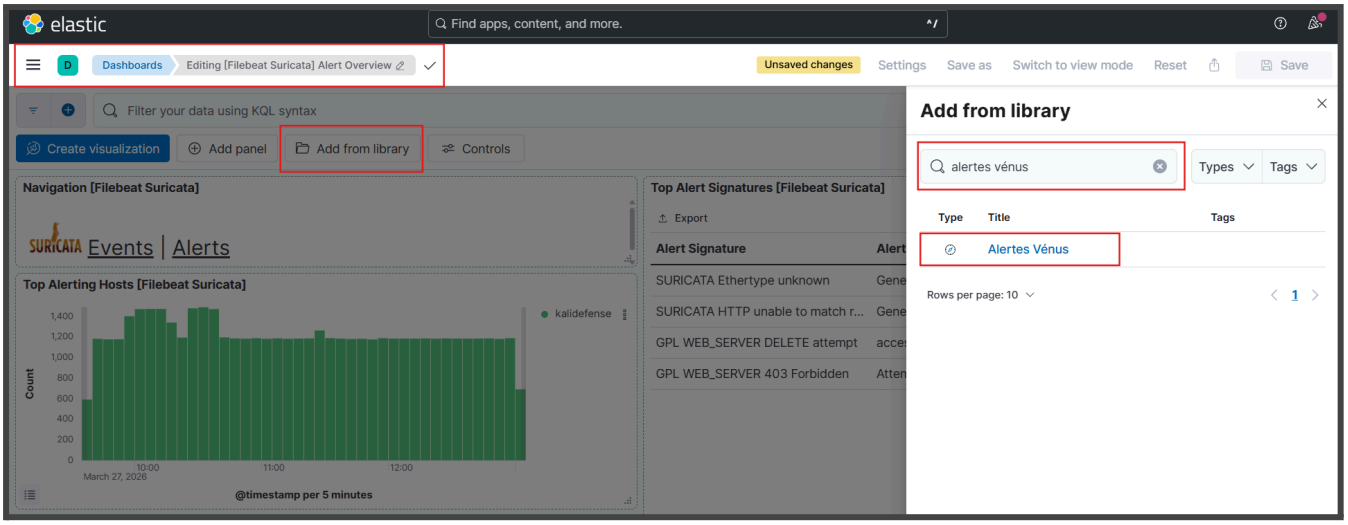


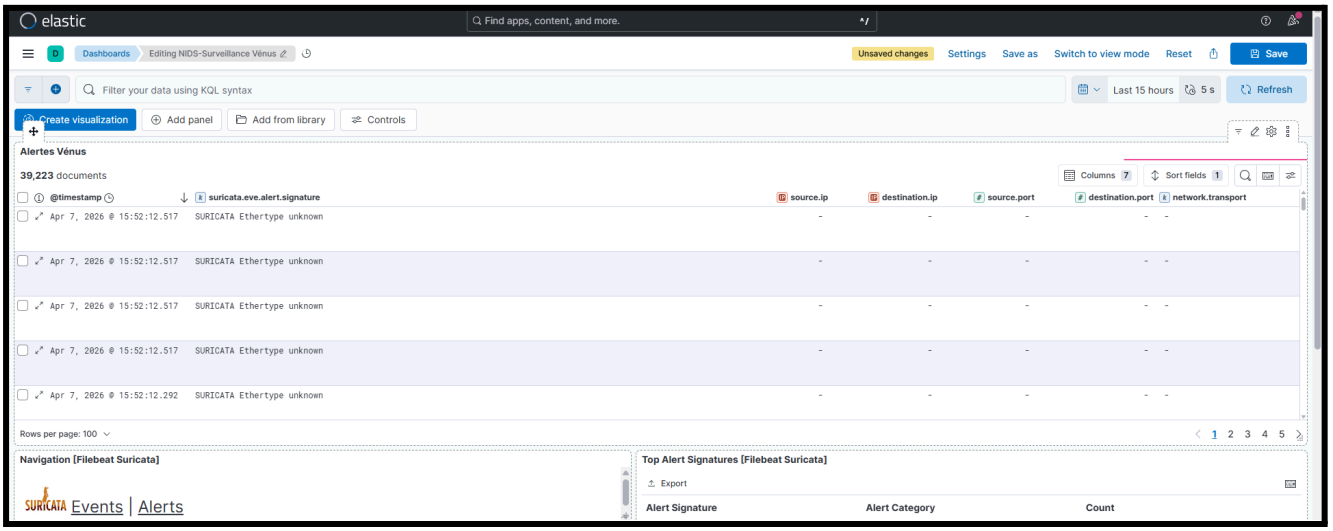
2. Sauvegarde de la vue

1. En haut à droite de l'écran Discover, cliquez sur l'icône de disquette "Save".
2. Dans la fenêtre qui s'ouvre, nommez cette vue : "Alerte VENUS".
3. Cliquez sur le bouton bleu "Save".

3. Création du Dashboard final

1. Allez dans le menu principal (☰) > Analytics > Dashboard.
2. Cliquez sur le bouton bleu "Create new dashboard" en haut à droite.
3. Au centre, cliquez sur "Add from library" (ou "Add panels").
4. Dans la barre de recherche, tapez "Vue Alertes VENUS" et cliquez dessus.
5. Cliquez sur "Save" en haut à droite.
6. Nommez le tableau de bord : exemple :"NIDS - Surveillance VENUS"





Panneau “Alertes Vénus” ajouté avec succès à la sexion “Dashboard”