Ecole Supérieure d'Informatique et du Numérique
COLLEGE OF ENGINEERING & ARCHITECTURE

Université Internationale de Rabat

College of Engineering & Architecture

École Supérieure d'Informatique et du Numérique (ESIN)

1ère Année Cycle d'Ingénieur

Rapport de Stage Technique

Conception et mise en œuvre d'un laboratoire SOC virtualisé

pour la détection et la supervision des cybermenaces

Stage réalisé au sein de CIH Bank

CIH بنك

Réalisé par	:	**Achraf ACHOUKHANE**
Filière	:	Cybersecurity
Encadrants professionnels	:	M. EL MOSTAFA ED-DAIYF
		M. Oussama AZZAM
Encadrante pédagogique	:	Mme Hajar EL GADI
Période du stage	:	du 22 juin 2026 au 15 août 2026

Année académique : 2025–2026

Remerciements

Je remercie **CIH Bank** pour son accueil et pour les conditions d'apprentissage offertes durant ce stage.

J'adresse mes sincères remerciements à mes encadrants professionnels, **M. EL MOSTAFA EDDAIYF** et **M. Oussama AZZAM**, pour leurs orientations, leur disponibilité et leurs retours tout au long du projet.

Je remercie également mon encadrante pédagogique, **Mme Hajar EL GADI**, ainsi que l'**Université Internationale de Rabat** et les enseignants de la filière **Cybersecurity**, pour leur accompagnement et la formation dispensée.

Résumé

Ce rapport présente la réalisation d'un laboratoire SOC virtualisé dans le cadre d'un stage au sein de CIH Bank. L'environnement relie un pare-feu OPNsense, un serveur Wazuh, des machines Linux et Windows, ainsi qu'une machine Kali Linux réservée aux tests contrôlés.

Le laboratoire centralise les journaux, supervise les hôtes et détecte plusieurs événements : tentatives SSH, échecs d'authentification Web, scan réseau, création de processus Windows, modification de fichier et erreur IIS. Les résultats sont illustrés par les captures obtenues pendant la validation.

Mots-clés : SOC, SIEM, Wazuh, OPNsense, Suricata, Sysmon, supervision, cybersécurité.

Abstract

This report presents a virtualized SOC laboratory developed during an internship at CIH Bank. The environment connects an OPNsense firewall, a Wazuh server, Linux and Windows machines, and a Kali Linux machine used for controlled tests.

The laboratory centralizes logs, monitors endpoints, and detects several events: SSH attempts, Web authentication failures, network scanning, Windows process creation, file modification, and an IIS error. The main results are supported by screenshots collected during validation.

Keywords: SOC, SIEM, Wazuh, OPNsense, Suricata, Sysmon, monitoring, cybersecurity.

Liste des acronymes

Acronyme	Signification
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
EVE	Format JSON utilisé par Suricata
FIM	File Integrity Monitoring
HIDS	Host-based Intrusion Detection System
IIS	Internet Information Services
NIDS	Network Intrusion Detection System
SIEM	Security Information and Event Management
SOC	Security Operations Center
SOAR	Security Orchestration, Automation and Response
SSH	Secure Shell
VM	Virtual Machine

Table des matières

Table des figures

1 Présentation de CIH Bank et du stage

1.1 Présentation de CIH Bank

Figure 1.1. Logo de CIH Bank

CIH Bank est un établissement bancaire marocain qui propose des services aux particuliers, aux professionnels et aux entreprises. Comme toute organisation financière, la banque s'appuie sur un système d'information disponible, fiable et surveillé. La cybersécurité participe donc directement à la continuité des services et à la protection des données [CIH Bank, 2025].

Table 1.1. Informations générales sur le stage

Organisme d'accueil	CIH Bank
Secteur	Banque et services financiers
Période	Du 22 juin 2026 au 15 août 2026
Sujet	Mise en place d'un laboratoire SOC virtualisé
Filière	Cybersecurity

1.2 Organisation de l'entreprise

L'organigramme fourni présente les principales directions de CIH Bank. La branche dédiée aux services technologiques comprend notamment la sécurité des systèmes d'information et le pôle des systèmes d'information. C'est dans ce contexte que s'inscrit le sujet du stage.

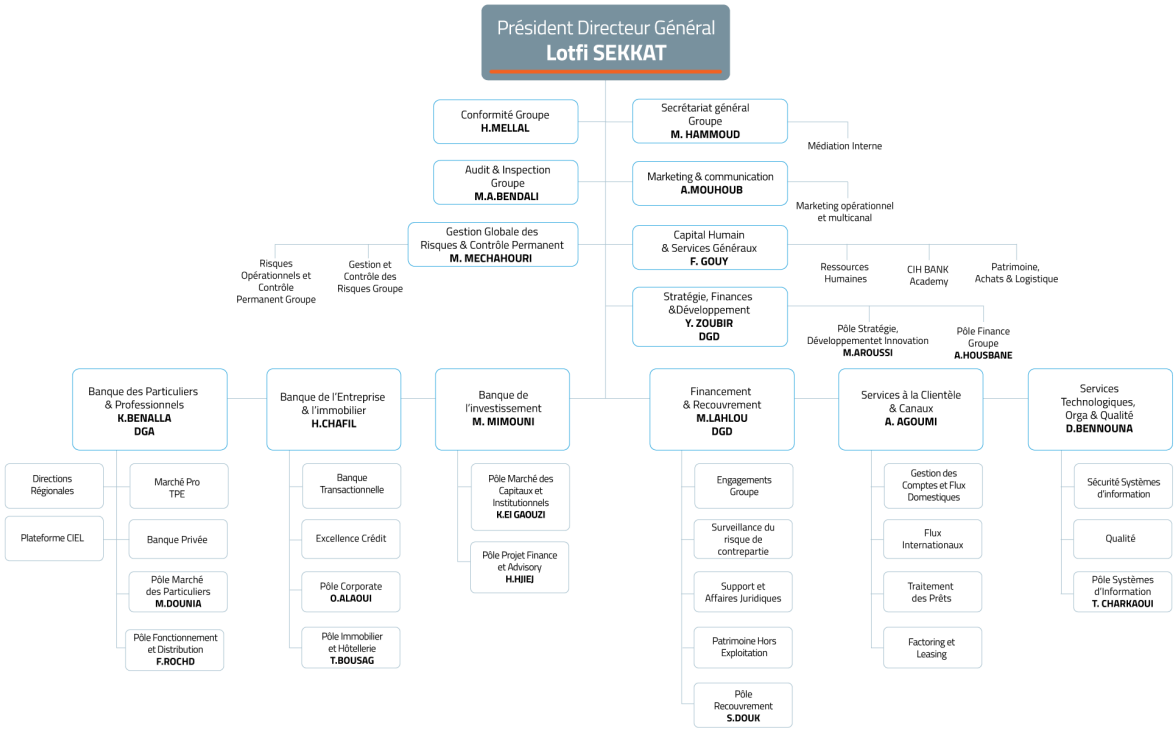


FIGURE 1.2. Organigramme général de CIH Bank fourni dans les ressources du stage

1.3 Contexte du stage

Le projet consiste à construire un environnement isolé permettant de comprendre le fonctionnement d'un SOC sans intervenir sur le réseau de production. Le laboratoire reproduit une chaîne simple :

Générer un événement → collecter le journal → détecter → analyser l'alerte

Les tests sont réalisés avec des machines virtuelles et des adresses privées. Ils concernent le réseau, un serveur Linux, un serveur Windows et une application Web de démonstration.

1.4 Enjeux de sécurité

Dans un environnement bancaire, la supervision ne consiste pas uniquement à bloquer une connexion. Elle doit aussi conserver une trace exploitable afin de comprendre l'événement et d'identifier la machine concernée. Le laboratoire reprend cette logique à une échelle réduite.

Table 1.2. Enjeux de sécurité représentés dans le laboratoire

Enjeu	Exemple	Réponse du laboratoire
Confidentialité	Accès non autorisé	Filtrage réseau et contrôle des connexions
Intégrité	Modification d'un fichier	Surveillance FIM
Disponibilité	Activité répétée ou scan	Blocage OPNsense et détection Suricata
Traçabilité	Recherche après incident	Centralisation des journaux dans Wazuh

À retenir. Le SOC regroupe des informations provenant de plusieurs sources afin de transformer un journal isolé en événement compréhensible.

1.5 Objectifs du projet

L'objectif principal est de mettre en place un petit laboratoire capable de centraliser et de visualiser des événements de sécurité. Les objectifs retenus sont :

— séparer le réseau de test, le réseau des serveurs et le réseau de supervision ;
— filtrer les communications avec OPNsense ;
— centraliser les journaux dans Wazuh ;
— superviser un serveur Linux et un serveur Windows ;
— détecter quelques scénarios simples et contrôlés ;
— conserver des captures permettant de vérifier les résultats.

1.6 Déroulement du stage

Table 1.3. Étapes principales du travail

Étape	Travail réalisé
Cadrage	Compréhension du sujet et choix du périmètre
Conception	Définition des machines, des zones et des flux
Mise en place	Installation et connexion des composants
Validation	Exécution des tests et collecte des captures
Documentation	Analyse des résultats et rédaction du rapport

1.7 Livrables réalisés

Le travail produit quatre livrables principaux :

— une architecture réseau segmentée ;
— un ensemble de machines virtuelles configurées ;
— des règles et scénarios de validation ;

— un dossier de captures et le présent rapport.

1.8 Organisation du rapport

Le rapport présente d'abord l'entreprise et l'environnement utilisé, puis l'architecture, la mise en place et les tests. Seuls les résultats visibles dans les captures sont retenus.

2 Environnement de travail et outils utilisés

2.1 Notions utiles

Le laboratoire reprend trois fonctions simples d'un SOC :

Table 2.1. Fonctions représentées dans le laboratoire

Fonction	Rôle dans le projet
Collecte	Récupérer les journaux réseau, Linux et Windows
Détection	Appliquer des règles et produire des alertes
Analyse	Rechercher les événements dans Wazuh Threat Hunting

Wazuh joue le rôle de plateforme SIEM et de supervision des hôtes. OPNsense filtre le réseau, tandis que Suricata inspecte le trafic. Sysmon complète les journaux Windows avec des informations sur les processus.

2.2 Fonctionnement simplifié d'une supervision

Une alerte est le résultat de plusieurs opérations successives. La source crée d'abord un journal. Celui-ci est transporté vers Wazuh, décodé, comparé à une règle, puis affiché si les conditions sont vérifiées.

Table 2.2. Étapes de traitement d'un événement

Étape	Élément	Résultat
Production	Pare-feu, système ou application	Ligne de journal
Transport	Agent Wazuh ou Syslog	Événement transmis
Décodage	Décodeur Wazuh	Champs identifiés
Détection	Règle Wazuh	Niveau et description
Analyse	Threat Hunting	Alerte consultable

Cette chaîne est importante : une règle ne peut fonctionner que si le journal est reçu et correctement décodé. La gestion des journaux constitue donc une base du travail d'un SOC [Kent and Souppaya, 2006].

2.2.1 NIDS et HIDS

TABLE 2.3. Complémentarité de la détection réseau et hôte

Type	Point observé	Exemple du projet
NIDS	Trafic traversant le réseau	Scan SSH détecté par Suricata
HIDS	Activité interne d'une machine	Processus Sysmon ou modification FIM

2.3 Machines virtuelles

VMware
Plateforme de virtualisation

Kali Linux
Machine de test contrôlé

Ubuntu Server
Serveur Linux et application Web

Windows Server
IIS, Sysmon et agent Wazuh

OPNsense
Pare-feu, routage et filtrage

Wazuh
Collecte, règles et visualisation

FIGURE 2.1. Principales machines et plateformes du laboratoire

2.4 Outils de sécurité et de supervision

Suricata
Détection d'intrusion réseau

Sysmon
Télémétrie Windows

IIS
Service Web Windows supervisé

MITRE ATT&CK
Référentiel de techniques

ET Open
Règles utilisées par Suricata

Threat Hunting
Recherche dans les alertes

FIGURE 2.2. Outils utilisés pour la détection et l'analyse

Table 2.4. Rôle des principaux outils

Outil	Catégorie	Utilisation
OPNsense	Pare-feu	Séparation des zones et blocage de flux
Wazuh	SIEM/HIDS	Agents, règles, alertes et Threat Hunting
Suricata	NIDS	Inspection du trafic avec ET Open
Sysmon	Journal Windows	Création de processus et détails associés
Apache	Serveur Web Linux	Hébergement de l'application de test
IIS	Serveur Web Windows	Génération de journaux HTTP

Les rôles décrits correspondent aux fonctions documentées par les projets Wazuh, OPNsense, Suricata et Microsoft [Wazuh, 2026b, OPNsense, 2026b, Open Information Security Foundation, 2026, Russinovich and Garnier, 2026].

2.5 Inventaire des machines

Table 2.5. Machines du laboratoire

Machine	Adresse	Fonction
Kali Linux	`10.10.10.10`	Génération des tests
OPNsense	`10.10.10.1`	Passerelle de la zone ATTACK
OPNsense	`192.168.56.1`	Passerelle de la zone LAN
OPNsense	`172.16.10.1`	Passerelle de la zone SIEM
SRV Linux Web	`192.168.56.20`	Apache, SSH, application et FIM
SRV Windows	`192.168.56.30`	IIS, Sysmon et agent Wazuh
Wazuh SIEM	`172.16.10.10`	Centralisation et analyse

2.6 Réseaux virtuels et interfaces

VMware relie les cartes réseau virtuelles aux trois segments du laboratoire. OPNsense possède une interface dans chaque zone et devient le seul point de passage entre elles. Une machine de la zone ATTACK ne communique donc pas directement avec la zone SIEM.

Table 2.6. Connexion des machines aux réseaux virtuels

Machine	Zone	Connexion
Kali Linux	ATTACK	Une carte réseau de test
SRV Linux Web	LAN	Une carte réseau serveur
SRV Windows	LAN	Une carte réseau serveur
Wazuh SIEM	SIEM	Une carte réseau de supervision
OPNsense	ATTACK, LAN, SIEM	Trois interfaces de routage

Cette séparation facilite les tests : une règle peut être appliquée à une zone précise sans modifier

les autres réseaux. Elle améliore également la lecture des journaux, car l'interface d'origine indique le chemin suivi par le trafic.

À retenir. La virtualisation permet de regrouper tout le laboratoire sur une même plateforme, tout en gardant des rôles et des réseaux séparés.

---

3 Architecture du laboratoire

3.1 Topologie générale

L'architecture est divisée en trois zones. Cette séparation permet de contrôler les échanges et de distinguer clairement la machine de test, les serveurs supervisés et la plateforme de sécurité.

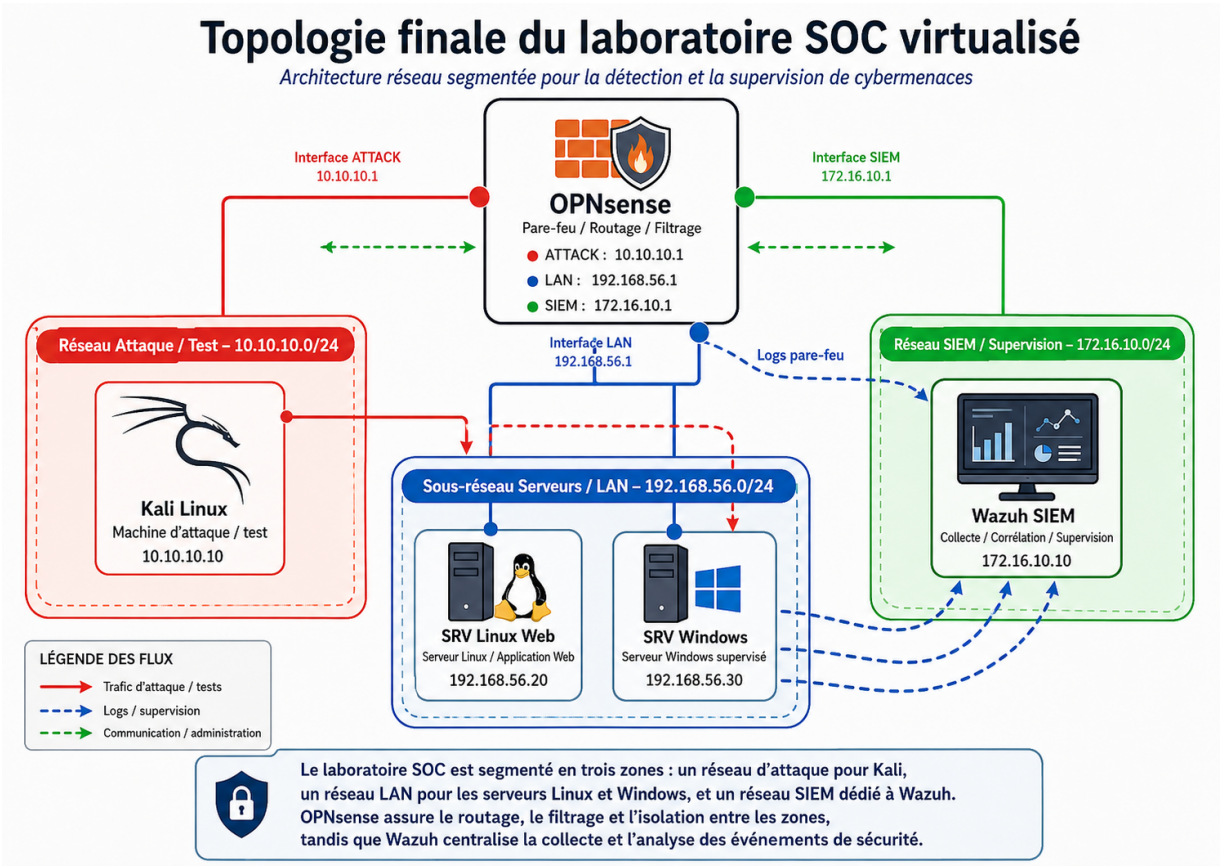


FIGURE 3.1. Topologie du laboratoire SOC virtualisé

TABLE 3.1. Segmentation réseau

Zone	Réseau	Éléments
ATTACK	`10.10.10.0/24`	Kali Linux et interface OPNsense
LAN	`192.168.56.0/24`	Serveurs Linux et Windows
SIEM	`172.16.10.0/24`	Serveur Wazuh

3.2 Règles de circulation

OPNsense applique un filtrage avec suivi d'état. Lorsqu'une connexion est autorisée, le pare-feu suit la session et accepte le trafic de retour correspondant. Les autres communications restent bloquées si aucune règle ne les autorise.

TABLE 3.2. Flux nécessaires au fonctionnement du laboratoire

Source	Destination	Port/service	Utilité
Kali Linux	Serveurs LAN	22, 80	Scénarios contrôlés
Agents Wazuh	Wazuh SIEM	TCP 1514	Envoi des événements
OPNsense	Wazuh SIEM	Syslog 514	Journaux du pare-feu
Suricata	Wazuh SIEM	EVE/Syslog	Alertes du NIDS
Administrateur	Interfaces Web	HTTPS	Configuration et analyse

Les flux d'administration sont séparés des actions de test. Cette distinction évite de confondre une opération normale avec un événement de sécurité.

3.3 Circulation des événements

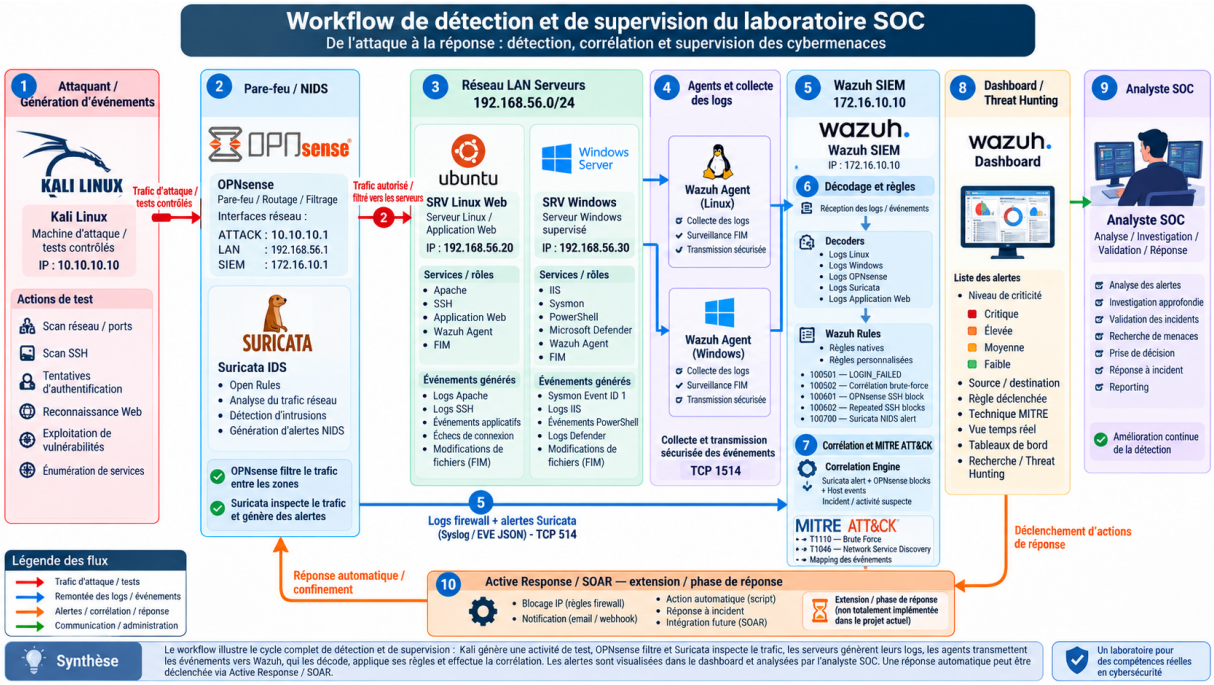


FIGURE 3.2. Workflow de détection ; le bloc Active Response/SOAR représente une évolution future non implémentée

Le fonctionnement du laboratoire peut être résumé en cinq étapes :

1. Kali Linux génère une action de test ;
2. OPNsense autorise ou bloque le trafic selon les règles ;
3. les hôtes et Suricata produisent des journaux ;
4. les agents et Syslog transmettent les événements à Wazuh ;
5. Wazuh applique les règles et affiche les alertes dans Threat Hunting.

3.4 Sources de journaux

Table 3.3. Journaux exploités

Source	Type de journal	Exemple d'événement
OPNsense	Pare-feu/Syslog	Connexion SSH bloquée
Suricata	EVE JSON/Syslog	Scan SSH détecté
Linux	Système et application	Authentification, FIM
Windows	Event Log et Sysmon	Création de processus
IIS	Journal HTTP	Requête retournant 404

3.5 Chaînes de détection

La même plateforme reçoit des événements de nature différente. Le chemin exact dépend de la source :

Table 3.4. Exemples de chaînes de détection

Scénario	Chemin suivi
Scan SSH	Kali → OPNsense → Suricata → EVE/Syslog → Wazuh
Échec Web	Application → journal JSON → agent Wazuh → décodeur → règle
Processus Windows	Sysmon → Event Log → agent Wazuh → Threat Hunting
Modification FIM	Fichier → agent Wazuh → comparaison d'intégrité → alerte

3.6 Critères de validation

Un scénario est considéré comme validé lorsque les quatre points suivants sont vérifiés :

1. l'action de test est identifiable ;
2. la source produit un journal ;
3. Wazuh reçoit un événement contenant les champs attendus ;
4. la règle et son niveau sont visibles dans l'alerte.

À retenir. Une capture de tableau de bord ne suffit pas seule : elle doit pouvoir être reliée à une action, une source et une règle.

3.7 Périmètre retenu

Le laboratoire est volontairement limité à quelques machines et scénarios faciles à reproduire. Il ne contient ni haute disponibilité ni SOAR opérationnel. Ces éléments sont présentés uniquement comme perspectives.

À retenir. L'architecture privilégie une chaîne courte et vérifiable : réseau, hôtes, collecte Wazuh, règle de détection et alerte visible.

4 Mise en place du laboratoire

4.1 Ordre de déploiement

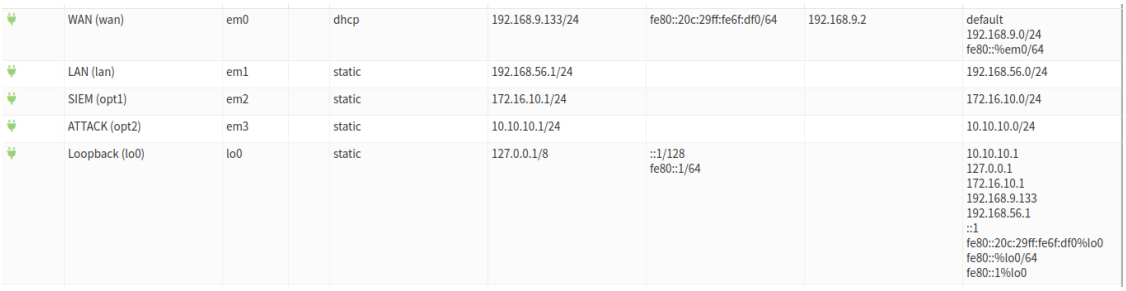
La mise en place a suivi un ordre progressif afin d'éviter de rechercher plusieurs pannes en même temps.

TABLE 4.1. Ordre de déploiement des composants

Étape	Composant	Vérification
1	Réseaux VMware	Cartes connectées aux bonnes zones
2	OPNsense	Interfaces et passerelles accessibles
3	Serveur Wazuh	Tableau de bord disponible
4	Serveurs Linux/Windows	Services et adresses configurés
5	Agents Wazuh	Agents actifs dans le tableau de bord
6	Suricata et règles	Premiers événements reçus

4.2 Configuration d'OPNsense

OPNsense possède une interface dans chaque zone. Il assure le routage entre les réseaux et applique les règles de filtrage.



	WAN (wan)	em0		dhcp	192.168.9.133/24	fe80::20c:29ff:fe6f:df0/64	192.168.9.2	default 192.168.9.0/24 fe80::%em0/64
	LAN (lan)	em1		static	192.168.56.1/24			192.168.56.0/24
	SIEM (opt1)	em2		static	172.16.10.1/24			172.16.10.0/24
	ATTACK (opt2)	em3		static	10.10.10.1/24			10.10.10.0/24
	Loopback (lo0)	lo0		static	127.0.0.1/8	::1/128 fe80::1/64		10.10.10.1 127.0.0.1 172.16.10.1 192.168.9.133 192.168.56.1 ::1 fe80::20c:29ff:fe6f:df0%lo0 fe80::%lo0/64 fe80::1%lo0

FIGURE 4.1. Interfaces réseau configurées sur OPNsense

La politique de filtrage autorise uniquement les communications nécessaires au laboratoire. Les tests SSH provenant de la zone ATTACK peuvent ainsi être bloqués et journalisés.

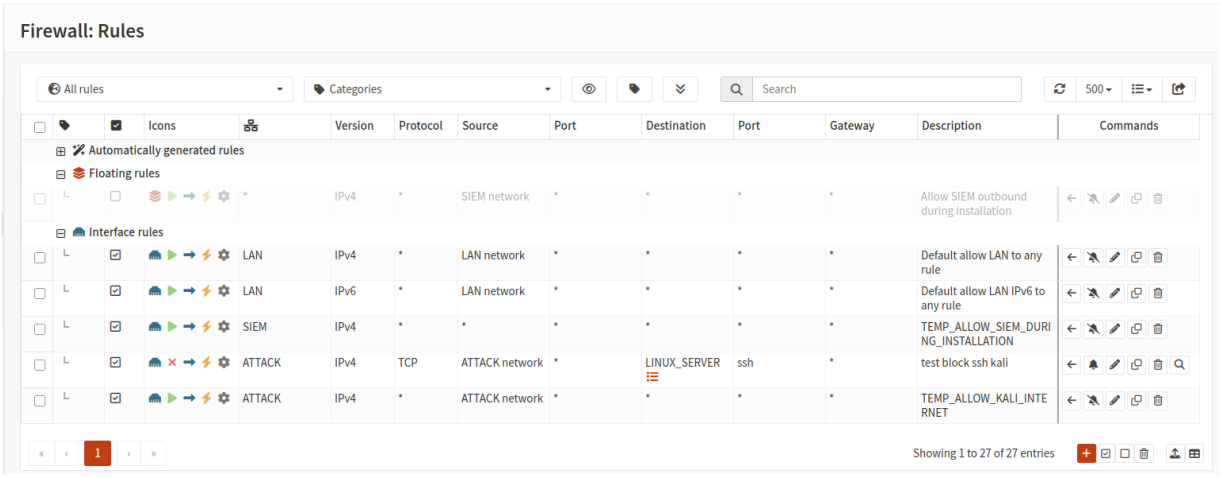


FIGURE 4.2. Règles de filtrage appliquées à la zone ATTACK

4.3 Installation de Wazuh

Le serveur Wazuh est installé dans la zone SIEM. Les agents Linux et Windows lui transmettent les événements des deux serveurs supervisés.

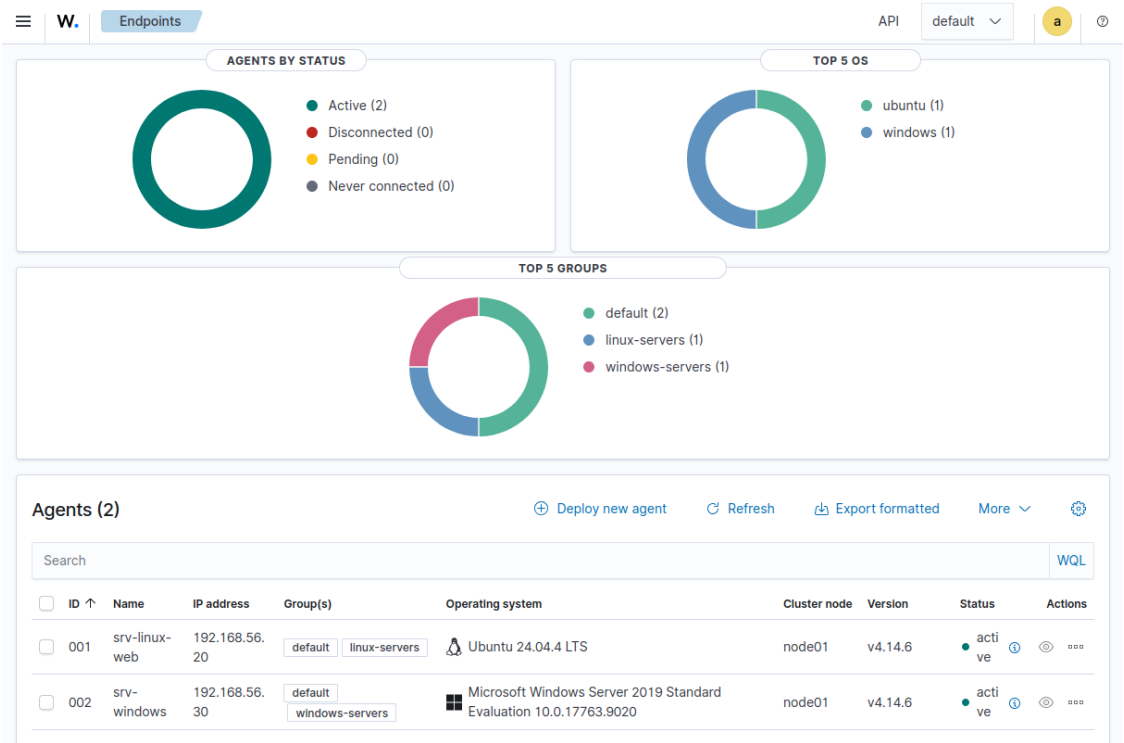


FIGURE 4.3. Agents Linux et Windows actifs dans Wazuh

4.3.1 Communication entre les agents et le serveur

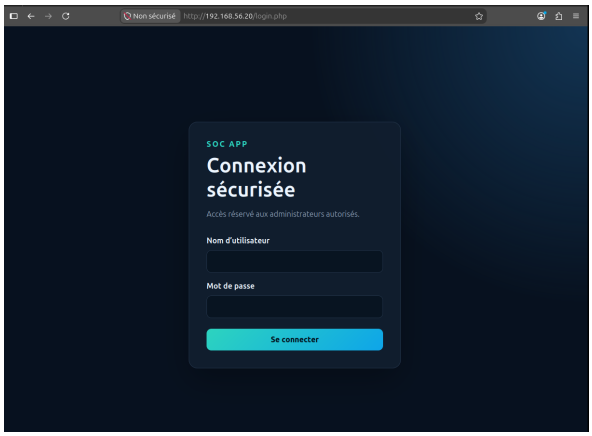
Chaque agent possède un identifiant et transmet les événements de sa machine au serveur Wazuh. Le serveur associe ensuite l'événement au nom et à l'adresse de l'agent. Cette association est visible dans les champs `agent.name`, `agent.ip` et `agent.id`.

TABLE 4.2. Événements collectés par les agents

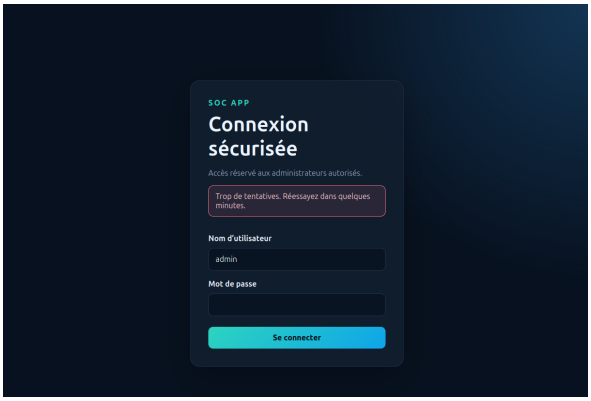
Agent	Principales données collectées
SRV Linux Web	Journaux système, application JSON et FIM
SRV Windows	Event Log, Sysmon, IIS et FIM

4.4 Serveur Linux et application Web

Le serveur Ubuntu héberge Apache, SSH, l'agent Wazuh et une application Web de test. L'application enregistre les échecs d'authentification dans un journal surveillé par Wazuh.



(a) Application hébergée par Apache



(b) Blocage après plusieurs échecs

FIGURE 4.4. Application Web utilisée pour le scénario d'authentification

4.5 Serveur Windows

Le serveur Windows reçoit IIS, Sysmon et l'agent Wazuh. IIS fournit les journaux Web, tandis que Sysmon ajoute des informations sur les processus. Le FIM de Wazuh surveille également les modifications de fichiers.

4.6 Intégration de Suricata

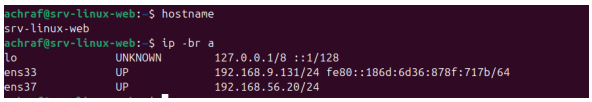
Suricata est activé sur l'interface ATTACK avec les règles ET Open. Ses alertes sont transmises à Wazuh afin d'être recherchées dans Threat Hunting.

TABLE 4.3. Paramètres retenus pour Suricata

Paramètre	Valeur
Interface surveillée	ATTACK
Jeu de règles	Emerging Threats Open / ET Open
Format d'événement	EVE JSON
Destination	Wazuh SIEM
Validation	Signature ET SCAN Potential SSH Scan

4.7 Vérifications de base

Avant les scénarios, l'adresse du serveur Linux et la réponse du service Apache sont contrôlées. Ces tests simples confirment que le problème ne vient pas de la configuration réseau ou du serveur Web.



(a) Nom et adresse du serveur

(b) Réponse HTTP reçue

FIGURE 4.5. Contrôles réalisés avant les scénarios

TABLE 4.4. Contrôles réalisés après l'installation

Composant	Contrôle
OPNsense	Interfaces visibles et règles actives
Wazuh	Serveur accessible et agents connectés
Linux	Apache, SSH et agent Wazuh actifs
Windows	IIS, Sysmon et agent Wazuh actifs
Suricata	Alertes reçues dans Wazuh

À retenir. La mise en place est validée lorsque les machines communiquent selon les règles et que les événements sont visibles dans Wazuh.

5 Tests et résultats

5.1 Principe de validation

Chaque test suit la même logique : une action est lancée depuis une machine du laboratoire, puis le résultat est vérifié dans OPNsense, dans le journal de l'hôte ou dans Wazuh.

Table 5.1. Scénarios retenus

Scénario	Source	Résultat attendu
Échecs de connexion Web	Application Linux	Alerte et corrélation
Tentatives SSH bloquées	OPNsense	Règles `100601`/`100602`
Création de processus	Sysmon	Event ID 1 visible
Scan SSH	Suricata	Alerte `100700`
Modification de fichier	Wazuh FIM	Alerte d'intégrité
Requête IIS invalide	IIS	Code HTTP 404 collecté

5.2 Règles applicatives Wazuh

L'application Web écrit ses événements d'authentification au format JSON. Les règles personnalisées Wazuh reconnaissent ces événements, puis les corrèlent selon l'adresse IP et la période observée. Cette partie présente les cinq règles applicatives utilisées dans le laboratoire.

Table 5.2. Explication des règles applicatives Wazuh

Identifiant	Niveau	Explication
`100501`	3	Reconnaît un événement `LOGIN_FAILED`. L'option `no_log` évite de créer une alerte isolée : cette règle sert de base aux corrélations.
`100502`	10	Détecte cinq échecs provenant de la même adresse IP en 300 secondes. Elle signale une tentative de brute force (MITRE ATT&CK T1110).
`100510`	3	Reconnaît un événement `LOGIN_SUCCESS`. Comme la règle 100501, elle sert de règle de base et n'est pas affichée seule.
`100503`	12	Signale une connexion réussie après trois échecs issus de la même adresse IP en cinq minutes. Ce comportement peut indiquer un compte compromis (MITRE ATT&CK T1078).
`100504`	10	Détecte l'événement `LOGIN_RATE_LIMIT` lorsque l'application limite les tentatives après plusieurs échecs.

Les règles `100501` et `100510` identifient donc les événements simples. Les règles `100502` et `100503` analysent leur enchaînement, tandis que `100504` confirme que la protection applicative a limité le rythme des tentatives.

```
achraf@wazuh-siem:~$ sudo grep -n -A 15 -B 3 '100501' /var/ossec/etc/rules/soc_app_rules.xml
1-<group name="soc_app,web_application,attack_detection,">
2-
3-  <!-- Échec de connexion conservé uniquement pour la corrélation -->
4:  <rule id="100501" level="3">
5-    <decoded_as>json</decoded_as>
6-    <field name="event" type="pcre2">^LOGIN_FAILED$</field>
7-    <description>SOC App: échec de connexion depuis $(src_ip)</description>
8-    <options>no_log</options>
9-  </rule>
10-
11-  <!-- Cinq échecs depuis la même IP en cinq minutes -->
12-  <rule id="100502" level="10" frequency="5" timeframe="300" ignore="60">
13:    <if_matched_sid>100501</if_matched_sid>
14-    <same_field>src_ip</same_field>
15-    <description>SOC App: tentative de brute force Web depuis $(src_ip)</description>
16-    <group>authentication_failures,brute_force,</group>
17-    <mitre>
18-      <id>T1110</id>
19-    </mitre>
20-  </rule>
21-  <!-- Connexion réussie conservée uniquement comme événement parent -->
22-  <rule id="100510" level="3">
23-    <decoded_as>json</decoded_as>
24-    <field name="event" type="pcre2">^LOGIN_SUCCESS$</field>
25-    <description>SOC App: connexion réussie depuis $(src_ip)</description>
26-    <options>no_log</options>
27-  </rule>
28-
29-  <!-- Connexion réussie après trois échecs depuis la même IP -->
30-  <rule id="100503" level="12" frequency="4" timeframe="300" ignore="60">
31-    <if_sid>100510</if_sid>
32:    <if_matched_sid>100501</if_matched_sid>
```

FIGURE 5.1. Règles applicatives de détection et de corrélation des échecs

```
32:    <if_matched_sid>100501</if_matched_sid>
33-    <same_field>src_ip</same_field>
34-    <description>SOC App: connexion réussie après plusieurs échecs depuis $(src_ip) pour $(dstuser)</description>
35-    <group>authentication_success,possible_account_compromise,</group>
36-    <mitre>
37-      <id>T1078</id>
38-    </mitre>
39-  </rule>
40-
41-  <!-- L'application a appliqué sa limitation de tentatives -->
42-  <rule id="100504" level="10">
43-    <decoded_as>json</decoded_as>
44-    <field name="event" type="pcre2">^LOGIN_RATE_LIMIT$</field>
45-    <description>SOC App: limitation déclenchée après plusieurs échecs depuis $(src_ip)</description>
46-    <group>authentication_failures,brute_force,</group>
47-    <mitre>
achraf@wazuh-siem:~$
```

FIGURE 5.2. Règles applicatives de connexion réussie et de limitation

5.2.1 Exemple de corrélation : règle 100502

L'extrait suivant montre la logique réellement utilisée pour la règle `100502`. Elle ne recherche pas directement un mot dans le journal : elle s'appuie sur les alertes déjà produites par la règle `100501`.

```
<rule id="100502" level="10" frequency="5"
      timeframe="300" ignore="60">
  <if_matched_sid>100501</if_matched_sid>
  <same_field>src_ip</same_field>
  <description>Tentative de brute force Web</description>
</rule>
```

Table 5.3. Lecture des paramètres de la règle 100502

Paramètre	Signification
`level="10"`	Niveau élevé attribué à l'alerte corrélée.
`frequency="5"`	Cinq alertes de base sont nécessaires.
`timeframe="300"`	Les cinq alertes doivent apparaître en cinq minutes.
`ignore="60"`	Après une alerte, les doublons sont ignorés pendant 60 secondes.
`if_matched_sid`	La corrélation porte sur la règle de base `100501`.
`same_field`	Les événements doivent concerner la même adresse source.

Cette séparation entre règle unitaire et règle corrélée limite le bruit : un seul échec reste peu critique, alors qu'une répétition depuis la même source produit une alerte plus importante.

5.3 Autres règles Wazuh validées

Les autres scénarios du laboratoire s'appuient sur les règles suivantes :

Table 5.4. Règles utilisées pour le réseau et l'intégrité

Identifiant	Niveau	Explication
`100601`	7	Signale une tentative SSH bloquée et journalisée par OPNsense.
`100602`	10	Corrèle plusieurs blocages SSH et augmente la criticité de l'activité répétée.
`100700`	7	Transforme l'alerte Suricata *ET SCAN Potential SSH Scan* en alerte Wazuh consultable.
`550`	7	Règle native FIM indiquant qu'un fichier surveillé a été modifié.

5.4 Échecs d'authentification Web

Une règle unitaire détecte l'événement `LOGIN_FAILED`. Une seconde règle augmente le niveau lorsque plusieurs échecs sont observés sur une courte période.

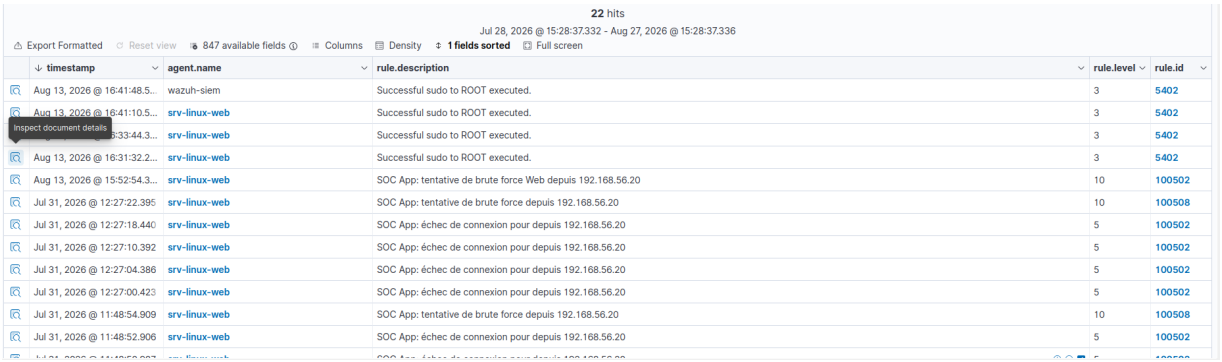


Figure 5.3. Alertes applicatives visibles dans Wazuh

5.4.1 Lecture de la règle corrélée

t	rule.description	SOC App: tentative de brute force Web depuis 192.168.56.20
#	rule.firedtimes	1
#	rule.frequency	5
t	rule.groups	soc_app, web_application, attack_detection, authentication_failures, brute_force
t	rule.id	100502
#	rule.level	10
◕	rule.mail	false
t	rule.mitre.id	T1110
t	rule.mitre.tactic	Credential Access
t	rule.mitre.technique	Brute Force

FIGURE 5.4. Champs de la règle applicative corrélée

La capture montre la règle `100502`, le niveau 10 et une fréquence de cinq événements. Le groupe `authentication_failures` indique la nature du cas d'usage. L'association MITRE T1110 qualifie la technique de brute force [MITRE, 2026].

TABLE 5.5. Champs utiles pour interpréter la règle

Champ	Interprétation
`rule.id`	Identifiant de la règle déclenchée
`rule.level`	Niveau de criticité Wazuh
`rule.frequency`	Nombre d'occurrences corrélées
`rule.mitre.id`	Technique MITRE associée

5.5 Blocages SSH par OPNsense

Les tentatives SSH issues de `10.10.10.10` sont bloquées par le pare-feu. Wazuh reçoit les journaux et applique la règle `100601`. La répétition déclenche la règle corrélée `100602`, de niveau plus élevé.

	timestamp	agent.name	rule.description	rule.level	rule.id
	Aug 13, 2026 @ 15:07:00.6...	srv-linux-web	Integrity checksum changed.	7	550
	Aug 13, 2026 @ 15:07:00.6...	srv-linux-web	Integrity checksum changed.	7	550
	Aug 13, 2026 @ 15:05:37.5...	srv-windows	Wazuh agent disconnected.	3	504
	Aug 13, 2026 @ 15:03:53.0...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:03:52.0...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:03:51.0...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:03:26.0...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:02:51.9...	wazuh-siem	OPNsense repeated SSH blocks/scan from 10.10.10.10 to 192.168.56.20: suspicious activity.	10	100602
	Aug 13, 2026 @ 15:02:35.8...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:02:27.7...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:02:23.7...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:02:21.7...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:02:20.7...	wazuh-siem	OPNsense repeated SSH blocks/scan from 10.10.10.10 to 192.168.56.20: suspicious activity.	10	100602
	Aug 13, 2026 @ 15:02:19.7...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601
	Aug 13, 2026 @ 15:02:18.7...	wazuh-siem	OPNsense blocked SSH connection from 10.10.10.10 to 192.168.56.20.	7	100601

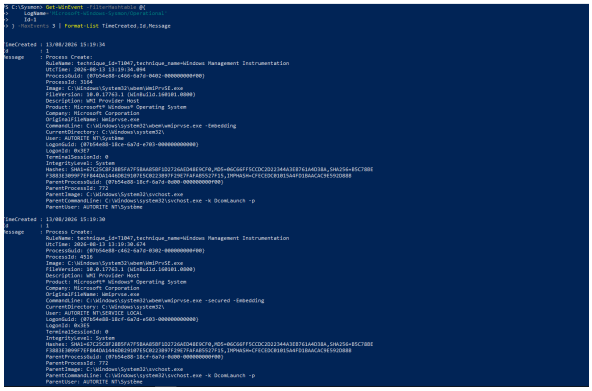
FIGURE 5.5. Blocages SSH OPNsense corrélés dans Wazuh

TABLE 5.6. Logique des règles OPNsense dans Wazuh

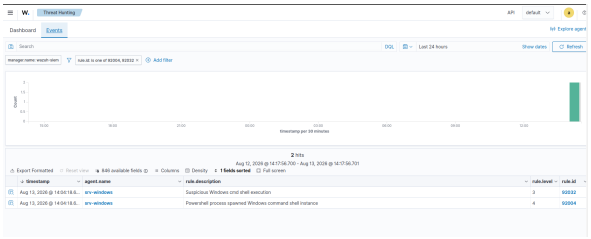
Règle	Niveau	Signification
`100601`	7	Une connexion SSH a été bloquée
`100602`	10	Plusieurs blocages sont corrélés

5.6 Création de processus Windows

Sysmon enregistre la création d'un processus avec l'Event ID 1. L'agent Wazuh transmet ensuite cet événement au serveur de supervision.



(a) Événement Sysmon

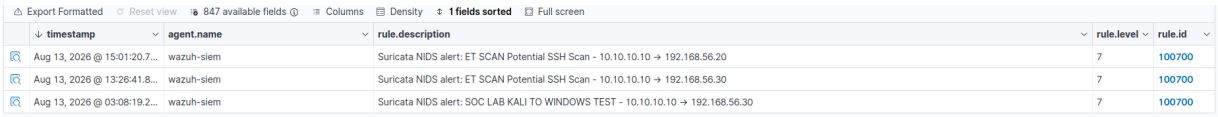


(b) Événement reçu dans Wazuh

FIGURE 5.6. Chaîne de supervision Sysmon vers Wazuh

5.7 Détection d'un scan SSH par Suricata

Le test part de Kali Linux vers le serveur Windows sur le port 22. Suricata détecte la signature *ET SCAN Potential SSH Scan*. L'alerte apparaît dans Wazuh avec la règle `100700`.



timestamp	agent.name	rule.description	rule.level	rule.id
Aug 13, 2026 @ 15:01:20.7…	wazuh-siem	Suricata NIDS alert: ET SCAN Potential SSH Scan - 10.10.10.10 → 192.168.56.20	7	100700
Aug 13, 2026 @ 13:26:41.8…	wazuh-siem	Suricata NIDS alert: ET SCAN Potential SSH Scan - 10.10.10.10 → 192.168.56.30	7	100700
Aug 13, 2026 @ 03:08:19.2…	wazuh-siem	Suricata NIDS alert: SOC LAB KALI TO WINDOWS TEST - 10.10.10.10 → 192.168.56.30	7	100700

FIGURE 5.7. Alerte Suricata reçue dans Wazuh

TABLE 5.7. Éléments vérifiés dans l'alerte Suricata

Élément	Valeur
Signature	ET SCAN Potential SSH Scan
SID	2001219
Source	`10.10.10.10`
Destination	`192.168.56.30:22`
Règle Wazuh	`100700`

5.8 Surveillance de l'intégrité

Une modification effectuée sur un fichier surveillé du serveur Linux génère une alerte FIM. Wazuh affiche le fichier concerné et le type de changement.

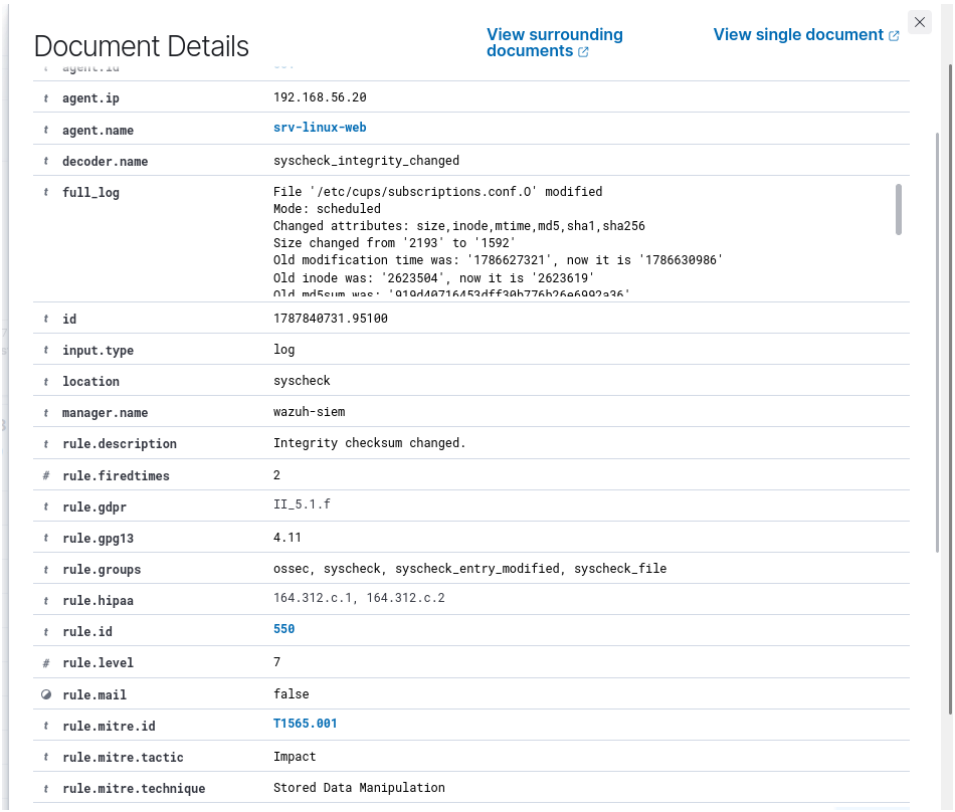


FIGURE 5.8. Modification de fichier détectée par Wazuh FIM

Le FIM calcule une référence pour les fichiers surveillés. Lorsqu'un changement est observé, Wazuh compare l'état courant à cette référence et génère un événement. La règle `550` confirme ici une modification ; elle ne signifie pas automatiquement que le changement est malveillant.

5.9 Supervision IIS

Le serveur Windows héberge une page IIS. Une requête vers une ressource inexistante produit un code 404 qui est collecté puis visible dans Wazuh.

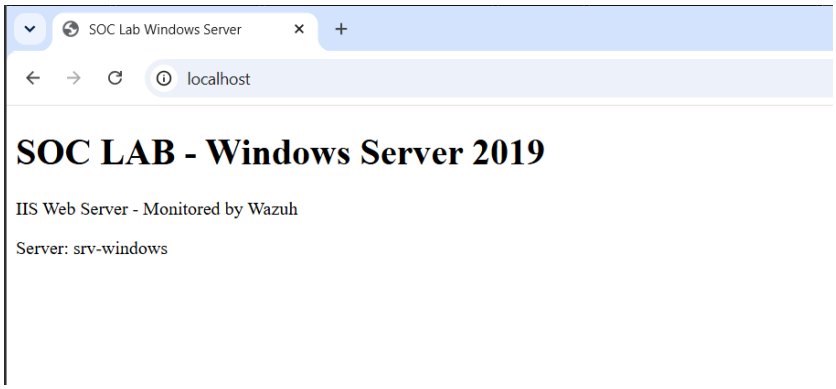


FIGURE 5.9. Page Web IIS du serveur Windows

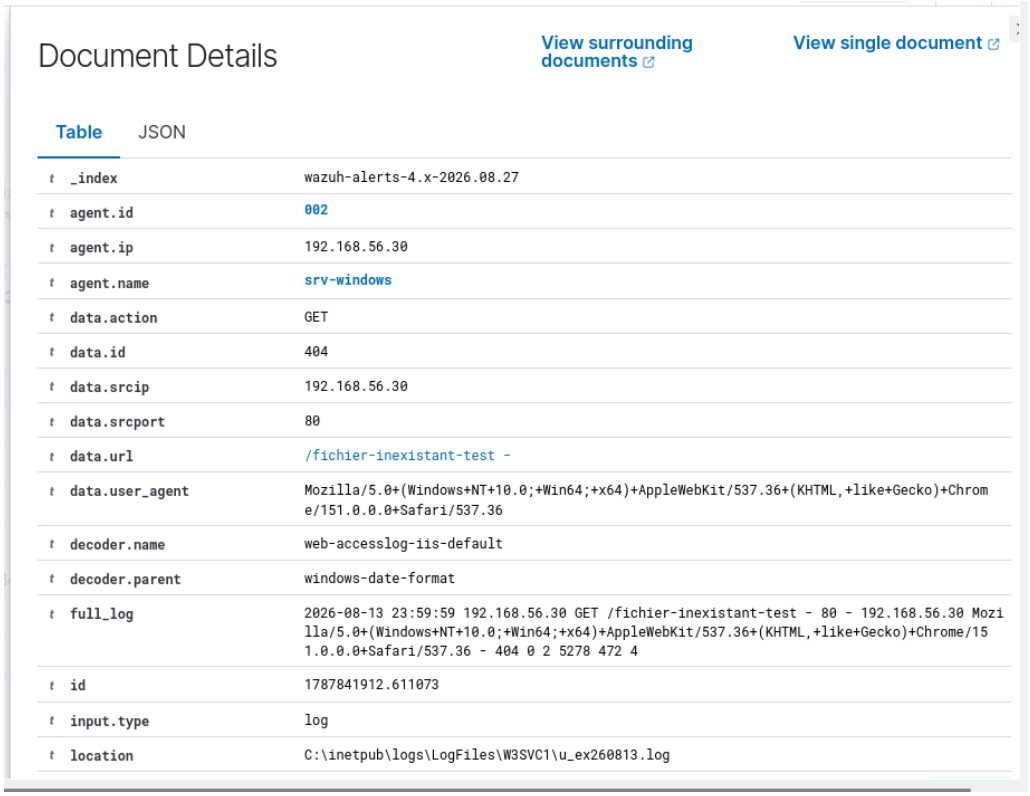


FIGURE 5.10. Requête IIS retournant un code 404 visible dans Wazuh

5.10 Synthèse

Table 5.8. Résultats des validations

Validation	Résultat
Alertes applicatives et corrélation	Validé
Blocage SSH OPNsense et remontée Wazuh	Validé
Événement Sysmon Event ID 1	Validé
Alerte Suricata ET Open	Validé
Modification FIM	Validé
Journal IIS avec code 404	Validé

6 Bilan et perspectives

6.1 Bilan du laboratoire

Le laboratoire obtenu permet de :

— séparer les machines en trois zones réseau ;
— filtrer et journaliser les communications avec OPNsense ;
— centraliser les événements Linux, Windows et réseau dans Wazuh ;
— détecter des événements simples avec Suricata, Sysmon et le FIM ;
— corréler des répétitions d'événements ;
— rechercher les alertes dans Threat Hunting.

Table 6.1. Couverture obtenue

Domaine	Éléments validés
Réseau	Filtrage OPNsense et détection Suricata
Linux	Apache, SSH, application et FIM
Windows	IIS, Sysmon et agent Wazuh
Centralisation	Wazuh, règles et Threat Hunting

6.2 Interprétation technique

Les tests montrent l'intérêt de croiser plusieurs points de vue. OPNsense et Suricata observent le réseau, tandis que les agents Wazuh et Sysmon observent l'activité interne des machines.

Table 6.2. Apport des différentes sources

Source	Ce qu'elle montre	Limite
OPNsense	Flux autorisé ou bloqué	Peu de détails sur l'hôte
Suricata	Signature détectée dans le trafic	Dépend du trafic visible
Sysmon	Processus créé sur Windows	Limité à l'hôte configuré
FIM	Fichier modifié	Ne détermine pas seul l'intention
Application/IIS	Action HTTP et résultat	Dépend de la qualité des journaux

La corrélation apporte un niveau supplémentaire : plusieurs événements faibles peuvent devenir une alerte plus importante lorsqu'ils sont répétés dans une même période.

6.3 Limites

Table 6.3. Principales limites

Limite	Conséquence
Environnement virtualisé	Résultats limités au laboratoire
Peu de machines	Volume d'événements réduit
Architecture centralisée	Pas de haute disponibilité
Scénarios contrôlés	Ne couvrent pas tous les cas réels

6.4 Perspectives

Les améliorations possibles sont :

— ajouter d'autres postes et serveurs ;
— créer de nouveaux cas d'usage et tableaux de bord ;
— sauvegarder les configurations et automatiser le déploiement ;
— étudier une plateforme SOAR pour la réponse aux incidents.

Le SOAR ne fait pas partie de la réalisation actuelle. Le principal résultat reste une chaîne de supervision complète, de la génération d'un événement jusqu'à son affichage dans Wazuh.

6.5 Recommandations

Pour conserver un laboratoire stable et réutilisable, il est recommandé de sauvegarder la configuration OPNsense, les règles Wazuh et les fichiers de configuration des agents. Les horloges des machines doivent également rester synchronisées afin de comparer correctement les événements.

— conserver un instantané propre de chaque machine virtuelle ;
— documenter les ports ouverts et les règles ajoutées ;
— tester une règle après chaque modification ;
— archiver les captures avec la date et le scénario ;
— séparer les essais de validation des essais exploratoires.

Conclusion générale

Ce stage a permis de concevoir et de mettre en place un laboratoire SOC virtualisé autour d'OPNsense, Wazuh, Suricata, Linux et Windows. L'architecture sépare les machines de test, les serveurs et la supervision.

Les validations montrent que les événements réseau et hôte peuvent être centralisés dans Wazuh. Les scénarios retenus couvrent les tentatives SSH, les échecs d'authentification Web, Sysmon, Suricata, le FIM et IIS.

Le laboratoire reste volontairement limité, mais il constitue une base claire pour ajouter de nouvelles machines, de nouvelles règles et des fonctions de réponse aux incidents.

A Vérifications du serveur Linux

A.1 Identité et adressage

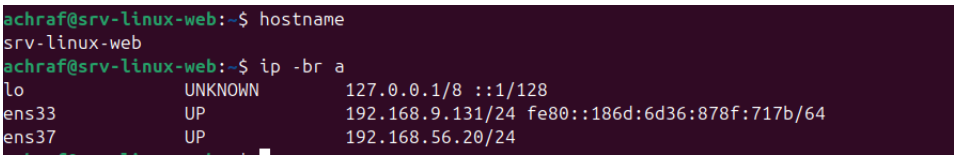
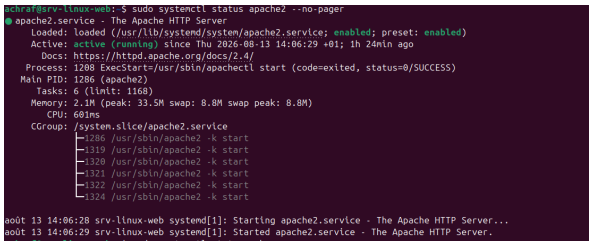
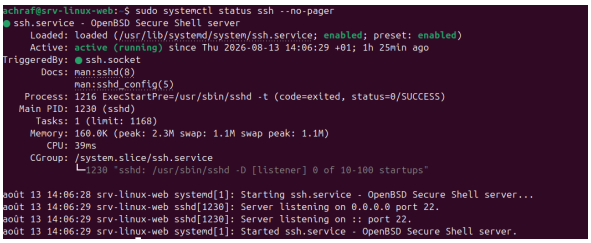


FIGURE A.1. Nom et adresses du serveur Linux

A.2 Services Apache et SSH



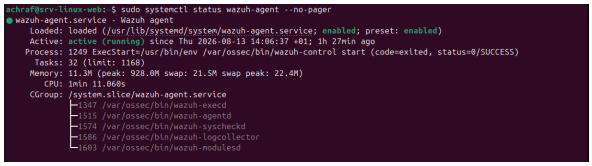
(a) Service Apache actif



(b) Service SSH actif

FIGURE A.2. Vérification des services du serveur Linux

A.3 Agent Wazuh et réponse HTTP



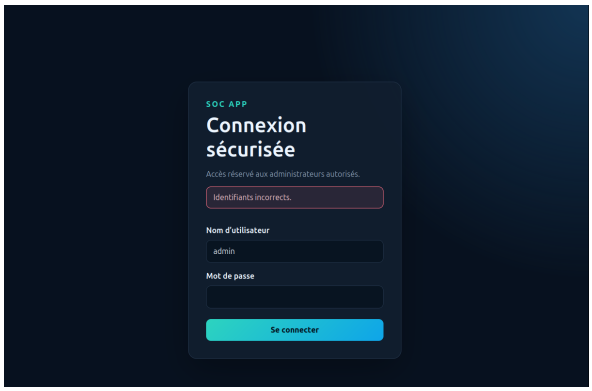
(a) Agent Wazuh actif

(b) Réponse HTTP du serveur

FIGURE A.3. Contrôles complémentaires sur le serveur Linux

B Preuves complémentaires

B.1 Étapes du scénario applicatif



(a) Échec de connexion

(b) Journal de l'application

FIGURE B.1. Éléments du scénario d'authentification

B.2 Erreur IIS côté navigateur

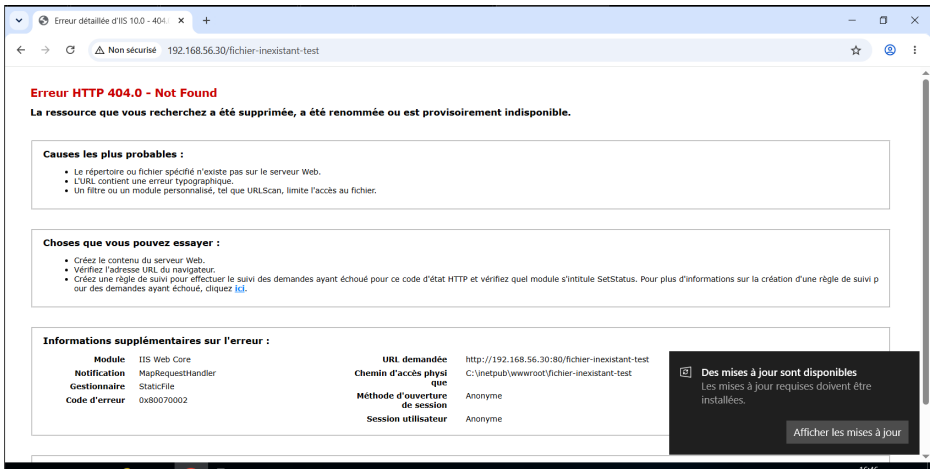


FIGURE B.2. Page d'erreur 404 générée par IIS

C Lecture des événements Wazuh

C.1 Détail d'un événement applicatif

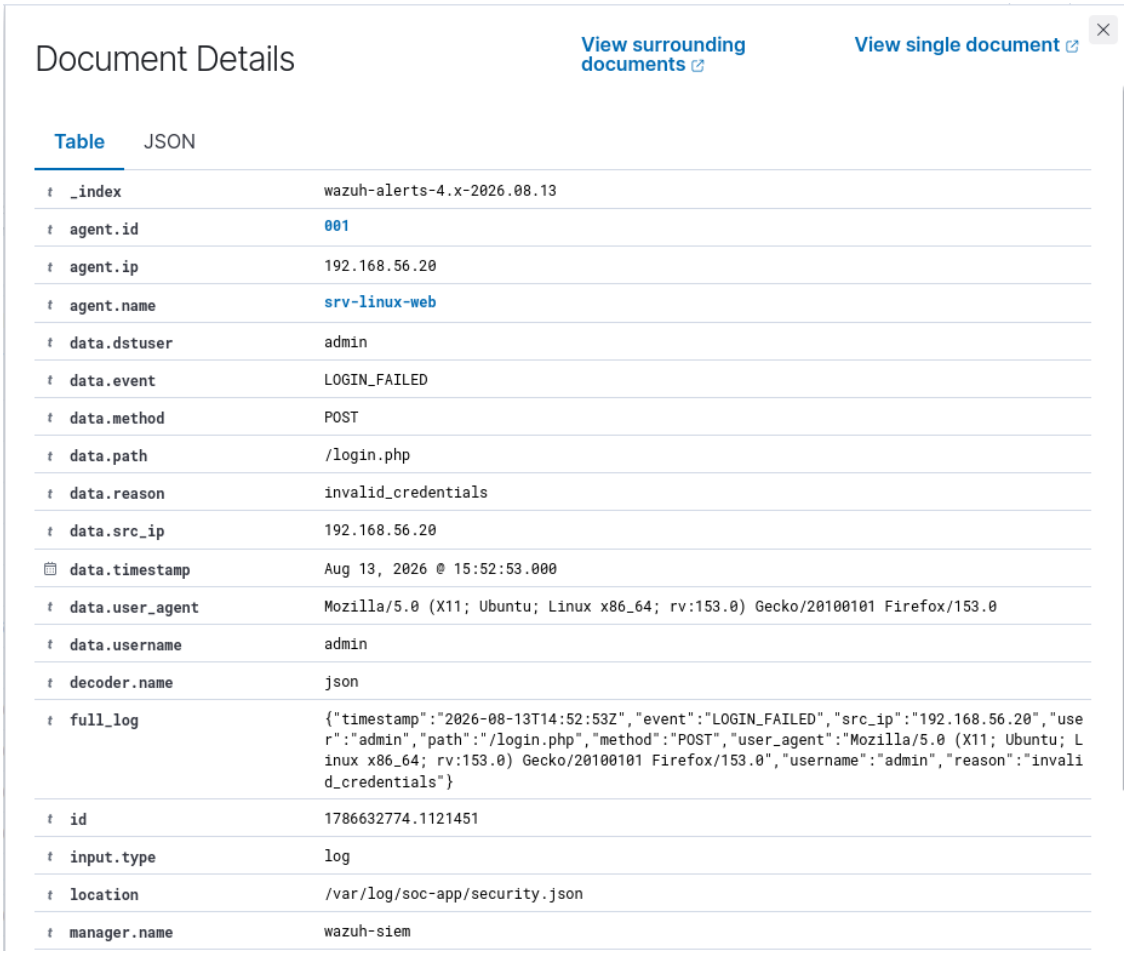


FIGURE C.1. Détail d'un événement LOGIN_FAILED reçu par Wazuh

TABLE C.1. Champs principaux de l'événement

Champ	Utilité
`agent.name`	Machine qui a transmis le journal
`data.event`	Type d'événement applicatif
`data.src_ip`	Adresse associée à l'action
`location`	Fichier de journal surveillé
`full_log`	Ligne complète reçue par Wazuh

C.2 Règle de blocage du pare-feu

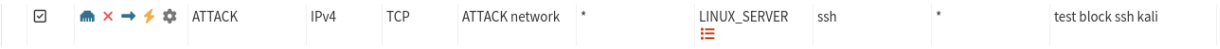


FIGURE C.2. Règle OPNsense bloquant SSH depuis la zone ATTACK

TABLE C.2. Lecture de la règle OPNsense

Élément	Valeur observée
Interface	ATTACK
Protocole	TCP/IPv4
Source	Réseau ATTACK
Destination	Alias du serveur Linux
Port	SSH
Action	Blocage et journalisation

La journalisation est nécessaire pour que Wazuh reçoive la trace du blocage. Sans journal du pare-feu, la règle réseau serait appliquée mais aucune alerte ne pourrait être corrélée dans le SIEM.

D Commandes de validation

D.1 Contrôles utilisés

```
ip -br a
systemctl status apache2 --no-pager
systemctl status ssh --no-pager
systemctl status wazuh-agent --no-pager
curl -I http://192.168.56.20
nmap -sS -p 22 192.168.56.30
```

Table D.1. Objectif des commandes

Commande	Vérification
`ip -br a`	Interfaces et adresses IP
`systemctl status`	État des services Linux
`curl -I`	Code de réponse HTTP
`nmap -sS`	Génération du test de scan SSH

Ces commandes sont exécutées uniquement dans le laboratoire. Elles permettent de confirmer l'état de l'environnement avant de rechercher les alertes dans Wazuh.

Bibliographie

Broadcom. Installing VMware Workstation Pro, 2026. URL https://knowledge.broadcom.com/external/article/387947/installing-vmware-workstation-pro.html. Guide officiel d'installation, consulté le 27 août 2026.

Canonical. Basic installation - Ubuntu Server, 2026. URL https://ubuntu.com/server/docs/tutorial/basic-installation/. Documentation officielle, consultée le 27 août 2026.

CIH Bank. Rapport annuel 2024. Technical report, CIH Bank, 2025. URL https://www.cihbank.ma/sites/default/files/2025-06/Rapport%20annuel%202024_compressed.pdf. Consulté le 27 août 2026.

Kali Linux. Installing Kali Linux, 2025. URL https://www.kali.org/docs/installation/hard-disk-install/. Documentation officielle, consultée le 27 août 2026.

Karen Kent and Murugiah Souppaya. Guide to computer security log management. Technical Report NIST SP 800-92, National Institute of Standards and Technology, 2006. URL https://csrc.nist.gov/pubs/sp/800/92/final.

Microsoft. Install Windows Server from installation media, 2026. URL https://learn.microsoft.com/en-us/windows-server/get-started/install-windows-server. Microsoft Learn, consulté le 27 août 2026.

MITRE. Brute force, technique t1110, 2026. URL https://attack.mitre.org/techniques/T1110/. MITRE ATT&CK, consulté le 27 août 2026.

Open Information Security Foundation. Suricata user guide : Eve json output, 2026. URL https://docs.suricata.io/en/latest/output/eve/eve-json-output.html. Consulté le 27 août 2026.

OpenAI. Assistant IA : ChatGPT, 2026. URL https://learn.chatgpt.com/docs/use-chatgpt. Documentation officielle de ChatGPT, consultée le 27 août 2026.

OPNsense. Initial Installation and Configuration, 2026a. URL https://docs.opnsense.org/manual/install.html. Documentation officielle, consultée le 27 août 2026.

OPNsense. Firewall rules, 2026b. URL https://docs.opnsense.org/manual/firewall.html. Consulté le 27 août 2026.

Mark Russinovich and Thomas Garnier. Sysmon – sysinternals, 2026. URL https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon. Microsoft Learn, consulté le 27 août 2026.

Wazuh. Quickstart installation, 2026a. URL https://documentation.wazuh.com/current/quickstart.html. Documentation officielle, consultée le 27 août 2026.

Wazuh. Architecture – getting started with wazuh, 2026b. URL https://documentation.wazuh.com/current/getting-started/architecture.html. Consulté le 27 août 2026.