

RAPPORT DE LABORATOIRE

Intégration Wazuh SIEM / Suricata NIDS

Déploiement, Configuration et Simulation d'Attaques

Nom	Rôle
Echahbouni Issam	Administrateur Wazuh Manager — 192.168.56.101
Qiyaoui Mohamed Reda	Machine cible (Wazuh Agent + Suricata) — 192.168.56.107
Abdeluahab Majida	Attaquante 1 — Kali Linux — 192.168.56.104
Benmlih-Taya Mehdi	Attaquant 2 — Kali Linux ----192.168.56.108

Date : 2 mai 2026

EST Tétouan — Année académique 2025–2026

Table des matières

1. Introduction Générale.....	5
1.1 Présentation de Wazuh.....	5
1.2 Le Tableau de Bord Wazuh	5
2. Topologie du Laboratoire	5
2.1 Architecture Réseau	5
2.2 Répartition des rôles.....	6
3. Déploiement et Configuration	6
3.1 Déploiement de l'Agent Wazuh.....	6
Étape 1 — Sélection de la cible et génération de la commande.....	6
Étape 2 — Exécution et démarrage du service.....	8
3.2 Installation et Configuration de Suricata.....	8
Étape 1 — Installation du paquet Suricata	8
Étape 2 — Téléchargement des règles Emerging Threats.....	9
Étape 3 — Configuration de suricata.yaml	10
Étape 4 — Intégration avec l'Agent Wazuh (ossec.conf)	11
4. Simulation et Détection des Attaques	11
4.1 Reconnaissance initiale — Ping ICMP	11
Action offensive	11
Détection Wazuh / Suricata	12
4.2 Scan de ports avancé — Nmap	12
Action offensive	12
Détection Wazuh / Suricata	13
4.3 Analyse de vulnérabilités Web — Nikto	14
Action offensive	14
Détection Wazuh / Suricata	14
4.4 Requête HTTP suspecte — cURL	14
Action offensive	14
Détection Wazuh / Suricata	15
4.5 Force Brute SSH — Hydra	15
Action offensive (Majida)	15
Détection Wazuh / Suricata	15
4.6 Force Brute FTP — Hydra.....	16
Action offensive	16
Détection Wazuh / Suricata	16
4.7 Force Brute RDP — Hydra.....	16
Action offensive (Mehdi)	17
Détection Wazuh / Suricata	17
4.8 Injection SQL — SQLMap (DVWA).....	17
Action offensive (Majida)	17
Détection Wazuh / Suricata	18
4.9 Attaque DoS par Flood SYN — hping3	18
Action offensive	18
Détection Wazuh / Suricata	19
4.10 Élévation de Privilèges — sudo	19
Action post-exploitation	19
Détection Wazuh — FIM et Audit	19

4.11 Transmission de Mots de Passe en Clair — HTTP POST	20
5. Intégration pfSense — Firewall et Remote	20
Étape 1 — Démarrage et vérification de pfSense	21
Étape 2 — Vérification du tableau de bord pfSense.....	21
Étape 3 — Configuration du Remote Syslog vers Wazuh	23
Étape 4 — Règles de Firewall OPT.....	23
6. Conclusion	24
Apports de Suricata.....	24
Apports de Wazuh.....	24
Limites observées.....	24
Bilan général	24

Liste des figures

Section 1 — Déploiement et Configuration

- Figure 1 — Interface Deploy new agent : sélection du paquet et configuration du serveur
- Figure 2 — Commandes générées par le Dashboard : téléchargement et démarrage de l'agent
- Figure 3 — Terminal Ubuntu : téléchargement, installation du paquet wazuh-agent 4.14.4-1 et activation systemd
- Figure 4 — Terminal : installation de Suricata via apt-get avec résolution des dépendances
- Figure 5 — Extraction de l'archive Emerging Threats : 40+ catégories de règles
- Figure 6 — suricata.yaml : définition de HOME_NET sur le sous-réseau 192.168.56.0/24
- Figure 7 — suricata.yaml : default-rule-path et chargement de toutes les règles
- Figure 8 — suricata.yaml : interface d'écoute enp0s3
- Figure 9 — ossec.conf : bloc de collecte du fichier eve.json de Suricata

Section 2 — Simulation et Détection des Attaques

- Figure 10 — Commande ping depuis Kali : 4 paquets ICMP transmis avec 0% packet loss
- Figure 11 — Dashboard Wazuh : 4 alertes "GPL ICMP_INFO PING *NIX" (rule.id 86601, level 3)
- Figure 12 — Résultats Nmap -A : ports 21/FTP, 22/SSH, 80/HTTP, 3389/RDP découverts
- Figure 13 — Dashboard : burst d'alertes Nmap (ET SCAN Possible Nmap User-Agent...)
- Figure 14 — Nikto v2.6.0 : Apache 2.4.58 obsolète, en-têtes sécurité manquants (CVE-2003-1418)
- Figure 15 — Dashboard : alertes GPL EXPLOIT, ET WEB_SPECIFIC_APPS (JBoss, WordPress, LFI...)
- Figure 16 — curl vers la cible : réponse 404 Apache 2.4.58, accès direct à une IP nue
- Figure 17 — Dashboard : alerte "ET HUNTING curl User-Agent to Dotted Quad"
- Figure 18 — Hydra v9.6 : attaque SSH avec rockyou.txt, 14 344 399 tentatives
- Figure 19 — Dashboard : cascade d'alertes SSH BruteForce (level 10)
- Figure 20 — Hydra attaquant le service FTP (port 21) avec le compte admin
- Figure 21 — Dashboard : rafale d'alertes "ET SCAN Potential FTP Brute-Force attempt"
- Figure 22 — Hydra RDP : 1 mot de passe valide trouvé pour le compte administrator (123456)
- Figure 23 — Dashboard : alertes RDP (ET POLICY MS Remote Desktop Administrator Login Request)
- Figure 24 — SQLMap 1.10.4 : paramètre id vulnérable (Boolean-blind, Error-based, UNION, Time-based)
- Figure 25 — Dashboard : 210+ alertes SQLi (ET WEB_SERVER SQL Injection, Information Schema...)
- Figure 26 — hping3 flood SYN : 919 549 paquets transmis, 100% packet loss
- Figure 27 — Dashboard : alertes level 9 "Agent event queue is full. Events may be lost"
- Figure 28 — Terminal cible : escalade sudo vers root, puis modification de /etc/hosts
- Figure 29 — Dashboard : "Successful sudo to ROOT" (rule 5402) et "Integrity checksum changed" (rule 550)
- Figure 30 — Dashboard : alertes "ET POLICY HTTP POST contains pass= in cleartext"
- Figure 31 — Console pfSense 2.8.1 : interfaces WAN/LAN/OPT1
- Figure 32 — Tableau de bord pfSense : version, ressources, état
- Figure 33 — pfSense Remote Logging : syslog UDP/514 vers Wazuh Manager
- Figure 34 — Règles pfSense OPT1 : blocage Kali spécifique et autorisation OPT1"

Liste des Tableaux

- Tableau 1 — Équipe du laboratoire : noms et rôles (page de garde)
- Tableau 2 — Répartition des rôles : membres, machines, adresses IP et fonctions (§ 2.2)

1. Introduction Générale

1.1 Présentation de Wazuh

Wazuh est une plateforme open-source de sécurité unifiée qui combine les capacités d'un **SIEM** (Security Information and Event Management) et d'un **XDR** (Extended Detection and Response). Elle offre une visibilité en temps réel sur les événements de sécurité d'un parc de machines hétérogènes, qu'elles soient physiques, virtuelles ou hébergées dans le cloud.

Architecture agent/serveur :

- **Wazuh Manager (Serveur) :** Le cerveau central. Il reçoit, analyse et corrèle tous les événements remontés par les agents. Il héberge le moteur de règles et gère les alertes.
- **Wazuh Agent :** Un démon léger installé sur chaque endpoint surveillé. Il collecte les journaux système, surveille l'intégrité des fichiers (FIM), détecte les vulnérabilités et transmet les données au Manager via un canal chiffré.
- **Wazuh Indexer :** Basé sur OpenSearch, il stocke et indexe les données d'alertes pour une recherche rapide.
- **Wazuh Dashboard :** L'interface graphique web centrale permettant aux analystes de visualiser les alertes et de superviser le parc.

1.2 Le Tableau de Bord Wazuh

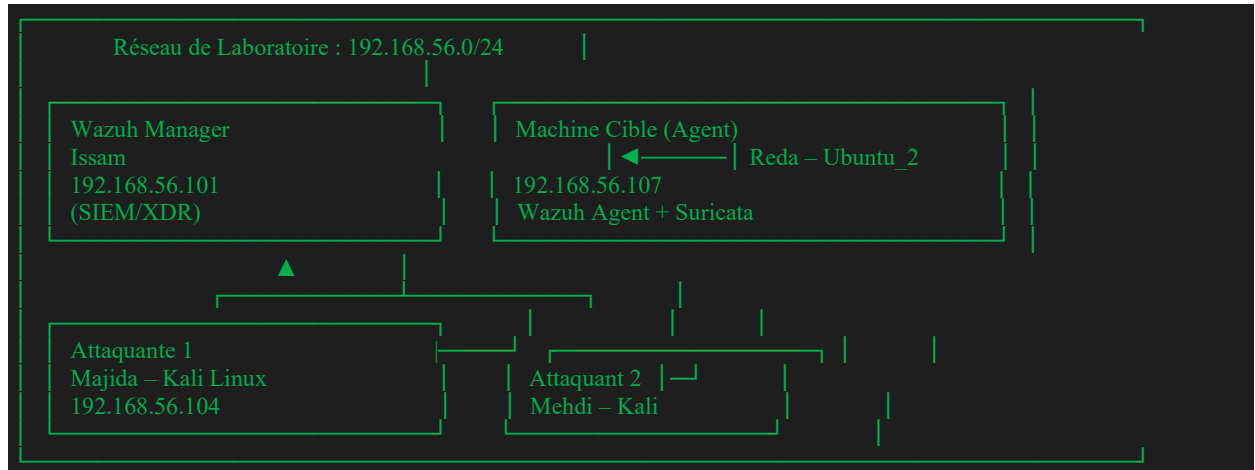
Le Wazuh Dashboard est l'interface de supervision utilisée par l'administrateur (Issam) tout au long de ce projet. Il propose plusieurs modules essentiels :

- **Security Events :** Vue agrégée de toutes les alertes classées par niveau de criticité (0 à 15), par agent, par règle et par horodatage. Vue principale utilisée pour l'observation des attaques.
- **Threat Hunting :** Permet des recherches avancées dans les logs pour identifier des comportements suspects.
- **File Integrity Monitoring (FIM) :** Détecte les modifications non autorisées sur les fichiers systèmes critiques.
- **Vulnerability Detector :** Inventorie les CVEs présentes sur les machines supervisées.
- **MITRE ATT&CK :** Cartographie les alertes sur le framework MITRE pour contextualiser les tactiques et techniques adversariales.

2. Topologie du Laboratoire

2.1 Architecture Réseau

L'environnement de test est un réseau isolé de type Host-Only / NAT sous VirtualBox, simulant un réseau d'entreprise. Toutes les machines communiquent sur le sous-réseau 192.168.56.0/24.



2.2 Répartition des rôles

Membre	Machine	IP	Rôle
Echahbouni Issam	Wazuh Manager	192.168.56.101	Administration SIEM, supervision des alertes, rédaction du guide
Qiyaoui Mohamed Reda	Ubuntu (Agent)	192.168.56.107	Machine cible : FTP, SSH, HTTP, RDP — Wazuh Agent + Suricata
Abdeluahab Majida	Kali Linux	192.168.56.104	Attaquante : scans, force brute, injection SQL, DoS, élévation
Benmlih-Taya Mehdi	Kali Linux	—	Attaquant : scans réseau, brute-force FTP/SSH/RDP

3. Déploiement et Configuration

3.1 Déploiement de l'Agent Wazuh

Étape 1 — Sélection de la cible et génération de la commande

Depuis l'interface Wazuh Dashboard d'Issam (Endpoints → Deploy new agent), la configuration suivante a été saisie : Système Linux/DEB amd64, adresse Manager 192.168.56.101, nom d'agent Ubuntu_2.

Deploy new agent

✓ **Select the package to download and install on your system:**

LINUX

☐ RPM amd64 ☐ RPM aarch64

☒ DEB amd64 ☐ DEB aarch64

① For additional systems and architectures, please check our [documentation](#).

✓ **Server address:**

This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name.

Assign a server address ①

192.168.56.101

☐ Remember server address

✓ **Optional settings:**

By default, the deployment uses the hostname as the agent name. Optionally, you can use a different agent name.

Assign an agent name ①

Ubuntu_2

Figure 1 — Interface Deploy new agent : sélection du paquet et configuration du serveur

Le Dashboard génère automatiquement la commande d'installation paramétrique :

```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.14.4-1_amd64.deb \
&& sudo WAZUH_MANAGER='192.168.56.101' WAZUH_AGENT_NAME='Ubuntu_2' \
dpkg -i ./wazuh-agent_4.14.4-1_amd64.deb
```

4 Run the following commands to download and install the agent:

```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.14.4-1_amd64.deb && sudo WAZUH_MANAGER='192.168.56.101' WAZUH_AGENT_NAME='Ubuntu_2' dpkg -i ./wazuh-agent_4.14.4-1_amd64.deb
```

① **Requirements**

- You will need administrator privileges to perform this installation.
- Shell Bash is required.

Keep in mind you need to run this command in a Shell Bash terminal.

5 Start the agent:

```
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

Figure 2 — Commandes générées par le Dashboard : téléchargement et démarrage de l'agent

Les variables WAZUH MANAGER et WAZUH AGENT NAME sont injectées dans dpkg lors de l'installation, configurant automatiquement le fichier /var/ossec/etc/ossec.conf sans intervention manuelle post-installation.

Étape 2 — Exécution et démarrage du service

Sur la machine de Reda, la commande a été exécutée dans un terminal Bash. La sortie confirme le téléchargement (12.61 Mo à 9.29 MB/s), l'installation et l'activation du service systemd.

```
ubuntu2@ubuntu:~$ wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.14.4-1_and64.deb && sudo WAZUH_MANAGER='192.168.56.101' WAZUH_AGENT_NAME='Ubuntu2' dpkg
--2026-05-02 13:28:42-- https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.14.4-1_and64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 99.84.9.57, 99.84.9.126, 99.84.9.92, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|99.84.9.57|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13220908 (13M) [application/vnd.debian.binary-package]
Saving to: 'wazuh-agent_4.14.4-1_and64.deb'

wazuh-agent_4.14.4-1_and64.deb      100%[=====]
2026-05-02 13:28:54 (9.29 MB/s) - 'wazuh-agent_4.14.4-1_and64.deb' saved [13220908/13220908]

[sudo: authenticate] Password:
Selecting previously unselected package wazuh-agent.
(Reading database... 191803 files and directories currently installed.)
Preparing to unpack ./wazuh-agent_4.14.4-1_and64.deb...
Unpacking wazuh-agent (4.14.4-1)...
Setting up wazuh-agent (4.14.4-1)...
ubuntu2@ubuntu:~$ sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink '/etc/systemd/system/multi-user.target.wants/wazuh-agent.service' → '/usr/lib/systemd/system/wazuh-agent.service'.
ubuntu2@ubuntu:~$
```

Figure 3 — Terminal Ubuntu : téléchargement, installation du paquet wazuh-agent 4.14.4-1 et activation systemd

```
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

3.2 Installation et Configuration de Suricata

La procédure suivie est celle du guide rédigé par Issam (Suricata NIDS Configuration & Wazuh SIEM Integration).

Étape 1 — Installation du paquet Suricata

```
sudo apt-get update
sudo apt-get install suricata -y
```

```
ubuntu2@Ubuntu:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu resolute-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu resolute InRelease
Hit:3 http://archive.ubuntu.com/ubuntu resolute-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu resolute-backports InRelease
Reading package lists... Done
ubuntu2@Ubuntu:~$ sudo apt-get install suricata -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Solving dependencies... Done
The following packages were automatically installed and are no longer required:
  linux-headers-7.0.0-14 linux-headers-7.0.0-14-generic linux-image-unsigned-7.0.0-14-g
Use 'sudo apt autoremove' to remove them.
```

Figure 4 — Terminal : installation de Suricata via apt-get avec résolution des dépendances

Étape 2 — Téléchargement des règles Emerging Threats

```
sudo mkdir -p /etc/suricata/rules
wget https://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz
tar -zxvf emerging.rules.tar.gz
sudo cp rules/*.rules /etc/suricata/rules/
```

```
Processing triggers for libc-bin (2.43-2ubuntu2)...
shant@shant:~$ sudo mkdir -p /etc/suricata/rules
shant@shant:~$ wget https://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz
--2026-05-02 13:40:36-- https://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz
Resolving rules.emergingthreats.net (rules.emergingthreats.net)... 35.153.32.75, 13.210.147.68, 52.86.126.48, ...
Connecting to rules.emergingthreats.net (rules.emergingthreats.net)|35.153.32.75|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5469944 (5.2M) [application/octet-stream]
Saving to: 'emerging.rules.tar.gz'

emerging.rules.tar.gz           100%[=====] 5.22M  4.02MB/s   in 1.3s

2026-05-02 13:40:36 (4.02 MB/s) - 'emerging.rules.tar.gz' saved [5469944/5469944]

shant@shant:~$ tar -zxvf emerging.rules.tar.gz
rules/
rules/bsd-license.txt
rules/LICENSE
rules/botcc-portgroup.rules
rules/botcc.rules
rules/ciarmy.rules
rules/classification.config
rules/compromised-ips.txt
rules/compromised.rules
rules/drop.rules
rules/dshield.rules
rules/emerging-active.rules
rules/emerging-ahave-pop.rules
rules/emerging-attack_response.rules
rules/emerging-chat.rules
rules/emerging-cobminer.rules
rules/emerging-current_events.rules
rules/emerging-deleted.rules
rules/emerging-dns.rules
rules/emerging-dos.rules
rules/emerging-exploit.rules
rules/emerging-exploit_kit.rules
rules/emerging-ftp.rules
rules/emerging-games.rules
rules/emerging-hunting.rules
rules/emerging-icmp.rules
rules/emerging-icmp_info.rules
rules/emerging-icmp.rules
rules/emerging-inappropriate.rules
rules/emerging-info.rules
rules/emerging-isa.rules
rules/emerging-malware.rules
rules/emerging-misc.rules
rules/emerging-mobile_malware.rules
rules/emerging-netbios.rules
rules/emerging-p2p.rules
rules/emerging-phishing.rules
rules/emerging-policy.rules
rules/emerging-pool.rules
rules/emerging-retired.rules
rules/emerging-rpc.rules
rules/emerging-scan.rules
rules/emerging-shellcode.rules
rules/emerging-smtp.rules
rules/emerging-smp.rules
rules/emerging-sql.rules
rules/emerging-telnet.rules
rules/emerging-tftp.rules
rules/emerging-user_agents.rules
rules/emerging-web_client.rules
rules/emerging-web_server.rules
rules/emerging-web_specific_apps.rules
rules/emerging-worm.rules
rules/gpl-2.0.txt
rules/isa-mg-map
rules/suricata-5.0-enhanced-open.txt
rules/threatview_C2.rules
rules/tor.rules
shant@shant:~$ sudo cp rules/* /etc/suricata/rules/
shant@shant:~$
```

Figure 5 — Extraction de l'archive Emerging Threats : 40+ catégories de règles (scan, exploit, dos, web_server, malware...)

Étape 3 — Configuration de suricata.yaml

Trois paramètres critiques ont été modifiés dans le fichier de configuration principal :

```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.56.0/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    #EXTERNAL_NET: "!$HOME_NET"
    EXTERNAL_NET: "any"
```

Figure 6 — suricata.yaml : définition de HOME_NET sur le sous-réseau 192.168.56.0/24

```
default-rule-path: /var/lib/suricata/rules

rule-files:
  - *.rules
```

Figure 7 — suricata.yaml : default-rule-path et chargement de toutes les règles (*.rules)

```
# Linux high speed capture support
af-packet:
  - interface: enp0s3
    # Number of receive threads. "auto" will use the number of
```

Figure 8 — suricata.yaml : interface d'écoute enp0s3 (interface réseau active de la machine Ubuntu)

*HOME NET définit le périmètre protégé. Le wildcard *.rules charge automatiquement toutes les règles du répertoire. L'interface enp0s3 est celle identifiée via 'ip a' sur la machine de Reda.*

Étape 4 — Intégration avec l'Agent Wazuh (ossec.conf)

Le bloc suivant a été ajouté dans /var/ossec/etc/ossec.conf pour que l'agent lise le fichier d'alertes JSON de Suricata :

```
<localfile>
  <log_format>json</log_format>
  <location>/var/log/suricata/eve.json</location>
</localfile>
```

```
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/dpkg.log</location>
</localfile>
<localfile>
  <log_format>json</log_format>
  <location>/var/log/suricata/eve.json</location>
</localfile>
</ossec_config>
```

Figure 9 — ossec.conf : bloc de collecte du fichier eve.json de Suricata, déclenché par le format JSON

Suricata écrit ses alertes dans /var/log/suricata/eve.json au format EVE (Extensible Event Format). L'agent Wazuh parse chaque entrée JSON, l'enrichit et la transmet au Manager où la règle 86601 la transforme en alerte Dashboard.

4. Simulation et Détection des Attaques

Les attaques ont été menées sur la machine cible de Reda (Ubuntu_2 — 192.168.56.107) exposant les services FTP (21), SSH (22), HTTP (80) et RDP (3389).

4.1 Reconnaissance initiale — Ping ICMP

Action offensive

```
ping 192.168.56.107
```

```
(kali@kali)-[~]
$ ping 192.168.56.107
PING 192.168.56.107 (192.168.56.107) 56(84) bytes of data.
64 bytes from 192.168.56.107: icmp_seq=1 ttl=64 time=5.80 ms
64 bytes from 192.168.56.107: icmp_seq=2 ttl=64 time=0.990 ms
64 bytes from 192.168.56.107: icmp_seq=3 ttl=64 time=0.760 ms
64 bytes from 192.168.56.107: icmp_seq=4 ttl=64 time=0.945 ms
^C
--- 192.168.56.107 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 0.760/2.124/5.801/2.124 ms
```

Figure 10 — Commande ping depuis Kali : 4 paquets ICMP transmis avec 0% packet loss

Détection Wazuh / Suricata

timestamp	agent.name	rule.description	rule.level	rule.id
May 2, 2026 @ 12:18:53.667	Ubuntu	Suricata: Alert - GPL ICMP_INFO PING *NIX	3	86601
May 2, 2026 @ 12:18:52.867	Ubuntu	Suricata: Alert - GPL ICMP_INFO PING *NIX	3	86601
May 2, 2026 @ 12:18:51.667	Ubuntu	Suricata: Alert - GPL ICMP_INFO PING *NIX	3	86601
May 2, 2026 @ 12:18:50.868	Ubuntu	Suricata: Alert - GPL ICMP_INFO PING *NIX	3	86601

Figure 11 — Dashboard Wazuh : 4 alertes 'GPL ICMP_INFO PING *NIX' (rule.id 86601, level 3)

La règle Emerging Threats GPL ICMP_INFO PING *NIX a déclenché une alerte par paquet ICMP, horodatée à la milliseconde.

4.2 Scan de ports avancé — Nmap

Action offensive

```
nmap -A 192.168.56.107
```

```
$ nmap -A 192.168.56.107
Starting Nmap 7.98 ( https://nmap.org ) at 2026-05-02 08:15 -0400
Nmap scan report for 192.168.56.107
Host is up (0.0013s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.5
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.15 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 256 93:a8:29:4f:ce:e9:40:d7:9d:e7:d5:18:83:d9:5e:a9 (ECDSA)
|_ 256 3c:f1:67:5a:bc:9a:2f:96:0f:8d:09:63:27:f3:74:2d (ED25519)
80/tcp    open  http     Apache httpd 2.4.58 ((Ubuntu))
|_ http-title: Apache2 Ubuntu Default Page: It works
|_ http-server-header: Apache/2.4.58 (Ubuntu)
3389/tcp  open  ms-wbt-server Microsoft Terminal Service
MAC Address: 08:00:27:5B:55:3A (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19
Network Distance: 1 hop
Service Info: OSs: Unix, Linux, Windows; CPE: cpe:/o:linux:linux_kernel, cpe:/o:microsoft:windows

TRACEROUTE
HOP RTT ADDRESS
1 1.33 ms 192.168.56.107

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 19.43 seconds
```

Figure 12 — Résultats Nmap -A : ports 21/FTP, 22/SSH, 80/HTTP, 3389/RDP découverts, OS Linux 4.15-5.19

Détection Wazuh / Suricata

timestamp	agent.name	rule.description	rule.level	rule.id
May 2, 2026 @ 13:16:12.193	Ubuntu	Suricata: Alert - ET SCAN Potential SSH Scan OUTBOUND	3	86601
May 2, 2026 @ 13:16:12.184	Ubuntu	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed	3	86601
May 2, 2026 @ 13:16:12.172	Ubuntu	Suricata: Alert - ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	3	86601
May 2, 2026 @ 13:16:12.163	Ubuntu	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed	3	86601
May 2, 2026 @ 13:16:12.163	Ubuntu	Suricata: Alert - ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	3	86601
May 2, 2026 @ 13:16:10.645	Ubuntu	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed	3	86601
May 2, 2026 @ 13:16:10.642	Ubuntu	Suricata: Alert - ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	3	86601
May 2, 2026 @ 13:16:10.633	Ubuntu	PAM: User login failed.	5	5503
May 2, 2026 @ 13:16:10.628	Ubuntu	PAM: User login failed.	5	5503
May 2, 2026 @ 13:16:10.619	Ubuntu	PAM: User login failed.	5	5503
May 2, 2026 @ 13:16:10.530	Ubuntu	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed	3	86601
May 2, 2026 @ 13:16:10.528	Ubuntu	Suricata: Alert - ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	3	86601
May 2, 2026 @ 13:16:10.519	Ubuntu	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed	3	86601
May 2, 2026 @ 13:16:10.516	Ubuntu	Suricata: Alert - ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	3	86601
May 2, 2026 @ 13:16:10.507	Ubuntu	Suricata: Alert - ET SCAN Possible Nmap User-Agent Observed	3	86601

Figure 13 — Dashboard : burst d'alertes Nmap (ET SCAN Possible Nmap User-Agent, Scripting Engine, SSH Scan OUTBOUND)

Les alertes identifient le User-Agent caractéristique du Nmap Scripting Engine, révélant l'outil utilisé par l'attaquant.

4.3 Analyse de vulnérabilités Web — Nikto

Action offensive

```
nikto -h http://192.168.56.107
```

```

Session Actions Edit View Help
(kali@kali)~$ nikto -h http://192.168.56.107
- Nikto v2.6.0

+ Your Nikto installation is out of date.
+ Target IP: 192.168.56.107
+ Target Hostname: 192.168.56.107
+ Target Port: 80
+ Platform: Linux/Unix
+ Start Time: 2026-04-27 10:29:25 (GMT-4)

+ Server: Apache/2.4.58 (Ubuntu)
+ No CGI Directories found (use '-C all' to force check all possible dirs). CGI tests skipped.
+ [600050] Apache/2.4.58 appears to be outdated (current is at least 2.4.66).
+ [740000] Multiple index files found (all unique): /index.php, /index.html.
+ [999984] /: Server may leak inodes via ETags, header found with file //. inode: 29af, size: 6503679082221, mtime: gzip. See: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ [013587] /: Suggested security header missing: strict-transport-security. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ [013587] /: Suggested security header missing: permissions-policy. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Permissions-Policy
+ [013587] /: Suggested security header missing: x-content-type-options. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options
+ [013587] /: Suggested security header missing: content-security-policy. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP
+ [013587] /: Suggested security header missing: referrer-policy. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy
+ [999990] OPTIONS: Allowed HTTP Methods: OPTIONS, HEAD, GET, POST .

```

Figure 14 — Nikto v2.6.0 : Apache 2.4.58 obsolète, en-têtes sécurité manquants, méthodes permissives (CVE-2003-1418)

Détection Wazuh / Suricata

Apr 27, 2026 @ 15:30:13.562	Ubuntu	Suricata: Alert - GPL EXPLOIT .cnf access	3	86601
Apr 27, 2026 @ 15:30:13.554	Ubuntu	Suricata: Alert - GPL EXPLOIT .cnf access	3	86601
Apr 27, 2026 @ 15:30:13.548	Ubuntu	Suricata: Alert - GPL EXPLOIT .cnf access	3	86601
Apr 27, 2026 @ 15:30:13.176	Ubuntu	Suricata: Alert - ET INFO Dotted Quad Host DLL Request	3	86601
Apr 27, 2026 @ 15:30:13.148	Ubuntu	Suricata: Alert - GPL WEB_SERVER author.exe access	3	86601
Apr 27, 2026 @ 15:30:13.146	Ubuntu	Suricata: Alert - ET INFO Executable Download from dotted-quad Host	3	86601
Apr 27, 2026 @ 15:30:13.136	Ubuntu	Suricata: Alert - ET INFO Executable Download from dotted-quad Host	3	86601
Apr 27, 2026 @ 15:30:13.122	Ubuntu	Suricata: Alert - ET INFO Dotted Quad Host DLL Request	3	86601
Apr 27, 2026 @ 15:30:12.672	Ubuntu	Suricata: Alert - ET WEB_SPECIFIC_APPS Possible JBoss JMX Console Beanshell Deployer WAR Upload and Depl...	3	86601
Apr 27, 2026 @ 15:30:12.655	Ubuntu	Suricata: Alert - ET WEB_SERVER ColdFusion componentutils access	3	86601
Apr 27, 2026 @ 15:30:11.526	Ubuntu	Suricata: Alert - ET WEB_SPECIFIC_APPS Horde type Parameter Local File Inclusion Attempt	3	86601
Apr 27, 2026 @ 15:30:10.933	Ubuntu	Suricata: Alert - ET WEB_SPECIFIC_APPS Request to Wordpress W3TC Plug-in dbcache Directory	3	86601
Apr 27, 2026 @ 15:30:10.924	Ubuntu	Suricata: Alert - ET WEB_SPECIFIC_APPS Request to Wordpress W3TC Plug-in dbcache Directory	3	86601
Apr 27, 2026 @ 15:30:10.802	Ubuntu	Suricata: Alert - ET WEB_SPECIFIC_APPS Wordpress LifeSpeed Cache Plugin debug.log Access Attempt (CVE-2...	3	86601
Apr 27, 2026 @ 15:30:10.766	Ubuntu	Suricata: Alert - ET WEB_SERVER /etc/shadow Detected in URI	3	86601

Figure 15 — Dashboard : alertes GPL EXPLOIT, ET WEB_SPECIFIC_APPS (JBoss, WordPress, ColdFusion, LFI, /etc/shadow...)

La règle 5551 (PAM: Multiple failed logins) atteint le niveau 10 — criticité élevée — signalant une attaque par force brute active. Corrélé avec ET SCAN LibSSH Frequent SSH Connections Likely BruteForce.

4.6 Force Brute FTP — Hydra

Action offensive

```
hydra -l admin -P /usr/share/wordlists/rockyou.txt ftp://192.168.100.188
```

```
(kali@kali)-[~]
$ hydra -l admin -P /usr/share/wordlists/rockyou.txt ftp://192.168.100.188
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-04-24 11:27:31
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://192.168.100.188:21/
```

Figure 20 — Hydra attaquant le service FTP (port 21) avec le compte admin

Détection Wazuh / Suricata

timestamp	agent.name	rule.description	rule.level	rule.id
Apr 24, 2026 @ 16:29:02.770	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:02.747	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:02.740	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:02.734	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:01.174	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:00.945	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:00.784	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:00.784	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:00.780	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601
Apr 24, 2026 @ 16:29:00.758	Ubuntu	Suricata: Alert - ET SCAN Potential FTP Brute-Force attempt response	3	86601

Figure 21 — Dashboard : rafale d'alertes 'ET SCAN Potential FTP Brute-Force attempt response' — 10+ en < 1 seconde

4.7 Force Brute RDP — Hydra

Action offensive (Mehdi)

```
hydra -l administrator -P /usr/share/wordlists/rockyou.txt rdp://192.168.56.107
```

```
(kali@kali)-[~]
$ hydra -l administrator -P /usr/share/wordlists/rockyou.txt rdp://192.168.56.107
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-04-27 10:40:33
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a pr
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100
[DATA] attacking rdp://192.168.56.107:3389/
[ERROR] freerdp: The connection failed to establish.
[3389][rdp] host: 192.168.56.107 login: administrator password: 123456
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-04-27 10:40:47
```

Figure 22 — Hydra RDP : 1 mot de passe valide trouvé pour le compte administrator (123456)

Détection Wazuh / Suricata

timestamp	agent.name	rule.description	rule.level	rule.id
Apr 27, 2026 @ 15:40:46.298	Ubuntu	Suricata: Alert - ET INFO RDP - Response To External Host	3	86601
Apr 27, 2026 @ 15:40:46.298	Ubuntu	Suricata: Alert - ET INFO RDP - Response To External Host	3	86601
Apr 27, 2026 @ 15:40:46.290	Ubuntu	Suricata: Alert - ET POLICY MS Remote Desktop Administrator Login Request	3	86601
Apr 27, 2026 @ 15:40:46.288	Ubuntu	Suricata: Alert - ET POLICY MS Remote Desktop Administrator Login Request	3	86601
Apr 27, 2026 @ 15:40:46.283	Ubuntu	Suricata: Alert - ET POLICY MS Remote Desktop Administrator Login Request	3	86601
Apr 27, 2026 @ 15:40:46.280	Ubuntu	Suricata: Alert - ET INFO RDP - Response To External Host	3	86601

Figure 23 — Dashboard : alertes RDP (ET POLICY MS Remote Desktop Administrator Login Request / ET INFO RDP Response)

Résultat critique : Hydra a trouvé le mot de passe administrator: 123456, illustrant le danger réel des mots de passe faibles sur des services exposés.

4.8 Injection SQL — SQLMap (DVWA)

Action offensive (Majida)

```
sqlmap -u "http://192.168.56.107/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
--cookie="PHPSESSID=agnisokjueobku7irarebfcqbh; security=low" \
--batch --dbs
```

```

(kali@kali)~$ sqlmap -u "http://192.168.56.107/dvwa/vulnerabilities/sqli/?id=126Submit=Submit" \
--cookie="PHPSESSID=agnisokjueobku7irarebfcqbh; security=low" \
--batch --dbs

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable
responsible for any misuse or damage caused by this program

[*] starting @ 16:37:50 /2026-04-26/

[16:37:51] [INFO] testing connection to the target URL
[16:37:51] [INFO] checking if the target is protected by some kind of WAF/IPS
[16:37:51] [INFO] testing if the target URL content is stable
[16:37:51] [INFO] target URL content is stable
[16:37:51] [INFO] testing if GET parameter 'id' is dynamic
[16:37:51] [WARNING] GET parameter 'id' does not appear to be dynamic
[16:37:51] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[16:37:51] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[16:37:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[16:37:51] [WARNING] reflective value(s) found and filtering out
[16:37:52] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[16:37:52] [INFO] testing 'Generic inline queries'
[16:37:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[16:37:52] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[16:37:52] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[16:37:53] [INFO] GET parameter 'id' appears to be 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)' injectable (with --not-string="Me")
[16:37:53] [INFO] GET parameter 'id' is 'MySQL > 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)' injectable
[16:37:53] [INFO] testing 'MySQL inline queries'
[16:37:53] [INFO] testing 'MySQL > 5.0.12 stacked queries (comment)'
[16:37:53] [INFO] testing 'MySQL > 5.0.12 stacked queries'
[16:37:53] [INFO] testing 'MySQL > 5.0.12 stacked queries (query SLEEP - comment)'
[16:37:53] [INFO] testing 'MySQL > 5.0.12 stacked queries (query SLEEP)'
[16:37:53] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[16:37:53] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[16:37:53] [INFO] testing 'MySQL > 5.0.12 AND time-based blind (query SLEEP)'
[16:38:03] [INFO] GET parameter 'id' appears to be 'MySQL > 5.0.12 AND time-based blind (query SLEEP)' injectable
[16:38:03] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[16:38:03] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[16:38:03] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[16:38:03] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extendi
[16:38:03] [INFO] target URL appears to have 2 columns in query
[16:38:03] [INFO] GET parameter 'id' is 'MySQL UNION query (NULL) - 1 to 20 columns' injectable
[16:38:03] [WARNING] in OR boolean-based injection cases, please consider usage of switch '--drop-set-cookie' if you experience any problems during data retriev
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N

```

Figure 24 — SQLMap 1.10.4 : paramètre id vulnérable (Boolean-blind, Error-based EXTRACTVALUE, UNION, Time-based SLEEP)

Détection Wazuh / Suricata

timestamp	agent.name	rule.description	rule.level	rule.id
Apr 26, 2026 @ 21:38:03.774	Ubuntu	Suricata: Alert - ET WEB_SERVER Possible SQL Injection SELECT CAST in HTTP URI	3	86601
Apr 26, 2026 @ 21:38:03.772	Ubuntu	Suricata: Alert - ET WEB_SERVER Possible MySQL SQLi Attempt Information Schema Access	3	86601
Apr 26, 2026 @ 21:38:03.762	Ubuntu	Suricata: Alert - ET WEB_SERVER Possible Attempt to Get SQL Server Version in URI using SELECT VERSION	3	86601
Apr 26, 2026 @ 21:38:03.754	Ubuntu	Suricata: Alert - ET WEB_SERVER Possible Attempt to Get SQL Server Version in URI using SELECT VERSION	3	86601
Apr 26, 2026 @ 21:38:03.742	Ubuntu	Suricata: Alert - ET WEB_SERVER Possible attempt to enumerate MS SQL Server version	3	86601
Apr 26, 2026 @ 21:38:03.740	Ubuntu	Suricata: Alert - ET WEB_SERVER Possible Attempt to Get SQL Server Version in URI using SELECT VERSION	3	86601

Figure 25 — Dashboard : 210+ alertes SQLi (ET WEB_SERVER SQL Injection, Information Schema Access, SELECT VERSION, Sleep Time Delay...)

14 pages d'alertes ont été générées ($14 \times 15 = 210+$), couvrant toutes les techniques SQLi automatiquement testées par SQLMap sur la cible MySQL.

4.9 Attaque DoS par Flood SYN — hping3

Action offensive

```
sudo hping3 -S --flood -V -p 80 192.168.56.107
```

```
(kali@kali)-[~]
$ sudo hping3 -S --flood -V -p 80 192.168.56.107
using eth0, addr: 192.168.56.104, MTU: 1500
HPING 192.168.56.107 (eth0 192.168.56.107): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
— 192.168.56.107 hping statistic —
919549 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figure 26 — hping3 flood SYN : 919 549 paquets transmis, 100% packet loss, interface eth0 depuis 192.168.56.104

Détection Wazuh / Suricata

Apr 26, 2026 @ 21:41:30.024	Ubuntu	Agent event queue is full. Events may be lost.	9	203
Apr 26, 2026 @ 21:41:28.396	Ubuntu	New dpkg (Debian Package) requested to install.	3	2901
Apr 26, 2026 @ 21:41:27.955	Ubuntu	Agent event queue is full. Events may be lost.	9	203
Apr 26, 2026 @ 21:41:23.856	Ubuntu	Agent event queue is full. Events may be lost.	9	203
Apr 26, 2026 @ 21:41:21.832	Ubuntu	Agent event queue is full. Events may be lost.	9	203
Apr 26, 2026 @ 21:41:19.741	Ubuntu	Agent event queue is full. Events may be lost.	9	203
Apr 26, 2026 @ 21:41:15.527	Ubuntu	Agent event queue is full. Events may be lost.	9	203
Apr 26, 2026 @ 21:41:13.535	Ubuntu	Agent event queue is full. Events may be lost.	9	203

Figure 27 — Dashboard : alertes level 9 'Agent event queue is full. Events may be lost' (rule.id 203)

Impact critique : le flood a saturé la file d'événements de l'agent Wazuh lui-même. Ce comportement réel illustre comment une attaque DoS volumétrique peut dégrader les capacités de surveillance.

4.10 Élévation de Privilèges — sudo

Action post-exploitation

```
sudo echo "HACKED" >>> /etc/hosts # Permission denied
sudo -i # Élévation vers root
sudo echo "HACKED" >>> /etc/hosts # Succès
```

```

Session  Actions  Edit  View  Help
vboxuser@Ubuntu:~$ sudo echo "HACKED" >> /etc/hosts
-bash: /etc/hosts: Permission denied
vboxuser@Ubuntu:~$ sudo -i
[sudo] password for vboxuser:
root@Ubuntu:~# sudo echo "HACKED" >> /etc/hosts
root@Ubuntu:~#

```

Figure 28 — Terminal cible : escalade sudo vers root, puis modification du fichier /etc/hosts

Détection Wazuh — FIM et Audit

Apr 26, 2026 @ 21:54:02.488	Ubuntu	Integrity checksum changed.	7	550
Apr 26, 2026 @ 21:54:00.194	Ubuntu	Successful sudo to ROOT executed.	3	5402

Figure 29 — Dashboard : 'Successful sudo to ROOT' (rule 5402) et 'Integrity checksum changed' (rule 550, level 7)

Détection double : Wazuh capte l'élévation de privilèges via l'audit sudo (5402) ET la modification du fichier système via le module FIM (550). Ces détections sont indépendantes de Suricata.

4.11 Transmission de Mots de Passe en Clair — HTTP POST

timestamp	agent.name	rule.description	rule.level	rule.id
Apr 24, 2026 @ 16:35:41.849	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.843	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.835	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.826	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.818	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.733	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.717	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.705	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.693	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.685	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.684	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.585	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601
Apr 24, 2026 @ 16:35:41.580	Ubuntu	Suricata: Alert - ET POLICY HTTP POST contains pass= in cleartext	3	86601

Figure 30 — Dashboard : alertes 'ET POLICY HTTP POST contains pass= in cleartext' (rule 86601) — mots de passe FTP envoyés en clair

Suricata détecte les mots de passe transmis en clair dans les requêtes HTTP POST, alertant sur l'absence de chiffrement (absence de HTTPS) dans les échanges applicatifs.

5 Intégration pfSense — Firewall et Remote Syslog

Dans la phase étendue du laboratoire, un pare-feu pfSense 2.8.1-RELEASE (FreeBSD 15.0-CURRENT) a été déployé sous VirtualBox pour segmenter le réseau et centraliser les logs firewall dans Wazuh via le protocole syslog. pfSense agit comme passerelle entre la zone attaquante (OPT1 — 192.168.57.0/24) et la zone cible/SIEM (LAN — 192.168.56.0/24).

Étape 1 — Démarrage et vérification de pfSense

Au démarrage de la VM pfSense, la console affiche la configuration réseau active :

```
VirtualBox Virtual Machine - Netgate Device ID: 39e27eb4b80e8994b72f

*** Welcome to pfSense 2.8.1-RELEASE (amd64) on pfsense-lab ***

WAN (wan)   -> em0 -> v4/DHCP4: 10.0.2.15/24
              v6/DHCP6: fd17:625c:f037:2:a00:27ff:fe1f:5dff/64
LAN (lan)   -> em1 -> v4: 192.168.56.2/24
OPT1 (opt1) -> em2 -> v4: 192.168.57.1/24
```

Figure 31 — Console pfSense 2.8.1 : WAN (em0) → 10.0.2.15/24, LAN (em1) → 192.168.56.2/24, OPT1 (em2) → 192.168.57.1/24

Les trois interfaces sont actives : **WAN** (em0, 10.0.2.15 — accès NAT Internet), **LAN** (em1, 192.168.56.2 — réseau SIEM/cible) et **OPT1** (em2, 192.168.57.1 — réseau attaquant Kali Linux).

Étape 2 — Vérification du tableau de bord pfSense

Depuis l'interface web de pfSense (accédée via 192.168.56.107 — machine de Reda), le tableau de bord système confirme l'état opérationnel du firewall :

The screenshot displays the pfSense web interface. The left sidebar shows 'System Information' with details like Name (pfsense-lab.local), User (admin@192.168.56.107), System (VirtualBox Virtual Machine), BIOS (Innotek GmbH), Version (2.8.1-RELEASE (amd64)), CPU Type (Intel(R) Core(TM) i5-8250U), Hardware crypto (Inactive), Kernel PTI (Enabled), MDS Mitigation (Inactive), Uptime (00 Hour 56 Minutes 13 Seconds), Current date/time (Thu May 14 17:58:18 +01 2026), DNS server(s) (127.0.0.1, ::1, 10.0.2.3, 8.8.8.8, 8.8.4.4), Last config change (Thu May 14 17:47:35 +01 2026), State table size (0%), MBUF Usage (0%), Load average (0.59, 0.98, 1.72), CPU usage (11%), Memory usage (33% of 968 MiB), and SWAP usage (0% of 1024 MiB). The right sidebar shows 'Netgate Services And Support' with contract type 'Community Support' and a list of support resources. Below this, the 'Interfaces' section lists three active interfaces: WAN (10.0.2.15), LAN (192.168.56.2), and OPT1 (192.168.57.1).

Figure 32 — Tableau de bord pfSense : version 2.8.1-RELEASE, FreeBSD 15.0, CPU i5-8250U, uptime 56 minutes, interfaces WAN/LAN/OPT1 actives

Informations clés relevées :

- **Version** : pfSense 2.8.1-RELEASE (amd64), built on Mon Dec 15 2025
- **Système** : VirtualBox VM, Netgate Device ID 39e27eb4b80e8994b72f
- **Ressources** : CPU 11%, RAM 33% de 968 MiB — charge légère
- **DNS** : 127.0.0.1, ::1, 10.0.2.3, 8.8.8.8, 8.8.4.4
- **Date/heure** : Thu May 14 17:58:18 +01 2026 — synchronisée avec le laboratoire

Étape 3 — Configuration du Remote Syslog vers Wazuh

La fonctionnalité Remote Logging de pfSense a été configurée pour envoyer tous les événements système et firewall au Wazuh Manager via UDP/514 :

The screenshot shows the 'Remote Logging Options' configuration page in pfSense. The 'Enable Remote Logging' checkbox is checked. The 'Source Address' dropdown is set to 'LAN'. The 'IP Protocol' dropdown is set to 'IPv4'. The 'Remote log servers' field contains '192.168.56.108'. The 'Remote Syslog Contents' section has 'Everything' selected, with a list of other options like 'System Events', 'Firewall Events', etc., all unchecked. A 'Save' button is at the bottom.

Figure 33 — pfSense Remote Logging Options : envoi syslog UDP/514 vers 192.168.56.108 (Wazuh Manager), source LAN, protocole IPv4, contenu « Everything » coché

Paramètres de configuration appliqués :

- **Enable Remote Logging** : activé (case cochée)
- **Source Address** : LAN — le trafic syslog part de l'interface 192.168.56.2
- **IP Protocol** : IPv4
- **Remote log server** : 192.168.56.108 (Wazuh Manager) sur le port 514 UDP (par défaut syslog)
- **Remote Syslog Contents** : Everything — tous les événements sont transmis (système, firewall, DNS, DHCP, auth, VPN, routage, NTP, etc.)

Cette configuration permet au Wazuh Manager de recevoir et corréler les logs pfSense avec les alertes Suricata et les événements agent, offrant une vue unifiée de l'ensemble de la chaîne réseau.

Note : Sur le Wazuh Manager, le démon syslog (port 514 UDP) doit être activé dans `ossec.conf` via le bloc `<remote><connection>syslog</connection><port>514</port></remote>` pour que pfSense puisse y envoyer ses logs.

Étape 4 — Règles de Firewall OPT1 (Isolation des Attaquants)

Des règles firewall ont été configurées sur l'interface OPT1 pour contrôler le trafic issu des machines attaquantes Kali Linux (192.168.57.0/24) :

Floating
WAN
LAN
OPT1

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>		0/132 KiB	IPv4 *	192.168.57.107	*	*	*	*	none	Block Kali Linux - 192.168.57.107	
<input type="checkbox"/>		0/12 KiB	IPv4 *	OPT1 subnets	*	*	*	*	none		

Add
 Add
 Delete
 Toggle
 Copy
 Save
 Separator

Figure 34 — Règles pfSense OPT1 : blocage explicite de 192.168.57.107 (Kali Linux spécifique) et autorisation générale du sous-réseau OPT1

Architecture des règles (ordre d'évaluation descendant) :

- **Règle 1 (bloquante, X) :** Block Kali Linux — 192.168.57.107, IPv4 *, source 192.168.57.107, toutes destinations. Cette règle bloque explicitement une machine Kali spécifique (0/132 KiB de trafic traité lors de la capture).
- **Règle 2 (permissive, ✓) :** Pass, IPv4 *, source OPT1 subnets, toutes destinations. Autorise le reste du sous-réseau OPT1 à communiquer (0/12 KiB).

Cette approche de segmentation réseau permet de simuler des scénarios réalistes où certains attaquants sont isolés (par exemple suite à une détection) tandis que d'autres conservent un accès contrôlé à la cible.

Apport sécurité : L'intégration pfSense ajoute une couche de défense périmétrique au laboratoire. Les logs firewall transmis à Wazuh permettent de corréler les décisions de filtrage réseau avec les alertes IDS/SIEM, enrichissant l'analyse de la chaîne d'attaque complète (reconnaissance → exploitation → post-exploitation).

6. Conclusion

Ce laboratoire démontre de manière concrète la **complémentarité indispensable** entre un SIEM (Wazuh) et un NIDS (Suricata) dans une stratégie de défense en profondeur.

Apports de Suricata

En tant que sonde réseau passive, Suricata a permis la détection précoce d'activités offensives dès la phase de reconnaissance (pings ICMP, scans Nmap, Nikto), bien avant toute intrusion réussie. La richesse des règles Emerging Threats (40+ catégories) lui confère une couverture transversale : SQL Injection, Brute Force, DoS, RDP, curl suspect, mots de passe en clair.

Apports de Wazuh

En centralisant les alertes Suricata et en les corrélant avec les journaux système natifs (PAM, sshd, FIM), Wazuh offre une vue unifiée et horodatée de toute la chaîne d'attaque — du scan initial (level 3) jusqu'à l'élévation de privilèges (level 10). Des détections comme la règle 5551 (multiple failed logins) ou 550 (integrity checksum changed) seraient invisibles sans l'agent.

Limites observées

L'attaque DoS par flood hping3 a saturé la file d'événements de l'agent Wazuh (rule 203 — Agent event queue is full), démontrant qu'une attaque volumétrique intense peut dégrader la surveillance elle-même. En production, un rate-limiting sur l'agent, l'augmentation des buffers ou une architecture Wazuh multi-tier seraient nécessaires.

Apports de pfSense

L'intégration de pfSense comme pare-feu périmétrique a enrichi le laboratoire d'une troisième couche défensive. En segmentant le réseau (LAN, OPT1, WAN) et en centralisant ses logs via syslog UDP/514 vers le Wazuh Manager, pfSense permet de corréler les décisions de filtrage réseau avec les alertes Suricata et les événements agents. Cette visibilité sur la chaîne complète — de la décision firewall jusqu'à la détection SIEM — rapproche l'architecture d'un vrai SOC en profondeur.

Bilan général

Ce projet valide que le triptyque **Wazuh + Suricata + pfSense** constitue une solution SOC open-source robuste et viable, combinant détection d'intrusion réseau, corrélation SIEM et contrôle périmétrique.