

RÉPUBLIQUE TUNISIENNE



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

DIRECTION GÉNÉRALE DES ÉTUDES TECHNOLOGIQUES

Institut Supérieur des Études Technologiques de Siliana
Département technologie de l'informatique



Projet de fin d'études

RSI 2024/2025

Mise en place d'une solution de gestion des logs avec le SIEM QRadar

Élaboré par :

Maram Sendesni
Sadok Lamiri

Encadré dans l'ISET par :

Pr. Ameur Salem ZAIDOUN

Encadré dans :



Encadré dans la société par :

Mlle. Ameni BEN KHADOUR

Année universitaire : 2024/2025

Dédicace

Je dédie ce travail :

À jamais dans mon cœur, car sans vous, rien n'aurait été possible.

Maram SENDESNI

Dédicace

Je dédie ce travail à :

Avec une immense reconnaissance pour votre présence et votre soutien.

Sadok LAMIRI

Remerciements

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui nous ont soutenus et guidés tout au long de la réalisation de ce projet :

Nous souhaitons exprimer, à travers ces quelques lignes, notre profonde gratitude envers toutes les personnes qui, par leur soutien, leurs conseils et leur bienveillance, nous ont permis de mener à bien ce projet de fin d'études.

Nous adressons nos sincères remerciements au Professeur Ameur Salem ZAIDOUN, notre encadrant universitaire à l'ISET Siliana, pour son accompagnement, sa disponibilité et ses conseils avisés tout au long de ce projet. Son expertise et son soutien nous ont été d'une grande aide dans l'élaboration et la réussite de notre travail.

Nos remerciements vont également à Mme Ameny Ben Khadhour, notre encadrante au sein de Network Associates, pour son encadrement, ses précieux conseils techniques et son accompagnement tout au long de notre mission en entreprise. Son expertise et son engagement nous ont permis d'enrichir nos compétences et d'évoluer dans un environnement professionnel enrichissant.

Nous exprimons également notre reconnaissance aux membres du jury qui nous font l'honneur d'évaluer notre travail. Nous les remercions pour le temps qu'ils nous accordent ainsi que pour leurs remarques et recommandations constructives.

Enfin, nous tenons à remercier l'ensemble des enseignants de l'ISET Siliana, ainsi que toutes les personnes qui nous ont soutenus de près ou de loin durant notre parcours académique. Leur engagement et leur bienveillance ont été essentiels dans notre apprentissage et notre développement professionnel.

Table des matières

Introduction Générale	1
1 Contexte général du projet	2
Introduction	2
1.1 Organisme d'accueil : Network Associates	2
1.1.1 Présentation générale	2
1.1.2 Organigramme	3
1.1.3 Services Offerts	3
1.2 Projet : Solution SIEM, QRadar	4
1.2.1 État des lieux et problématique	4
1.2.2 Solution envisagée	4
1.2.3 Objectifs	5
1.2.4 Cahier des Charges	5
1.3 Méthodologie de travail	5
1.3.1 Méthodologies prévalentes	5
1.3.2 Méthodologie choisie : RAD	6
conclusion	7
2 État de l'art	8
Introduction	8
2.1 Cybersécurité	8
2.2 Outils SIEM	9
2.2.1 Technologie SIEM	9
2.2.2 Solutions Existantes	9
2.2.3 SIEM retenu	10
2.3 IBM QRadar	10
2.3.1 Présentation de la Solution	10
2.3.2 Architecture Générale	11
2.3.3 Fonctionnalités	11
2.3.4 Infractions	12
2.4 Gestion des Logs	14
2.4.1 Types	14
2.4.2 Protocoles de Collecte	14
2.4.3 Automatisation	14
2.5 Plateforme EVE-NG	15
2.5.1 Présentation	15
2.5.2 Rôle	15
2.5.3 Avantages	16
conclusion	16

3 Stratégie de mise en place de la solution	17
Introduction	17
3.1 Étude préalable	17
3.1.1 Problématique	17
3.1.2 Objectifs	17
3.1.3 Choix IBM QRadar	17
3.2 Solution adoptée	18
3.2.1 Architecture réseaux	18
3.2.2 Composants principaux	19
3.2.3 Réponse automatisée aux incidents	21
3.3 Étude fonctionnelle	21
3.3.1 Fonctionnalités réseau	21
3.3.2 Fonctionnalités SIEM	22
3.3.3 Fonctionnalités d'automatisation	22
Conclusion	22
4 Déploiement des outils	23
Introduction	23
4.1 Mise en place d'émulateur : EVE-NG	23
4.1.1 Téléchargement et installation	23
4.1.2 Préparation d'environnement	26
4.1.3 Création de topologie	37
4.2 Configuration du réseau sécurisé	38
4.2.1 paramétrage réseau	38
4.2.2 Configuration du VPN IPsec	42
4.2.3 Tests réseau	44
4.3 Mise en œuvre du SIEM : QRadar	49
4.3.1 Implémentation et paramétrage initial	49
4.3.2 Intégration au réseau	53
4.3.3 Configuration des sources de logs	60
4.4 Collecte des logs : WinCollect	62
4.4.1 Intégration d'agent	62
4.4.2 Validation de la centralisation des logs	65
4.4.3 Détection d'infractions	66
4.5 Automatisation : Python	70
4.5.1 Préparation de l'environnement	70
4.5.2 Script de surveillance	72
4.5.3 Validation d'automatisation	76
Conclusion	78
5 Tests et validation	79
Introduction	79
5.1 Environnement de travail	79
5.1.1 Infrastructure matérielle	79
5.1.2 Infrastructure logicielle	79
5.1.3 Contraintes techniques	81
5.2 Scénarios de test	81
5.2.1 Scénario 1 : Collecte des logs	81
5.2.2 Scénario 2 : Détection d'activités malveillantes	81
5.2.3 Scénario 3 : Réaction automatisée	82
5.2.4 Scénario 4 : Simulation d'une attaque complète	82

5.3 Analyse des résultats	82
5.3.1 Limites du projet	82
5.3.2 Axes d'amélioration	83
Conclusion	83
Conclusion Générale	84
Webographie	86

Table des figures

1.1	Logo d'Organisme d'accueil	2
1.2	Organigramme d'Organisme d'accueil	3
1.3	Processus RAD	7
2.1	Les couches de la cybersécurité	9
2.2	Architecture fonctionnelle	11
2.3	Cycle de vie d'une infraction	12
2.4	Logo EVE-NG	15
3.1	Architecture réseau	19
3.2	VPN site à site	20
4.1	Résumé de la configuration de la machine virtuelle EVE-NG sous Hyper-V	24
4.2	Montage du CD-ROM et préparation de l'installation	24
4.3	Écran de démarrage	25
4.4	Avertissement de l'absence de virtualisation matérielle	25
4.5	Activation de la virtualisation via PowerShell	25
4.6	Connexion réussie à l'interface web	26
4.7	Résumé des paramètres d'installation	26
4.8	page login d'outil WinSCP	27
4.9	Images OS Fortinet	27
4.10	images OS Windows	28
4.11	images OS switechs cisco	28
4.12	Configuration de la session SFTP	29
4.13	Interface WinSCP	29
4.14	dossier "fortinet-FGTV5" pour l'image Fortinet	30
4.15	Transfert de "fortios.qcow2" dans le dossier approprié	30
4.16	Accès au répertoire QEMU	31
4.17	Création du dossier <code>win-10</code>	31
4.18	Transfert de l'image Windows 10	32
4.19	Vérification du dossier <code>win-10</code>	32
4.20	Transfert des images IOL Cisco dans le dossier <code>/opt/unetlab addons/iol/bin/</code>	33
4.21	Répertoires d'extensions dans EVE-NG	33
4.22	Contenu du répertoire <code>/opt/unetlab addons/iol/</code>	34
4.23	Téléchargement du Windows Integration Pack	35
4.24	Assistant d'installation du Windows Integration Pack	35
4.25	Installation de UltraVNC	36
4.26	Installation de Wireshark	36
4.27	Fin de l'installation du Windows Integration Pack	37
4.28	Interface création lab	37
4.29	L'ajout du Fortinet FortiGate	38
4.30	FW FortiGate	38
4.31	Préparation de la topologie réseau sur EVE-NG	39

4.32 Ajout de deux interfaces réseau (LAN/WAN)	39
4.33 Lancement des équipements et ouverture du terminal Putty	40
4.34 Connexion à l'interface CLI de FortiGate avec l'utilisateur admin	40
4.35 Configuration des interfaces et de la route statique sur FortiGate	41
4.36 Sauvegarde de la configuration via la CLI FortiGate	41
4.37 Dashboard de l'interface graphique web du pare-feu FortiGate	41
4.38 Configuration d'un VPN IPsec site-à-site	42
4.39 Phase d'authentification du VPN IPsec	42
4.40 Définition des réseaux internes dans le VPN	42
4.41 Aperçu du tunnel IPsec en cours	43
4.42 État du tunnel VPN : Down	43
4.43 Ajout de routes statiques	43
4.44 Configuration du routage de FortiGate	44
4.45 Configuration des politiques de sécurité sur le FortiGate 1	44
4.46 Configuration des politiques de sécurité sur le FortiGate 2	44
4.47 Test de connectivité entre les FortiGate	45
4.48 RéPLICATION de la configuration sur le second FortiGate	45
4.49 Tunnel VPN actif	45
4.50 Lancement des machines Windows via UltraVNC	46
4.51 Configuration réseau des PC	46
4.52 Ajout d'un PC supplémentaire à la topologie	47
4.53 Vérification du pare-feu Windows	47
4.54 Attribution des adresses IP des postes	48
4.55 Test de tentative de connexion échouée	48
4.56 Page de téléchargement de l'ISO	49
4.57 État initial de la mémoire	50
4.58 Téléchargement de RAMMap depuis Sysinternals	50
4.59 Exécution de RAMMap	50
4.60 Fonctions de nettoyage de la mémoire dans RAMMap	51
4.61 État de la mémoire après l'utilisation de RAMMap	51
4.62 Résumé des paramètres initiaux de la VM	52
4.63 Résumé de la création du disque dur virtuel	53
4.64 Écran de démarrage de la VM QRadar	53
4.65 Démarrage de l'installation du système	54
4.66 Choix du mode d'installation	54
4.67 Définition du rôle	55
4.68 Sélection du type d'installation	55
4.69 Choix du rôle	55
4.70 Type d'installation sélectionné	56
4.71 Configuration date et de l'heure	56
4.72 Sélection du protocole réseau et mode de configuration	57
4.73 Configuration des paramètres réseau de QRadar	57
4.74 Téléchargement de Nmap depuis Nmap.org	57
4.75 Choix des composants à installer avec Nmap	58
4.76 Choix du dossier d'installation de Nmap	58
4.77 Résultat du ping scan	59
4.78 Utilisation de Zenmap avec un scan personnalisé	59
4.79 Paramètres réseau	60
4.80 Interface système de FortiGate	60
4.81 Paramétrage Syslog dans "Log Settings"	61
4.82 Liste des événements reçus	61

4.83	Test de ping entre le FW 0% de perte	62
4.84	Extrait des événements	62
4.85	Sélection du package	63
4.86	L'exécution de l'installation	63
4.87	Sélection des composants à installer	63
4.88	Paramètre de destination des logs	64
4.89	Confirmation de l'installation	64
4.90	Écran final de l'installation	65
4.91	Étapes complètes de vérification	65
4.92	Affichage des logs reçus dans QRadar	66
4.93	Extrait des journaux relatifs à l'activité utilisateur	66
4.94	Affichage de la règle personnalisée QRadar	67
4.95	Configuration de l'audit des événements	67
4.96	Notification générée suite à une infraction détectée	68
4.97	Consultation des événements associés à l'alerte	68
4.98	Confirmation de l'alerte répétée suite à des connexions suspectes	69
4.99	Multiples infractions détectées par la même règle	69
4.100	Détails de l'infraction détectée	70
4.101	Connexion SSH vers la machine cible	70
4.102	Alerte déclenchée par une connexion SSH suspecte	70
4.103	Installation réussie	71
4.104	Installation des bibliothèques via pip	71
4.105	Vérification de la bibliothèque <code>requests</code>	72
4.106	Création du dossier <code>PFE_QRadar</code>	72
4.107	Aperçu de la configuration initiale des bibliothèques et des variables email	72
4.108	Fonction d'envoi d'alerte par email	73
4.109	Fonction de vérification de l'état du pare-feu	73
4.110	Fonction d'activation automatique du pare-feu si désactivé	74
4.111	Fonction principale de gestion du pare-feu	75
4.112	Point d'entrée principal du script de surveillance	75
4.113	Lancement du script depuis PowerShell en mode administrateur	76
4.114	désactivation du pare-feu Windows	76
4.115	Sortie console lors de la détection d'un pare-feu désactivé	77
4.116	Email de confirmation reçu après réactivation automatique	77
5.1	Affichage du contenu du répertoire Downloads via PowerShell	87
5.2	Vérification de la présence du fichier d'installation	87
5.3	Contenu du dossier d'installation	87
5.4	Vérification de l'intégrité du fichier d'installation à l'aide de SHA256	88

Liste des tableaux

1.1	Comparaison entre méthode classique et méthode agile	5
1.2	Comparaison entre approches agiles	6
2.1	Comparaison des solutions SIEM	10
2.2	Composants d'une infraction QRadar	13
5.1	Outils déployés dans l'environnement	80
5.2	Outils utilisés pour la gestion du projet	81

Liste des abréviations

SIEM	Security Information and Event Management
SIM	Security Information Management
SEM	Security Event Management
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
VPN	Virtual Private Network
IPsec	Internet Protocol Security
API	Application Programming Interface
SNMP	Simple Network Management Protocol
ELK	Elasticsearch, Logstash, Kibana
OSSIM	Open Source Security Information Management
RGPD	Règlement Général sur la Protection des Données
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
PCI-DSS	Payment Card Industry Data Security Standard
GDPR	General Data Protection Regulation
EVE-NG	Emulated Virtual Environment – Next Generation
ML	Machine Learning
IA	Intelligence Artificielle
X-Force	Base de renseignements sur les menaces d'IBM
Power BI	Outil de reporting et de visualisation de Microsoft
WinCollect	Agent Windows de collecte de logs pour QRadar
DSDM	Dynamic Systems Development Method
RAD	Rapid Application Development
SOC	Security operation center
UEFI	Unified Extensible Firmware Interface
BIOS	Basic Input/Output System
VM	Virtual Machine
FW	Firewall

Introduction Générale

Le Projet de Fin d'Études (PFE) représente une étape décisive dans le parcours académique des étudiants en licence professionnelle. Il constitue un pont entre les acquis théoriques de la formation et les exigences du monde professionnel. Cette expérience immersive permet de développer des compétences pratiques, de travailler sur des problématiques réelles et de s'intégrer dans la dynamique de l'entreprise d'accueil.

Dans un contexte marqué par l'évolution rapide des technologies de l'information et l'augmentation constante des menaces numériques, la sécurité des systèmes informatiques s'impose aujourd'hui comme une priorité stratégique pour toutes les organisations. La capacité à détecter, analyser et répondre aux incidents de sécurité est devenue indispensable pour assurer la protection du patrimoine informationnel.

C'est dans cette optique que s'inscrit le présent projet de fin d'études, réalisé au sein de l'entreprise **Network Associates**. Il vise à contribuer à l'amélioration de la surveillance et de la gestion des événements de sécurité à travers une solution adaptée aux besoins de l'entreprise. Ce projet a été l'occasion d'approfondir nos compétences dans le domaine de la cybersécurité, tout en développant une méthodologie rigoureuse de gestion de projet.

Le présent rapport rend compte du travail réalisé tout au long de cette expérience professionnelle. Il est structuré en cinq chapitres, chacun apportant une contribution complémentaire à la compréhension du projet :

— **Chapitre 1 : Contexte général du projet**

Ce chapitre présente l'environnement de stage, la problématique identifiée, les objectifs du projet ainsi que la méthodologie adoptée.

— **Chapitre 2 : État de l'art**

Cette partie expose les notions théoriques et les concepts clés liés à la cybersécurité et aux mécanismes de gestion des événements de sécurité. Elle fournit également un aperçu des solutions existantes dans ce domaine.

— **Chapitre 3 : Stratégie de mise en place de la solution**

Ce chapitre détaille la démarche choisie pour concevoir une réponse adaptée à la problématique, en précisant les grandes étapes de planification, de choix des outils et de conception générale.

— **Chapitre 4 : Déploiement des outils**

Il présente le processus de mise en œuvre de la solution, en mettant en évidence les différentes phases techniques de configuration et d'intégration, tout en respectant les exigences initiales.

— **Chapitre 5 : Tests et validation**

Cette dernière section évalue l'efficacité de la solution mise en place à travers une série de tests, met en lumière les résultats obtenus, et propose une analyse critique des limites rencontrées ainsi que des perspectives d'amélioration.

Ainsi, ce projet nous a permis de mobiliser nos acquis académiques dans un cadre professionnel stimulant, de mieux appréhender les enjeux actuels de la sécurité informatique, et de renforcer notre autonomie dans la gestion de projets techniques. Il constitue une expérience enrichissante, tant sur le plan personnel que professionnel, et s'inscrit pleinement dans notre projet de carrière.

Chapitre 1

Contexte général du projet

Introduction

Ce chapitre a pour objectif de présenter le cadre global dans lequel s'inscrit le projet de mise en place d'une solution SIEM basée sur IBM QRadar. Il débute par une description de l'organisme d'accueil, Network Associates, en exposant ses missions, son organisation interne et les services proposés. Par la suite, il détaille le projet confié dans le cadre du stage, en partant de l'état des lieux existant, en identifiant les problématiques rencontrées en matière de gestion des logs et de sécurité, et en expliquant la solution envisagée pour y répondre. Les objectifs du projet ainsi que le cahier des charges sont ensuite explicités afin de cadrer les attendus. Enfin, la méthodologie de travail adoptée, notamment le choix du modèle RAD (Rapid Application Development), est justifiée pour répondre efficacement aux besoins spécifiques du projet dans un délai contraint.

1.1 Organisme d'accueil : Network Associates

1.1.1 Présentation générale

Network Associates (NA) est une entreprise fondée en 2015, spécialisée dans l'intégration et la fourniture de solutions TIC, de logiciels et de services cloud. Elle propose aux entreprises des solutions avancées en matière de communication, d'infrastructure et de sécurité informatique. Grâce à son expertise et à son savoir-faire, NA accompagne ses clients dans leur transformation numérique en leur fournissant des produits performants et adaptés à leurs besoins spécifiques, la figure 1.1 présente le logo de NA.



FIGURE 1.1 – Logo d'Organisme d'accueil

NA se distingue comme l'un des principaux intégrateurs et revendeurs de technologies de l'information en Tunisie. L'entreprise met un point d'honneur à simplifier la gestion des infrastructures informatiques en fournissant des solutions évolutives et sécurisées. Elle propose un accompagnement

sur mesure aux entreprises de toutes tailles et met en place des infrastructures robustes adaptées aux exigences technologiques du marché.

1.1.2 Organigramme

La figure 1.2 ci-dessous présente de manière visuelle l'organigramme de l'organisme d'accueil, permettant de mieux comprendre sa structure hiérarchique ainsi que la répartition des responsabilités entre les différents services.

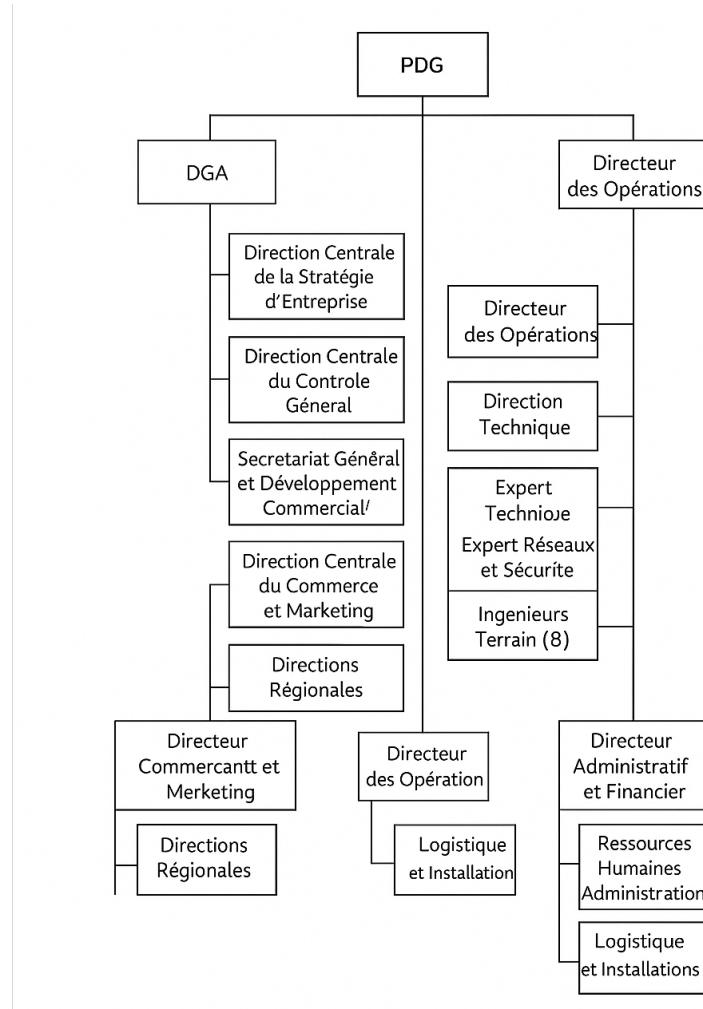


FIGURE 1.2 – Organigramme d'Organisme d'accueil

1.1.3 Services Offerts

NA propose une gamme variée de services conçus pour répondre aux besoins complexes des entreprises modernes. Elle aide ses clients à naviguer dans les défis technologiques actuels en fournissant des solutions complètes dans les domaines suivants :

Consultation

En fournissant des services de consultation complets, NA apporte une valeur ajoutée importante aux entreprises en proposant des stratégies, des architectures, des mises en œuvre et des services d'intégration pour aider à planifier, construire, améliorer et innover les solutions et les plans d'affaires dans divers domaines tels que :

- Gestion de l'exploitation des infrastructures.

- Gestion des applications et des systèmes informatiques.
- Procédure de récupération suite à une catastrophe.

Conception

Avec son équipe d'experts spécialisés dans les performances système, la sécurité et la convivialité du réseau, NA entreprend une analyse approfondie de votre infrastructure. Cette évaluation se concentre sur plusieurs traits essentiels, tels que la convivialité, la flexibilité et la fiabilité.

Installation et optimisation

Dans un contexte technologique en constante évolution, NA accompagne les entreprises dans le choix, l'installation, l'intégration et l'optimisation des technologies adaptées à leurs besoins. Grâce à l'expertise en réseautage, routage, commutation, infrastructure sans fil et sécurité, cette entreprise assure une performance et une fiabilité accrues pour offrir une expérience client optimale. De plus, l'automatisation de l'analyse réseau par NA procure une visibilité améliorée, permettant aux entreprises de tirer le meilleur parti de leurs investissements technologiques et de rester compétitives.

1.2 Projet : Solution SIEM, QRadar

1.2.1 État des lieux et problématique

L'entreprise dispose déjà de plusieurs dispositifs de sécurité tels que des pare-feu, des systèmes de détection d'intrusion (IDS) et des scanners de vulnérabilités ... , qui génèrent quotidiennement une quantité importante de journaux d'activité. Cependant, ces logs sont répartis sur différents équipements, rendant leur consultation, leur analyse et leur corrélation particulièrement complexes. Cette dispersion des données nuit à la réactivité des équipes de sécurité et ralentit considérablement la détection des incidents.

Dans un environnement où les attaques sont de plus en plus nombreuses et sophistiquées, la gestion manuelle des journaux devient rapidement inefficace. Les administrateurs doivent faire face à un volume massif d'événements provenant de sources hétérogènes, ce qui complique l'identification rapide des menaces et l'analyse approfondie des incidents. Il devient alors indispensable de mettre en place une solution de gestion centralisée, capable de collecter, corrélérer et visualiser les événements de manière intelligente et automatisée, afin d'optimiser la surveillance de l'infrastructure et de renforcer la posture de sécurité de l'entreprise.

1.2.2 Solution envisagée

La mise en place d'une solution SIEM (Security Information and Event Management) permet de centraliser, analyser et corrélérer les logs en temps réel. Cette solution fournit des tableaux de bord interactifs, des alertes automatiques en cas de comportement suspect et une réponse rapide aux incidents. Elle facilite également la conformité réglementaire (GDPR, ISO, etc.) en stockant les journaux et générant des rapports d'audit. Afin de renforcer l'efficacité de la détection et de la réponse aux incidents, Des scripts de réponse automatisée peuvent être développés pour traiter immédiatement certains incidents, comme la mise en quarantaine d'un équipement compromis ou le blocage d'une adresse IP malveillante. Cette automatisation permet de libérer du temps pour les analystes et d'améliorer la réactivité face aux menaces.

1.2.3 Objectifs

L'objectif principal de ce projet est la mise en place d'une solution de gestion des logs avec QRadar, permettant de centraliser, corréler et analyser les menaces et les incidents de sécurité. Les objectifs spécifiques sont les suivants :

- Élaboration d'une architecture adaptée pour déployer QRadar.
- Identification des sources de logs à intégrer.
- Mise en place de la collecte, de la corrélation et de l'analyse des logs.
- Analyse des incidents et des vulnérabilités.

1.2.4 Cahier des Charges

Pour répondre aux besoins identifiés, le projet suivra les étapes suivantes :

- Analyse des besoins.
- Conception : Définition de l'architecture QRadar et des spécifications techniques.
- Mise en œuvre : Installation, configuration et intégration de la solution QRadar.
- Intégration des équipements réseaux avec QRadar.
- Test et validation : Test de la collecte des logs et analyse des incidents.

1.3 Méthodologie de travail

1.3.1 Méthodologies prévalentes

La gestion de projet informatique repose sur deux grandes familles de méthodes : les **méthodes classiques** et les **méthodes agiles**. Le choix de la méthodologie à adopter ne doit pas être fait au hasard, mais guidé par une analyse fine du contexte : nature du projet, contraintes temporelles, ressources humaines et techniques, complexité des livrables, évolutivité des exigences et degré d'implication des utilisateurs finaux. Ces critères déterminent la capacité d'une méthode à garantir l'efficacité, la maîtrise des risques et la qualité du produit livré. Le tableau 1.1 présente une comparaison synthétique entre les deux grandes approches, mettant en évidence leurs points forts et leurs limites respectives.

Critère	Méthode classique	Méthode agile
Approche	Linéaire et séquentielle	Itérative et incrémentale
Flexibilité	Faible, difficile à ajuster	Forte, changement intégré en continu
Documentation	Complète et rigide	Allégée, ciblée sur l'essentiel
Livraison	Produit livré à la fin	Versions fonctionnelles continues
Implication du client	Faible, début/fin du projet	Forte, permanente et active
Gestion des risques	Planifiés en amont	Évalués à chaque itération

TABLE 1.1 – Comparaison entre méthode classique et méthode agile

Les méthodes classiques, comme le cycle en V ou le modèle en cascade, sont fondées sur une planification rigoureuse dès le départ, suivie d'une exécution séquentielle sans retour arrière. Ces approches conviennent aux projets stables et bien spécifiés, où les exigences sont peu susceptibles d'évoluer. En revanche, elles sont peu adaptées aux environnements dynamiques ou incertains, où les besoins évoluent fréquemment. De plus, la faible implication du client au cours du processus peut engendrer des écarts entre les attentes et le produit final.

À l'inverse, les méthodes agiles reposent sur une philosophie de développement adaptatif. Elles privilégient des cycles courts, appelés itérations, au cours desquels des fonctionnalités opérationnelles sont développées, testées et validées. L'accent est mis sur l'implication constante du client, la réactivité au changement et la production de valeur dès les premières étapes. Elles permettent un pilotage progressif et une amélioration continue du produit.

Dans le cadre de notre projet — qui consiste à concevoir et déployer une solution SIEM à l'aide de QRadar dans un environnement simulé sous EVE-NG — plusieurs facteurs nous ont conduits à opter pour une approche **semi-agile**. En effet, les contraintes de temps (quelques semaines), la taille réduite de l'équipe (deux personnes), ainsi que la nature progressive des livrables (collecte, centralisation, détection, automatisation) nécessitaient une méthode à la fois structurée et flexible.

Nous avons alors comparé trois méthodes agiles : Scrum, DSDM et RAD. Cette analyse avait pour but d'identifier l'approche la mieux adaptée à notre organisation, notre rythme de travail, ainsi qu'aux objectifs spécifiques de notre projet. Le tableau suivant résume les principales différences entre ces trois méthodes.

Critère	Scrum	DSDM	RAD
Organisation	Rôles bien définis	Structure rigoureuse	Flexible, sans hiérarchie imposée
Itérations	Sprint de 2 à 4 semaines	Cycles structurés	Cycles rapides et adaptables
Document	Légère	Moyenne	Minimale
Livrailles	Par fonctionnalités	Par version complète	Par prototypes rapides
Objectif	Pilotage d'équipe agile	Conformité métier	Rapidité de développement
Adapté à	Projets complexes	Projets collaboratifs lourds	Projets courts, souples

TABLE 1.2 – Comparaison entre approches agiles

Après analyse, nous avons choisi d'adopter la méthode **RAD (Rapid Application Development)**. Cette approche repose sur des cycles courts et itératifs, favorisant le prototypage rapide, l'implication active des utilisateurs et une forte réactivité. Elle se distingue par sa capacité à fournir rapidement des résultats concrets, ce qui est essentiel dans des projets à durée limitée comme le nôtre.

1.3.2 Méthodologie choisie : RAD

La méthode RAD repose sur quatre phases principales qui se succèdent de manière itérative :

- **Planification des exigences** : cette phase consiste à identifier les besoins fonctionnels essentiels du système. Dans notre cas, cela incluait la mise en place d'une infrastructure réseau, la simulation des flux, et la sélection des sources de logs pertinentes.
- **Conception utilisateur** : nous avons développé des prototypes de scénarios (ex. : détection d'une attaque brute-force via QRadar), que nous avons ajustés progressivement en fonction des résultats de simulation obtenus dans l'environnement EVE-NG.
- **Construction rapide** : les composants (QRadar, agents de logs, scripts Python) ont été développés, configurés et testés par étapes, avec des feedbacks immédiats sur leur bon fonctionnement.
- **Mise en service** : cette dernière étape a consisté à valider le comportement global du système, notamment la détection d'événements de sécurité, la génération d'alertes, et le bon fonctionnement des automatisations.

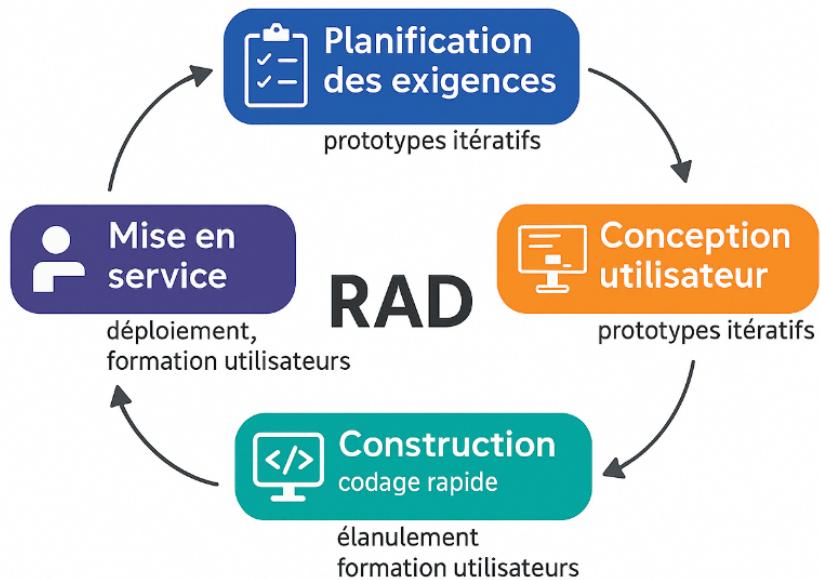


FIGURE 1.3 – Processus RAD

L'approche RAD s'est révélée être un excellent compromis entre agilité, efficacité et simplicité de mise en œuvre. Elle nous a permis de structurer notre travail autour de livrables concrets et d'intégrer rapidement les retours pour améliorer la solution. De plus, la flexibilité qu'elle offre a facilité l'ajustement aux imprévus techniques ou aux nouvelles idées qui ont émergé au cours du développement. Cette méthode a joué un rôle déterminant dans le succès de notre projet, en assurant une progression régulière et une livraison fonctionnelle et maîtrisée de la solution SIEM.

Conclusion

Ce premier chapitre a permis de poser les fondations du projet en présentant le contexte professionnel, les enjeux de sécurité auxquels l'entreprise est confrontée, ainsi que les orientations retenues pour y remédier. La solution IBM QRadar, choisie pour ses capacités avancées de collecte, de corrélation et d'analyse de logs, s'inscrit dans une démarche structurée et méthodique guidée par le modèle RAD. Cette base contextuelle est essentielle pour comprendre la suite du rapport, qui approfondira l'état de l'art, la stratégie de mise en place et la réalisation technique de la solution.

Chapitre 2

État de l'art

Introduction

Ce chapitre présente les concepts clés liés à la cybersécurité et à la gestion centralisée des journaux d'événements. Il s'agit d'établir un socle théorique solide afin de mieux appréhender les enjeux actuels de la sécurité des systèmes d'information, notamment dans le contexte de la détection d'incidents et de la réponse rapide aux menaces. Le chapitre s'articule autour de la notion de SIEM (*Security Information and Event Management*), en présentant ses composantes, ses avantages, ainsi que les solutions existantes sur le marché. Un accent particulier est mis sur la solution IBM QRadar, avec une présentation de ses fonctionnalités, de son architecture et de son rôle dans une infrastructure de sécurité. Enfin, le chapitre traite des méthodes de collecte et de traitement des logs, en intégrant la plateforme EVE-NG utilisée pour l'émulation de l'environnement cible.

2.1 Cybersécurité

La cybersécurité regroupe l'ensemble des pratiques, technologies et processus visant à protéger les systèmes d'information contre les menaces numériques. Elle repose sur trois piliers fondamentaux : la **confidentialité**, l'**intégrité** et la **disponibilité** des données.

Face à la sophistication croissante des attaques (phishing, malwares, DDoS, exploitations de vulnérabilités...), les entreprises doivent mettre en œuvre une défense multicouche. Si les outils traditionnels (pare-feux, antivirus, IDS/IPS) sont indispensables, leur efficacité est limitée sans une vision centralisée et corrélée des événements de sécurité. C'est pourquoi les solutions SIEM (*Security Information and Event Management*) comme **IBM QRadar** sont devenues cruciales : elles permettent de centraliser les journaux issus de différentes sources, de les analyser intelligemment et de déclencher des alertes en cas d'activités suspectes.



FIGURE 2.1 – Les couches de la cybersécurité

2.2 Outils SIEM

2.2.1 Technologie SIEM

Le SIEM, ou *Security Information and Event Management*, est une technologie clé dans la sécurité des systèmes d'information. Il permet de centraliser la collecte de logs, d'analyser les événements de sécurité et de détecter des comportements malveillants en temps réel. Un SIEM regroupe généralement deux fonctions essentielles : la gestion des informations de sécurité (SIM) et la gestion des événements de sécurité (SEM). Grâce à ces fonctionnalités combinées, les SIEM assurent une visibilité complète sur l'ensemble du réseau et facilitent la prise de décision rapide lors d'incidents.

2.2.2 Solutions Existantes

De nombreuses solutions SIEM sont disponibles sur le marché, adaptées à différents contextes d'entreprise. Parmi les plus connues, on trouve :

- **IBM QRadar** : Solution robuste et largement adoptée par les grandes entreprises, elle offre des capacités de corrélation avancée, une architecture modulaire et une grande flexibilité.
- **Splunk** : Connue pour sa puissance d'analyse de données et ses visualisations, il permet une indexation rapide et une recherche en langage naturel.
- **ArcSight** : Proposé par Micro Focus, il est souvent utilisé dans des environnements très réglementés.
- **AlienVault OSSIM** : Une solution open source qui combine des fonctions SIEM avec des outils de détection d'intrusions.
- **ELK Stack** : Regroupe Elasticsearch, Logstash et Kibana, très flexible et personnalisable, mais nécessite plus de configuration.
- **Microsoft Sentinel** : Une solution cloud-native proposée par Azure, qui combine des fonctions SIEM et SOAR pour collecter, analyser et automatiser la réponse aux incidents de sécurité à grande échelle.

Le choix d'une solution SIEM (Security Information and Event Management) dépend de plusieurs critères comme les capacités de collecte, la corrélation, l'intelligence des menaces, l'intégration SOAR,

les tableaux de bord, la scalabilité ou encore le coût. Le tableau suivant compare quatre solutions populaires du marché :

Critères	IBM QRadar	Splunk	ArcSight	Microsoft Sentinel
Collecte des logs	Excellente	Bonne	Bonne	Basée sur Azure
Corrélation avancée	IA + ML	Bonne	Règles statiques	Limitée cloud
Threat intelligence	X-Force	Marketplace rapide	Externe	Intégrée, rapide
Temps réel	Très rapide	Rapide	Moins réactif	Rapide
SOAR	Intégré	En option	Limité	Azure Logic Apps
Analyse réseau	Incluse	Non	Non	Partielle
Tableaux de bord	Personnalisables	Très détaillés	Complexes	Power BI
Scalabilité	Excellente	Haute	Complexe	Facile
Déploiement	Simple	Complexe	Difficile	Cloud-first
Coût	Moyen	Très cher	Cher	Abonnement mensuel

TABLE 2.1 – Comparaison des solutions SIEM

Après comparaison, *IBM QRadar* se démarque par sa richesse fonctionnelle, sa compatibilité avec les environnements hybrides, sa capacité d'analyse en temps réel et son bon rapport qualité/prix.

2.2.3 SIEM retenu

Dans le cadre de ce projet, la solution retenue est **IBM QRadar**, pour ses nombreuses qualités techniques et sa compatibilité avec les environnements professionnels. QRadar se distingue par sa capacité à collecter et analyser les événements de sécurité provenant de sources hétérogènes, à détecter des comportements anormaux grâce à des règles de corrélation, et à fournir des alertes pertinentes pour une réponse rapide. Il permet aussi de répondre aux exigences réglementaires comme le RGPD, ISO/IEC 27001, ou encore PCI-DSS.

2.3 IBM QRadar

2.3.1 Présentation de la Solution

QRadar est un système SIEM développé par IBM, conçu pour offrir une plateforme complète de surveillance et de gestion des incidents de sécurité. Il collecte des données provenant de différentes sources (firewalls, systèmes, bases de données, etc.) et les analyse afin de détecter automatiquement des activités malveillantes ou suspectes. L'un de ses avantages majeurs est sa capacité à s'adapter à une architecture réseau complexe sans compromettre les performances.

2.3.2 Architecture Générale

L'architecture de QRadar repose sur plusieurs composants distribués, chacun ayant un rôle bien défini dans le processus de gestion des événements :

- **Event Collector** : recueille les logs depuis les sources.
- **Event Processor** : normalise et analyse les logs collectés.
- **Flow Processor** : analyse les flux réseau pour détecter des anomalies comportementales.
- **Console** : interface graphique pour l'analyse, la visualisation et la gestion des incidents.

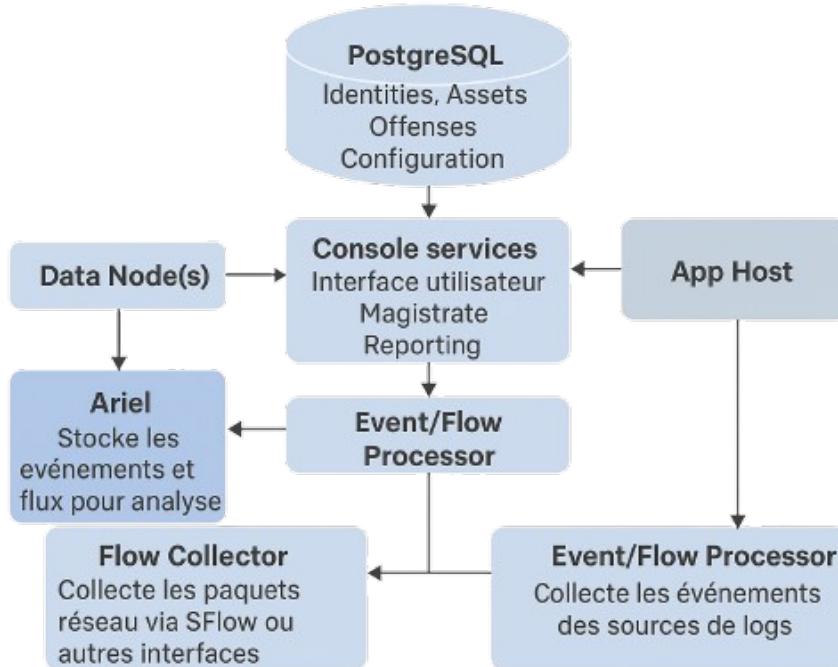


FIGURE 2.2 – Architecture fonctionnelle

Le schéma de la figure 2.3 présente l'architecture fonctionnelle de la solution SIEM IBM QRadar. Les collecteurs d'événements et de flux récupèrent les données des sources réseau, qui sont ensuite traitées par des processeurs pour analyse et corrélation. Les résultats sont stockés dans des bases comme Ariel et PostgreSQL, tandis que la Console assure l'accès utilisateur et la gestion centralisée du système.

2.3.3 Fonctionnalités

QRadar dispose d'un ensemble riche de fonctionnalités qui couvrent l'ensemble du cycle de gestion des incidents. Il permet notamment :

- La collecte centralisée de données à partir de multiples sources.
- La normalisation des événements pour une meilleure lisibilité et traitement.
- La corrélation d'événements afin d'identifier des menaces complexes.
- La génération automatique d'alertes de sécurité.
- L'analyse comportementale et l'intelligence artificielle pour prédire et prévenir les attaques.
- La création de rapports conformes aux normes de sécurité.

L'un des points forts de QRadar est sa capacité d'intégration avec d'autres outils de cybersécurité. Il peut se connecter nativement à des firewalls comme Cisco ASA ou Fortinet Fortigate, à des scanners

de vulnérabilités comme Nessus, et à des systèmes de détection d'intrusions comme Snort. Cela permet d'enrichir l'analyse des logs et d'avoir une visibilité complète sur l'état de sécurité du système.

2.3.4 Infractions

La notion d'infraction (ou *offense*) constitue le cœur du mécanisme de détection des menaces dans IBM QRadar. Contrairement à une alerte isolée issue d'un seul événement, une infraction est un regroupement contextualisé d'événements et de flux réseau corrélés, permettant de détecter des comportements anormaux ou des attaques avancées. QRadar applique des règles de corrélation à l'ensemble des données collectées, afin d'identifier automatiquement des incidents de sécurité pertinents.

Principe de fonctionnement

QRadar collecte les événements à partir de multiples sources (pare-feux, systèmes d'exploitation, applications, bases de données, équipements réseau, etc.) et applique des règles de corrélation en temps réel. Lorsqu'un ensemble d'événements correspond à une règle (ex. : tentative de scan suivie d'une connexion suspecte), QRadar génère une infraction. Cela permet de dépasser la simple surveillance d'événements individuels pour fournir une analyse plus intelligente et automatisée du contexte global de sécurité.

La figure 2.4 illustre le cycle de vie d'une infraction, qui débute par la détection automatique d'un comportement suspect via une règle de corrélation. L'alerte générée est ensuite analysée par un analyste, qui évalue les éléments liés et les impacts. Si nécessaire, des actions correctives sont engagées avant que l'infraction ne soit clôturée, manuellement ou automatiquement.



FIGURE 2.3 – Cycle de vie d'une infraction

Composants d'une infraction

Le tableau 2.2 ci-dessous présente les différents composants d'une infraction dans QRadar, en décrivant leur rôle et leur contribution à l'identification et à l'analyse des incidents de sécurité.

Composant	Description
Règle déclenchée	Décrit le scénario détecté, par exemple : attaque brute force suivie d'une exfiltration de données.
Événements / Flux réseau	Ensemble d'événements et/ou de flux corrélés, accompagnés de métadonnées telles que l'horodatage, les utilisateurs, les ports ou services impliqués.
Hôtes source / destination	Identifiés à partir des ressources du système, ils représentent les machines ou utilisateurs impliqués dans l'infraction.
Score de priorité	Évalue la criticité de l'infraction selon trois axes : - Gravité : niveau de dangerosité de l'activité - Crédibilité : fiabilité de la source ou de l'événement - Importance : valeur critique de l'actif visé

TABLE 2.2 – Composants d'une infraction QRadar

Types courants d'infractions

QRadar fournit de nombreuses règles par défaut (default rules), mais permet également de créer des règles personnalisées (custom rules). Quelques exemples d'infractions fréquentes :

- Tentatives de brute force SSH sur plusieurs machines,
- Scan réseau distribué (slow scan),
- Connexion à un serveur interdit hors des heures de travail,
- Exfiltration de données sur un port non autorisé,
- Changements non autorisés de priviléges utilisateurs.

Visualisation et analyse dans QRadar

Depuis la console QRadar, les infractions sont accessibles via l'onglet *Offenses*. Chaque infraction donne accès à une interface détaillée :

- Timeline des événements corrélés,
- Source et cible identifiées,
- Contexte de l'attaque (utilisateur, géolocalisation IP...),
- Vue graphique des relations entre événements.

L'analyste peut alors prioriser les alertes les plus critiques, marquer les offenses, assigner à un analyste, ou encore archiver les incidents résolus.

Avantages clés du système d'infractions

- Corrélation intelligente des événements isolés,
- Réduction du bruit grâce à l'agrégation des alertes redondantes,
- Hiérarchisation automatique des incidents,
- Gain de temps pour les équipes SOC,
- Base d'analyse complète pour la réponse aux incidents.

2.4 Gestion des Logs

2.4.1 Types

Dans le contexte d'une solution SIEM comme QRadar, les logs représentent des sources d'information critiques pour la surveillance. On distingue plusieurs types de journaux :

- **Les logs systèmes** (Windows, Linux) : Ils contiennent des informations sur l'activité du système d'exploitation, comme les connexions, les erreurs ou les redémarrages.
- **Les logs réseaux** (switches, firewalls) : Ils permettent de suivre le trafic réseau, les connexions autorisées ou bloquées, et les anomalies de communication.
- **Les journaux de sécurité** (IDS/IPS) : Ils signalent les menaces détectées, les attaques bloquées et les comportements malveillants potentiels.

2.4.2 Protocoles de Collecte

QRadar prend en charge plusieurs protocoles standards pour la collecte des logs, permettant une compatibilité étendue avec l'infrastructure existante :

- **Syslog** : protocole standard largement utilisé pour la transmission des logs.
- **SNMP** : utilisé pour la supervision et les alertes d'équipements réseau.
- **WinCollect** : agent de collecte pour les environnements Windows.

2.4.3 Automatisation

Dans une architecture SIEM telle que **QRadar**, la gestion centralisée des logs permet non seulement de surveiller les événements en temps réel, mais aussi de déclencher automatiquement des réponses face à certaines menaces détectées. C'est dans ce contexte que l'automatisation des réponses prend tout son sens.

En effet, chaque log collecté et corrélé dans QRadar peut devenir un déclencheur d'action. Grâce à l'analyse des journaux (logs) provenant des pare-feux, serveurs, IDS ou autres dispositifs, QRadar peut détecter des comportements suspects ou des incidents critiques, puis appliquer automatiquement des mesures de remédiation.

Dans le cadre de ce projet, l'automatisation repose sur des **scripts personnalisés**, écrits en Python, déclenchés par des règles définies à partir des événements consignés dans les journaux. Ce mécanisme s'intègre directement dans le processus de gestion des logs pour transformer une simple détection en action concrète.

Exemples d'actions automatisées basées sur l'analyse des logs :

- **Blocage d'IP malveillante** : si un log indique une tentative d'accès non autorisée, un script peut envoyer une commande au pare-feu FortiGate pour bloquer immédiatement l'adresse IP en cause.
- **Alerte par e-mail** : lorsqu'un type spécifique de log critique est détecté, un message peut être automatiquement envoyé à l'administrateur.
- **Journalisation des incidents** : les détails d'un événement peuvent être enregistrés dans un fichier texte pour archivage ou audit.
- **Isolation d'un hôte** : si un log signale une compromission, un script peut lancer une commande à distance pour désactiver temporairement la connexion réseau de la machine ciblée.

Ce type d'automatisation améliore considérablement la **réactivité** du système de sécurité. Elle **réduit le temps de réponse** aux incidents, tout en assurant un traitement homogène et traçable

des alertes issues des logs. Cette approche reste volontairement simple, afin de rester compatible avec les ressources disponibles et les objectifs pédagogiques du projet.

2.5 Plateforme EVE-NG

2.5.1 Présentation

EVE-NG (*Emulated Virtual Environment – Next Generation*) est une plateforme de virtualisation et d'émulation réseau professionnelle, utilisée dans les environnements académiques, industriels et notamment chez **Network Associates** (NA) pour la conception, les tests et la simulation d'architectures complexes.

Elle permet de créer des laboratoires virtuels en intégrant une grande diversité d'éléments : routeurs, pare-feux, machines Windows/Linux, solutions de cybersécurité, etc. Accessible via une interface web intuitive, EVE-NG supporte l'importation de nombreuses images système telles que celles de Cisco, Palo Alto, Fortinet, Windows Server, Ubuntu, ainsi que des appliances comme QRadar. Elle offre aux utilisateurs un contrôle total sur la topologie réseau, les connexions et les interactions entre les composants.



FIGURE 2.4 – Logo EVE-NG

2.5.2 Rôle

Dans le cadre de ce projet, EVE-NG a été choisie pour servir de plateforme de simulation et de test. Son rôle a été central dans toutes les phases techniques du projet, remplaçant une infrastructure physique coûteuse et difficile à déployer.

La plateforme a permis de :

- Créer un environnement réseau réaliste, composé de deux sites sécurisés par des pare-feux FortiGate, interconnectés via un tunnel VPN IPsec.
- Déployer et configurer une instance de QRadar dans une machine virtuelle dédiée, pour la collecte et l'analyse des logs.
- Simuler plusieurs sources de logs, telles que des serveurs (Linux, Windows), un IDS (Snort), un scanner de vulnérabilités (Nessus), et des postes clients.
- Configurer des scénarios d'attaque simulés afin de générer des incidents détectables par le SIEM QRadar.
- Tester les scripts Python d'automatisation dans un environnement isolé, en s'assurant qu'ils n'affectent pas le système global.

EVE-NG étant déjà utilisée par Network Associates dans ses environnements internes, sa maîtrise et son intégration dans ce projet se sont avérées naturelles et cohérentes avec les pratiques professionnelles de l'entreprise.

2.5.3 Avantages

L'utilisation d'EVE-NG dans ce projet a offert plusieurs bénéfices déterminants pour la réussite du déploiement de la solution SIEM :

- **Souplesse de conception** : l'ajout, la suppression ou la modification de nœuds réseau se fait rapidement et sans interruption.
- **Réalisme des scénarios** : les appliances et systèmes utilisés sont identiques à ceux déployés dans des environnements professionnels.
- **Sécurité de l'environnement** : tout est isolé du réseau réel, ce qui permet de tester des attaques, des erreurs ou des comportements critiques sans aucun risque.
- **Réduction des coûts** : aucun matériel physique n'est requis pour la simulation, ce qui rend le projet plus accessible.
- **Expérimentation illimitée** : les snapshots et la sauvegarde des topologies permettent de revenir à un état antérieur pour relancer un test ou corriger une erreur.
- **Approche pédagogique** : EVE-NG offre une visualisation claire des liens réseau et des configurations, ce qui facilite l'apprentissage et la documentation.

En résumé, EVE-NG a permis de construire une architecture réseau complète, modulaire, et adaptée au déploiement de QRadar dans un cadre de test. Cette plateforme, largement utilisée par Network Associates, a été un élément clé dans la validation des choix techniques, la formation à l'utilisation des outils de sécurité, et la mise en œuvre des scénarios d'automatisation.

Conclusion

En résumé, ce chapitre a permis de dresser un panorama des technologies de sécurité centrées sur la gestion des événements et des informations de sécurité. L'étude des solutions SIEM a mis en lumière les avantages de l'approche centralisée pour améliorer la visibilité, la détection et la réaction face aux incidents. L'analyse spécifique de QRadar a apporté une compréhension approfondie de ses capacités, justifiant son intégration dans la solution proposée. Cette base théorique constitue un appui essentiel pour le choix de l'architecture et la mise en œuvre décrite dans les chapitres suivants.

Chapitre 3

Stratégie de mise en place de la solution

Introduction

Ce chapitre décrit la stratégie suivie pour concevoir et planifier la mise en œuvre de la solution de sécurité basée sur IBM QRadar. Il commence par une étude préalable qui comprend l'analyse de la problématique de sécurité, la définition des objectifs à atteindre, et les critères ayant conduit au choix de QRadar parmi d'autres SIEM. Ensuite, le chapitre présente l'architecture réseau cible, les composants principaux à intégrer (comme les pare-feu, les agents de collecte, les serveurs, etc.), ainsi que les mécanismes de réponse automatique aux incidents. Enfin, une étude fonctionnelle est menée afin de spécifier les fonctionnalités attendues, aussi bien au niveau réseau, SIEM que de l'automatisation des tâches de sécurité.

3.1 Étude préalable

3.1.1 Problématique

Face à l'augmentation du volume des attaques informatiques et au besoin croissant de supervision efficace, Network Associates ne disposait pas d'une solution centralisée capable d'analyser, de corrélérer et de traiter automatiquement les événements de sécurité. Le principal défi résidait dans la mise en place d'un mécanisme de collecte centralisée des journaux de sécurité et dans l'automatisation de la détection des menaces afin de renforcer la posture de sécurité globale de l'entreprise.

3.1.2 Objectifs

L'objectif principal de la solution est de mettre en place un système de gestion centralisée des logs capable de collecter, corrélérer et analyser les événements provenant de l'ensemble des équipements critiques tels que les pare-feux, les serveurs et les postes utilisateurs. Il s'agit de détecter en temps réel toute activité suspecte, de générer automatiquement des alertes de sécurité et d'automatiser certaines réponses aux incidents afin de renforcer la posture de cybersécurité de l'organisation. La solution doit également faciliter la supervision à travers des tableaux de bord dynamiques et l'édition de rapports personnalisés, tout en garantissant la conformité avec les exigences réglementaires telles que le RGPD, la norme ISO/IEC 27001 et PCI-DSS.

3.1.3 Choix IBM QRadar

Le choix de QRadar repose sur ses performances reconnues dans le domaine des SIEM (Security Information and Event Management). QRadar offre une gamme complète de fonctionnalités allant de la collecte multi-source (via Syslog, WinCollect, agents tiers ou API REST) à l'analyse en temps réel avec détection d'incidents, jusqu'à la génération de rapports personnalisés.

L'un de ses atouts majeurs est sa capacité à assurer la centralisation des logs de manière efficace et sécurisée. QRadar permet de regrouper les journaux provenant de sources hétérogènes — pare-feux, serveurs, postes clients, équipements réseau — dans une base de données unifiée. Cette centralisation simplifie l'analyse, favorise la détection de menaces transversales, et permet la mise en place de règles de corrélation sophistiquées. Chaque événement journalisé est normalisé, enrichi, indexé, puis mis à disposition à travers une interface centralisée et intuitive.

Cette approche permet de :

- Réduire les silos d'information et améliorer la visibilité globale du système d'information
- Identifier plus rapidement les comportements anormaux répartis sur plusieurs machines
- Traiter les logs en temps réel avec un moteur de corrélation performant
- Faciliter les audits et la traçabilité à travers des rapports détaillés et filtrables.

De plus, sa version Community Edition permet de tester l'ensemble de ses fonctionnalités dans un cadre académique, tout en reproduisant une architecture réaliste. QRadar est également compatible avec l'intégration de scripts externes (notamment en Python) et supporte une API REST complète, ce qui le rend particulièrement adapté à l'automatisation des tâches de remédiation ou de notification. Ce choix s'aligne donc parfaitement avec les objectifs pédagogiques, opérationnels et techniques du projet.

3.2 Solution adoptée

3.2.1 Architecture réseaux

L'architecture réseau mise en œuvre dans notre solution est basée sur deux sites géographiquement distants, reliés de manière sécurisée par un tunnel VPN IPsec configuré directement au niveau des pare-feux FortiGate. Cette configuration permet une communication chiffrée et fiable entre les deux réseaux internes tout en assurant l'intégrité et la confidentialité des données échangées.

Chaque site est composé des éléments suivants, organisés selon une topologie hiérarchique :

- Un pare-feu FortiGate agissant comme point d'entrée et de sécurisation du réseau.
- Un commutateur (switch) pour l'interconnexion des équipements locaux.
- Un poste de travail (PC utilisateur ou serveur).
- Une connexion Internet via un routeur ou modem.

Sur chaque FortiGate, l'interface port1 est dédiée à la connexion WAN (Internet), tandis que port2 assure la liaison avec le réseau local interne à travers le commutateur. Le tunnel VPN IPsec est établi entre les deux équipements FortiGate, à l'aide d'une configuration complète intégrant :

- des adresses IP publiques statiques pour chaque pare-feu.
- une négociation IKE (Internet Key Exchange) en phase 1 avec chiffrement AES et authentification par clé pré-partagée.
- une phase 2 précisant les sous-réseaux internes autorisés à communiquer à travers le tunnel.
- des routes statiques pour acheminer le trafic vers le réseau distant.
- des règles de pare-feu permettant le passage du trafic IPsec.

Cette infrastructure assure une connectivité inter-site robuste, permettant à chaque poste de travail de communiquer avec les ressources du site distant de manière transparente et sécurisée, tout en préparant le terrain à une centralisation efficace des journaux de logs par la solution SIEM QRadar.

La configuration réseau repose principalement sur la mise en place d'un tunnel VPN IPsec entre deux pare-feux FortiGate situés sur les sites A et B. Chaque FortiGate a été configuré avec une interface WAN connectée à Internet et une interface LAN reliée à l'infrastructure locale. Le tunnel

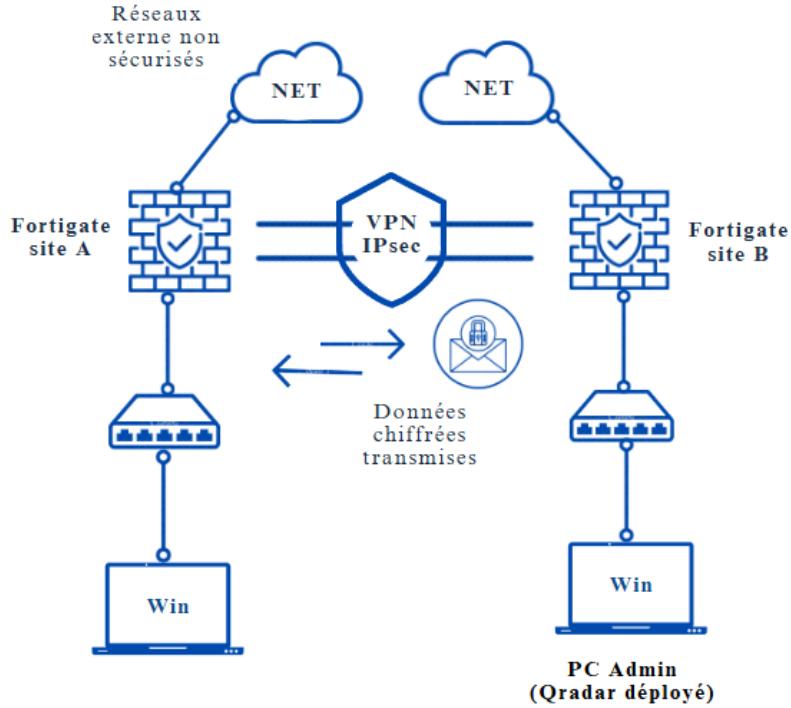


FIGURE 3.1 – Architecture réseau

VPN IPsec est configuré avec une politique de phase 1 (IKE) et phase 2 (IPsec), définissant les paramètres de chiffrement, l’authentification (pré-partagé ou certificat) et les réseaux locaux à inclure dans le tunnel. Les règles de pare-feu (firewall policies) sont ajoutées pour autoriser le trafic entre les sous-réseaux locaux à travers le VPN. Des routes statiques sont également configurées pour assurer la bonne redirection du trafic entre les deux LAN via le tunnel. Cette configuration permet une communication sécurisée entre les équipements des deux sites, tout en protégeant les données échangées contre les interceptions extérieures.

3.2.2 Composants principaux

Cependant, pour que l’outil SIEM atteigne sa pleine efficacité, il doit être alimenté par des sources fiables et variées. C’est dans cette optique qu’interviennent des outils complémentaires tels que les pare-feux de nouvelle génération et, potentiellement, des scanners de vulnérabilités. Leur intégration avec QRadar renforce considérablement les capacités de détection et de réponse, tout en augmentant la visibilité globale sur les infrastructures informatiques.

Pare-feu et fonctionnalités IDS/IPS intégrées

Le pare-feu constitue la première ligne de défense contre les menaces extérieures. Il filtre le trafic réseau en fonction de règles prédéfinies, empêchant ainsi les connexions non autorisées et assurant la protection des ressources critiques. Dans notre projet, le choix s’est porté sur la solution **FortiGate**, un NGFW (*Next-Generation Firewall*) réputé pour sa richesse fonctionnelle.

FortiGate intègre nativement un moteur **IPS** avancé, ce qui permet de détecter, prévenir et bloquer automatiquement les activités malveillantes sans nécessiter l’ajout d’un IDS externe.

Les principales fonctionnalités offertes par FortiGate sont :

- l’inspection approfondie des paquets (DPI),
- un moteur IPS embarqué pour la prévention proactive,
- le filtrage web et applicatif pour contrôler l’accès aux ressources,

- o la gestion unifiée des menaces (UTM).

Les logs générés par FortiGate, relatifs aux connexions autorisées, bloquées et aux tentatives d'intrusion détectées, sont directement envoyés à QRadar pour :

- o assurer une visibilité globale sur le trafic réseau,
- o identifier rapidement les anomalies de communication,
- o corrélérer efficacement ces événements avec d'autres sources de sécurité.

VPN IPsec Site-to-site

Pour garantir la confidentialité, l'intégrité et la disponibilité des communications entre les différentes infrastructures du projet, un tunnel **VPN IPsec site-to-site** a été établi entre les deux équipements FortiGate. Ce tunnel sécurisé assure l'échange fiable des journaux d'événements (logs) avant leur centralisation dans le système QRadar.

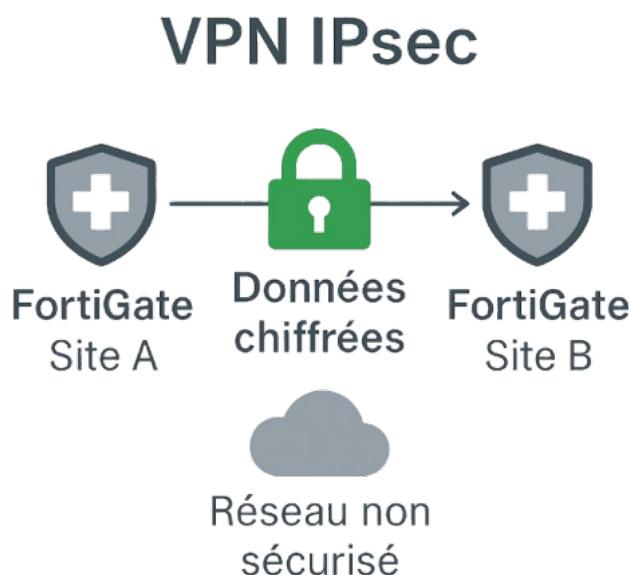


FIGURE 3.2 – VPN site à site

Le VPN IPsec site-to-site offre plusieurs avantages :

- o Chiffrement des communications pour protéger les données sensibles contre les interceptions.
- o Authentification mutuelle entre les équipements pour éviter tout accès non autorisé.
- o Résistance accrue aux attaques de type *Man-in-the-Middle*.
- o Garantie de la confidentialité, de l'intégrité et de l'authenticité des données échangées entre les sites.
- o Maintien d'une connectivité sécurisée permanente entre les réseaux distants.

Ainsi, le VPN IPsec site-to-site constitue un élément clé dans l'architecture réseau, en assurant une collecte des logs fiable et sécurisée pour leur traitement par QRadar.

Transfert d'images système avec WinSCP

Dans le cadre de la mise en place de l'environnement de test virtualisé et de l'intégration des équipements réseau, **WinSCP** a été utilisé comme outil principal de transfert de fichiers. WinSCP est un client graphique pour les protocoles SFTP, SCP et FTP, permettant un échange sécurisé entre une machine locale Windows et un hôte distant Linux.

Son utilisation a été cruciale dans ce projet :

- **Transfert des images des systèmes d'exploitation des équipements (OS)** : WinSCP a permis d'importer les images nécessaires au bon fonctionnement des routeurs, firewalls et autres appliances dans la plateforme EVE-NG.

Grâce à cet outil, les opérations de déploiement ont pu être menées de manière fluide, sécurisée et centralisée, contribuant à la mise en œuvre efficace de la plateforme SIEM.

3.2.3 Réponse automatisée aux incidents

Dans le cadre du projet, une réponse automatisée aux incidents a été mise en place afin d'améliorer la réactivité du système de détection basé sur QRadar. Cette automatisation s'appuie à la fois sur les règles de détection intégrées dans QRadar et sur des scripts Python développés spécifiquement pour surveiller et sécuriser certains composants critiques, comme le pare-feu Windows.

L'implémentation repose sur l'utilisation de l'API REST de QRadar, qui permet de transmettre des alertes vers un script externe en cas de détection d'événements critiques, tels qu'une désactivation du pare-feu, une tentative d'accès non autorisé ou un comportement anormal. Le script principal, a été conçu pour surveiller en temps réel l'état du pare-feu local. Lorsqu'un incident est détecté, il exécute automatiquement une série d'actions correctives :

- Vérification de l'état des trois profils réseau (Domaine, Privé, Public) toutes les 10 secondes ;
- Réactivation automatique du pare-feu si l'un des profils est désactivé ;
- Envoi d'un e-mail à l'administrateur système avec les détails de l'incident (heure, profil concerné, action entreprise) ;
- Affichage d'un message de confirmation dans la console PowerShell ;
- Enregistrement local de l'événement dans un fichier journal.

L'exécution du script nécessite l'environnement Python 3.13.3 ainsi que les bibliothèques `requests` et `pandas`, installées via `pip`. L'automatisation a été validée par des scénarios de test réalistes, comme la désactivation manuelle du pare-feu, simulant une faille. Le script a correctement détecté l'anomalie, rétabli la configuration sécurisée, et notifié l'équipe SOC.

Cette approche démontre l'intérêt d'un couplage entre un moteur SIEM comme QRadar et des scripts personnalisés, sans recourir à des plateformes externes d'orchestration. Elle assure une réponse rapide, traçable et adaptable, tout en étant réalisable dans un environnement à ressources limitées comme celui du projet.

3.3 Étude fonctionnelle

3.3.1 Fonctionnalités réseau

Dans le but de garantir une collecte sécurisée et fiable des logs dans l'environnement de Network Associates, plusieurs fonctionnalités réseau ont été mises en place pour répondre aux exigences de stabilité, de confidentialité et de performance.

Ces fonctionnalités sont les suivantes :

- Mise en place d'un tunnel **VPN IPsec** entre les deux équipements FortiGate pour assurer un transfert sécurisé des journaux d'événements.
- Attribution d'**adresses IP fixes** aux serveurs critiques, pare-feux et à l'instance QRadar pour garantir une communication stable et prédictible.
- Utilisation de **règles de filtrage** sur les équipements réseau pour contrôler le trafic autorisé vers le serveur QRadar.

3.3.2 Fonctionnalités SIEM

Le déploiement de la solution SIEM basée sur IBM QRadar vise à fournir une surveillance avancée des événements de sécurité et à optimiser la détection des menaces dans l'infrastructure existante. Les fonctionnalités exploitées dans ce projet sont :

- **Collecte centralisée** des événements via les protocoles Syslog et WinCollect pour Windows.
- **Normalisation automatique** des logs collectés afin d'uniformiser les formats d'événements provenant de sources hétérogènes.
- **Corrélation intelligente** entre différents événements pour identifier des incidents complexes et détecter des comportements suspects.
- **Priorisation des alertes** selon leur criticité pour permettre aux administrateurs de se concentrer sur les incidents les plus urgents.
- **Génération de rapports de sécurité** personnalisés et exportables pour répondre aux besoins d'audit et de conformité.

3.3.3 Fonctionnalités d'automatisation

Dans le but de renforcer l'efficacité de la détection et de la réponse aux incidents, plusieurs mécanismes d'automatisation ont été intégrés dans notre projet. Plutôt que de s'appuyer uniquement sur les fonctions natives de QRadar, nous avons mis en œuvre une approche hybride basée sur des scripts Python autonomes, exécutés localement à partir d'une surveillance régulière du système.

Les principales fonctionnalités automatisées sont les suivantes :

- **Surveillance périodique du pare-feu Windows** : un script Python vérifie toutes les 10 secondes l'état de protection des trois profils réseau (Domaine, Privé, Public).
- **Réactivation automatique du pare-feu** : en cas de désactivation détectée, le script déclenche une commande de réactivation immédiate pour restaurer la protection.
- **Envoi d'un e-mail d'alerte** : une notification est générée automatiquement et envoyée par courrier électronique à l'administrateur, avec les détails de l'incident détecté.
- **Traçabilité et archivage local** : les événements détectés, ainsi que les actions correctives effectuées, sont enregistrés dans des fichiers journaux dédiés pour consultation ultérieure.

Conclusion

Ce chapitre a permis de poser les bases conceptuelles et organisationnelles nécessaires à la réalisation technique du projet. Grâce à une analyse rigoureuse des besoins et à un choix réfléchi des outils, une stratégie claire a été définie pour garantir l'efficacité et la cohérence de la solution. Cette stratégie guide les étapes de déploiement technique présentées dans le chapitre suivant, en assurant une transition fluide entre la phase de conception et celle de mise en œuvre.

Chapitre 4

Déploiement des outils

Introduction

Ce chapitre retrace les différentes étapes techniques mises en œuvre pour déployer les composants nécessaires à la solution de gestion des logs basée sur IBM QRadar. Le déploiement s'est appuyé sur l'utilisation de la plateforme d'émulation EVE-NG, l'installation de QRadar, la mise en place de l'agent WinCollect sur des hôtes Windows, ainsi que le développement de scripts Python pour automatiser certaines réponses aux incidents. L'objectif est d'obtenir une infrastructure fonctionnelle, intégrée et adaptée au traitement centralisé des journaux.

4.1 Mise en place d'émulateur : EVE-NG

4.1.1 Téléchargement et installation

L'image ISO officielle de l'émulateur EVE-NG (Community Edition) a été téléchargée depuis le site officiel, puis utilisée pour l'installation dans une machine virtuelle via un hyperviseur.

Création d'une VM EVE-NG sur Hyper-V

La plateforme EVE-NG (Emulated Virtual Environment Next Generation) a été déployée dans le cadre de notre infrastructure virtuelle afin de simuler un environnement réseau complet et interactif pour les besoins du projet. L'installation a été réalisée sur une machine virtuelle Hyper-V, en utilisant l'image ISO officielle de la version Community Edition, disponible sur le site du projet. Cette version gratuite offre les fonctionnalités essentielles pour la création de laboratoires virtuels multi-vendeurs.

Les paramètres de configuration appliqués à la machine virtuelle sont les suivants :

- **RAM** : 4 Go de mémoire vive statique, allouée pour assurer le bon fonctionnement du système EVE-NG. Bien que cela représente le minimum requis, cette quantité est suffisante pour des laboratoires de petite à moyenne taille.
- **Disque virtuel** : 50 Go d'espace disque alloué dynamiquement, permettant d'installer le système de base et d'ajouter ultérieurement des images d'équipements réseau (routeurs, pare-feu, machines virtuelles, etc.).
- **Génération** : 2, avec prise en charge du démarrage UEFI, ce qui améliore la compatibilité avec les systèmes modernes et permet une meilleure gestion des ressources.
- **ISO EVE-NG** : montée en tant que lecteur CD/DVD virtuel pour procéder à l'installation initiale du système.

Cette configuration a été définie avant la création finale de la VM, afin d'assurer une compatibilité optimale avec les exigences de la plateforme EVE-NG.

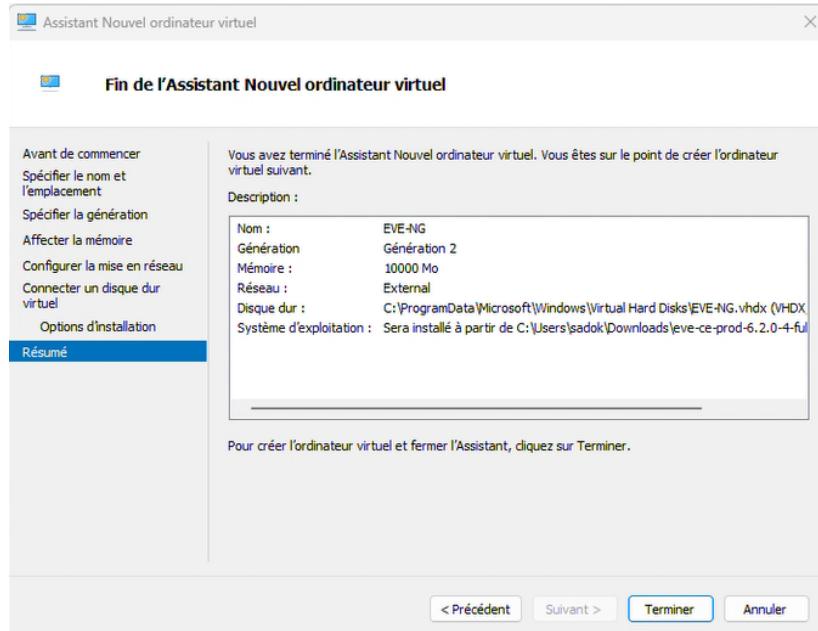


FIGURE 4.1 – Résumé de la configuration de la machine virtuelle EVE-NG sous Hyper-V

L'image ci-dessus illustre le récapitulatif des paramètres appliqués à la machine virtuelle juste avant son déploiement. Ces réglages sont essentiels pour garantir la stabilité de l'environnement de simulation, particulièrement lors de l'exécution simultanée de plusieurs nœuds virtuels.

Configuration avant l'installation d'EVE-NG

Avant de procéder à l'installation d'EVE-NG, une configuration de base est nécessaire. Celle-ci inclut le montage du CD-ROM d'installation, la sélection de la langue, du clavier, la configuration du nom d'hôte, du mot de passe administrateur, et surtout les paramètres réseau pour l'accès à distance via l'interface web.

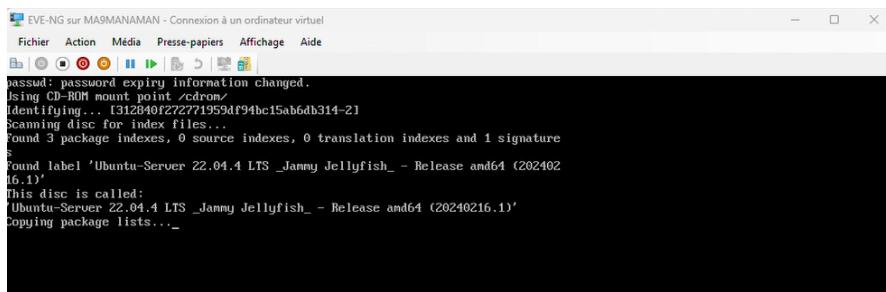


FIGURE 4.2 – Montage du CD-ROM et préparation de l'installation

Lors de l'installation, les paramètres réseau sont configurés manuellement comme suit :

- **Adresse IP** : 192.168.100.55
- **Masque de sous-réseau** : 255.255.255.0
- **Passerelle par défaut** : 192.168.100.1
- **DNS primaire** : 8.8.8.8
- **DNS secondaire** : 8.8.4.4

Ces paramètres permettent d'assurer l'accessibilité de la machine EVE-NG depuis le réseau local, notamment via son interface web. Une fois la configuration de base terminée, l'installation se poursuit normalement jusqu'à l'affichage de l'écran de démarrage.

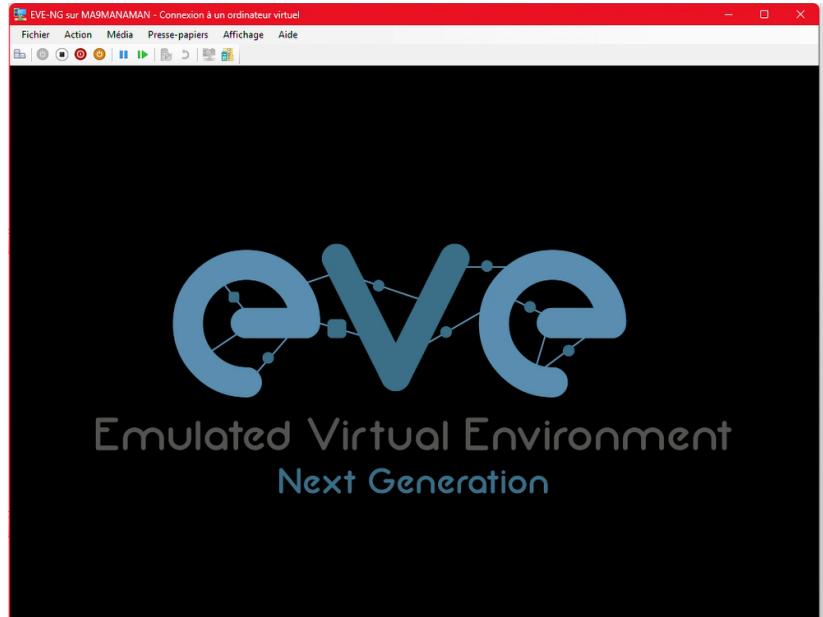


FIGURE 4.3 – Écran de démarrage

Cependant, au premier démarrage, un message d'avertissement est apparu signalant l'absence de support pour la virtualisation matérielle (Intel VT-x ou AMD-V), nécessaire au bon fonctionnement d'EVE-NG.

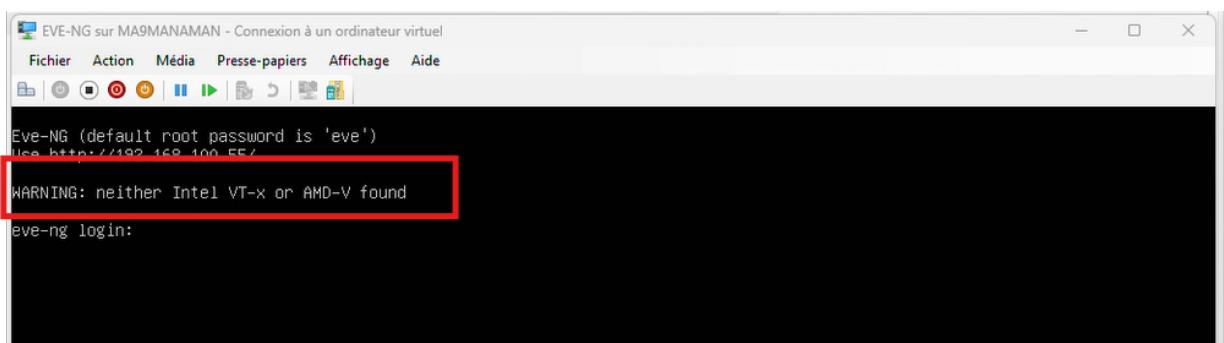


FIGURE 4.4 – Avertissement de l'absence de virtualisation matérielle

Pour corriger cela, une commande PowerShell a été utilisée afin d'activer manuellement l'extension de virtualisation.

```
PS C:\WINDOWS\system32> Set-VMProcessor -VMName "EVE-NG" -ExposeVirtualizationExtensions
Set-VMProcessor : Argument manquant pour le paramètre «ExposeVirtualizationExtensions». Spécifiez un paramètre de type «System.Nullable`1[System.Boolean]» et réessayez.
Au caractère Ligne:1 : 34
+ Set-VMProcessor -VMName "EVE-NG" -ExposeVirtualizationExtensions
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
+ CategoryInfo          : InvalidArgument : () [Set-VMProcessor], ParameterBindingException
+ FullyQualifiedErrorId : MissingArgument,Microsoft.PowerShell.Commands.SetVMProcessor

PS C:\WINDOWS\system32> Set-VMProcessor -VMName "EVE-NG" -ExposeVirtualizationExtensions $true
PS C:\WINDOWS\system32>
```

FIGURE 4.5 – Activation de la virtualisation via PowerShell

Enfin, après redémarrage, la connexion au système a été effectuée avec succès en tant que root.

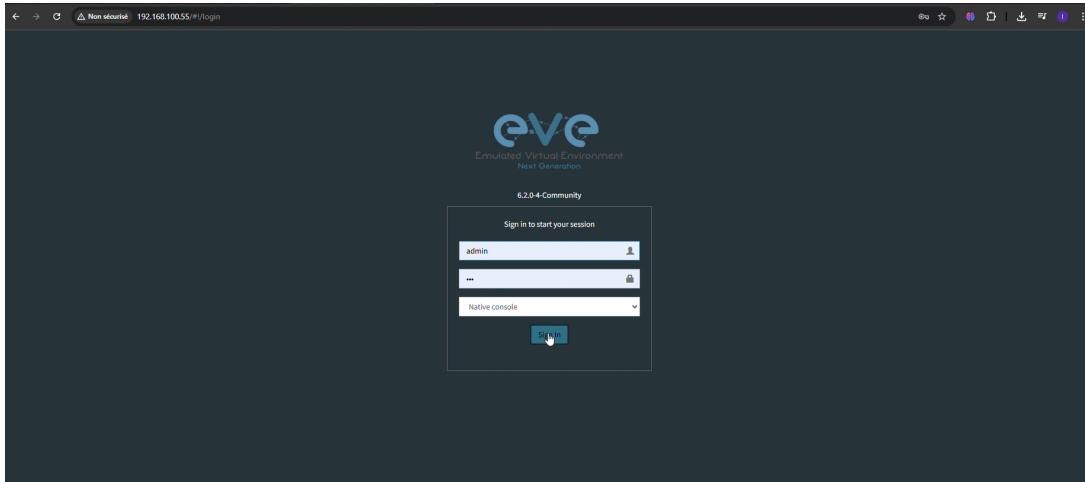


FIGURE 4.6 – Connexion réussie à l’interface web

4.1.2 Préparation d’environnement

WinSCP

WinSCP est un client SFTP, FTP et SCP pour Windows. Il permet de transférer des fichiers de manière sécurisée entre une machine locale Windows et un serveur distant via SSH. Dans le cadre de notre projet, il est utilisé pour transférer les images d’équipements (Windows, Fortinet, etc.) vers la machine virtuelle EVE-NG. Le logiciel est disponible gratuitement sur le site officiel. Il est conseillé de télécharger la dernière version stable.

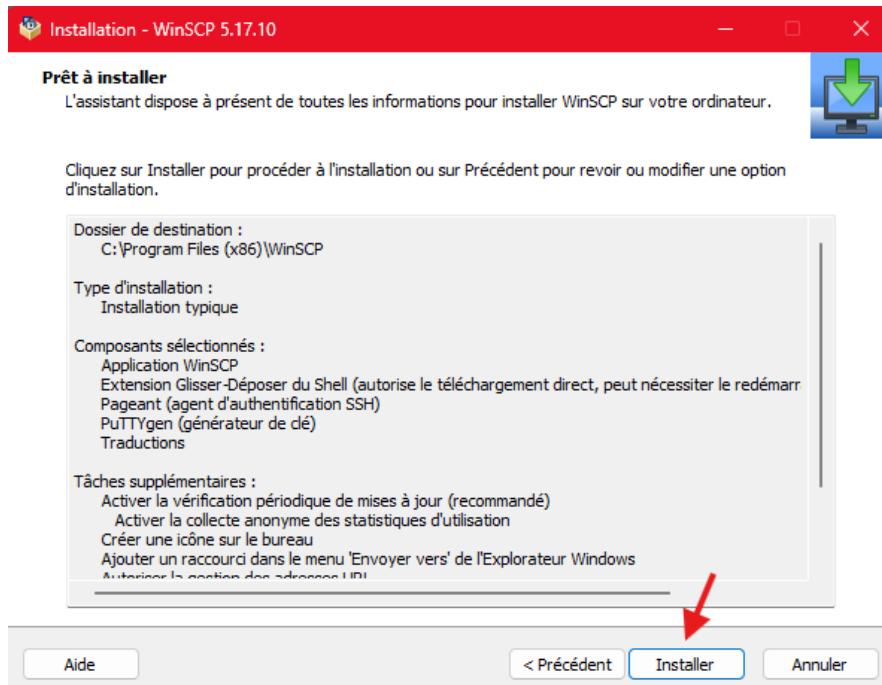


FIGURE 4.7 – Résumé des paramètres d’installation

L’assistant d’installation permet de configurer le type d’installation, les composants à inclure et le chemin d’installation. Une fois ces options validées, l’installation peut démarrer. Une fois l’installation terminée, WinSCP est prêt à être utilisé pour les transferts de fichiers via SCP vers la machine virtuelle EVE-NG.

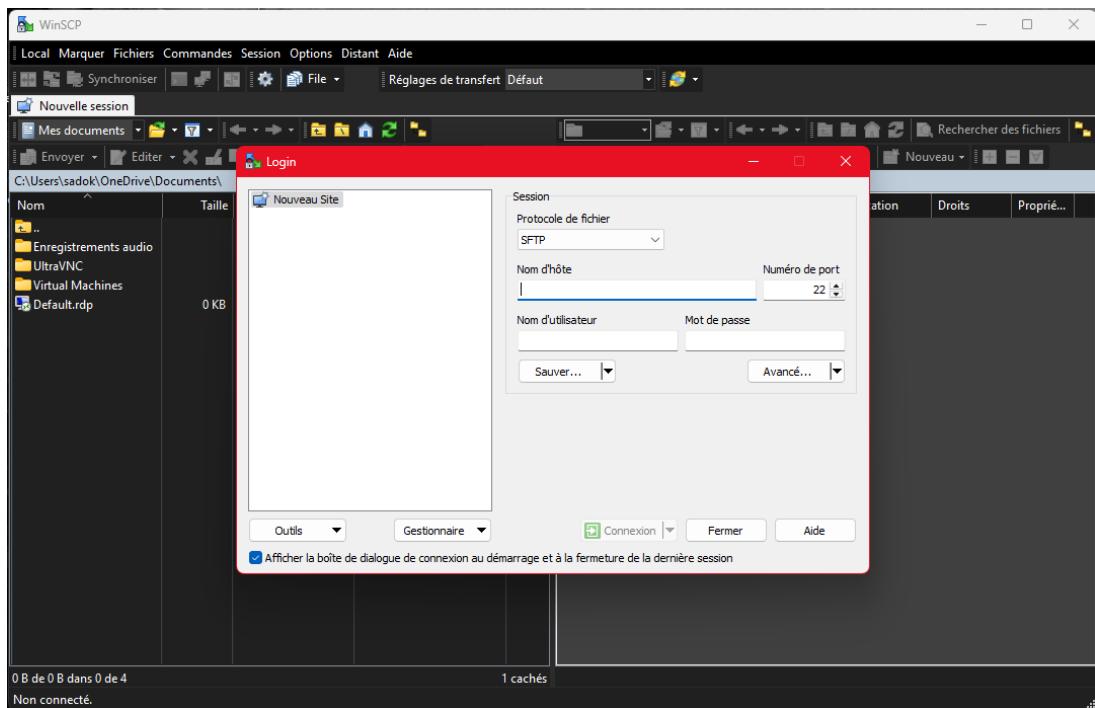


FIGURE 4.8 – page login d’outil WinSCP

ISO des équipements

Après l’installation de la plateforme EVE-NG et de l’outil WinSCP, il a été nécessaire de préparer l’environnement de simulation en y ajoutant les images des équipements réseau. Nous avons téléchargé les images ISO et IOS compatibles avec EVE-NG, en nous assurant qu’elles soient au format supporté (souvent .qcow2) et adaptées à l’émulation QEMU/KVM utilisée par la plateforme.

Les images ont été transférées via WinSCP dans les répertoires appropriés sur EVE-NG, en respectant la structure et la nomenclature exigées (ex. : **fortinet-FGTv5**). Ce respect du format et du nommage permet à EVE-NG de reconnaître et intégrer automatiquement les équipements dans les topologies réseau.

Fortinet		
SR	NAME	DOWNLOAD
1	fortinet-FGT-v7.0.9-build0444	Download here
2	FFW_VM64_KVM-v7.0.13.M-build0566-FORTINET.out.kvm	Download here
3	FFW_VM64_KVM-v7.4.2.F-build2571-FORTINET.out.kvm	Download here
4	fortinet-FGT-v7.2.0-build1157.tgz	Download here
5	fortinet-FGT-v7.2.4.F-build1396(1).tgz	Download here
6	fortinet-FMG-v6.2.3-build1235.zip	Download here
7	fortinet-FMG-v7.2.2-build1334.tgz	Download here

FIGURE 4.9 – Images OS Fortinet

Windows			
SR	NAME	DOWNLOAD	Password[U/P]
1	win-10-Ent-main.zip	Download here	
2	win-10-x64-20H2v2.tgz	Download here	
3	win-10-x64-21H1v1.tgz	Download here	
4	win-10-x86-20H2v3.tgz	Download here	
5	win-10-x86-21H1v1.tgz	Download here	
6	win-11-x64-DEV.tgz	Download here	
7	win-7-Ent-main.zip	Download here	
8	win-8.1-x64-ENT.tgz	Download here	
9	win10-rs5-ltsc-kvm-ttys3.zip	Download here	
10	winserver-2012R2.zip	Download here	
11	winserver-S2012-R2-x64-rev2.tgz	Download here	
12	winserver-S2012-R2-x64.tgz	Download here	
13	winserver-S2019-R2-x64-rev3.tgz	Download here	

FIGURE 4.10 – images OS Windows

Cisco Switches			
SR	CISCO IOS NAME(Folder Name)	DOWNLOAD	TYPE
1	i86bi-linux-l2-adventerprise-15.1b.bin	Download here	L2
2	i86bi-linux-l2-adventerprisek9-15.1a.bin	Download here	L2
3	i86bi-linux-l2-adventerprise-15.1b.zip	Download here	L2
4	i86bi-linux-l2-ipbasek9-15.1a.zip	Download here	L2
5	i86bi-linux-l2-ipbasek9-15.1c.bin	Download here	L2
6	i86bi-linux-l2-ipbasek9-15.1d.bin	Download here	L2
7	i86bi-linux-l2-ipbasek9-15.1e.zip	Download here	L2
8	i86bi-linux-l2-ipbasek9-15.1f.zip	Download here	L2
9	i86bi-linux-l2-ipbasek9-15.1g.zip	Download here	L2
10	i86bi-linux-l2-upk9-12.2.bin	Download here	L2
11	i86bi-linux-l2-upk9-15.0a.bin	Download here	L2
12	i86bi-linux-l2-upk9-15.0a.bin	Download here	L2
13	i86bi-linux-l3-jk9s-15.0.1.bin	Download here	L3
14	i86bi-linux-l3-p-15.0a.bin	Download here	L3
15	i86bi-linux-l3-p-15.0b.bin	Download here	L3
16	i86bi-linux-l3-tpgen-adventerprisek9-12.4.bin	Download here	L3
17	i86bi-linux-l3-tpgen-ipbase-12.4.bin	Download here	L3

FIGURE 4.11 – images OS switechs cisco

Cette section présente les images système utilisées dans EVE-NG, notamment celles de Fortinet (FortiGate, FortiManager) et de Windows. Ces images permettent de simuler différents environnements pour tester des scénarios de sécurité réseau et d'administration système. Nous avons téléchargé les OS de FortiGate (version gratuite), ainsi que celles de Windows 10 et Windows 7 et des switechs cisco, afin de les intégrer à notre environnement virtuel.

Transfert d'images vers EVE-NG

Une fois les fichiers ISO téléchargés, nous avons utilisé l'outil WinSCP pour les transférer vers la machine virtuelle EVE-NG. Ce processus est essentiel pour permettre à EVE-NG d'accéder aux images des équipements à simuler, qu'il s'agisse de systèmes Windows ou de pare-feux Fortinet.

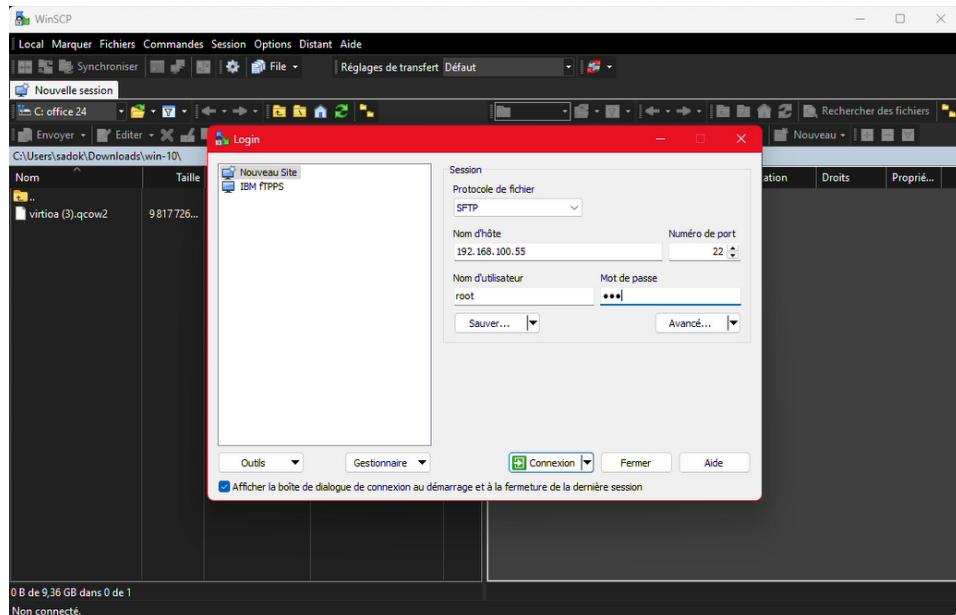


FIGURE 4.12 – Configuration de la session SFTP

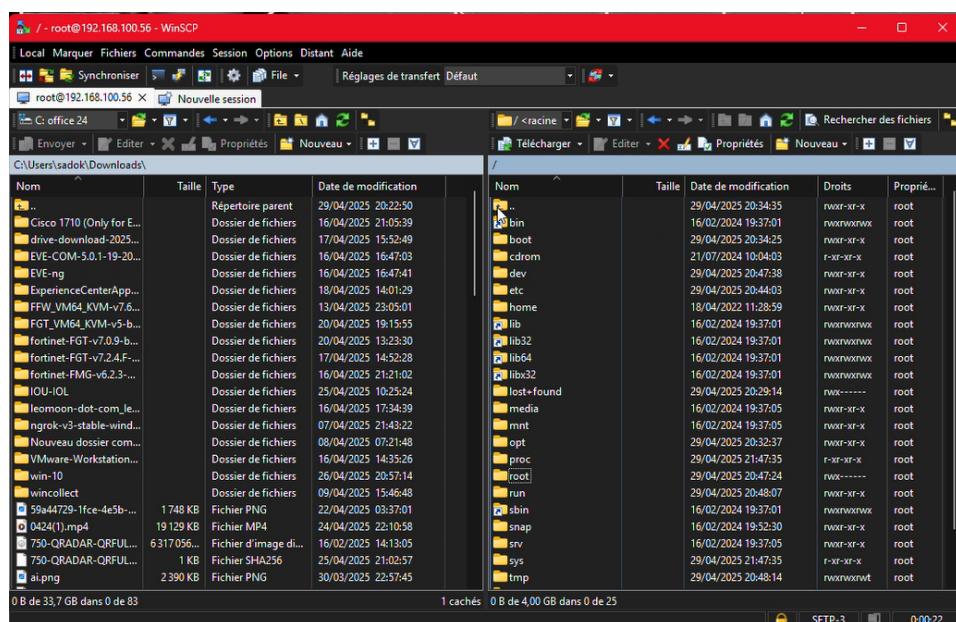


FIGURE 4.13 – Interface WinSCP

L'interface graphique de WinSCP facilite le transfert de fichiers entre le système hôte (Windows) et la machine virtuelle (EVE-NG) via le protocole SFTP. Dans notre cas, nous avons connecté WinSCP

à EVE-NG avec le compte *root*, puis transféré les images nécessaires vers les répertoires appropriés. Cette étape est indispensable pour intégrer les équipements dans la plateforme de simulation.

Ajout d'une image IOS Fortiget

Les images suivantes illustrent les étapes de création du dossier pour l'image Fortinet dans EVE-NG, le transfert de l'image dans ce dossier, puis la vérification du nom du fichier pour garantir sa reconnaissance par la plateforme. sans quoi EVE-NG ne pourra pas les reconnaître ni les utiliser correctement. Par exemple, le nom du dossier doit suivre un format précis comme :

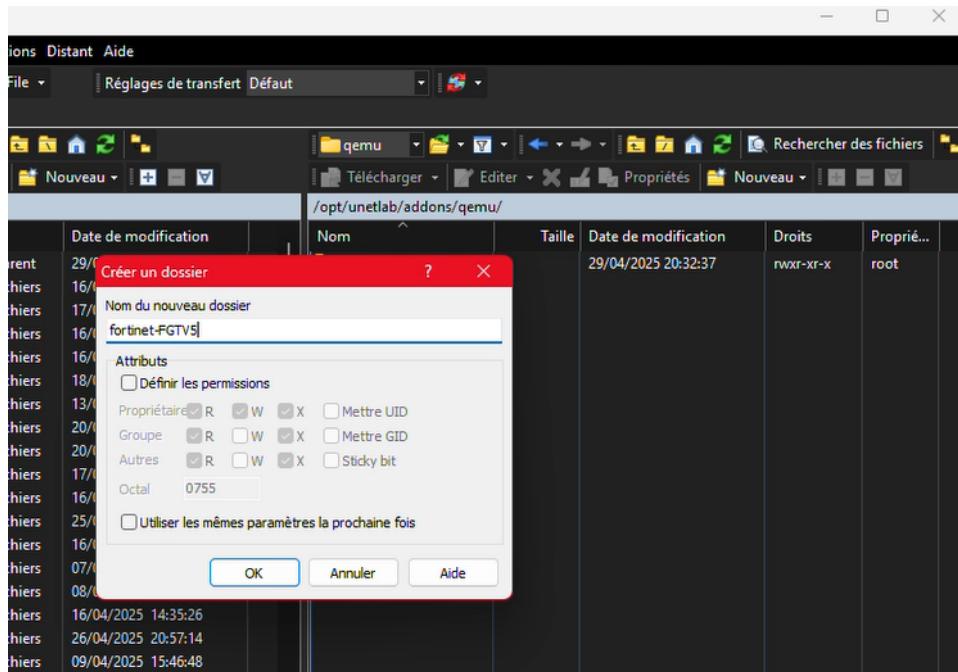


FIGURE 4.14 – dossier "fortinet-FGTv5" pour l'image Fortinet

Le fichier *fortios.qcow2* a ensuite été transféré dans ce dossier.

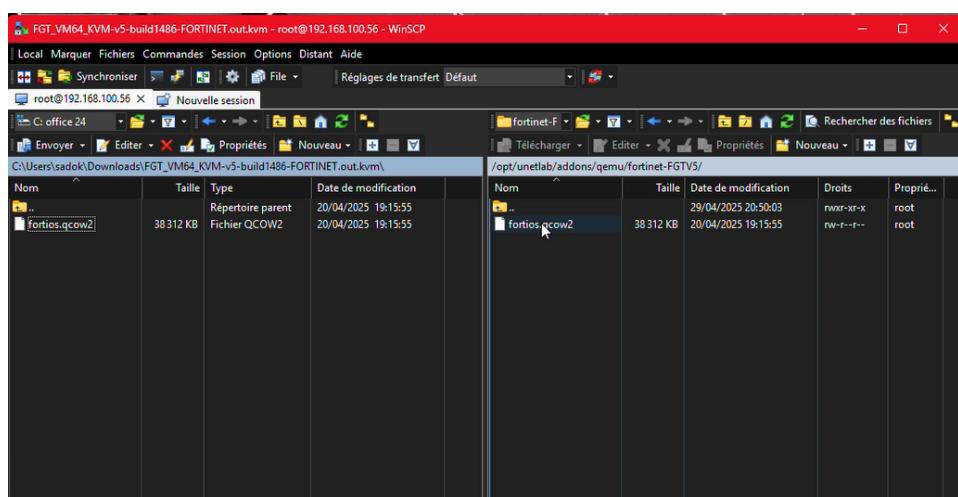


FIGURE 4.15 – Transfert de "fortios.qcow2" dans le dossier approprié

Après chaque transfert d'image, il est indispensable d'exécuter la commande suivante dans le terminal de la VM EVE-NG pour corriger les permissions :

```
/opt/unetlab/wrappers/unl_wrapper -a fixpermissions
```

Cette commande permet l'exécution correcte des images ajoutées, en leur attribuant les droits d'accès adéquats.

Ajout d'une image OS Windows 10

Pour faciliter le diagnostic et les tests dans notre topologie EVE-NG, nous avons intégré une machine virtuelle Windows 10. Cette VM servira de client réseau pour interagir avec le pare-feu FortiGate et vérifier les connexions.

Tout d'abord, nous accédons via WinSCP au répertoire des images QEMU d'EVE-NG afin d'y créer un nouveau dossier dédié à l'image Windows 10.

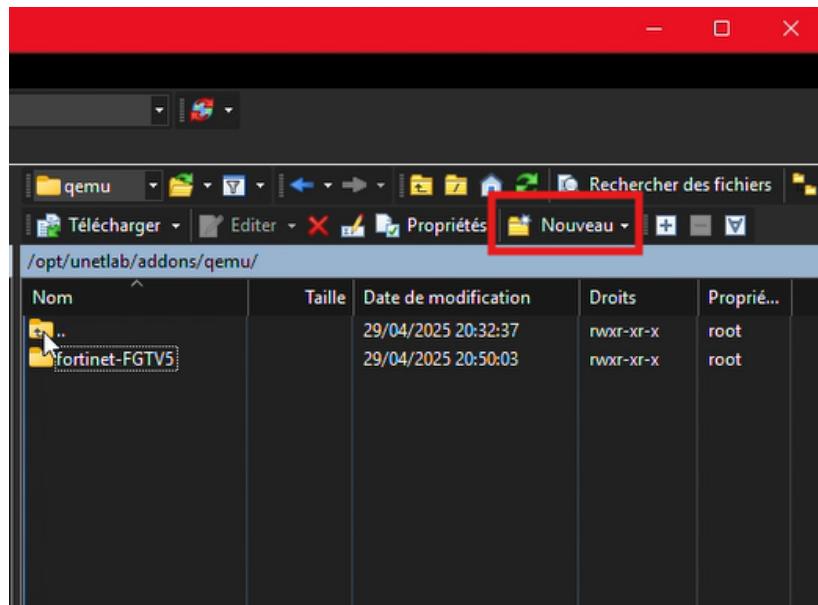


FIGURE 4.16 – Accès au répertoire QEMU

Ensuite, nous créons un dossier nommé **win-10**, en respectant la convention de nommage imposée par EVE-NG (nom en minuscules et sans espaces).

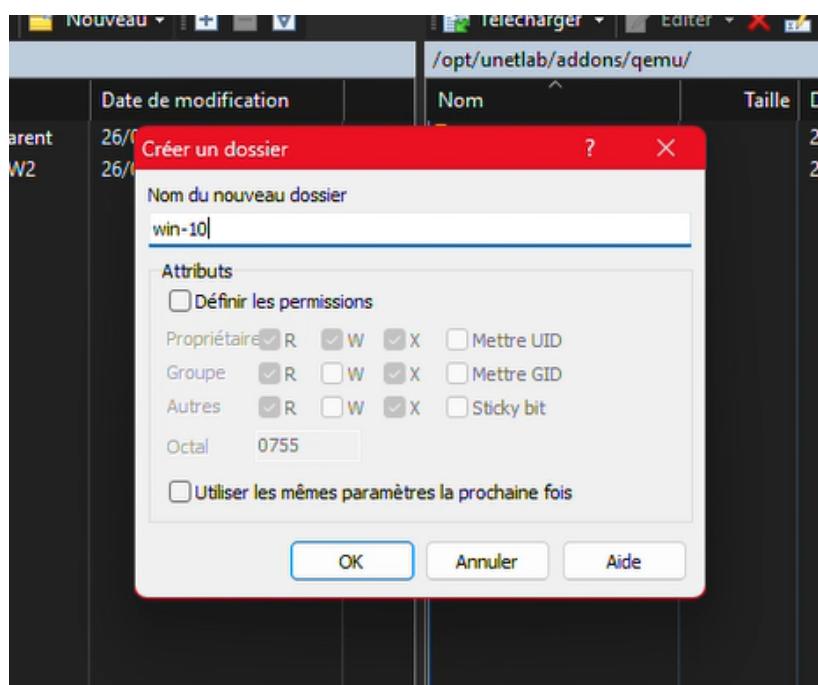


FIGURE 4.17 – Crédit de dossier win-10

Nous transférons ensuite le fichier image .qcow2 de Windows 10 vers ce dossier via l'interface de transfert de WinSCP.

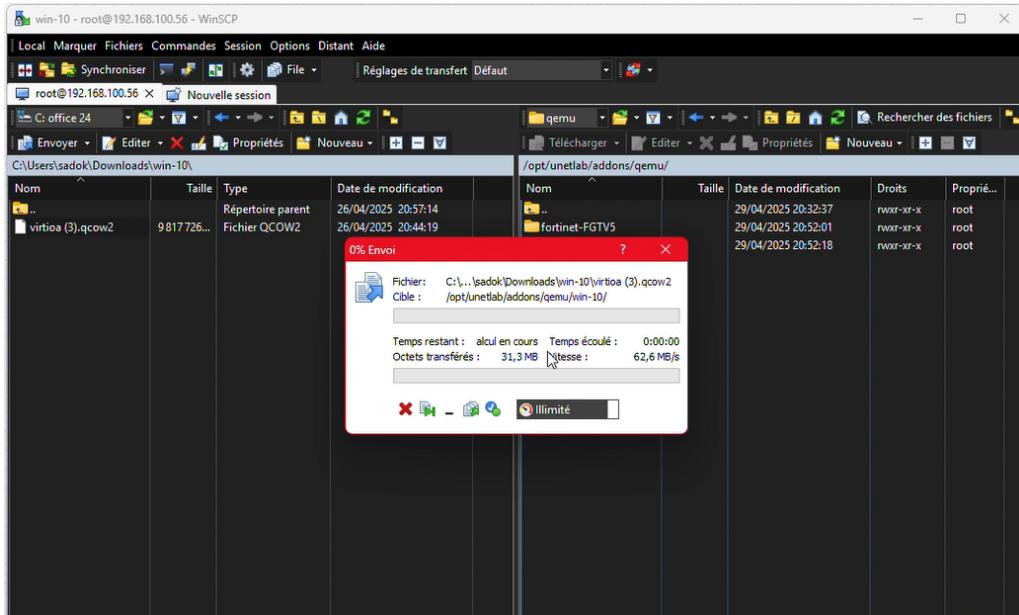


FIGURE 4.18 – Transfert de l'image Windows 10

Une fois le transfert terminé, nous vérifions que le fichier a bien été copié dans le bon répertoire.

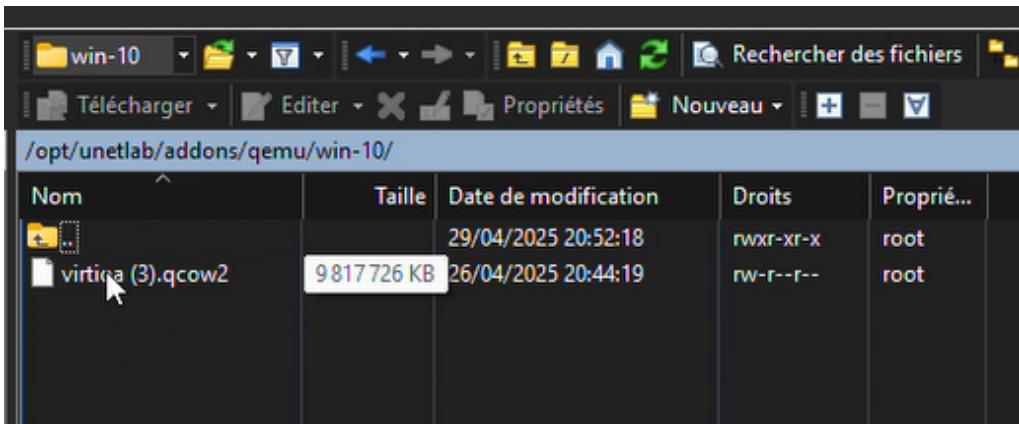


FIGURE 4.19 – Vérification du dossier win-10

Finalement, nous confirmons que le fichier `virtioa (3).qcow2` est bien placé, ce qui rend la VM prête à être utilisée dans EVE-NG.

Ajout d'une image OS Cisco IOL (Switch)

Pour enrichir notre topologie avec des équipements de commutation Cisco, nous avons intégré une image IOS de type IOL (IOS on Linux). EVE-NG permet la simulation de commutateurs et routeurs Cisco via des images BIN, à condition qu'elles soient correctement installées et autorisées via une licence.

Tout d'abord, nous avons utilisé **WinSCP** pour accéder au répertoire `/opt/unetlab/addons/iol/bin/` dans la VM EVE-NG, où sont stockés les fichiers BIN de Cisco.

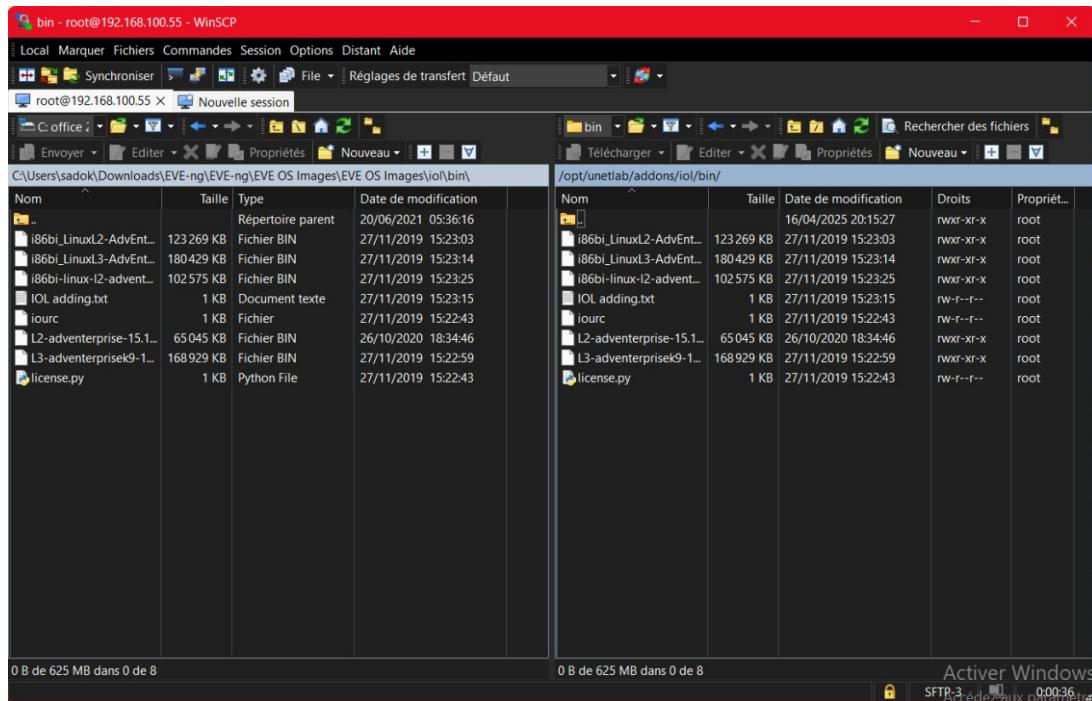


FIGURE 4.20 – Transfert des images IOL Cisco dans le dossier /opt/unetlab/addons/iol/bin/

Plusieurs fichiers BIN ont été ajoutés, tels que :

- i86bi_LinuxL2-AdvEnterpriseK9 : pour la simulation de switches L2
- i86bi_LinuxL3-AdvEnterpriseK9 : pour la simulation de routeurs L3
- Le fichier iourc : contenant la clé de licence indispensable au fonctionnement de ces images

Nous avons aussi vérifié que la structure des répertoires est bien conforme, notamment la présence des dossiers requis à l'intérieur de /opt/unetlab/addons/ :

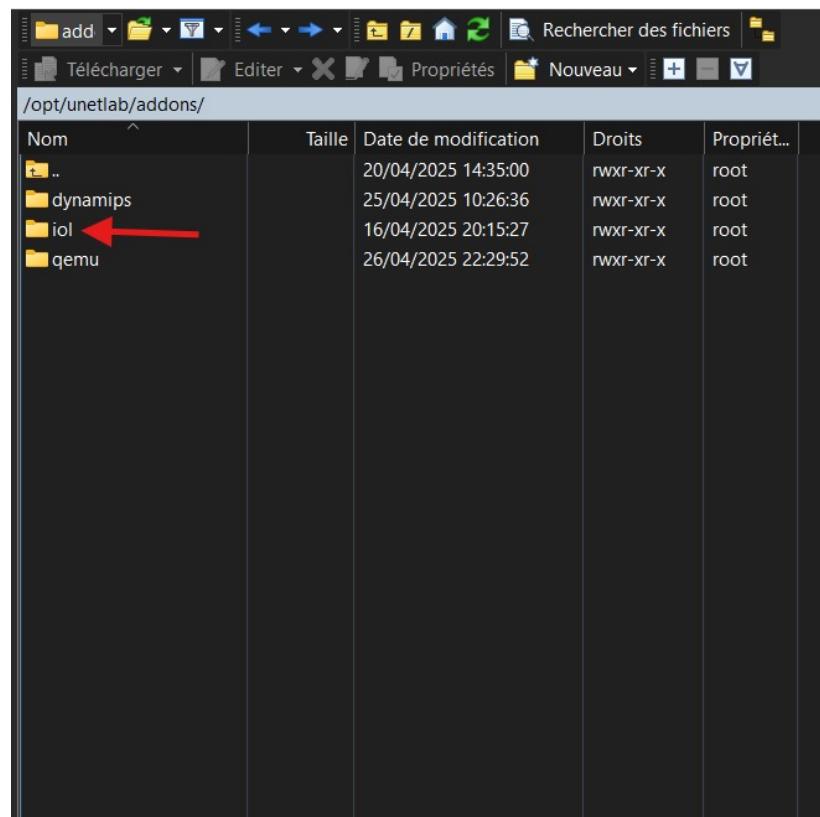
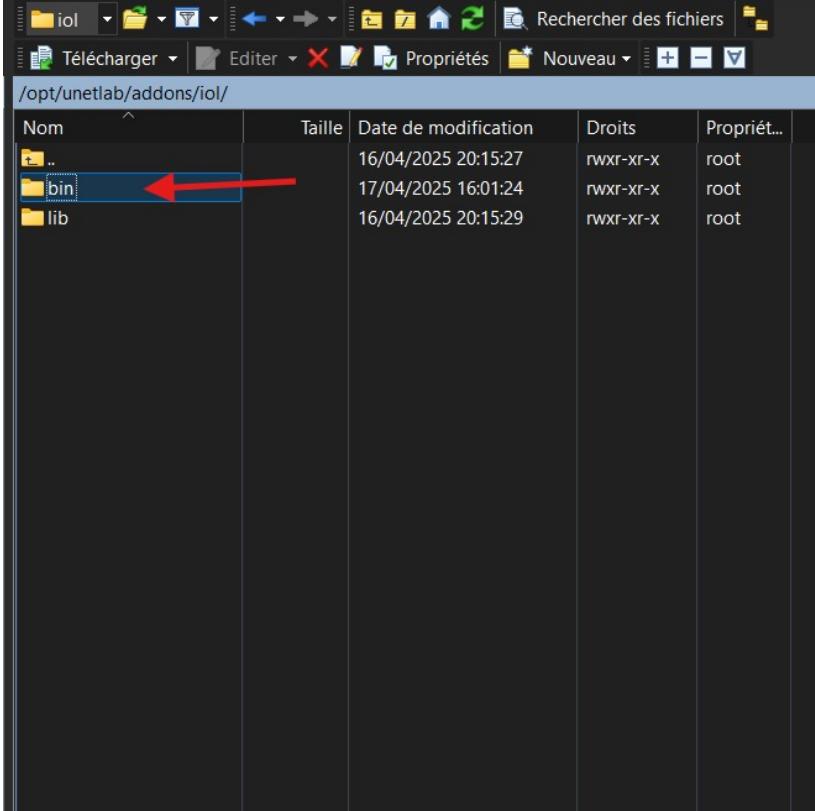


FIGURE 4.21 – Répertoires d'extensions dans EVE-NG

Et à l'intérieur de `/opt/unetlab addons/iol/`, la structure est bien respectée avec les répertoires `bin` et `lib` :



Nom	Taille	Date de modification	Droits	Propriét...
..		16/04/2025 20:15:27	rwxr-xr-x	root
bin		17/04/2025 16:01:24	rwxr-xr-x	root
lib		16/04/2025 20:15:29	rwxr-xr-x	root

FIGURE 4.22 – Contenu du répertoire `/opt/unetlab addons/iol/`

Après avoir transféré les fichiers nécessaires, il est crucial d'attribuer les bonnes permissions pour permettre à EVE-NG de les utiliser correctement. Cela se fait avec la commande suivante sur la machine EVE-NG :

```
/opt/unetlab/wrappers/unl_wrapper -a fixpermissions
```

Cette commande applique les permissions adéquates à tous les fichiers et dossiers des images, garantissant ainsi leur fonctionnement lors de la création de laboratoires sur la plateforme.

Installation du client Windows

Dans le cadre de notre projet, nous avons utilisé une topologie réseau simulée sur EVE-NG. Pour interagir correctement avec les équipements virtuels, notamment :

- accéder à l'interface en ligne de commande (CLI) du pare-feu via *PuTTY*,
- et ouvrir un poste client *Windows* à distance via *UltraVNC*,

il est nécessaire d'installer un ensemble d'outils spécifiques sur notre machine hôte Windows. Ces outils sont regroupés dans ce que l'on appelle le **Windows Integration Pack** fourni par EVE-NG.

Ce pack est indispensable pour assurer une interaction fluide entre la machine hôte Windows et les équipements simulés dans EVE-NG.

Windows Client Side

Below one can find a Windows client side pack that will install everything necessary for running telnet, vnc, wireshark, rdp applications when working on/building labs on EVE-NG it includes:

- Wireshark 3.0.6.0 installation
- UltraVNC 1.2.3.1 installation
- putty 0.73 (used as default telnet client)
- plink 0.73 (for wireshark)
- all necessary wrappers
- It will modify windows registry files for proper work
- It will save all the files on the local PC if one would like to modify for example, using SecureCRT instead of default Putty.
- Windows 8 and 10 reg files to support tabbed SecureCRT
- Auto detection of Windows version (7, 8, 10) (x64 only supported)

Download links:

- [Windows Integration pack](#)
- [Windows Integration pack mirror](#)

FIGURE 4.23 – Téléchargement du Windows Integration Pack

Après téléchargement, il suffit de lancer l'installateur et de suivre les instructions affichées à l'écran. Cela va automatiquement configurer l'environnement pour permettre une interaction fluide avec les noeuds du lab EVE-NG.

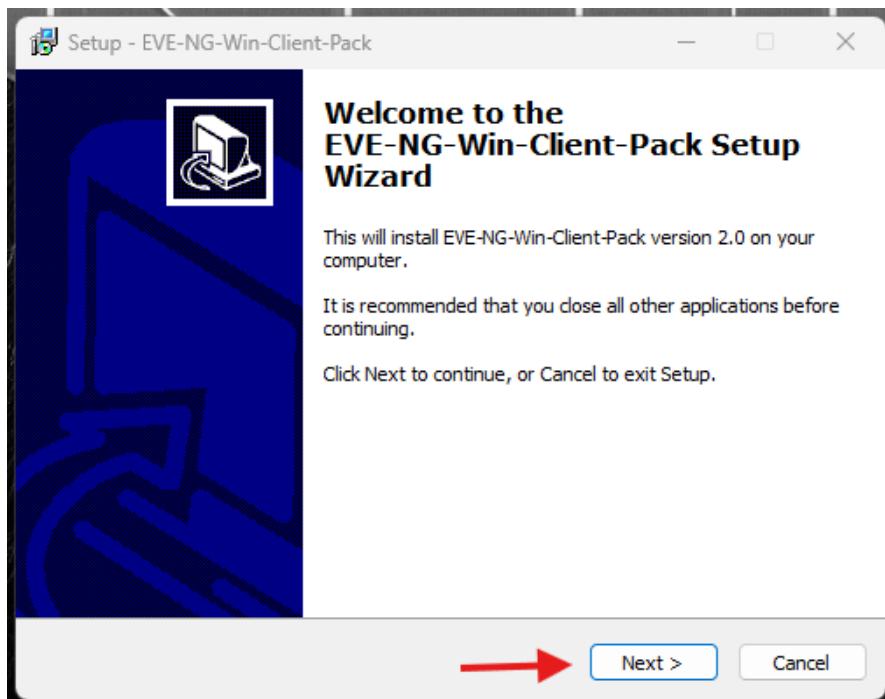


FIGURE 4.24 – Assistant d'installation du Windows Integration Pack

Ce pack facilite considérablement la préparation de l'environnement côté client en installant automatiquement l'ensemble des outils nécessaires pour interagir avec les équipements virtualisés dans EVE-NG. Il permet d'éviter les configurations manuelles souvent sources d'erreurs, et garantit ainsi une compatibilité immédiate avec la plateforme. Le pack comprend notamment :

- **Wireshark 3.0.6.0** : outil d'analyse de trafic réseau permettant la capture et l'inspection des paquets échangés entre les noeuds,
- **UltraVNC 1.2.3.1** : utilisé pour accéder aux interfaces graphiques (GUI) de certains systèmes comme Windows ou FortiGate,
- **PuTTY et Plink 0.73** : clients SSH et Telnet pour accéder aux consoles des équipements en ligne de commande,
- des **fichiers de registre Windows** : qui assurent l'association correcte des protocoles (telnet, vnc, ssh...) avec les logiciels installés,

- o le support pour **SecureCRT** (optionnel) : si cet outil est déjà présent sur la machine, il peut être utilisé comme client par défaut,
- o une **détection automatique de la version de Windows** (7, 8, 10 – uniquement en 64 bits), afin d'adapter l'installation selon l'OS.

L'exemple ci-dessous illustre l'installation de *UltraVNC*, un outil permettant d'accéder graphiquement aux interfaces des équipements simulés.

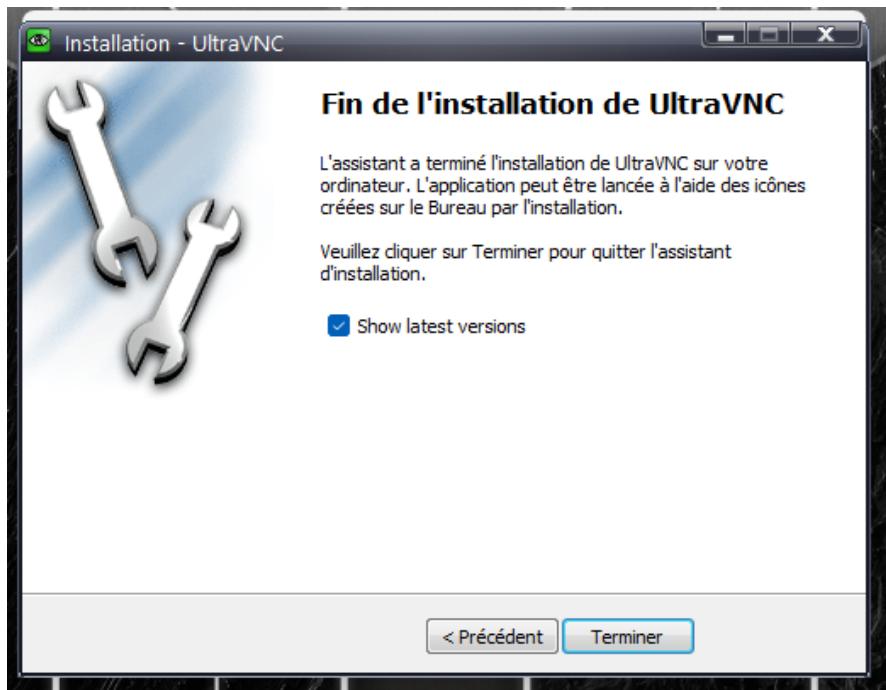


FIGURE 4.25 – Installation de UltraVNC

Le même principe s'applique à l'installation de *Wireshark*, un outil indispensable pour la capture et l'analyse des paquets réseau. Il est automatiquement configuré pour être utilisé avec EVE-NG, ce qui évite les réglages manuels souvent nécessaires dans une installation classique.

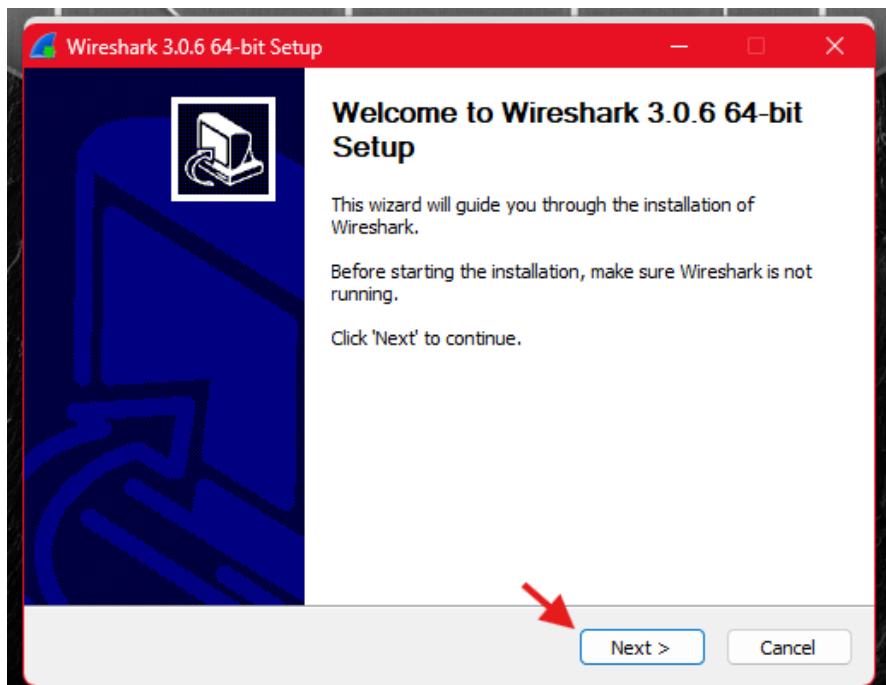


FIGURE 4.26 – Installation de Wireshark

Une fois l'ensemble des composants installés, l'assistant termine par un écran de confirmation indiquant que l'environnement est prêt à l'emploi. Cette finalisation garantit que tous les outils nécessaires sont bien intégrés et fonctionnels.

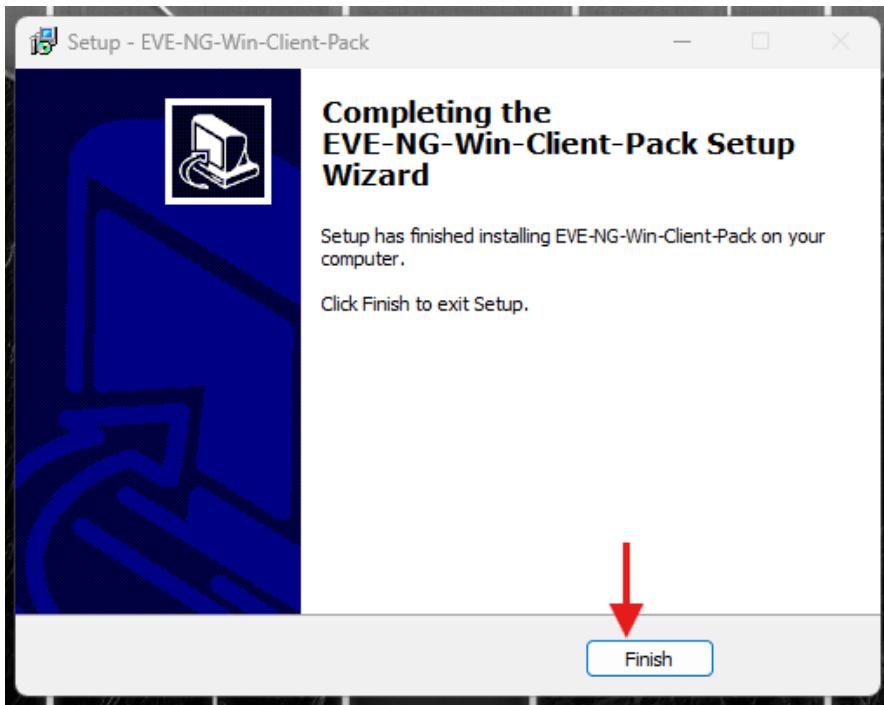


FIGURE 4.27 – Fin de l'installation du Windows Integration Pack

4.1.3 Création de topologie

La création de la topologie consiste à concevoir l'architecture réseau virtuelle en ajoutant et configurant les différents équipements nécessaires à travers l'interface web d'EVE-NG.

Ajout du OS Fortinet FortiGate

Avant d'ajouter un nœud Fortinet, il est nécessaire d'accéder à l'interface de la plateforme EVE-NG et de créer un nouveau lab.

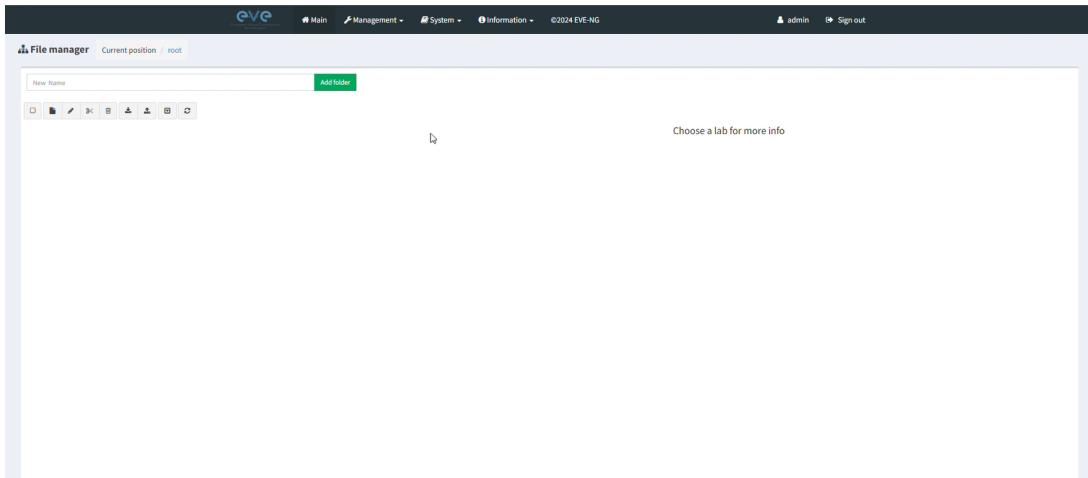


FIGURE 4.28 – Interface création lab

Nous avons créé un lab nommé **eve**, dans lequel nous allons ajouter et configurer le pare-feu Fortinet FortiGate.

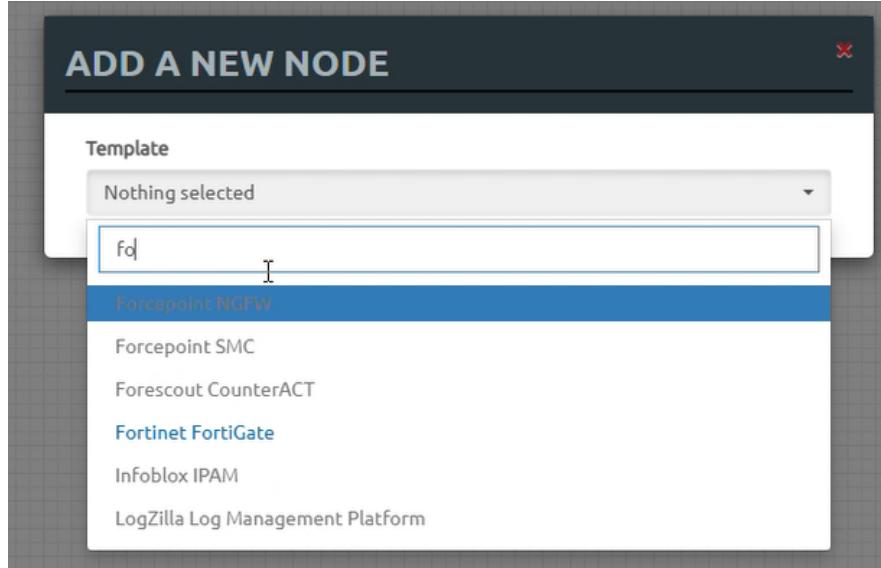


FIGURE 4.29 – L’ajout du Fortinet FortiGate

Dans un premier temps, nous créons un nouveau dossier dans le répertoire `/opt/unetlab/addons/qemu/` afin d’y ajouter l’image du pare-feu Fortinet.

Ensuite, nous ajoutons un nouveau noeud au sein de notre topologie réseau virtuelle via l’interface graphique. Lors de l’ajout du noeud, nous sélectionnons le modèle *Fortinet FortiGate* depuis la liste proposée.

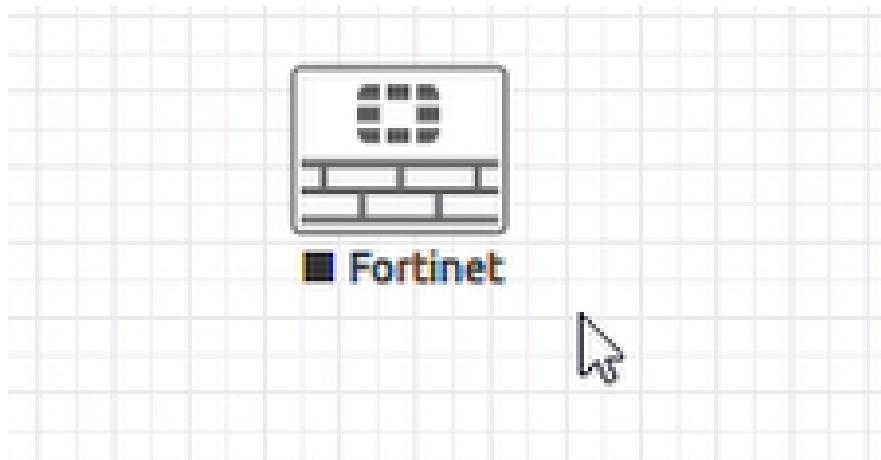


FIGURE 4.30 – FW FortiGate

Après l’ajout, nous constatons que le pare-feu Fortinet démarre brièvement puis s’éteint automatiquement. Cela peut être dû à une configuration incorrecte des paramètres QEMU ou à une erreur dans la structure du dossier contenant l’image. En effet, EVE-NG requiert que le fichier image soit nommé précisément `hda.qcow2` pour qu’il soit correctement monté au démarrage du noeud. Initialement nommé `fortios.qcow2`, le fichier a donc été renommé en `hda.qcow2` conformément aux exigences de la plateforme.

4.2 Configuration du réseau sécurisé

4.2.1 paramétrage réseau

Après avoir intégré les machines virtuelles dans EVE-NG, nous passons à la configuration de la topologie réseau. Cette étape est essentielle pour assurer la communication entre les différents

équipements : pare-feu FortiGate, VM Windows 10, et VPCS.

Tout d'abord, nous préparons la plateforme EVE-NG en lançant l'interface de création d'un nouveau projet. Il s'agit ici de la première étape dans la construction de notre maquette virtuelle.

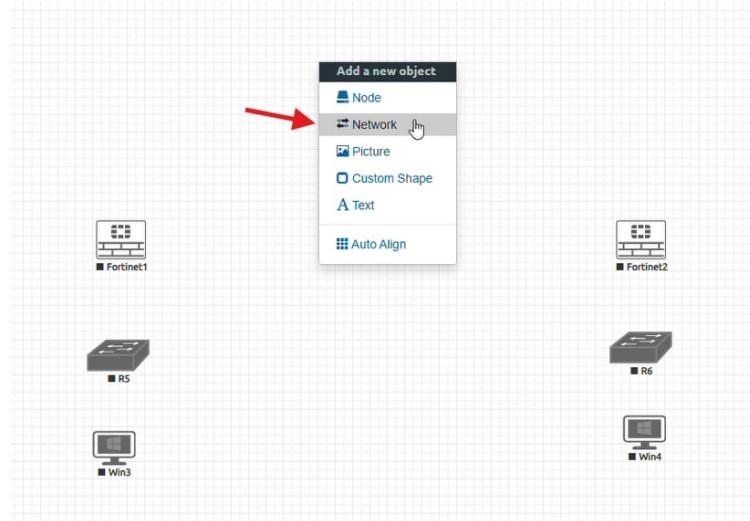


FIGURE 4.31 – Préparation de la topologie réseau sur EVE-NG

Ensuite, nous ajoutons deux interfaces réseau : l'une dédiée au réseau interne (LAN) et l'autre à l'accès externe (WAN). Cette séparation permet de simuler un vrai environnement sécurisé.

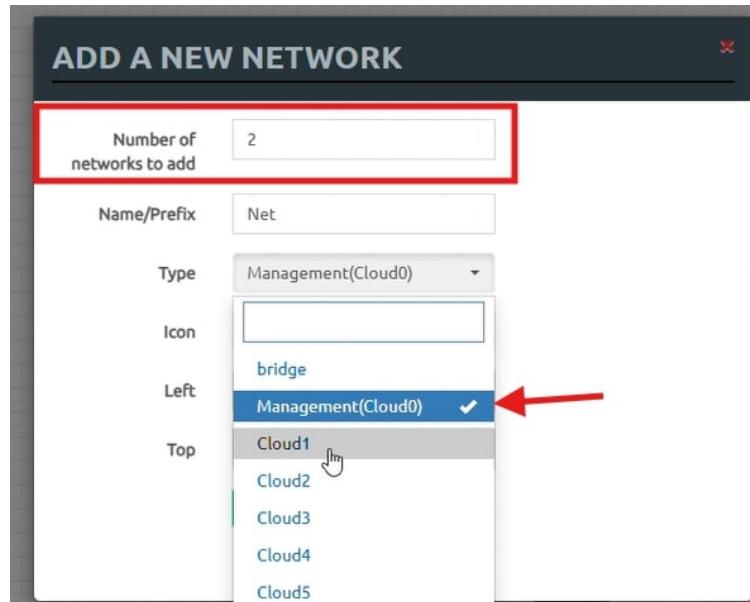


FIGURE 4.32 – Ajout de deux interfaces réseau (LAN/WAN)

Une fois la topologie en place, nous procédons au démarrage des noeuds. Pour ce faire, un clic droit sur les équipements permet de les lancer, et un message contextuel apparaît pour proposer une connexion via Putty.

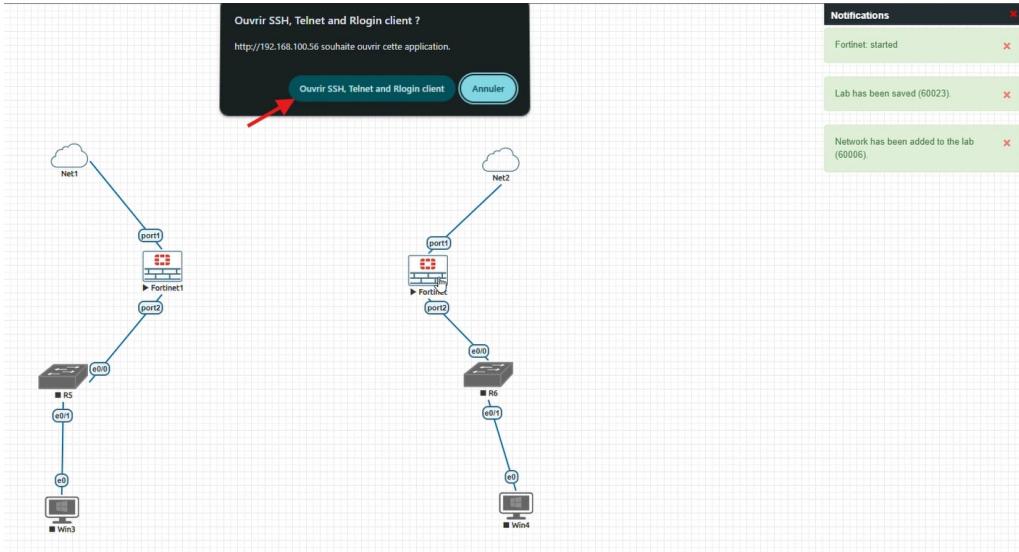


FIGURE 4.33 – Lancement des équipements et ouverture du terminal Putty

Une fois la console ouverte via Putty, nous accédons à l’interface de configuration CLI du pare-feu FortiGate. Par défaut, l’authentification se fait avec l’utilisateur **admin** sans mot de passe.

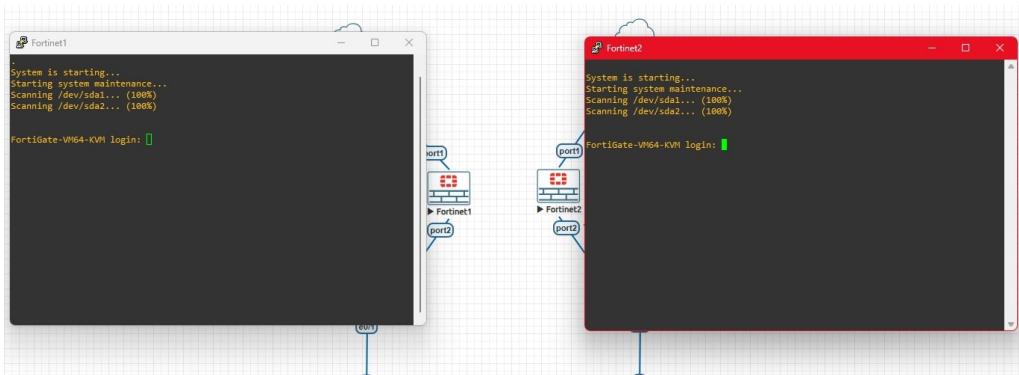


FIGURE 4.34 – Connexion à l’interface CLI de FortiGate avec l’utilisateur **admin**

Nous commençons ensuite par la configuration des interfaces réseau sur chaque pare-feu. Chaque interface est affectée à un sous-réseau spécifique et configurée avec une adresse IP, ainsi que des autorisations d’accès (ping, http) pour permettre la gestion à distance.

```

FortiGate-VM64-KVM # config system interface
FortiGate-VM64-KVM (interface) # edit port1
FortiGate-VM64-KVM (port1) # set ip 192.168.100.150 255.255.255.0
FortiGate-VM64-KVM (port1) # set allowaccess ping http
FortiGate-VM64-KVM (port1) # next
FortiGate-VM64-KVM (interface) # edit port2
FortiGate-VM64-KVM (port2) # set ip 192.168.1.1 255.255.255.0
FortiGate-VM64-KVM (port2) # set allowaccess ping http
FortiGate-VM64-KVM (port2) # next
FortiGate-VM64-KVM (interface) # end
FortiGate-VM64-KVM # config router static
FortiGate-VM64-KVM (static) # edit 1
new entry '1' added
FortiGate-VM64-KVM (1) #      set gateway 192.168.100.1
FortiGate-VM64-KVM (1) #      set device "port1"
FortiGate-VM64-KVM (1) #      next
FortiGate-VM64-KVM (static) # end
FortiGate-VM64-KVM # Timeout
exit

FortiGate-VM64-KVM login: [REDACTED] 192.168.1.10
```



```

FortiGate-VM64-KVM (interface) # edit port1
FortiGate-VM64-KVM (port1) # set ip 192.168.100.24
FortiGate-VM64-KVM (port1) # set allowaccess ping http
FortiGate-VM64-KVM (port1) # next
FortiGate-VM64-KVM (interface) # edit port2
FortiGate-VM64-KVM (port2) # set ip 192.168.2.1/24
FortiGate-VM64-KVM (port2) # set allowaccess ping http
FortiGate-VM64-KVM (port2) # next
FortiGate-VM64-KVM (interface) # config router static
FortiGate-VM64-KVM (static) #      edit 1
new entry '1' added
FortiGate-VM64-KVM (1) #      set gateway 192.168.100.1
FortiGate-VM64-KVM (1) #      set device "port1"
FortiGate-VM64-KVM (1) #      next
FortiGate-VM64-KVM (static) # end
FortiGate-VM64-KVM #
FortiGate-VM64-KVM # Timeout
exit

FortiGate-VM64-KVM login: [REDACTED]
```

FIGURE 4.35 – Configuration des interfaces et de la route statique sur FortiGate

Une fois cette configuration effectuée, il est essentiel de sauvegarder la configuration système afin d'éviter toute perte en cas de redémarrage.

```

FortiGate-VM64-KVM (1) # FortiGate-VM64-KVM (1) #
Unknown action 0

FortiGate-VM64-KVM (1) #      set gateway 192.168.100.1
FortiGate-VM64-KVM (1) #      set device "port1"
FortiGate-VM64-KVM (1) #      next
FortiGate-VM64-KVM (static) # end
FortiGate-VM64-KVM # Timeout
exit

FortiGate-VM64-KVM login: admin
Password: Welcome !
FortiGate-VM64-KVM # execute backup config flash
Please wait...
Config backed up to flash disk done.

FortiGate-VM64-KVM #
```



```

FortiGate-VM64-KVM (1) # set gateway 192.168.100.1
Fortigate-VM64-KVM (1) # set device "port1"
token line: Unmatched double quote.

FortiGate-VM64-KVM (1) # set device "port1"
FortiGate-VM64-KVM (1) # next
FortiGate-VM64-KVM (static) # end
FortiGate-VM64-KVM # Timeout
exit

FortiGate-VM64-KVM login: admin
Password: Welcome !
FortiGate-VM64-KVM # execute backup config flash
Please wait...
Config backed up to flash disk done.

FortiGate-VM64-KVM #
```

FIGURE 4.36 – Sauvegarde de la configuration via la CLI FortiGate

Par la suite, l'administration du pare-feu peut se faire de manière plus conviviale via l'interface web. Pour cela, nous accédons à l'interface graphique en saisissant l'adresse IP du pare-feu dans un navigateur.

Une fois authentifié, l'administrateur accède au tableau de bord principal affichant les informations système du pare-feu ainsi que les menus de configuration.

FortiGate VM64-KVM

System Information

- Hostname: FortiGate-VM64-KVM
- Serial Number: FGVMEVON84-ARL7A
- Firmware: v5.6.2 build1486 (GA)
- Mode: NAT (Flow-based)
- System Time: 2025/05/19 04:43:44
- Uptime: 00:00:38:03
- WAN IP: [REDACTED]

Virtual Machine

Evaluation Used	1 / 15 Days
Allocated vCPUs	1 / 1
Allocated RAM	995 MIB / 1 GB

FortiCloud

Status: Not Activated

FortiGate VM64-KVM

System Information

- Hostname: FortiGate-VM64-KVM
- Serial Number: FGVMEVK-WADAW887
- Firmware: v5.6.2 build1486 (GA)
- Mode: NAT (Flow-based)
- System Time: 2025/05/19 04:43:44
- Uptime: 00:00:38:01
- WAN IP: [REDACTED]

Virtual Machine

Evaluation Used	1 / 15 Days
Allocated vCPUs	1 / 1
Allocated RAM	995 MIB / 1 GB

FortiCloud

Status: Not Activated

FIGURE 4.37 – Dashboard de l'interface graphique web du pare-feu FortiGate

4.2.2 Configuration du VPN IPsec

Une fois la connectivité de base établie entre les deux pare-feu, nous avons procédé à la mise en place d'un tunnel VPN IPsec site-à-site afin de sécuriser les communications entre les deux réseaux internes.

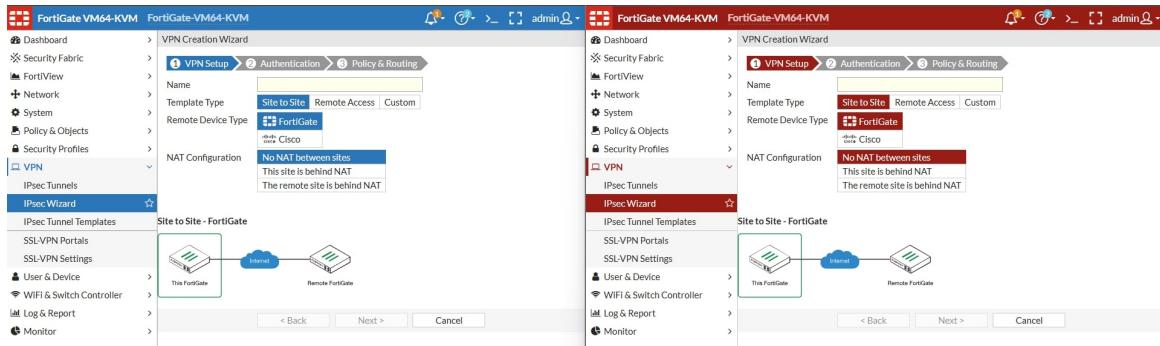


FIGURE 4.38 – Configuration d'un VPN IPsec site-à-site

Le processus débute par la configuration de la phase d'authentification. Chaque pare-feu spécifie l'adresse IP du pair et une clé prépartagée identique.

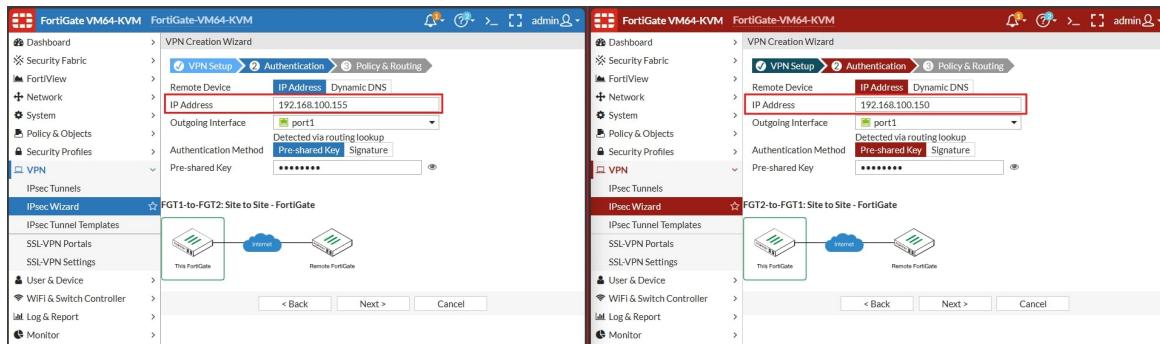


FIGURE 4.39 – Phase d'authentification du VPN IPsec

Ensuite, les réseaux locaux à inclure dans le tunnel sont définis. Ces sous-réseaux seront ceux qui bénéficieront de la connectivité via le VPN.

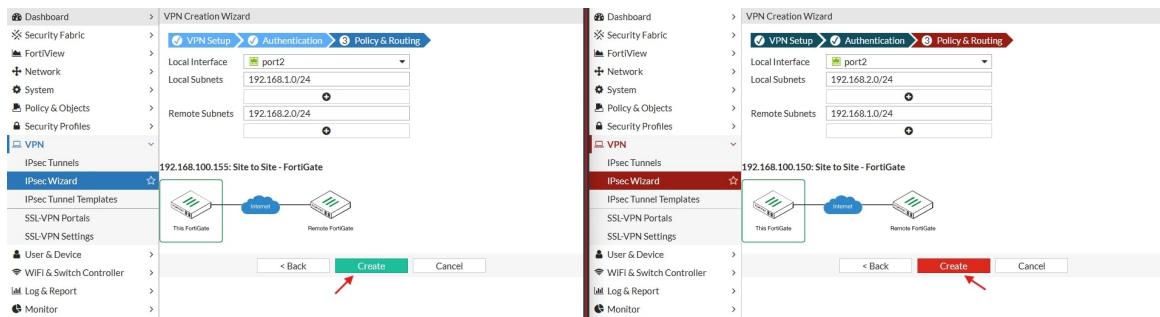


FIGURE 4.40 – Définition des réseaux internes dans le VPN

Un aperçu de l'interface du tunnel VPN permet de visualiser les détails de configuration.

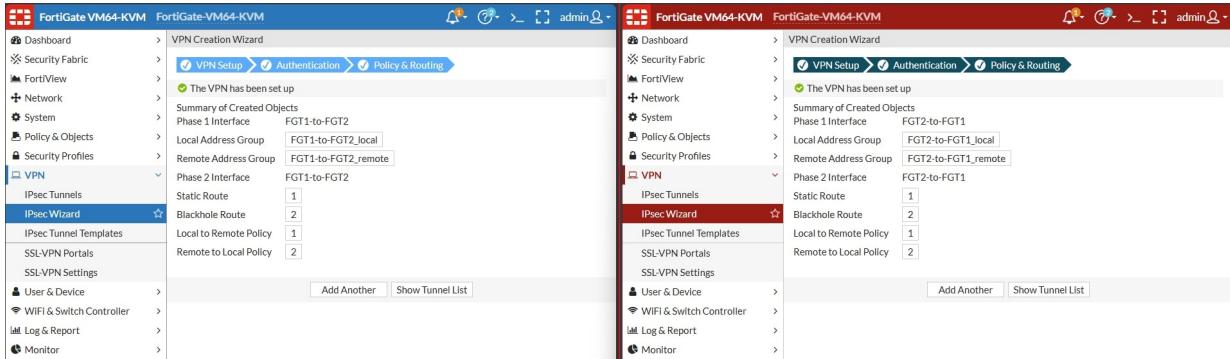


FIGURE 4.41 – Aperçu du tunnel IPsec en cours

Tant que la configuration n'est pas complète sur le second FortiGate, le tunnel reste inactif.



FIGURE 4.42 – État du tunnel VPN : Down

Des routes statiques sont ensuite ajoutées pour permettre le routage du trafic destiné au réseau distant via le tunnel.

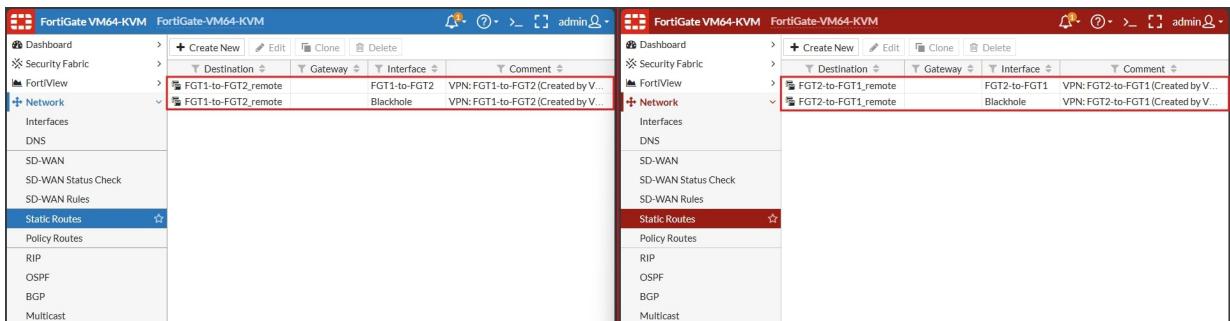


FIGURE 4.43 – Ajout de routes statiques

Afin de permettre la circulation du trafic entre les deux réseaux locaux via le tunnel VPN, il est nécessaire de définir des politiques de sécurité (ou *firewall policies*) sur chaque pare-feu. Ces règles autorisent explicitement le passage du trafic entre les interfaces VPN et les interfaces LAN.

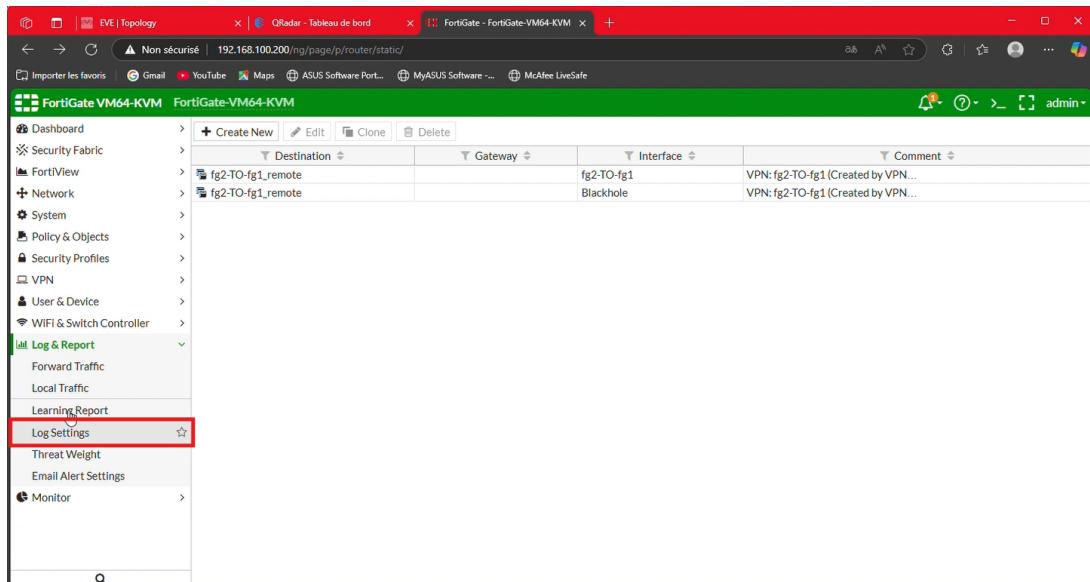


FIGURE 4.44 – Configuration du routage de FortiGate

On s’assure qu’une route statique vers l’IP de QRadar est présente, garantissant la connectivité réseau entre les deux équipements.

FIGURE 4.45 – Configuration des politiques de sécurité sur le FortiGate 1

Sur le FortiGate 1, deux règles ont été créées : l’une permettant au trafic issu du réseau local de transiter vers le réseau distant via le tunnel VPN, et l’autre dans le sens inverse pour assurer la communication bidirectionnelle.

FIGURE 4.46 – Configuration des politiques de sécurité sur le FortiGate 2

De manière similaire, deux règles équivalentes ont été mises en place sur le FortiGate 2. Ces configurations garantissent que le trafic autorisé peut circuler librement dans les deux sens à travers le tunnel IPsec, conformément à la stratégie de sécurité définie.

4.2.3 Tests réseau

Une fois la configuration du second FortiGate terminée, un test de connectivité est réalisé à l’aide de la commande `ping` entre les deux pare-feu.



FIGURE 4.47 – Test de connectivité entre les FortiGate

Les mêmes étapes de configuration sont reproduites sur le second pare-feu.

FIGURE 4.48 – RéPLICATION de la configuration sur le second FortiGate

Lorsque le tunnel est actif, l'interface affiche l'état up, indiquant que la connexion est établie.

FIGURE 4.49 – Tunnel VPN actif

Pour tester le fonctionnement du réseau simulé, les machines Windows sont démarrées à distance via UltraVNC.

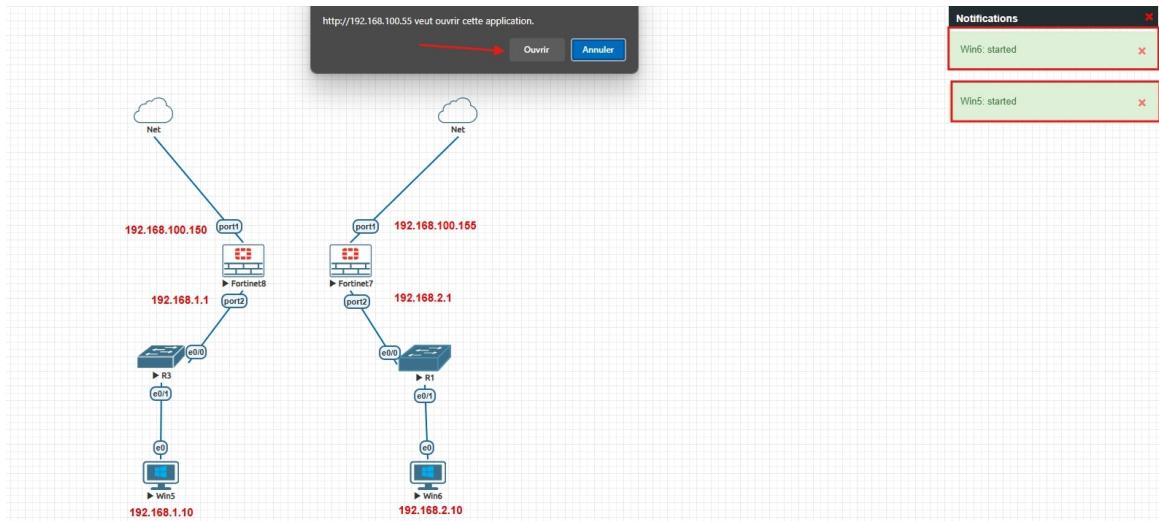


FIGURE 4.50 – Lancement des machines Windows via UltraVNC

Les deux VM démarrent correctement, permettant d'effectuer des tests réseau. Chaque machine reçoit une configuration IP statique manuellement.

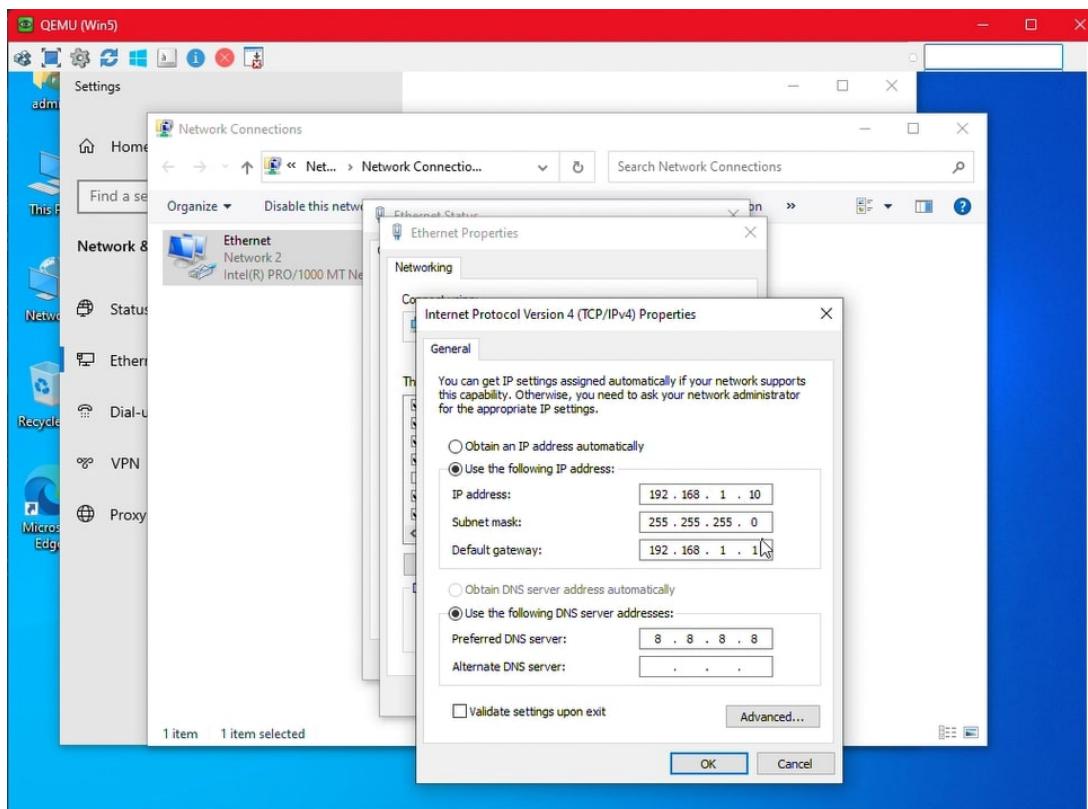


FIGURE 4.51 – Configuration réseau des PC

Un poste client supplémentaire est ajouté pour enrichir le scénario.

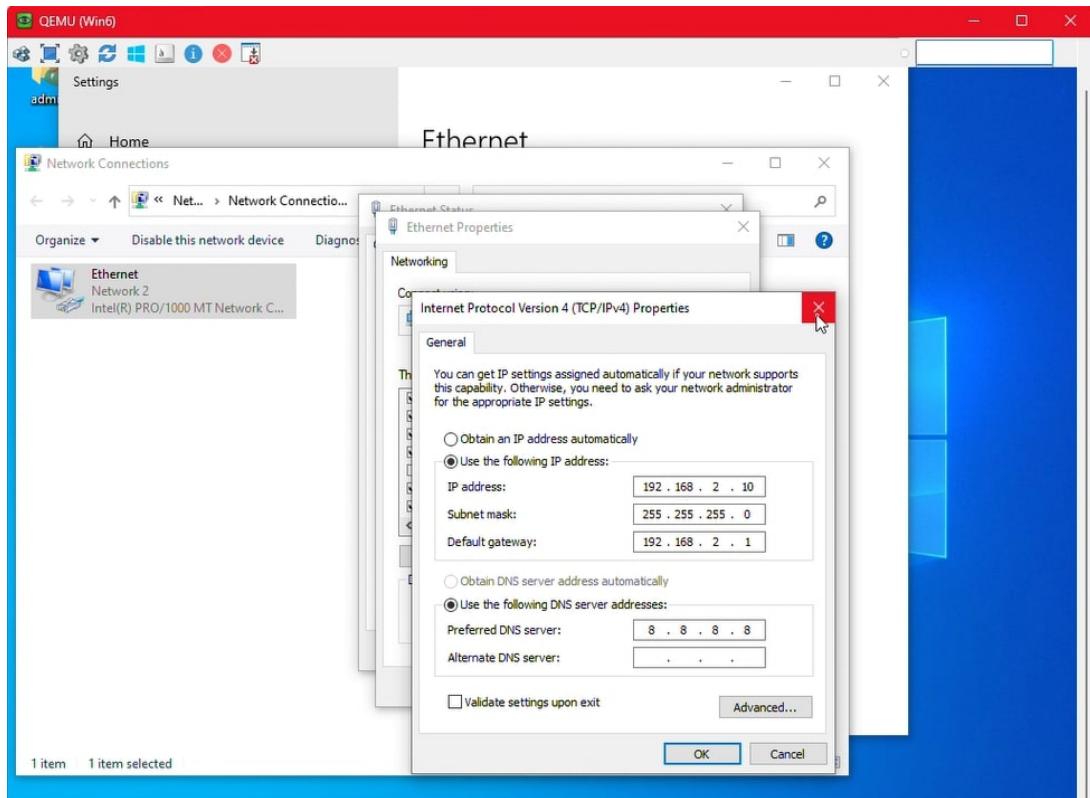


FIGURE 4.52 – Ajout d'un PC supplémentaire à la topologie

Chaque poste est vérifié pour s'assurer que le pare-feu Windows ne bloque pas les communications.

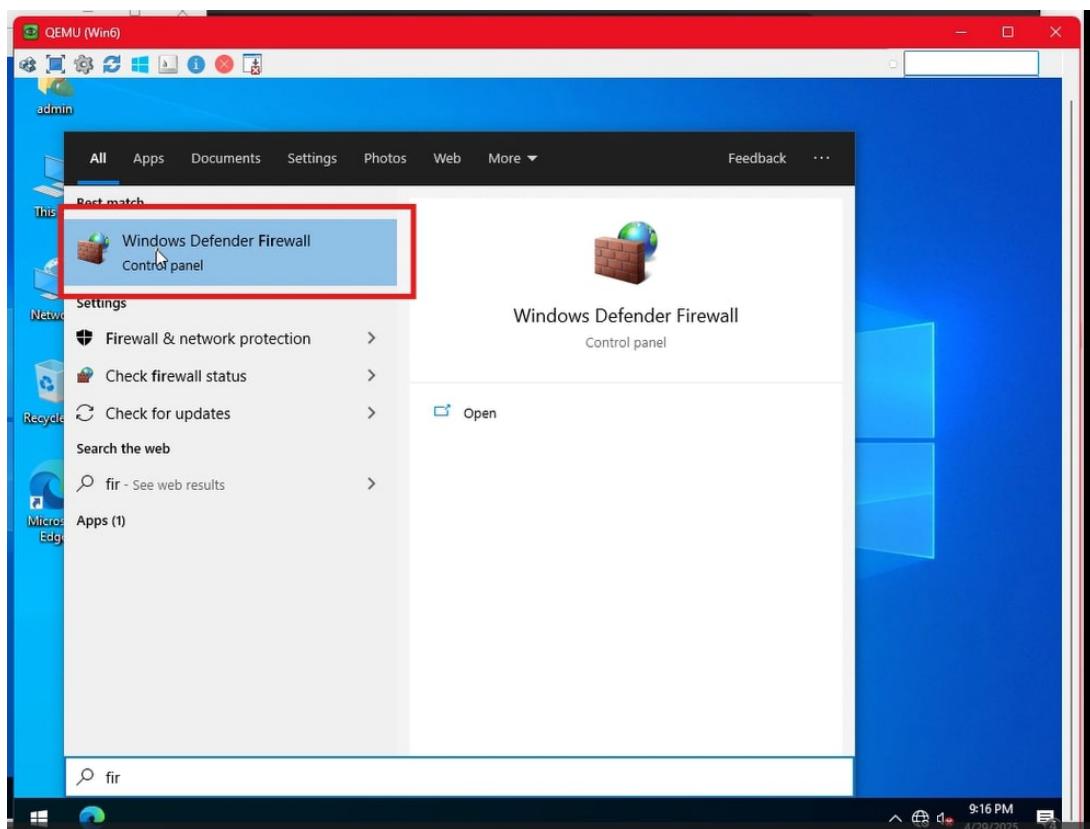


FIGURE 4.53 – Vérification du pare-feu Windows

Chaque PC dispose maintenant d'une adresse IP statique unique.

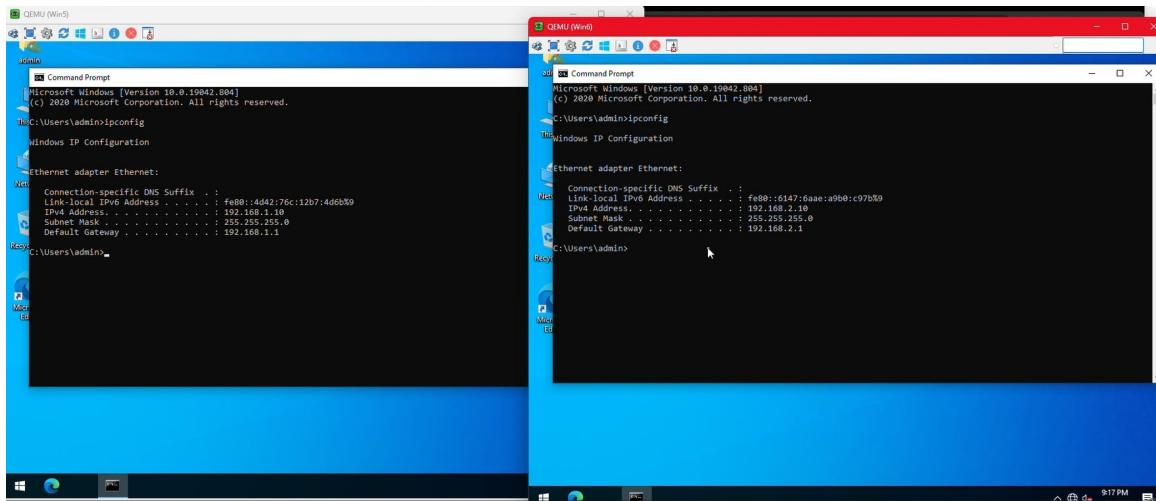


FIGURE 4.54 – Attribution des adresses IP des postes

Pour simuler un incident de sécurité, une tentative de connexion échouée est lancée.

```
Fortinet8

FortiGate-VM64-KVM login: azeazeaz
Password:
Login incorrect

FortiGate-VM64-KVM login: pp
Password:
Login incorrect

FortiGate-VM64-KVM login: mmm
Password:
Login incorrect

FortiGate-VM64-KVM login: ooo
Password:
Login incorrect

FortiGate-VM64-KVM login: nbhy
Password:
Login incorrect

FortiGate-VM64-KVM login: aaa
Password:
Login incorrect

FortiGate-VM64-KVM login: wwdwd
Password:
Login incorrect

FortiGate-VM64-KVM login: bvbv
Password:
Login incorrect

FortiGate-VM64-KVM login: esfesfs
Password:
Login incorrect

FortiGate-VM64-KVM login: sefsefs
Password:
Login incorrect

FortiGate-VM64-KVM login:
```

FIGURE 4.55 – Test de tentative de connexion échouée

4.3 Mise en œuvre du SIEM : QRadar

4.3.1 Implémentation et paramétrage initial

IBM QRadar est une solution SIEM (Security Information and Event Management) puissante permettant de centraliser, analyser et corrélérer les événements de sécurité issus de multiples sources. Dans le cadre de notre projet, nous avons utilisé la l'image ISO version Community Edition de QRadar, disponible gratuitement sur le site officiel d'IBM.

Lors du téléchargement, un fichier .sha256 est également fourni avec l'image ISO. Celui-ci permet de vérifier l'intégrité de l'image téléchargée à l'aide d'une somme de contrôle. Cette vérification est essentielle pour s'assurer que le fichier n'a pas été corrompu pendant le téléchargement ou modifié de manière malveillante.

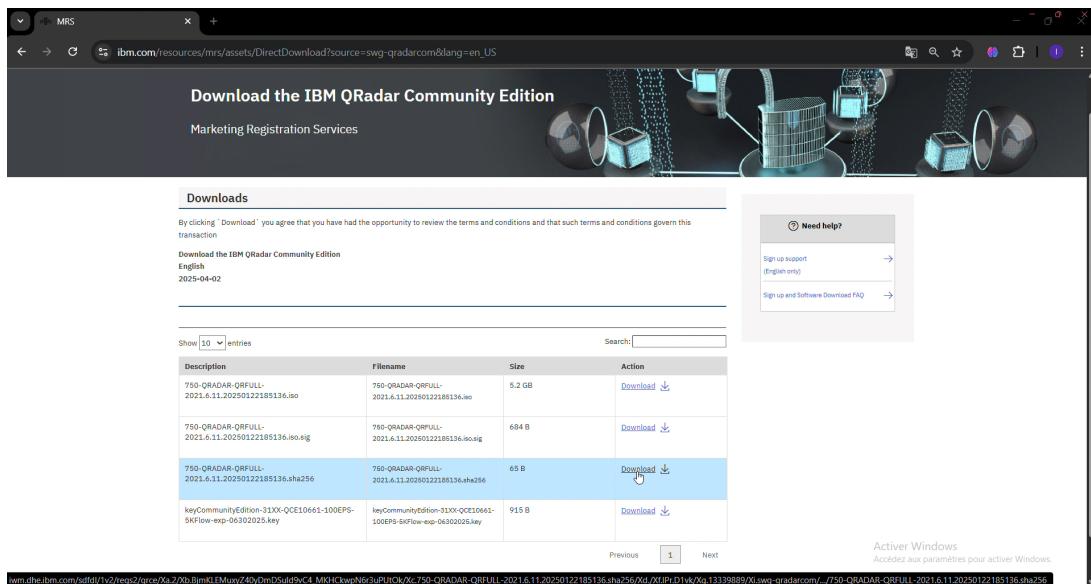


FIGURE 4.56 – Page de téléchargement de l'ISO

optimisation de la mémoire RAM avec RAMMap

QRadar étant une solution SIEM particulièrement gourmande en ressources, nous avons rencontré des limitations au niveau de la mémoire vive (RAM) sur notre infrastructure virtuelle EVE-NG. Afin de garantir la stabilité de la VM Windows 10 et des autres composants, nous avons utilisé l'outil RAMMap de Microsoft. Cet utilitaire, issu de la suite Sysinternals, offre une vue détaillée de l'utilisation de la mémoire physique. Il permet notamment d'identifier les zones utilisées par le système, les fichiers en cache ou les pilotes, et de libérer manuellement certaines zones comme la *Standby List*.

Cette intervention a permis d'optimiser l'utilisation de la RAM sans redémarrage, améliorant ainsi les performances globales nécessaires au bon fonctionnement de QRadar.

Avant son utilisation, nous avons observé l'état de la mémoire via le Gestionnaire des tâches :

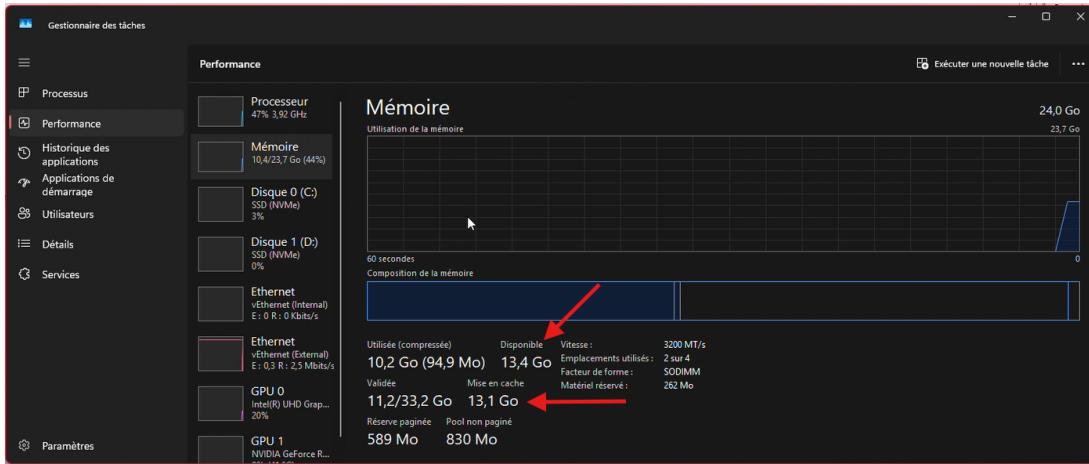


FIGURE 4.57 – État initial de la mémoire

Nous avons ensuite téléchargé RAMMap depuis le site officiel de Sysinternals :

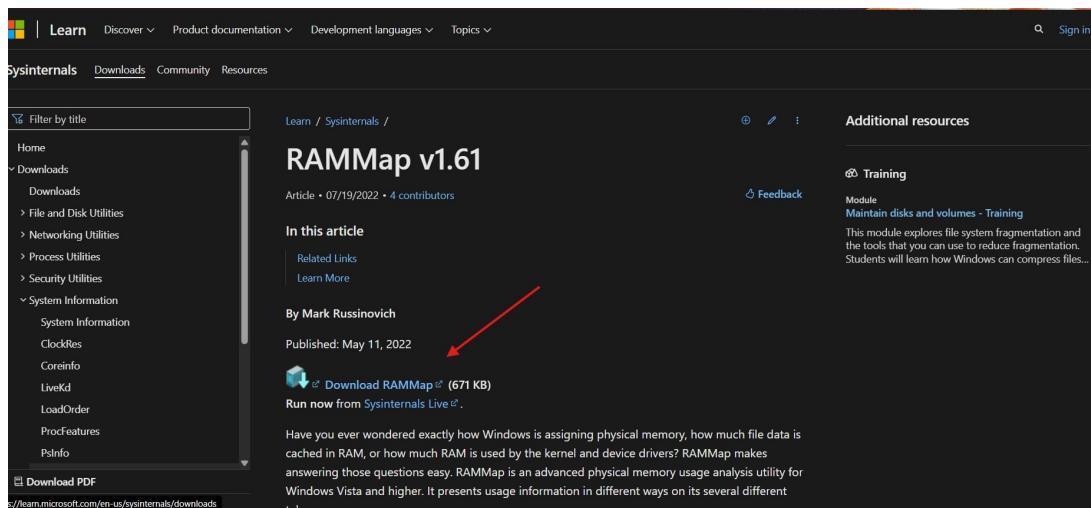


FIGURE 4.58 – Téléchargement de RAMMap depuis Sysinternals

Le fichier exécutable RAMMap64a.exe a été lancé directement, l'outil étant portable :

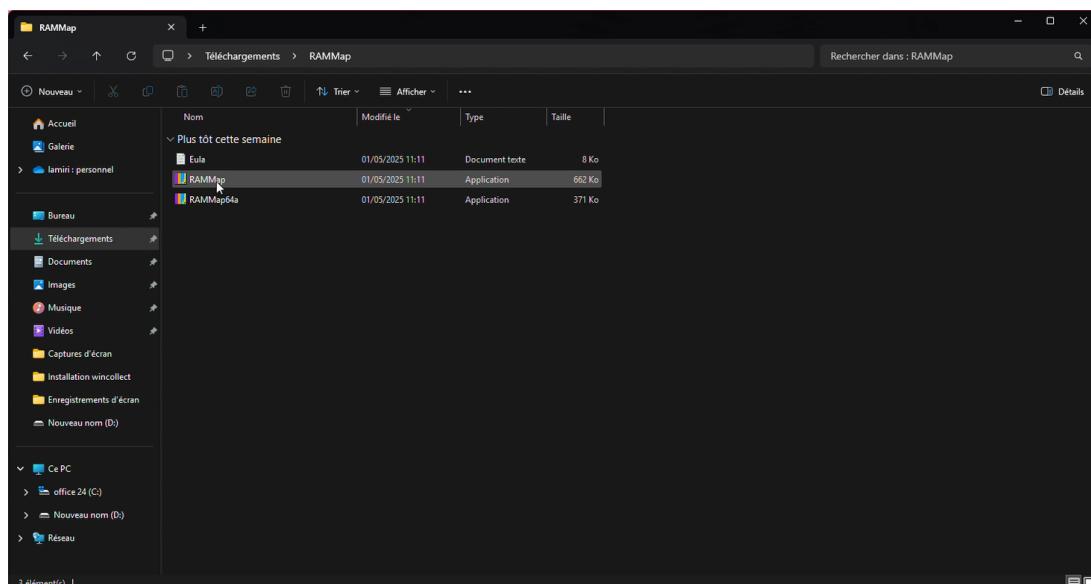


FIGURE 4.59 – Exécution de RAMMap

Dans son interface, RAMMap propose plusieurs onglets informatifs (Use Counts, Processes, Physical Pages, File Summary...) et des fonctionnalités puissantes comme le vidage manuel de la mémoire en veille ou modifiée :

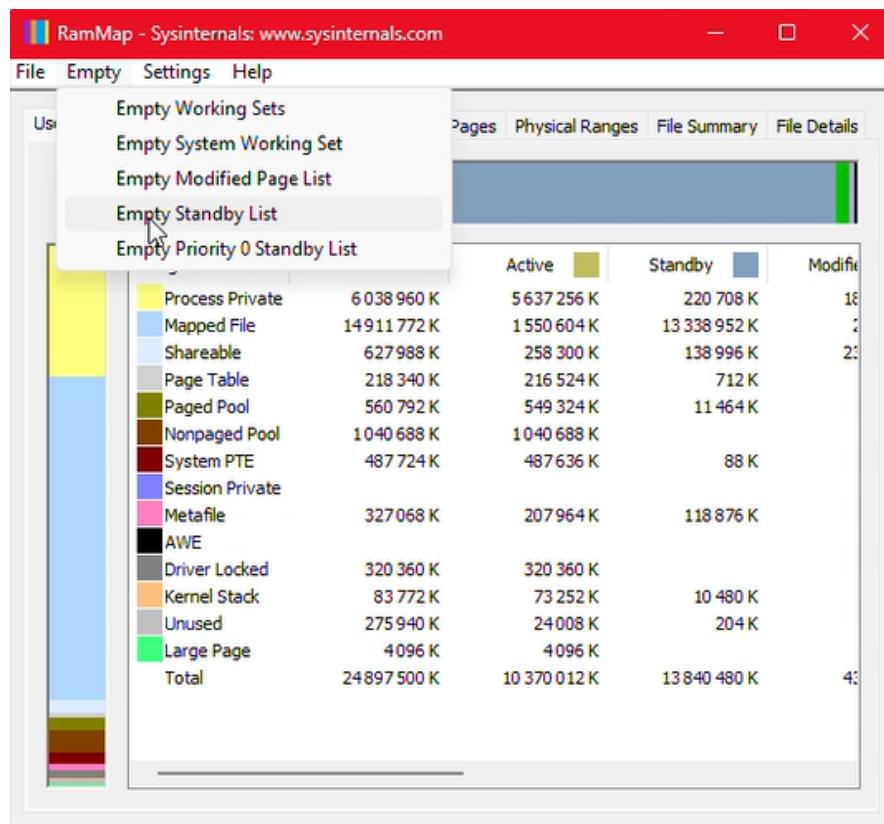


FIGURE 4.60 – Fonctions de nettoyage de la mémoire dans RAMMap

Cette opération nous a permis de récupérer une quantité significative de mémoire, optimisant ainsi les performances globales de la machine. Après l'utilisation de RAMMap, l'état de la mémoire était sensiblement amélioré :

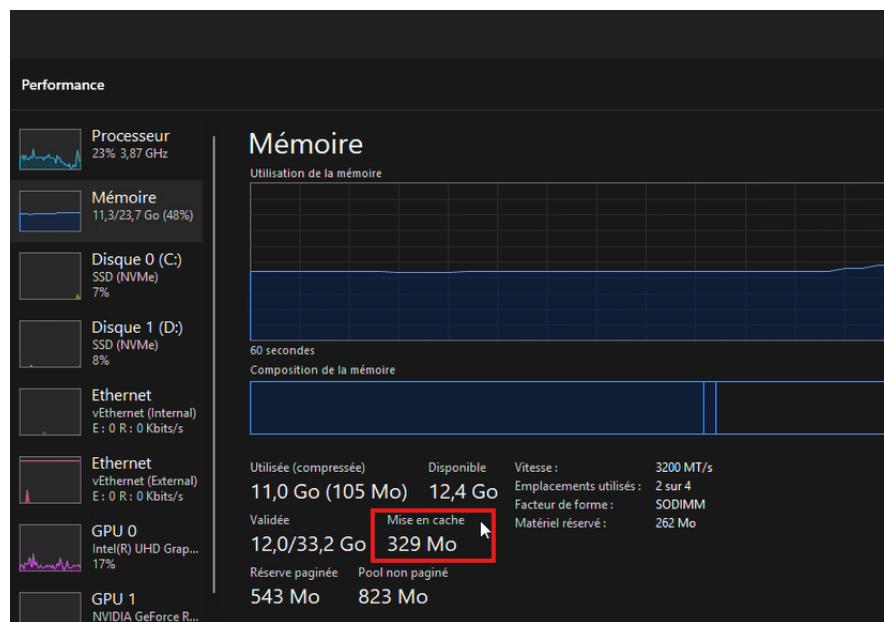


FIGURE 4.61 – État de la mémoire après l'utilisation de RAMMap

Dans le contexte de notre projet, l'usage de RAMMap a donc été essentiel pour libérer les ressources nécessaires à l'exécution fluide de la solution SIEM. Grâce à cet outil, nous avons pu améliorer

la disponibilité de la RAM sans avoir à modifier la configuration matérielle de la VM, ce qui s'est révélé crucial pour éviter les ralentissements ou plantages lors de l'exécution des services de QRadar, tels que le traitement des logs ou l'analyse comportementale en temps réel.

Paramétrage initial de la VM QRadar

L'environnement QRadar a été déployé au sein d'une machine virtuelle hébergée sur Hyper-V, afin de bénéficier à la fois d'une isolation complète du système hôte et d'un niveau de performance adapté aux exigences d'une plateforme SIEM. Ce choix offre également une grande flexibilité dans la gestion des ressources matérielles et une meilleure maîtrise de l'environnement réseau.

Les étapes clés de la configuration initiale de cette machine virtuelle sont les suivantes :

- Création d'une nouvelle machine virtuelle nommée **QRadar**, dédiée exclusivement à l'environnement SIEM.
- Sélection de la **Génération 2** pour profiter du microprogramme UEFI, du démarrage sécurisé, ainsi que du support natif des systèmes 64 bits requis par QRadar.
- Attribution d'une mémoire vive de **13 000 Mo**, garantissant ainsi une exécution fluide des services QRadar et une marge suffisante pour l'analyse en temps réel des journaux.
- Ajout d'un adaptateur réseau connecté à un commutateur virtuel **External**, permettant à la machine virtuelle d'interagir avec le réseau physique et de recevoir les flux de logs provenant de diverses sources.
- Création d'un disque dur virtuel au format VHDX, en mode **taille fixe** pour de meilleures performances d'écriture, d'une capacité de **900 Go**, et stocké dans le répertoire D:\qradar, afin de centraliser les fichiers liés à la VM.
- Enfin, montage de l'image ISO d'installation de la solution IBM QRadar, configurée comme source de démarrage pour initier le processus d'installation du système.

Ces actions ont permis de préparer la VM pour le démarrage et l'installation de QRadar Community Edition, tout en garantissant des ressources suffisantes pour le traitement des logs et l'analyse en temps réel.

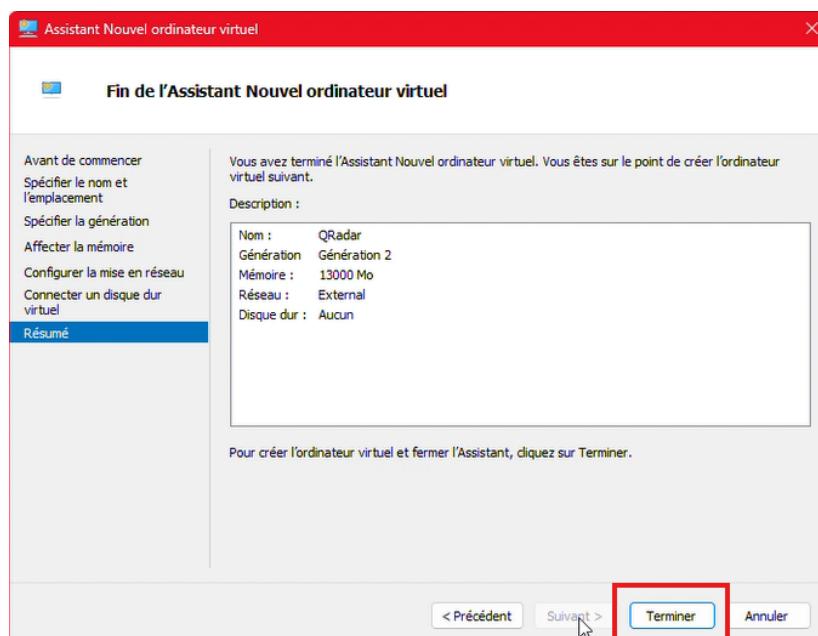


FIGURE 4.62 – Résumé des paramètres initiaux de la VM

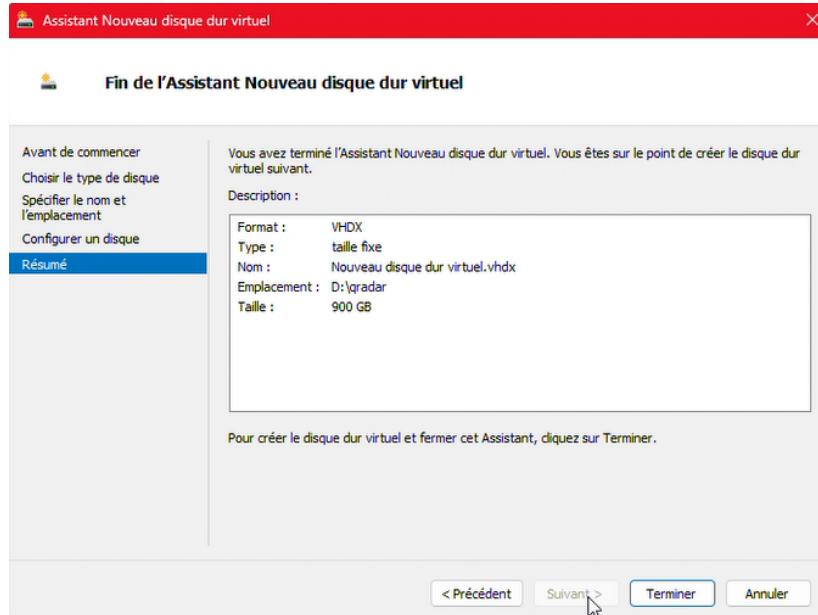


FIGURE 4.63 – Résumé de la création du disque dur virtuel

Après avoir terminé l'ensemble des configurations initiales, nous avons redémarré la machine virtuelle QRadar. L'apparition de l'écran de démarrage suivant confirme que les paramètres ont été appliqués avec succès et que QRadar est prêt à être utilisé.

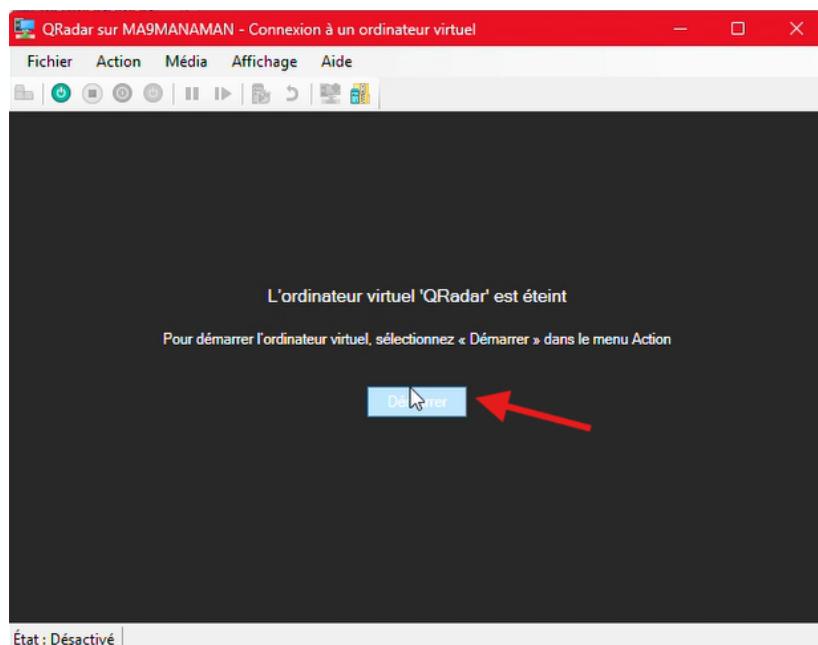


FIGURE 4.64 – Écran de démarrage de la VM QRadar

4.3.2 Intégration au réseau

Après avoir terminé le paramétrage initial, la prochaine étape consiste à configurer les éléments nécessaires au bon fonctionnement de la solution QRadar.

Installation de l'appliance QRadar

La première action effectuée a été le démarrage de l'installation du système d'exploitation requis pour QRadar. Une fois la machine virtuelle lancée, l'écran suivant s'affiche :

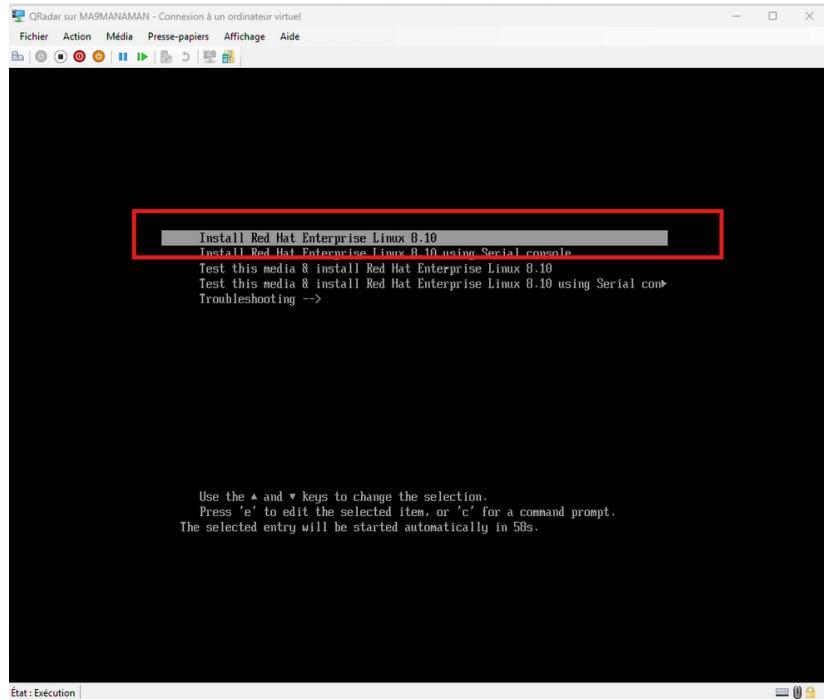


FIGURE 4.65 – Démarrage de l’installation du système

Le système propose le choix entre différents modes de démarrage. Nous avons sélectionné l’option « Factory install » afin d’effectuer une installation propre et complète de QRadar 7.5.0 :

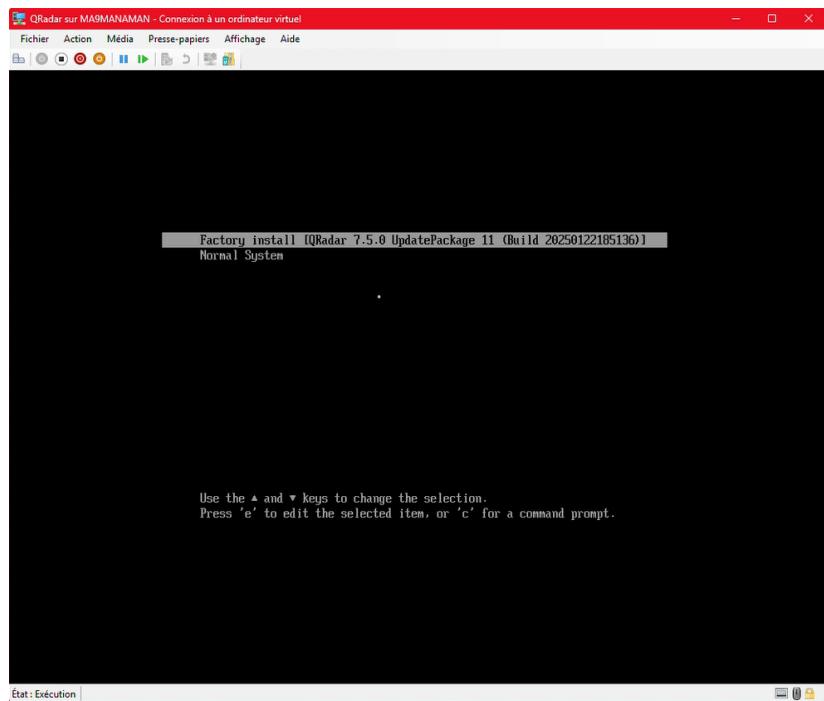


FIGURE 4.66 – Choix du mode d’installation

Nous avons ensuite spécifié le rôle de l’appliance en sélectionnant **Appliance Install**, puis défini le type d’installation sur **Software Install**, et enfin choisi l’option “**All-In-One Console**” pour regrouper toutes les fonctionnalités sur une seule machine :

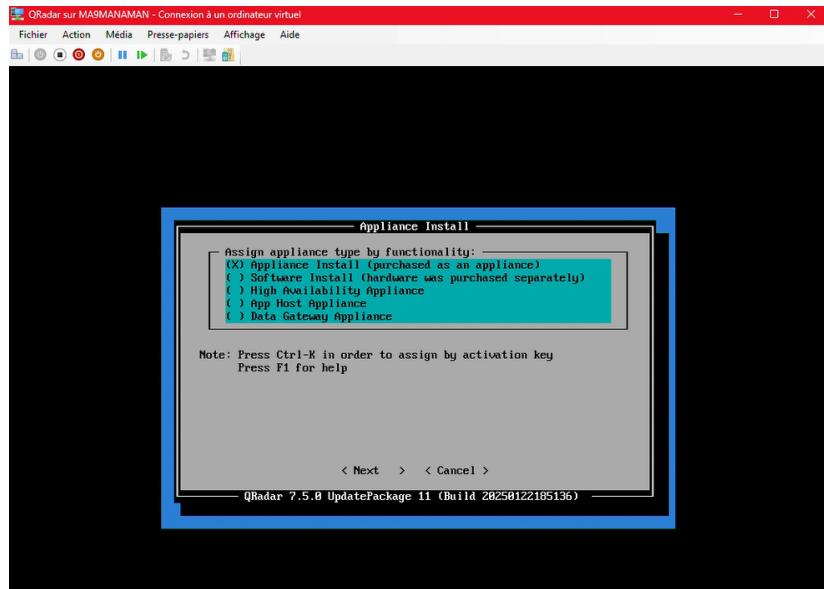


FIGURE 4.67 – Définition du rôle

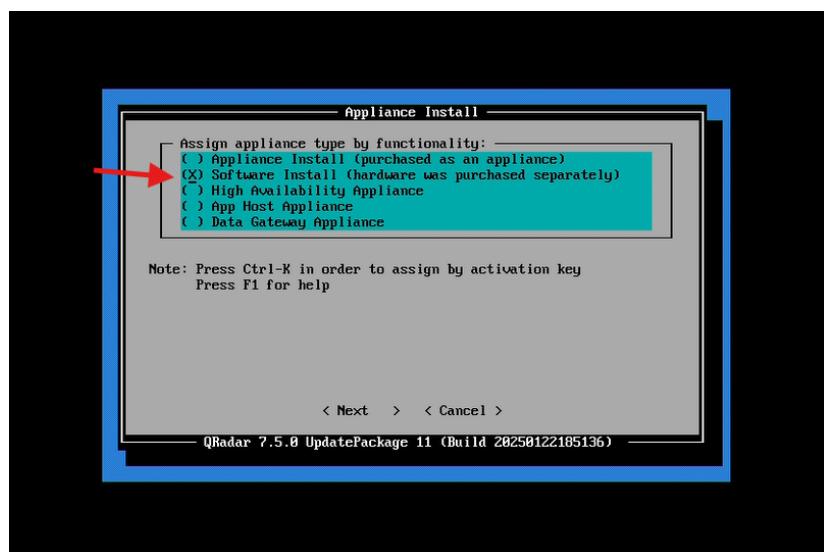


FIGURE 4.68 – Sélection du type d'installation

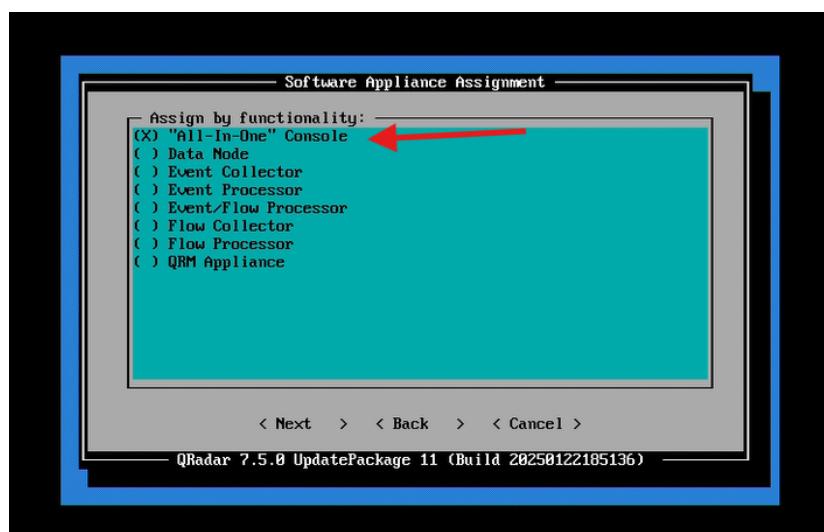


FIGURE 4.69 – Choix du rôle

Puis, nous avons confirmé la configuration standard (Normal Setup (default)) et ajusté la date et l'heure manuellement pour assurer la synchronisation temporelle des événements :

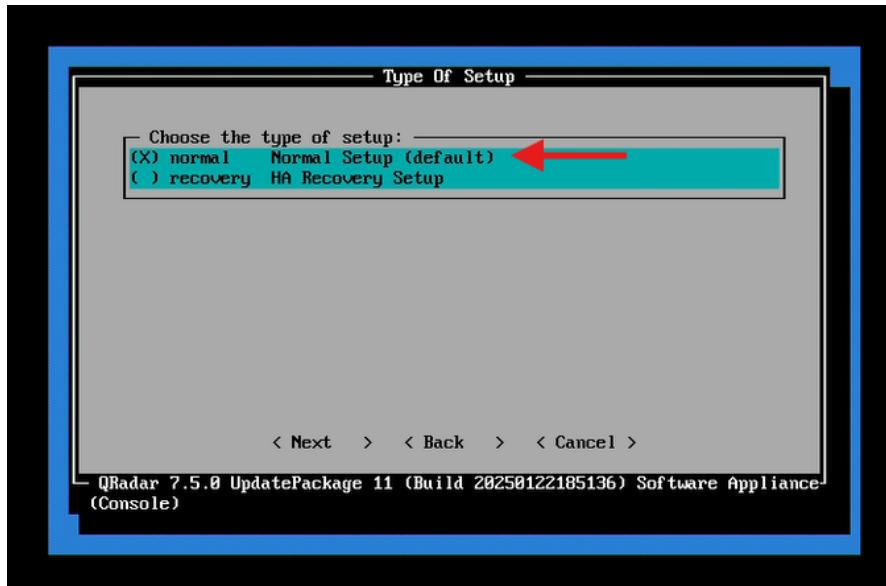


FIGURE 4.70 – Type d’installation sélectionné

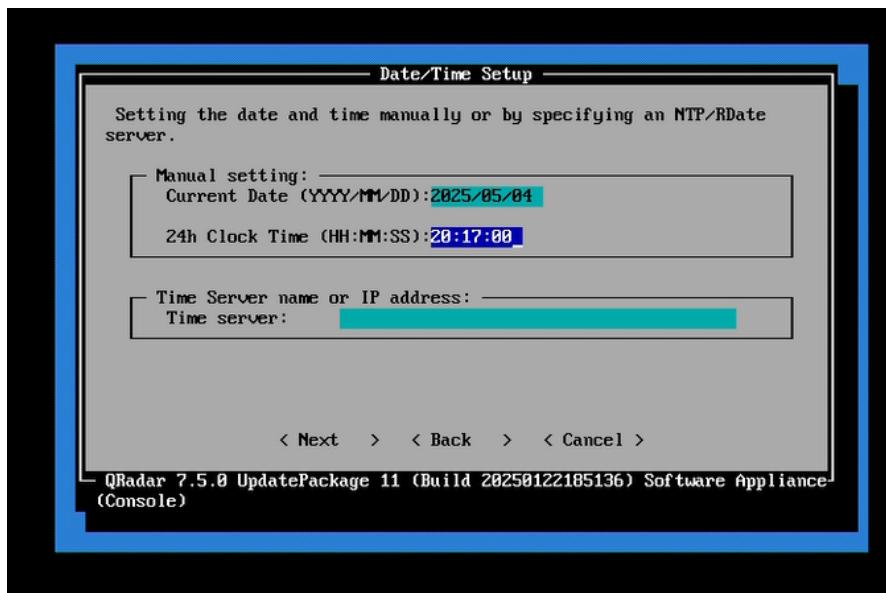


FIGURE 4.71 – Configuration date et de l’heure

vérification de l’adresse IP : Nmap

La configuration du protocole réseau a été effectuée en choisissant IPv4, sans interface liée (bonded), pour un environnement de test simplifié. Nous avons ensuite défini les paramètres suivants :

- **Hostname** : gradar.lab
- **Adresse IP** : 192.168.100.60
- **Masque de sous-réseau, Passerelle et DNS** : laissés vides temporairement

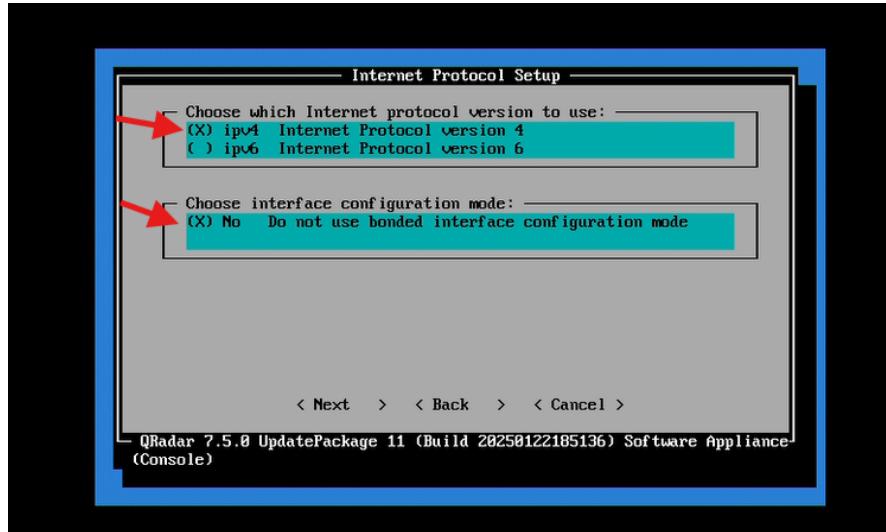


FIGURE 4.72 – Sélection du protocole réseau et mode de configuration

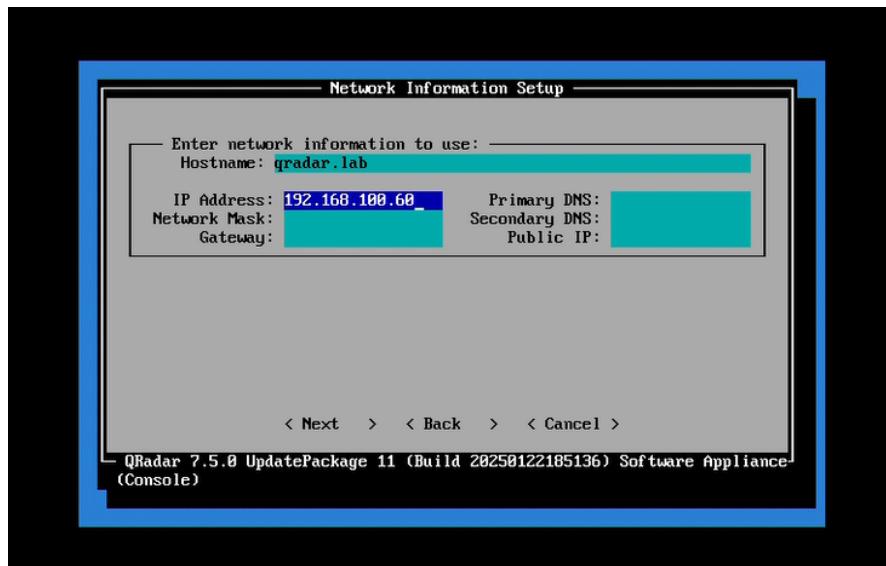


FIGURE 4.73 – Configuration des paramètres réseau de QRadar

Afin de valider la disponibilité de l'adresse IP avant son assignation, nous avons d'abord installé l'outil **Nmap** et son interface graphique **Zenmap** :

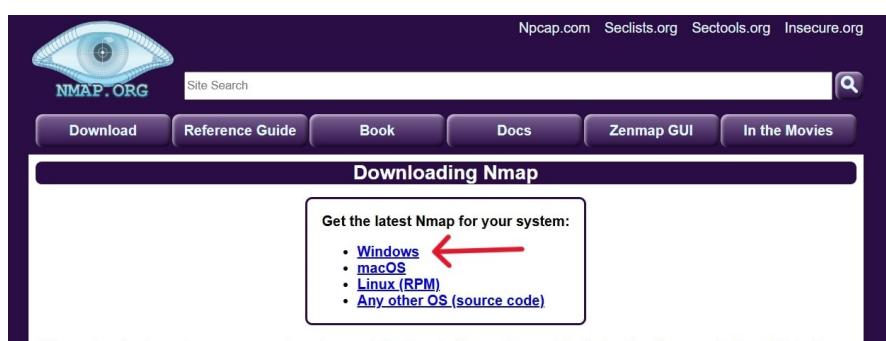


FIGURE 4.74 – téléchargement de Nmap depuis Nmap.org

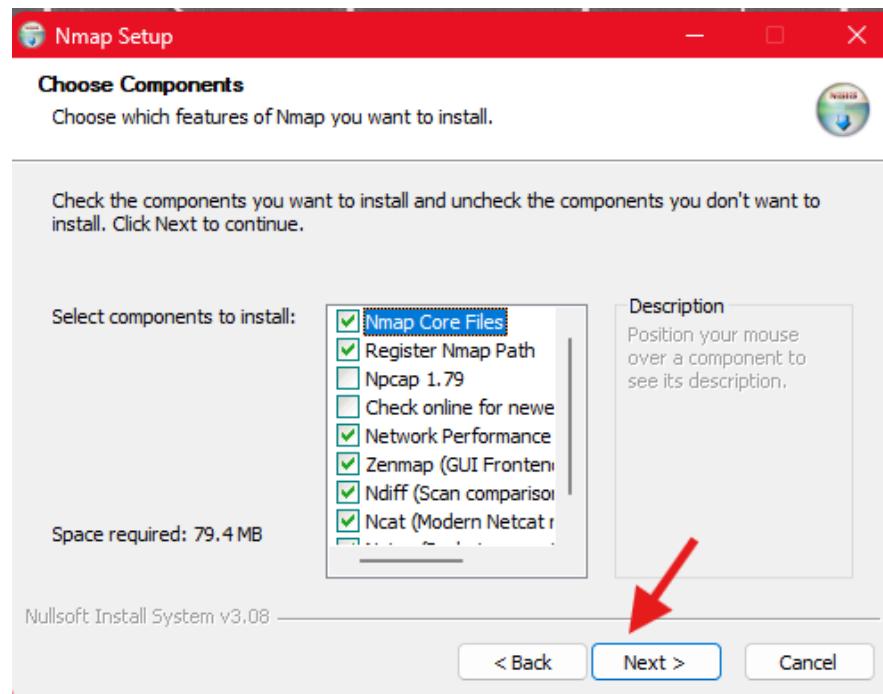


FIGURE 4.75 – Choix des composants à installer avec Nmap

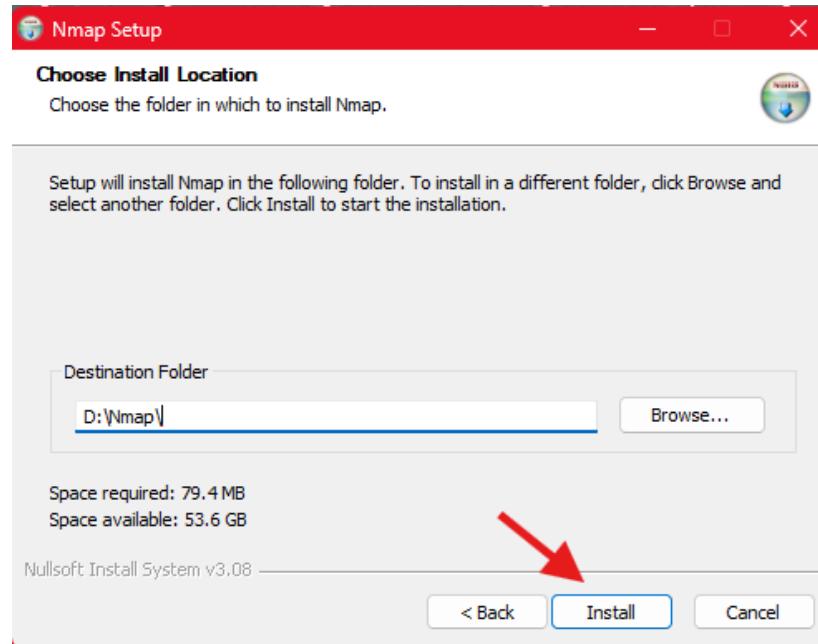


FIGURE 4.76 – Choix du dossier d'installation de Nmap

Ensuite, nous avons exécuté un *ping scan* sur l'adresse cible : nmap -sn 192.168.100.60

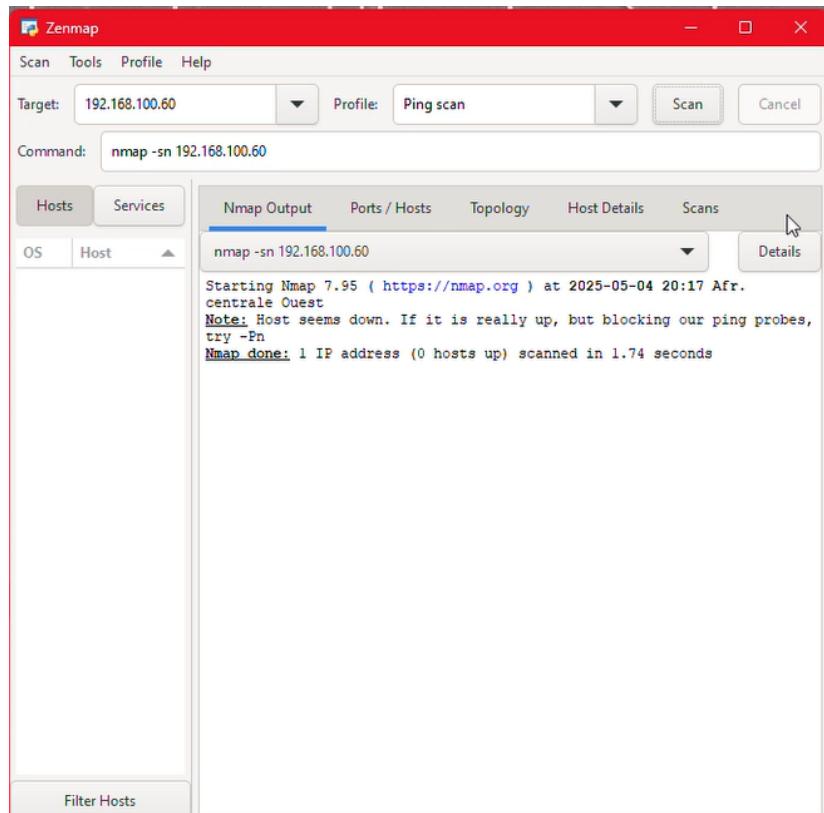


FIGURE 4.77 – Résultat du ping scan

Le retour «0 hosts up» confirme que l'adresse IP 192.168.100.60 est libre et peut être assignée à QRadar sans risques de conflit.

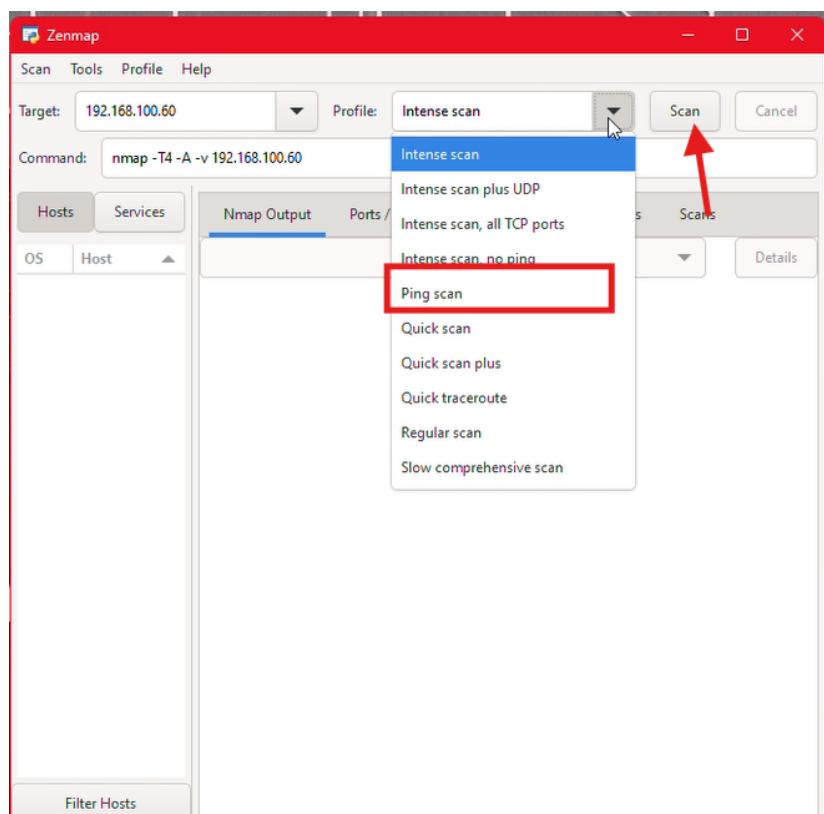


FIGURE 4.78 – Utilisation de Zenmap avec un scan personnalisé

Cette étape de vérification est essentielle pour :

- éviter les conflits d'adresse IP sur le réseau ;
- garantir la stabilité de la configuration réseau ;
- s'assurer que QRadar sera accessible sans interférence.

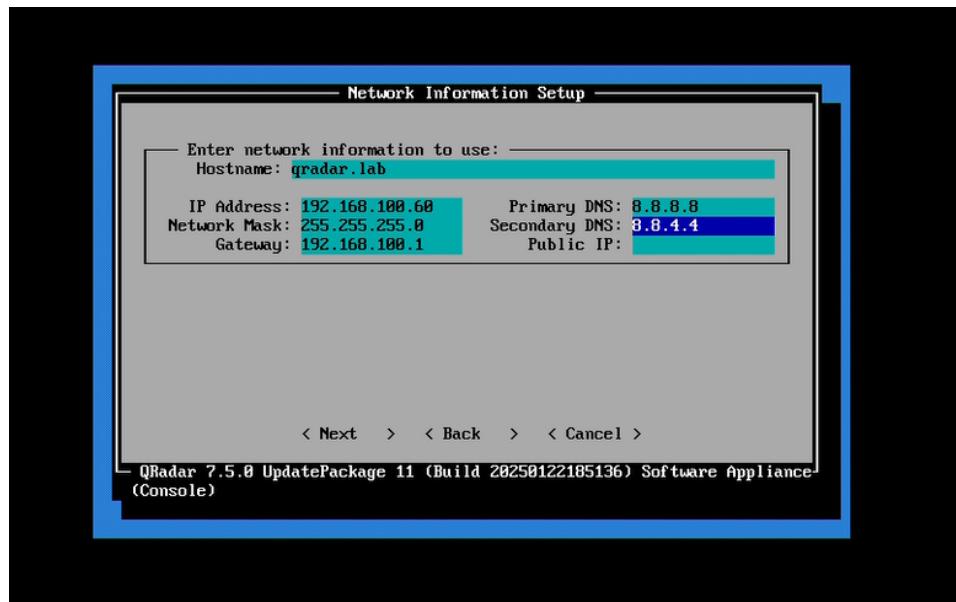


FIGURE 4.79 – Paramètres réseau

Cette étape assure que la plateforme QRadar est correctement intégrée au réseau, avec une connectivité stable et sans conflit IP.

4.3.3 Configuration des sources de logs

Dans cette étape, nous configurons le pare-feu **FortiGate** pour qu'il envoie ses journaux d'activités vers IBM QRadar, permettant ainsi la collecte centralisée et l'analyse des événements réseau.

Vérification de l'état du pare-feu FortiGate

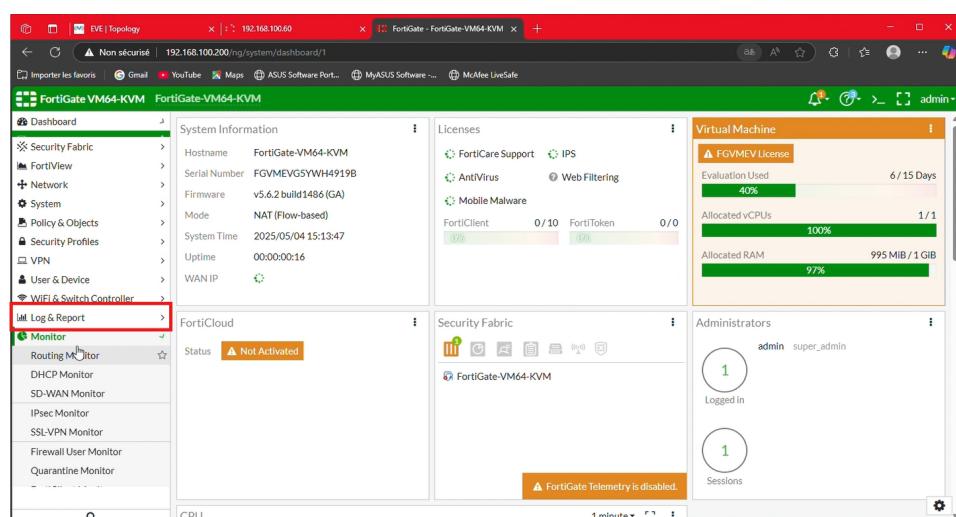


FIGURE 4.80 – Interface système de FortiGate

La capture ci-dessus montre l'interface d'administration de FortiGate :

- Adresse IP : 192.168.100.200
- Firmware : v5.6.2 build 1486
- Mode : NAT (Flow-based)
- Uptime, CPU et RAM utilisées

Activation de l'envoi de logs vers QRadar

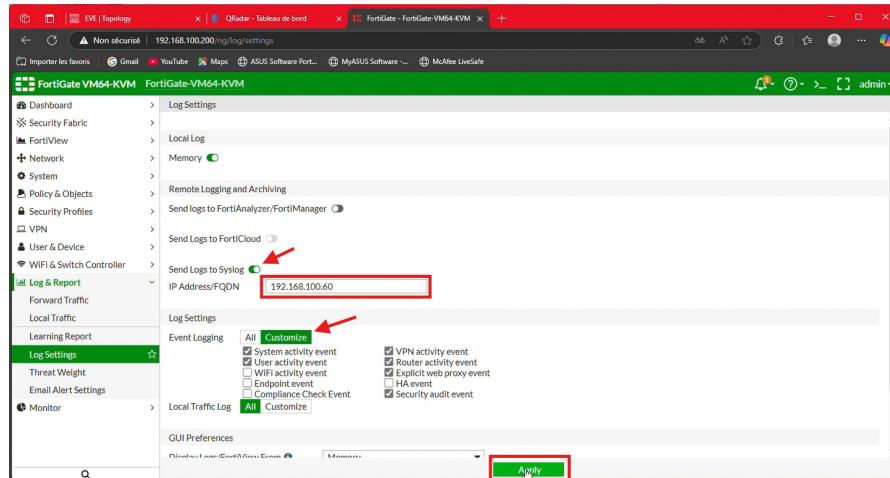


FIGURE 4.81 – Paramétrage Syslog dans “Log Settings”

Dans la section *Log Settings* de FortiGate, nous avons :

- Ajouté l'adresse du serveur Syslog : 192.168.100.60
- Activé l'envoi des types d'événements : système, utilisateur, VPN, trafic, etc.

Vérification de la réception des logs dans QRadar

Src journal	Heure	IP source	Port source	IP de destination	Port de destination
FortiGate @ 192.168.100.200	4 mai 2025 à 23:22:26	192.168.100.69	33169	192.168.100.200	80
FortiGate @ 192.168.100.200	4 mai 2025 à 23:22:15	192.168.100.100	500	192.168.100.200	500
FortiGate @ 192.168.100.200	4 mai 2025 à 23:22:11	192.168.100.69	33158	192.168.100.200	80
FortiGate @ 192.168.100.200	4 mai 2025 à 23:22:01	192.168.100.69	33147	192.168.100.200	60
FortiGate @ 192.168.100.200	4 mai 2025 à 23:21:50	192.168.100.69	33131	192.168.100.200	80
FortiGate @ 192.168.100.200	4 mai 2025 à 23:21:45	192.168.100.100	500	192.168.100.200	500
FortiGate @ 192.168.100.200	4 mai 2025 à 23:21:39	192.168.100.69	33125	192.168.100.200	80
FortiGate @ 192.168.100.200	4 mai 2025 à 23:21:27	192.168.100.69	33113	192.168.100.200	80
FortiGate @ 192.168.100.200	4 mai 2025 à 23:20:45	192.168.100.100	500	192.168.100.200	500

FIGURE 4.82 – Liste des événements reçus

En filtrant par l'IP du FortiGate, on constate l'arrivée continue des journaux. Le test de ping confirme la communication bidirectionnelle sans perte de paquets :

- Ping du PC vers FortiGate

```

Microsoft Windows [version 10.0.26100.3775]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\sadok>ping 192.168.100.200

Envoi d'une requête 'Ping' 192.168.100.200 avec
32 octets de données :
Réponse de 192.168.100.200 : octets=32 temps=1 ms
TTL=255
Réponse de 192.168.100.200 : octets=32 temps<1ms
TTL=255
Réponse de 192.168.100.200 : octets=32 temps<1ms
TTL=255
Réponse de 192.168.100.200 : octets=32 temps<1ms
TTL=255

Statistiques Ping pour 192.168.100.200:
Paquets : envoyés = 4, reçus = 4, perdus = 0
(perde 0%),
Durée approximative des boucles en millisecondes
:
Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms

C:\Users\sadok>

```

FIGURE 4.83 – Test de ping entre le FW 0% de perte

Src journal	Heure	IP source	Port source	IP de destination	Port de destination
FortiGate @ 192.168.100.200	4 mai 2025 à 23:23:15	192.168.100.100	500	192.168.100.200	500
FortiGate @ 192.168.100.200	4 mai 2025 à 23:23:15	192.168.100.69	33213	192.168.100.200	80
FortiGate @ 192.168.100.200	4 mai 2025 à 23:23:01	192.168.100.69	33202	192.168.100.200	80
FortiGate @ 192.168.100.200	4 mai 2025 à 23:22:51	192.168.100.69	33190	192.168.100.200	80
FortiGate @ 192.168.100.200	4 mai 2025 à 23:22:45	192.168.100.100	500	192.168.100.200	500

FIGURE 4.84 – Extrait des événements

Les logs présents confirment la bonne intégration et fournissent les informations nécessaires pour les analyses et corrélations dans QRadar.

4.4 Collecte des logs : WinCollect

Dans cette partie, nous décrivons les deux méthodes principales de collecte des journaux mises en place : l'agent WinCollect pour les machines Windows, et la méthode Syslog pour les équipements réseaux

4.4.1 Intégration d'agent

L'agent WinCollect est utilisé pour collecter les logs des machines Windows et les transmettre à QRadar. Son installation et sa configuration ont été réalisées selon les étapes suivantes :



FIGURE 4.85 – Sélection du package

L'utilisateur sélectionne le fichier d'installation de WinCollect, préalablement téléchargé depuis le portail officiel IBM QRadar.

```
PS C:\Users\sadok\Downloads\wincollect>
PS C:\Users\sadok\Downloads\wincollect> Start-Process .\wincollect-10.1.13-12.x64.msi
```

FIGURE 4.86 – L'exécution de l'installation

L'installation est lancée soit par double-clic sur le fichier exécutable, soit en ligne de commande.

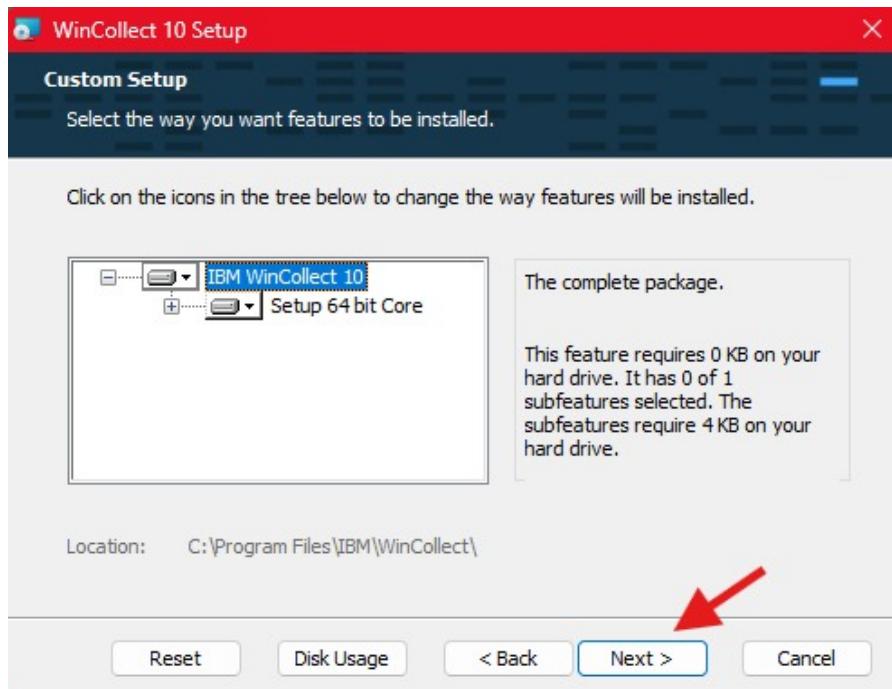


FIGURE 4.87 – Sélection des composants à installer

L'utilisateur accède à l'étape de configuration personnalisée où il peut choisir les composants de WinCollect à installer, avant de cliquer sur Next pour poursuivre l'installation.

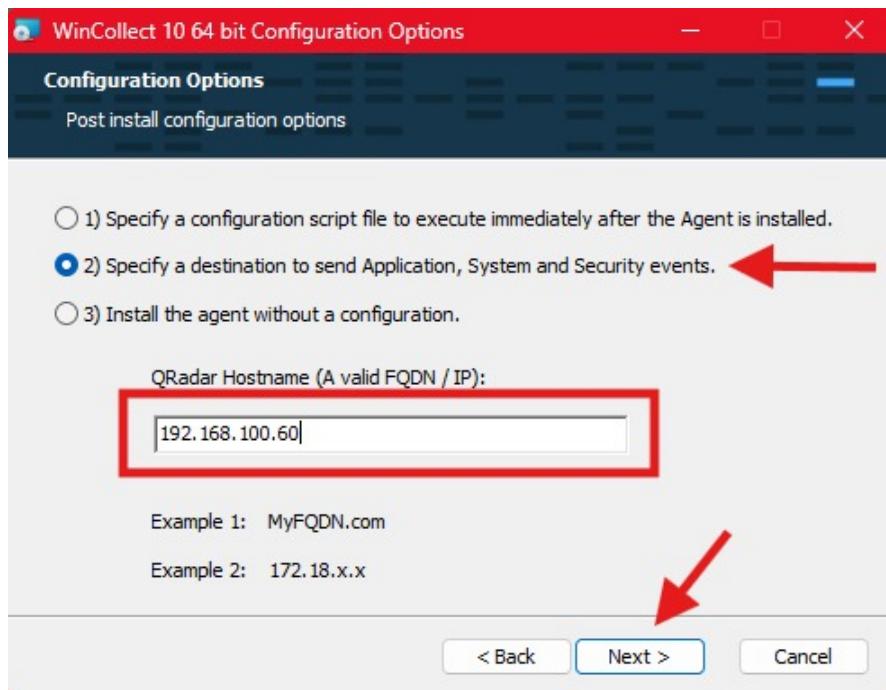


FIGURE 4.88 – Paramètre de destination des logs

Cette étape permet de spécifier où seront envoyés les journaux collectés par l'agent WinCollect.

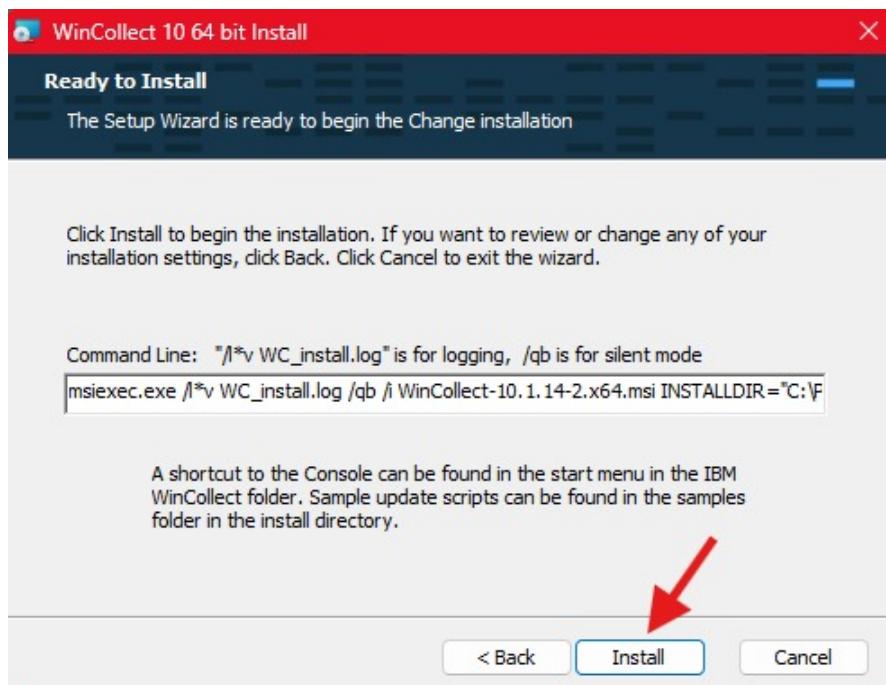


FIGURE 4.89 – Confirmation de l'installation

Le flèche indique le bouton *Install*, que l'utilisateur doit cliquer pour démarrer effectivement le processus d'installation.



FIGURE 4.90 – Écran final de l'installation

Une fois l'installation terminée, un message de confirmation s'affiche indiquant que WinCollect a été correctement installé. L'utilisateur peut alors lancer la console via le menu Démarrer.

FIGURE 4.91 – Étapes complètes de vérification

L'installation de WinCollect peut également être effectuée et vérifiée en ligne de commande à l'aide de PowerShell. Les étapes incluent la vérification de l'intégrité du fichier d'installation via une somme de hachage SHA256, suivie de l'exécution du fichier MSI.

4.4.2 Validation de la centralisation des logs

Une fois l'agent WinCollect correctement installé et configuré sur la machine Windows cible, il est essentiel de vérifier que les journaux (logs) sont bien transmis à la plateforme QRadar. Cette

étape permet de valider le bon fonctionnement de la chaîne de collecte d'événements, un préalable indispensable pour toute détection de menace.

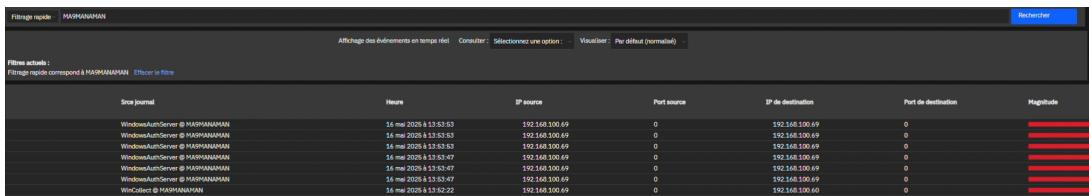


FIGURE 4.92 – Affichage des logs reçus dans QRadar

Cette capture montre l'arrivée des événements dans QRadar depuis une source Windows. Les logs sont affichés en temps réel via la console, prouvant que la communication entre l'agent WinCollect et QRadar est bien établie.

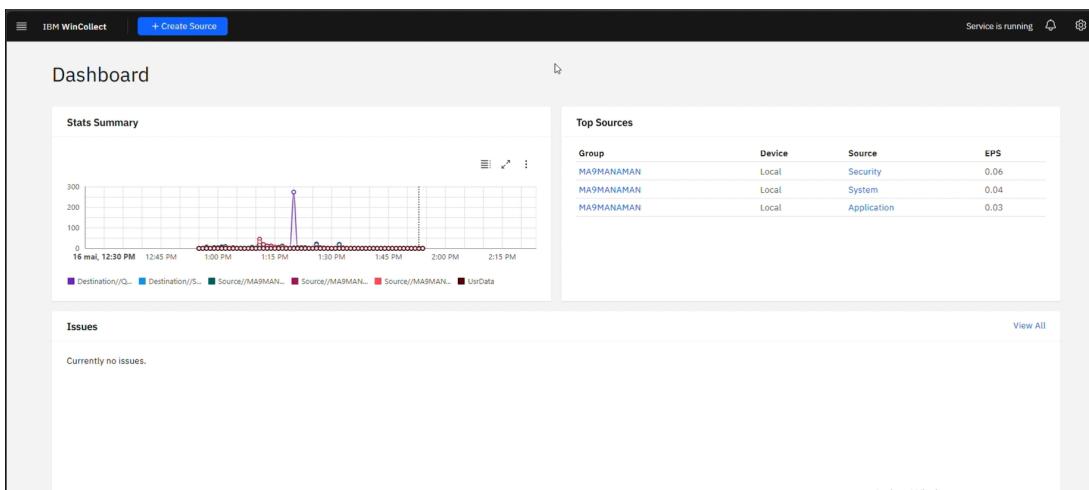


FIGURE 4.93 – Extrait des journaux relatifs à l'activité utilisateur

Ici, QRadar affiche des journaux liés à l'activité de l'utilisateur. Cela permet de suivre les connexions, les exécutions de commandes, ou encore les accès aux ressources critiques.

4.4.3 Détection d'infractions

Après la validation de la collecte, nous passons à la vérification de la détection d'une infraction via une règle personnalisée dans QRadar. Cette étape consiste à simuler un comportement anormal afin d'évaluer la capacité du SIEM à le détecter et à le signaler comme événement de sécurité.

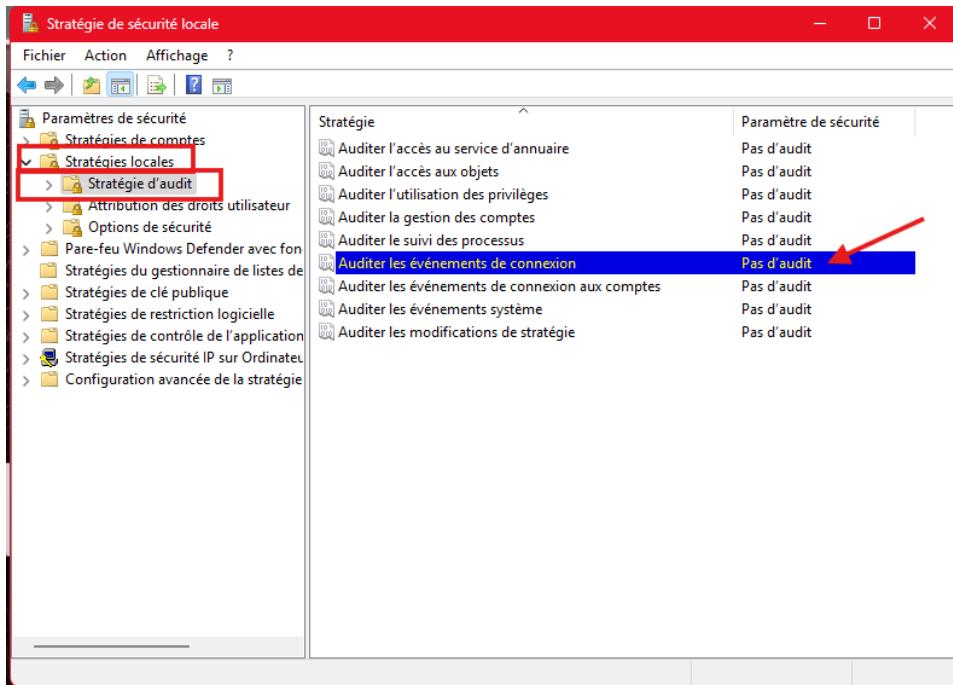


FIGURE 4.94 – Affichage de la règle personnalisée QRadar

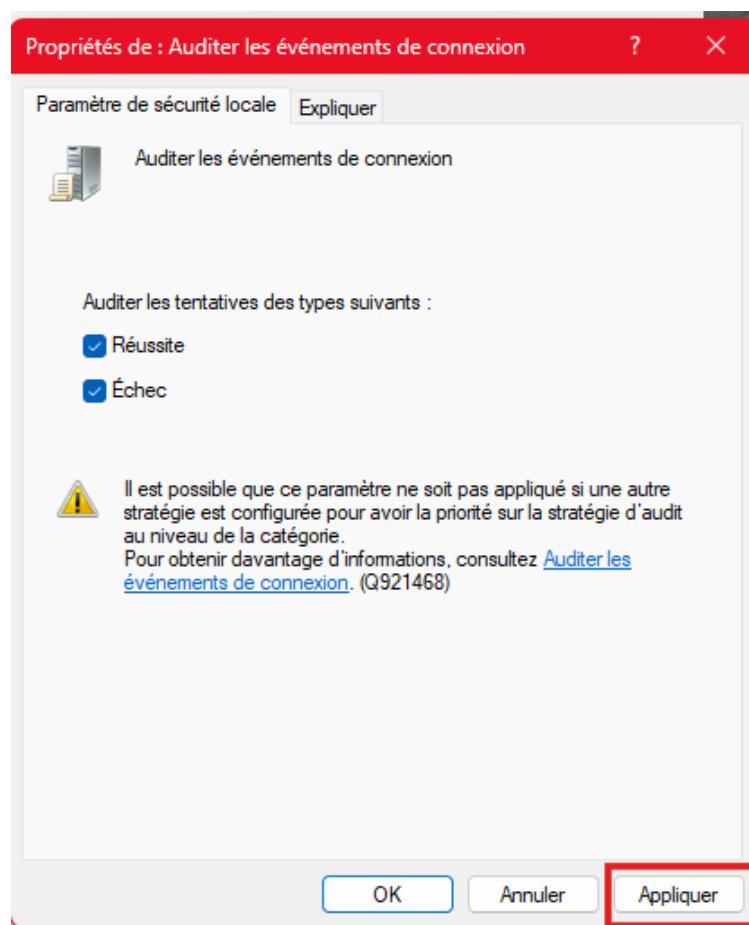


FIGURE 4.95 – Configuration de l'audit des événements

la configuration de l'audit des connexions, permettant d'enregistrer toutes les tentatives, réussies ou non. Cela garantit que les événements soient correctement journalisés dans Windows, puis collectés par WinCollect pour être envoyés à QRadar, qui les analyse et les classe comme possiblement suspects en cas de multiples échecs, comme lors d'un test de faux login.

QRadar surveille en continu les événements entrants et déclenche automatiquement une alerte lorsqu'un comportement suspect est détecté.

The screenshot shows the IBM QRadar web interface. The top navigation bar includes links for Tableau de bord, Infractions (selected), Activité des journaux, Activité réseau, Actifs, Rapports, Risks, Vulnerabilities, Admin, Sources de journaux, and Pulse. The main content area is titled 'Visualiser : Règles' and shows a table of rules. The table columns are: Performances, Nom de la règle, Groupe, Catégorie de règle, Type de règle, Activé, Réponse, Nombre d'évén., Nombre d'infraç., Origine, Date de création, and Date de modifica. A red arrow points to the 'Règles' link in the left sidebar under the 'Infractions' section.

FIGURE 4.96 – Notification générée suite à une infraction détectée

Ici, QRadar a généré une alerte en réponse à une infraction détectée, ce qui démontre que la règle fonctionne comme prévu.

The screenshot shows the IBM QRadar web interface. The top navigation bar includes links for Tableau de bord, Infractions, Activité des journaux, Activité réseau, Actifs, Rapports, Risks, Vulnerabilities, Admin, Sources de journaux, and Pulse. The main content area is titled 'Visualiser : Règles' and shows a table of events. The table columns are: Performances, Nom de la règle, Groupe, Catégorie de règle, Type de règle, Activé, Réponse, Nombre d'évén., Nombre d'infraç., Origine, Date de création, and Date de modifica. A red box highlights the 'Groupe : Authentication' dropdown menu. A red arrow points to the 'Multiple Login F...' rule in the list.

FIGURE 4.97 – Consultation des événements associés à l'alerte

L'analyste peut consulter directement les événements ayant déclenché l'alerte pour mener une investigation approfondie.

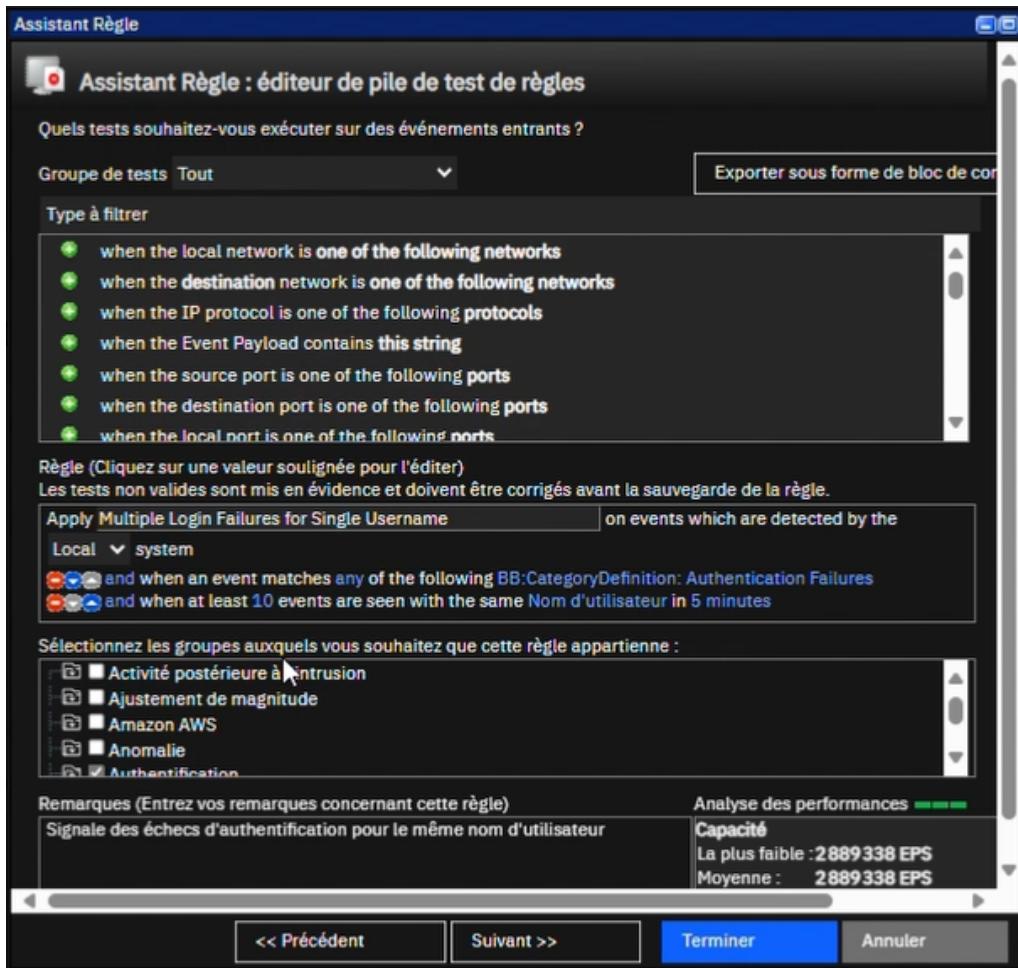


FIGURE 4.98 – Confirmation de l’alerte répétée suite à des connexions suspectes

Cette interface montre la configuration d’une règle personnalisée dans QRadar visant à détecter des tentatives de connexion répétées échouées pour un même nom d’utilisateur. La règle déclenche une alerte lorsqu’au moins 10 échecs d’authentification sont observés dans un intervalle de 5 minutes, ce qui permet d’identifier rapidement un comportement suspect de type attaque par force brute ou tentative d’intrusion persistante.

```

Administrator : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> # Configuration
>> $username = "fauxutilisateur"
>> $program = "notepad.exe"
>>
>> # Nombre de tentatives
>> for ($i=1; $i -le 5; $i++) {
>>     # Exécute la commande runas
>>     Start-Process -FilePath "runas" -ArgumentList "/user:$username $program"
>>
>>     # Attend 1 seconde
>>     Start-Sleep -Seconds 1
>>
>>     # Envoie un mot de passe erroné via SendKeys (simulateur de frappe clavier)
>>     [System.Windows.Forms.SendKeys]::SendWait("motdepasseinvalide{ENTER}")
>> }
>>
>> Write-Host "[+] 5 tentatives effectuées."

```

FIGURE 4.99 – Multiples infractions détectées par la même règle

Plusieurs infractions peuvent être détectées par la même règle. Cela peut indiquer une attaque automatisée ou un comportement malveillant persistant.



FIGURE 4.100 – Détails de l'infraction détectée

Cette capture illustre la tentative d'exécution d'une commande en tant qu'utilisateur fictif, ici nommé *fauxutilisateur*. La fenêtre d'invite de mot de passe, affichée en rouge, indique que l'opération est bloquée ou échoue, ce qui génère une alerte dans QRadar. Cette simulation permet de vérifier que la règle de détection fonctionne correctement.

Src journal	Heure	IP source	Port source	IP de destination	Port de destination
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 14:00:27	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 14:00:27	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 14:00:27	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 14:00:27	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:45	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:45	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:08	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:08	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:08	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:14	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:33	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:41	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:26	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:37	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:33	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:33	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:47	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:47	192.168.100.69	0	192.168.100.69	0
WindowsAuthServer @ MASHANAMAN	16 mai 2026 à 13:59:47	192.168.100.69	0	192.168.100.69	0
WinCollect @ MASHANAMAN	16 mai 2026 à 13:59:22	192.168.100.69	0	192.168.100.60	0

FIGURE 4.101 – Connexion SSH vers la machine cible

Cette action (connexion SSH) est celle qui a été détectée comme suspecte par QRadar. Elle permet de simuler une tentative d'accès non autorisé pour tester la règle.



FIGURE 4.102 – Alerte déclenchée par une connexion SSH suspecte

4.5 Automatisation : Python

L'automatisation est une étape clé pour optimiser la gestion des logs et la détection des incidents dans un SIEM comme QRadar. Python, grâce à sa simplicité et à ses nombreuses bibliothèques, permet de développer des scripts efficaces pour la collecte, l'analyse et la notification automatisée.

4.5.1 Préparation de l'environnement

Avant tous, la dernière version de Python pour Windows (ici, la version 3.13.3) a été téléchargée et installée sur la machine de travail. Cette étape garantit la compatibilité avec les bibliothèques récentes et les meilleures performances pour le développement des scripts d'automatisation.

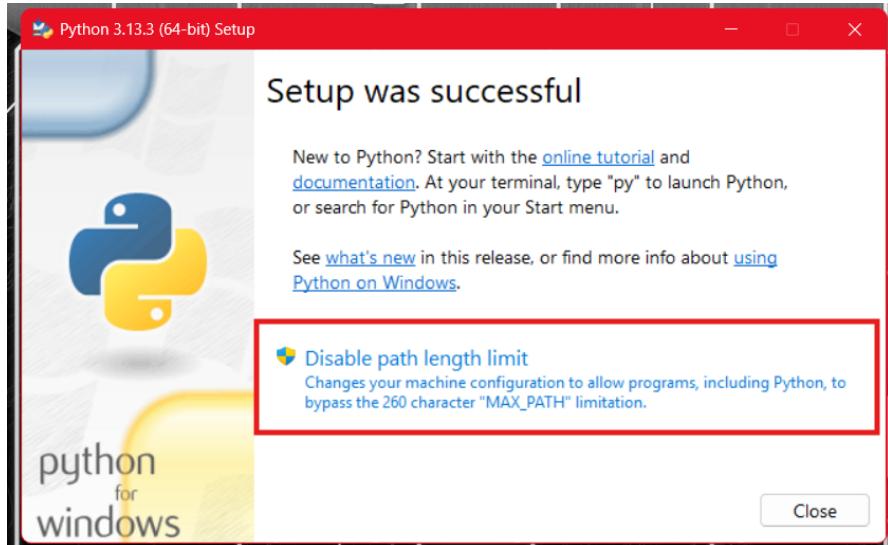


FIGURE 4.103 – Installation réussie

Une fois Python installé, il est nécessaire d'ajouter les bibliothèques indispensables au projet. Les bibliothèques `requests` et `pandas` sont installées via la commande `pip install`, comme illustré ci-dessous. Ces outils sont essentiels pour effectuer des appels API et manipuler efficacement les données issues des logs..

A screenshot of an Administrator Windows PowerShell window. The title bar says "Administrateur : Windows PowerShell". The command entered is "PS C:\WINDOWS\system32> pip install requests pandas". The output shows the progress of the package download and installation, including dependencies like requests, pandas, and various metadata files. A red box highlights the command "pip install requests pandas".

```
Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows | s
PS C:\WINDOWS\system32> pip install requests pandas
Collecting requests
  Downloading requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting pandas
  Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl.metadata (19 kB)
Collecting charset-normalizer<4,>=2 (from requests)
  Downloading charset_normalizer-3.4.2-cp313-cp313-win_amd64.whl.metadata (36 kB)
Collecting idna<4,>=2.5 (from requests)
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)
Collecting urllib3<3,>=1.21.1 (from requests)
  Downloading urllib3-2.4.0-py3-none-any.whl.metadata (6.5 kB)
Collecting certifi=>2017.4.17 (from requests)
  Downloading certifi-2025.4.26-py3-none-any.whl.metadata (2.5 kB)
Collecting numpy=1.26.0 (from pandas)
  Downloading numpy-2.2.5-cp313-cp313-win_amd64.whl.metadata (60 kB)
Collecting python-dateutil=>2.8.2 (from pandas)
  Downloading python_dateutil-2.9.0.post0-py3-none-any.whl.metadata (8.4 kB)
Collecting pytz=>2020.1 (from pandas)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata=>2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting six=>1.5 (from python-dateutil=>2.8.2->pandas)
  Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Downloading requests-2.32.3-py3-none-any.whl (64 kB)
Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl (11.5 MB)
-----
```

FIGURE 4.104 – Installation des bibliothèques via pip

Après installation, la vérification de la présence de la bibliothèque `requests` permet de s'assurer que l'environnement est prêt.

```

PS C:\WINDOWS\system32> pip install requests
Requirement already satisfied: requests in c:\users\sadok\appdata\local\programs\python\python313\lib\site-packages (2.3.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\sadok\appdata\local\programs\python\python313\lib\site-packages (from requests) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\sadok\appdata\local\programs\python\python313\lib\site-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\sadok\appdata\local\programs\python\python313\lib\site-packages (from requests) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sadok\appdata\local\programs\python\python313\lib\site-packages (from requests) (2025.4.26)

[notice] A new release of pip is available: 25.0.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\WINDOWS\system32>

```

FIGURE 4.105 – Vérification de la bibliothèque `requests`

La création d'un dossier dédié au projet facilite l'organisation des scripts et données.

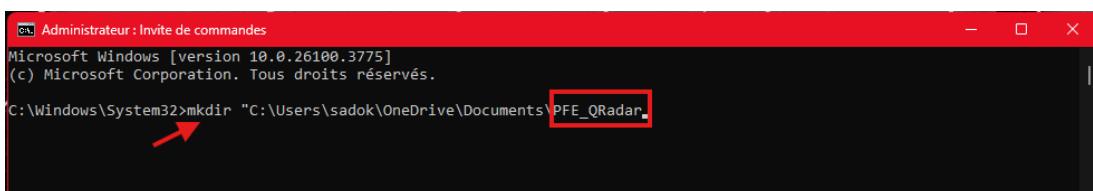


FIGURE 4.106 – Crédit du dossier PFE_QRadar

4.5.2 Script de surveillance

```

import subprocess
import smtplib
from email.message import EmailMessage
from datetime import datetime
import time
# =====
# Configuration email
# =====
SENDER_EMAIL = "██████████"
SENDER_PASSWORD = "██████████"
RECEIVER_EMAIL = "██████████"

# =====

```

FIGURE 4.107 – Aperçu de la configuration initiale des bibliothèques et des variables email

Cette première partie du script importe les bibliothèques nécessaires comme `subprocess`, `smtplib`, et `datetime`, et configure les paramètres d'authentification pour l'envoi d'alertes par email. On y définit l'adresse email de l'expéditeur, son mot de passe, et le destinataire. Cette configuration est cruciale pour automatiser la réponse par alerte.

```

# =====
# Fonction : Envoyer une alerte par email
# =====
def envoyer_alerte_email(destinataire, sujet, corps):
    msg = EmailMessage()
    msg.set_content(corps)
    msg["Subject"] = sujet
    msg["From"] = SENDER_EMAIL
    msg["To"] = destinataire

    try:
        with smtplib.SMTP("smtp.gmail.com", 587) as server:
            server.starttls()
            server.login(SENDER_EMAIL, SENDER_PASSWORD)
            server.send_message(msg)
        print("[INFO] Email d'alerte envoyé.")
    except Exception as e:
        print(f"[ERREUR] Échec de l'email : {e}")

```

FIGURE 4.108 – Fonction d'envoi d'alerte par email

Cette fonction nommée `envoyer_alerte_email` permet d'envoyer un email en cas d'événement critique détecté. Elle utilise la bibliothèque `smtplib` pour se connecter au serveur SMTP de Gmail, puis envoie le message construit à l'aide de la classe `EmailMessage`. En cas d'échec, un message d'erreur est affiché dans la console.

```

# =====
# Fonction : Vérifier l'état du pare-feu Windows
# =====
def verifier_parefeu():
    try:
        # Vérifie si le service MpsSvc (pare-feu) est démarré
        result_service = subprocess.run(
            ["powershell.exe", "-Command", "Get-Service -Name MpsSvc"],
            capture_output=True,
            text=True,
            check=True
        )
        if "Running" not in result_service.stdout:
            return False, "service_stopped"

        # Vérifie chaque profil de pare-feu
        profiles = ["Domain", "Private", "Public"]
        disabled_profiles = []

        for profile in profiles:
            cmd = f"Get-NetFirewallProfile -Name {profile} | Select-Object Enabled"
            result = subprocess.run(
                ["powershell.exe", "-Command", cmd],
                capture_output=True,
                text=True,
                check=True
            )
            output = result.stdout.strip().lower()

            if "false" in output:
                disabled_profiles.append(profile)

        if disabled_profiles:
            return False, disabled_profiles
        else:
            return True, None

    except subprocess.CalledProcessError as e:
        print(f"[ERREUR] Impossible de vérifier le pare-feu : {e}")
        return None, "error"

```

FIGURE 4.109 – Fonction de vérification de l'état du pare-feu

La fonction `verifier_parefeu()` joue un rôle central dans la détection des vulnérabilités liées à la désactivation du pare-feu sur les machines Windows. Elle commence par vérifier l'état du service système `MpsSvc`, qui correspond au service de pare-feu Windows. Si ce dernier n'est pas actif, l'état est immédiatement signalé comme critique.

Ensuite, la fonction exécute une commande PowerShell pour récupérer les paramètres de configuration des trois profils de pare-feu : `Domain`, `Private` et `Public`. Pour chacun de ces profils, elle inspecte si le pare-feu est activé. Si un ou plusieurs profils sont désactivés, leurs noms sont ajoutés à une liste d'alertes. Cette liste est ensuite utilisée pour informer l'utilisateur ou déclencher une remédiation.

La fonction gère également les erreurs d'exécution, notamment en cas de défaillance de la commande PowerShell ou de permissions insuffisantes. Dans ces cas, un message d'erreur est retourné, et le code `None` est utilisé comme indicateur d'échec. Cette robustesse permet de sécuriser davantage la supervision automatique.

```
# =====
# Fonction : Activer le pare-feu si désactivé
# =====
def activer_parefeu_auto():
    print("[ACTION] Tentative de réactivation du pare-feu...")

    try:
        # Démarrer le service s'il est arrêté
        subprocess.run(
            ["powershell.exe", "-Command", "Start-Service -Name MpsSvc"],
            check=True,
            capture_output=True,
            text=True
        )

        # Activer les profils
        profiles = ["Domain", "Private", "Public"]
        for profile in profiles:
            subprocess.run(
                ["powershell.exe", "-Command", f"Set-NetFirewallProfile -Name {profile} -Enabled True"],
                check=True,
                capture_output=True,
                text=True
            )
            print(f"[INFO] Profil {profile} activé.")

        print("[SUCCÈS] Pare-feu entièrement réactivé.")
        return True

    except subprocess.CalledProcessError as e:
        print(f"[ERREUR] Échec lors de la réactivation du pare-feu : {e}")
        return False
```

FIGURE 4.110 – Fonction d'activation automatique du pare-feu si désactivé

La fonction `activer_parefeu_auto()` tente de redémarrer le service du pare-feu Windows (`MpsSvc`) si celui-ci est arrêté, puis active les trois profils (domaine, privé, public) via des commandes PowerShell. En cas de succès, un message de confirmation est affiché ; sinon, une erreur est capturée et signalée.

```

# =====
# Fonction : Gestion complète du pare-feu (vérification + activation auto)
# =====
def gestion_parefeu_complet():
    print("[INFO] Début de la vérification du pare-feu...")
    etat, info = verifier_parefeu()

    if etat is False:
        print("[ALERTE] Le pare-feu est désactivé !")
        reussite = activer_parefeu_auto()

    if reussite:
        message = f"{{datetime.now().strftime('%Y-%m-%d %H:%M:%S')}}\n\n"
        message += "[ALERTE RÉSOLUE]\n"
        message += "Le pare-feu a été trouvé désactivé et vient d'être réactivé.\n"
        message += f"Profils concernés : {', '.join(info)} if isinstance(info, list) else 'Service pare-feu arrêté'{}\n\n"
        message += "✓ Protection restaurée."

        envoyer_alerte_email(RECEIVER_EMAIL, "[ALERTE RÉSOLUE] Pare-feu réactivé", message)
    else:
        message = f"{{datetime.now().strftime('%Y-%m-%d %H:%M:%S')}}\n\n"
        message += "[ALERTE] Activation du pare-feu échouée.\n"
        message += "Le pare-feu est désactivé et ne peut pas être activé automatiquement.\n"
        message += f"Profil(s) impacté(s) : {', '.join(info)} if isinstance(info, list) else 'Service pare-feu arrêté'{}\n\n"
        message += "⚠ Intervention manuelle nécessaire."

        envoyer_alerte_email(RECEIVER_EMAIL, "[ALERTE] Échec de réactivation du pare-feu", message)

    elif etat is True:
        print("[INFO] Le pare-feu est actif. Aucune action nécessaire.")

    else:
        print("[ERREUR] État du pare-feu inconnu ou erreur système.")

```

FIGURE 4.111 – Fonction principale de gestion du pare-feu

La fonction `gestion_parefeu_complet()` assure une supervision continue : elle vérifie l'état du pare-feu, tente une réactivation si nécessaire, puis envoie une alerte par email selon le résultat. Cette fonction constitue le cœur de la logique de remédiation automatique.

```

# =====
# Point d'entrée principal
# =====
if __name__ == "__main__":
    try:
        while True:
            gestion_parefeu_complet()
            print("[INFO] Attente 10sc avant prochaine vérification...")
            time.sleep(10)
    except KeyboardInterrupt:
        print("\n[INFO] Script arrêté par l'utilisateur (Ctrl + C).")
        print("[INFO] Surveillance terminée."]

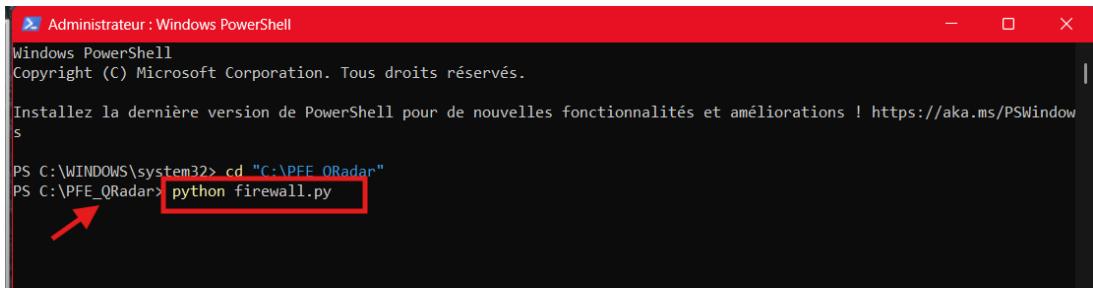
```

FIGURE 4.112 – Point d'entrée principal du script de surveillance

Le point d'entrée `main` lance une boucle infinie qui appelle régulièrement la fonction de gestion du pare-feu toutes les 10 secondes. Ce mécanisme permet d'assurer une surveillance continue, avec la possibilité d'interruption manuelle via `Ctrl+C`.

Le script final, enregistré sous le nom `firewall.py`, intègre l'ensemble des fonctions développées pour une surveillance complète du pare-feu Windows. L'utilisation de l'extension `.py` est essentielle pour garantir une exécution correcte par l'interpréteur Python.

Pour lancer le script, une console PowerShell avec droits administrateur est nécessaire. La navigation vers le dossier du projet suivie de la commande `python firewall.py` initie la surveillance.



```
Administrator : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> cd "C:\PFE_QRadar"
PS C:\PFE_QRadar> python firewall.py
```

FIGURE 4.113 – Lancement du script depuis PowerShell en mode administrateur

4.5.3 Validation d’automatisation

Une batterie de tests a été réalisée pour valider le fonctionnement du script dans différents scénarios :

Désactivation manuelle du pare-feu

Le pare-feu est volontairement désactivé via l’interface Windows pour simuler une faille de sécurité. Cette configuration permet de vérifier la détection par le script.

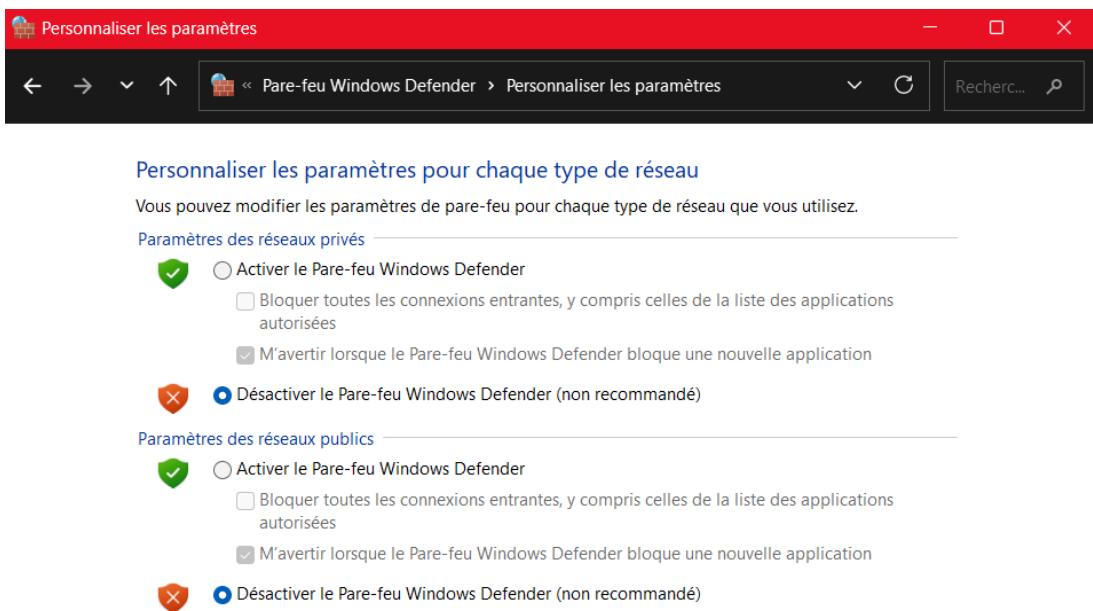


FIGURE 4.114 – désactivation du pare-feu Windows

Résultats du test :

- Le script détecte instantanément l’état désactivé du pare-feu
- La fonction de réactivation automatique s’exécute avec succès
- Une notification email est générée avec les détails de l’incident

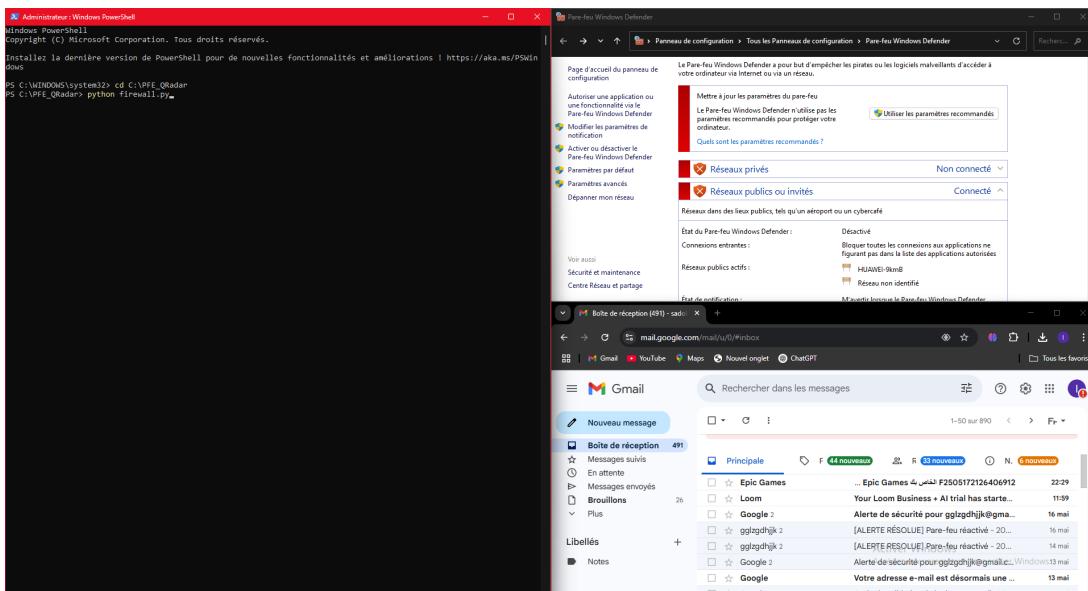


FIGURE 4.115 – Sortie console lors de la détection d'un pare-feu désactivé

Notification par email

L'efficacité du système d'alerte est vérifiée par la réception effective des notifications dans la boîte mail configurée.

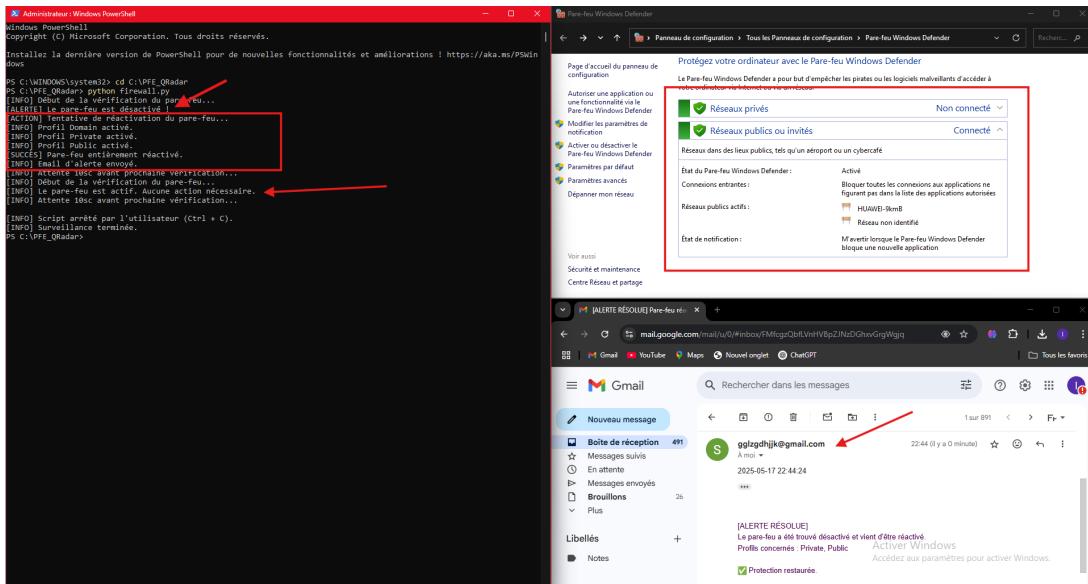


FIGURE 4.116 – Email de confirmation reçu après réactivation automatique

Cette validation complète démontre que la solution répond aux exigences fonctionnelles :

- Surveillance en temps réel (toutes les 10 secondes)
- Détection précise de l'état des trois profils (Domaine/Privé/Public)
- Correction automatique des anomalies détectées
- Notification immédiate de l'administrateur
- Gestion propre de l'arrêt (Ctrl+C)

L'ensemble du processus, depuis l'implémentation jusqu'aux tests de validation, confirme la robustesse de cette solution d'automatisation pour la sécurité des systèmes Windows.

Conclusion

La mise en place des différents composants a permis de créer un environnement de test fonctionnel et représentatif des objectifs du projet. QRadar a été intégré avec succès aux principales sources de logs, notamment les machines Windows grâce à l'agent WinCollect, et le pare-feu FortiGate. L'ajout de scripts Python a introduit une automatisation de base, comme le blocage d'adresses IP suspectes. Chaque fonctionnalité a été testée individuellement, assurant ainsi une cohérence globale et une validation efficace de l'environnement en conditions quasi-réelles.

Chapitre 5

Tests et validation

Introduction

Ce chapitre présente la phase finale du projet, dédiée à la mise en œuvre pratique des scénarios de test dans l'environnement construit. L'objectif est de vérifier le bon fonctionnement de l'architecture SIEM déployée, notamment sa capacité à collecter, corréler et analyser les logs provenant de différentes sources. Les tests sont conçus pour simuler des situations réalistes, telles que des comportements malveillants, des tentatives d'intrusion ou des anomalies réseau. Ils permettent d'observer les réactions du système, d'évaluer la précision des alertes générées, et d'analyser l'efficacité des mécanismes de réponse automatique mis en place. Cette étape est essentielle pour valider l'adéquation de la solution aux exigences de sécurité définies, et pour identifier d'éventuelles améliorations à apporter.

5.1 Environnement de travail

5.1.1 Infrastructure matérielle

Les travaux ont été réalisés sur un poste personnel offrant des performances suffisantes pour exécuter plusieurs machines virtuelles et outils de sécurité de manière fluide. La configuration matérielle est la suivante :

- **Processeur** : AMD Ryzen™ 7 7435HS
- **Mémoire RAM** : 24 Go DDR5
- **Stockage** : SSD de 512 Go
- **Carte graphique** : NVIDIA GeForce RTX 2050 avec 4 Go de mémoire dédiée GDDR6
- **Système d'exploitation** : Windows 11 Pro

Cette configuration a permis le déploiement stable de plusieurs environnements virtualisés via la plateforme EVE-NG, essentielle à la simulation du réseau et des outils de sécurité dans le cadre du projet.

5.1.2 Infrastructure logicielle

L'infrastructure logicielle utilisée dans ce projet repose sur une combinaison de systèmes d'exploitation, d'outils de sécurité et de solutions de simulation réseau. Voici les composants principaux :

- **Systèmes d'exploitation** : Red Hat Enterprise Linux (pour QRadar), Windows Server
- **Plateforme de simulation** : EVE-NG pour l'interconnexion des équipements (pare-feux, routeurs, serveurs, postes clients)
- **Outils déployés** :

Outil	Logo	Fonction principale
<i>IBM QRadar + agent WinCollect</i>		Solution SIEM utilisée pour la collecte, la corrélation et l'analyse des événements de sécurité. WinCollect est un agent Windows que nous avons installé et configuré afin d'assurer la collecte efficace des journaux des machines vers QRadar.
<i>FortiGate</i>		Pare-feu matériel/virtuel utilisé pour le filtrage, la sécurisation du trafic et la mise en place d'un VPN IPsec.
<i>WinSCP</i>		Transfert de fichiers sécurisé via SCP/SFTP, utilisé pour l'échange de journaux et de configurations.
<i>Nmap</i>		Outil de scan de ports et de découverte réseau, utile pour la détection des services actifs.
<i>RamMap</i>		Analyse détaillée de l'utilisation de la mémoire système pour optimiser les performances de virtualisation.
<i>Scripts Python</i>		Automatisation de certaines tâches comme le blocage d'adresses IP malveillantes.

TABLE 5.1 – Outils déployés dans l'environnement

- Outils utilisés :

Outil	Logo	Fonction principale
<i>Hyper-V</i>		Virtualiseur de Microsoft utilisé pour héberger et gérer les machines virtuelles nécessaires à l'environnement du projet, utilisable uniquement avec les éditions Windows Professionnel.
<i>Overleaf (LaTeX)</i>		Éditeur en ligne collaboratif utilisé pour la rédaction du rapport PFE avec un format professionnel en LaTeX.
<i>Canva</i>		Outil de conception graphique en ligne utilisé pour la création des supports visuels de présentation du projet.

<i>Visual Studio Code (VS Code)</i>		Environnement de développement léger utilisé pour la rédaction et l'édition des scripts PowerShell, facilitant le développement avec des fonctionnalités telles que la coloration syntaxique, l'autocomplétion et l'intégration terminal.
<i>Windows PowerShell</i>		Interface de ligne de commande puissante permettant l'exécution des scripts d'automatisation, la gestion des services Windows, ainsi que la collecte et la surveillance des événements système.

TABLE 5.2 – Outils utilisés pour la gestion du projet

5.1.3 Contraintes techniques

Au cours de la mise en place de l'environnement, plusieurs contraintes ont été rencontrées :

- **Limites matérielles** : QRadar est exigeant en ressources (RAM, CPU, disque). Le lancement simultané de plusieurs VMs a nécessité une optimisation rigoureuse des ressources.
- **Compatibilité logicielle** : certaines sources de logs n'étaient pas directement compatibles avec QRadar sans configuration avancée.
- **Virtualisation sous Windows Famille** : la version initiale du système (Windows 11 Famille) ne supportait pas certaines fonctionnalités de virtualisation comme Hyper-V. Une mise à niveau vers Windows 11 Pro a été nécessaire.
- **Limitations réseau simulé** : certaines fonctionnalités comme la capture de flux NetFlow n'étaient pas entièrement prises en charge dans l'environnement de simulation.

5.2 Scénarios de test

Dans cette section, nous présentons les principaux scénarios de test validant la mise en œuvre de la solution SIEM QRadar dans l'environnement simulé. Chaque scénario évalue une étape clé du processus de surveillance et de réponse aux incidents : collecte, détection, réaction et résilience.

5.2.1 Scénario 1 : Collecte des logs

- **Objectif** : Vérifier que QRadar est capable de collecter des logs depuis différentes sources du réseau.
- **Sources configurées** :
 - Firewall FortiGate (logs de trafic, logs d'attaque IPS)
 - Machines Windows (logs système et sécurité via WinCollect)
 - IDS (Snort) pour les alertes d'intrusion
- **Procédure** :
 - Génération de trafic réseau simulé entre les machines virtuelles.
 - Observation en temps réel dans QRadar des logs entrants par source.
- **Résultat attendu** : QRadar affiche les événements provenant de chaque source dans l'onglet “Log Activity”, avec les métadonnées correctes.

5.2.2 Scénario 2 : Détection d'activités malveillantes

- **Objectif** : Vérifier la capacité de QRadar à identifier des comportements suspects ou malveillants.

- **Procédure :**
 - Simulation d'une tentative de scan réseau depuis une machine compromise.
 - Surveillance des alertes générées automatiquement dans QRadar.
- **Règles utilisées :** Règles QRadar intégrées de type "Port Scan Detected" ou "Reconnaissance Activity".
- **Résultat attendu :** Déclenchement d'un événement de sécurité (offense) avec détails sur l'attaquant, la victime et la nature de l'activité.

5.2.3 Scénario 3 : Réaction automatisée

- **Objectif :** Tester un script Python déclenché automatiquement lorsqu'un incident critique est détecté.
- **Procédure :**
 - Détection d'une tentative d'intrusion simulée.
 - Déclenchement d'un script QRadar Offense API ou webhook.
 - Le script met automatiquement en liste noire l'IP source sur le firewall FortiGate via son API.
- **Résultat attendu :** L'IP attaquante est bloquée en temps réel, l'événement est consigné dans QRadar et dans les logs du pare-feu.

5.2.4 Scénario 4 : Simulation d'une attaque complète

- **Objectif :** Simuler une attaque de type chaîne de compromission (kill chain) pour tester l'ensemble du processus SIEM.
- **Étapes de l'attaque simulée :**
 1. Reconnaissance (scan de ports)
 2. Exploitation (tentative d'accès RDP avec brute force)
 3. Déploiement d'un malware
 4. Communication vers un serveur C&C
- **Évaluation :**
 - Détection successive des activités par QRadar
 - Corrélation des événements pour générer une offense globale
 - Réponse automatisée par blocage ou alerte
- **Résultat attendu :** QRadar identifie toutes les étapes de l'attaque, génère une offense enrichie, et active une réponse si configurée.

5.3 Analyse des résultats

5.3.1 Limites du projet

Cette section présente une analyse détaillée des principales limitations rencontrées lors du déploiement de la solution de gestion des logs et de détection des incidents avec IBM QRadar. Ces limites concernent l'environnement d'exécution, la configuration des outils et les choix technologiques adoptés.

1. Performance limitée dans un environnement virtualisé :

QRadar a été déployé dans un environnement EVE-NG avec des ressources matérielles restreintes. Cette contrainte a provoqué des lenteurs importantes dans l'analyse des événements, affectant la réactivité du système et la fluidité de l'interface. Un tel environnement n'est pas optimal pour simuler une détection en temps réel.

2. Collecte de logs partielle ou instable :

Certains équipements, comme les clients Windows ou le pare-feu, n'ont pas systématiquement transmis leurs journaux. Cela est dû à une configuration Syslog incorrecte, une mauvaise connectivité ou l'absence de configuration de l'agent WinCollect, ce qui a limité l'analyse complète des événements de sécurité.

3. Alertes peu pertinentes (faux positifs) :

L'utilisation des règles de détection par défaut a entraîné un grand nombre d'alertes non critiques. Cela rend difficile la distinction entre un incident sérieux et un événement bénin, ce qui compromet l'efficacité du tri et de la réponse aux incidents.

4. Script Python limité en fonctionnalités :

Le script Python conçu pour bloquer des IP malveillantes fonctionne dans des cas simples, mais il est insuffisant pour gérer des scénarios complexes. Il ne dispose ni de gestion des logs, ni d'interface graphique, ni d'intégration avec d'autres outils.

5. Outils en version gratuite ou limitée :

L'utilisation de QRadar Community Edition et FortiGate en version d'évaluation introduit des restrictions majeures : limite d'événements (EPS), absence de support technique, et impossibilité d'utiliser certaines fonctionnalités avancées comme des règles personnalisées ou des intégrations spécifiques.

5.3.2 Axes d'amélioration

Des actions concrètes peuvent être envisagées pour surmonter les limites précédentes et rendre la solution plus robuste, efficace et proche d'un usage en entreprise réelle.

- **Renforcer l'infrastructure matérielle** : Migrer QRadar vers une machine avec plus de RAM/CPU ou un hyperviseur dédié (comme VMware) pour garantir des performances optimales.
- **Optimiser la collecte des journaux** : Configurer de manière correcte tous les équipements pour qu'ils envoient leurs logs en continu. L'installation et le bon paramétrage d'agents comme WinCollect ou NXLog sont essentiels.
- **Personnaliser les règles de corrélation** : Créer des règles spécifiques adaptées à l'environnement simulé et intégrer des sources de Threat Intelligence fiables (ex : IBM X-Force, MITRE ATT&CK).
- **Évoluer vers une réponse automatique avancée** : Étendre le script Python avec une interface web (par exemple avec Flask), un moteur de règles, et un système de journalisation des actions.
- **Adopter des licences officielles** : Investir dans des versions complètes de QRadar et FortiGate pour bénéficier de toutes les fonctionnalités, d'un support professionnel et d'une meilleure intégration avec d'autres solutions de sécurité.

Conclusion

Les résultats obtenus confirment que la solution mise en place répond globalement aux objectifs de détection et de réaction face aux incidents de sécurité. Les événements critiques ont pu être identifiés, et des réponses automatisées simples ont été déclenchées avec succès. Toutefois, certaines limites ont été constatées, notamment en ce qui concerne la couverture de scénarios complexes et la sophistication des mécanismes d'automatisation. Ces constats ont permis de dégager plusieurs axes d'amélioration pour renforcer la robustesse, la réactivité et l'adaptabilité de la solution, notamment dans des contextes de production à plus grande échelle.

Conclusion Générale

Ce projet de fin d'études a représenté une opportunité précieuse de mettre en pratique les acquis théoriques et pratiques développés au cours de notre parcours universitaire. Il s'est inscrit dans un contexte où la cybersécurité occupe une place stratégique croissante dans les organisations, face à l'augmentation constante des menaces et à la complexité des environnements numériques modernes. Réalisé au sein de l'entreprise **Network Associates**, spécialisée dans l'intégration de solutions réseau et la sécurité informatique, ce stage avait pour objectif de concevoir et de mettre en œuvre une solution de gestion des logs à travers la plateforme **IBM QRadar**. L'enjeu était de permettre une centralisation efficace des événements de sécurité, une corrélation pertinente des données collectées, ainsi qu'une réaction automatisée aux incidents détectés. Ce projet nous a permis de répondre à une problématique actuelle en sécurité informatique tout en consolidant notre compréhension des architectures réseau, des outils de surveillance et des méthodes de réponse aux menaces.

Le travail réalisé s'est articulé autour de plusieurs étapes majeures, complémentaires et interdépendantes. Il a d'abord fallu s'approprier les fondamentaux des systèmes SIEM et évaluer les différentes approches existantes, afin d'identifier les meilleures pratiques pour assurer une intégration cohérente dans l'environnement ciblé. Ensuite, une architecture réseau a été conçue et simulée à l'aide de la plateforme d'émulation **EVE-NG**, intégrant des pare-feu **FortiGate**, des machines générant des logs, et un serveur QRadar. Une attention particulière a été portée à la configuration des protocoles de collecte comme **Syslog** et **WinCollect**, pour garantir une transmission fiable et sécurisée des événements. Enfin, des scripts Python ont été développés dans une logique d'automatisation, afin de réagir de manière adaptée à certains incidents critiques, et l'ensemble du système a été soumis à une série de tests afin de valider son bon fonctionnement et sa robustesse. Ce processus nous a également permis d'identifier les limites de notre solution et d'ouvrir des perspectives d'amélioration à moyen et long terme.

Au-delà des apports techniques, cette expérience a été humainement et professionnellement formatrice. Elle nous a permis de développer une méthodologie de travail rigoureuse, une capacité d'analyse approfondie et une aptitude à résoudre des problèmes concrets dans des délais parfois contraints. Le travail en collaboration avec les encadrants et les professionnels de l'entreprise nous a sensibilisés aux exigences du monde de l'ingénierie réseau et de la sécurité informatique, tout en renforçant notre autonomie et notre sens des responsabilités. Ce projet a été une étape déterminante dans notre parcours académique, marquant la transition vers le monde professionnel avec des compétences renforcées, une vision plus claire des enjeux du secteur, et une motivation renouvelée pour continuer à progresser dans le domaine de la cybersécurité et de l'administration des systèmes et réseaux.

Résumé

Ce projet a été réalisé dans le cadre d'un stage au sein de l'entreprise Network Associates, avec pour objectif la mise en place d'une solution de gestion centralisée des journaux (logs) à l'aide de la plateforme SIEM IBM QRadar. Le contexte de ce projet repose sur le besoin d'identifier, collecter, corréler et analyser les événements de sécurité générés par divers équipements et serveurs. Après une phase d'analyse des besoins, une architecture réseau simulée a été conçue sur la plateforme EVE-NG, incluant deux sites interconnectés via un tunnel VPN IPsec, chacun protégé par un pare-feu FortiGate. Des machines virtuelles Windows ont été configurées et intégrées à QRadar comme sources de logs. Des scripts PowerShell ont été développés pour automatiser la surveillance de l'état de sécurité. Les résultats obtenus ont démontré l'efficacité de la corrélation d'événements dans QRadar et une amélioration notable de la détection des incidents. Le projet ouvre la voie à des extensions futures, notamment l'intégration de scanners de vulnérabilité et des systèmes de réponse automatique.

Abstract

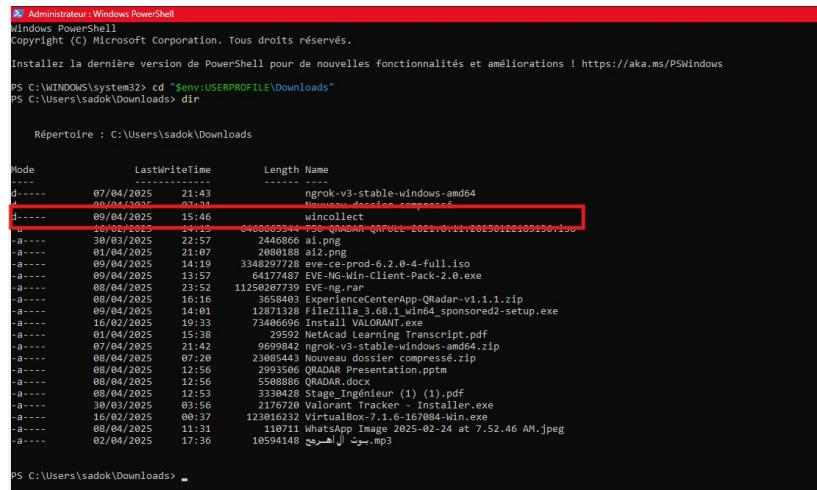
This project was carried out during an internship at Network Associates and focused on deploying a centralized log management solution using IBM QRadar SIEM. The project context lies in the need to identify, collect, correlate, and analyze security events generated by various devices and servers. Following a requirement analysis, a simulated network architecture was designed on the EVE-NG platform, including two sites interconnected via an IPsec VPN tunnel, each protected by a FortiGate firewall. Windows virtual machines were configured and integrated into QRadar as log sources. PowerShell scripts were developed to automate security state monitoring. The results demonstrated the effectiveness of event correlation in QRadar and a significant improvement in incident detection. This project paves the way for future enhancements, such as the integration of vulnerability scanners and automated response systems.

Webographie

La liste des sites web consultés durant la période du stage et durant la préparation du rapport :

- [1]. Téléchargement d'émulateur EVE, consulté le 15/03/2025
- [2]. Documentation EVE-NG, consulté le 18/03/2025
- [3]. Téléchargement d'outil WinSCP, consulté le 28/03/2025
- [4]. Documentation IBM QRadar, consulté le 01/04/2025
- [5]. Téléchargement IBM QRadar, consulté le 06/04/2025
- [6]. IBM WinCollect Agent, consulté le 10/04/2025
- [7]. Méthode RAD, consulté le 16/02/2025
- [8]. Fortinet Documentation Library, consulté le 21/04/2025
- [9]. Images IOS pour EVE-NG, consulté le 29/03/2025
- [10]. Téléchargement RAMmap Microsoft, consulté le 04/04/2025
- [12]. QRadar Architecture – Présentation SlideShare, consulté le 25/02/2025
- [11]. **Téléchargement Nmap Scanner** : <https://nmap.org/download>, consulté le 10/04/2025

Annexes



```
Administrator : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> cd "$env:USERPROFILE\Downloads"
PS C:\Users\sadok\Downloads> dir

Répertoire : C:\Users\sadok\Downloads

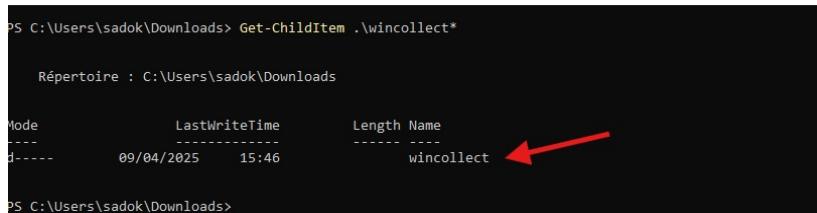
Mode                LastWriteTime       Length Name
----                -----        ---- 
d----
```

Mode	LastWriteTime	Length	Name
d----	07/04/2025 21:43		ngrok-v3-stable-windows-amd64
d----	09/04/2025 15:46		wincollect
a----	30/03/2025 22:57	2446866	ai.png
a----	01/04/2025 21:07	2880188	ai2.png
a----	09/04/2025 14:19	3348297788	eve-ce-prod-6.2.0-4-full.iso
a----	09/04/2025 14:57	6417648	EVE-Next-Client-Pack-2.0.exe
a----	09/04/2025 22:53	1125097739	img.png
a----	08/04/2025 16:16	3658493	ExperienceCenterApp-QRadar-v1.1.1.zip
a----	09/04/2025 14:01	12871328	filezilla 3.68.1 wind64_sponsored2-setup.exe
a----	16/02/2025 19:33	7340669	Install VALORANT.exe
a----	01/04/2025 15:38	29592	NetAcad Learning Transcript.pdf
a----	07/04/2025 07:29	9592	ngrok-v3-stable-windows-amd64.zip
a----	09/04/2025 07:20	23801443	nvuvuvv dossier compressé.zip
a----	09/04/2025 12:56	2993566	QRADAR Presentation.pptm
a----	08/04/2025 12:56	558888	QRADAR.docx
a----	08/04/2025 12:53	3330428	Stage_Ingenieur (1) (1).pdf
a----	30/03/2025 03:56	2176720	Valorant Tracker - Installer.exe
a----	16/02/2025 00:37	123010232	VirtualBox-7.1.6-1670884-Win.exe
a----	08/04/2025 11:31	118971	WhatsApp Image 2025-02-24 at 7.52.46 AM.jpeg
a----	02/04/2025 17:36	10594148	لـ.mp3

```
PS C:\Users\sadok\Downloads>
```

FIGURE 5.1 – Affichage du contenu du répertoire Downloads via PowerShell

Une commande PowerShell est utilisée pour afficher le contenu du dossier «Downloads» et s’assurer de la présence du fichier.



```
PS C:\Users\sadok\Downloads> Get-ChildItem .\wincollect*
```

Mode	LastWriteTime	Length	Name
d----	09/04/2025 15:46		wincollect

```
PS C:\Users\sadok\Downloads>
```

FIGURE 5.2 – Vérification de la présence du fichier d’installation

Cette étape confirme la présence du fichier exécutable de WinCollect avant de procéder à l’installation.



```
PS C:\Users\sadok\Downloads> cd .\wincollect
PS C:\Users\sadok\Downloads\wincollect> Get-ChildItem

Répertoire : C:\Users\sadok\Downloads\wincollect

Mode                LastWriteTime       Length Name
----                -----        ---- 
-a----
```

Mode	LastWriteTime	Length	Name
a----	08/04/2025 14:14	8314880	wincollect-10.1.13-12.x64.msi
a----	09/04/2025 15:46	126	wincollect-10.1.13-12.x64.sha256

```
PS C:\Users\sadok\Downloads\wincollect>
```

FIGURE 5.3 – Contenu du dossier d’installation

Le répertoire d’installation contient les différents fichiers nécessaires au bon déroulement de l’installation.

```
PS C:\Users\sadok\Downloads\wincollect> Get-FileHash -Algorithm SHA256 .\wincollect-10.1.13-12.x64.msi
Algorithm      Hash
-----
SHA256        21FCF2CD92A569550AC237252EC7B06E821D30ACBDB708EBC92713F5BD501D6E
Path          C:\Users\sadok\Downloads\wincollect\wincollect-10.1.13-12.x64.msi

PS C:\Users\sadok\Downloads\wincollect> Get-Content .\wincollect-10.1.13-12.x64.sha256
Algorithm      Hash
-----
SHA256        21fcf2cd92a569550ac237252ec7b06e821d30acb708ebc92713f5bd501d6e
PS C:\Users\sadok\Downloads\wincollect>
```

FIGURE 5.4 – Vérification de l'intégrité du fichier d'installation à l'aide de SHA256

Une commande de hachage est utilisée pour générer et vérifier la signature SHA256 du fichier téléchargé, assurant ainsi son authenticité.