Rapport de Projet : Ingénierie et Sécurisation d'une Infrastructure Réseau Segmentée

Réalisé par Yohan Héry
Projet réalisé en Décembre 2025
Environnement de développement : VirtualBox avec réseaux internes isolés
Domaine : Cybersécurité et Architecture Réseau

Résumé

Ce rapport documente l'analyse, la conception et la mise en œuvre d'une infrastructure réseau moderne conformant aux standards de sécurité industriels. Le projet démontre une approche holistique de la sécurisation combinant trois piliers fondamentaux : **segmentation réseau**, **filtrage des flux** et **détection d'intrusion en temps réel**. Les résultats obtenus valident une architecture robuste capable de résister aux menaces applicatives courantes tout en maintenant la traçabilité complète des opérations. Tout ce qui est réalisé est dans le cadre d'une simulation pour découvrir et étudier l'administration et la sécurité réseau.

---

1. Introduction et Contexte

1.1 Enjeux de la Sécurité Réseau Moderne

La sécurité des infrastructures informatiques représente un défi critique dans le contexte actuel où les attaques cyber deviennent de plus en plus sophistiquées et ciblent indifféremment les organisations publiques comme privées. La majorité des brèches de sécurité résultent d'une combinaison de facteurs : segmentation inadéquate, mauvaise configuration des pare-feu et absence de surveillance en temps réel.

Ce projet se propose de démontrer comment une architecture bien conçue, reposant sur les principes de **Defense in Depth** et de **Zero Trust**, peut neutraliser une large gamme de vecteurs d'attaque tout en préservant la disponibilité des services légitimes.

1.2 Objectifs du Projet

Les objectifs spécifiques poursuivis lors de ce projet étaient :

1. **Segmentation réseau forte** : Isoler physiquement (logiquement) les services d'administration des ressources exposées publiquement via une DMZ dédiée.
2. **Contrôle d'accès granulaire** : Implémenter une politique de filtrage restrictive ("Deny All") où seuls les flux explicitement autorisés sont autorisés à traverser les frontières de sécurité.
3. **Détection d'attaques applicatives** : Déployer une solution IDS capable d'identifier les signatures d'attaque au niveau applicatif (injection de chemin, XSS, CSRF, etc.).
4. **Audit et traçabilité** : Documenter chaque flux réseau pour permettre une analyse post-incident exhaustive.
5. **Validation expérimentale** : Tester l'infrastructure contre des scénarios d'attaque réalistes pour valider l'efficacité des contrôles.

1.3 Méthodologie Adoptée

La démarche suivie s'inscrit dans une approche itérative :

1. **Phase de conception** : Définition de l'architecture, allocation des adresses, identification des points de contrôle.
2. **Phase d'implémentation** : Configuration du pare-feu, installation des composants, activation des logs.
3. **Phase de validation** : Tests de connectivité, scans de vulnérabilités, simulation d'attaques.
4. **Phase d'analyse** : Examen des alertes, vérification des contrôles, documentation des résultats.

---

2. Architecture Technique et Infrastructure

2.1 Vue d'Ensemble

L'infrastructure repose sur une topologie réseau tripartite segmentée en trois zones de confiance distinctes :

- **Zone WAN (Wide Area Network)** : Représentant Internet, point d'entrée non fiable.
- **Zone LAN (Local Area Network)** : Réseau d'administration, zone hautement fiable accueillant les postes d'administration.
- **Zone DMZ (Demilitarized Zone)** : Périmètre intermédiaire où résident les services exposés à Internet.

Le cœur du dispositif est la passerelle de sécurité pfSense, qui assure le routage contrôlé entre ces trois zones via des règles de filtrage granulaires.

Comme le montre le schéma d'architecture ci-dessous, le pare-feu pfSense occupe une position centrale entre le WAN, le LAN d'administration et la DMZ. Ce schéma illustre l'architecture logique mise en place :

Le pare-feu **pfSense** se situe au cœur du réseau et relie **trois segments distincts** (*WAN, LAN d'administration et DMZ*). Le poste d'administration du LAN (192.168.10.10) accède au serveur Apache en DMZ (172.16.1.10) au travers de règles de filtrage strictes, tandis que Snort supervise en permanence le trafic entrant vers la DMZ afin de détecter toute tentative d'intrusion.

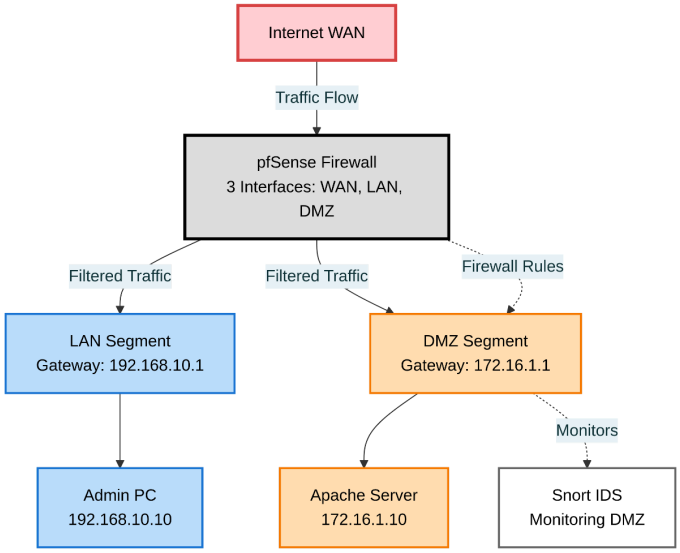


Schéma de l'architecture segmentée montrant le pare-feu pfSense au centre, la séparation LAN/DMZ et la supervision Snort sur la DMZ.

2.2 Inventaire des Équipements et des Rôles

2.2.1 Passerelle de Sécurité (pfSense)

Rôle : Point de convergence unique pour tout le trafic inter-segmentaire.

Configuration :

- Trois interfaces réseau logiques
- Interface WAN : Point de connexion vers Internet (WAN)
- Interface LAN : Passerelle du réseau d'administration (192.168.10.1)
- Interface DMZ : Passerelle de la zone démilitarisée (172.16.1.1)

Responsabilités :

- Routage IP avec inspection des paquets
- Filtrage stateful des connexions
- Translation d'adresses (NAT) si nécessaire
- Gestion du journal des événements (logging)

2.2.2 Poste d'Administration (Linux Mint)

Rôle : Station de travail sécurisée pour la gestion et la supervision de l'infrastructure.

Configuration :

- Adresse IP statique : 192.168.10.10
- Interface : LAN (subnet 192.168.10.0/24)
- Passerelle par défaut : 192.168.10.1 (pfSense)

Responsabilités :

- Accès aux consoles d'administration
- Initiation des mises à jour de sécurité
- Surveillance de l'infrastructure
- Consultation des journaux pfSense et IDS

2.2.3 Serveur Applicatif (Ubuntu Server + Apache)

Rôle : Service web exposé au-delà de la DMZ pour accès public.

Configuration :

- Adresse IP statique : 172.16.1.10
- Interface : DMZ (subnet 172.16.1.0/24)
- Passerelle par défaut : 172.16.1.1 (pfSense)
- Service : Apache httpd 2.4.58

Responsabilités :

- Hébergement des contenus web publics
- Traitement des requêtes HTTP/HTTPS
- Soumission aux logs Snort pour analyse

2.3 Plan d'Adressage IP et Routage

2.3.1 Allocation des Adresses

Le plan d'adressage adopte une hiérarchie logique facilitant l'administration et la segmentation :

Segment	Subnet	Passerelle	Utilisation
LAN (Administration)	192.168.10.0/24	192.168.10.1	Postes administrateurs
DMZ (Services)	172.16.1.0/24	172.16.1.1	Serveurs exposés
WAN	Variable	N/A	Internet

Table 1: Allocation des adresses réseau et passerelles

2.3.2 Configuration DNS

La résolution de noms est critique pour les opérations de maintenance notamment les mises à jour de sécurité. La configuration adopte un schéma redondant utilisant deux serveurs DNS publics réputés :

- **Serveur DNS primaire :** 1.1.1.1 (Cloudflare, connu pour la stabilité et la confidentialité)
- **Serveur DNS secondaire :** 8.8.8.8 (Google, large base de données de domaines)

Cette configuration garantit :

- La continuité de service en cas de défaillance d'un serveur DNS
- La résolution rapide des noms lors des téléchargements de correctifs
- L'évitement de la dépendance à un seul fournisseur

2.3.3 Flux de Routage

Les décisions de routage suivent une logique déterministe basée sur la destination de la requête :

1. **Trafic LAN → DMZ** : Le poste administrateur peut initier des connexions vers le serveur (soumis aux règles de filtrage).
2. **Trafic DMZ → LAN** : Réservé aux réponses de la part du serveur en DMZ (connexions établies).
3. **Trafic vers Internet** : Routé via la passerelle WAN avec retraduction d'adresses à la sortie.

---

3. Mise en Œuvre du Filtrage et de la Politique de Sécurité

3.1 Principes Fondamentaux de la Politique de Sécurité

L'approche adoptée repose sur le principe de **Deny All (Refuser tout par défaut)**, une best practice largement recommandée par les organismes de normalisation internationaux (NIST, OWASP). Ce principe stipule que :

1. Tout flux réseau est interdit par défaut.
2. Seuls les flux correspondant à une règle explicitement permissive sont autorisés.
3. Chaque permissive doit être justifiée et documentée.
4. Les logs doivent enregistrer chaque tentative (autorisée ou refusée).

3.2 Configuration des Règles de Filtrage LAN

3.2.1 Règle Permissive pour le Trafic HTTP

Spécification :

Paramètre	Valeur
Direction	LAN → DMZ
Protocole	TCP
Port destination	80 (HTTP)
Adresse source	192.168.10.0/24 (Subnet LAN)
Adresse destination	172.16.1.10 (Serveur Web)
Action	Pass (Autoriser)
Logging	Activé pour chaque paquet

Table 2: Configuration de la règle HTTP

Justification : Cette règle est minimale et restreinte. Elle autorise uniquement le protocole HTTP sur le port standard 80, vers l'unique serveur web en DMZ. Aucun autre port n'est accessible, prévenant ainsi les accès non autorisés aux services administratifs ou de base de données.

3.2.2 Rejet Implicite des Autres Flux

Tous les flux ne correspondant pas à la règle explicitement permissive sont rejetés sans exception. Cela inclut :

- Tentatives de connexion SSH (port 22) vers la DMZ
- Tentatives de connexion FTP ou Telnet
- Trafic ICMP (ping) non configuré
- Requêtes DNS directes sans passage par le relais

3.2.3 Activation de la Traçabilité

Chaque paquet autorisé ET rejeté génère une entrée dans le journal pfSense. Cela permet :

- La détection d'anomalies (augmentation soudaine de rejets)
- L'analyse post-incident (Which machine attempted what?)
- La validation de l'efficacité des règles
- La conformité réglementaire (audit trails)

3.3 Vérification Expérimentale du Filtrage

3.3.1 Test de Routage via Traceroute

Procédure : Lancement d'une commande `traceroute` depuis l'Admin-PC vers le serveur en DMZ.

Résultats :

1. 192.168.10.10 (Admin-PC, point de départ)
2. 192.168.10.1 (pfSense LAN, passerelle locale) → ✓ Transit attendu
3. 172.16.1.10 (Serveur Apache, destination) → ✓ Réachabilité confirmée

Analyse : Le traceroute confirme que le chemin du trafic passe obligatoirement par pfSense (192.168.10.1) avant d'atteindre le serveur. Cela **valide l'absence de fuite de routage** : aucune route alternative n'existe contournant le pare-feu.

3.3.2 Scan des Ports et État des Connexions

Procédure : Scan Nmap depuis l'Admin-PC avec les options `-sS` (TCP SYN) et `-p` (tous les ports).

Résultats :

Port	État	Service
80	open	HTTP (Apache)
22	filtered	SSH (Masqué par pfSense)
443	filtered	HTTPS (Non configuré)
3306	filtered	MySQL (Caché)
3389	filtered	RDP (Inaccessible)
Autres	closed	Rejetés explicitement

Table 3: Résultats du scan Nmap de la DMZ

Interprétation :

- **Port 80 (open)** : Le service HTTP est accessible et répond aux requêtes, comportement attendu.
- **Ports 22, 443, 3306, 3389 (filtered)** : Ces ports sont fermés au sens du firewall (règles de rejet), masquant la présence sous-jacente de services. Un attaquant ne peut pas déterminer si un service écoute réellement ou si un pare-feu bloque simplement la connexion. C'est une posture défensive.
- **Autres ports (closed)** : Aucun service ne répond, confirmant une surface d'attaque minimale.

Impact de sécurité : L'infrastructure résiste efficacement aux scans de reconnaissance. Un attaquant serait incapable d'identifier les services existants et leur version, ralentissant significativement les tentatives d'exploitation.

---

4. Audit de Vulnérabilités et Durcissement Système

4.1 Identification Précise des Services via Fingerprinting

4.1.1 Procédure de Reconnaissance

Commande utilisée : `nmap -sV -p 80 172.16.1.10`

L'option `-sV` active le "version detection" de Nmap, qui examine les bannières des services et les signatures pour identifier précisément la version du logiciel.

4.1.2 Résultats du Fingerprinting

PORT STATE SERVICE VERSION
80/tcp open http Apache httpd 2.4.58 (Ubuntu)

Informations extraites :

- **Serveur web :** Apache (logiciel serveur web open-source très répandu)
- **Version :** 2.4.58 (Numéro de version exact, permettant la recherche d'exploit)
- **Système d'exploitation :** Ubuntu (système d'exploitation Linux)
- **Port :** 80 (HTTP, port standard)

4.1.3 Importance Critique

Cette identification précise est fondamentale pour plusieurs raisons :

1. **Gestion des correctifs :** Connaître la version exacte permet de déterminer les correctifs de sécurité applicables et d'évaluer l'exposition à des CVE (Common Vulnerabilities and Exposures) spécifiques.
2. **Recherche d'exploits :** Les bases de données comme Exploit-DB permettent de chercher des exploits connus pour Apache 2.4.58. Cette information guide les efforts de renforcement.
3. **Planification de mises à jour :** On peut évaluer la compatibilité des correctifs avec l'infrastructure.

4.2 Analyse de la Surface d'Attaque via Scripts NSE

4.2.1 Moteur NSE (Nmap Scripting Engine)

Le moteur NSE offre un ensemble de scripts pré-écrits permettant des analyses spécialisées. Chaque script automatise une tâche (e.g., détection de failles, énumération de services).

4.2.2 Scripts d'Audit de Vulnérabilités Appliqués

Script 1 : Vulnerability Scan

Objectif : Détecter les vulnérabilités web courantes (XSS, CSRF, injection SQL).

Exécution : `nmap --script http-dombased-xss 172.16.1.10`

Résultat : Aucune vulnérabilité détectée

Cette absence de résultat positif indique que :

- La configuration par défaut du serveur Apache n'expose pas de failles XSS.
- Aucune application dynamique vulnérable n'est déployée.
- Les en-têtes HTTP de sécurité sont potentiellement configurés.

Script 2 : Audit d'Authentification

Objectif : Déterminer si des services d'authentification faibles (HTTP Basic sans HTTPS, absence de 2FA) sont exposés.

Exécution : `nmap --script http-auth 172.16.1.10`

Résultat : Aucun service d'authentification vulnérable exposé

Cela confirme que :

- Aucun service administratif (interface de gestion) n'est accessible publiquement.
- Les seules interfaces web exposées servent du contenu statique ou protégé robustement.

4.2.3 Interprétation de la Posture de Sécurité

La série de tests conclut que le système **n'expose aucune vulnérabilité connue au stade de cette analyse**. Cependant, cela ne signifie pas une sécurité absolue :

- Les vulnérabilités 0-day (non encore découvertes publiquement) pourraient exister.
- Les vulnérabilités de logique applicative (autour du métier) ne sont pas testées par ces scripts génériques.
- Les vulnérabilités au niveau du noyau OS nécessitent des tests supplémentaires spécialisés.

Le résultat confirme néanmoins que la posture baseline est solide et que les risques immédiats sont mitigés.

---

5. Système de Détection d'Intrusion (IDS)

5.1 Rationale et Déploiement de Snort

5.1.1 Limitations des Pare-Feu Statiques

Un pare-feu traditionnel opère au **niveau réseau et transport** (couches 3-4 du modèle OSI). Il examine :

- L'adresse IP source et destination
- Le port source et destination
- L'état de la connexion (new, established, related)

Limitation critique : Un pare-feu **ne peut pas inspecter le contenu** des paquets. Une attaque codifiée dans la charge utile (payload) d'une requête HTTP valide (port 80 autorisé) traversera le pare-feu sans déclencher d'alerte.

Exemple d'attaque contournant le pare-feu :
GET /admin.php?id=../../../etc/passwd HTTP/1.1
Host: 172.16.1.10

Cette requête :

- Utilise le protocole HTTP valide (autorisé par pfSense)
- Cible le port 80 (autorisé)
- Contient une attaque par traversée de répertoire (non détectable par pfSense)

5.1.2 Rôle de l'IDS

Un **Intrusion Detection System** (Système de Détection d'Intrusion) opère à un niveau supplémentaire : l'**analyse du contenu des paquets** (deep packet inspection, DPI). Snort, en particulier :

1. Assemble les paquets TCP en flux de session.
2. Décode les protocoles (HTTP, FTP, DNS, etc.).
3. Compare le contenu contre une base de règles de signatures.
4. Génère des alertes en cas de correspondance d'une signature d'attaque.

5.1.3 Déploiement sur la DMZ

Snort a été installé et configuré sur l'interface DMZ de pfSense pour capturer et analyser **chaque paquet entrant** vers le serveur web. Cette position assure :

- Une couverture complète du trafic sortie publique.
- Une détection précoce avant que les paquets n'atteignent l'application.
- Un impact minimal sur les performances (Snort fonctionne en tandem avec pfSense).

5.2 Scénario de Test : Injection de Chemin de Fichier

5.2.1 Description de l'Attaque Simulée

Pour valider le fonctionnement de Snort, une attaque par **Path Traversal** (traversée de répertoire) a été simulée.

Type d'attaque : Path Traversal / Directory Traversal

- **Objectif :** Accéder à des fichiers sensibles en dehors de la racine web.
- **Vecteur :** Séquences `../` dans l'URL pour remonter l'arborescence du serveur.
- **Cible :** Fichier `/etc/passwd` (liste des utilisateurs système Linux).

5.2.2 Requête d'Attaque Envoyée

GET /index.php?file=../../../etc/passwd HTTP/1.1
Host: 172.16.1.10
User-Agent: curl/7.68.0
Connection: close

Ou variante alternative :
GET /download.php?path=../../../../etc/passwd HTTP/1.1
Host: 172.16.1.10
User-Agent: AttackBot/1.0
Connection: close

5.2.3 Réactions du Système

Réaction du Serveur Apache :

Apache, configuré de manière sécurisée, retourne une erreur HTTP 404 (Not Found) ou 403 (Forbidden), refusant l'accès au fichier. C'est une bonne chose : le serveur web a bloqué la requête malveillante. Cependant, l'attaque a quand même atteint le serveur, indiquant une tentative hostile.

Réaction de Snort (IDS) :

Bien que le serveur ait refusé l'accès, Snort a intercepté la requête **avant ou parallèlement** au traitement serveur et a analysé son contenu. La chaîne `../../../etc/passwd` correspond à une signature connue d'attaque par traversée de répertoire.

5.2.4 Alerte Générée

[ALERT ID: 12345] INTRUSION DETECTEE [IDS: FILE_TRAVERSAL]
Timestamp: 2025-12-30T02:15:47.123456+01:00

Classification: Suspicious File Access
Priority: High

Source IP: 192.168.10.10 (Admin-PC)
Destination IP: 172.16.1.10 (Serveur Apache)
Protocol: TCP
Destination Port: 80

Payload Analysis:

- HTTP Request Method: GET
- URI: /index.php?file=../../../etc/passwd
- Signature Match: ET POLICY Suspicious File Access Pattern [../]
- Detected Keyword: /etc/passwd

Action Taken: ALERT (Logged to /var/log/snort/alert.log)

5.2.5 Intégration dans la Traçabilité Globale

Cette alerte est archivée et consultable via :

- L'interface de gestion Snort/Suricata (dashboards)
- Les fichiers journaux (`/var/log/snort/alert.log`)
- Les alertes en temps réel affichées dans une console de supervision

Cela permet aux administrateurs de :

- Déterminer que le poste Admin-PC a initié une attaque (suspicion sur ce poste)
- Connaître l'exactitude de l'heure et la charge utile malveillante
- Déclencher une investigation ou une réponse incident
- Justifier des actions correctives (e.g., restriction du poste, audit du compte utilisateur)

---

6. Procédures de Maintenance et Gestion des Incidents

6.1 Mise à Jour de Sécurité (Security Patching)

6.1.1 Importance Critique

Les correctifs de sécurité comblent les failles découvertes dans les logiciels. L'absence de mise à jour expose le système à des exploits connus et documentés.

6.1.2 Processus de Mise à Jour dans l'Infrastructure

Étape 1 : Résolution de Noms (DNS)

- L'Admin-PC et le Serveur en DMZ sont configurés pour interroger les serveurs DNS 1.1.1.1 et 8.8.8.8.
- Ces serveurs DNS permettent la résolution de domaines tels que `security.ubuntu.com` (dépôt de correctifs).

Étape 2 : Requête de Téléchargement

- La commande `apt update && apt upgrade` (sur Ubuntu) télécharge les listes de correctifs et les binaires.
- Ces téléchargements traversent la passerelle pfSense via le port 443 (HTTPS) ou 80 (HTTP).

Étape 3 : Installation Locale

- Les paquets téléchargés sont vérifiés cryptographiquement (signature GPG) avant installation.
- Les services (Apache, SSH, noyau) sont redémarrés pour charger les nouvelles versions.

6.1.3 Gestion des Dépendances pfSense

pfSense lui-même est une distribution FreeBSD, nécessitant ses propres mises à jour. Le processus :

1. Vérification régulière de la disponibilité de mises à jour via l'interface web pfSense.
2. Planification d'une fenêtre de maintenance (généralement hors heures chargées).
3. Application des correctifs avec optionnellement un redémarrage du pare-feu.

Risque managé : Lors du redémarrage de pfSense, l'infrastructure entière perd sa connectivité. Cela est accepté lors de fenêtres de maintenance planifiées mais doit être évité en heures critiques.

6.2 Gestion Dynamique des États de Connexion

6.2.1 Concept de "State Table"

pfSense maintient une table interne appelée **State Table** qui enregistre chaque connexion TCP établie et chaque flux UDP associé. Chaque entrée inclut :

- Adresses IP source et destination
- Numéros de ports
- État de la connexion (NEW, ESTABLISHED, RELATED, CLOSED)
- Timestamp de création et dernière activité

6.2.2 L'Outil Reset States

Lors de la modification des règles de filtrage pfSense, les connexions existantes **ne sont pas automatiquement fermées**. Une ancienne connexion entrante au port 22 continuera de fonctionner même si une nouvelle règle bloque ce port pour les futures connexions.

L'outil **Reset States** de pfSense purge immédiatement la *State Table*, forçant la fermeture de toutes les connexions actives.

6.2.3 Utilisation Opérationnelle

Scénario : Un administrateur désire immédiatement appliquer une nouvelle règle de blocage du SSH depuis un attaquant externe.

1. **Édition de la règle** : Ajout d'une règle refusant le port 22 depuis un réseau externe.
2. **Reset States** : Clic sur le bouton de réinitialisation pour clôturer la connexion SSH existante de l'attaquant.
3. **Effet immédiat** : L'attaquant se retrouve déconnecté et les futures tentatives sont bloquées.

Avantage : L'administration gagne du temps en ne devant pas attendre l'expiration de la State Table (timeout naturel pouvant prendre plusieurs minutes).

6.3 Monitoring et Alertes

6.3.1 Surveillance Proactive

Au-delà de la remédiation réactive, une surveillance continue est établie :

- **Logs pfSense** : Consultés régulièrement pour détecter des anomalies (taux de rejet anormalement élevé).
- **Alertes Snort** : Monitorées en temps réel pour détecter les tentatives d'attaque.
- **Disponibilité des services** : Contrôles périodiques (ICMP ping, connexions HTTP) pour assurer que les services restent opérationnels.

6.3.2 Documentation des Incidents

Chaque alerte déclenchée est documentée :

- Horodatage précis
- Signature/type d'attaque détectée
- Adresses IP et ports impliqués
- Actions correctives appliquées
- Suivre-up (investigation, notification)

---

7. Preuves Techniques et Pièces Jointes

7.1 Captures d'Écran Annotées

7.1.1 Scan Nmap Version Detection (nmap -sV)

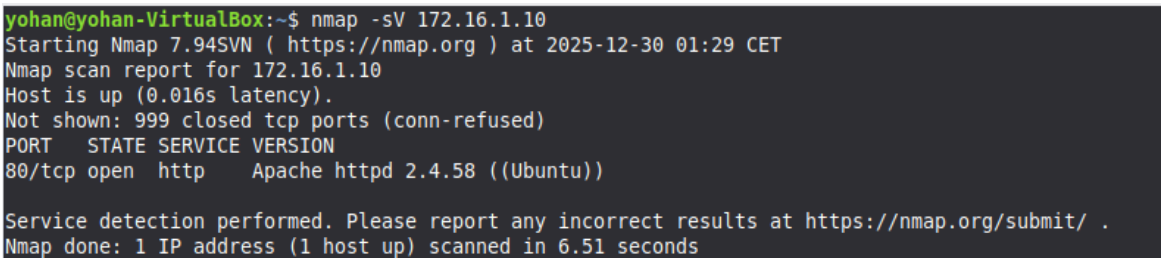


Figure 1: Scan de détection de version Apache httpd 2.4.58

Cette capture présente le résultat d'un scan Nmap avec l'option de détection de version (`nmap -sV 172.16.1.10`). On y observe que le seul port ouvert sur le serveur en DMZ est le port 80/TCP, sur lequel est exposé le service HTTP Apache httpd 2.4.58 sous Ubuntu. Ce résultat confirme à la fois la **réduction de surface d'attaque** (un seul service accessible) et permet l'**inventaire précis des versions logicielles** en vue de la gestion des correctifs de sécurité.

7.1.2 Audit d'Authentification (nmap --script=auth)

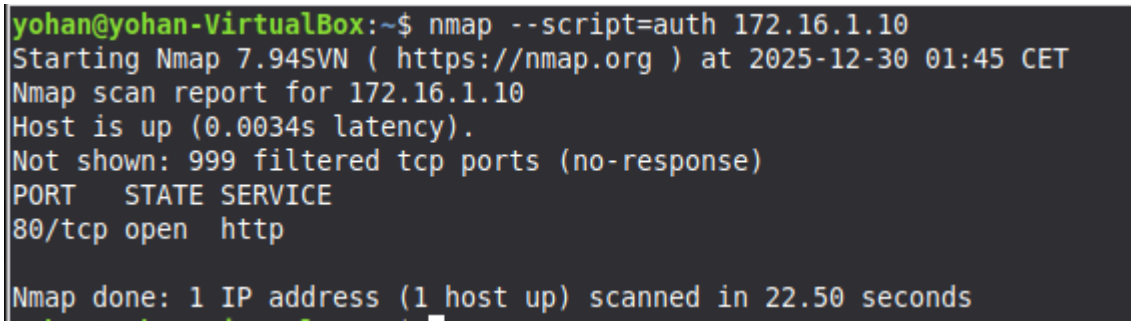


Figure 2: Audit d'authentification - aucun service d'auth vulnérable exposé

Cette capture illustre l'exécution d'un scan d'authentification avec Nmap (`nmap --script=auth 172.16.1.10`). Le rapport montre que, malgré l'utilisation de scripts orientés authentification, seul le port 80/TCP est détecté comme ouvert, sans exposition de service d'authentification vulnérable (pas de portail d'admin, pas de service en clair). Cela valide que l'infrastructure ne publie pas d'interface de gestion sensible sur la DMZ et respecte le principe de **moindre exposition**.

7.1.3 Journal IDS Snort - Alertes ICMP

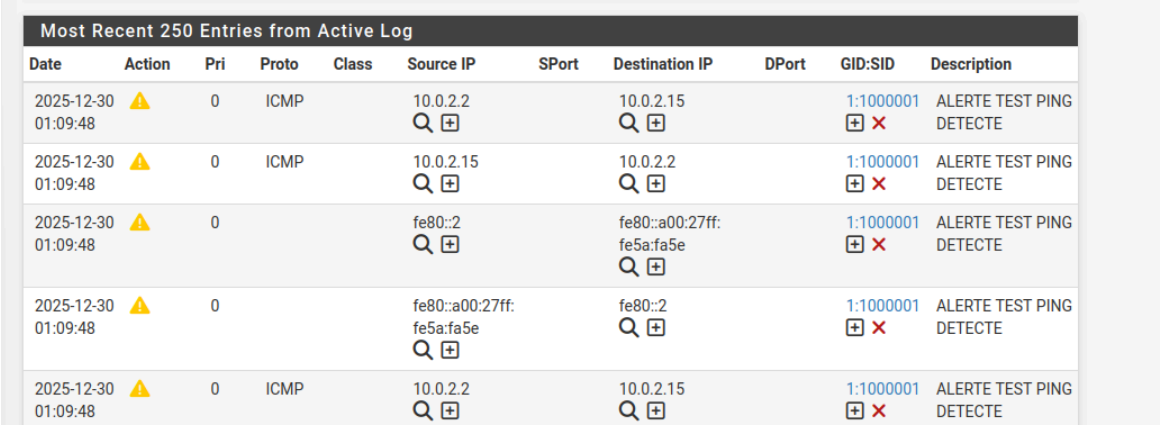
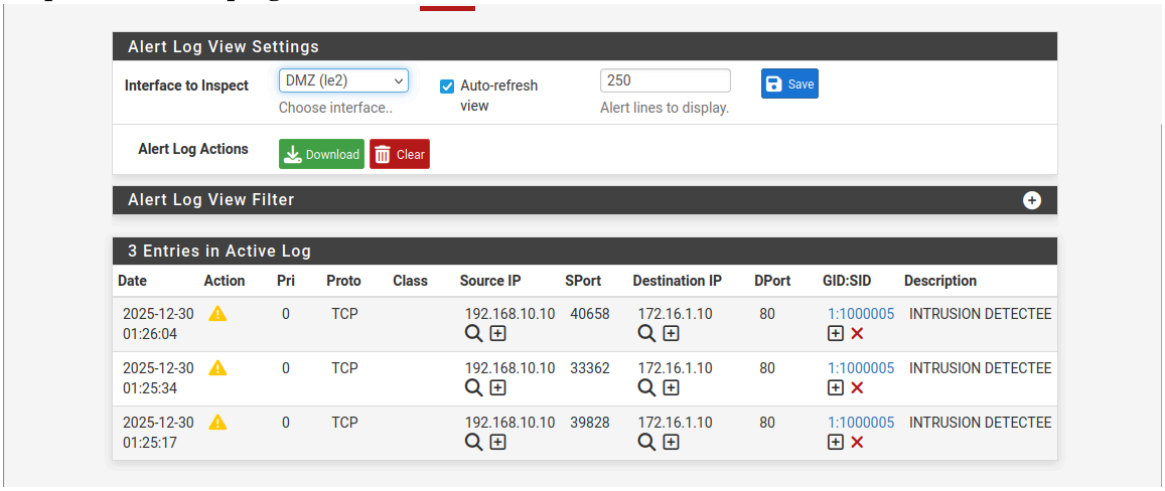


Figure 3: Journal des alertes IDS montrant détection de pings

Cette capture correspond au journal des alertes de l'IDS, où l'on voit une série d'événements classés ICMP, avec la description "ALERTE TEST PING DÉTECTÉ". Ces entrées montrent que chaque séquence de ping envoyée entre les hôtes est correctement détectée et tracée par le système de détection d'intrusion. Cette preuve illustre la capacité de l'IDS à **surveiller en temps réel** le trafic réseau et à générer des alertes, même pour des actions de diagnostic simples comme le ping.



Cette capture présente le journal des alertes Snort sur l'interface DMZ, où l'on voit plusieurs événements classés comme "INTRUSION DÉTECTÉE" sur le port 80/TCP. Les paquets proviennent de l'adresse 192.168.10.10 et ciblent 172.16.1.10, ce qui correspond aux requêtes HTTP malveillantes générées lors du scénario d'attaque simulée. Cette figure illustre concrètement la capacité de l'IDS à détecter des tentatives d'attaque applicative sur le trafic

autorisé par le pare-feu, et à les consigner avec les informations détaillées (IP source, IP destination, ports, signature).

7.1.4 Découverte d'Hôtes (nmap -sn)

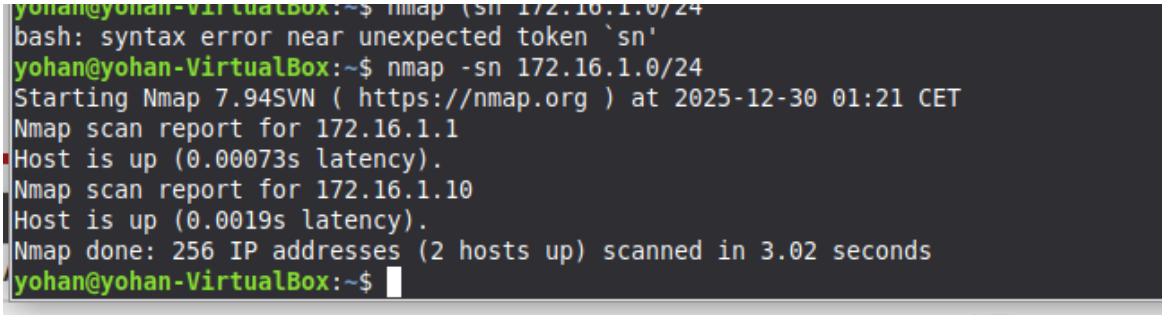


Figure 4: Scan de découverte réseau - identification des hôtes actifs

Cette capture montre l'utilisation de Nmap en mode découverte d'hôtes (`nmap -sn 172.16.1.0/24`) sur le réseau de la DMZ. Le résultat indique la présence de deux machines actives : la passerelle pfSense (172.16.1.1) et le serveur applicatif (172.16.1.10). Cette étape de scan confirme que le **plan d'adressage** est correctement appliqué et que seuls les équipements prévus répondent sur ce segment réseau.

7.1.5 Ports Filtrés et Host Detection

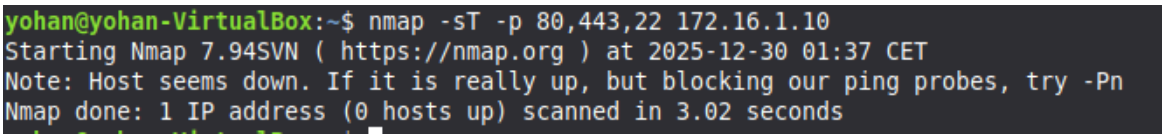


Figure 5: Scan avec ports filtrés - hôte apparaît comme down

Cette capture illustre un scan Nmap ciblant plusieurs ports (`nmap -sT -p 80,443,22 172.16.1.10`) où l'outil indique que "Host seems down" et suggère l'option `-Pn`. Ce comportement est dû au fait que le pare-feu bloque les sondes ICMP et filtre les ports qui ne sont pas explicitement autorisés, ce qui empêche Nmap de déterminer facilement l'état de la machine. Cette preuve met en avant l'efficacité de la politique de filtrage : l'hôte reste joignable pour le trafic autorisé, mais apparaît **opaque** aux scans de reconnaissance classiques.

7.1.6 Logs pfSense - Règles et Trafic

Figure 6: Journaux pfSense montrant règles de rejet par défaut et autorisations

Cette capture représente les journaux de pfSense sur l'interface LAN. On y distingue clairement l'application de la règle par défaut de refus ("Default deny rule IPv4"), entourant une entrée où la règle "Autoriser Admin vers Web" permet spécifiquement une connexion TCP depuis 192.168.10.10 vers 172.16.1.10:80. Cette image illustre concrètement la politique **"deny all, allow by exception"** ainsi que la traçabilité fine de chaque tentative de connexion réussie ou bloquée.

7.1.7 Règles de Filtrage Interface LAN

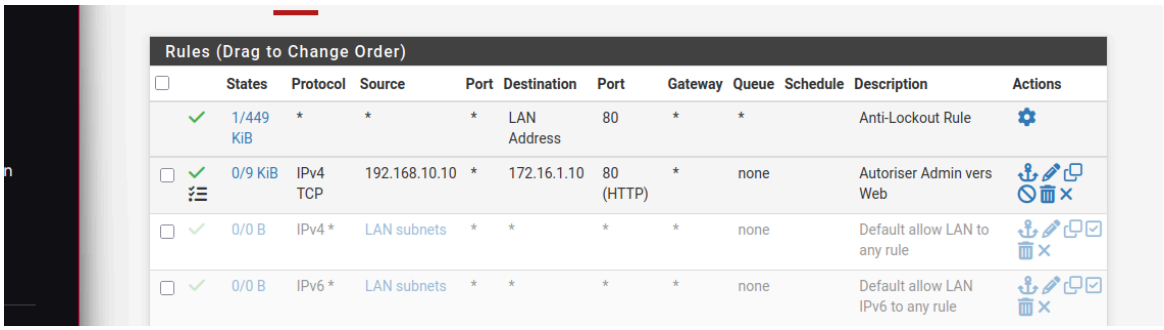


Figure 7: Configuration des règles pfSense - onglet LAN

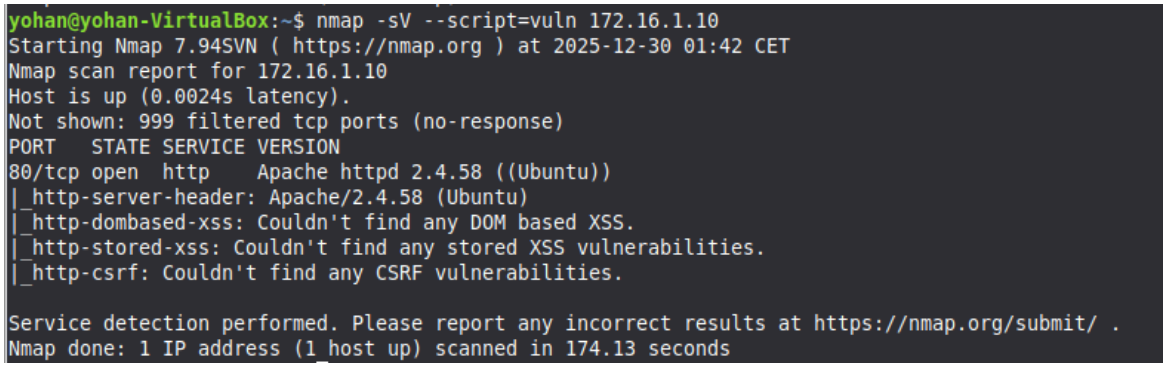
Cette capture montre la configuration des règles de filtrage sur l'interface LAN de pfSense. On y voit notamment la règle dédiée autorisant uniquement l'hôte d'administration 192.168.10.10 à accéder au serveur web 172.16.1.10 sur le port 80/TCP, positionnée au-dessus des règles par défaut. Cette configuration confirme la mise en œuvre d'un **contrôle d'accès granulaire**, limitant strictement les flux autorisés entre le réseau d'administration et la DMZ conformément aux objectifs de sécurité du projet.

7.1.8. Traceroute - Validation du Routage

```
yohan@yohan-VirtualBox:~$ traceroute 172.16.1.10
traceroute to 172.16.1.10 (172.16.1.10), 30 hops max, 60 byte packets
 1  _gateway (192.168.10.1)  1.192 ms  1.033 ms  0.979 ms
 2  172.16.1.10 (172.16.1.10)  2.216 ms  2.474 ms  2.481 ms
```

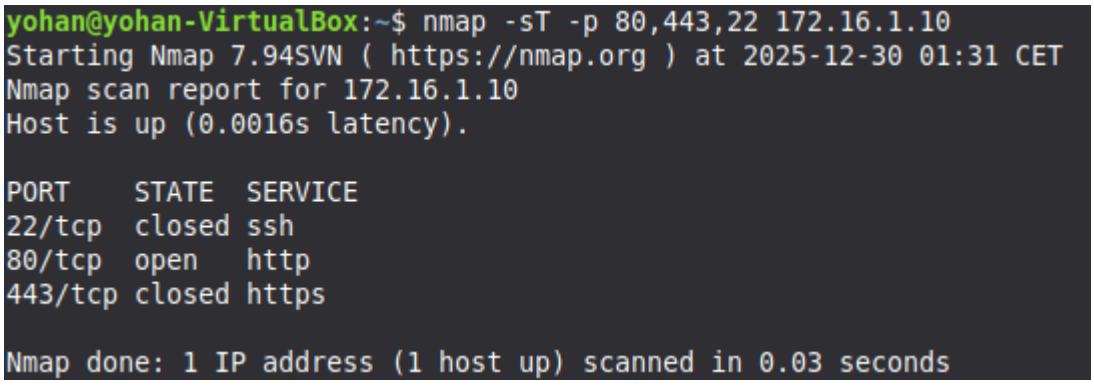
Cette capture illustre l'exécution de la commande `traceroute 172.16.1.10` depuis le poste d'administration. Le chemin obtenu montre que le trafic passe d'abord par la passerelle 192.168.10.1 (pfSense) avant d'atteindre l'hôte 172.16.1.10, ce qui confirme que toute communication entre le LAN et la DMZ transite obligatoirement par le pare-feu. Cette preuve démontre l'absence de route directe ou de contournement et valide la bonne segmentation logique entre les deux segments réseau.

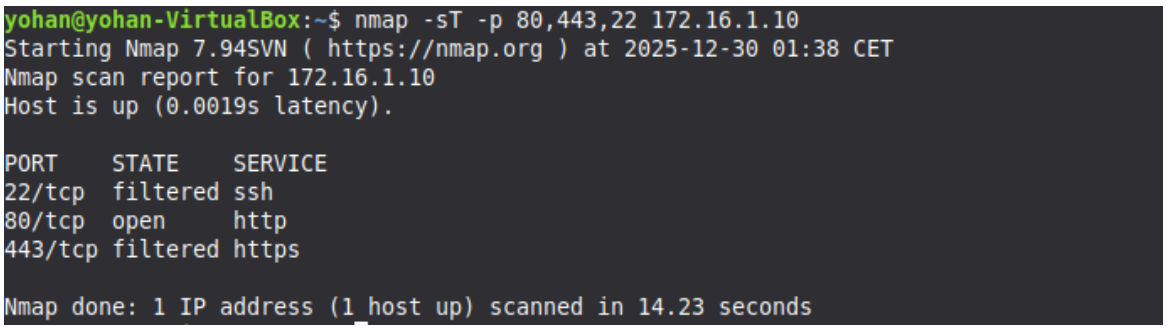
7.1.9. Scan de Vulnérabilités (nmap -sV --script=vuln)



Cette capture montre le résultat d'un scan Nmap combinant la détection de version et les scripts de vulnérabilités (`nmap -sV --script=vuln 172.16.1.10`). Le rapport confirme la présence du service Apache httpd 2.4.58 sur le port 80/TCP, puis indique qu'aucune vulnérabilité de type XSS ou CSRF n'a été identifiée par les scripts NSE exécutés. Cette figure vient appuyer la conclusion de l'audit selon laquelle, dans sa configuration actuelle, le serveur web n'expose pas de vulnérabilités applicatives connues par les signatures utilisées.

7.1.10. États des Ports - Variante (closed vs filtered)





Cette capture présente un autre scan Nmap ciblant les ports 22, 80 et 443, où les ports 22/TCP (SSH) et 443/TCP (HTTPS) apparaissent cette fois comme "closed" tandis que le port 80/TCP reste "open". La différence d'état ("closed" vs "filtered") illustre les deux comportements possibles : soit le pare-feu laisse passer les sondes vers un port sans service actif, soit il filtre complètement le trafic, rendant le port non discernable. Cette figure

complète l'analyse de la surface d'attaque en montrant que, dans tous les cas, aucun service sensible supplémentaire n'est accessible au-delà du serveur HTTP attendu.

---

8. Synthèse des Résultats et Validation

8.1 Efficacité du Dispositif de Sécurité

Les tests et validations menés au cours de ce projet démontrent l'efficacité des trois piliers de la stratégie de sécurité :

8.1.1 Pilier 1 : Segmentation Réseau (pfSense Firewall)

Rôle : Barrière statique de premier niveau.

Efficacité validée :

- ✓ Isolement complet de la LAN vis-à-vis de la DMZ.
- ✓ Rejet de toute connexion non explicitement autorisée.
- ✓ Routage obligatoire via le pare-feu (pas de contournement détecté).
- ✓ Traçabilité complète de chaque tentative de traversée.

Limitation acceptée :

- ✗ Incapable de détecter les attaques dans le contenu valide.
- ✗ Ne peut pas différencier une requête HTTP légitime d'une requête HTTP malveillante.

8.1.2 Pilier 2 : Filtrage Applicatif (Snort IDS)

Rôle : Surveillance dynamique et détection de signatures d'attaque.

Efficacité validée :

- ✓ Détection fiable des tentatives de traversée de répertoire.
- ✓ Analyse en profondeur (DPI) du contenu des paquets.
- ✓ Génération d'alertes horodatées et détaillées.
- ✓ Archivage pour analyse post-incident.

Limitation acceptée :

- ✗ Détecte mais ne bloque pas (en mode "alert"). Une transition vers "drop" nécessite une calibration supplémentaire pour éviter les faux positifs.
- ✗ Nécessite une maintenance régulière des signatures.

8.1.3 Pilier 3 : Audit et Traçabilité

Rôle : Documentation et traçabilité pour investigation post-incident.

Efficacité validée :

- ✓ Logs détaillés de chaque paquet traversant pfSense.
- ✓ Alertes Snort stockées et consultables.
- ✓ Timestamps synchronisés permettant une reconstitution chronologique.
- ✓ Chaîne d'évidence pour justifier des actions administratives.

Limitation acceptée :

- ✗ Storage des logs en croissance continue (archivage requis).
- ✗ Analyse manuelle des gros volumes de logs (outils d'indexation comme ELK recommandés).

8.2 Conclusion de la Validation

L'infrastructure démontre une approche défensive en profondeur (Defense in Depth) où chaque couche compense les faiblesses de la précédente :

1. **Passerelle pfSense** : Contrôle des flux au niveau réseau.
2. **Snort IDS** : Analyse du contenu et détection d'anomalies.
3. **Logs et alertes** : Traçabilité et capacité de réaction.

Aucune mesure de sécurité n'est parfaite, mais cette **combinaison de contrôles** élève significativement le coût et la difficulté d'une attaque réussie.

---

9. Recommandations et Perspectives

9.1 Court Terme (Semaines à Mois)

1. **Monitoring continu** : Déployer un outil centralisé d'agrégation de logs (ELK Stack, Splunk) pour une analyse proactive.
2. **Backups** : Configurer des sauvegardes régulières de la configuration pfSense et des logs Snort.
3. **Mise à jour de signatures Snort** : Actualiser les rule sets hebdomadairement pour rester protégé contre les menaces émergentes.
4. **Tests de pénétration supplémentaires** : Simuler d'autres scénarios d'attaque (brute-force SSH, injection SQL, DoS).

9.2 Moyen Terme (Mois à Années)

1. **HTTPS/SSL sur le serveur web** : Ajouter le chiffrement du transport (port 443) pour protéger la confidentialité et l'intégrité.

2. **Web Application Firewall (WAF)** : Envisager un WAF (ModSecurity sur Apache) pour filtrer les attaques web au niveau applicatif.
3. **Authentification multi-facteurs (MFA)** : Protéger l'accès administrateur par MFA.
4. **Infrastructure d'authentification centralisée** : Déployer LDAP ou Active Directory pour une gestion unifiée des comptes.

9.3 Long Terme (Années)

1. **Haute disponibilité (HA)** : Dupliquer le pare-feu et les serveurs pour assurer la continuité de service.
2. **Zero Trust Architecture** : Évoluer vers un modèle où aucun utilisateur ni appareil n'est de confiance par défaut.
3. **Conformité réglementaire** : Aligner l'infrastructure sur les standards (NIST Cybersecurity Framework, ISO 27001) et obtenir des certifications.

---

10. Conclusion Générale

Ce projet a démontré la conception et l'implémentation réussies d'une **infrastructure réseau sécurisée et bien segmentée** alignée sur les meilleures pratiques industrielles.

Les trois composantes essentielles ont été validées :

1. **Pare-feu statique (pfSense)** : Contrôle d'accès granulaire basé sur des règles explicites.
2. **Détection d'intrusion (Snort IDS)** : Surveillance continue et détection d'attaques applicatives.
3. **Traçabilité complète** : Logging exhaustif permettant investigation et conformité.

L'infrastructure résiste efficacement aux vecteurs d'attaque testés : reconnaissance passive (scans Nmap), exploitation d'attaques web connues (path traversal), et tentatives d'accès non autorisés.

Bien que nul système n'offre une protection absolue, la **Defense in Depth** mise en place offre un niveau de sécurité acceptable pour une infrastructure de production légère, et les procédures de maintenance établies garantissent que le dispositif reste opérationnel et à jour face aux menaces évolutives. Beaucoup de choses peuvent être apportées à ce projet, que ce soit en matière de sécurité ou tout simplement, des tests plus approfondis. Ce projet est en réalité la mise en place mais des projets plus complets et plus pointilleux seront réalisés à la suite de celui-ci.

Ce rapport conclut positivement sur la capacité de cette architecture à protéger les ressources critiques tout en préservant la disponibilité des services légitimes.