AYI NEDJIMI Consultants

Nessus et Greenbone : Guide Scanners Vulnérabilités

Catégorie : Cybersécurité Générale Lecture : 15 min Publié le : 26/03/2026 Auteur : Ayi NEDJIMI

Guide complet Nessus et Greenbone (OpenVAS) 2026 : installation, configuration scans authentifiés, exploitation résultats, comparaison et.

Voici un chiffre qui surprend toujours lors des missions d'audit : **un scan sans credentials trouve en moyenne 3 fois moins de vulnérabilités qu'un scan authentifié** sur le même hôte. La plupart des équipes sécurité font tourner des scans réseau non-authentifiés en se disant qu'elles couvrent leur périmètre — et passent à côté de l'essentiel. Les ***scanners de vulnérabilités*** comme **nessus greenbone scanner vulnerabilites** ne sont pas de simples outils de détection réseau : ce sont des moteurs d'évaluation de posture de sécurité qui, correctement configurés, peuvent identifier des failles critiques avant qu'un attaquant ne les exploite. Dans le cycle de sécurité NIST CSF et la démarche ISO 27001, la phase "Identify" repose largement sur cette capacité à cartographier précisément ce qui est exposé et vulnérable. Ce guide complet couvre l'installation de A à Z, la configuration des scans avec et sans credentials, la lecture et priorisation des résultats CVSS, ainsi que la comparaison objective entre Nessus et Greenbone (OpenVAS) — pour que vous sachiez laquelle choisir selon votre contexte et votre budget. J'y ajoute un cas pratique d'audit Windows complet, l'automatisation via API, et les bonnes pratiques légales et opérationnelles que j'applique en mission.

En bref : Nessus et Greenbone (OpenVAS) sont les deux scanners de vulnérabilités les plus déployés au monde, couvrant ensemble plus de 80 % des audits techniques en entreprise. Nessus propose une couverture plugin inégalée avec plus de 180 000 détections, tandis que Greenbone offre une alternative open-source robuste via son architecture GVM. Ce guide couvre l'installation, la configuration, l'exploitation des résultats et la comparaison détaillée des deux outils — avec des retours terrain sur ce que ces scans trouvent vraiment en production.

Pourquoi Scanner les Vulnérabilités : Scan vs Pentest

Un scanner de vulnérabilités et un pentest ne font pas la même chose, et confondre les deux est une erreur que je vois régulièrement dans les organisations qui débutent leur programme de sécurité. Le scanner est **automatisé, exhaustif et non-destructif** : il parcourt l'ensemble du parc pour identifier les failles connues, cataloguées dans des bases comme le NVD (National Vulnerability Database) ou listées via les identifiants ***CVE (Common Vulnerabilities and Exposures)***.

Le pentest, lui, est ciblé, créatif et souvent destructif (dans un périmètre contrôlé) : il cherche à enchaîner des vulnérabilités pour démontrer un impact réel. Le scanner prépare le terrain du pentest — il n'en est pas le substitut.

CVSS (Common Vulnerability Scoring System) : Système de notation standardisé pour évaluer la gravité d'une vulnérabilité. La version 3.1 est la plus répandue, la version 4.0 (publiée fin 2024) introduit de nouvelles métriques d'environnement et de sécurité. Un score CVSS va de 0 à 10 : Critical (9-10), High (7-8.9), Medium (4-6.9), Low (0.1-3.9).
Dans le référentiel **NIST CSF (Cybersecurity Framework)**, le scan de vulnérabilités s'inscrit dans la fonction "Identify" (ID.RA — Risk Assessment) et conditionne directement l'efficacité des fonctions "Protect" et "Respond". Pour **ISO 27001:2022**, l'annexe A.8.8 impose explicitement la gestion des vulnérabilités techniques, avec évaluation régulière.

Chiffre clé : Selon le rapport Tenable 2025 sur l'exposition cybernétique, 40 % des brèches analysées exploitaient des vulnérabilités pour lesquelles un patch existait depuis plus de 90 jours. Le problème n'est pas toujours l'absence de détection — c'est l'absence de remédiation structurée.

Nessus (Tenable) : Présentation et Versions

Nessus est né en 1998 sous la plume de Renaud Deraison. Longtemps open-source, il est devenu propriétaire en 2005 lors du rachat par Tenable Network Security. Aujourd'hui, **Tenable** propose plusieurs déclinaisons selon les besoins :

- **Nessus Essentials** : gratuit, limité à 16 IPs. Idéal pour les labs et l'apprentissage.
- **Nessus Professional** : illimité en IPs, environ 3 500 $/an. Standard pour les équipes sécurité.
- **Nessus Expert** : inclut les scans d'infrastructure cloud (AWS, GCP, Azure) et l'analyse de code IaC (Terraform, CloudFormation).
- **Tenable.io** : plateforme SaaS, gestion multi-sites, intégrations SIEM/SOAR natives.
- **Tenable.sc** (ex SecurityCenter) : on-premise, pour les grandes entreprises et OIV.

La force de Nessus réside dans ses **180 000+ plugins** de détection — chaque plugin correspond à une vérification spécifique, qu'il s'agisse d'un CVE, d'une mauvaise configuration ou d'une information de découverte. Cette couverture est mise à jour quotidiennement par les équipes de Tenable Research.

Installation et Configuration de Nessus

Installation sur Debian/Ubuntu

Téléchargez le paquet depuis tenable.com/downloads puis installez :

```
# Téléchargement (remplacer par la version courante)
wget https://www.tenable.com/downloads/api/v1/public/pages/nessus/downloads/XXXX/download?i_agree_to_tenable_license_agreement=true   -O Nessus-10.x.x-debian10_amd64.deb

# Installation
sudo dpkg -i Nessus-10.x.x-debian10_amd64.deb

# Démarrage du service
sudo systemctl enable nessusd
sudo systemctl start nessusd

# Vérification
sudo systemctl status nessusd
```

L'interface web est accessible sur **https://localhost:8834**. Acceptez le certificat auto-signé et suivez l'assistant d'activation. Pour **Nessus Essentials**, saisissez votre code d'activation obtenu gratuitement sur tenable.com — la mise à jour initiale des plugins prend 10-20 minutes.

```
# Mise à jour manuelle des plugins via CLI
sudo /opt/nessus/sbin/nessuscli update --all

# Liste des plugins disponibles
sudo /opt/nessus/sbin/nessuscli fetch --list

# Vérification version et statut
sudo /opt/nessus/sbin/nessuscli --version
```

Conseil terrain : Si vous déployez Nessus dans un réseau segmenté, prévoyez l'accès sortant vers updates.nessus.org (port 443) pour les mises à jour de plugins. En environnement air-gapped, utilisez `nessuscli update --file plugins.tgz` avec un bundle téléchargé manuellement depuis le portail Tenable.

Types de Scans Nessus et Configuration

Nessus propose une bibliothèque de **templates de scan** couvrant tous les cas d'usage :

Template	Usage	Credentials requis	Durée typique
Basic Network Scan	Découverte et vulnérabilités générales	Non	15-60 min
Advanced Scan	Contrôle fin de tous les paramètres	Optionnel	30-120 min
Credentialed Patch Audit	Vérification patches OS/applicatifs	Oui (SSH/WMI)	20-45 min
Web Application Tests	Vulnérabilités web (SQLi, XSS, etc.)	Optionnel	30-90 min
PCI DSS Compliance	Conformité PCI DSS 4.0	Recommandé	45-90 min
HIPAA Compliance	Conformité HIPAA (environnements santé)	Oui	45-90 min
Malware Scan	Détection malwares connus	Oui	20-60 min

Pour créer un scan, naviguez dans **Scans** → **New Scan**, choisissez votre template, puis configurez :

- **Targets** : IPs, plages CIDR (192.168.1.0/24), noms d'hôtes, fichier texte d'IPs
- **Schedule** : scan unique ou récurrent (quotidien, hebdomadaire)
- **Credentials** : SSH (clé ou mot de passe), Windows (domaine ou local), SNMP v1/v2/v3, base de données
- **Assessment** : niveau d'intrusion (safe, non-disruptive, disruptive)
- **Advanced** : ports à scanner, timeouts, parallélisme

Lecture et Exploitation des Résultats Nessus

Un rapport Nessus classe les findings par sévérité : **Critical**, **High**, **Medium**, **Low**, et **Info**. Chaque finding affiche :

- Le **Plugin ID** : identifiant unique du plugin Nessus (ex: 93562 pour MS16-120)
- Le **CVE associé** et son score CVSS v3.x
- La **description** de la vulnérabilité et son impact
- La **solution** recommandée (patch, configuration)
- Le **output brut** du plugin (ce que le scan a réellement détecté)

REX terrain : Lors d'un audit d'infrastructure Windows en 2025, j'ai trouvé que 60 % des findings "Critical" remontés par Nessus concernaient des versions TLS obsolètes (TLS 1.0/1.1 activées) et des algorithmes de chiffrement dépréciés. Ces éléments, invisibles depuis un scan réseau non-authentifié, sont apparus uniquement grâce aux credentials Windows configurés. Le scan sans auth avait remonté 12 findings — le scan authentifié en avait trouvé 87 sur les mêmes hôtes.

Pour exporter les résultats : **Report** → **Export**, formats disponibles :

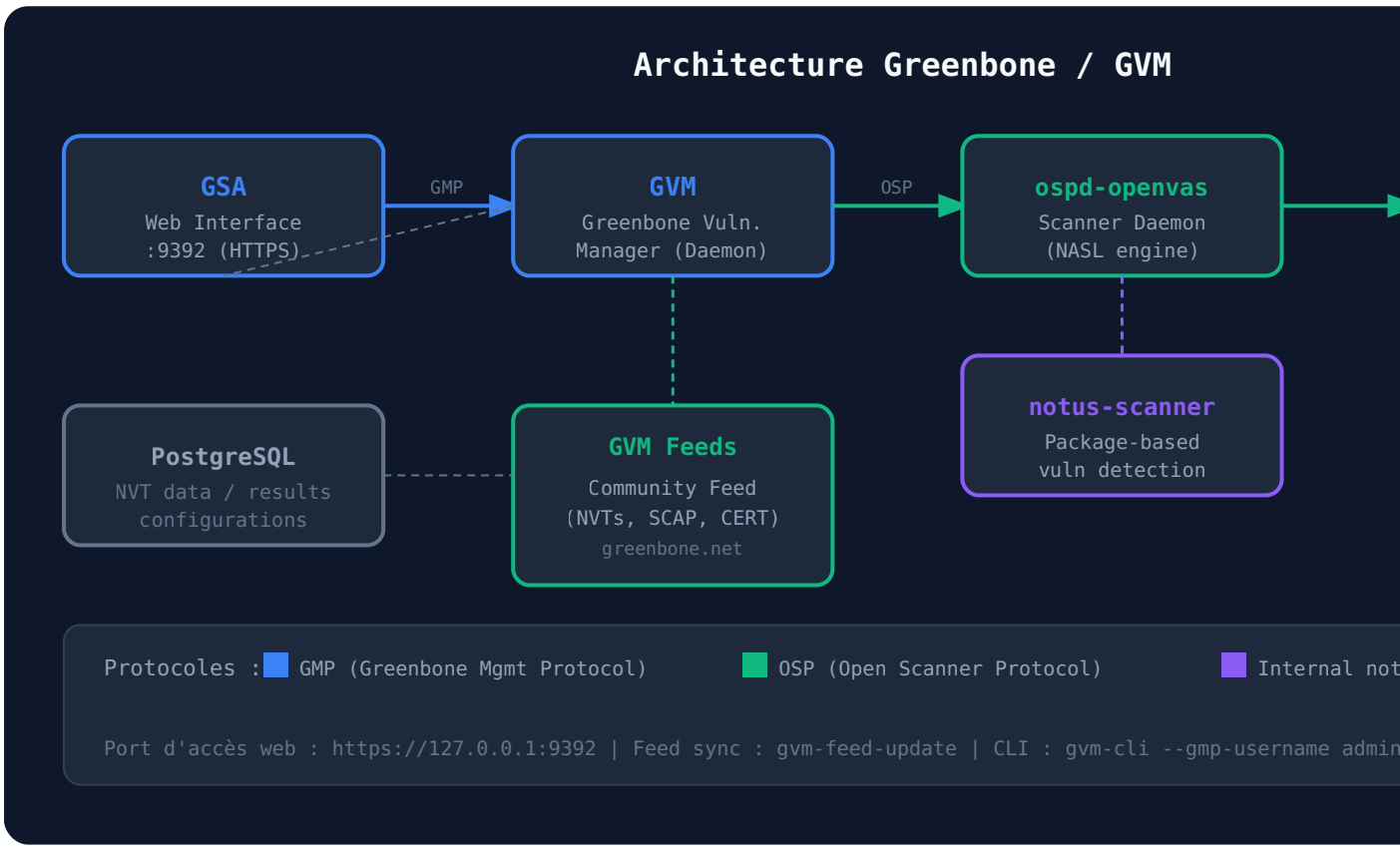
- **PDF** : rapport exécutif ou technique, personnalisable avec logo
- **CSV** : pour traitement dans Excel, Power BI, ou scripts Python
- **Nessus XML** : format natif pour import dans d'autres outils (Metasploit, Dradis)
- **HTML** : rapport web standalone

```
# Export via nessuscli (pratique pour l'automatisation)
sudo /opt/nessus/sbin/nessuscli report export --id SCAN_ID --format pdf --output /tmp/
rapport.pdf

# Lister les scans disponibles
sudo /opt/nessus/sbin/nessuscli report list
```

Greenbone / OpenVAS : Architecture et Composants

OpenVAS (Open Vulnerability Assessment Scanner) est devenu **Greenbone Community Edition** sous l'égide de Greenbone Networks. L'architecture est modulaire et distribuée :



Les composants principaux de la stack Greenbone sont :

- ***GSA (Greenbone Security Assistant)*** : interface web accessible sur le port 9392, construit avec React. C'est le tableau de bord principal pour gérer les scans, visualiser les résultats et générer les rapports.
- ***GVM (Greenbone Vulnerability Manager)*** : le cœur du système, qui orchestre les scans, stocke les résultats en PostgreSQL et expose l'API GMP.
- **ospd-openvas** : le démon scanner qui exécute les scripts NASL (Nessus Attack Scripting Language) — oui, Greenbone utilise le même langage de scripting que Nessus.
- **notus-scanner** : scanner spécialisé dans la détection de vulnérabilités par analyse de paquets installés (plus rapide que NASL pour les CVE OS).
- **Greenbone Community Feed** : base de NVTs (Network Vulnerability Tests), mise à jour quotidiennement. La version Enterprise Feed offre une couverture plus large.

Installation de Greenbone sur Kali Linux

Kali Linux intègre Greenbone/OpenVAS dans ses dépôts officiels. L'installation est simplifiée par le script `gvm-setup` :

```
# Installation des paquets GVM
sudo apt update && sudo apt install -y gvm

# Setup automatique (télécharge feeds, crée DB PostgreSQL, génère certs)
# Durée : 15-30 min selon connexion internet
sudo gvm-setup

# Récupérer le mot de passe admin généré automatiquement
# Il s'affiche à la fin de gvm-setup — à sauvegarder IMMÉDIATEMENT
# Si perdu : sudo gvm-cli --gmp-username admin --gmp-password PASS #   socket --
socketpath /run/gvmd/gvmd.sock --xml ""

# Démarrage des services
sudo gvm-start

# Vérification que tout est up
sudo gvm-check-setup
```

L'interface est accessible sur **https://127.0.0.1:9392**. Connectez-vous avec l'utilisateur `admin` et le mot de passe généré par gvm-setup.

```
# Mise à jour manuelle du feed
sudo gvm-feed-update

# Vérifier le statut des feeds
sudo gvm-cli --gmp-username admin --gmp-password VOTRE_PASS   socket --socketpath /run/
gvmd/gvmd.sock   --xml ""

# Lister les scan configs disponibles
sudo gvm-cli --gmp-username admin --gmp-password VOTRE_PASS   socket --socketpath /run/
gvmd/gvmd.sock   --xml ""
```

Attention : `gvm-setup` peut échouer si PostgreSQL n'est pas correctement initialisé. Si vous voyez des erreurs `pg_ctlcluster`, lancez d'abord `sudo pg_lsclusters` pour vérifier l'état du cluster PostgreSQL, puis `sudo pg_ctlcluster 16 main start` si nécessaire.

Configuration des Scans Greenbone

Dans l'interface GSA, la configuration d'un scan passe par :

1. **Hosts** (Configuration → Targets) : définir les cibles avec IP, FQDN ou fichier
2. **Credentials** (Configuration → Credentials) : SSH, SMB, ESXi, SNMP
3. **Scan Config** : choisir parmi les profils prédéfinis
4. **Task** (Scans → Tasks → New Task) : assembler cibles + credentials + config

Les **Scan Configs** principales sont :

- **Full and Fast** : recommandé pour la plupart des cas — bon équilibre couverture/vitesse
- **Full and Deep** : tests plus intrusifs, plus lent, peut perturber certains services
- **System Discovery** : découverte réseau uniquement, sans tests de vulnérabilités
- **Host Discovery** : simple ping + port scan, très rapide

La métrique ***QoD (Quality of Detection)*** est propre à Greenbone : elle indique la fiabilité de la détection, de 0 à 100 %. Un QoD de 70 % et plus est considéré comme fiable. Filtrer par QoD >= 70 % réduit considérablement les faux positifs.

Comparaison Nessus vs Greenbone : Tableau Complet

Nessus — Avantages

- 180 000+ plugins, la couverture la plus large du marché
- Interface intuitive, prise en main rapide
- Compliance checks natifs (PCI DSS, HIPAA, CIS)
- Support Tenable, documentation exhaustive
- Intégrations SIEM/SOAR prêtes à l'emploi

Nessus — Inconvénients

- Coût élevé (Pro ~3 500 $/an)
- Essentials limité à 16 IPs
- Code propriétaire, pas d'audit du moteur
- Dépendance Tenable pour les mises à jour

Scans avec Credentials : La Différence Clé

On ne répètera jamais assez : **un scan non-authentifié est un scan incomplet**. Les études terrain montrent systématiquement que les scans avec credentials trouvent **3 à 4 fois plus de vulnérabilités** sur les mêmes hôtes. La raison est simple : sans accès au système, le scanner est limité à ce qui est visible depuis le réseau — ports ouverts, bannières de services, protocoles exposés. Avec credentials, il peut lire la liste des paquets installés, vérifier les versions exactes, tester les configurations locales et valider l'application des patches.

Configuration SSH Credentials (Nessus et Greenbone)

Pour les systèmes Linux/Unix, la méthode recommandée est l'**authentification par clé SSH** avec un compte dédié au scan :

```
# Créer un compte de scan avec privilèges minimaux
sudo useradd -m -s /bin/bash nessus-scan
sudo passwd nessus-scan

# Générer une clé SSH dédiée (sur le serveur Nessus)
ssh-keygen -t ed25519 -f ~/.ssh/nessus_scan_key -C "nessus-scanner"

# Copier la clé publique sur les cibles
ssh-copy-id -i ~/.ssh/nessus_scan_key.pub nessus-scan@192.168.1.100

# Configurer sudo sans mot de passe pour les commandes nécessaires
# /etc/sudoers.d/nessus-scan :
nessus-scan ALL=(ALL) NOPASSWD: /usr/bin/dpkg, /bin/rpm, /usr/bin/yum, /usr/bin/apt
```

Dans Nessus : **Credentials** → **SSH** → **Authentication method : Public Key**, uploader la clé privée. Dans Greenbone : **Configuration** → **Credentials** → **New Credential** → **SSH** → **Private Key**.

Configuration Windows Credentials

Pour Windows, Nessus et Greenbone utilisent WMI/SMB. Le compte doit avoir les droits suivants :

- Membre du groupe **Administrators** local (ou **Domain Admins** si scan AD complet)
- Accès WMI activé (par défaut sur les systèmes Windows)
- Partage `ADMIN$` accessible (pour certains plugins)
- Firewall Windows : autoriser les règles "Remote Administration" et "Windows Management Instrumentation"

Principe de moindre privilège : En production, on peut utiliser un compte avec des droits réduits pour la majorité des checks. Créez un GPO dédié "Scanner Account" qui accorde les droits WMI et Remote Registry sans être administrateur local complet. La plupart des checks de patch level fonctionnent avec ces droits réduits.

Exploiter les Résultats : CVSS ne suffit pas

Une erreur classique : trier les findings par CVSS et attaquer les Critical en premier sans réfléchir. **Le CVSS mesure la gravité intrinsèque d'une vulnérabilité, pas le risque réel pour votre organisation.** Une faille CVSS 9.8 sur un serveur sans accès internet, dans un VLAN isolé, sans données sensibles, est moins prioritaire qu'une faille CVSS 6.5 sur un serveur web public exposant des données clients.

Le **CVSS v4.0** (publié par le FIRST en novembre 2024) tente d'adresser ce problème avec de nouvelles métriques supplémentaires :

- **CVSS-BTE** : Base + Threat (exploitabilité réelle connue) + Environmental (contexte de votre infra)
- Nouveau groupe de métriques **Supplemental** : Safety, Automatable, Recovery, Value Density
- Score **Safety** pour les systèmes OT/ICS critiques

Workflow de Remédiation Recommandé

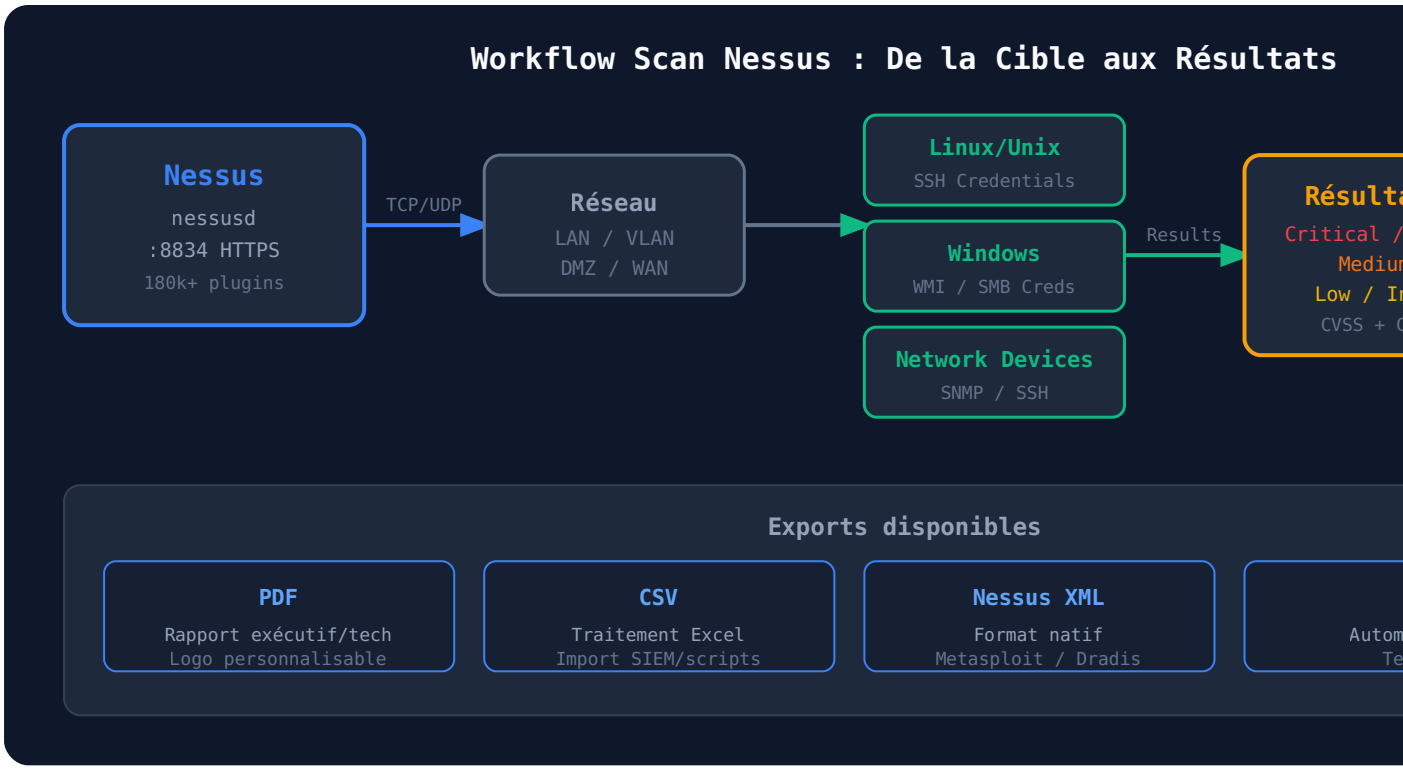
1. **Identification** : export CSV du scan, filtrage par sévérité et contexte métier
2. **Enrichissement** : croiser avec NVD, vérifier si exploit public disponible (ExploitDB, Metasploit)
3. **Priorisation** : pondérer CVSS × exposition × exploitabilité × valeur du système
4. **Ticketing** : créer un ticket ITSM (Jira, ServiceNow) par finding ou groupe de findings
5. **Patch** : appliquer en environnement de test d'abord, puis production
6. **Rescan ciblé** : valider la remédiation sur les hosts patchés
7. **Clôture** : mettre à jour le ticket, archiver dans le CMDB

Métriques de suivi essentielles

Pour mesurer l'efficacité de votre programme de gestion des vulnérabilités :

- ***MTTR (Mean Time To Remediate)*** : temps moyen entre la détection et le patch. Cible recommandée : < 15 jours pour Critical, < 30 jours pour High.
- **Patch Compliance Rate** : pourcentage de systèmes patchés dans les SLA définis. Cible : > 95 % à 30 jours pour Critical.
- **Vulnerability Age** : âge moyen des vulnérabilités ouvertes. Un indicateur clé de la dette sécurité.
- **Scan Coverage** : pourcentage du parc effectivement scanné. Beaucoup d'organisations scannent 60-70 % de leur parc — les 30-40 % restants sont souvent les plus risqués.

Architecture d'un Scan Nessus en Réseau



Automatisation et Intégration CI/CD

Les scanners de vulnérabilités modernes offrent des APIs complètes pour s'intégrer dans les pipelines DevSecOps. La tendance **Shift Left Security** consiste à scanner avant le déploiement en production, pas après.

API Tenable.io — Exemples Python

```
import requests

# Configuration API Tenable.io
API_BASE = "https://cloud.tenable.com"
HEADERS = {
    "X-ApiKeys": "accessKey=VOTRE_ACCESS_KEY;secretKey=VOTRE_SECRET_KEY",
    "Content-Type": "application/json"
}

# Lancer un scan
def launch_scan(scan_id):
    resp = requests.post(f"{API_BASE}/scans/{scan_id}/launch", headers=HEADERS)
    return resp.json()

# Récupérer les vulnérabilités d'un scan
def get_vulnerabilities(scan_id):
    resp = requests.get(f"{API_BASE}/scans/{scan_id}", headers=HEADERS)
    data = resp.json()
    return data.get('vulnerabilities', [])

# Exporter en CSV
def export_scan_csv(scan_id):
    payload = {"format": "csv", "chapters": "vuln_hosts_summary"}
    resp = requests.post(f"{API_BASE}/scans/{scan_id}/export",
                         headers=HEADERS, json=payload)
    file_id = resp.json()['file']
    # Polling du statut d'export
    while True:
        status = requests.get(f"{API_BASE}/scans/{scan_id}/export/{file_id}/status",
                              headers=HEADERS).json()
        if status['status'] == 'ready':
            break
    # Téléchargement
    content = requests.get(f"{API_BASE}/scans/{scan_id}/export/{file_id}/download",
                           headers=HEADERS)
    return content.content
```

API Greenbone via gvm-cli

```
# Installation du client Python GVM
pip install python-gvm

# Script Python pour lancer un scan et récupérer les résultats
from gvm.connections import UnixSocketConnection
from gvm.protocols.gmp import Gmp
from gvm.transforms import EtreeCheckCommandTransform

connection = UnixSocketConnection(path='/run/gvmd/gvmd.sock')
transform = EtreeCheckCommandTransform()

with Gmp(connection=connection, transform=transform) as gmp:
    # Authentification
    gmp.authenticate('admin', 'VOTRE_MOT_DE_PASSE')

    # Lancer une tâche de scan
    gmp.start_task(task_id='TASK_UUID')

    # Récupérer les rapports
    reports = gmp.get_reports(filter_string="sort-reverse=date rows=10")

    # Exporter en PDF
    pdf_report = gmp.get_report(
        report_id='REPORT_UUID',
        report_format_id='c402cc3e-b531-11e1-9163-406186ea4fc5',  # PDF format ID
        ignore_pagination=True
    )
```

Intégration GitLab CI/CD : Ajoutez un stage de scan de vulnérabilités dans votre pipeline avant le déploiement en staging. Utilisez l'image Docker `instrumentisto/nmap` pour les scans rapides ou l'API Tenable.io pour des scans complets. Configurez des seuils : bloquer le pipeline si un finding Critical est détecté sur le nouveau code ou les nouvelles dépendances.

Cadre Légal et Bonnes Pratiques

Avertissement légal : Scanner des systèmes sans autorisation explicite et écrite du propriétaire est une infraction pénale dans la quasi-totalité des pays. En France, les articles 323-1 à 323-7 du Code pénal couvrent l'accès frauduleux à des systèmes informatiques. Assurez-vous d'avoir une autorisation formelle avant tout scan. Voir aussi notre guide sur la sécurité des pipelines CI/CD.

Au-delà du cadre légal, plusieurs bonnes pratiques opérationnelles s'imposent :

- **Fenêtres de scan** : planifiez les scans hors heures de bureau ou en heures creuses. Un scan "Full and Deep" peut saturer le réseau et impacter les performances des applications.
- **Throttling** : limitez la vitesse du scanner (max_checks, max_hosts dans Nessus) sur les systèmes sensibles. Certains équipements réseau (switches anciens, imprimantes) peuvent crasher sous la charge d'un scan agressif.
- **RGPD** : les rapports de scan peuvent contenir des données personnelles (noms d'utilisateurs, configurations). Classifiez ces rapports comme confidentiels et appliquez une politique de rétention (recommandation : 1-3 ans selon votre contexte).

- **Documentation** : maintenez un registre des scans réalisés avec date, périmètre, compte utilisé, et résultats. Essentiel pour les audits de conformité ISO 27001 / NIS2.

Cas Pratique : Audit d'un Réseau Windows d'Entreprise

Environnement de test : Active Directory Windows Server 2022, 50 workstations Windows 10/11, Nessus Professional 10.x, scan avec credentials de domaine (compte dédié "svc_nessus"). Périmètre : 192.168.10.0/24.

Voici ce qu'on trouve typiquement lors d'un audit d'entreprise Windows avec Nessus :

1. **MS17-010 (EternalBlue)** — Plugin 97737 : encore présent sur des systèmes Windows 7/ Server 2008 non mis à jour. CVSS 9.8. Si vous en trouvez en 2026, c'est un signal d'alerte majeur sur la gestion des patches.
2. **PrintNightmare patches manquants** (CVE-2021-34527) : 3 ans après la divulgation, on en trouve encore sur des serveurs d'impression en production.
3. **SMBv1 activé** — Plugin 57608 : protocole obsolète depuis 2014, encore activé sur 20-30 % des parcs enterprise pour des raisons de compatibilité applicative.
4. **TLS 1.0/1.1 activés** — Plugins 104743/157288 : les plus fréquents dans nos scans. Simple à désactiver via GPO, souvent négligé.
5. **Credentials par défaut** sur des équipements réseau (imprimantes, NAS, switches) — Plugins multiples : souvent les findings les plus critiques en termes d'exploitation réelle.
6. **Antivirus définitions obsolètes** — Plugins Windows Defender/WSUS : témoin d'une gestion défaillante des mises à jour.

```
$ nessuscli scan list | grep "Windows Audit Q1 2026"
ID: 452 | Status: completed | Targets: 192.168.10.0/24 | Duration: 47min

$ nessuscli report export --id 452 --format csv --output /tmp/audit_q1.csv
Export completed: 1247 findings | Critical: 8 | High: 47 | Medium: 156 | Low: 312 | Info: 724

# Filtrer uniquement les Critical :
$ grep ",Critical," /tmp/audit_q1.csv | wc -l
8
```

Mon workflow post-scan sur ce type d'audit : exporter en CSV, importer dans un tableur Python (pandas), grouper par hôte et par plugin, puis générer un rapport de priorisation par Business Unit. Les équipes IT apprécient un rapport qui liste "les 10 actions critiques à faire cette semaine" plutôt qu'un dump de 1247 findings sans contexte.

Alternatives : Qualys, Rapid7, OpenSCAP

Le marché des scanners de vulnérabilités va bien au-delà de Nessus et Greenbone :

- ***Qualys VMDR*** : SaaS, excellent pour les grandes entreprises multi-sites. Gestion d'actifs intégrée, patch management, conformité cloud. Coût élevé mais ROI mesurable pour les équipes > 10 personnes.
- **Rapid7 InsightVM** (ex-Nexpose) : concurrent direct de Nessus Professional, fort en remédiation guidée et intégration avec InsightIDR (leur SIEM). Interface moderne, reporting très lisible.
- **OpenSCAP** : outil open-source basé sur les profils SCAP/XCCDF du NIST. Parfait pour la conformité CIS Benchmarks sur Linux. Moins orienté découverte réseau, plus orienté hardening configuration. Disponible dans les repos Kali/Debian : `apt install openscap-scanner`.
- **Nuclei** (ProjectDiscovery) : scanner de vulnérabilités basé sur des templates YAML, très utilisé par les bug bounty hunters. Particulièrement efficace sur les APIs REST/GraphQL.

Points clés à retenir

- **Nessus vs Greenbone** : Nessus domine en couverture (180k+ plugins vs 80k NVTs) et en interface, Greenbone est l'alternative gratuite robuste pour les labs et PME.
- **Scans authentifiés obligatoires** : un scan sans credentials trouve 3-4x moins de vulnérabilités. Configurez des comptes dédiés avec principe de moindre privilège.
- **CVSS ≠ priorité réelle** : contextualisez toujours les scores avec l'exposition réseau, l'exploitabilité et la valeur du système cible.
- **QoD Greenbone** : filtrez à >= 70 % pour réduire les faux positifs et concentrer les efforts de remédiation.
- **Automatisation** : les deux outils ont des APIs REST/GMP complètes — intégrez les scans dans vos pipelines CI/CD pour du shift left security.
- **Cadre légal** : autorisation écrite obligatoire, fenêtres de scan définies, rapports classifiés conformément au RGPD.
- **CVSS v4.0** (2024) introduit des métriques Safety et Supplemental — adoptez-le progressivement pour une priorisation plus fine.

Conclusion

Le choix entre Nessus et Greenbone n'est pas une question de "meilleur outil" absolu — c'est une question de contexte. Si votre organisation a un budget sécurité et des besoins de conformité (PCI DSS, HIPAA, ISO 27001), Nessus Professional s'amortit rapidement. Si vous démarrez votre programme de gestion des vulnérabilités, montez en compétence sur un lab, ou êtes une PME aux ressources limitées, Greenbone Community Edition couvre amplement 80 %

des besoins. Ce que je recommande systématiquement à mes clients : commencez par Greenbone pour vous familiariser avec les concepts, le workflow et l'exploitation des résultats — puis évaluez Nessus si les limitations du Community Feed vous bloquent sur des cas spécifiques.

La vraie valeur d'un scanner n'est pas dans l'outil lui-même, mais dans le **processus de remédiation** qu'il alimente. Un scan sans plan de remédiation structuré n'est qu'un rapport qui prend la poussière. Construisez d'abord le workflow (ticketing, SLAs, validation), puis automatisez les scans autour de ce processus.

Sources et références : CERT-FR · MITRE ATT&CK

Articles connexes

- Darkweb Monitoring : Outils et Techniques 2026 en 2026
- InfoStealers 2026 : Lumma, Raccoon et RedLine en 2026

Questions Fréquentes

Quelle est la différence entre Nessus Essentials et Nessus Professional ?

Nessus Essentials est gratuit mais limité à 16 adresses IP. Il offre les mêmes templates de scan que la version Professional, mais sans les scans de conformité (PCI DSS, HIPAA, CIS), sans le scan de vulnérabilités web avancé, et sans le support Tenable. Nessus Professional (~3 500 $/an) est illimité en IPs, inclut tous les templates de conformité, le support technique et les rapports personnalisables. Pour un lab personnel ou apprendre, Essentials est largement suffisant. En production d'entreprise avec plus de 16 hôtes, Professional est indispensable.

Comment réduire les faux positifs dans un scan Greenbone ?

La méthode la plus efficace est de filtrer par **QoD (Quality of Detection) >= 70 %** dans les résultats. Ce filtre est disponible directement dans l'interface GSA via Scans → Results → Filtres → QoD >= 70. Les findings avec un QoD bas (30-50 %) sont souvent des détections par déduction ou empreinte de version, sans vérification active de l'exploitabilité réelle. Par ailleurs, configurez des scans authentifiés (credentials SSH/SMB) : les faux positifs diminuent drastiquement car le scanner peut vérifier directement les versions des paquets installés plutôt que d'inférer depuis les bannières réseau.

Peut-on utiliser Nessus ou Greenbone dans un environnement cloud (AWS, Azure) ?

Oui, avec quelques adaptations. Pour AWS : déployez une instance Nessus ou GVM dans votre VPC, configurez les Security Groups pour autoriser les flux de scan depuis l'instance scanner vers les cibles. Nessus Expert inclut des connecteurs cloud natifs pour scanner les assets AWS/Azure/GCP via API sans déploiement agent. Pour Greenbone, les connecteurs cloud sont disponibles dans la version Enterprise. Dans les deux cas, vérifiez les conditions d'utilisation de votre cloud provider : certains types de scans peuvent déclencher des alertes IDS/IPS ou nécessitent une notification préalable (notamment sur AWS qui impose une autorisation pour certains types de scans réseau).

Quelle fréquence de scan recommander pour une infrastructure d'entreprise ?

La recommandation standard pour PCI DSS est un scan trimestriel minimum sur le périmètre externe et un scan après chaque changement significatif d'infrastructure. En pratique, je recommande à mes clients un cycle de scan hebdomadaire sur les systèmes critiques (serveurs exposés, contrôleurs de domaine, bases de données), mensuel sur le reste du parc, et un scan immédiat après tout déploiement majeur ou publication d'une CVE Critical touchant votre stack. Les environnements DevOps avec déploiement continu devraient intégrer les scans dans les pipelines CI/CD pour un scan à chaque build.

Comment intégrer les résultats de scan dans un SIEM comme Splunk ou QRadar ?

Nessus Professional et Tenable.sc disposent de connecteurs natifs pour Splunk (Tenable App for Splunk disponible sur Splunkbase) et IBM QRadar (DSM Tenable). Pour Greenbone, l'intégration se fait via export CSV/XML traité par un script Python ou via l'API GMP. Le pattern recommandé pour Splunk : configurer un Universal Forwarder sur le serveur Nessus pour envoyer les exports CSV vers un index dédié "vulnerability_scans", puis créer des dashboards Splunk pour le suivi du MTTR et de la compliance rate. Pour une intégration plus fine, les rapports Nessus XML peuvent être parsés et indexés comme des événements structurés dans Splunk.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr