

PROJET 2

ANALYSE DES VULNERABILITES AVEC OPENVAS



CONTEXTE DU PROJET

L'entreprise Aptsol, spécialisée dans les services IT, cherche à renforcer la sécurité de ses infrastructures en intégrant un outil de scan des vulnérabilités.

Jusqu'à présent, plusieurs solutions propriétaires ont été testées, notamment Nessus et Qualys, mais avant de faire un choix définitif, la direction IT souhaite évaluer une alternative open source : OpenVAS. L'objectif est de déterminer si cet outil peut répondre aux besoins de l'entreprise en matière de détection des vulnérabilités et de gestion des risques.

Pour cela, un audit de sécurité sera réalisé sur un serveur Windows Server 2012 R2, permettant ainsi de mesurer l'efficacité et la pertinence d'OpenVAS dans un environnement de production.

OBJECTIF DU PROJET

Déployer OpenVAS et l'évaluer en conditions réelles

Réaliser un audit de sécurité sur un serveur Windows Server 2012 R2

Identifier les vulnérabilités et proposer des solutions correctives

Méthodologie

Installation d'OpenVAS via Docker

Définition des cibles et paramétrage des scans

Lancement des scans et collecte des résultats

Analyse des vulnérabilités trouvées

Proposition de correctifs

Rédaction d'un rapport d'évaluation

CONFIGURATION DES MACHINES VIRTUELLES

VM	RAM	PROCESSEUR	STOCKAGE
Pare-feu	4Go	2 cœurs	20Go
OPENVAS	4Go	2 cœurs	60Go
SERVEUR 2012R2	4Go	2 cœurs	60Go

Plan d'adressage IP

VM	Adresse IP	CIDR	PASSERELLE
Pare-feu	192.168.2.1	255.255.255.248	192.168.2.1
OPENVAS	192.168.2.4	255.255.255.248	192.168.2.1
SERVEUR 2012R2	192.168.3.2	255.255.255.252	192.168.3.1

Installation de docker

- 1- Mise à jour du système

```
sudo apt update
```

```
sudo apt upgrade
```

- 2- Installation des dépendances

```
Sudo apt install -y ca-certificates curl gnupg lsb-release
```

- 3- Récupération de la clé GPG afin de valider les paquets récupérés depuis le dépôt Docker

```
Sudo mkdir -p /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -  
o /etc/apt/keyrings/docker.gpg
```

- 4- Ajout du dépôt officiel de Docker à la liste des sources de la machine

```
echo \ "deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- 5- Mise à jour du cache des paquets pour prendre en compte les paquets de ce nouveau dépôt et installation des paquets Docker

`Sudo apt update`

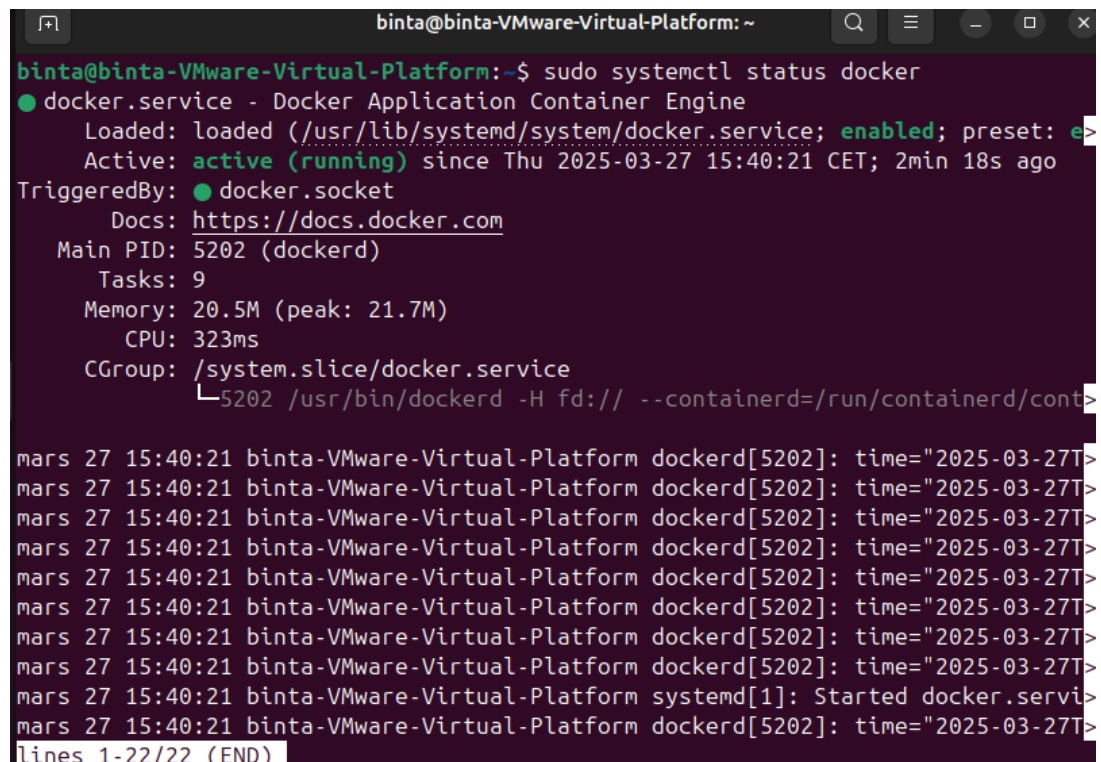
`Sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`

- 6- Pour le démarre automatique de docker

`sudo systemctl enable docker`

- 7- Vérification de l'installation de docker

`Sudo systemctl status docker`



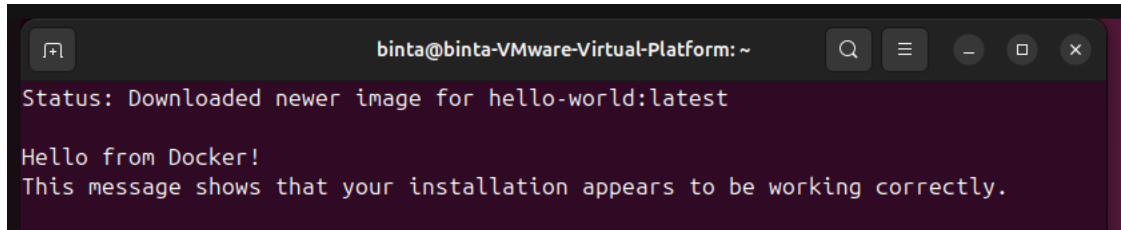
```
binta@binta-VMware-Virtual-Platform: ~  
binta@binta-VMware-Virtual-Platform:~$ sudo systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: e>  
   Active: active (running) since Thu 2025-03-27 15:40:21 CET; 2min 18s ago  
 TriggeredBy: ● docker.socket  
     Docs: https://docs.docker.com  
    Main PID: 5202 (dockerd)  
       Tasks: 9  
    Memory: 20.5M (peak: 21.7M)  
       CPU: 323ms  
    CGroup: /system.slice/docker.service  
            └─5202 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>  
  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
mars 27 15:40:21 binta-VMware-Virtual-Platform systemd[1]: Started docker.servi>  
mars 27 15:40:21 binta-VMware-Virtual-Platform dockerd[5202]: time="2025-03-27T>  
lines 1-22/22 (END)
```

- 8- Donner les droits Docker à mon utilisateur pour éviter d'utiliser sudo à chaque commande Docker

`sudo usermod -aG docker $binta`

`newgrp docker`

- 9- Exécution du container "hello-world" pour tester le fonctionnement de docker
- ```
docker run hello-world
```

A terminal window titled 'binta@binta-VMware-Virtual-Platform: ~' showing the output of the 'docker run hello-world' command. The output indicates that a newer image for 'hello-world:latest' was downloaded and then displays the message 'Hello from Docker! This message shows that your installation appears to be working correctly.'

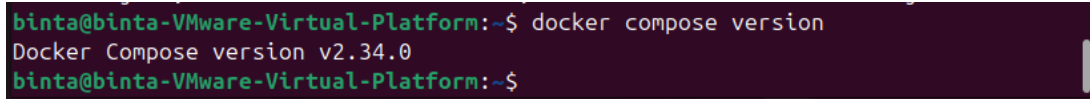
- 10- Installation de docker compose

```
sudo curl -L
```

```
"https://github.com/docker/compose/releases/latest/download/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

Vérification de l'installation

A terminal window titled 'binta@binta-VMware-Virtual-Platform: ~\$' showing the command 'docker compose version' being executed. The output is 'Docker Compose version v2.34.0'.

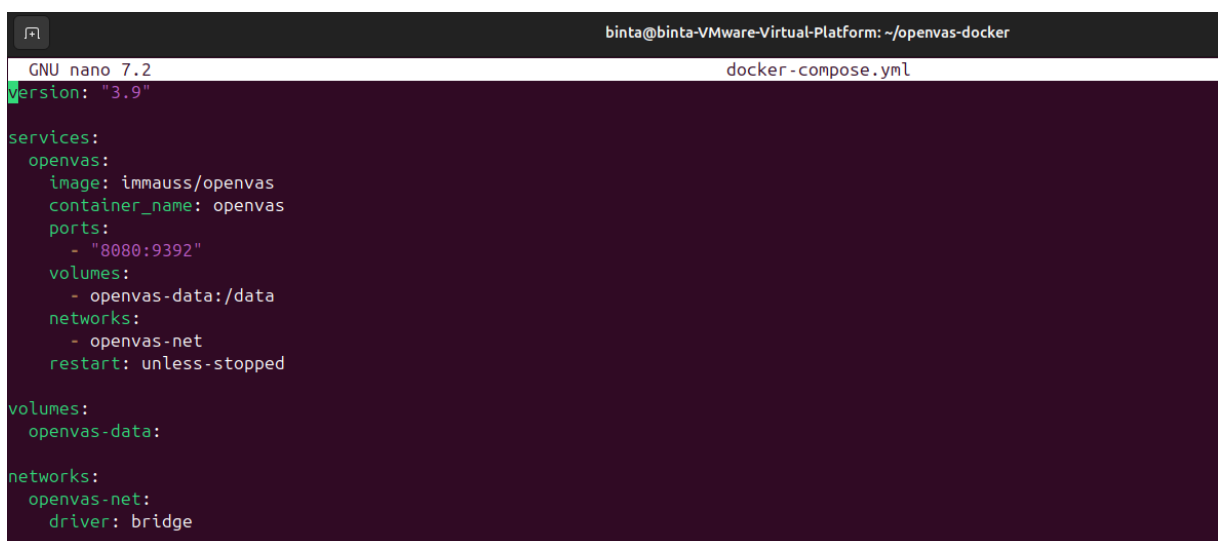
Après l'installation de docker et docker compose on passe à l'installation de Openvas

## Déploiement de Openvas avec Docker Compose

- 1- Création d'un dossier openvas-docker dans lequel je vais créer mon fichier docker-compose.yaml

```
mkdir openvas-docker && cd openvas-docker
```

- 2- Création du fichier docker-compose

A terminal window titled 'binta@binta-VMware-Virtual-Platform: ~/openvas-docker' showing the nano editor editing 'docker-compose.yml'. The content of the file is: version: '3.9', services: openvas (image: immauss/openvas, container\_name: openvas, ports: '8080:9392', volumes: openvas-data:/data, networks: openvas-net, restart: unless-stopped), volumes: openvas-data, networks: openvas-net (driver: bridge).

### 3- Démarrage de Openvas

`docker compose up -d`

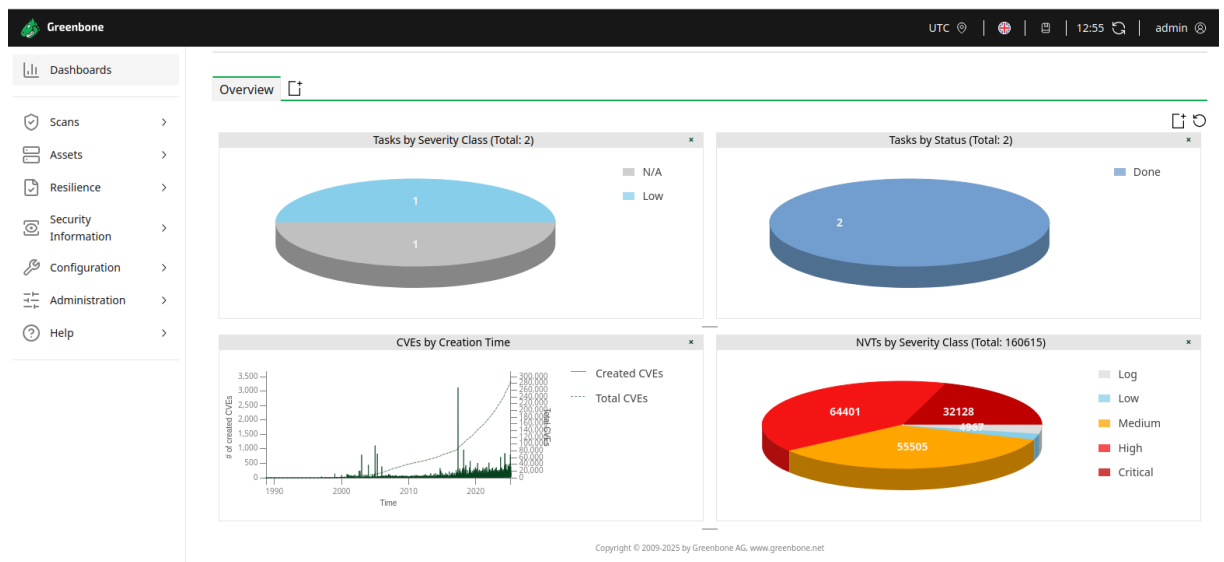
```
binta@binta-VMware-Virtual-Platform:~/openvas-docker$ sudo docker compose up -d
[+] Running 1/1
 ✓ Container openvas Running
```

### 4- Vérification du statut

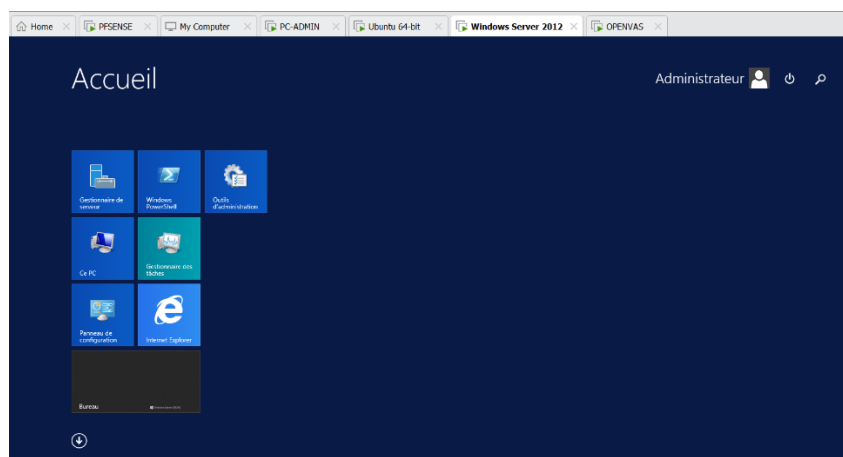
`docker compose ps`

```
binta@binta-VMware-Virtual-Platform:~/openvas-docker$ sudo docker compose ps
NAME IMAGE COMMAND SERVICE CREATED STATUS PORTS
openvas immauss/openvas "/scripts/start.sh" openvas 2 days ago Up 2 hours (healthy) 0.0.0.0:8080->9392/tcp, [::]:8080->9392/tcp
binta@binta-VMware-Virtual-Platform:~/openvas-docker$
```

### 5- Accès à l'interface web de Openvas



### Installation du serveur 2012R2

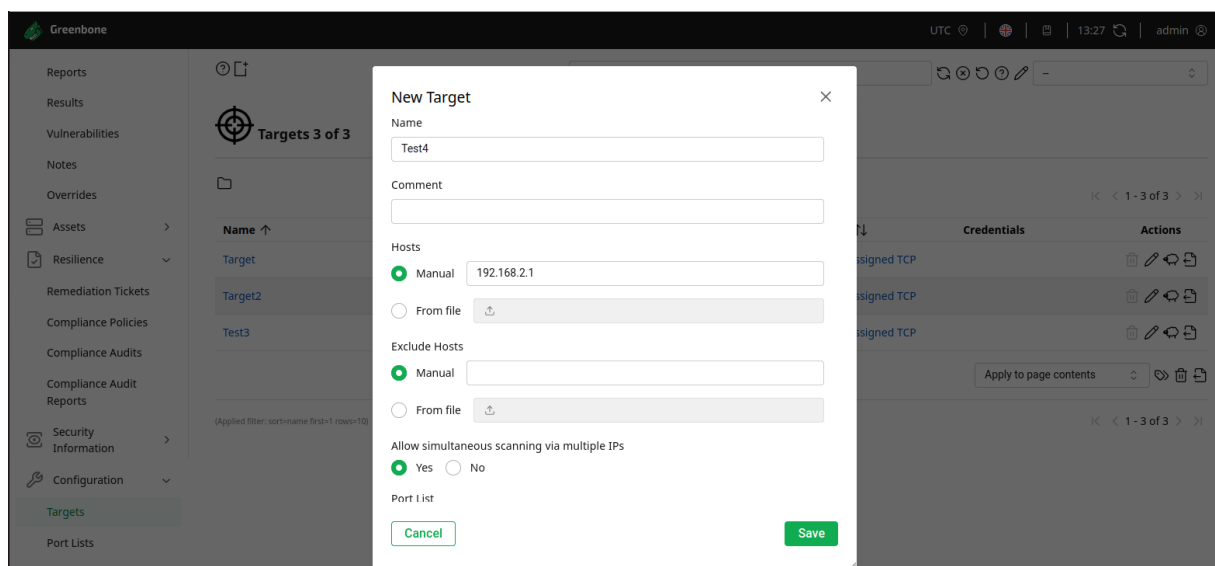


Après l'installation et la configuration du serveur sur le bon réseau, celui-ci ne répondait pas aux requêtes ping envoyées depuis Openvas. Or, pour que le scan puisse se dérouler correctement, il est essentiel que les deux machines puissent se pinger mutuellement.

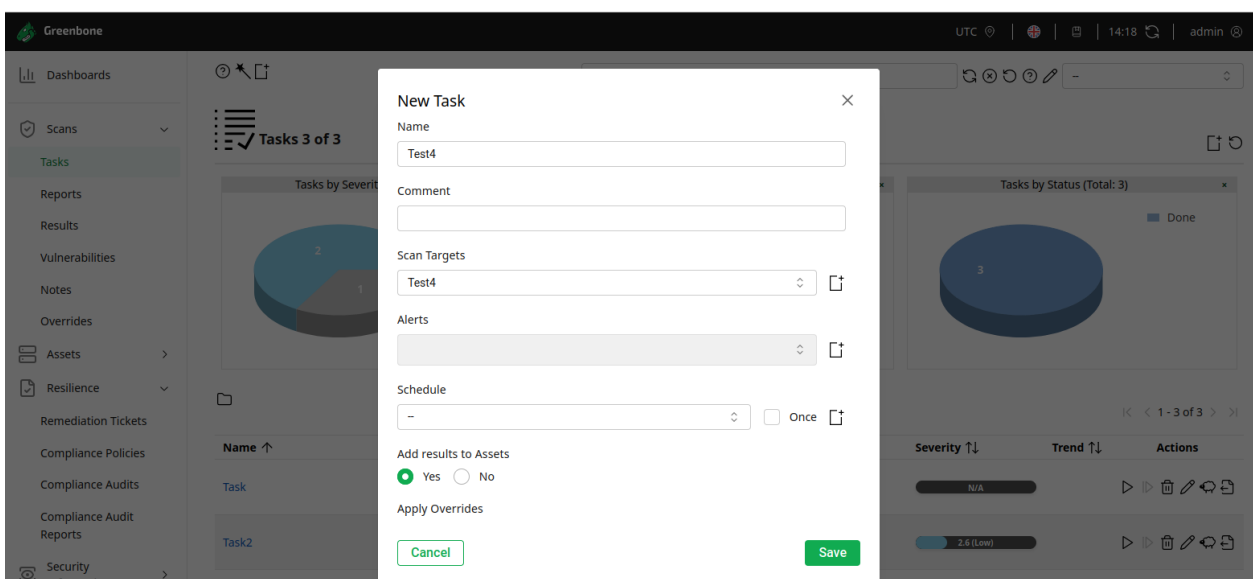
Par défaut, Windows Server bloque les requêtes ICMP. J'ai donc exécuté la commande suivante pour les autoriser :

```
netsh advfirewall firewall add rule name="Autoriser Ping" protocol=ICMPv4 dir=in action=allow
```

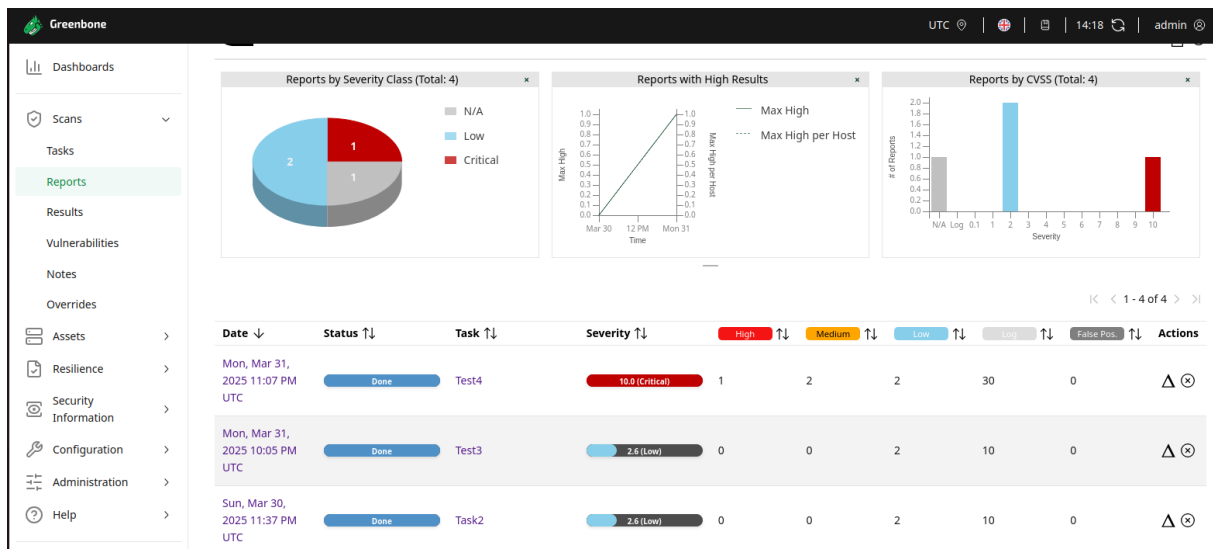
## Ajout du serveur à scanner



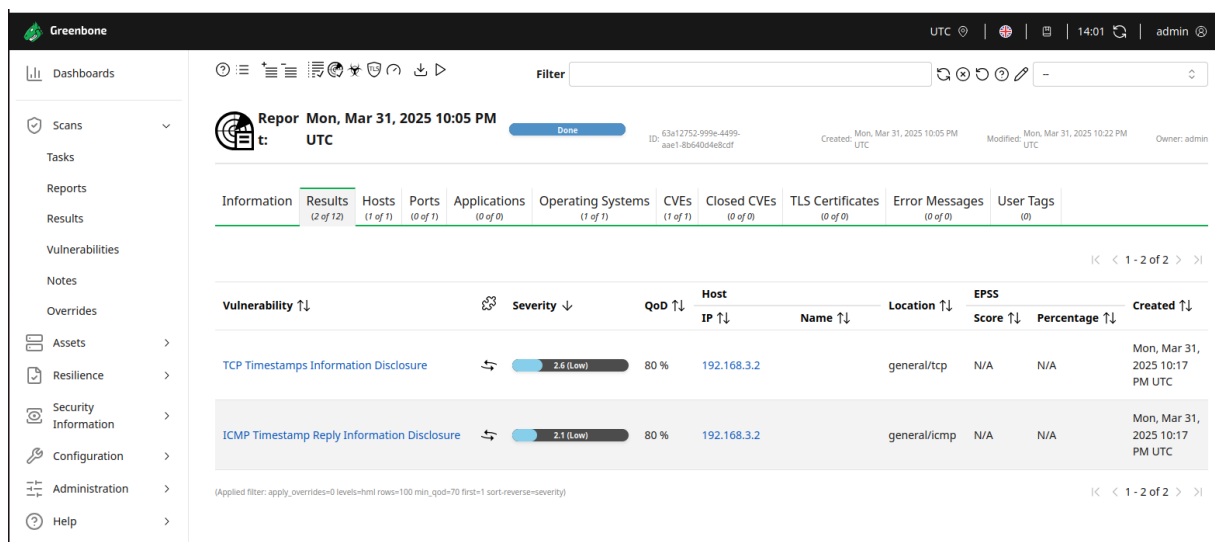
## Création de la tâche pour scanner



## Remontée des vulnérabilités



## Exemple de vulnérabilités



On peut voir ici tous les détails de ce que l'outil remonte sur le serveur, sur les ports, les CVEs...

Exemple détaillé :

Sur ce cas nous avons la réponse d'une requête icmp de type timestamp (Un timestamp ou horodatage) est une information qui indique une date et une heure précises.



ICMP Timestamp Reply Information Disclosure

2.1 (Low)

80 %

192.168.3.2

general/icmp

N/A

N/A

Mon, Mar 31,  
2025 10:17  
PM UTC

### Summary

The remote host responded to an ICMP timestamp request.

### Detection Result

The following response / ICMP packet has been received:

- ICMP Type: 14
- ICMP Code: 0

### Insight

The Timestamp Reply is an ICMP message which replies to a Timestamp message. It consists of the originating timestamp sent by the sender of the Timestamp as well as a receive timestamp and a transmit timestamp.

### Detection Method

Sends an ICMP Timestamp (Type 13) request and checks if a Timestamp Reply (Type 14) is received.

Details: [ICMP Timestamp Reply Information Disclosure OID: 1.3.6.1.4.1.25623.1.0.103190](#)

Version used: 2025-01-21T05:37:33Z

**Impact :** Les timestamps peuvent être utilisés pour mesurer les délais de latence sur le réseau et synchroniser des attaques, comme les attaques de type "man-in-the-middle" ou d'autres formes de surveillance avancée.

**Solution :**

Différentes mesures d'atténuation sont possibles :

- Désactiver complètement la prise en charge des requêtes ICMP timestamp sur l'hôte distant.
- Protéger l'hôte distant avec un pare-feu et bloquer les paquets ICMP qui passent à travers le pare-feu dans les deux directions (soit complètement, soit uniquement pour les réseaux non fiables).

## Evaluation Openvas

| Critère                                             | Description                                                                                                     | (Notes /10) | Commentaires                                                                                                       |
|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-------------|--------------------------------------------------------------------------------------------------------------------|
| Facilité d'installation et d'utilisation            | Simplicité du processus d'installation, accessibilité de l'interface utilisateur                                | 7/10        | Installation via docker facile, mais nécessite des commandes spécifiques et du temps pour la mise à jour des NVTs. |
| Qualité et précision des scans                      | Fiabilité des résultats, taux de faux positifs/négatifs, rapidité d'exécution des scans.                        | 7/10        | Détection correcte des vulnérabilités, mais quelques faux positifs détectés.                                       |
| Taux de détection des vulnérabilités                | Nombre de vulnérabilités détectées par rapport à une base de référence, comparaison avec d'autres scanners.     | 8/10        | Bonne couverture des vulnérabilités connues, mais en dessous de solutions propriétaires comme Nessus.              |
| Pertinence des recommandations de remédiation       | Qualité des solutions proposées pour corriger les failles détectées (exploitation, mise à jour, configuration). | 7/10        | Les correctifs sont clairs, mais certaines recommandations restent génériques.                                     |
| Compatibilité avec l'infrastructure de l'entreprise | Capacité d'OpenVAS à s'adapter aux systèmes et configurations de l'entreprise.                                  | 8/10        | Fonctionne bien sur Linux et Windows, mais nécessite quelques ajustements sur les scans Windows Server.            |