

Rapport d'avancement

PROJET DE FIN D'ETUDES

En vue d'obtention du diplôme

D'Ingénieur d'état

En Génie Télécommunications et Réseaux

**Etude et Mise en place d'une solution d'Audit de
vulnérabilités pour les Datacenters de CGI**

Effectué à

CGI

Technologies et solutions MAROC

Réalisé par

- **MOUSTAHDID Sara**

Encadré à l'ENSAS par :

- **Prof. CHOUKRI Ali**

Encadré à CGI par :

- **Mr. ENNAJAH Youssef**

Année universitaire: 2018/2019

Liste des Figures

Figure 1: Clients de CGI par secteur	4
Figure 2: Projets GTO.....	5
Figure 3: Fiche technique du projet GTO	5
Figure 4: Diagramme de GANTT.....	7
Figure 5:Evaluation et rapport de scan de ELASTIC Detector	19
Figure 6:Vue d'ensemble d'Elastic Detector	19
Figure 7:Stratégies et modèles prédéfinis sur Nessus	20
Figure 8:Rapports personnalisables	21
Figure 9:Vue d'ensemble de Nexpose.....	22
Figure 10:Vue d'ensemble de Retina	23
Figure 11: Vue d'ensemble d'OpenVAS.....	24
Figure 12: Architecture d'OpenVAS.....	28
Figure 13:Démarche de mise en place	31
Figure 14: Conception de la solution.....	32
Figure 15:Spécifications techniques de l'environnement d'installation d'OpenVAS	34
Figure 16: architecture de l'environnement de Test	35
Figure 17: Architecture de simulation sous GNS3.....	35

Table des matières

CHAPITRE I	1
La Genèse duProjet	1
Introduction	1
I. Présentation de CGI technologies et solutions	2
1. Présentation générale	2
2. Principales solutions	3
3. Clients de CGI	3
4. CGI Maroc	4
5. Présentation du Projet GTO.....	5
II. Contexte générale du projet	5
1. Cadre du projet	5
2. Problématique.....	Erreur ! Signet non défini.
3. Cahier des charges.....	6
4. Diagramme de GANTT.....	6
Conclusion	7
CHAPITRE II	8
Audit de vulnérabilités.....	8
I. Audit de Vulnérabilités	9
1. Les phases d’audit de vulnérabilités	9
1.1. Définition du périmètre de l'audit de vulnérabilités	9
1.2. Phase de découverte de vulnérabilités	9
1.3. Analyse des vulnérabilités.....	10
1.4. Mise en place d’un plan de remédiation.....	10
II. Scanner de vulnérabilités.....	10
1. Définition d’un scanneur de vulnérabilité.....	10
2. Principes de fonctionnement	11
3. Cibles	12
4. Méthodes de détection.....	12
1.1. Fingerprint de version.....	12
1.2. Exploitation active	13
1.3. Scan de configuration.....	13
1.4. Scans authentifiés	14
5. Restitution des résultats	15

CHAPITRE III	17
Etude et choix de la solution adéquate.....	17
I. Présentation des scanners de vulnérabilités.....	18
1. Elastic Detector.....	18
2. Nessus	19
3. Nexpose.....	21
4. Retina.....	22
5. OpenVAS.....	24
II. Critère de comparaison.....	24
III. Tableau comparatif	25
IV. Présentation d'OpenVAS.....	26
1. Architecture d'OpenVAS	28
CHAPITRE IV	30
Pré-étude et implémentation	30
I. Pré-étude	31
1. Démarche de mise en place.....	31
2. Etude de faisabilité	31
3. Etude conceptuelle.....	31
3.1. Critères de choix	32
3.2. Architecture de la solution réseau choisi.....	32
3.3. Les équipements réseau	32
4. Prérequis techniques	34
II. Implémentation virtuelle.....	34
1. Configuration de CentOS 7	35
2. Installation d'OpenVAS (GVM 10)	38
Annexe.....	43
I. Configuration vSRX.....	44

CHAPITRE I

La Genèse du Projet

Introduction

Dans ce premier chapitre nous présentons l'organisme d'accueil, Groupe CGI Maroc, ses activités et sa structure interne. Ensuite nous aborderons le cadre général du projet, ses objectifs et les missions effectuées ainsi que la planification en élaboration du diagramme de Gantt.

I. Présentation de CGI technologies et solutions

1. Présentation générale

CGI est l'acronyme de « Conseillers en gestion et informatique », c'est un groupe canadien reconnu pour être parmi les leaders dans le domaine des technologies de l'information et de la communication et en gestion des processus d'affaires. Le groupe a vu le jour en 1976 grâce à Serge Godin et André Imbeau, en tant que petite firme personnelle dans le sous-sol de la résidence de Serge spécialisée dans la consultation informatique pour les administrations et les industriels au Québec.

Grâce à ses 74000 professionnels répartis dans divers emplacements dans le monde, CGI aide ses clients à devenir des organisations numériques centrées sur le client, en offrant des services conseils en management ainsi que des services d'intégration de systèmes de grande qualité. Ces services sont conjugués à plus de 150 solutions de propriété intellectuelle afin d'aider ses clients à devenir des organisations numériques de bout en bout.

CGI se classe parmi les cinq plus grands groupes mondiaux dans son secteur, Groupe CGI fait partie de la liste Forbes Global 2000, un classement des 2000 plus grandes entreprises au monde publié par le magazine économique américain Forbes⁶. Elle fait également partie du S&P/TSX 60(*indice boursier*), la liste des 60 plus grandes entreprises canadiennes par capitalisation boursière.

Le groupe se caractérise actuellement par son étendue géographique qui englobe les 5 continents, en étant présents dans 40 pays dont le Maroc, tout en conservant un indice de satisfaction client très élevé (9/10). Cet indice de satisfaction est possible grâce entre autres au respect des échéances et des budgets pour plus de 95% des projets.

De plus, parmi les clients de CGI, nous pouvons trouver 39 entreprises du CAC 40(*indice boursier*) ainsi que de nombreuses grandes entreprises dans le monde entier avec un carnet de commandes de plus de 20 milliards de dollars et un revenu annuel de plus de 10 milliards.

2. Principales solutions

CGI met à profit ses connaissances sectorielles approfondies et son expertise technologique afin de fournir à ses clients des repères dans les dédales, qui constitue la numérisation des ressources humaines, des processus et des technologies.

CGI offre une gamme de services dont :

- Services conseils-stratégiques en technologie.
- Services d'intégration des systèmes.
- Développement et gestion des applications.
- Services d'infrastructures.
- Gestion des processus d'affaires.
- Solutions d'affaires exclusives.

3. Clients de CGI

CGI compte 5000 clients en services complets et 25000 clients en services en gestion des processus d'affaires et propriété intellectuelle. Elle intervient principalement dans les secteurs économiques présents sur la figure ci-dessous :



SECTEUR MANUFACTURIER	PÉTROLE ET GAZ	COMMERCE DE DÉTAIL ET SERVICES AUX CONSOMMATEURS	TRANSPORT ET LOGISTIQUE	SERVICES PUBLICS

Figure 1: Clients de CGI par secteur

4. CGI Maroc

Présente au Maroc depuis 2004, sous le cap de l'ex SSII multinationale Logica, avant d'être réaffecté au Groupe CGI suite à l'absorption de cette dernière pour l'ex-logica.

CGI Maroc est une BU de la SBU Fr-Ma-Lu(France-Maroc-Luxembourg), dont les activités se concentrent essentiellement autour de l'offshoring des projets de la SBU Francophone de CGI. Présente sur 3 sites, à Casa Near Shore, au Technopolis de Rabat et Fès, caractérisée par son modèle de gouvernance inspiré de la norme **ISO 9001**, et par son modèle de sécurité de l'information inspiré également de la norme **ISO 27001**, CGI Maroc assure un nombre considérable des métiers appartenant à la sphère des activités CGI.

CGI Maroc emploie plus de 700 fonctionnels majoritairement issus de formations d'ingénierie ou techniques en relation avec la technologie de l'information pour répondre au besoin des clients avec des solutions robustes.

Les sièges de service de CGI Maroc sont répartis en 3 sièges :

CGI Casablanca : Casa NearShore, Bât. Shore 5, 2ème étage, Casablanca, contient plus que 450 membres et abrite la majorité des projets actifs de CGI Maroc ainsi que les services administratifs.

CGI Rabat : Technopolis, Bât. Shore 3 Sala El Jadida, contient plus que 250 membres et abrite quelques extensions de projets situés au site de Casablanca ainsi que des projets propres au site.

CGI Fès : Fès Shore Park, Route de Sidi Hrazem.

5. Présentation du Projet GTO

GTO est l'acronyme de Global Technology Operations, c'est un projet qui a démarré en 2016 sur les 3 sites de CGI Maroc, et qui regroupe l'ensemble des activités techniques et infrastructures de CGI France.

GTO travaille actuellement sur 6 projets à savoir :



Figure 2: Projets GTO

Ces références touchent chaque secteur d'activités auxquels s'adressent le projet GTO :

- Energie et services
- Manufacturing
- Finance /Banques/ Assurances
- Retail
- Télécommunications /Transport
- Secteur publique

Ce projet possède une production dédiée aux activités de pilotage des services Infrastructures réseaux et cloud solution et englobe des équipes qui travaillent sur divers services réseaux tel que l'équipe SUPPORT qui gère la partie Réseau des services fournis aux clients, l'équipe virtualisation, l'équipe base de données et le service IT.

Projet	GTO
Collaborateurs	270 (experts et architectes)
Sites	Casablanca- Rabat-Fès
Certifications	ITIL

Figure 3: Fiche technique du projet GTO

II. Contexte générale du projet

1. Cadre du projet

Ce projet rentre dans le cadre d'un projet de fin d'études réalisé au sein de l'entreprise CGI, et principalement au profit du projet GTO.

- Stagiaire à l'œuvre : MOUSTAHFID Sara
- Responsable du projet : MIRI Saoussane
- Encadrant professionnel : ENNAJAH Youssef
- Encadrant pédagogique : CHOUKRI Ali
- Durée du projet : 6 mois, Février-Août

2. Cahier des charges

L'objectif principal est de mettre en place une solution d'Audit de vulnérabilités propre au projet GTO .

Le projet consiste à mener les actions suivantes :

- Etude des solutions de Scan de vulnérabilités.
- Conception d'une architecture sécurisée pour la solution.
- Installation et Configuration de la solution sur un environnement de Test.
- Mise en place de la solution sur le Datacenter VDR.
- Tests du bon fonctionnement de la solution.
- Traitement et mise en place des correctifs.

3. Diagramme de GANTT

La planification du projet est par définition une organisation du déroulement des différentes étapes d'un projet. Elle permet de suivre la concrétisation des objectifs et la réalisation des différentes tâches pour fournir une vision globale du projet et de son déroulement.

Parmi les méthodes de planification de projet, j'ai choisi d'utiliser le diagramme de GANTT. Il permet de superviser aussi bien la réalisation d'une tâche unique, avec ses dates de début et de fin et la durée qui lui a déjà été consacrée, que l'avancement du projet dans son ensemble, avec les répercussions de chaque tâche sur la date de fin du projet.

Ci-dessous la figure représentant l'ensemble des tâches à accomplir lors de ce stage.

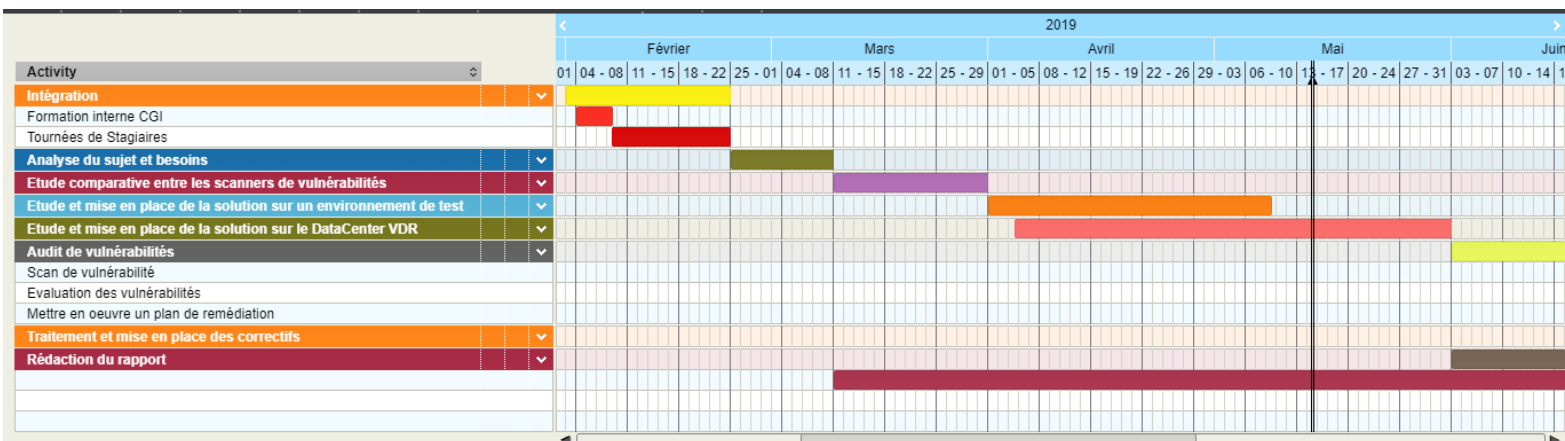


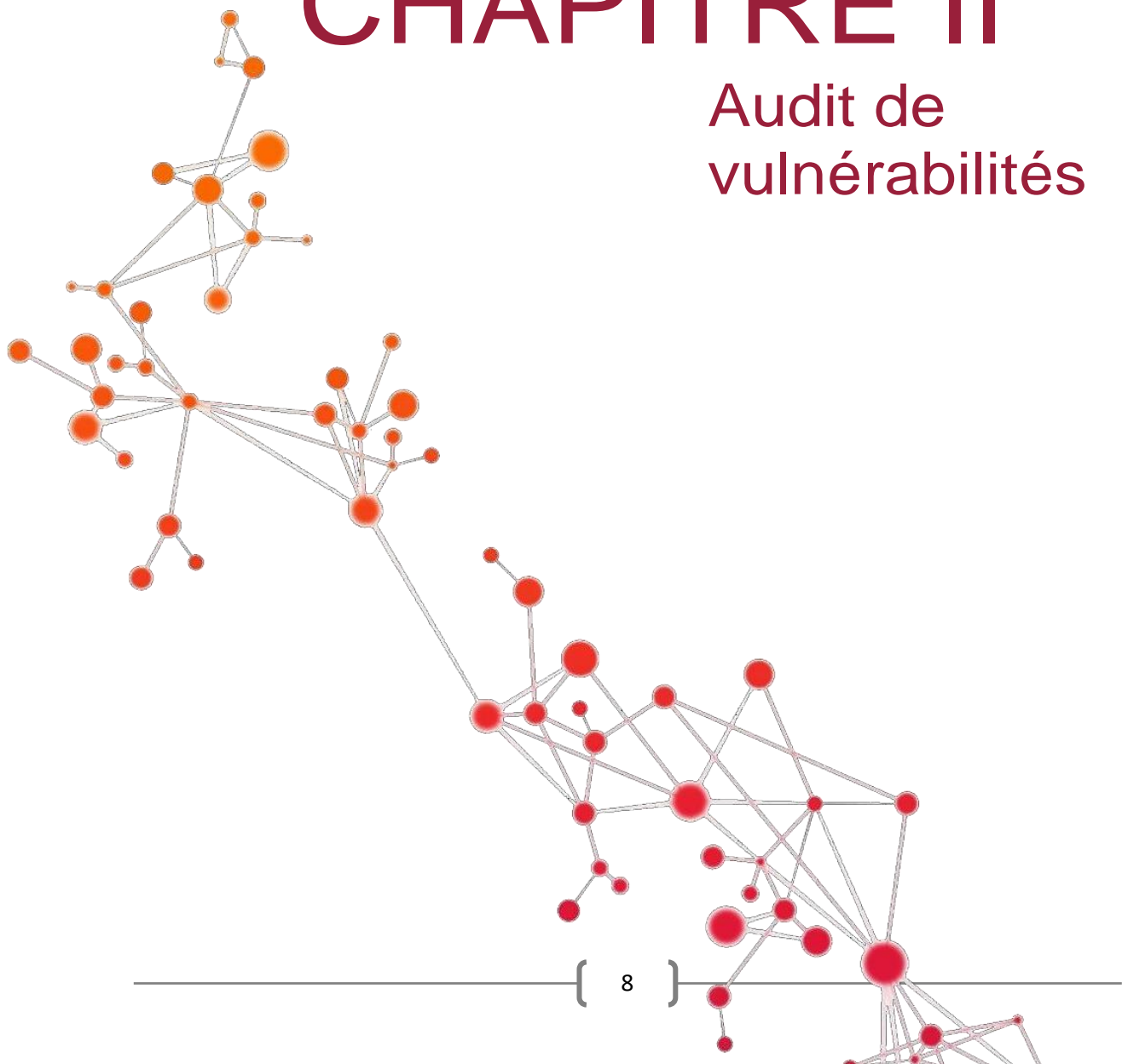
Figure 4: Diagramme de GANTT

Conclusion

Nous avons défini, au cours de ce chapitre, dans un premier lieu l'organisme d'accueil ensuite le contexte général du projet selon ses différents aspects. Après nous avons présenté la méthodologie de travail et le diagramme de GANTT de notre projet à l'heure actuel

CHAPITRE II

Audit de vulnérabilités



I. Audit de Vulnérabilités

L'audit de vulnérabilités a pour but de mesurer le niveau de sécurité d'un système ou d'un périmètre défini, de déterminer les failles de sécurité et faiblesses dans les mécanismes de sécurité et de pouvoir ainsi définir le degré d'exposition aux risques et menaces et de mettre en œuvre un plan de remédiation avec des actions correctives.

1. Les phases d'audit de vulnérabilités

1.1. Définition du périmètre de l'audit de vulnérabilités

Il est crucial de définir le périmètre de l'audit de vulnérabilités afin d'analyser un ou plusieurs éléments souhaités, sans impacter d'éléments tiers. L'objectif de cette phase consiste en la récolte d'informations sur le périmètre à analyser et son exposition externe grâce à la validation de l'accessibilité des composants inscrits au périmètre et l'identification d'outils qui seront utilisés durant l'audit de vulnérabilité. Par exemple, un audit d'infrastructure, un audit applicatif ou un audit de serveur web auront des approches différentes, définies par les consultants afin d'appréhender au mieux chaque scénario et chaque environnement donné.

1.2. Phase de découverte de vulnérabilités

L'objectif de cette phase consiste en la recherche de vulnérabilités potentielles sur les composants inscrits au périmètre et validés à la phase précédente. Cette recherche est conduite à l'aide d'un panel d'outils spécifiques, tels **les scanners de vulnérabilités**.

1.3. Analyse des vulnérabilités

L'objectif de cette phase consiste en l'approfondissement de l'analyse dans le but d'identifier des vulnérabilités spécifiques nécessitant une expertise et une investigation plus complète. Les résultats de l'audit de vulnérabilités technique sont décrits au sein de deux rapports distincts:

- un rapport destiné aux équipes techniques: listant de manière exhaustive les éléments analysés et les résultats obtenus. Décrire un plan d'action de remédiation prenant en compte les niveaux de priorité, et aussi décrire les recommandations de correction et les moyens à mettre en œuvre pour corriger les points de vulnérabilités détectés.
- un second rapport est destiné à l'équipe managériale de l'entreprise: ce document de synthèse donne une vision opérationnelle globale, et permet de prendre rapidement connaissance du degré d'exposition aux risques et menaces du système étudié.

1.4. Mise en place d'un plan de remédiation

Cette dernière phase a pour but de gérer au mieux les interventions qui font suite aux découvertes. Certaines solutions d'audits de vulnérabilités fournissent une plateforme d'attribution de ticket lors de la découverte d'une vulnérabilité. Le ticket adressé à un ingénieur lui permet de mettre en place une action curative et de tracer les événements de remédiation. Le processus de remédiation doit être l'aboutissement d'un audit de vulnérabilités.

II. Scanner de vulnérabilités

1. Définition d'un scanner de vulnérabilité

Un scanner de vulnérabilités est un outil conçu pour identifier des faiblesses dans une application, un système d'exploitation, ou un réseau permettant à un individu mal attentionné de les exploiter. Ces programmes proposent en général un lien qui détail la faille de sécurité et explique comment la corriger.

Les scanners de vulnérabilités sont répartis en 3 parties :

- **Les scanners spécialisés** : permettent d'analyser un équipement informatique précis. Cela permet de découvrir des vulnérabilités sur une partie précise
- **Les scanners complets** : Les scanners complets permettent d'analyser un ensemble d'équipement SI pour également découvrir les vulnérabilités. Cela permet d'avoir une

unique interface pour scanner les différents structures SI. Cela fait qu'il est plus aisé et rapide de scanner un réseau informatique.

- **Les scanners continus** : permettent d'avoir une analyse constante du réseau et ainsi de découvrir plus rapidement les vulnérabilités d'un réseau entier. Cette méthode est la plus couteuse en ressource humaine et technique mais elle est également la solution la plus réactive.

Bien évidemment, une solution peut être continue et spécialisée ou continue et complète.

2. Principes de fonctionnement

Les scanners de vulnérabilités se présentent sous plusieurs formes:

- logiciel à installer sur son système,
- machine virtuelle pré-configurée (*virtual appliance*)
- ou encore en SaaS dans le *cloud*.

Un scanner de vulnérabilités se "lance" sur une ou plusieurs cibles, dans un réseau interne ou sur Internet. Ces cibles (URL, adresse IP, sous-réseau) sont renseignées par l'utilisateur lorsqu'il désire mener son scan.

La plupart des outils suivent le schéma de scan suivant:

- Détection des cibles actives (attente d'une réponse ICMP, ARP, TCP, etc. pour déterminer si la cible répondra au scanner).
- Détection des ports TCP et UDP accessibles sur la cible (scan de ports).
- Détection des services actifs (SSH, HTTP, etc.) sur chacun de ces ports et de leurs versions (phase de "*fingerprint*").
- Eventuellement: utilisation d'un compte fourni pour se connecter sur la machine et lister les programmes non visibles depuis le réseau (navigateur, suite bureautique, etc.).
- Eventuellement: reconnaissance des applications Web accessibles et construction de l'arbre de chaque site Web (phase dite de "*crawling*").
- Sélection des modules de sécurité à lancer sur la cible selon les services précédemment reconnus.
- Lancement des modules de sécurité.
- Génération du rapport de sécurité.

Un scanner de vulnérabilités est donc un outil complexe qui peut faire appel à de nombreux programmes spécifiques pour chacune des tâches précitées.

3. Cibles

Un scanner de vulnérabilités est théoriquement capable de tester tout élément joignable par une adresse IP.

Le fait de pouvoir joindre un élément n'implique cependant pas forcément que son niveau de sécurité puisse être audité correctement. Pour cela, le scanner doit embarquer les modules de sécurité idoines dans son catalogue.

4. Méthodes de détection

Pour établir la présence d'une vulnérabilité, un scanner dispose de plusieurs méthodes:

1.1. Fingerprint de version

Cette première méthode est la plus répandue. L'outil tente de déterminer la nature et la version d'un service.

Pour cela, il peut se fier aux bannières présentées par les différents services, rechercher des motifs caractéristiques d'une version dans les réponses du serveur (notamment dans les en-têtes), etc.

Une fois que la version est déterminée, l'outil utilise une base qui associe à chaque version d'un outil, la liste des vulnérabilités publiquement connues sur celui-ci.

De telles bases sont publiquement accessibles et enrichies par l'éditeur de la solution concernée, par un État ou encore par une communauté.

- La base la plus connue est celle des CVE: Common Vulnerabilities and Exposures publiée par le MITRE, une organisation à but non lucratif soutenue par le département de la Sécurité intérieure des États-Unis.
- Une alternative indépendante est l'OSVDB (Open Source Vulnerability Database).

Ces bases tentent donc de recenser toutes les vulnérabilités découvertes et de les associer aux produits vulnérables. Généralement seuls les produits les plus connus font l'objet d'une entrée dans ces deux bases.

Les éditeurs les plus connus maintiennent aussi leur propres base:

- Debian Security Advisory (DSA) pour Debian,
- CSA pour Cisco,
- RHSA pour RedHat, *idem* pour Wordpress, Gentoo, Drupal, etc.

De manière symétrique, la dénomination des programmes et de leurs versions a été normalisée avec les CPE (*Common Platform Enumeration* également maintenu par le MITRE) afin de pouvoir faire des associations en base de données plus simple.

Cette méthode est cependant notoirement sujette aux faux positifs et aux faux négatifs. De nombreux programmes ne laissent pas transparaître leur version (désactivation des bannières, etc.), le scanner ne pourra donc pas en déduire leurs vulnérabilités (faux négatifs). Inversement, des programmes peuvent donner leur version mais avoir subi un rétroportage des correctifs et donc, ne pas souffrir des vulnérabilités officiellement rattachées à cette version (faux positifs). Ce dernier cas se produit très fréquemment sur les versions paquetées par les distributions Linux comme Debian ou Ubuntu.

En outre, les informations fournies par la base CVE et par la normalisation CPE ne sont pas toujours suffisantes pour identifier la présence d'une vulnérabilité. Par exemple, dans le cas d'une vulnérabilité comme la CVE-2017-0143 (liée à l'attaque WannaCry), on constate que toute machine Microsoft Windows 8.1 utilisant SMB v1 est vulnérable d'après les informations fournies par le NVD et la terminologie CPE. Or, une telle machine disposant des correctifs de sécurité appropriés (KB4012213) sera bel et bien protégée de cette vulnérabilité, et représentera ainsi un "faux-positif". Il faut donc affiner une telle analyse à partir des alertes de sécurité en provenance des éditeurs, qui indiquent de manière plus précise les conditions de vulnérabilité.

1.2. Exploitation active

Lorsque des vulnérabilités sont publiquement dévoilées, elles sont parfois accompagnées d'un "exploit" qui est un programme permettant de les exploiter automatiquement.

Un scanner de vulnérabilités peut donc recourir à cet exploit pour vérifier la présence d'une vulnérabilité. Dans ce cas-là, le scanner n'a pas besoin de se fier à la version du programme qu'il audit, il se fie à la réussite ou à l'échec de l'exploit.

1.3. Scan de configuration

Certaines vulnérabilités ne proviennent pas d'un défaut dans le code source du programme mais simplement d'un mauvais réglage de celui-ci dans un certain contexte. Les scanners de

vulnérabilités peuvent donc détecter certaines vulnérabilités en analysant la configuration distante du service audité (ce qui est un service généralement prévu par celui-ci).

Ceci concerne par exemple les options de sécurité activées sur les cookies, les cipher suite dans la configuration SSL/TLS, les transferts de zone pour un serveur DNS, les mots de passe par défaut inchangés, les services par défaut laissés activés, etc.

Cette méthode est généralement sans risque (excepté tester les mots de passe par défaut qui peut bloquer des comptes après trop d'échecs) et la base des bonnes pratiques est relativement simple à maintenir (par rapport à celle des nouvelles vulnérabilités où l'on compte en moyenne 13 nouvelles CVE par jour). Cependant, il y a des risques de faux positifs puisque le scanner de vulnérabilités n'est pas forcément à même de connaître le contexte et donc de juger si la configuration en question est adaptée ou non (exemple: un transfert de zone DNS est problématique vers Internet mais relativement sans danger dans un réseau interne).

1.4. Scans authentifiés

Bien que la plupart des scanners soient "sans agents", ils fournissent souvent la possibilité d'utiliser un compte renseigné par l'utilisateur pour mener des tests authentifiés.

Lors d'un scan authentifié, le scanner utilise des modules appelés "*local security check*" qui consistent à consulter la base des programmes installés et leur version (en interrogeant le registre Windows ou Aptitude, pacman, emerge, etc. pour Linux).

L'intérêt principal de cette méthode est de ne pas se limiter à la surface visible de la machine depuis le réseau. En effet, une machine ayant un navigateur obsolète pourrait facilement être compromise par un maliciel, mais le navigateur étant un programme "client" et non pas "serveur", il n'émet pas sur un port (puisque'il consomme un service) comme le ferait par exemple un serveur SSH (qui fournit un service). Donc, en l'absence de scan authentifié, le risque peut demeurer totalement ignoré de l'utilisateur du scanner de vulnérabilités.

Avec cette méthode, le scanner de vulnérabilités remplit donc également une fonction proche du "*patch management*" (il permet la détection des anomalies même s'il ne les corrige pas lui-même).

Cette méthode est fiable (pas ou peu de faux positifs/négatifs), ne risque pas d'engendrer d'instabilités sur le système audité et est rapide. Elle peut se substituer aux méthodes de fingerprint et d'exploitation active mais pas à celle du scan de configuration qui demeure

complémentaire. En revanche, elle nécessite de fournir des comptes valides pour chaque machine auditée.

5. Restitution des résultats

La visualisation et la restitution des résultats se fait traditionnellement via deux paradigmes.

- Premièrement, une vue "par vulnérabilité" permettant de lister toutes les vulnérabilités identifiées dans le scan et de donner pour chacune d'elles la liste des machines affectées.
- Deuxièmement, une vue "par machine" listant les cibles de l'audit associées à la liste de leur vulnérabilités respectives.

Traditionnellement, en sécurité informatique, les vulnérabilités sont restituées par ordre de criticité suivant une échelle à 4 niveaux:

- ✓ Critiques: les vulnérabilités permettant généralement une prise de contrôle ou une exécution de commande à distance dont l'exploitation est relativement simple
- ✓ Majeures: les vulnérabilités permettant une prise de contrôle ou une exécution de commande à distance dont l'exploitation demeure complexe
- ✓ Moyennes: les vulnérabilités ayant des impacts limités ou dont l'exploitation nécessite des conditions initiales non triviales
- ✓ Mineures: les vulnérabilités ayant des impacts faibles ou nuls à moins d'être combinées à d'autres vulnérabilités plus importantes

L'état de l'art associe à chaque vulnérabilité un score entre 0 et 10 appelé CVSS (*Common Vulnerability Scoring System*) qui dépend des caractéristiques de cette vulnérabilité. La version 3 du CVSS prend en compte, au minimum, les éléments suivants :

- ✓ Le vecteur d'attaque : est-ce que l'attaquant peut venir de n'importe où ou bien doit-il avoir une position de départ privilégiée ?
- ✓ Complexité : exploiter la vulnérabilité est-il trivial (par exemple si un exploit existe) ou bien hautement techniques ?
- ✓ Privilèges requis : l'attaquant doit-il disposer d'accès préalables (un compte utilisateur par exemple) pour pouvoir mener son action ?
- ✓ Interaction avec un utilisateur: l'attaquant doit-il amener la victime à effectuer une action pour que son attaque réussisse (comme l'inciter à cliquer sur un lien) ?

- ✓ Périmètre : est-ce qu'une exploitation permet à l'attaquant d'avoir accès à de nouvelles cibles ?
- ✓ Impacts : une exploitation réussie entraîne-t-elle des pertes de confidentialité/disponibilité/intégrité ?

Le score CVSS est régulièrement utilisé par les scanners de vulnérabilités pour pondérer les risques associés à une cible.

Les scanners doivent donner à l'utilisateur tous les éléments pertinents pour sa compréhension de la vulnérabilité. Traditionnellement, une description de vulnérabilité comporte les éléments suivants :

- Le nom de la vulnérabilité.
- Sa criticité.
- La cible touchée.
- Une brève description de sa nature.
- Une référence à une base de connaissance type CVE, DSA...
- Une mention de la simplicité de l'exploitation.
- Une description de l'impact en cas d'exploitation réussie.
- Une ou plusieurs préconisations pour la résoudre.

Parfois, d'autres éléments y sont ajoutés:

- Le niveau de confiance à accorder à la vulnérabilité : quantification du risque qu'il s'agisse ou non d'un faux positif.
- S'il existe ou non un exploit automatique.
- Un extrait des données ayant permis au module de sécurité de conclure à la présence de cette faille.
- La famille de vulnérabilité (authentification, mise à jour, etc.).
- Des liens pour en savoir plus (notamment pour des explications plus détaillées du fonctionnement technique de la vulnérabilité).
- Les rapports sont souvent exportables aux formats PDF, CSV, HTML, etc.

CHAPITRE III

Etude et choix de la solution adéquate



I. Présentation des scanners de vulnérabilités

1. Elastic Detector

Elastic Detector est un outil qui permet de scanner le réseau, les serveurs, les applications, les systèmes d'exploitation et les données de manière continue et fonctionne de manière automatique sans intervention extérieure. L'outil est très simple à utiliser et a une interface très facile à comprendre.

Les Réseaux et serveurs sont découverts automatiquement dans les environnements Cloud, virtuels, physiques et hybrides.

Elastic Detector surpasse les scanners de vulnérabilités existants par la mise en œuvre d'un nouveau modèle d'audit de sécurité breveté : la surveillance continue et adaptative.

Cet outil propose :

- Mise à jour et lancement automatiquement des tests de sécurité : Nouveaux serveurs immédiatement surveillés.
- Analyse approfondie de l'intérieur du serveur : Analyse poussée sans impact sur la production et moins de faux-positifs.
- Un fonctionnement sans agent : pas besoin d'opération de déploiement ou de maintenance et pas de ressources utilisées.
- Rapports détaillés en temps réel : Rapports configurables, indicateurs clés de risque, proposition de solutions.
- Connaître à tout moment son état d'exposition aux vulnérabilités pour pouvoir agir et prendre des décisions bien informées.
- Il n'y a pas d'impact sur les performances des serveurs car ils sont dupliqués avant leurs scans.
- Le système de monitoring, de génération de rapport et d'alerte est configurable.

Scan Reports ?

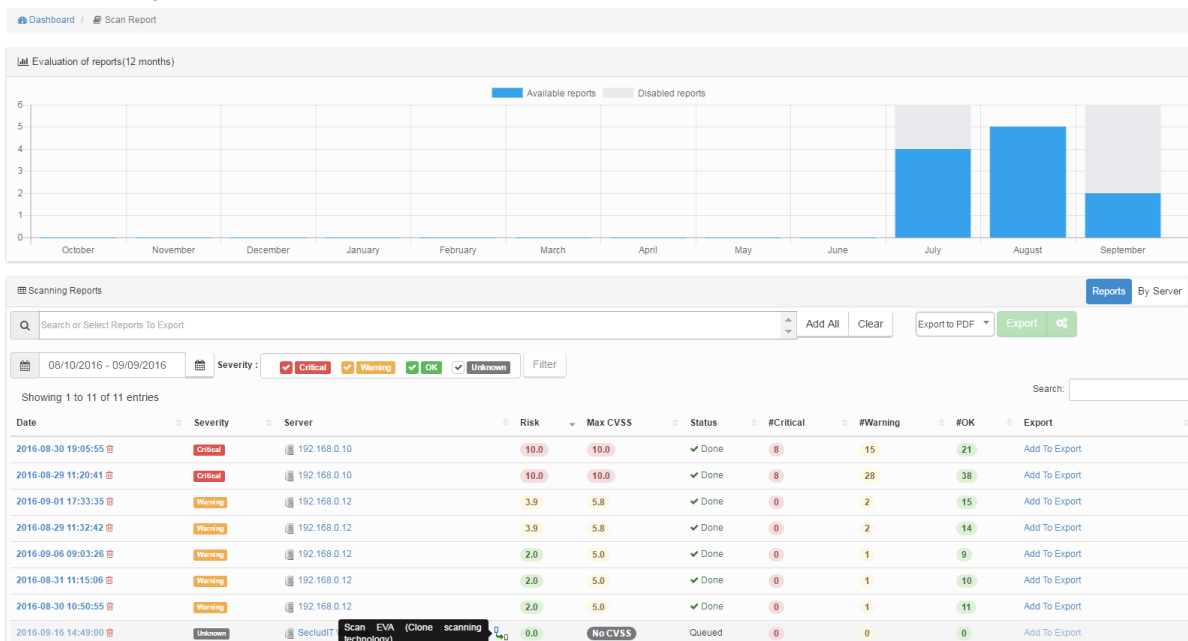


Figure 5: Evaluation et rapport de scan de ELASTIC Detector

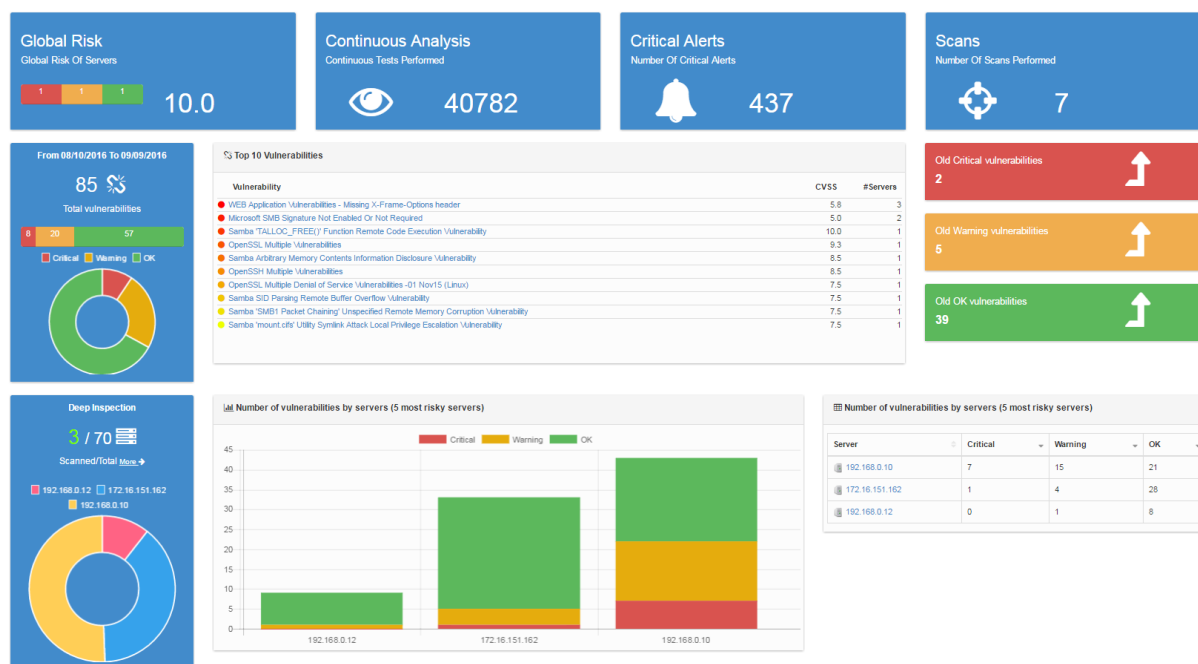


Figure 6: Vue d'ensemble d'Elastic Detector

2. Nessus

Nessus est un outil de scan de vulnérabilité qui permet de rechercher les différentes failles existantes au sein d'un réseau d'un système d'information. Cet outil permet de détecter les systèmes qui ne sont pas à jour, les mots et les services de passes jugés faibles, les mauvaises configurations et les failles de sécurité.

Ce logiciel :

- Détecte les équipements branchés sur le réseau
- Vérifie la sécurité d'un point de vu extérieur en observant le réseau et en déduisant les logiciels et leurs versions utilisés grâce à leurs bannières.
- Se connecte à un ordinateur en tant qu'utilisateur pour les scanner de manières plus pertinentes.
- Présente un rapport clair et facilement exportable sous différents formats.
 - Présente de nombreuses attaques pour des attaques (Déni de services, attaques simples, attaques destructives etc.).

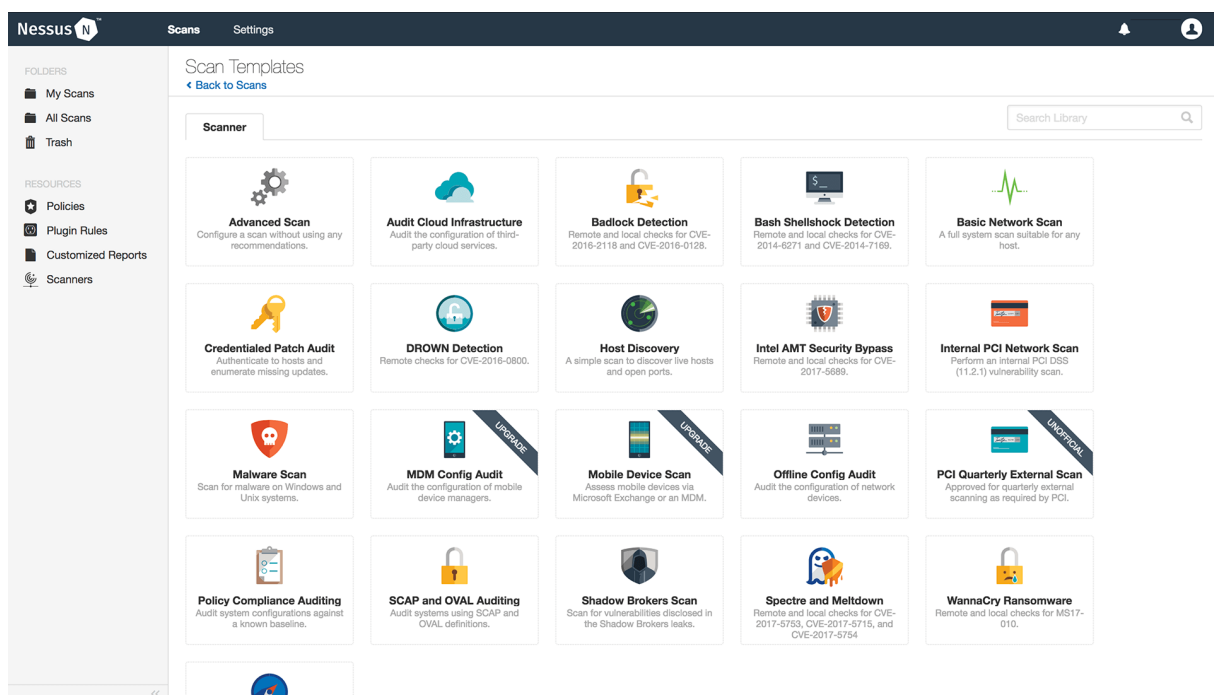


Figure 7:Stratégies et modèles prédéfinis sur Nessus

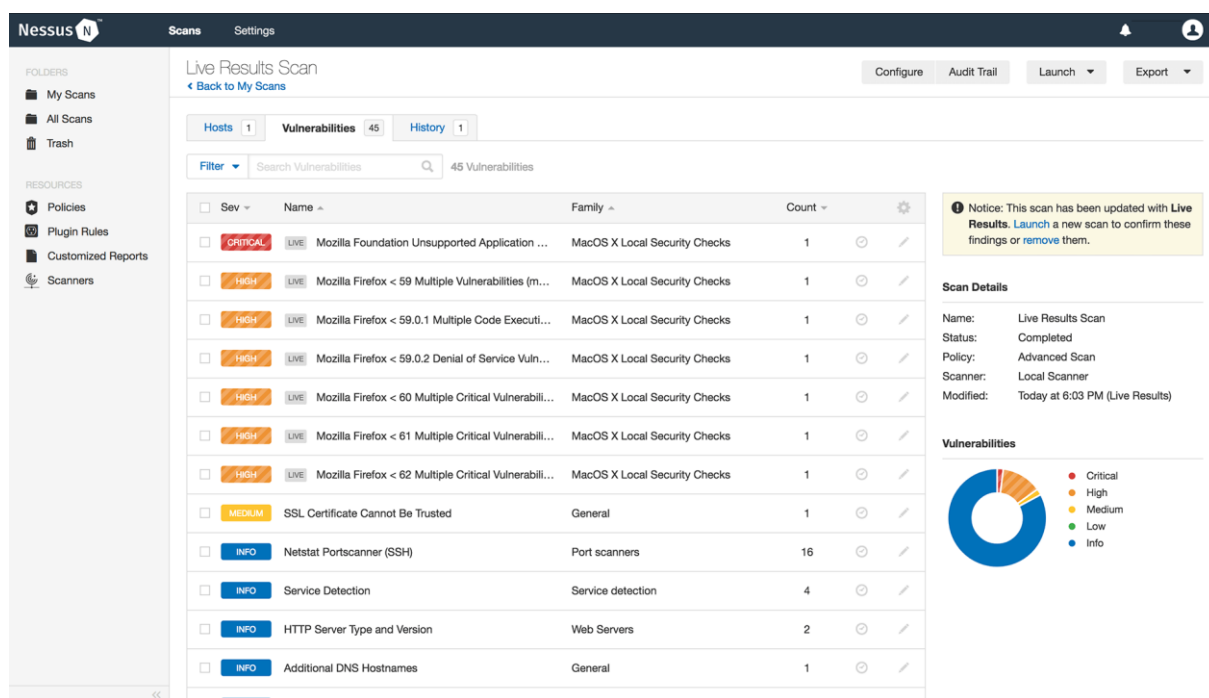


Figure 8: Rapports personnalisables

3. Nexpose

Nexpose est un outil qui permet de scanner les vulnérabilités d'un système informatique (Scan du réseau, d'OS, BDD, application web et environnement virtuel) et vise à prendre en charge l'ensemble du cycle de vie de la gestion de la vulnérabilité, y compris la découverte, la détection, la vérification, la classification des risques, l'analyse d'impact, les rapports et les mesures d'atténuation.. C'est un outil qui fonctionne en continu et détecte automatiquement la présence de nouveaux équipements lors de leurs connexions. Le logiciel est très réactif, il se met à jour très régulièrement.

Le logiciel est facile à utiliser grâce à une bonne documentation et une interface très claire et épuré. Les vulnérabilités détectées ont un score de 0 à 1000 pour nous aider à comprendre quelle vulnérabilité est la plus importante à traiter.

Enfin il est possible de créer des groupes pour que chaque groupe ait son rapport et sache ce qu'il doit faire. Les informations étant mise à jour en temps réelle, dès qu'une faille est repérée, l'équipe concernée par la faille reçoit le rapport et peu rapidement traiter le problème.

De plus, cette outil est développé par Rapid7 et peut être facilement complémentaire d'autre outil de cette compagnie comme Metasploit pour l'exploitation des vulnérabilités.

Il est vendu sous forme de logiciel autonome, d'Appliance, de machine virtuelle ou de déploiement de service géré ou de cloud privé. L'interaction de l'utilisateur se fait via un navigateur Web. Il existe une édition communautaire gratuite qui permet d'analyser 32 hôtes, ainsi que des versions commerciales qui commencent à 2 000 USD par utilisateur et par an.

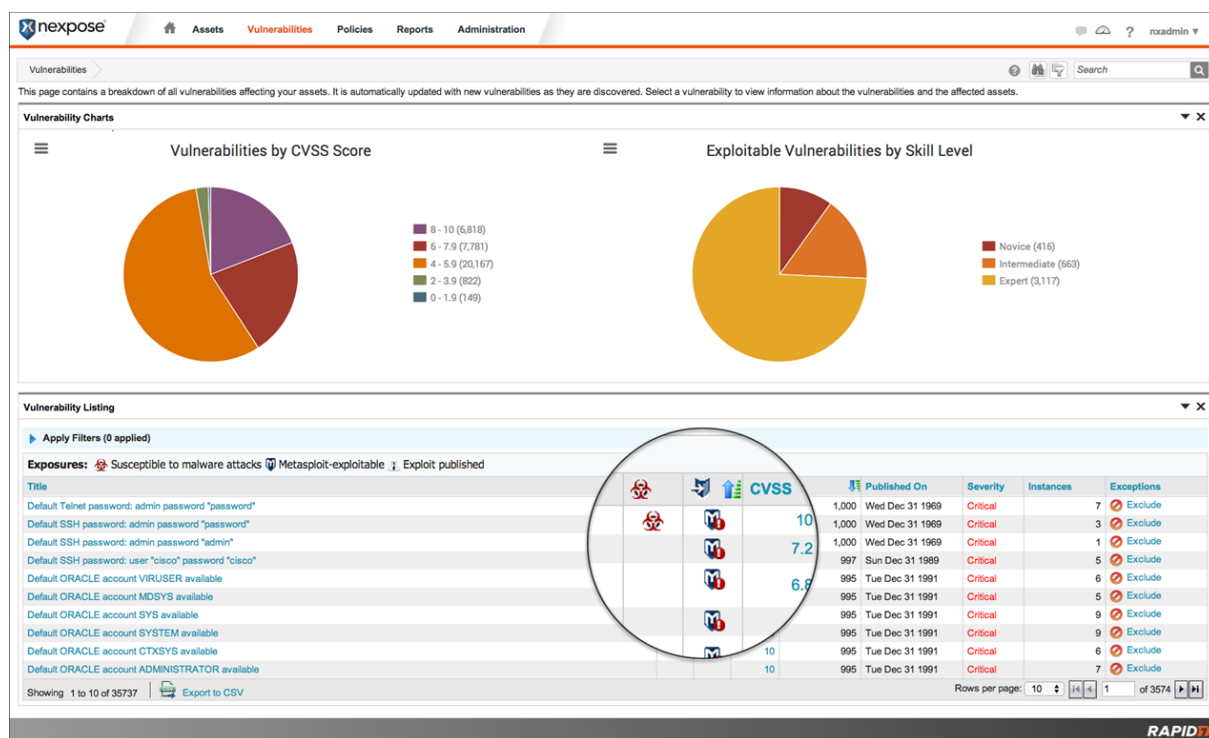


Figure 9: Vue d'ensemble de Nexpose

4. Retina

Retina Network Security Scanner est la solution d'évaluation de la vulnérabilité la plus sophistiquée du marché. Disponible en tant qu'application autonome ou en tant que composant de la plate-forme de gestion de vulnérabilités unifiée Retina CS, Retina Network Security Scanner vous permet d'identifier efficacement les expositions informatiques et de hiérarchiser les mesures de correction à l'échelle de l'entreprise.

Retina Network Security Scanner permet de :

- Découvrir tous les actifs réseau (locaux et distants), Web, de base de données et virtuels.
- Révéler des informations personnelles identifiables (PII) et autres données sensibles à risque.

- Identifier les vulnérabilités du système, des applications, des bases de données, du système d'exploitation et des applications Web via un contrôle basé sur et / ou sans agent.
- Évaluer les risques et hiérarchiser les mesures de correction en fonction de l'exploitabilité (à partir de Core Impact®, Metasploit®, Exploit-db), de CVSS et d'autres facteurs.
- Confirmer l'exploitabilité au moyen de tests de pénétration, en un clic dans le framework Metasploit.
- Signaler les progrès et les résultats à des collègues dans les domaines de la gestion, de la conformité, de la vérification, des risques et autres.
- Analyser les menaces et obtenez des informations sur la sécurité via la console de gestion des vulnérabilités optionnelle Retina CS.
- Partager des données avec des solutions populaires pour SIEM, GRC et d'autres plateformes de gestion de la sécurité.

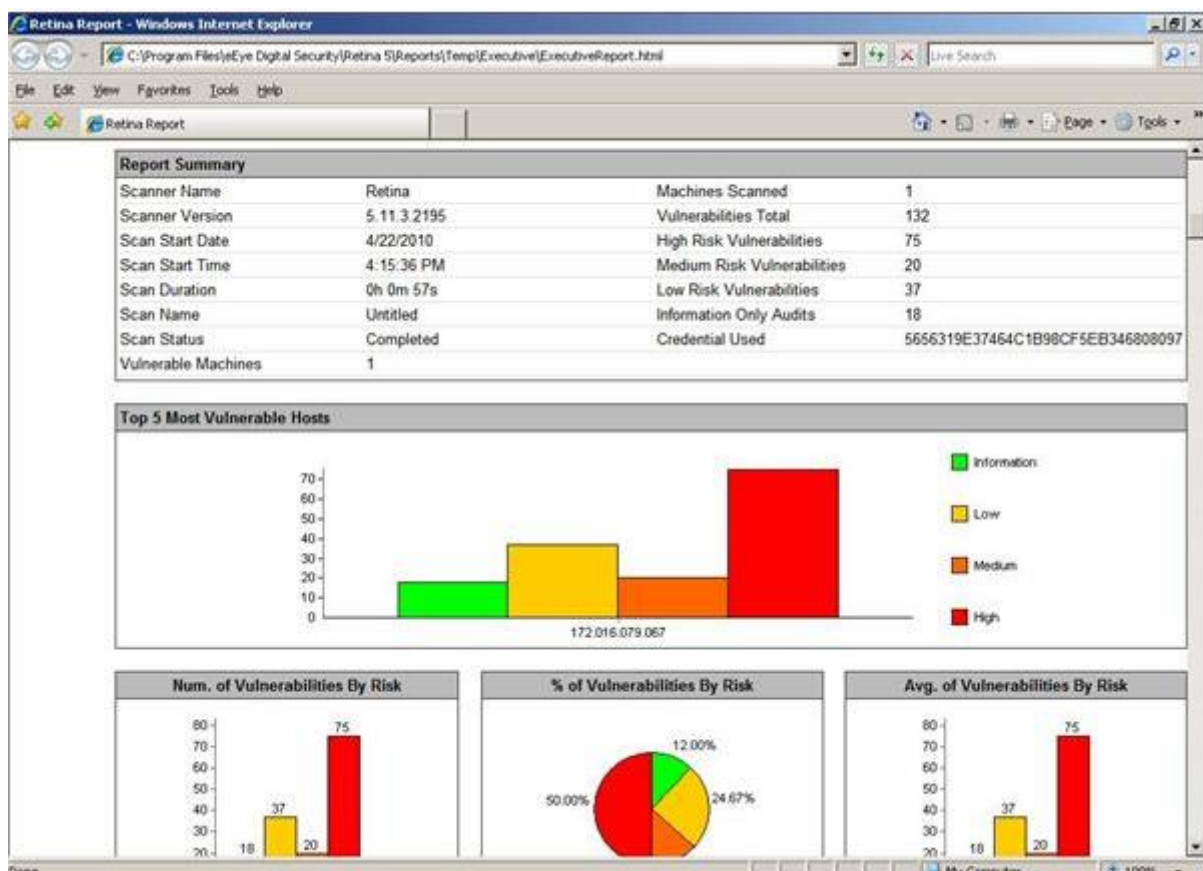


Figure 10: Vue d'ensemble de Retina

5. OpenVAS

OpenVAS est un fork du logiciel NESSUS. Comme son confrère, il est capable de scanner un SI entier et indiquer les faiblesses potentielles de chaque équipement repéré. Il fournit une liste des faiblesses et un lien pour pouvoir les corriger. OpenVAS est construit en trois parties indépendantes. La première partie permet de préciser les paramètres du scan (OS, rapidité, option...) et de visualiser le rapport. Le scan peut être lancé depuis le site fourni ou en ligne de commande. La seconde partie est OpenVAS manager qui permet de préparer le scan selon les paramètres que nous avons rentré dans la première partie et récupérera les résultats du scan. Enfin, la troisième partie est le Scanner de Vulnérabilité. Cette répartition en trois parties simplifie l'intégration de l'outil dans un projet.

OpenVAS accepte l'ajout de plugin pour permettre de lui rajouter des fonctionnalités. Ainsi, il est possible de créer des vulnérabilités contre des applications internes et ainsi les tester pour vérifier que les patchs ont bien été appliqués. Il est également possible d'ajouter des plugins existants comme Wapiti, Arachni, Nikto et Dirb. Le rapport peut être généré sous différents formats.

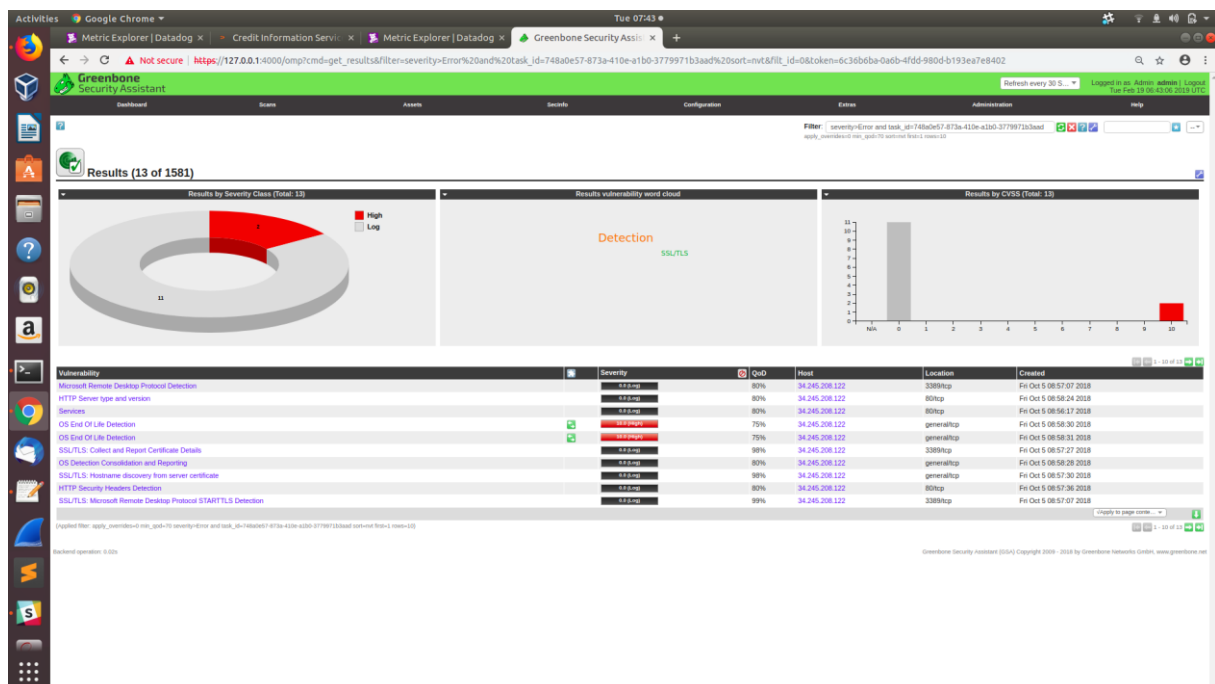


Figure 11: Vue d'ensemble d'OpenVAS

II. Critère de comparaison

Suite au diagnostic et à l'identification des besoins, la comparaison entre ces solutions se fera sur la base de critères qui découlent de notre problématique ainsi que de l'étude de l'existant.




Les critères de comparaison définis sont les suivants :



1. Multiplateformes.
2. Scan authentifié: Bien que la plupart des scanners soient "sans agents", ils fournissent souvent la possibilité d'utiliser un compte renseigné par l'utilisateur pour mener des tests authentifiés.
3. Scan automatique : permet de notifier des vulnérabilités trouvées et de Gagner du temps en analyse manuelle et de recevoir une notification dès que des vulnérabilités sont détectées , Et il permet de Garder une trace de l'analyse , et réduire les risques de faux positifs et de faux négatifs.
4. Intégration des modules supplémentaires.
5. Scan complet.

III. Tableau comparatif

Le tableau suivant est une étude comparative entre les solutions citées selon les critères présentés.

Tableau 1:Tableau comparatif des scanners de vulnérabilités

Nom	Rayon d'action	Licence	scan authentifié	Scan automatique	Environnement
Elastic Detector 	Serveur, Réseau, ordinateur, hyperviseurs (Vsphere, hyper-V, Citrix XenServer)	Payant	Non	Oui	Cloud
Nessus 	Serveur, Réseau, OS, Application WEB, appareil mobile	Payant(22 870,09 MAD per year)	Oui	Non	Unix , Windows
Nexpose 	Scan du réseau, d'OS, Serveur, BDD, application web et environnement virtuel	Payant(28733 1,42 MAD)	Oui	Oui	Windows , Linux

		Gratuit(limité à 32 adresses IP)			
	OpenVAS Serveur, Réseau, OS, Application WEB, appareil mobile	Gratuit	Oui	Oui	Multiplateforme
	Retina Scan du réseau, d'OS, Serveur, BDD, application web et environnement virtuel	Payant(17271,00MAD) Gratuit(limité à 256 adresses)	Non	Oui	Windows

IV. Présentation d'OpenVAS

OpenVAS (Open Vulnerability Assessment System) est un scanner de vulnérabilités complet, un framework, et un fork (ou une branche dérivée) de NESSUS.

Nessus étant sous licence propriétaire, OpenVAS s'est développé sous licence GNU GPL.

En 2005, les développeurs du scanner de vulnérabilités Nessus ont décidé de mettre fin aux travaux sous licences Open Source et de passer à un modèle commercial propriétaire.

En 2006, plusieurs fourchettes de Nessus ont été créées en réaction à l'arrêt de la solution Open Source. Parmi ces fourchettes, une seule a continué à faire preuve d'activité: OpenVAS.

OpenVAS est développé et mis à jour par Greenbone Networks depuis 2009. Les travaux sont fournis en tant que source ouverte à la communauté sous licence GNU General Public (GNU GPL).

Greenbone développe OpenVAS dans le cadre de sa famille de produits de gestion de vulnérabilités commerciale "Greenbone Security Manager" (GSM). OpenVAS est l'un des éléments d'une architecture plus grande. Associé à des modules Open Source supplémentaires, il constitue la solution Greenbone Vulnerability Management.

Greenbone est la société qui exploite OpenVAS et propose le scanner de vulnérabilité en version gratuite ou payante. La principale différence réside dans l'alimentation des tests de vulnérabilité de réseau (NVT) utilisés par le scanner.

La version payante du flux s'appelle Greenbone Security Feed, tandis que la version gratuite du flux s'appelle Greenbone Community Feed. Les deux flux sont mis à jour quotidiennement et incluent les menaces les plus récentes.

Ses capacités incluent des tests non authentifiés, des tests authentifiés, divers protocoles Internet et industriels de haut niveau et de bas niveau, le réglage des performances pour les analyses à grande échelle et un puissant langage de programmation interne permettant de mettre en œuvre tout type de test de vulnérabilité.

la solution OpenVAS s'appuie sur un ensemble de vulnérabilités connues (environ +55000 NVTs connues en 2017).

1. Architecture d'OpenVAS

Le schéma présente les différents composants de l'architecture d'OpenVAS ainsi que les sources d'alimentation des NVTs.

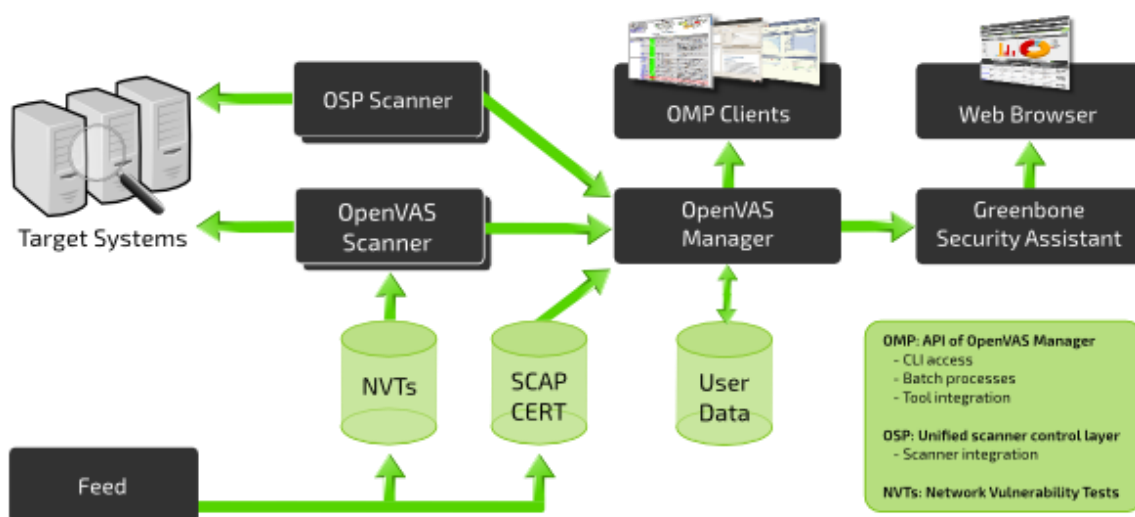


Figure 12: Architecture d'OpenVAS

- **Greenbone Security Assistant (GSA)** fournit une interface utilisateur Web pour l'administration et la gestion des analyses, rapports, rapports OpenVAS...
- **OpenVAS Manager** qui est l'artère principale de l'OpenVAS. Il reçoit diverses tâches/instructions de l'administrateur via les composants client, WEB / GUI / CLI, et utilise ces instructions pour contrôler OpenVAS Scanner, qui effectue l'évaluation de la vulnérabilité. Il contrôle également une base de données SQL où toutes les données de configuration et d'analyse des résultats sont stockées de manière centralisée. Enfin, il gère également la gestion des utilisateurs, notamment le contrôle d'accès avec des groupes et des rôles.
- **OMP Clients:** c'est une API d'OpenVAS Manager :

- ✓ CLI OpenVAS qui fournit l'interface de ligne de commande pour l'administration OpenVAS.
- ✓ Intégration d'outils.
- **OpenVAS Scanner** est le composant qui évalue la vulnérabilité par rapport aux cibles spécifiées.
- **Les cibles d'analyse** sont les points d'extrémité faisant l'objet d'une évaluation de toutes les vulnérabilités.

CHAPITRE IV

Pré-étude et implémentation



I. Pré-étude

1. Démarche de mise en place

Pour mener à bien le déroulement du projet, son implémentation doit passer par différentes étapes et qui doivent en premier lieu être approuvée et validée par mon encadrant.

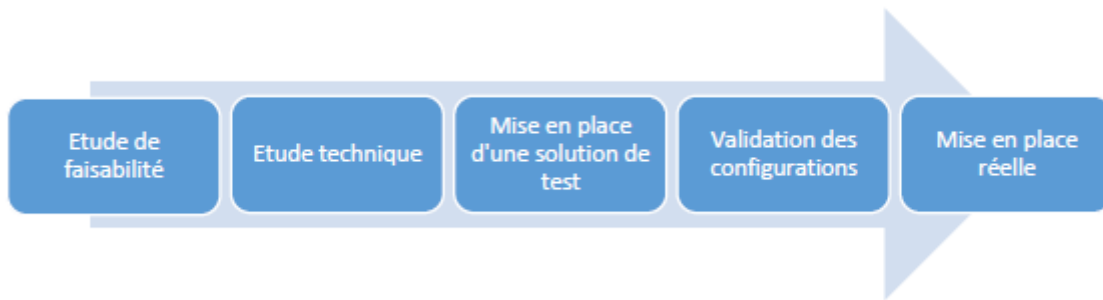


Figure 13:Démarche de mise en place

2. Etude de faisabilité

L'étude de faisabilité est une étude théorique dont l'objectif est de vérifier que le projet est conforme à la stratégie et les moyens de l'entreprise, ainsi cette étude sera abordée selon deux volets, on va tout d'abord vérifier la faisabilité économique de la mise en place de la solution ainsi que la faisabilité organisationnelle.

- **Faisabilité économique**

Avant toute implémentation, il faut s'assurer que les coûts soient acceptables, pour le cas d'une mise en place d'une solution ayant une licence à payer, il faudra mobiliser des ressources financières pour mener à bien le projet. Dans notre cas, la solution à implémenter est une solution libre, n'impliquant ainsi aucun coût supplémentaire, de même pour le système d'exploitation où on va installer la solution.

- **Faisabilité organisationnelle**

Le projet rentre dans la stratégie de CGI, puisque la solution va contribuer à une meilleure gestion du patrimoine informatique de CGI- Maroc et ainsi augmenter la productivité de cette entreprise.

3. Etude conceptuelle

Cette étude va pouvoir déterminer le choix de la solution réseau pour l'hébergement de la VM d'OpenVAS sur le DataCenter VDR.

3.1. Critères de choix

- Solution permettant de délivrer un service d'hébergement sur le datacenter de VDR.
- Environnement accessible depuis le réseau CGI.
- Environnement communiquant avec l'extérieur : il s'agit des environnements accessibles depuis le réseau CGI et depuis l'extérieur (par ex. par le client ou des utilisateurs externes à CGI)

3.2. Architecture de la solution réseau choisi

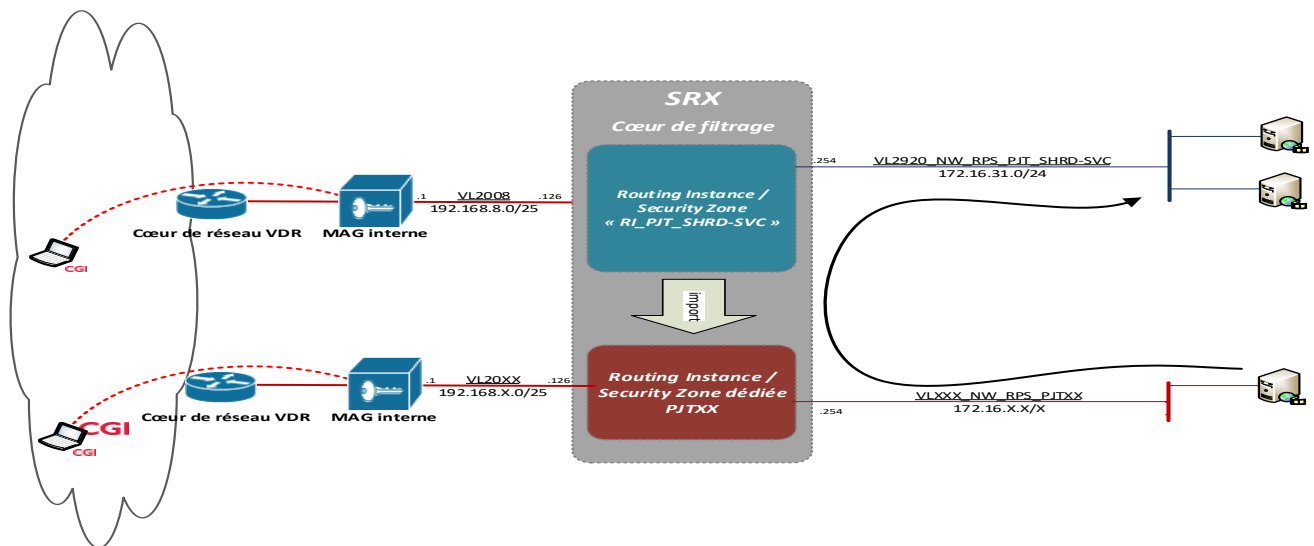


Figure 14: Conception de la solution

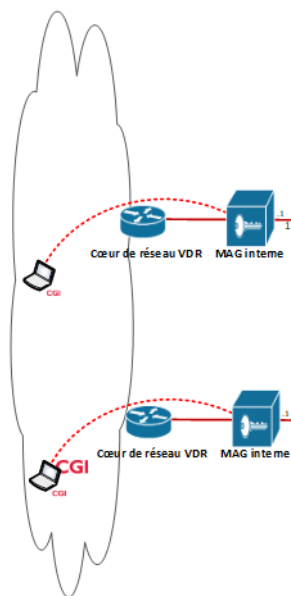
La solution comprend une solution d'hébergement, Le sous-réseaux d'hébergement sont portés par le SRX, Il est la passerelle pour les hôtes de ce réseau pour les accès distants.

Ce sous-réseau peut accéder aux ressources distantes des différents projets GTO.

Ce/ces sous-réseau(x) seront accessibles par les utilisateurs des solutions SAS au travers du client Pulse. Ils pourront également être accédés depuis les réseaux client, auquel cas les flux réseau à destination de CGI seront proscrits.

Le/les sous-réseau(x) d'hébergement sont positionnés dans la « routing instance » dédiée à l'environnement projet en question.

3.3. Les équipements réseau



- Un cluster de **CISCO Nexus 7000** positionné en point de routage entre le MAN et le réseau de VDR.
- Deux clusters de **PulseSecure MAG SM 360** Ces équipements sont les passerelles d'accès aux environnements projet sécurisés.

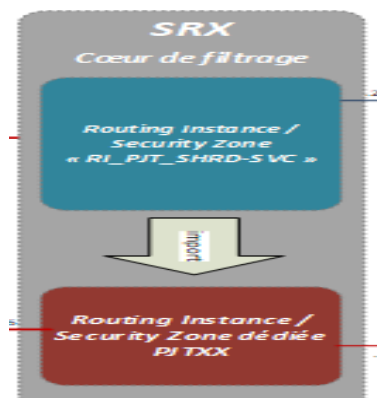
Un premier cluster est destiné à offrir un accès depuis le réseau CGI, il s'agit du MAG interne (FR-DC1LVP001/2). L'autre cluster permet les accès depuis Internet pour satisfaire, entre autre les besoins en télétravail, il s'agit du MAG externe (FR-DC1LVP003/4).

Le MAG dispose de trois interfaces :

Interface de management : Elle est uniquement dédiée aux tâches d'administration de l'équipement.

Interface externe : Cette interface est celle sur laquelle sont initiées les connexions distante depuis les postes utilisateurs.

Interface interne : le MAG communique avec le réseau CGI par le biais de cette interface. C'est celle qui est utilisée pour porter les différents réseaux d'interconnexion projet avec le SRX.



- Un cluster de Juniper SRX 1400, pare-feu centraux qui assureront le cloisonnement des environnements projet et le routage vers les ressources spécifiques.

Le SRX permet le filtrage et le routage.

Afin d'assurer le cloisonnement logique des différents environnements, chaque profil projet sur le MAG dispose de sa propre interconnexion avec le SRX.

Dans un même souci de sécurité et de cloisonnement, chaque solution dédiée à un projet donne lieu à la création d'un routeur virtuel sur le SRX appelé « routing instance » et une zone de sécurité appelée « security zone » correspondant à une zone logique de filtrage.

une interface dédiée à l'interconnexion MAG/SRX pour un environnement projet donné est créée sur le SRX et se trouve positionnée dans la routing-instance dédiée.

Ceci permet d'assurer que la ségrégation des réseaux est respectée sur l'ensemble de la chaîne de routage.

4. Prérequis techniques

Pour l'implémentation de notre solution, il nous faudra les éléments techniques suivants :

- Une machine Virtuelle CentOS :

SYSTÈME				
Système	<input type="text" value="Autres (préselectionner CentOS 7)"/>	Version	64 bits	

COMPOSANTS				
Processeur	2vCPU	RAM	4 Go	
Disque	50 Go			
Lun (si stockage SAN)	Disque de base	Partition	Taille	Volum Group
Non	sda	/dev/sda1	200Mo	N/A
	sda	/dev/sda2	48,8Go	vg_system

Figure 15:Spécifications techniques de l'environnement d'installation d'OpenVAS

II. Implémentation virtuelle

Avant de pouvoir mettre en place la solution dans l'environnement de CGI-Maroc, il m'a était important de commencer par implémenter la solution virtuellement, afin de se familiariser avec l'environnement et l'outil, de prendre connaissance des enjeux de la solution et aussi de valider les configurations au préalable.

Ainsi, j'ai réalisé une simulation de l'architecture de CGI- Maroc sur GNS 3, et j'ai créé une machine virtuelle sur Vmware qui a comme environnement d'exploitation la version 7 de CentOS.

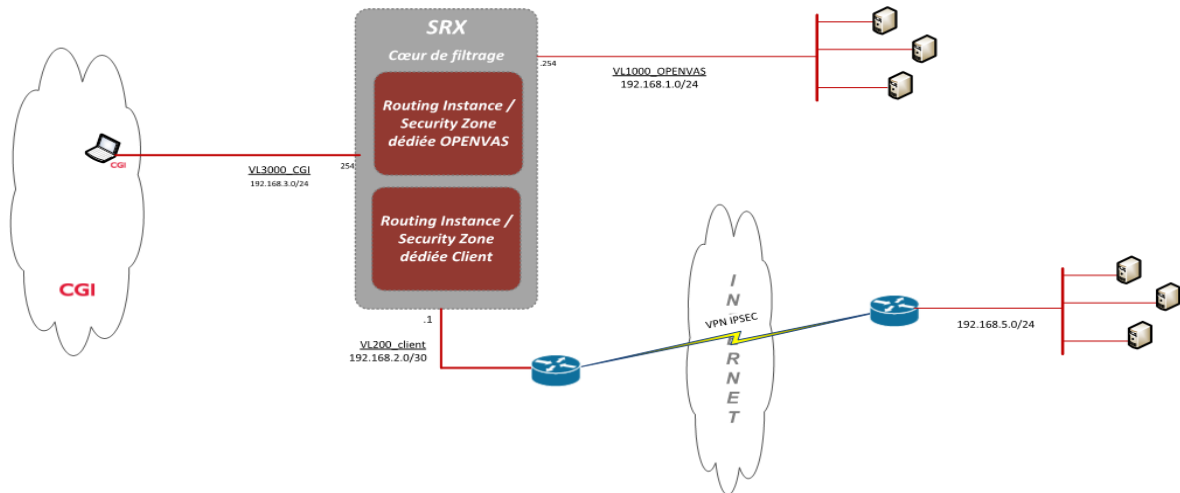


Figure 16: architecture de l'environnement de Test

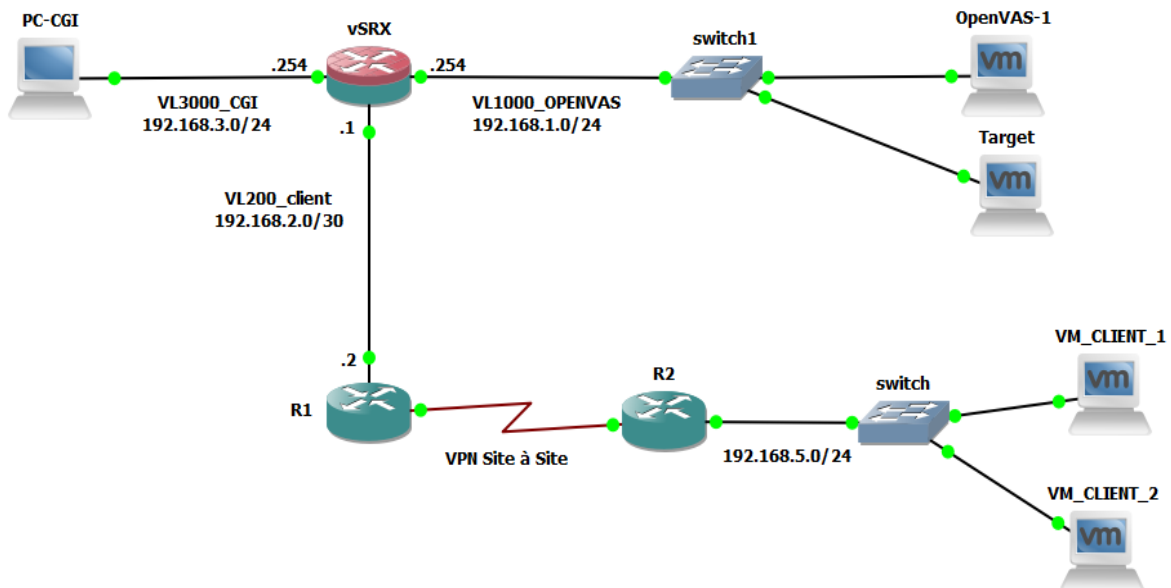


Figure 17: Architecture de simulation sous GNS3

1. Configuration de CentOS 7

- OpenVAS (GVM) se plaindra si on laisse SELinux activé

SELinux :Security-Enhanced Linux ,est un Linux security module (LSM), qui permet de définir une politique de contrôle d'accès obligatoire aux éléments d'un système issu de Linux.

Alors désactiver le en utilisant la commande suivante :

```
[root@localhost ~]# sed -i ' s/=applique /=disabled/' /etc/selinux/config
[root@localhost ~]# _
```

- Ouvrir le port nécessaire pour l'interface Web OpenVAS :

```
[root@localhost ~]# firewall-cmd --zone=public --add-port=9392/tcp --permanent
Warning: ALREADY_ENABLED: 9392:tcp
success
[root@localhost ~]# firewall-cmd --reload
success
```

- mettre à jour et redémarrer CentOS :

```
[root@localhost ~]# yum -y update && reboot
```

- Installation des paquets:

- ✓ wget
- ✓ net-tools
- ✓ bzip2

- Dépôts de paquets :

Pour CentOS, OpenVAS est fourni par deux dépôts de paquets tiers concurrents, Atomiccorp et EPEL.

- ✓ **Atomicorp**: Le référentiel RPM de Atomic aka ART (Atomic Rocket Turtle) est une archive open source non prise en charge de packages logiciels destinés à la communauté Centos et Redhat. Cela inclut PHP, l'hébergement de projets spécifiques, les archives yum rpm et les forums de la communauté. Fournissant un seul paquet **Openvas** qui contient tout ce qu'il faut.

Installation et configuration du référentiel à partir d'Atomic Corp :


```
[root@localhost ~]# wget -q -O - https://updates.atomicorp.com/installers/atomic | sh

Atomic Free Unsupported Archive installer, version 4.0.1

BY INSTALLING THIS SOFTWARE AND BY USING ANY AND ALL SOFTWARE
PROVIDED BY ATOMICORP LIMITED YOU ACKNOWLEDGE AND AGREE:

THIS SOFTWARE AND ALL SOFTWARE PROVIDED IN THIS REPOSITORY IS
PROVIDED BY ATOMICORP LIMITED AS IS, IS UNSUPPORTED AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ATOMICORP LIMITED, THE
COPYRIGHT OWNER OR ANY CONTRIBUTOR TO ANY AND ALL SOFTWARE PROVIDED
BY OR PUBLISHED IN THIS REPOSITORY BE LIABLE FOR ANY DIRECT,
INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.

For supported software packages please contact us at:

    sales@atomicorp.com

Do you agree to these terms? (yes/no) [Default: yes] yes_
```

- ✓ **EPEL(Extra Packages for Enterprise Linux):**est un groupe d'intérêt spécial de Fedora qui crée, met à jour et gère un ensemble de packages de haute qualité pour Enterprise Linux, notamment Red Hat Enterprise Linux (RHEL), CentOS et Scientific Linux (SL), Oracle Linux (OL). Les paquets EPEL fournissent une série de paquets pour OpenVAS, étant donné qu'il s'agit là d'un véritable framework avec une série de composants.

- ☐ openvas-scanner
- ☐ openvas-libraries
- ☐ openvas-manager
- ☐ openvas-cli
- ☐ openvas-gsa

Installation et configuration du référentiel à partir d'EPEL :

yum install <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

- ✓ Choix du dépôt de paquets:

Après vérification de l'installation d'OpenVAS, j'ai constaté que Le dépôt de paquets EPEL fournit le paquet OpenVAS dans sa version 8 et de ce fait l'installation des flux (NVTs) est devenu impossible, alors que Atomicorp fournit OpenVAS dans sa dernière version 10.

D'où la nécessité du choix du dépôt Atomicorp.

2. Installation d'OpenVAS (GVM 10)

Remarque: Greenbone désapprouve OpenVAS versions 9 et 10, il est maintenant connu sous le nom de Greenbone Vulnerability Manager (GVM). De même, les nouveaux fichiers rpms sont appelés «greenbone-vulnerability-manage» et «gvm-libs», qui remplacent les fichiers «openvas» et «openvas-libraries».

- ✓ Installation d' OpenVAS (GVM) et les dépendances associées se fait par commande :

```
[root@localhost ~]# yum -y install greenbone-vulnerability-manager
```

- ✓ **Le serveur Redis :** (Remote Dictionary Server) est un système de gestion de bases de données clé/valeur. Il fait partie de la famille des bases NoSQL et vise la performance. Et notre système OpenVAS a visiblement besoin d'un serveur Redis pour fonctionner correctement. nstallation du serveur redis se fait par la commande :

```
# yum install redis
```

- ✓ Éditer /etc/redis.conf et spécifier un emplacement approprié pour le fichier socket, aux alentours de la ligne 100.

```
# Specify the path for the Unix socket that will be used to listen for
# incoming connections. There is no default, so Redis will not listen
# on a unix socket when not specified.
#
unixsocket /tmp/redis.sock
unixsocketperm 700
```

- ✓ Nous devons maintenant activer le service Redis pour qu'il démarre après les prochains redémarrages. Nous allons également démarrer /start le service.

```
[root@localhost ~]# systemctl enable redis && systemctl restart redis
Created symlink from /etc/systemd/system/multi-user.target.wants/redis.service to /usr/lib/systemd/system/redis.service.
```

- ✓ Exécuter 'openvas-setup' et acceptez rsync par défaut.

```
[root@localhost ~]# openvas-setup
```

```
Openvas Setup, Version: 4.0.1
```

```
Redirecting to /bin/systemctl restart redis.service
```

```
Step 1: Update NVT, CERT, and SCAP data
```

```
Please note this step could take some time.
```

```
Once completed, this will be updated automatically every 24 hours
```

```
Select download method
```

```
* wget (NVT download only)
* curl (NVT download only)
* rsync
```

```
Note: If rsync requires a proxy, you should define that before this step.
```

```
Downloader [Default: rsync] █
```

- ✓ Une fois que openvas-setup est terminé et que certaines clés sont générées, on reçoit les invites suivantes. «Autoriser les connexions depuis n'importe quelle adresse IP?» en appuyant simplement sur Entrée en supposant qu'on souhaite accéder à l'interface Web à partir de n'importe quelle adresse IP. On peut changer le nom d'utilisateur et taper (deux fois) le mot de passe que vous souhaitez utiliser pour accéder à l'interface Web.

```
Step 2: Choose the GSAD admin users password.
```

```
The admin user is used to configure accounts,
Update NVT's manually, and manage roles.
```

```
Enter administrator username [Default: admin] :
```

```
Enter Administrator Password:
```

```
Verify Administrator Password:
```

```
Setup complete, you can now access GSAD at:
```

```
https://<IP>:9392
```

```
Redirecting to /bin/systemctl restart gsad.service
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/openvas-scanner.service to /usr/lib/systemd/system/openvas-scanner.service.
```

```
Created symlink from /etc/systemd/system/openvas-manager.service to /usr/lib/systemd/system/gvmd.service.
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/gvmd.service to /usr/lib/systemd/system/gvmd.service.
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/gsad.service to /usr/lib/systemd/system/gsad.service.
```

```
Vous avez du nouveau courrier dans /var/spool/mail/root
```

Le seul problème d'OpenVAS (GVM) c'est qu'il ne fonctionne pas sur 9392 en tant qu'états du package. Pour régler ce problème, exécuter les commandes suivantes :

```
#echo 'OPTIONS="--listen=0.0.0.0 --port=9392"' > /etc/sysconfig/gsad
```

```
#systemctl start gsad
```

- ✓ Accéder à l'interface Web OpenVAS (GVM) :

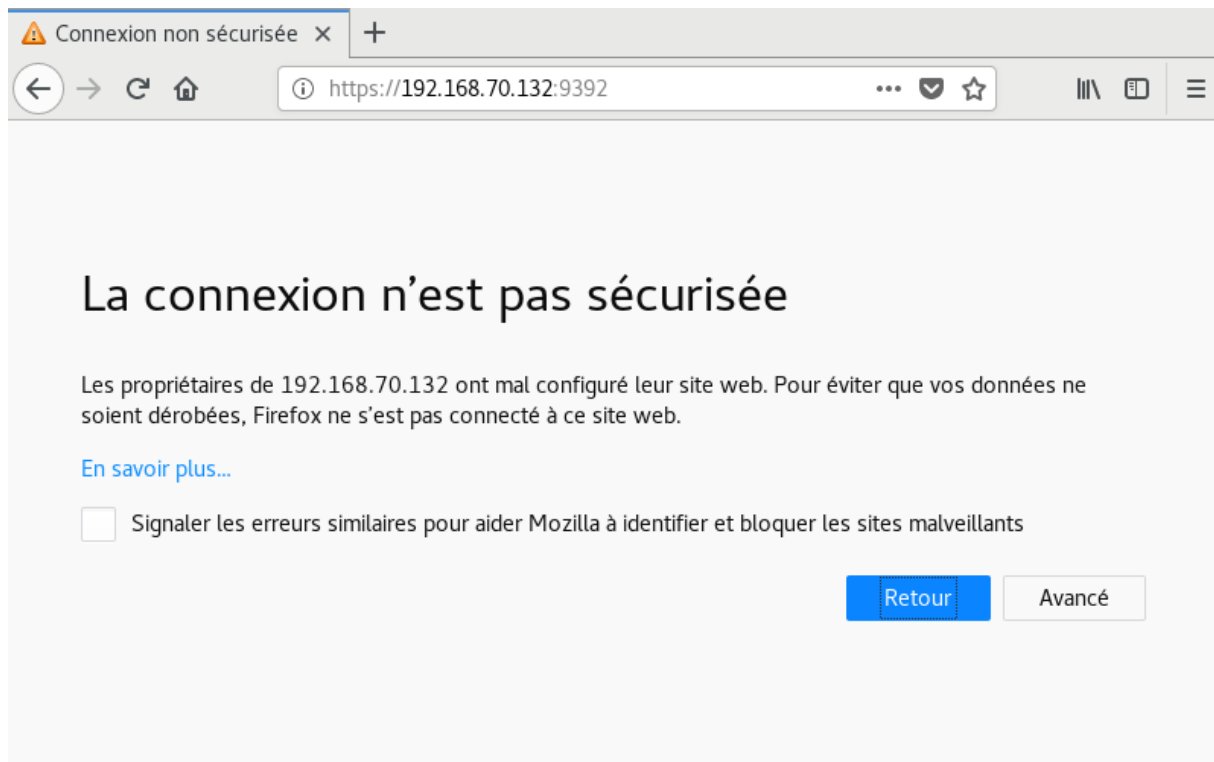
```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.70.132 netmask 255.255.255.0 broadcast 192.168.70.255
    inet6 fe80::e308:1bbf:4304:cabe prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:6b:9b:bf txqueuelen 1000 (Ethernet)
    RX packets 970383 bytes 1453449106 (1.3 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 402293 bytes 24194902 (23.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 85 bytes 6828 (6.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 85 bytes 6828 (6.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

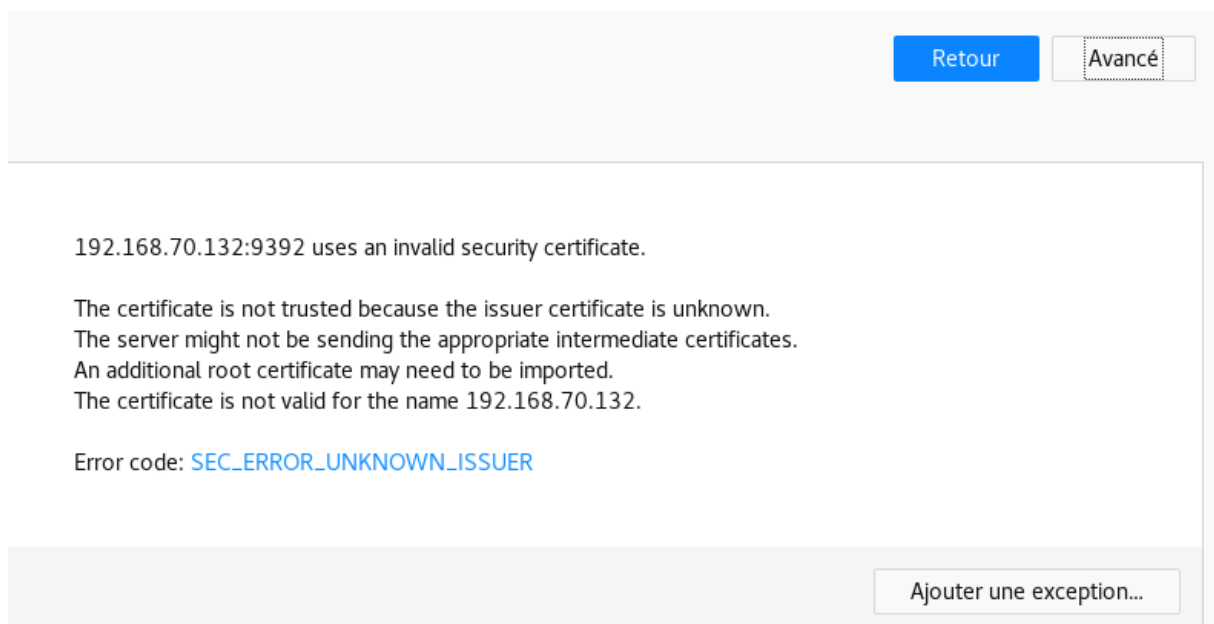
virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:02:a0:a0 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

On accède à l'interface Web à partir de n'importe quel navigateur en nous rendant à l'adresse [https:// 192.168.70.132 : 9392](https://192.168.70.132:9392)

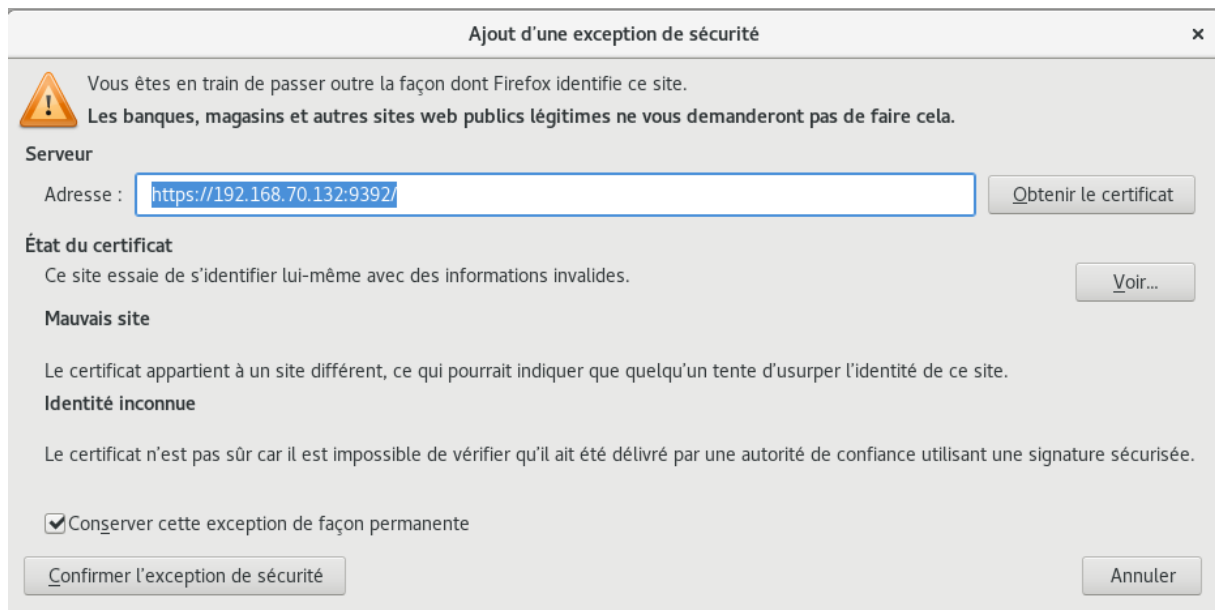
On reçoit une invite de sécurité concernant le certificat puisqu'il est auto-signé, mais on doit pouvoir nous connecter ensuite en cliquant sur avancé.



Je clique sur Ajouter une exception.



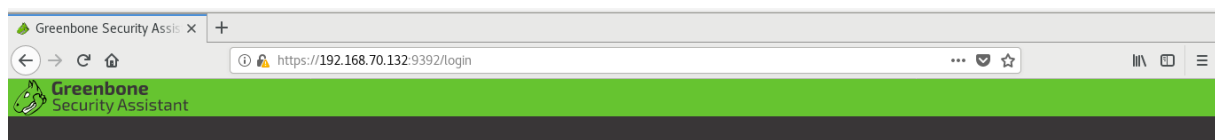
Et enfin, je confirme l'exception de sécurité de façon permanente.



La fenêtre de connexion du Greenbone Security Assistant s'affiche. Je fournis l'identifiant et le mot de passe définis lors de l'installation.



Greenbone Security Assistant (GSA) Copyright (C) 2009-2019 by Greenbone Networks GmbH, www.greenbone.net



Greenbone Security Assistant (GSA) Copyright (C) 2009-2019 by Greenbone Networks GmbH, www.greenbone.net

Annexe



I. Configuration vSRX

```
//Change Hostname and password of VSRX
set system root-authentication plain-text-password
Set system host-name vSRX
commit and-quit

//Configuration interco vSRX-OPENVAS
set interfaces ge-0/0/1.0 unit 0 description VL1000_OPENVAS
set interfaces ge-0/0/1.0 unit 0 vlan-id 1000
set interfaces ge-0/0/1.0 unit 0 family inet address 192.168.1.254/24

//Configuration interco PC-vSRX :
set interfaces ge-0/0/0.0 unit 0 description VL3000_CGI
set interfaces ge-0/0/0.0 unit 0 vlan-id 3000
set interfaces ge-0/0/0.0 unit 0 family inet address 192.168.3.254/24

//Configuration interco SRX-R1 :
set interfaces ge-0/0/2.0 unit 0 description VL200_client
set interfaces ge-0/0/2.0 unit 0 vlan-id 200
set interfaces ge-0/0/2.0 unit 0 family inet address 192.168.2.1/30

//Configuration Security Zone
Zone PC-SRX : zone Trust
ge0/0/0 appartient par défaut à Zone untrust .
edit security zones security-zone untrust
delete interfaces ge-0/0/0.0

Commit

set security zones security-zone INTERN_CGI
set security zones security-zone INTERN_CGI interfaces ge-0/0/0.0
set security zones security-zone INTERN_CGI host-inbound-traffic system-services ping
set security zones security-zone INTERN_CGI host-inbound-traffic system-services http
set security zones security-zone INTERN_CGI host-inbound-traffic system-services https
set security zones security-zone INTERN_CGI host-inbound-traffic system-services ssh

//Configuration routing instance RI_OPENVAS
set routing-instances RI_OPENVAS instance-type virtual-router
set routing-instances RI_OPENVAS interface ge-0/0/0.0
set routing-instances RI_OPENVAS interface ge-0/0/1.0
set routing-instances RI_OPENVAS routing-options static route 192.168.3.0/24 next-hop
192.168.3.254

Zone OpenVAS SRX : SZ_SRX_OPENVAS
set security zones security-zone SZ_SRX_OPENVAS
set security zones security-zone SZ_SRX_OPENVAS interfaces ge-0/0/1.0
set security zones security-zone SZ_SRX_OPENVAS host-inbound-traffic system-services any-
service
```


//Configuration routing instance RI_Client

```
set routing-instances RI_Client instance-type virtual-router
```

```
set routing-instances RI_Client interface ge-0/0/0.0
```

```
set routing-instances RI_Client interface ge-0/0/2.0
```

```
set routing-instances RI_Client routing-options static route 192.168.5.0/24 next-hop 192.168.2.1
```

To verify that routing instance has been created do "show route instance"

Zone Client SRX : SZ_SRX_Client

```
set security zones security-zone SZ_SRX_Client
```

```
set security zones security-zone SZ_SRX_Client interfaces ge-0/0/2.0
```

```
set security zones security-zone SZ_SRX_Client host-inbound-traffic system-services ping
```

```
set security zones security-zone SZ_SRX_Client host-inbound-traffic system-services ssh
```

```
set security zones security-zone SZ_SRX_Client host-inbound-traffic system-services telnet
```

Note : To verify "show security policies" first

//Security Policies

```
set security address-book global address NW_OPENVAS 192.168.1.0/24
```

```
set security address-book global address R1 192.168.2.1/30
```

```
set security address-book global address NW_CGI 192.168.3.0/24
```

Address books are simply a way of naming a host within a zone (or globally if required). Address books are used as a point of reference within any security policy configuration.

Now to create the policy:

//FOR OPENVAS

```
set security policies from-zone SZ_SRX_OPENVAS to-zone SZ_SRX_OPENVAS policy FW001
```

```
match source-address W_OPENVAS
```

```
set security policies from-zone SZ_SRX_OPENVAS to-zone SZ_SRX_OPENVAS policy FW001
```

```
match destination-address
```

```
set security policies from-zone SZ_SRX_OPENVAS to-zone SZ_SRX_OPENVAS policy FW001
```

```
match application any
```

```
set security policies from-zone SZ_SRX_OPENVAS to-zone SZ_SRX_OPENVAS policy FW001
```

```
then permit
```

//FOR INTERN CGI

```
set security policies from-zone INTERN_CGI to-zone INTERN_CGI policy FW002 match source-address NW_CGI
```

```
set security policies from-zone INTERN_CGI to-zone INTERN_CGI policy FW002 match any
```

```
set security policies from-zone INTERN_CGI to-zone INTERN_CGI policy FW002 match application any
```

```
set security policies from-zone INTERN_CGI to-zone INTERN_CGI policy FW002 then permit
```

//FOR CLIENT

set security policies from-zone SZ_SRX_Client to-zone SZ_SRX_Client policy FW003 match
source-address NW_CGI

set security policies from-zone SZ_SRX_Client to-zone SZ_SRX_Client policy FW003 match any

set security policies from-zone SZ_SRX_Client to-zone SZ_SRX_Client policy FW003 match
application any

set security policies from-zone SZ_SRX_Client to-zone SZ_SRX_Client policy FW003 then permit
Policy between OpenVAS and Client :

set security policies from-zone SZ_SRX_OPENVAS to-zone SZ_SRX_Client policy FW004 match
source-address any

set security policies from-zone SZ_SRX_OPENVAS to-zone SZ_SRX_Client policy FW004 match
destination-address any

set security policies from-zone SZ_SRX_OPENVAS to-zone SZ_SRX_Client policy FW004 match
application any

set security policies from-zone SZ_SRX_OPENVAS to-zone SZ_SRX_Client policy FW004 then
permit

Import Route :

- Apply a policy to routes being imported into each of the virtual routing instances:

set routing-instances RI_Client routing-options instance-import import-from-RI_OPENVAS

set routing-instances RI_OPENVAS routing-options instance-import import-from-RI_Client

- Create a policy that imports routes from routing instances RI_Client to RI_OPENVAS and
another policy that imports routes from routing instances RI_OPENVAS to RI_Client :

set policy-options policy-statement import-from-RI_Client term 1 from instance RI_Client

set policy-options policy-statement import-from-RI_Client term 1 then accept

set policy-options policy-statement import-from- RI_OPENVAS term 1 from instance
RI_OPENVAS

set policy-options policy-statement import-from- RI_OPENVAS term 1 then accept

Note : to verify "show configuration" it will show route options

Use the show route forwarding-table command to view the forwarding table information for each
routing instance: show route forwarding-table