MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLAB DE BLIDA 1

FACULTE DES SCIENCES

MEMOIRE DE MASTER II Option : Sécurité des Système d'Information

THEME :

Conception et réalisation d'un gestionnaire de vulnérabilités système

Réalisé par

BENTHAMEUR Mourad

&

BENKOULAL Ramzi Wassim

Année Universitaire: 2019-2020

Devant le Jury compose de :

Mme. N.BOUSTIA	**Présidente**
Mme ARKAM M.	**Promotrice**
Mme S.AROUSSI	**Examinatrice**

Remerciement

Avant tout, nous remercions Allah de nous avoir accordé la force et le courage nécessaire afin d'accomplir ce travail, et Continuer à progresser durant ces longues années d'études.

Nous tenons à remercier vivement Madame ARKAM Meriem, pour son apport scientifique, qui a été mis à notre disponibilité, par ses conseils fructifiés et directives lors de l'élaboration de ce mémoire, et aussi Messieurs OUCHENE Abdelaziz et BENKOULAL Abdelghani, qui nous ont été d'une aide indispensable pour la concrétisation de notre PFE.

Nous remercions également toute l'équipe pédagogique du département d'informatique de l'université de Blida 1 et aussi tous nos enseignants du cycle primaire au cycle universitaire.

Nous remercions nos camarades qui ont rendu meilleur nos années à l'université.

Nos remerciements vont aussi aux membres du jury qui ont pris de leur temps pour juger ce modeste travail, qu'ils trouvent ici l'expression de notre gratitude et tout notre respect.

Il est naturel que nos pensées les plus fortes aillent vers nos parents. Qu'ils sachent que l'amour qu'ils nous donnent continue à nous animer et nous permet d'envisager l'avenir comme un défi.

Nous tenons à exprimer nos sincères remerciements à frères et sœurs, ainsi qu'aux familles Benthameur & Benkoulal pour leurs conseils, et leur soutien, à la fois moral qu'économique.

Enfin, nous souhaitons adresser nos remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Merci à toutes et à tous.

Résumé :

Les gestionnaires de vulnérabilités système sont devenus des mécanismes de sécurité indispensables chez les grandes entreprises possédant une infrastructure informatisée, comme l'entreprise "ELIT" qui est spécialisée dans les technologies de l'information et de la communication. Mais le problème qui se pose devant ces sociétés, c'est que la quasi-totalité des gestionnaires de vulnérabilités système dans le marché sont chères, et on n'est jamais sûre de ce qui se passe en arrière-plan.

Pour ces raisons-là "ELIT" a décidé d'avoir son propre gestionnaire de vulnérabilités.

Dans ce travail, nous avons réalisé un motif qui répond aux besoins de l'entreprise, respecte sa politique et qui est compatible avec son architecture.

ملخص:

أصبحت برامج تسيير ثغرات الانظمة أحد آليات الأمان الأساسية في الشركات الكبيرة التي لديها بنية تحتية محوسبة، مثل المتخصصة في تقنيات المعلومات والاتصالات "الجزائر لتكنولوجيا المعلومات "، لكن المشكلة التي تظهر "ELIT"شركة أمام هذه الشركات هي أن برامج تسيير ثغرات الانظمة في السوق باهظة الثمن، بالإضافة إلى أن مورديهم لا يسمحون للشركة أن يكون لديهم تحكم 100٪ من هذا البرنامج ولا لرؤية وظائفهم الحقيقية المختبئة في الخلفية.

أن يكون لها مسير الثغرات الخاص بها "ELIT" لهذه الأسباب قررت.

في هذا العمل، حققنا نمطًا يلبي احتياجات الشركة ويحترم سياستها ويتوافق مع هندستها

Summary:

vulnerability managers have become one of the essential security mechanisms in big companies that have a computerized infrastructure, such as the "ELIT" company which is specialized in information and communication technologies, "El Djazaïr Information Technology », But the problem which arises in front of these companies is that all the vulnerability managers in the market are expensive, in addition to that their vondors do not allow the company nor to have a handling 100% of this software nor to see their real functionality hiding in the background, for these reasons "ELIT" has decided to have its own vulnerability manager.

In this work, we have achieved a pattern that meets the needs of the company, respects its policy and compatible with its architecture.

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GENERALE

Commenté [H1]: C'est ici qu'on doit trouver
1)Le context
2)Le sujet
3)La problématique
4)Les objectifs
5)L'organisation du mémoire
Et non pas dans le chapitre 1 !

L'humanité ne cesse de se développer. En notre ère, des milliers de logiciels sont développé chaque jour partout dans le monde, et dans tous les domaines, ce qui engendre des millions de failles.

Ces failles ne sont pas sans conséquences, au contraire, des millions de dollars sont perdu et des millions d'informations personnels sont divulgué chaque année à cause des cyber-attaques. La cyber-sécurité est donc le défi de toute entreprise utilisant l'outils informatique, et « EL-djazair Information Technology » n'en fait pas l'exception.

La détection de ces vulnérabilités permettra aux entreprises d'éviter des dégâts d'ampleur phénoménale, c'est donc une course contre les malfaiteurs, le premier à détecter les vulnérabilités est le gagnant, ces vulnérabilités peuvent être dues à des cause humaine ou naturelle, voir accidentelle ou bien intensuinnel.

NIST "National Institute of Standards and Technology" a créé NVD "National vulnerabilty database" qui contient les vulnérabilités trouvés dans tous les systèmes d'exploitation, ce qui a facilité la détection des vulnérabilités, même les plus récentes.

Le problème qui se pose est que la détection manuelle de ces vulnérabilités est absolument impossible, car chaque système peut contenir des milliers et des milliers de logiciels et paquets qui doivent être scannés.

Ce projet de fin d'étude découle de la conviction de l'entreprise « ELIT » que la sécurité de l'information doit être un des plus grands axes de sa politique.

« ELIT » a une architecture réseau énorme, avec des milliers de serveurs linux interconnectées, comme on a dit précédemment, les vulnérabilités sont impossibles à détecter manuellement, d'où la nécessité d'un gestionnaire de vulnérabilités afin que ces dernières soient corrigées avant qu'un malfaiteur en profite.

L'entreprise utilise déjà un gestionnaire de vulnérabilités très robuste, mais aussi excessivement cher, et en plus de ça ELIT ne peut se permettre une confiance totale à l'entreprise l'ayant développé, puisque cette dernière pourrait être entrain de dérober des données de chez ELIT en arrière-plan.

Pour mettre fin à ces doutes, l'entreprise nous a confier le développement d'un premier gestionnaire de vulnérabilités en Algérie, ce dernier devra être compatible avec plusieurs architectures et politiques y compris les leurs.

Dans ce travail nous allons présenter, étape par étape, le chemin que nous avons suivi pour atteindre nos objectifs, en passant par l'état de l'art, la conception et la réalisation, et finalement le test de la solution implémentée.

CHAPITRE I : CONCEPTS GENERAUX DE GESTIONNAIRE DE VULNERABILITES

1.1 Introduction

Les vulnérabilités sont des conséquences des faiblesses dans la conception, la mise en œuvre ou l'utilisation d'un composant matériel ou logiciel du système, elles représentent une porte ouverte à les malfaiteurs pour endommager le système, pour cela des grandes entreprises utilisent des gestionnaires de vulnérabilité pour assurer plus de sécurité, le rôle de ces gestionnaires est la détection automatique des vulnérabilités, la gestion des résultats et la proposition des solutions.

Dans ce chapitre nous essayerons de traiter ces différentes notions et concepts sur les vulnérabilités et leur gestion.

Vous trouverez dans ce chapitre la problématique à traiter, les généralités et les définitions sur la sécurité informatique, les vulnérabilités et leur gestion, avant de passer en revue quelques gestionnaires existants dans la littérature.

1.2 Sécurité informatique

La sécurité des systèmes informatiques est un ensemble de mécanismes permettant de garder tout système à l'abri des attaques. Nous parlerons dans cette partie sur les aspects généraux de la sécurité informatique

1.2.1 Définition sécurité informatique

Plusieurs définitions de la sécurité informatique existent dans la littérature, dans ce qui suit deux définitions des plus usuelles.

Définition 1 : La cybersécurité (sécurité informatique) consiste à protéger les ordinateurs, les serveurs, les appareils mobiles, les systèmes électroniques, les réseaux et les données contre les attaques malveillantes. On l'appelle également sécurité informatique ou sécurité des systèmes informatiques et électroniques. Le terme est large et englobe chaque élément, de la sécurité de l'ordinateur à la reprise de l'activité après sinistre et la formation des utilisateurs. (1)

Définition 2 : La cyber sécurité (sécurité informatique) désigne le rôle de l'ensemble des lois, politiques, outils, dispositifs, concepts et mécanismes de sécurité, méthodes de gestion des risques, actions, formations, bonnes pratiques et technologies qui peuvent être utilisés pour protéger les personnes et les actifs informatiques matériels et immatériels (connectés directement ou indirectement à un réseau) des états et des organisations (avec un objectif de disponibilité, intégrité & authenticité, confidentialité, preuve & non-répudiation) (2)

1.2.2 Les objectifs de la sécurité informatique

La sécurité informatique a plusieurs objectifs, chaque objectif est lié aux types de menaces, ainsi qu'aux types de ressources … etc. qu'on peut résumer dans les trois principaux objectifs suivants :

- Empêcher la divulgation non-autorisée de données
- Empêcher la modification non-autorisée de données
- Empêcher l'utilisation non-autorisée de ressources réseau ou informatiques de façon générale

1.2.3 Les champs d'application de la sécurité informatique

Ces objectifs s'appliquent dans différents domaines ou champs d'applications, chacun faisant appel à des techniques différentes pour atteindre le ou les mêmes objectifs ; ces champs sont:

- La sécurité physique.
- La sécurité personnelle.
- La sécurité procédurale (audit de sécurité., procédures informatiques...).
- La sécurité des émissions physiques (écrans, câbles d'alimentation, courbes de consommation de courant...).
- La sécurité des systèmes d'exploitation.
- La sécurité des communications.

1.2.4 Les types d'attaques

Les attaques peuvent à première vue être classées en 2 grandes catégories :

- Les attaques passives : consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible.

- Les attaques actives : consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

1.2.5 Services principaux de la sécurité informatique

Pour remédier aux failles et pour contrer les attaques, la sécurité informatique se base sur un certain nombre de services qui permettent de mettre en place une réponse appropriée à chaque menace. A ce niveau, aucune technique n'est encore envisagée ; il ne s'agit que d'un niveau d'abstraction visant à obtenir une granularité minimale pour déployer une politique de sécurité de façon optimale. Décrivons les principaux services de sécurité :

- **Confidentialité :**
  Ce service permet de contrôler l'accès aux données, en d'autres termes il répond à la question « qui peut voir quoi et quand ? »
- **Intégrité**
  Ce service assure l'intégrité des ressources, c'est-à-dire qu'aucune ressources ne peut être perdu.
- **Disponibilité**
  Ce service assure la disponibilité des ressources à chaque fois que l'on a besoin.
- **Non-répudiation**
  Ce service fournit des preuves d'émission et/ou de réception. C'est-à-dire qu'aucune partie ne peut nier le fait d'avoir envoyé ou reçu des données.

Notez que le chiffrement, les signatures digitales et autres techniques correspondent au niveau d'abstraction inférieur, décrit comme l'ensemble des **mécanismes de sécurité** permettant de réaliser les services décrits ci-dessus. Plusieurs mécanismes peuvent par exemple réaliser le service de confidentialité (schémas d'authentification, chiffrement ...). Néanmoins, ces mécanismes de sécurité ne correspondent pas encore aux solutions finales qui seront réellement implémentées. Il faudra pour cela effectuer un dernier raffinement, consistant à choisir les algorithmes symétriques, les algorithmes asymétriques, la tailles des clés, les droits d'accès etc... (3)

1.3 Les vulnérabilités :

Nous parlerons dans cette partie en général sur les vulnérabilités

1.3.1 Définition d’une vulnérabilité :

Dans cette partie nous répondrons à la question suivante :

Qu’Est-ce qu’une vulnérabilité ?

Définition 1 : C’est une faiblesse dans bien ou dans un groupe de biens qui peut être exploitée par une ou plusieurs cyber menaces, où un bien est tout ce qui a une valeur pour l'organisation, ses opérations commerciales et leur continuité, y compris les ressources d'information qui soutiennent la mission de l’organisation. [4]

Définition 2 : C’est une faiblesse au niveau d’un système d’ĭnformation, au niveau des procédures de sécurité du système, au niveau des contrôles internes ou au niveau de la mise en œuvre qui pourrait être exploitée ou déclenchée par une source de menace. [5]

1.3.2 Les causes des vulnérabilités

Les causes des vulnérabilités sont nombreuses, notamment :

La complexité : Les systèmes complexes augmentent la probabilité d'une faille, d'une mauvaise configuration ou d'un accès non intentionnel.

Familiarité : Le code, les logiciels, les systèmes d'exploitation et le matériel communs augmentent la probabilité qu'un attaquant puisse trouver ou avoir des informations sur des vulnérabilités connues.

Connectivité : Plus un appareil est connecté, plus la probabilité d'une vulnérabilité est élevée.

Mauvaise gestion des mots de passe : Les mots de passe faibles peuvent être cassés par la force brute et la réutilisation des mots de passe peut faire qu'une violation de données devient plusieurs.

Défaillances du système d'exploitation : Comme tout logiciel, les systèmes d'exploitation peuvent avoir des failles. Les systèmes d'exploitation qui ne sont pas sécurisés par défaut et qui donnent à Tous les utilisateurs un accès complet peuvent permettre à des virus et à des logiciels malveillants d'exécuter des commandes.

Utilisation d'Internet : L'Internet est rempli de logiciels espions et de logiciels publicitaires qui peuvent être installés automatiquement sur les ordinateurs.

Bugs de logiciels : Les programmeurs peuvent accidentellement ou délibérément laisser un bogue exploitable dans un logiciel.

Entrée utilisateur non vérifiée : Si votre site Web ou votre logiciel suppose que toutes les entrées sont sûres, il peut exécuter des commandes SQL non intentionnelles.

Les gens : La plus grande vulnérabilité dans toute organisation est l'humain derrière le système. L'ingénierie sociale est la plus grande menace pour la majorité des organisations.

1.3.3 Vulnérabilités et exploits

Une vulnérabilité -avec au moins un vecteur d'attaque connu et fonctionnel- est classée comme une vulnérabilité exploitable. Ainsi, alors que vulnérable signifie qu'il y a théoriquement un moyen d'exploiter quelque chose (c'est-à-dire qu'une vulnérabilité existe), exploitable signifie qu'il y a un chemin précis pour le faire dans la réalité. Naturellement, les attaquants veulent trouver des faiblesses qui sont réellement exploitables. En tant que défenseur, être vulnérable n'est pas très bien, mais vous devriez être particulièrement inquiet d'être exploitable.

Il y a quelques raisons principales pour lesquelles quelque chose qui est théoriquement vulnérable n'est pas réellement exploitable :

- Il se peut que les informations publiques soient insuffisantes pour permettre aux attaquants d'exploiter la vulnérabilité.
- Cela peut nécessiter une authentification préalable ou un accès au système local que l'attaquant n'a pas.
- Les contrôles de sécurité existants peuvent rendre les attaques difficiles.

1.3.4 La gestion des vulnérabilités

La gestion des vulnérabilités est une pratique cyclique qui consiste à identifier, classer, corriger et atténuer les vulnérabilités de sécurité. Les éléments essentiels de la gestion des

vulnérabilités comprennent la détection des vulnérabilités, l'évaluation des vulnérabilités et les mesures correctives.

Les méthodes de détection des vulnérabilités comprennent :

- **Scan des vulnérabilités :** un scanner (ou scanneur) de vulnérabilités est un programme conçu pour identifier des vulnérabilités dans une application, un système d'exploitation, ou un réseau.

- **Test d'intrusion** : Un test d'intrusion (« penetration test » ou « pentest », en anglais) est une méthode d'évaluation de la sécurité d'un système d'information ou d'un réseau informatique ; il est réalisé par un testeur (« pentester », en anglais), la méthode consiste généralement à analyser l'infrastructure d'un réseau informatique, afin de simuler l'attaque d'un utilisateur mal intentionné, voire d'un logiciel malveillant.

Le testeur analyse alors les risques potentiels dus à une mauvaise configuration d'un système d'information, d'un défaut de configuration, de programmation informatique ou encore d'une vulnérabilité liée à la solution testée

- **Google hacking :** Le Google hacking consiste à utiliser un moteur de recherche, tel que Google ou Bing de Microsoft, pour localiser des vulnérabilités de sécurité. Le Google hacking est réalisé par l'utilisation d'opérateurs de recherche avancée dans les requêtes qui localisent des informations difficiles à trouver ou des informations qui sont accidentellement exposées par une mauvaise configuration des services en nuage.

Les chercheurs en sécurité et les pirates utilisent ces requêtes ciblées pour localiser des informations sensibles qui ne sont pas destinées à être exposées au public.

Cela dit, la grande majorité des attaquants auront tendance à rechercher les erreurs de configuration courantes des utilisateurs qu'ils savent déjà exploiter et à rechercher simplement les systèmes qui présentent des failles de sécurité connues.

Pour empêcher le Google hacking, vous devez vous assurer que tous les services en nuage sont correctement configurés. Une fois qu'un objet est exposé à Google, il est public, que cela vous plaise ou non.

Une fois qu'une vulnérabilité est trouvée, elle passe par le processus d'évaluation de la vulnérabilité :

Identifier les vulnérabilités : Analyser les scans du réseau, les résultats des tests de pénétration, les journaux de pare-feu et les résultats des scans de vulnérabilité pour trouver des anomalies qui suggèrent qu'une cyberattaque pourrait tirer profit d'une vulnérabilité.

Vérifier les vulnérabilités : Décider si la vulnérabilité identifiée pourrait être exploitée et classer la gravité de l'exploitation pour comprendre le niveau de risque

Atténuer les vulnérabilités : Décider des contre-mesures et comment mesurer leur efficacité dans le cas où un correctif n'est pas disponible.

Remédier aux vulnérabilités : Mettez à jour les logiciels ou le matériel affecté lorsque c'est possible.

Étant donné que les cyberattaques sont en constante évolution, la gestion des vulnérabilités doit être une pratique continue et répétitive pour que votre organisation reste protégée.

1.3.5 Bases de données de vulnérabilités

Une base de données de vulnérabilités est une plate-forme qui collecte, maintient et partage des informations sur les vulnérabilités découvertes. MITRE utilise l'une des plus grandes bases de données appelées CVE (Common Vulnerabilities and Exposures) et attribue une note du CVSS (Common Vulnerability Scoring System) pour refléter le risque potentiel qu'une vulnérabilité pourrait présenter pour votre organisation.

Cette liste centrale des CVE sert de base à de nombreux scanners de vulnérabilité.

L'avantage des bases de données publiques sur les vulnérabilités est qu'elles permettent aux organisations de développer, de prioriser et d'exécuter des correctifs et d'autres mesures d'atténuation pour corriger les vulnérabilités critiques. (6)

1.4 Gestionnaire de vulnérabilités

Un gestionnaire de vulnérabilités est avant tout un scanner de vulnérabilité qui en plus de scanner, génère des rapports contenants différentes informations à propos des scans effectués tel que les vulnérabilités trouvées, leur criticité, des informations sur les machines scannées …etc. Ces scans peuvent être programmés de façon périodique pour assurer un maximum de sécurité.

1.4.1 Scanner de vulnérabilités

Un scanner de vulnérabilité est un programme informatique qui permet de scanner une application, un système d'exploitation ou un réseau afin de détecter les vulnérabilités existantes. (7)

1.4.2 Fonctionnement des gestionnaires de vulnérabilité

Il existe plusieurs architectures pour les gestionnaires de vulnérabilité, on citera :

1.4.2.1 Moteurs de scans hébergés

Les moteurs de scan hébergés permettent de voir le réseau tel qu'un attaquant externe sans autorisation d'accès le verrait. Ils scannent tout ce qui se trouve à la périphérie du réseau, en dehors du pare-feu. Il s'agit d'actifs qui, par nécessité, fournissent un accès public inconditionnel, tels que les sites web et les serveurs de messagerie électronique. La figure suivante (1) représente l'architecture des moteurs de scan hébergés

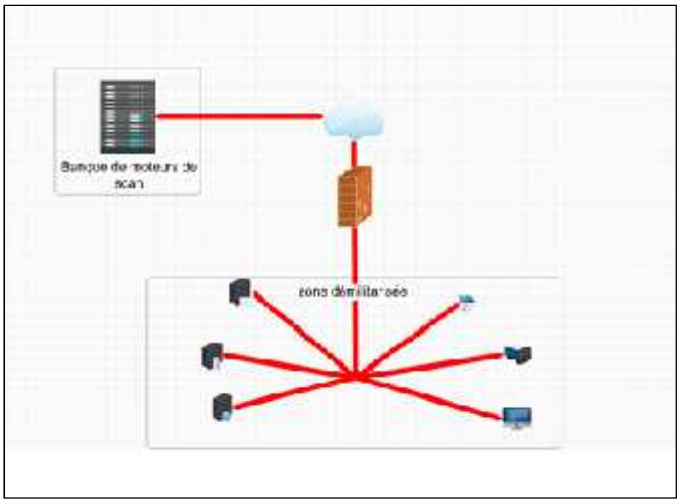

Figure 1 : l'architecture des moteurs de scan hébergés

1.4.2.2 Moteurs de scans distribués

Les moteurs de scan distribués permettent d'inspecter le réseau de l'intérieur. Ils sont idéaux pour les serveurs centraux et les stations de travail. Des moteurs d'analyse distribués peuvent être déployer n'importe où sur le réseau pour obtenir plusieurs vues. Cette flexibilité est

particulièrement précieuse lorsqu'il s'agit d'analyser un réseau comportant plusieurs sous-réseaux, pare-feu et autres formes de segmentation. La figure suivante (2) représente l'architecture des moteurs de scan distribués

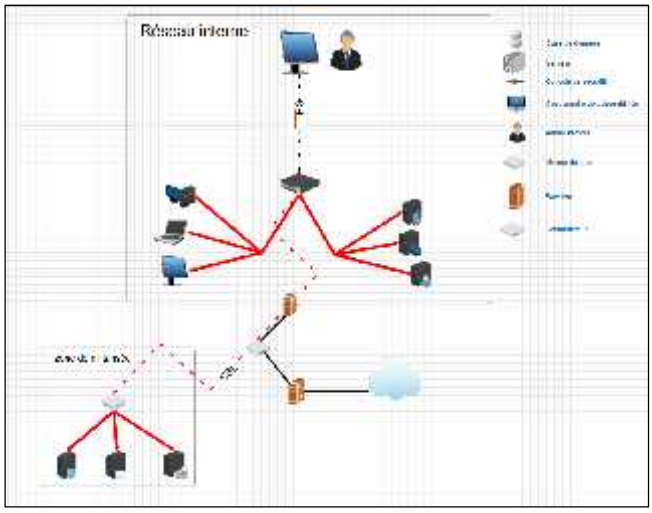

Figure 2 : l'architecture des moteurs de scan distribués

1.4.2.3 Avec agents

La technologie basée sur les agents permet une surveillance et une gestion approfondies. Tout éditeur de logiciel vous dira que ses agents fonctionnent le mieux avec leur plate-forme. Bien que cela puisse être vrai, cela peut aussi être dû au fait que leur plate-forme de gestion est conçue uniquement pour fonctionner avec leurs agents propriétaires.

Le résultat est des fournisseurs fixes, et le changement de fournisseur peut entraîner des déploiements coûteux, à grande échelle et à long terme de technologies de remplacement. Pour cette raison, lorsque les exigences informatiques changent, il peut être extrêmement coûteux de les satisfaire. En général, les normes ouvertes et la flexibilité fonctionnent bien mieux à long terme.

La figure suivante (3) représente un schéma avec l’utilisation des agents

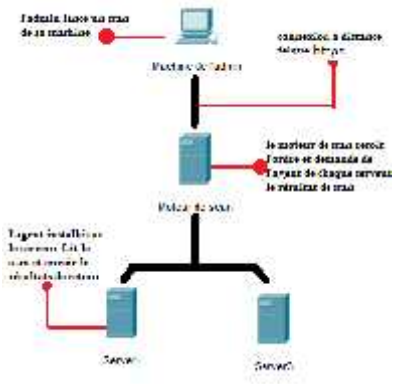

Figure 3: Schéma avec l’utilisation des agents

1.4.2.4 Sans agents

Pour l’architecture sans agent, il y a des agents SNMP intégrés à l'accès distant au Shell, tel que SSH. «sans agent» est un peu inapproprié. Toute gestion nécessite un agent, qu'il soit intégré à la plate-forme de gestion, au périphérique géré ou à un logiciel installé séparément.

L'industrie a accepté la définition de l'agentless comme agent de gestion intégré au logiciel de l'appareil ou comme capacité du gestionnaire, ne nécessitant aucune installation ou licence distincte. La surveillance sans agent signifie vraiment l'utilisation de capacités existantes et intégrées. La figure suivante (4) représente un schéma sans l’utilisation des agents

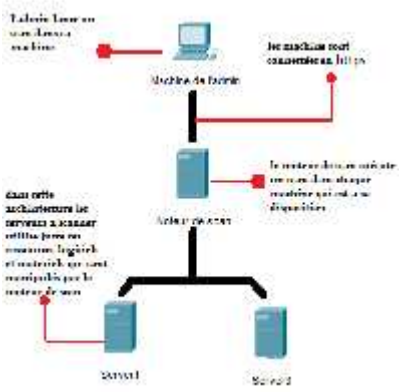

Figure 4 : schéma sans l'utilisation des agents

1.5 Étude des gestionnaires de vulnérabilité existants

Il est nécessaire avant d'essayer de développer un gestionnaire de vulnérabilité, de voir les produits existant sur le marché et les comparer afin d'avoir les critères et les exigences qu'on doit respecter et couvrir.

1.5.1 Les gestionnaires de vulnérabilités choisis pour la comparaison

Commenté [H2]: Arrangez les numéros de titres ainsi !

Pour que cette comparaison soit utile pour l'implémentation du projet, il fallait bien choisir des outils qui sont :

- Basés sur la machine.
- Connus sur le marché.
- Ont des documentations disponibles (afin d'avoir plus d'information pour la comparaison).

Avec les critères ci-dessus, on a choisi ces 4 gestionnaires de vulnérabilités :

- Nessus by tenable.
- Nexpose by rapide7.
- InsightVM by rapide7.
- Vuls scan.

Ci-dessous on va décrit chaque un de ces gestionnaires de vulnérabilités.

1.5.1.1 Nessus (by tenable) :

La société Tenable Network Security est spécialisée dans les produits de surveillance continue et d'évaluation des vulnérabilités tel que Nessus.

- **Architecture et types :**

La gamme de produits d'analyse de vulnérabilité Nessus de Tenable comprend Nessus Cloud, qui est une offre de logiciel en tant que service ; Nessus Manager, une application physique ou virtuelle sur site pour la gestion des vulnérabilités ; Nessus Professional, qui est un logiciel qui s'exécute sur un périphérique client tel qu'un ordinateur portable ; et Nessus Home est une version gratuite destinée aux consommateurs.

- **Tarifs, licences et assistance**

Les produits de la gestion des vulnérabilités Nessus sont des produits basés sur un abonnement annuel vendus via un partenaire commercial ou directement via le Tenable Store en ligne. Les abonnements Nessus Cloud et Nessus Manager ont le même prix par nombre d'hôtes ou d'agents ; 128 hôtes ou agents coûtent 2920 $, par exemple, tandis que 256 hôtes ou agents coûtent 4 745 $. Les clients possédant plus de 256 hôtes doivent contacter un représentant commercial pour connaître les tarifs spécifiques. Chaque abonnement comprend un an de mises à jour logicielles et de mises à jour de vulnérabilité. (8)

1.5.1.2 Nexpose (by Rapide7)

Rapid7 Nexpose est un gestionnaire de vulnérabilité qui vise à prendre en charge l'ensemble du cycle de vie de la gestion des vulnérabilités, y compris la découverte, la détection, la vérification, la classification des risques, l'analyse d'impact, le reporting et l'atténuation.

- **Fonctionnement**

Nexpose Cloud collecte les données de l'ensemble de l'environnement, ce qui permet aux équipes de gérer facilement les vulnérabilités, de surveiller les comportements malveillants, d'enquêter et d'arrêter les attaques et d'automatiser vos opérations.

- **Tarifs, licences et assistance**

Les versions express de Nexpose commencent à 2 000 $ (128 hôtes), une version PRO complète commence à 15 000 $ par an. (9)

1.5.1.3 InsightVM (by Rapide7) :

La plateforme Rapid7 Insight, lancée en 2015, rassemble la bibliothèque Rapid7 de recherche sur les vulnérabilités, exploite les connaissances, le comportement global des attaquants, les données d'analyse à l'échelle de l'Internet, les analyses d'exposition et les rapports en temps réel pour fournir un moyen entièrement disponible, évolutif et efficace de collectez vos données de vulnérabilité et transformez-les en réponses.

InsightVM exploite cette plate-forme pour l'analyse des vulnérabilités et des points de terminaison en direct. Des milliers de clients utilisent cette solution depuis juin 2016, date à laquelle elle a été publiée en BÊTA sous le nom de «Nexpose Now». Le 11 avril 2017, toutes les fonctionnalités de Nexpose Now sont devenues GA et la solution a été rebaptisée InsightVM pour refléter l'innovation passionnante disponible aujourd'hui et demain via des fonctionnalités et des fonctionnalités basées sur le cloud.

- **Tarifs, licences et assistance :**

Insight VM est le plus cher gestionnaire de vulnérabilité entre les quatre produits choisis, il coute au minimum 25$/hôte chaque année, pour 128 hôte 3200$, y compris les mises à jour et les maintenances. (10)

1.5.1.4 Vuls scan :

Vuls est un scanner de vulnérabilités en accès libre et sans agent, écrit en Go. Il automatise l'analyse des vulnérabilités de sécurité des logiciels installés sur un système, ce qui peut être une tâche fastidieuse pour les administrateurs système à effectuer manuellement dans un environnement de production. Vuls utilise plusieurs bases de données de vulnérabilités renommées, telles que la base de données nationale de vulnérabilité (NVD). Peu gourmand en ressources, Vuls peut analyser plusieurs systèmes à la fois et envoyer des rapports par e-mail ou par Slack. Il comporte trois modes d'analyse (fast, fast root et deep), que vous pouvez sélectionner en fonction de la situation.

- Tarifs, licences et assistance :

Vuls scan est gratuit, avec ses mis à jour. (11)

Pour assurer une bonne implémentation d'un outil de de gestion de vulnérabilité, et dans le cadre de définition des besoins, il est exigeant de faire une comparaison entre les outils

existants les plus utilisés, dans le cadre d'avoir une bonne vision sur les points à atteindre par rapport au besoin

1.5.2 Le cadre global de la comparaison

Les critères de comparaison globale sont divisés en portée, type des vulnérabilités, analyse et type de résultat.

1.5.2.1.1 Portée :

La portée d'un outil est la gamme d'un certain outil, et il est divisé en plusieurs sous-critères élaborés dans les sous-sections

1.5.2.1.1.1 Plateformes logicielles :

Les vulnérabilités peuvent résider et résident dans toutes les plates-formes telles que Windows, Unix, Linux, HP / UX, Solaris et Mac OS.

1.5.2.1.1.2 La magnitude :

La magnitude est l'échelle des vulnérabilités gérées par un outil. Il se peut qu'un outil soit uniquement capable d'identifier les vulnérabilités sur un seul hôte, ou que l'outil soit capable d'identifier les vulnérabilités dans une architecture entière au sein d'une organisation.

1.5.2.1.1.3 Les normes :

Les normes suivantes sont les normes les plus courantes pour la gestion des vulnérabilités et nous les expliquerons ci-dessous :

- XCCDF

  The Extensible Configuration Checklist Description Format est un langage de spécification pour écrire des listes de contrôle de sécurité, des tests de performance et des types de documents connexes. (14)

- CVE

  The Common Vulnerabilities and Exposures est une norme pour fournir des noms uniformes à travers les sources de rapports de vulnérabilité [34]. Par exemple. CVE-2007-3168 Résumé : Un certain contrôle ActiveX du composant EDraw Office Viewer (edrawofficeviewer.ocx) 4.0.5.20, et d'autres versions antérieures à 5.0, permet aux attaquants distants de supprimer des fichiers arbitraires via la méthode DeleteLocalFile. (14)

- CPE

The Common Platform enumeration est un schéma de dénomination structuré pour les systèmes, plates-formes et packages de technologie de l'information, fournissant des noms communs pour tous les systèmes logiciels. Cela facilite beaucoup la liaison des différentes vulnérabilités aux systèmes, car tout le monde parle du même système lorsque CPE est utilisé. La structure CPE : cpe: / {part}: {fournisseur}: {produit}: {version}: {mise à jour}: {édition}: {langue} [12]. Un exemple de CPE : cpe: / o: microsoft: windows-nt: 2000: sp4: pro. (14)

• CCE

The Common Configuration enumeration fournit des identificateurs uniques aux problèmes de configuration du système afin de faciliter une corrélation rapide et précise des données de configuration à travers plusieurs sources d'informations et outils. (14)

Le tableau suivant (1) représente un exemple de CCE :

CCE NR	CCE Definition	CCE Platform	CCE Parameters	CCE Technical measures
CCE-871	The "maximum password age" policy should meet minimum requirements.	Microsoft Windows Vista	(1) number of days	(1) defined by Local or Group Policy

Tableau 1:tableau représente un exemple CCE

Un élément de liste CCE contient : le numéro d'identification CCE, une description, les paramètres conceptuels, les mécanismes techniques associés et les citations. Le tableau 1 présente un exemple de CCE :

• CVSS

The Common Vulnerability Scoring system, fournit une évaluation standardisée pour les vulnérabilités présentes dans la base de données NIST (NVD), les scores CVSS se diffèrent entre 0 (gravité faible) et 10 (gravité élevée), Par exemple, CVE-2007-3168 a un score CVSS de 7,8 (élevé). (14)

1.5.2.1.2 Type de vulnérabilités

Les vulnérabilités peuvent résider dans trois domaines différents : la configuration, le code source et l'environnement des systèmes logiciels. La figure suivante (5) représente les différents types de vulnérabilités

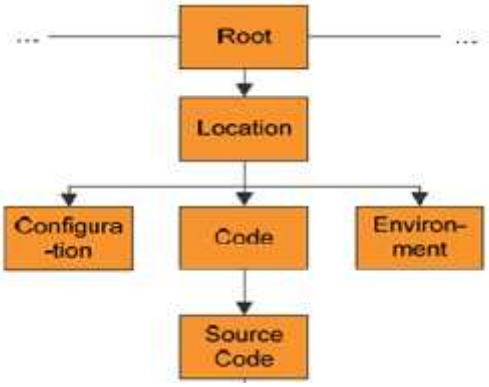

Figure 5: Les type de vulnérabilités

a. Configuration

Les vulnérabilités dans la configuration sont causées par des erreurs de configuration dans le logiciel qui peuvent être exploitées par des attaquants, par exemple, pour l'exécution de code à distance sur la machine attaquée.

b. Code source

Les vulnérabilités dans le code source sont causées par des erreurs dans le code source du logiciel. Ces erreurs pourraient, par exemple, permettre de provoquer une erreur de « buffer overflow » pouvant endommager le système attaqué.

1.5.2.1.3 L'analyse :

Les critères de cette sous-section précisent le type d'analyse qu'un outil fournit pour trouver et évaluer les vulnérabilités des produits informatique standard (COTS based system 'CBSs').

Nous distinguons quatre types d'analyses effectuées par les outils de gestion de vulnérabilités: vérification de la conformité, gestion des patchs, analyse de vulnérabilité et analyse corrélée, comme expliqué ci-après. Il convient de mentionner que lorsqu'un outil effectue un type d'analyse, cela ne signifie pas qu'un autre type d'analyse est exclu.

a. Vérification de la conformité

Actuellement, il existe plusieurs cadres et lois de conformité à la sécurité, tels que HIPAA, SOX, ISO17799, GBLA, FDCC, FISMA, PCI. Les outils peuvent automatiser le processus de vérification de la conformité avec ces Frameworks.

b. Gestion des patchs

Les outils qui en relèvent ne donnent qu'une liste, par ex. correctifs installés, logiciels installés, correctifs manquants, etc. Ainsi, seul un inventaire d'un hôte est créé, mais aucune évaluation supplémentaire n'est effectuée avec ces informations.

c. Analyse corrélée

Il est également possible qu'un outil corrèle ou agrège des vulnérabilités avec d'autres informations (ou vulnérabilités) afin d'effectuer une meilleure analyse de vulnérabilité. Il existe deux types de corrélation de vulnérabilité actuellement discutés dans la littérature. Premièrement, corrélation entre différents types de dispositifs de sécurité comme les IDS et les pare-feu. Deuxièmement, corrélation entre plusieurs vulnérabilités qui peuvent être utilisées comme tremplins pour un scénario d'attaque.

1.5.2.1.4 Types de résultats

De nombreux outils créent une vue d'ensemble du score de vulnérabilité combiné d'un système en termes de vulnérabilité d'un système ou d'une architecture. Cela peut être fait soit qualitativement, par exemple avec une couleur allant du vert au rouge, ou quantitativement avec un nombre qui peut être spécifique au fournisseur et / ou basé sur un score CVSS.

1.5.3 Comparaison entre les différents outils existants

En utilisant les critères ci-dessus, on a comparé les 4 gestionnaires des vulnérabilités choisis, les résultats sont dans le tableau suivant (2) :

vendeur	produit	La Plateforme			La magnitude		Les standards					Les types d'analyse					Les types de résultats	
		Windows	Linux	Autre	Utilisateur simple	Entreprise	XCCDF	CVE	CPE	CCE	CVSS	Configuration	Code source	Vérification de la conformité	Gestion des patchs	Analyse corrélé	Quantitative	Qualitative
TENABLE	NESSUS	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X
RAPIDE7	NEXPOSE	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X
RAPIDE7	INSIGHT VM	X	X	X	X	X	X	X	X	X	X	X	X		X		X	X
	VULS SCAN	X	X	X	X	X		X	X	X	X	X	X				X	X

Tableau 2 : Comparaison entre les différents outils existants

On constate que les 4 gestionnaire de vulnérabilités sont multiplateforme, ils sont aussi dédiés aux entreprises comme aux particuliers et respectent tous les standards du SCAP (Security Content Automation Protocol), ils permettent aussi l'analyse de la configuration et des codes source, et ils fournissent des résultats qualitatifs et quantitatifs, quant à la vérification de la conformité, seul NEXPOSE et NESSUS le font et en ce qui concerne la gestion des patchs NESSUS et INSIGHTVM l'assurent.

On remarque aussi que aucun des gestionnaire de vulnérabilité ne fait de l'analyse corrélé.

Ils sont tous excessivement chère pour une entreprise tel que ELIT, sauf « VULS SCAN » qui est open-source, mais là y a une possibilité de le contourner vue que les malfaiteurs peuvent aussi analyser son code et trouver des failles.

Ces gestionnaires sont propriétaires, sauf « VULS SCAN » c'est-à-dire que l'entreprise ne peut pas être sure qu'aucune fonctionnalité malveillante ne fonctionne en arrière-plan, chose qui intrigue ses dirigeants, quant à « VULS SCAN » et comme cité dans le paragraphe précédent, même les malfaiteurs peuvent analyser son code source et ainsi trouver des failles.

Notre solution devrait être multiplateforme, respectant tous les standards SCAP sauf le XCCDF qu'on a jugé pas nécessaire, elle est dédiée aux entreprises comme aux particuliers, génère des résultats qualitatifs et quantitatif, elle n'inclut pas l'analyse corrélé, l'analyse des code sources, ni la gestion des patchs. Par contre il est gratuit pour ELIT, et ils ont le code-source, l'entreprise peut scanner son code et s'assurer qu'il est sûr. ELIT aura le droit aussi de le vendre à d'autres entreprises au prix qu'elle juge convenable.

1.6 Conclusion

Nous avons vue dans ce chapitre les définitions essentielles pour comprendre le sujet, parlé de généralités sur la sécurité informatique, et sur les vulnérabilités, nous avons exposé aussi le fonctionnement des gestionnaires de vulnérabilité et nous avons terminé par une étude sur les gestionnaire de vulnérabilités existants et une comparaison entre eux.
Nous parlerons dans le chapitre suivant des choix conceptuels de la solution proposée.

CHAPITRE II : CONCEPTION DU GESTIONNAIRE DE VULNERABILITE « ELITVM »

2.1 Introduction

La première étape à suivre dans le développement d'un outil informatique est la bonnen conception.

Comme on a mentionné dans le chapitre précédent le gestionnaire de vulnérabilité que nous allons réaliser doit détecter les vulnérabilités dans les systèmes des machines du réseau « ELIT », en respectant la politique de l'entreprise et les aspects de la sécurité informatique.

Il est obligé alors de bien définir les actions du système, les données a traité, les acteurs principaux, composants principaux et le matériel de déploiement, tout ça par rapport à l'environnement.

Après définir les points précédents, on doit parler du langage de programmation et du type de stockage des données (type de BDD).

Cette travail va nous aider à avoir une bonne vision et facilité la réalisation d'un gestionnaire de vulnérabilité fonctionnel qui couvre des besoins que nous allons définir, et compatible avec l'environnement de l'entreprise.

2.2 Environnement

Sur le Datacenter d'ELIT, on trouve deux environnements, celui de test, et celui de production, et avant d'exploiter réellement un serveur, ELIT l'exploite et le teste dans un environnement de test où on cherche les différentes vulnérabilités existantes. Après avoir passé les tests, ce dernier sera déplacé vers l'environnement de production où il sera exploité et utilisé par les personnes concernées.

En résumé :

- L'environnement de PRODUCTION où ELITE héberge les serveurs validés dans l'environnement de test

- L’environnement de TEST où ELITE teste les serveurs avant de les exploiter dans l’environnement de production.

Chaque environnement contient plusieurs sous réseau et un nombre énorme de machines, alors l’objectif de notre système est d’assurer que toutes les machines des deux environnements soient bien sécurisées et de rapporter toutes les vulnérabilités qui existent à l’administrateur système avant qu’elle ne soient exploitées.

Elit utilise un système de gestion de vulnérabilité -dont on garde le nom anonyme comme demandé- développer par une entreprise étrangère, elle veut donc avoir son propre système de gestion de vulnérabilité pour éviter un potentiel risque de vol de données.

2.3 Problématique

La disponibilité, l’intégrité et la confidentialité des données est désormais ce qu’il y’a de plus important pour une organisation.

Mais l’hétérogénéité des systèmes, leur complexité et l’erreur humaine rendent l’inexistence des vulnérabilités un fait ressortant de l’impossible.

Si une vulnérabilité est exploitée, les attaquants pourront alors voler les données, les détruire, ou bien faire un déni de service ce qui rend les données indisponibles. Cela peut causer des dégâts partiels, voire total à l’organisation, cette dernière est, en conséquence, dans l’obligation d’assurer la protection de son système contre les cyberattaques.

Et pour cela il faut détecter et remédier aux vulnérabilités existantes avant qu’un malfaiteur ne les exploite, par contre il est impossible de le faire manuellement, surtout dans des systèmes complexes.

Il fallait donc penser à automatiser la détection des vulnérabilités, on a vu alors naitre les systèmes de scan des vulnérabilités puis les systèmes de gestion des vulnérabilités.

2.4 Description du projet

Notre projet consiste à concevoir et réaliser un système de gestion de vulnérabilités pour l'établissement d'accueil, à savoir "EL-djazair Information technology".

Ce système devra permettre à l'entreprise de détecter différents types de vulnérabilités avant qu'elles ne soient exploitées par des malfaiteurs, et de proposer des correctifs.

2.5 Identification des besoins

Les besoins exprimés par l'entreprise sont les suivants :

- Scanner les différents serveurs de l'entreprise à la recherche de vulnérabilités.
- Assurer deux types de scan :
  - Un scan avec authentification : scanner la machine cible en ayant accès à cette dernière.
  - Un scan sans authentification : scanner la machine cible de l'extérieur, c'est-à-dire sans avoir les identifiants pour s'authentifier.
- Le Scan avec authentification permet de détecter les vulnérabilités du SE, les ports ouverts, les services actifs, les méthodes http autorisés, et les chiffrements obsolètes.
- Le scan sans authentification nous permet quant à lui de détecter les ports ouverts, les services actifs, les méthodes http autorisés, et les chiffrements obsolètes.
- Développer une application en interne qui permet la gestion des scans, et donc d'y voir les résultats.
- L'application en question permet aux administrateurs systèmes de mieux voir l'état des machines qu'ils administrent.
- Générer et archiver des rapports de scan.
- Etablir une interaction efficace entre les administrateurs systèmes et les responsables de la sécurité des serveurs.
- Sécuriser le stockage et la transmission des données.
- Assurer la disponibilité, l'intégrité et la confidentialité des données traitées.
- Etre à jour avec les vulnérabilités découvertes.

Après plusieurs visites à l'entreprise nous avons défini trois acteurs principaux ayant accès au nouveau gestionnaire de vulnérabilité à savoir :

- L'administrateur principal : cet administrateur peut gérer les comptes des utilisateurs, les moteurs de scans et les sites, c'est l'acteur principal de l'application, et d'après la politique de l'entreprise « ELIT » le système doit avoir un seul administrateur principal. Ce dernier est désigné parmi les ingénieurs qualifiés de « ELIT ».
- L'administrateur de site : c'est le chargé de la gestion des scans, il lance qui consulte les résultats des scans, il consulte aussi le tableau de bord des sites qu'il gère, un site dans « ELIT » est ……..

Commenté [H3]: Complète la définition d'un site tu mets des termes qui ne sont pas clairs au lecteur.

- L’administrateur système : cet administrateur a un accès physique aux machines, il est chargé de rétablir l’états des machines dès qu’il les répare, en plus, il gère la table des identifiants qui contient toutes les informations des machines y compris leurs logins.

Chaque acteur de notre nouveau système a des droits d’accès et des fonctionnalités bien définies, ces dernières répondent aux différents besoins de l’entreprise, le diagramme de cas d’utilisation suivant représentera les principaux besoins par acteur. La figure suivante (6) représente le diagramme de cas d’utilisations.

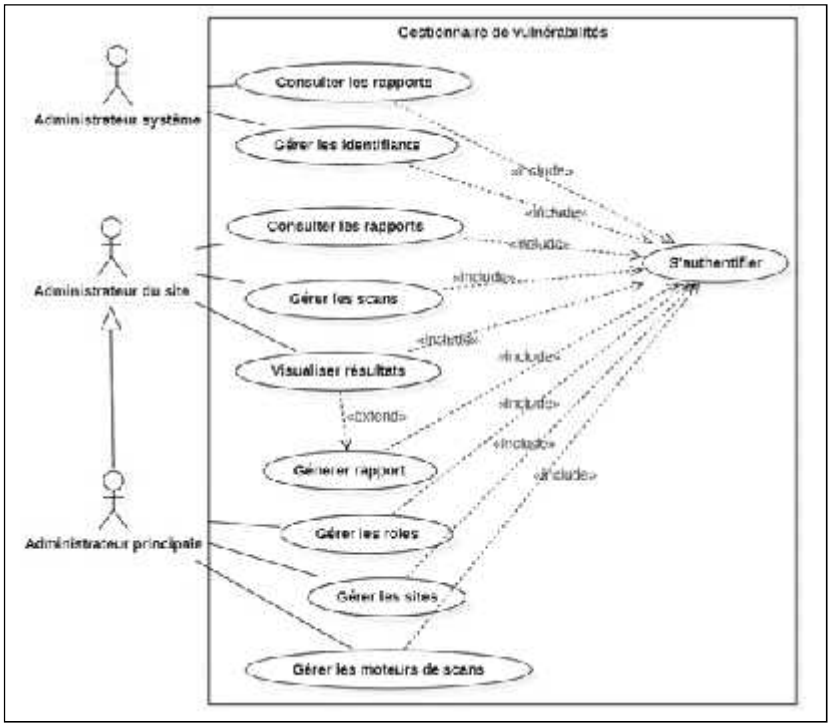

Figure 6 : diagramme de cas d'utilisation

Description des cas d’utilisations apparues dans la figure ci-dessus :

a- Consulter les rapports :

- L’administrateur du site peut consulter à tout moment n’importe quel rapport qu’il a déjà généré.

- L’administrateur système peut consulter à tout moment n’importe quel rapport généré à propos d’une machine qu’il administre.
- L’administrateur principale peut quant à lui peut à tout moment consulter n’importe quel rapport généré.

b- Gérer les identifiants (machines) :

- L’administrateur système gère les identifiants des machines qu’il administre, il peut ajouter un identifiant ou modifier la base de données contenante tous les identifiants. La suppression n’est pas permise pour garder l’historique des scans.

c- Gérer les scans :

-L’administrateur du site gère les scans des machines se trouvant sur son site, ceci inclus la sélection des machines a scanné, le type de scan(avec ou sans authentification), et la périodicité des scans.

d- Consulter le tableau de bord :

-L’administrateur du site peut à tout moments le tableau de bord des scans qu’il a déjà effectué, il peut les visualiser par plage d’adresse IP, le dernier scan effectué, par CVE, par criticité ou par période tel que les résultats des scans effectués cette semaine, cela lui permettra d’avoir une vue d’ensemble, sur les machine de son site.

e- Générer rapport :

-L’administrateur du site peut après avoir visualiser les résultats, générer un rapport -s’il juge cela nécessaire- que l’administrateur système consultera, et prendra les bonnes décisions pour fixer les vulnérabilités.

f- Gérer les rôles :

-L’administrateur principale est le seul apte à gérer les rôles, ceci inclus la désignation des administrateurs, leurs droits d’accès en les affilant aux sites, mais aussi leurs destitutions.

g- Gérer les sites :

-L’administrateur principale est le seul apte à gérer les sites, ceci inclus la création de sites, modification, La suppression n’est pas permise pour garder l’historique des scans.

h- Gérer les moteurs de scans :

-L’administrateur principale est le seul apte à gérer les moteurs de scan (ajout et modification).

2.6 L’architecture choisie pour le produit

Commenté [H4]: Ou est l’architecture en elle-même ?

Il est évident que l’architecture choisie pour notre produit doit être en fonction de l’environnement, et du domaine de l’application, pour assurer son bon fonctionnement. La figure suivante (7) représente l’architecture du système proposé.

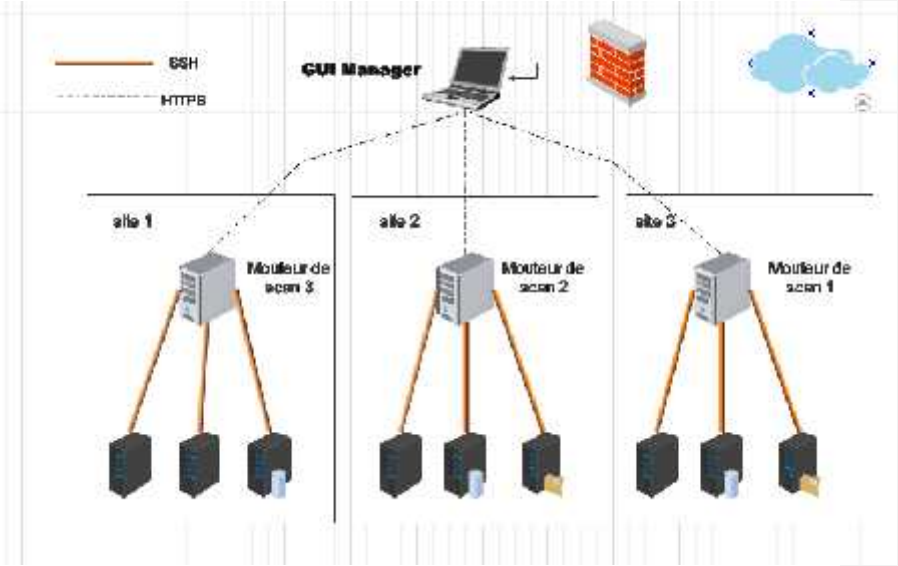

Figure 7 Architecture de lu système proposé

La fonction principale de l’application est la gestion des scans de tous les serveurs d’ELITE, qui sont très nombreux, placés dans plusieurs sous réseau séparés et isolés de l’internet. En plus de ça il faut prendre en compte que l’application ne doit pas occuper la bonde passante sur le réseau, pour éviter l’interruption de la transmission des donnés au sein de l’entreprise.

L’application doit collecter les informations des scans de tous les nœuds, et assurer la mise à jour de la base de données de vulnérabilité à partir d’internet.

A partir de tout ce qu'a été cité dans ce chapitre, nous concluons que notre système de gestion de vulnérabilités doit être :

a- 3-tiers :
    - GUI Manager : qui est une partie principale de la solution installée dans un nœud séparé, il permet la gestion de tout le système, la mise à jour de la liste des CVE et son envoie aux moteurs de scan.
    - Moteur de scan : installé dans plusieurs nœuds, chaque moteur de scan effectue les scans sur des machines qui lui sont autorisé à scanner, et envoie le résultat au GUI Manager
    - Machine à scanner : ce sont les serveurs d'ELITE qu'on va scanner

Cette solution permet de minimiser le temps du scan et facilité l'accès du moteur de scan aux machines à scanner. La base de données de vulnérabilités qui sera mis à jour à partir du nœud manager est aussi transmise et stockée dans les moteurs de scan.

b- Distribuée : la partie principale de l'application est installée dans le manager, et elle manipule la partie qui fait le scan et qui est placée dans les moteurs de scans, ces moteurs de scans sont placés dans chaque sous réseau. Cette solution assure la rapidité de l'exécution du scan sans trop occuper la bande passante ni trop chargé les ressources, vue que notre système utilise des ressources de plusieurs machines.

c- Sans agent : les points finaux dans notre système qui sont les serveurs a scanné ne contiennent pas d'agents, ils sont scannés à distance à partir des moteurs de scan du même site, même si l'installation de l'agent qui effectue le scan minimise un peu le trafic qui occupe la bonde passante, on a éliminé cette solution car il est très difficile d'installer des agents dans tous les points finaux qui se comptent par milliers.

A noter aussi que tous les canaux de communication entre les moteurs de scan et les machines à scanner et entre Serveur fournisseur de CVE et les moteurs de scan sont sécurisés via le protocole SSH, quant aux canaux de communication entre le GUI Manager et les moteurs de scans, ils sont sécurisés via le protocole HTTPS. Cela veut dire que tout ce qui circule dans le réseau est crypté et sécurisé.

2.7 Interaction Homme-Machine

Comme illustré dans la section précédente, notre outil est doté d'une interface graphique permettant les interactions entre les différents acteurs du système afin de réaliser les différentes fonctionnalités répondant aux besoins prédéfinis.

Nous présentons dans cette section ces différentes interactions à travers les diagrammes de séquences de UML, ces derniers sont très illustratifs et formels, ils permettent une meilleure vision des interactions effectuées.

Nous commençons alors avec le cas d'utilisation principale dans notre système qui est « Effectuer un scan »

2.7.1 Effectuer un scan

Le diagramme de séquence qui suit (Figure 9) détaille ce qui se passera quand l'administrateur du site voudra effectuer un scan sans le programmé, sur une machine qui se situe dans un site auquel il est affecté.

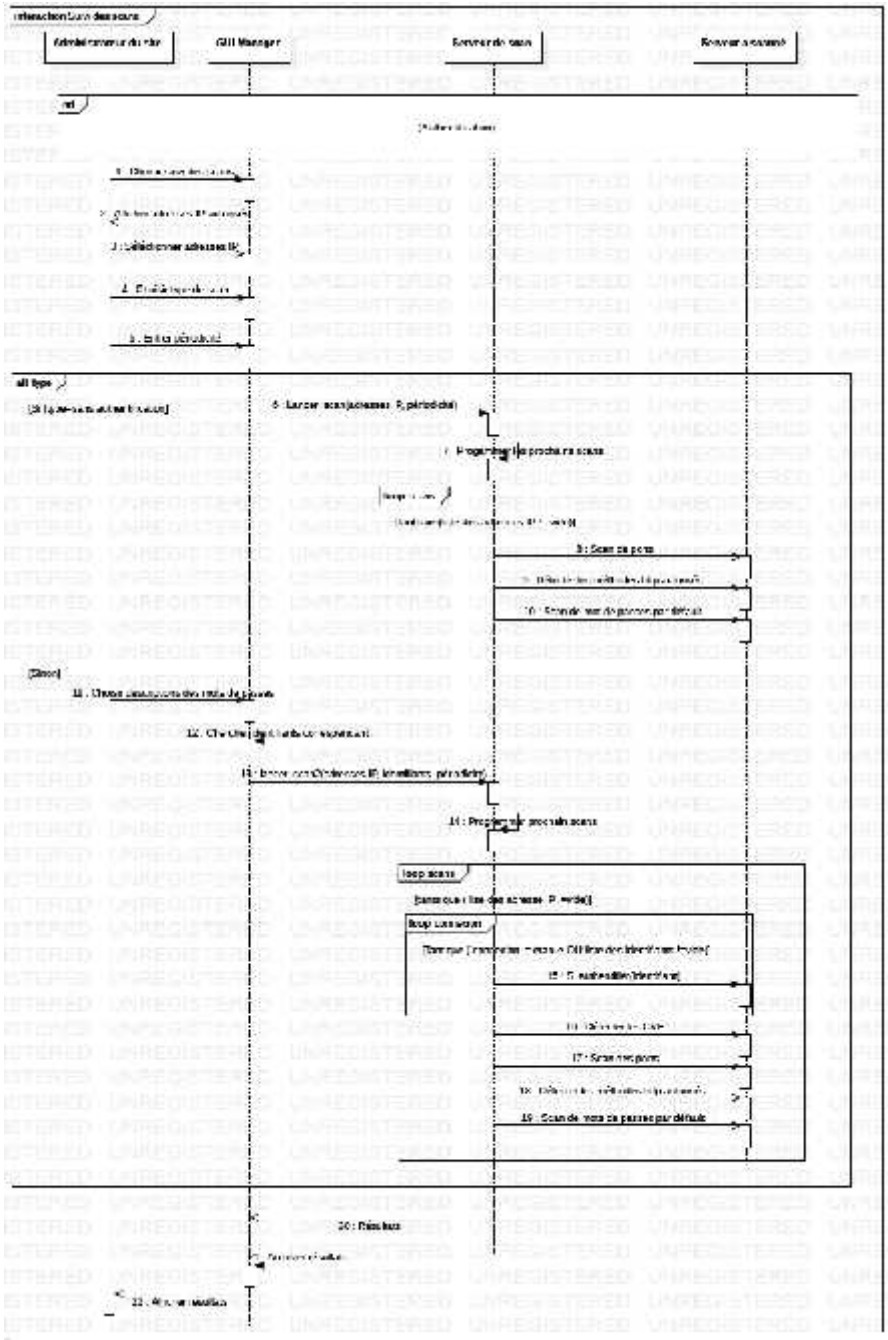

Figure 8 Diagramme de séquence montrant comment effectuer un scan non programmé

Commenté [H5]: On met le diagramme et puis les scénarios et non l'inverse !
Corrige pour tout ce qui suit.

Le tableau suivant (3) détaillera ce qu'on a vue plus haut dans la figure (8) et aidera pour sa compréhension :

<table>
<tr><td colspan="2">Scénario 1: Effectuer un scan non programmé d'un serveur</td></tr>
<tr><td colspan="2">Objectif : Montrer les interactions du système et ses principales opérations</td></tr>
<tr><td colspan="2">Acteurs : Administrateur du site</td></tr>
<tr><td colspan="2">Préconditions : L'administrateur du site doit exister dans la base de données, le serveur que l'administrateur souhaite scanner soit sur le même site que l'administrateur, connaître l'identifiant pour accéder à la machine à scanné.</td></tr>
<tr><td colspan="2">Post conditions : Envoyer le rapport à l'administrateur système responsable sur ce serveur et à l'administrateur principal</td></tr>
<tr><td>Scénario nominal :
1 : l'administrateur s'authentifie au système
2 : le système valide l'authentification
3 : l'administrateur choisi l'option « suivi du scan »
4 : le système affiche les adresses ip autorisé
5 : l'administrateur lui envoie les adresses IP des machines cibles pour les scanner
6 : l'administrateur choisis le type de scan
7 : l'administrateur entre la périodicité s'il veut des scans programmés, sinon il entre la valeur 0
8 : si l'administrateur choisis un scan sans authentification GUI Manager envoie alors au moteur de scan concerné les adresses ip à</td><td>14 : si l'administrateur choisis un scan avec authentification GUI manager envoie les adresses ip + la périodicité + les identifiants (login + mots de passe) de toutes les machine du site de l'administrateur
15 : si la périodicité > 0 le moteur de scan programme les scans suivants sinon il passe à l'étape 16
16 : le moteur de scan parcourt toutes les machines ciblées en appliquant les étapes 17 jusqu'à 21
17 : le moteur de scan essaye avec les identifiants envoyer par GUI Manager à accéder en SSH a la machine cible
18 : le moteur de scan cherche les paquets vulnérables qui sont associé à des CVE</td></tr>
</table>

scanner + la périodicité 9 : si la périodicité > 0 le moteur de scan programme les scans suivants sinon il passe à l'étape 10 10 : le moteur de scan parcourt toutes les machines ciblées en appliquant les étapes 11,12,13 11 : le moteur de scan scanner et découvrir tous les ports ouverts et leurs services 12 : le moteur de scan détecte toutes les méthodes http autorisées 13 : le moteur de scan découvrir s'il y a des mots de passe par défaut	19 : le moteur de scan scanner et découvrir tous les ports ouverts et leurs services 20 : le moteur de scan détecte toutes les méthodes http autorisées 21 : le moteur de scan découvrir s'il y a des mots de passe par défaut 22 : dès que le scan finis le moteur de scan collecte et envoie le résultat au GUI Manager 23 : GUI manager archive le résultat 24 : GUI manager affiche le résultat a l'administrateur

Tableau 3 : Scénario nominal pour effectuer un scan non programmé

2.7.2 Consulter un rapport

Cette action permet à l'administrateur système de consulter les rapports des scans déjà effectués, la figure suivante (9) représente le diagramme de séquence montrant comment visualiser un rapport.

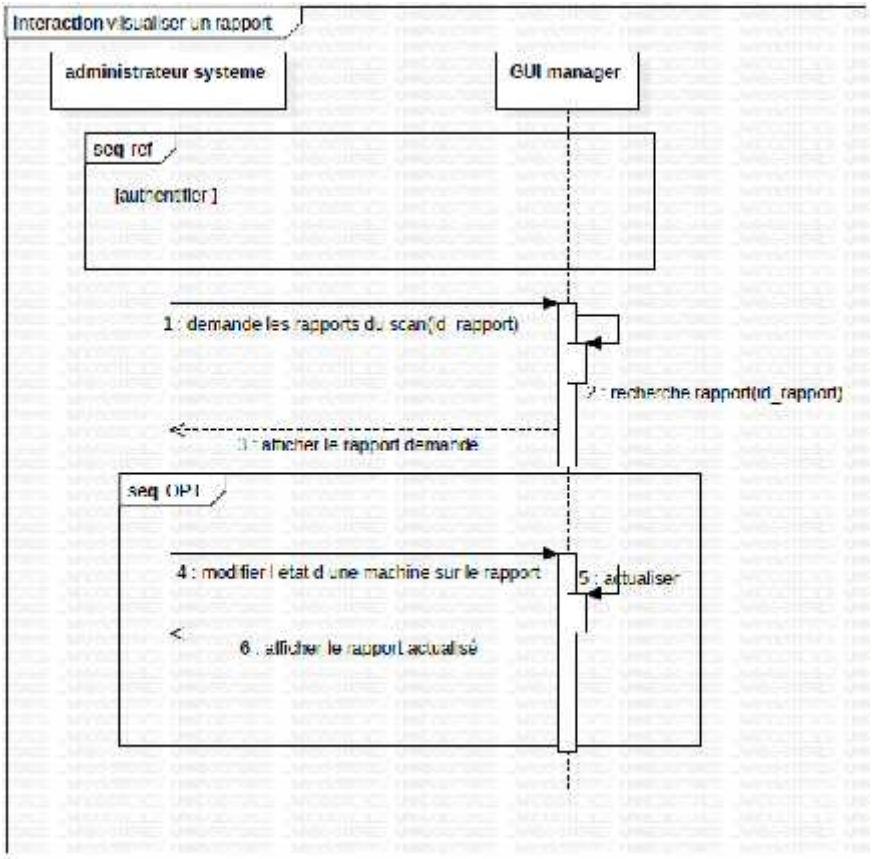

Figure 9 : Diagramme de séquence montrant comment visualiser un rapport

Le tableau suivant (4) détaillera ce qu'on a vue plus haut (figure 9) et aidera pour sa compréhension, il représente le scénario nominal de « consulter un rapport »

Scénario 1: visualiser un rapport	
Objectif : Montrer comment un administrateur système peut consulter les rapports des scans	
Acteurs : Administrateur système	
Préconditions : L'administrateur système doit exister dans la base de données, et pour demander un rapport du scan il doit connaître leur identifiant.	
Post conditions : actualiser le statut du rapport.	
Scénario nominal : 1 : l'administrateur s'authentifie au système 2 : le système valide l'authentification 3 : l'administrateur cherche un rapport en donnant leur identifiant. 4 : GUI Manager recherche le rapport demandé par l'utilisateur dans sa BD de résultats. 5 : GUI Manager affiche en retour le rapport trouvé à l'administrateur, avec la possibilité de le modifier.	6 : l'administrateur système rétablis l'état des machines vulnérables. 7 : GUI Manager actualise la nouvelle version du rapport et le stocke. 8 : GUI Manager affiche le nouveau rapport à l'administrateur système.

Tableau 4: Scénario nominal pour consulter un rapport pour l'administrateur système

2.7.3 Gérer les identifiants :

Cette action permet à l'administrateur système de gérer les identifiants de serveurs, c'est-à-dire ajouter un compte, modifier un compte. La figure suivante (10) représente le diagramme de séquence de « Gérer les identifiants »

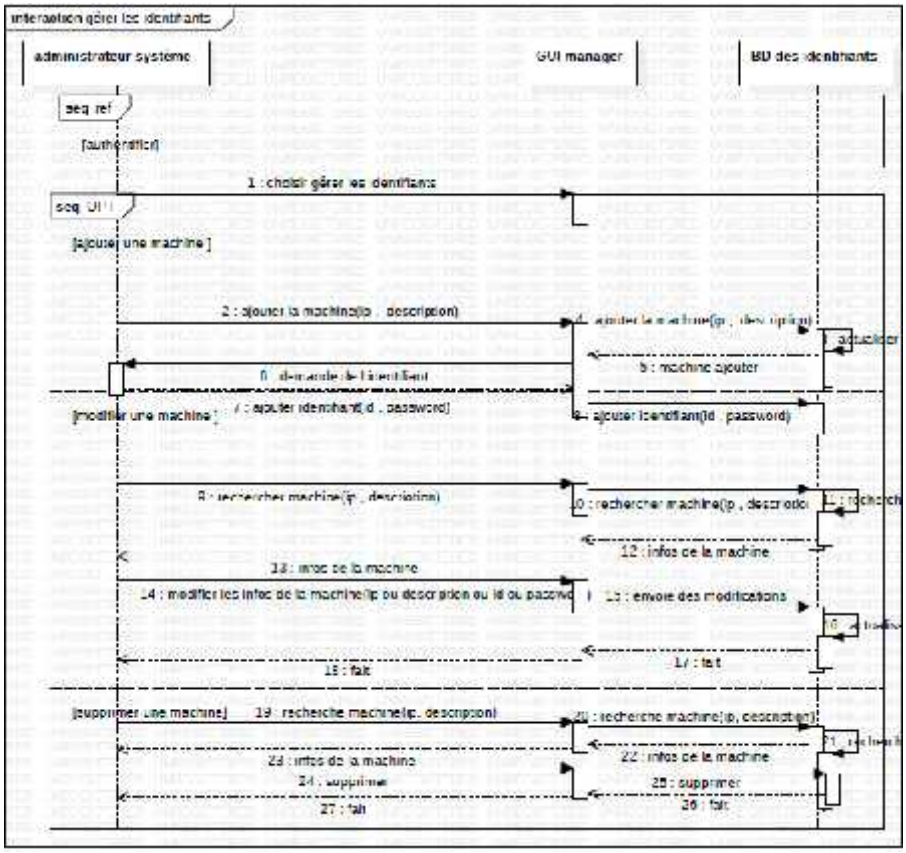

Figure 10: Diagramme de séquence montrant comment gèrer les identifiants

Le tableau suivant (5) détaillera ce qu'on a vue plus haut (figure 10) et aidera pour sa compréhension, il représente le scénario nominal de « Gérer les identifiants »

<table>
<tr><td colspan="2">Scénario 1: gérer les identifiants</td></tr>
<tr><td colspan="2">Objectif : Montrer comment notre système gère les identifiants des machine cibles</td></tr>
<tr><td colspan="2">Acteurs : Administrateur système</td></tr>
<tr><td colspan="2">Préconditions : L'administrateur système doit exister dans la base de données et il doit aussi connaitre les identifiants des machines qui sont à sa disposition.</td></tr>
<tr><td colspan="2">Post conditions : ajouter ou supprimer ou modifier les informations générale d'une machine cible (id, mot de passe, description).</td></tr>
<tr><td>Scénario nominal :

1 : l'administrateur s'authentifie au système

2 : le système valide l'authentification

3 : l'administrateur choisis l'option de gestion des identifiants.

4 : GUI Manager affiche l'interface correspondante avec trois options : ajout, modification et suppression.

5 : si l'administrateur système choisis l'ajout, il doit saisir alors les informations sur la nouvelle machine (ip, description). La description contient des informations sur la machine physique (@ MAC, Nom de la machine) et d'autre sur le système (nom et version de l'OS).

6 : GUI Manager reçoit ces informations et les envoie vers la BD des identifiants.

7 : la BD des identifiants vérifie l'existence</td><td>15 : la BD des identifiants vérifie l'existence de cette machine, si elle existe la BD des identifiants envoie un message « machine trouvée ».

16 : GUI Manger donne la main à l'administrateur pour modifier les informations de la machine (ip, description) ou de changer l'id et le mot de passe.

17 : l'administrateur saisie les informations modifiées (ip, description, id, password).

18 : GUI Manager envoie les modifications à la BD des identifiants et hacher le mot de passe s'il a été changé.

19 : la BD des identifiants envoie un message « fin de modification » à GUI manager.

20 : GUI manager renvoie le message à l'administrateur.

21 : si l'administrateur système choisis la</td></tr>
</table>

de cette machine, si elle n'existe pas il la stocke. 8 : la BD des identifiants envoie un message « ajout succès » a GUI Manager. 9 : GUI Manager transmis le message à l'administrateur système en demandant l'identifiant et le mot de passe de la machine. 10 : l'administrateur système saisie l'id et le mot de passe de la nouvelle machine 11 : GUI Manager transmis l'id et le mot de passe (crypté) à la BD des identifiants. 12 : la BD des identifiants stocke l'id et le mot de passe associés à la machine ajoutée. 13 : si l'administrateur système choisis la modification, il doit saisir alors les informations sur la machine à modifier (ip, description). 14 : GUI Manager reçoit ces informations et les envoie vers la BD des identifiants.	suppression, il doit saisir alors les informations sur la machine à modifier (ip, description). 22 : GUI Manager reçoit ces informations et les envoie vers la BD des identifiants. 23 : la BD des identifiants vérifie l'existence de cette machine, si elle existe la BD des identifiants envoie un message « machine trouvée ». 24 : GUI Manger donne la main à l'administrateur pour supprimer la machine. 25 : l'administrateur confirme la suppression. 26 : GUI manager envoie l'ordre de suppression a la BD des identifiants. 27 : la BD des identifiants supprime la machine. 28 : la BD des identifiants envoie un message « fin de suppression ». 29 : GUI manager transmis le message à l'administrateur système.

Tableau 5 : Scénario nominal pour gérer les identifiants des machines.

2.8 Notre moteur de scan

Le moteur de scan est un serveur avec des capacités élevées, ce serveur contient un script de scan des vulnérabilités, et il a accès à toute machine à sa disposition via le protocole SSH.

L'algorithme qui gère le scan dans le moteur de scan se base sur quatre points :

- Le parcours de toutes les machines.

Commenté [H6]: C'est votre proposition conceptuelle, cette partie devra etre dans le chapitre 2 juste avant conception de la base de donnée !!!!!!!!!!!!!

Mettez « Notre moteur de scan » comme titre, et enlevez le mot réalisation de cette section et mettez proposition conception….!!!!

C'est le plus important de tout votre travail !!!!!!!!!!

Dans la partie architecture de notre solution dans le chapitre précédent on devra trouver un composant moteur de scan !!!!!!
Aussi dans le chapitre précédent Il manque le canal de communication SSH !!!!! vous avez omis tout l'essentiel de votre travail en mettant les diagrammes UML alors que c ça l'essentiel !!!
Vous ne palez plus du choix agent less ……
Et dans ce chapitre dans cette partie mettez au pire une capture du code source du moteur de scan, enfin mettez des captures de lui, mais l'algorithme c la conception et non la réalisation !!!!!!!

- L’authentification : le moteur de scan demande l’identifiant de chaque machine à scanner, le gestionnaire des mots de passe lui donne les identifiants des machines si le scan est approfondi, sinon les machines devront être scanner sans authentification.
- La détection des vulnérabilités : c’est le corps de notre solution, et il y’a quatre types de détection incluses dans notre solution :
  -Le scan des CVE.
  -Le scan des ports.
  -Les mots de passe par défaut.
  -Le méthode http.
- La génération des rapports du scan, après chaque scan, le moteur du scan génère une variable de type dictionnaire contenant les résultats trouvés et l’envoie au GUI Manager, cette variable respecte la forme suivante pour faciliter le traitement textuel des résultats. L’arborescence est représentée dans la figure suivante (11) :

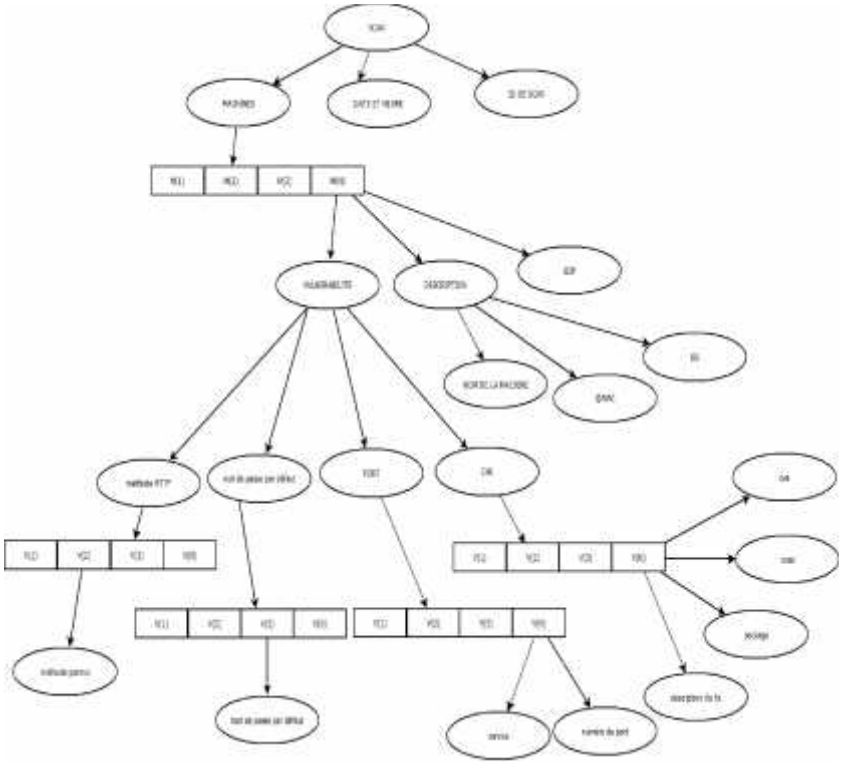

Figure 11: L’arborescence de variable de résultat

La figure suivante (13) représente l'organigramme de l'algorithme du scan :

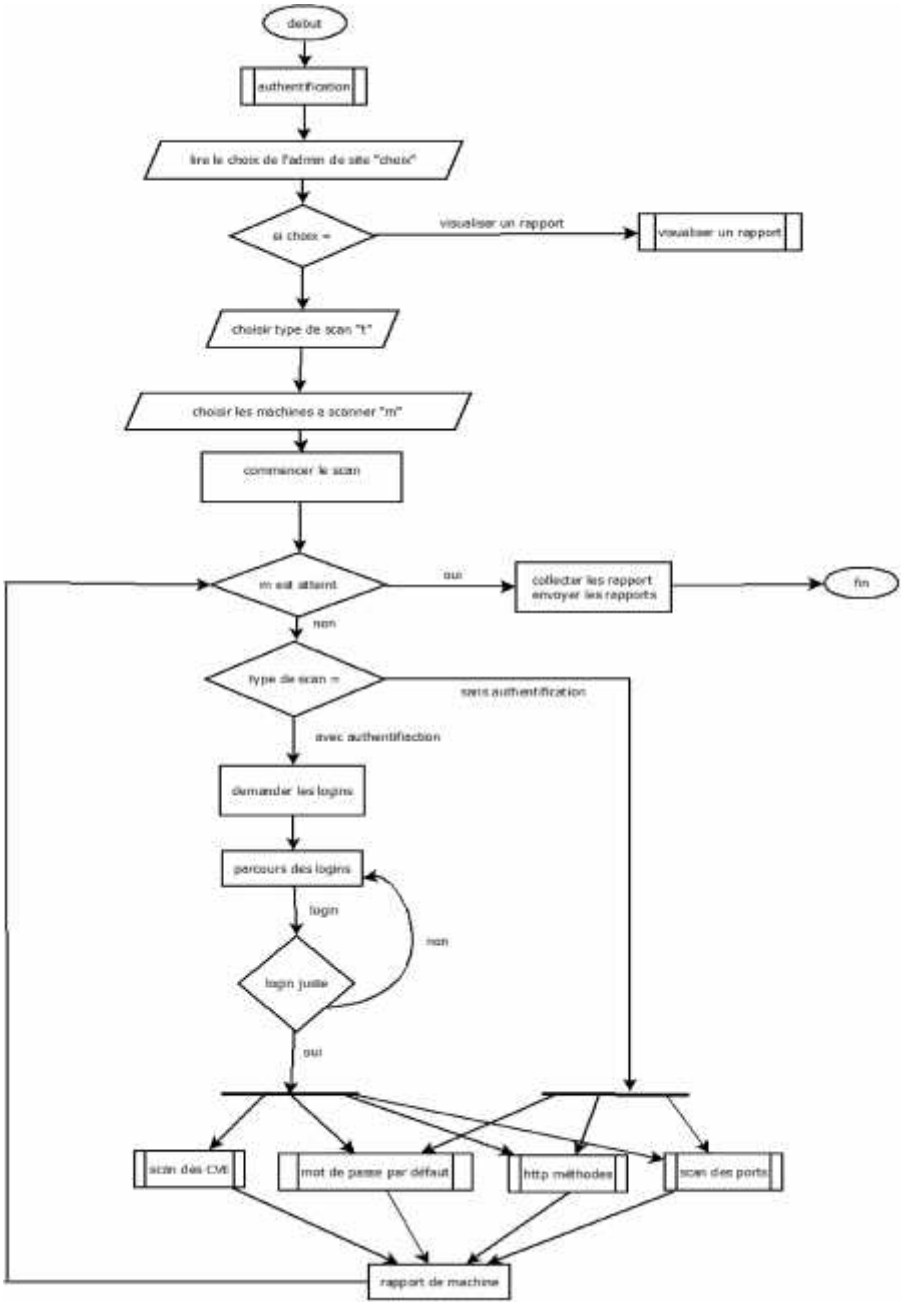

Figure 12: Organigramme représentant l'algorithme du scan

2.9 Conception de la base de données

Notre solution propose à l'entreprise la possibilité de visualiser les résultats des scans précédant, la gestion d'accès et aussi la gestion des identifiants, et pour atteindre ses objectifs le système doit être doté d'une base de données qui englobe le tout.

Notre base de données est composée de deux parties essentiels :

- Une pour la gestion d'accès qui permet d'autoriser qui à faire quoi et quand via le modèle RBAC.
- L'autre partie est dédié à la gestion des machines, et des résultats des scans

La figure suivante (13) représente le diagramme de classes

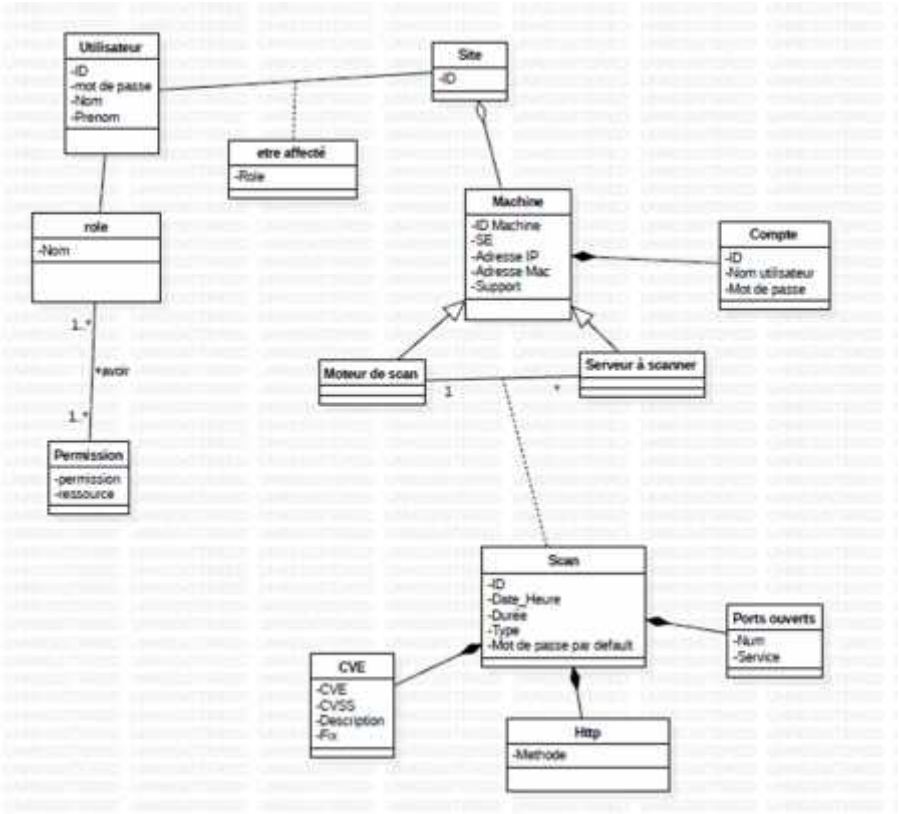

Figure 13: Diagramme de classe

Ci-dessous on va décrire les classes apparus dans le diagramme de classe.

Commenté [H7]: Explique pourquoi y a une nécessité de la base de données dans ton outil (pas produit mais outil).

Chaque le mot produit par outil ou système dans toute la conception !!!!

Commenté [H8]: Manque la description du diagramme. Il faudra expliquer chaque classe, association et classe associative, dan un tableau ou tu descrit chaque attribut sont type sa taille, et sa description…

- La classe Utilisateur : cette classe représente les utilisateurs de notre système, chaque utilisateur est caractérisés par un identifiant (id), un nom et un prénom qui forme son nom d'utilisateur et un mot de passe, chaque utilisateur a un ou plusieurs rôles, et il est affecté à un site s'il a le rôle d'administrateur de site.
- La classe rôle : cette classe représente les trois rôles dans notre système qui sont l'administrateur principale, l'administrateur de site et l'administrateur de système, chaque rôle est identifié par son nom, et il a ses propres permissions.
- La classe permission : cette classe représente les permissions de chaque rôle dans notre système, chaque permission est caractérisée par un nom et une liste des accès permis.
- La classe site : elle représente les sites des administrateurs de site, chaque site est caractérisé par un id, et il est responsable de plusieurs machines.
- La classe associative être affecté : cette classe représente la liaison entre chaque administrateur de site et les sites qu'il les dirige.
- La classe machine : elle représente toutes les machines, les serveurs à scanner et les machines qui contiennent les moteurs de scan, ces machines sont caractérisées par un identifiant (id machine), un nom du système d'exploitation (SE), une adresse IP et MAC et l'état (soit virtuel ou physique) (support).
- La classe hérité moteur de scan : cette classe représente la machine qui contient le moteur de scan, elle hérite toutes les caractéristiques de la classe machine.
- La classe hérité serveur à scanner : elle représente les serveurs à scanner par notre système, elle hérite toutes les caractéristiques de la classe machine.
- La classe compte : cette classe contient les identifiants des machines (nom d'utilisateur et mot de passe), et référenciée par un identificateur auto-incrémentale.
- La classe scan : cette classe représente les scans effectués par machine, elle est caractérisée par un identificateur (id), la date et l'heure et la durée du scan, le type de scan et le mot de passe par défaut s'il est trouvé.
- La classe CVE : elle représente tous les CVE trouvés dans un scan, cette classe est caractérisée par le numéro de CVE (CVE), le score (CVSS), la description et le fixe proposé.
- La classe http : elle contient les méthode http trouvés dans un scan sur une machine.

- La classe ports : cette classe représente chaque port ouvert trouvé dans un scan sur une machine, elle est caractérisée par le numéro de port et son service.

2.10 Déploiement de notre solution

La figure suivante (14) représente le diagramme de déploiement

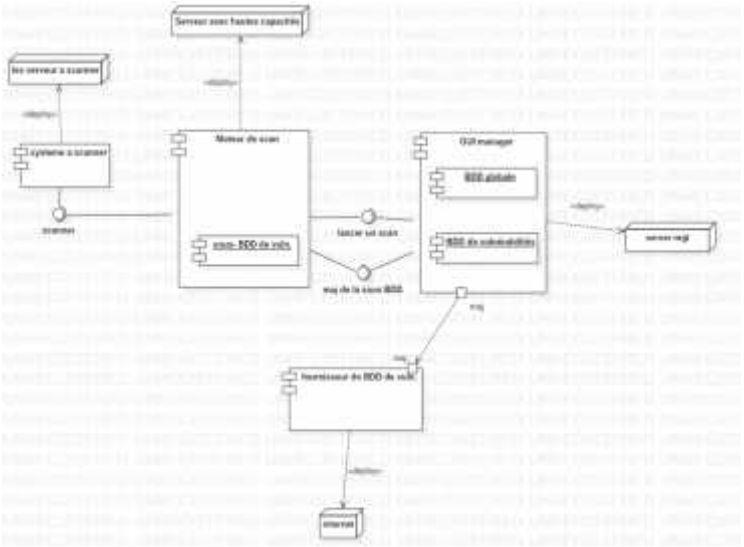

Figure 14 : Diagramme de déploiement

Description des nœuds de déploiement qui apparaissent dans la figure 14 :

a- Serveur MGT (de manager) : c'est la machine dont on installe la GUI manager, elle doit contenir un écran d'affichage, et car elle contient la BDD de vulnérabilités –qui devra être mis à jour de l'internet-, cette machine doit être performante et avec une connexion à l'internet

b- Serveur avec haute capacités : ces serveurs sont chargés d'exécuter des scans sur plusieurs machines en parallèle, reçoit les dernières versions de la BDD de vulnérabilités par la GUI manager et les listes des paquets de chaque machine avant de la scanner et envoie les résultats des scans a la GUI manager. Alors pour minimiser le temps d'exécution de ces processus, la machine dont on installe un moteur de scan doit être puissante

c- Les serveurs à scanner : ce sont les serveurs de l'entreprise.

2.11 Conclusion

Dans ce chapitre, on a identifié les diagrammes de cas d'utilisation, de séquence, de classe, de composant et de déploiement, pour obtenir une bonne vision, conclure une meilleure structure pour faciliter la réalisation de ce prototype.

Dans le chapitre suivant nous montrerons toutes les étapes, en détails, que nous avons suivies pour implémenter et réaliser notre système de gestion de vulnérabilités

CHAPITRE III : REALISATION

3.1 Introduction

Dans ce chapitre, nous allons décrire, étape par étape, la réalisation de notre produit qui répond aux besoins cités précédemment, et qui respecte la conception du chapitre précèdent, nous allons aussi montrer des solutions de sécurité pour assurer l'intégrité, la confidentialité, la disponibilité, et la non-répudiation.

Dans le chapitre précédent, nous avons divisé notre système en deux entités : le moteur de scan et GUI Manager, nous allons suivre la même division dans la représentation de notre réalisation de ce système.

3.2 Choix du SGBD

Comme on a évoqué précédemment la BDD est téléchargée sous forme d'un fichier json, ce type de fichier n'a pas d'index, et ça rend la recherche très couteuse et lente, pour cette raison en a décidé de convertir le fichier json en BDD relationnelle pour profiter de son index, malgré cette solution avait un inconvénient est que La BDD relationnelle est volumineuse par rapport le fichier json, ça pose un problème dans le transfert de cette BDD dans le réseau (le temps et la bonde passante), mais ce problème n'est pas aussi grave car la mise à jour est faite une fois par semaine ou plus.

3.3 Choix du langage de programmation

Notre système sera placé sur un grand réseau interne d'Elit, il touche la partie système et utilise des commandes, il transmit aussi des données critique sur le réseau, pour cela on a choisi le langage PYTHON pour la programmation back end et le HTML pour le front end, ce choix assure la rapidité de l'exécution, la facilité de l'écriture et la compréhension et l'intégrité car le python et un langage de haut niveau et aussi plus proche de la machine par rapport d'autre langage comme JAVA, il est aussi multiplateforme et possède de très riches bibliothèques, quant au choix de l' HTML, c'est parce que c'est une application web et c'est le mieux adapté, il suffit d'avoir un navigateur web, et il nous permet aussi d'utiliser le protocole HTTPS pour sécuriser le transfert de données.

3.4 Architecture composante de notre outil

La figure suivante (15) représente le diagramme de composants de notre solution

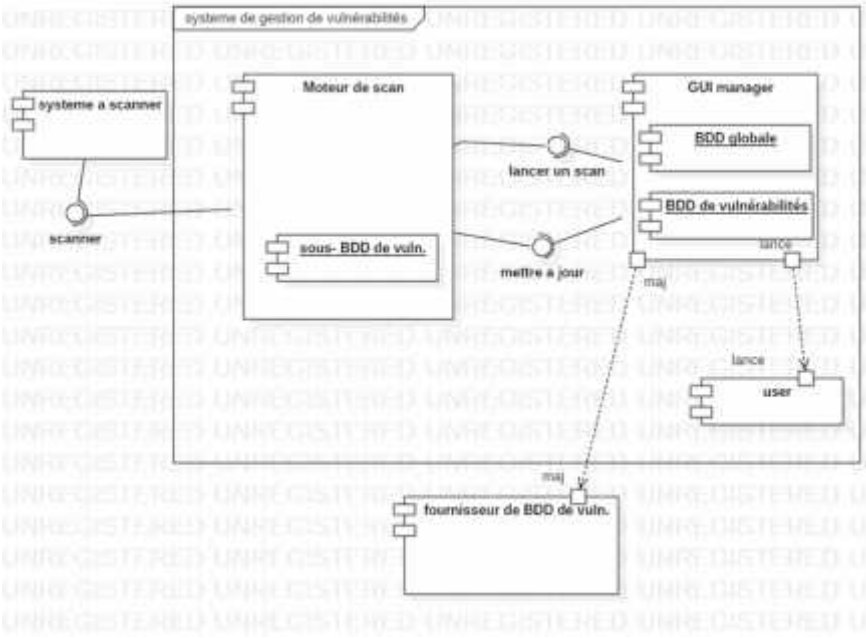

Figure 15: Diagramme de composants

Description des composants qui apparaissent dans la figure 15 :

a- Moteur de scan

Le moteur de scan est une petite application sans interface graphique, installé sur un hôte dans un sous réseau selon la politique de l'entreprise et manipulé par GUI manager, il est chargé d'effectuer des scans sur un plage d'hôtes, ce composant contient une base de données qui contiens juste les vulnérabilités des systèmes d'exploitation utilisés dans la plage d'hôtes du même moteur.

b- BDD de vulnérabilités

La base de données de vulnérabilités est une base de données relationnelle qui contient une liste des vulnérabilités, les vulnérabilités se présent dans notre BDD sous forme d'une table

Commenté [H9]: Il manque aussi une phrase introductive. Il faudra faire attention à mes remarques, je la mets une seule fois mais elle doit être appliquée partout.
1)Pas de titres qui se succèdent sans une phrase entre eux
2)Pas de figure/ tableau plaqué sans avoir un texte avant qui définie à quoi ça sert ceci.
3)Tu écrit toujours figure suivante, tableau suivant , il faudra rajouté entre parenthèse (figure 5) par exemple (Tableau 3) c'est très important de référencer tes figures/tableaux dans le texte.
4)Les mêms remarques à mettre avant diagramme de déploiement je vais pas écrire la même chose

Commenté [H10]: Je pense que ce titre avec autre choix conceptuel devront être dans le chapitre réalisation c'est mieux.
Le diagramme de déploiement est un passage entre les modules conception et réalisation, mais composant c'est dans la réalisation.
Aussi, la partie autre choi conceptuel je pense que c des choix technique il faudra les mettre dans la partie réalisation juste après ce diagramme et changer le titre.
Choix du SGBD
Choix du langage de programmation

contient le CVE, CVSS, la description et le fixe, cette table est associée à une autre table qui contient les paquets vulnérables.

c- GUI manager

Une interface graphique dédiée à manipuler les moteurs de scan par les différents utilisateurs du système, dont chaque utilisateur est affecté à un site, et chaque cite peut lancer plusieurs moteurs de scan selon la politique de l'entreprise appliquer par l'administrateur principale. Elle est chargée aussi de collecter, stocker et afficher les informations des scans, ainsi des statistiques.

d- User

Ce composant représente les ordres donnés par un utilisateur sur un site pour lancer les scans, come apparait dans la figure 2, les activités de GUI manager dépend de ces ordres.

e- Système à scanner connecté en SSH

Ce composant représente toutes les machines à scanner connectées dans notre réseau et reconnues par les moteurs de scan, le scan effectué dans ces machines à deux types :

Scan sans authentification : dont le moteur de scan n'a pas le login SSH de cette machine, alors ce scan touche juste les ports, les services actifs et les ciphers.

Scan avec authentification : en plus des taches qui peuvent être faites dans le premier type, dans ce type de scan, le moteur de scan peut effectuer un scan sur toutes les paquets installés dans le système d'exploitation de la machine cible en utilisant la connexion SSH.

3.5 Notre GUI Manager

GUI Manager est l'entité qui gère les scans, on parle ici des moteurs du scan, rapports, machines et identification, GUI Manager gère aussi les utilisateurs en utilisant des sites définis par l'administrateur principale.

Pour éclairer les choses on va expliquer les notions suivantes :

- Gestion des identifiants :

  Parmi les contraintes imposées par l'entreprise, les administrateurs des sites n'ont pas à connaitre les identifiants des machines à scanner, et ces identifiants doivent être bien sécurisés, pour ces raisons-là, on a ajouté un gestionnaire des identifiants, son rôle est

d'envoyer, au début de chaque scan, les identifiants des machines concernées (nom d'utilisateur, mot de passe) cryptés au moteur de scan qui effectuera le scan.

Avec cette solution, personne ne pourra jamais connaitre le login et le mot de passe d'une machine sauf l'administrateur système.

- Les sites :

Pour faciliter l'affectations des rôles des administrateurs qui lancent les scans, on a décidé d'introduire la notion de site.

Un site peut contenir plusieurs machines et n'a qu'un seul administrateur, une machine ne peut appartenir qu'a un seul site, et chaque site possède un moteur de scan.

La réalisation de GUI manager se base sur 2 parties :

3.5.1 La création des bases de données :

Nous avons 3 bases de données séparées dans notre système :

3.5.1.1 BDD des CVE :

Cette base de données contient les nouvelles vulnérabilités trouvées, elle est remplie en lisant un fichier JSON téléchargé de la source suivante :

https://nvd.nist.gov/feeds/xml/cve/misc/nvd-rss.xml

Les mis à jour de cette BDD est programmé automatiquement chaque semaine, et le fichier JSON de mise à jour est de la même source.

Pour plus de sécurité, on a décidé qu'il a qu'un serveur qui contient cette base et qui est chargé de faire les mises à jour de l'internet, cette solution minimise l'accès au réseau internet à partir du réseau interne de l'entreprise, après ça le serveur envoie les mises à jour aux moteur de scan.

3.5.1.2 BDD des mots de passes par défaut :

Ceci est un fichier contenant des centaines de mots de passes qu'on a tendance a utilisé, tel que ' admin ', '0000','root' …etc.

3.5.1.3 BDD principale :

Cette base contient les comptes des utilisateurs, les rapports des scans, la liste des moteurs de scans et des machines à scanner, pour une meilleure sécurité en a utilisé le modèle d'habilitation RBAC.

3.5.1.3.1 RBAC (Role-Based Access Control)

C’est un modèle de contrôle d’accès qui gère la sécurité des données à un niveau qui correspond étroitement à la structure de l'organisation, Chaque utilisateur se voit attribuer un ou plusieurs rôles, et chaque rôle se voit attribuer un ou plusieurs privilèges autorisés aux utilisateurs de ce rôle. L'administration de la sécurité avec RBAC consiste à déterminer les opérations qui doivent être exécutées par des personnes dans des emplois particuliers et à affecter les employés aux rôles appropriés. (15)

3.5.1.3.2 Projection du modèle RBAC dans notre système :

La BDD principale contient des données sensibles tel que les rapports, les comptes des administrateurs, les identifiants et les informations sur les machines, ces données ne sont pas accessibles à tous les administrateurs, pour cela on a attribué à chaque rôle ses permissions et ses ressources.

La figure suivante (16) représente la modélisation RBAC de notre système

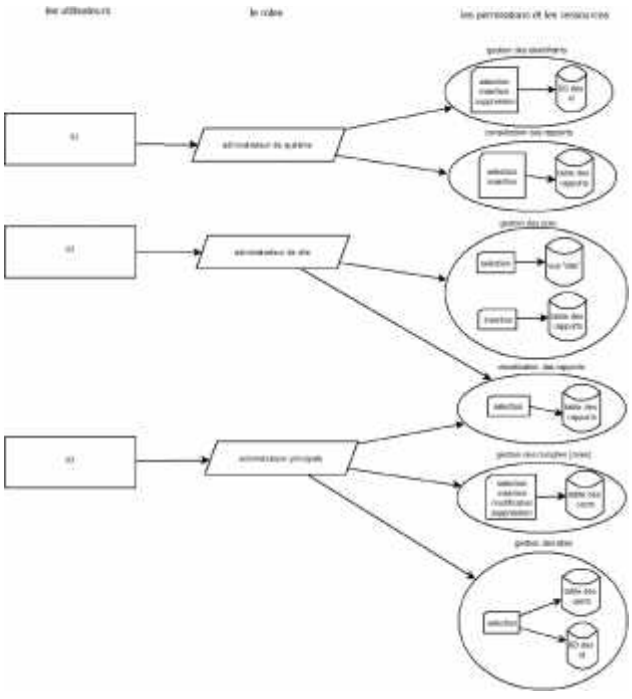

Figure 16 : la modélisation RBAC de notre système

3.6 La création des interfaces utilisateur

On va présenter ci-dessous les interfaces graphiques créer pour chaque type d'administrateur :

3.6.1 Interfaces graphique de l'administrateur de site :

Dès que l'administrateur de site s'authentifie, l'interface principale lui apparait, elle contient le lien de déconnexion plus deux boutons, un pour effectuer des scans et l'autre pour afficher les statistiques, le bouton de scan permet de passer vers l'interface de scan, et l'autre affiche une interface qui contient le tableau de bord, ci-dessous une capture d'écran de l'interface principale de l'administrateur de site :

Figure 17: l'interface principale de l'administrateur de site

Si l'administrateur de site décide de lancer un nouveau scan en appuyant sur le bouton de scan, le système lui affiche une interface de scan, cette dernière contient les sites associés à cet administrateur sous forme des zones de texte contiennent les adresses IP des machines du site, en plus cette interface permet à l'utilisateur de saisir la périodicité du scan, et de choisir le type de scan, soit approfondi (avec authentification), soit rapide (sans authentification).
Cette interface est la suivante :

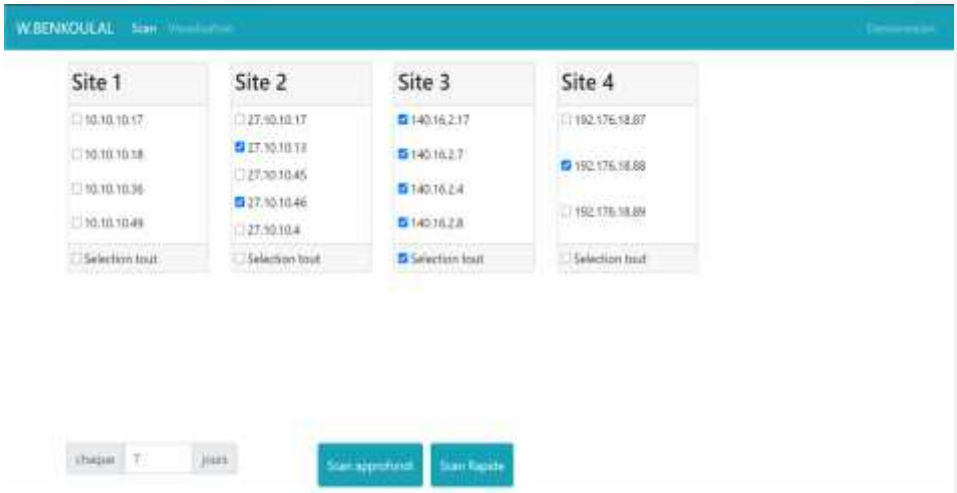

Figure 18: l'interface de lancement des scans de l'administrateur de site

Après chaque scan le système affiche les résultats du scan comme suite :

Figure 19 : interface qui représente le résultat de scan lancé

Si l'administrateur de site veut avoir les statistiques de tous les scans, il appuie sur le bouton des statistiques, le système lui retourne un tableau de bord détaillé, les deux figures suivantes représentent les données affichées par ce tableau de bord :

Figure 20: le tableau de bord des scans (1)

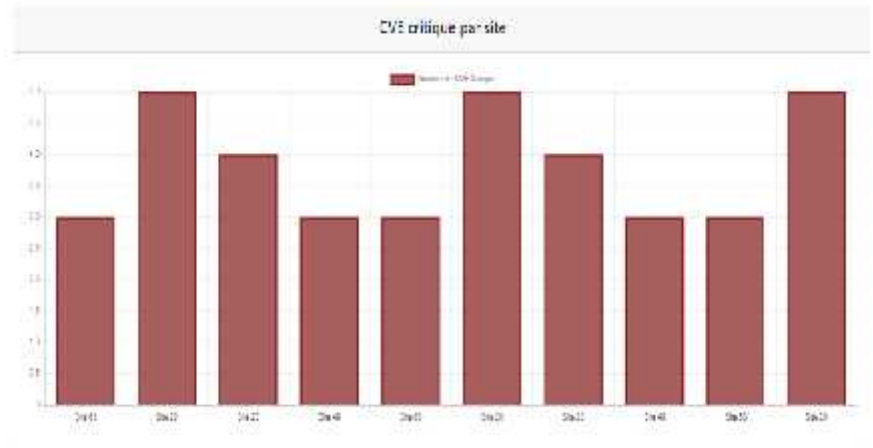

Figure 21: le tableau de bord des scans (2)

3.6.2 Interface graphique de l'administrateur principale :

L'administrateur principale a le rôle de créer et gérer les comptes d'administrateurs, il peut consulter le tableau de bord générale des scans.

S'il choisit la gestion des utilisateurs, un tableau qui contient la liste des utilisateurs lui apparait, il peut effectuer toutes les opérations de gestion sauf la suppression

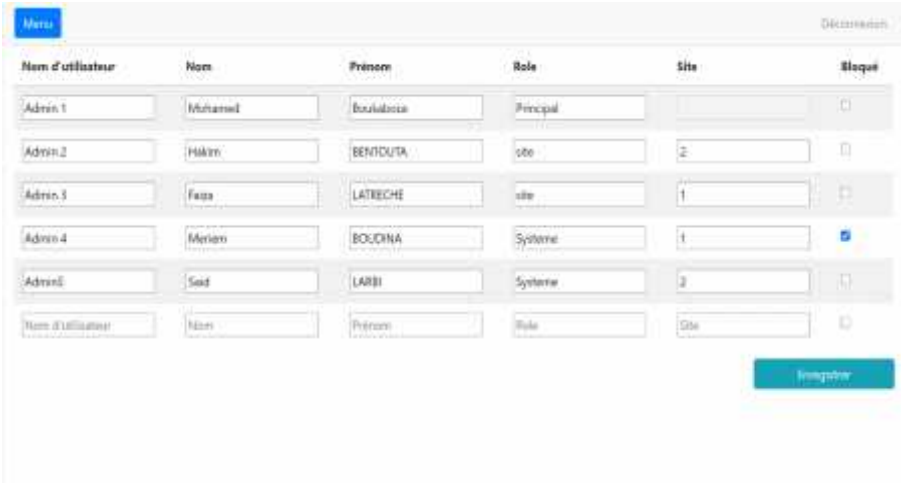

Figure 22: l'interface de gestion des utilisateurs

S'il choisit la consultation de tableau de bord, l'interface suivante lui apparait, elle contient toutes les informations et les statistiques des scans effectuer, et elle lui permet d'effectuer des différentes recherches.

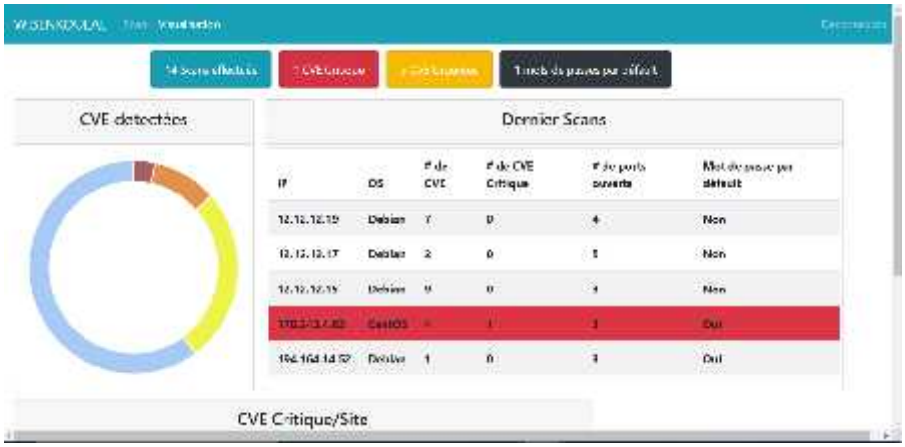

Figure 23 : le tableau de bord de l'administrateur principale

3.6.3 Interface graphique de l’administrateur système :

L’administrateur système est chargé de la gestion des identifiants, il est besoin alors des interfaces qui lui permet de créer et modifier des champs de la table des identifiants (la suppression n’est pas permise), et aussi de consulter les rapports du scan, modifier l’état d’une vulnérabilité existante de « non-fixé » à « fixé » dès qu’il l’a réparée.

Les interface apparues ci-dessous recouvrent ses besoins

L’interface de la gestion des machines :

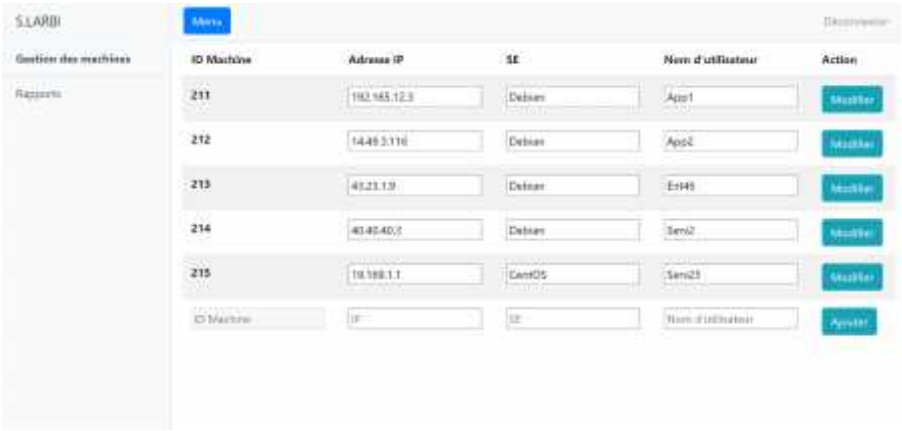

Figure 24: l’interface de gestion des machines

L’interface de l’ajout d’une machine :

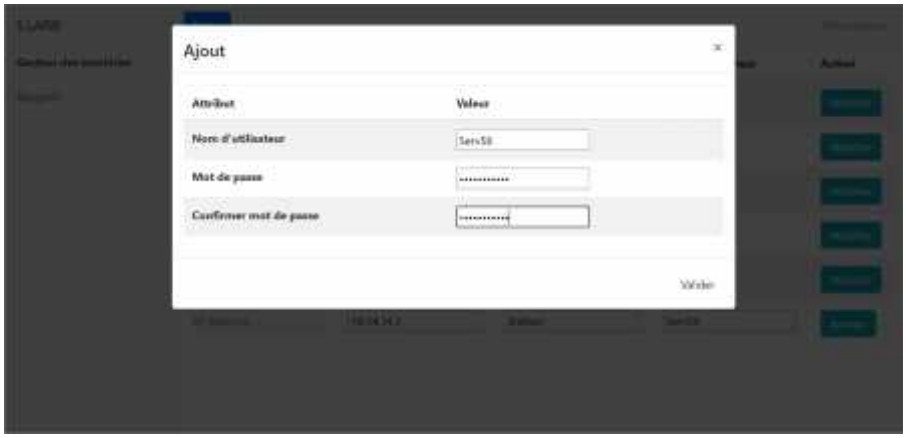

Figure 25: l’interface de l’ajout des machines

L’interface de modification d’une machine :

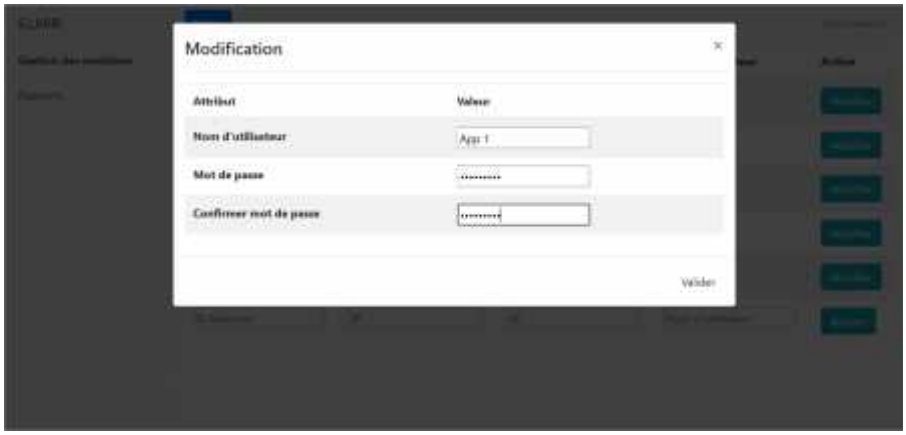

Figure 26: l’interface de modification des machines

L’interface de la consultation des rapports :

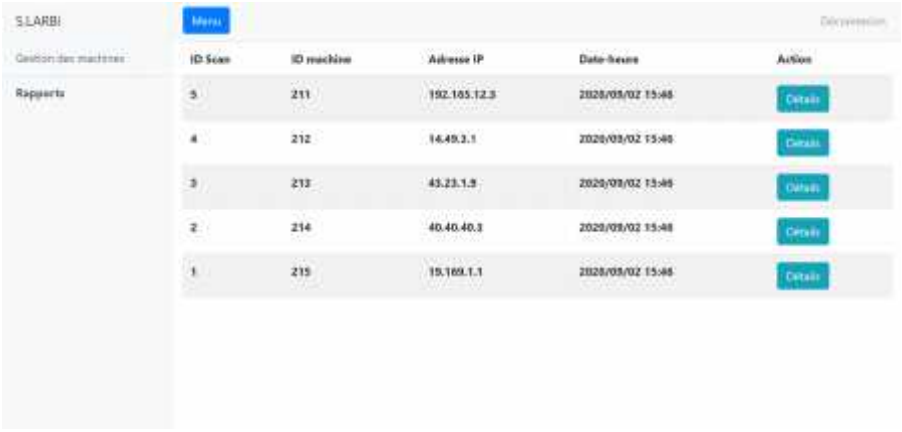

Figure 27: l'interface de consultation des rapports

Si l'administrateur système veut plus de détails sur un scan l'interface ci-dessous lui apparait :

Figure 28: l'interface de consultation des rapports (détails d'un scan)

S'il veut plus d'informations sur un CVE, cette interface lui apparait :

Figure 29: l'interface de consultation des rapports (détails d'un CVE)

3.7 Conclusion :

Comme nous avons dit au début, nous avons montré la réalisation de notre prototype, dans les deux partie, la gestion des données et l'implémentation des interfaces graphiques, on a aussi présenté quelques trucs qui augmentent le niveau de sécurité de notre gestionnaire des vulnérabilités

Dans le chapitre qui suit, nous allons tester notre système afin de montrer la fiabilité des résultats trouvés, et la sécurité des informations des machines et des utilisateurs.

CHAPITRE IV : TEST ET EXTENSION

4.1 Introduction

Après chaque réalisation d'un logiciel informatique, la dernière étape avant la commercialisation est le test, le produit sera prêt dès qu'il passe les tests.

Les tests de qualité de logiciel assurent que le produit répond aux besoins prédéfinis, ces tests-là suivent le programme tout au long de son cycle de développement. Il y a des tests a appliqués à chaque partie du programme dès qu'elle soit finie, et des autres après la réalisation de toute l'application.

Les tests de sécurité assurent que le programme est immunisé contre les attaques fréquents, dans notre cas on va prendre l'environnement de travail (le réseau de l'entreprise « ELIT » qui est très sécurisé) en considération.

4.2 Le test de logiciel

Le test du logiciel est une approche dynamique de la vérification, destinée à s'assurer que ce logiciel possède effectivement les caractéristiques requises pour son contexte d'utilisation. La première action à entreprendre est donc de décrire avec précision ce contexte, en particulier les fonctionnalités attendues, les contraintes d'environnement, ou encore les situations dangereuses à considérer. (16)

Le cycle de déploiement en V, qui apparait dans la figure ci-dessous (30), définie trois types de test qui assure le bon fonctionnement de logiciel.

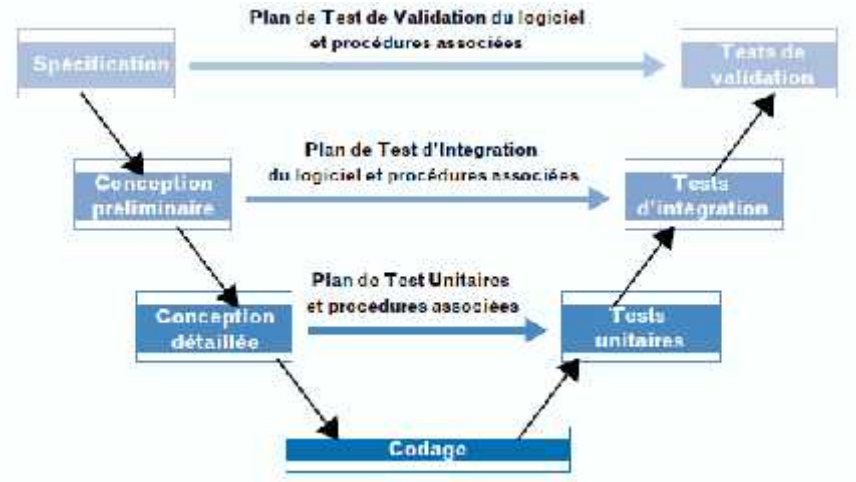

Figure 30: définition des tests dans le modèle de déploiement en V

Les tests unitaires : ces tests démontrent que chaque module effectue toute la fonction prévue et seulement cette fonction.

Tests d'intégration du logiciel : ils démontrent le bon fonctionnement d'unités fonctionnelles constituées d'un assemblage de modules.

Tests de validation : ils assurent que le logiciel implanté dans le matériel répond aux spécifications fonctionnelles.

Dans notre cas, on va appliquer les trois types de test afin de démontrer la qualité de notre motif. (16)

4.3 Définition des tests à appliquer :

Pour chaque type de tests on va décider la bonne technique à suivre (boite blanche ou boite noire), les variables d'entré et les résultats estimés.

4.3.1 Les tests unitaires

4.3.1.1 Définition des modules à tester

Comme il apparait dans les deux chapitres (conception et réalisation) notre système a deux modules principaux à tester, le module de scan et le module de gestion des résultats.

4.3.1.1.1 Le module de scan :

Ce module contient 4 types de scan, scan des CVE, des ports, des méthodes http et des mots de passe par défaut.

On va tester chaque une de ces 4 parties par la technique de test boite blanche, on doit alors définie les variables en entré et les résultats estimés.

4.3.1.1.1.1 Test de module de scan des CVE :

Ce test nous permettra de s'assurer du bon fonctionnement du module de détection de CVE, et qu'il donne des résultats fiables

Définition des entrés :

- Un serveur linux « Debian 8 » contenant des milliers de paquets.
- Une liste Contenant 5 paquets vulnérable -qui joue le rôle d'un fichier source- dont 2 sont installé dans notre serveur linux, un troisième qui est installé mais sur une autre version -non vulnérable-, un quatrième qui est installé mais vulnérable pour la distribution linux « CentOS 6 » et non pas « Debian 8 », quant au cinquième, il n'est tout simplement pas installé.

Résultats prédit :

Ci- dessous le résultat prédit, qui correspond aux deux paquets vulnérable installé :

Figure 31 : CVE prédits à être détecter par le module de scan

Résultats réels :

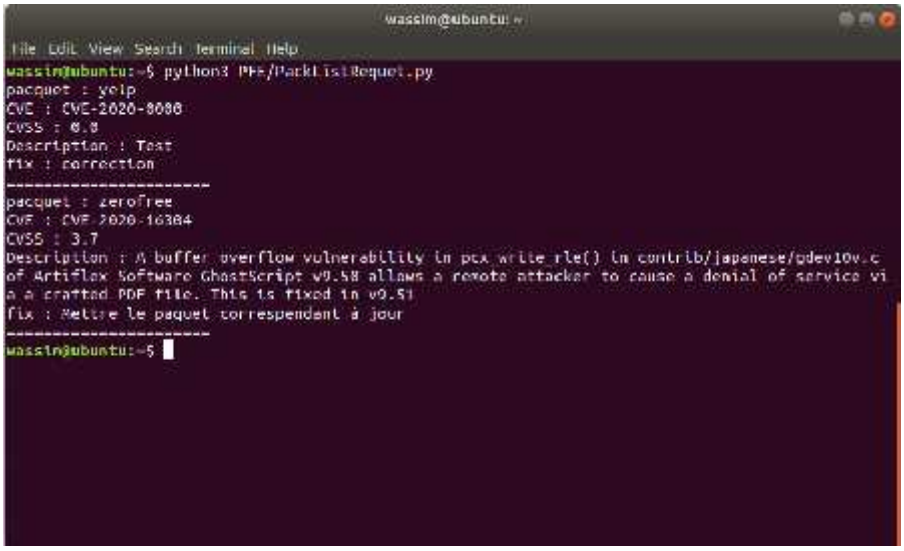

Figure 32 : CVE trouvés

4.3.1.1.1.2 Test de module de scan des ports :

Ce test nous permet de s'assurer du bon fonctionnement du module de scan des ports ouverts

Définition des entrés :

- Un serveur linux « Ubuntu 8.2 » avec seulement le service apache2 activé

Résultats prédits :

Afficher le port 80 comme étant ouvert.

Résultats réel :

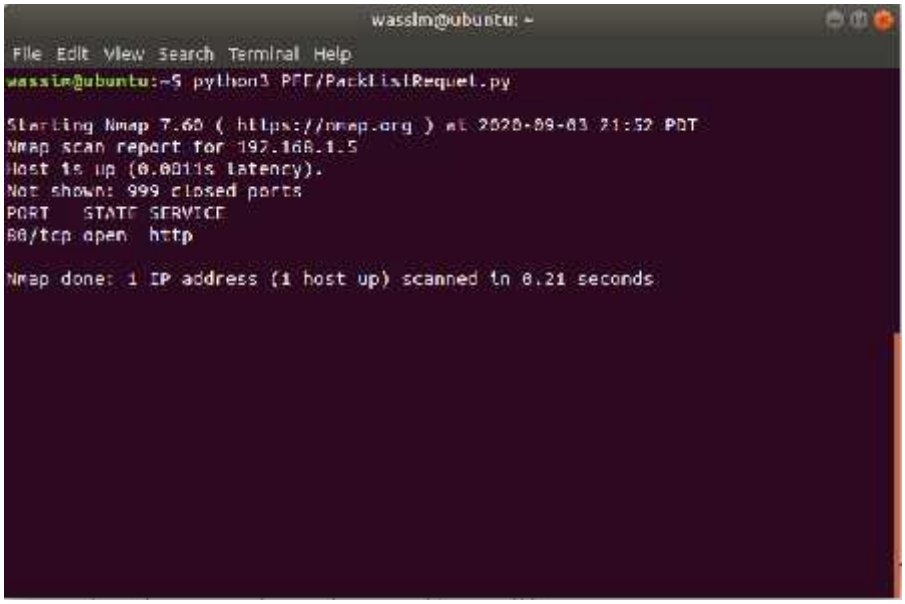

Figure 33 : les ports trouvés après le test

4.3.1.1.1.3 Test de module de scan des méthodes http :

Ce teste nous permettra de s'assurer du bon fonctionnement du module qui scan les méthodes http autorisés, sachant que c'est aux administrateurs de de définir si une méthode devrait être autorisé ou pas, selon la ou les applications installées sur le serveur.

Définition des entrés :

- Un serveur linux « Debian 8 » sur lequel le service Apache2 est installé.

Résultats prédits :

Par défaut le serveur Apache2 autorise 4 méthodes http qui sont :

- POST
- GET
- HEAD
- Options

Résultats réel :

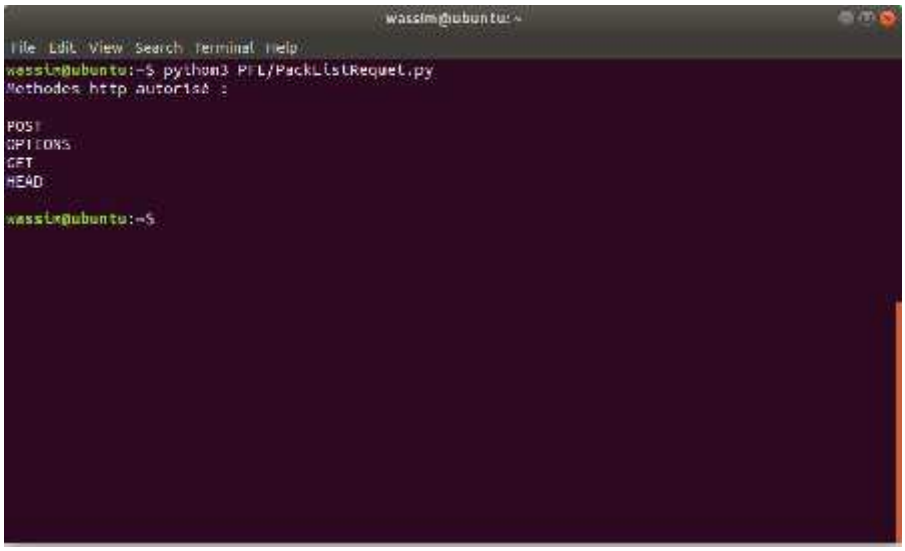

Figure 34 : les méthodes http permises

4.3.1.1.1.4 Test de module de scan des mots de passe par défaut :

Ce test nous permettra de s'assurer du bon fonctionnement du module qui détecte les mots de passes par défaut.

Avant d'établir la connexion SSH avec un serveur, ce module vérifie l'existence du mot de passe de l'utilisateur de ce serveur dans un fichier source contenant des milliers de mot de passes par défaut.

Définition des entrés :

- Un serveur linux « Debian 8 » contenant un utilisateur qui a « admin » comme mot de passe.
- Un fichier textuel contenant une liste des mots de passes par défaut -joue le rôle d'un fichier source-

Résultats prédits :

Signalement de l'existence d'un mot de passe par défaut, sans le divulguer, et sans interrompre le scan.

Résultats réel :

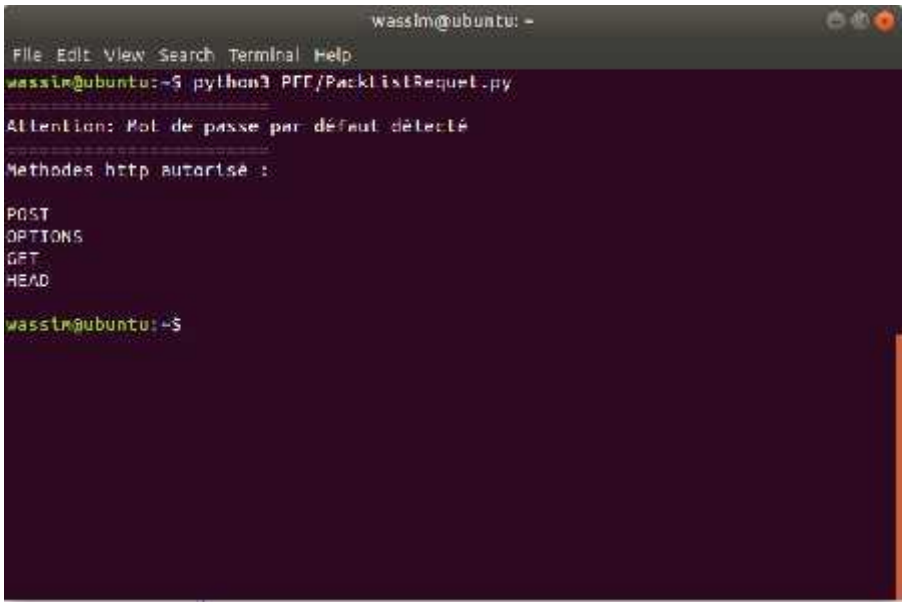

Figure 35 : la détection des mots de passe par défaut

4.3.1.1.2 Le module de gestion des résultats :

Ce module est chargé de gérer les résultats (stockage, traitement, affichage), alors en doit tester ces trois opérations de la gestion, pour cela on va lancer un test à partir de l'interface de l'administrateur de site dans une seule machine, après en va suivre les résultats dans les trois opérations, Les entrés de ce test dans la figure ci-dessous :

Figure 36 : les entrés de test du module de gestion des résultats

4.3.1.1.2.1 Test de stockage :

Pour la vérification du bon stockage, l'administrateur du site 1 va lancer un scan, et on vérifiera sur la base de données que les données du scan ont été enregistré avec succès.

Et voici ce qu'on trouve sur la base de données :

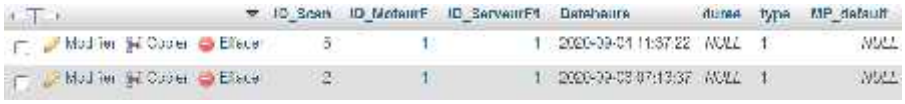

Figure 37 : résultat du stockage

4.3.1.1.2.2 Test de traitement et affichage :

Dans ce test on s'assure que les données stockés sont ceux affichés, pour cela on vérifie que ce qu'il est afficher sur l'interface du résultat du scan-05 est ce qui est enregistré sur la base de données.

Voici ce qui est stocké en base de données :

ID_Serveur	Adresse_IP	Adresse_Mac	SE	Support
1	192.168.1.4	B4-69-9F-CA-38-AC	Debian	VM

Figure 38 : les données stockées de scan05 (1)

Figure 39

ID_Serveur	Methode
1	GET
1	post
1	HEAD
1	Options

Figure 40: les données stockées de scan05 (2)

ID_Serveur	Num_Port	service
1	80	HTTP
1	443	SSH

Figure 41: les données stockées de scan05 (3)

Et voici ce qui est affiché sur l'interface graphique :

Figure 42 : les données traitées et affichées du scan05

Et quand on appuie sur le bouton « Plus » :

Figure 43: les détails des CVE du scan05

On constate donc que ce module fonctionne correctement.

4.3.2 Les tests de validation :

Les tests de validation vérifient la compatibilité du système avec le matériel, le fonctionnement des interfaces et le temps d'exécution. Dans notre cas on s'intéresse au temps d'exécution.

Voici les performances de la machine sur laquelle on effectue les tests :

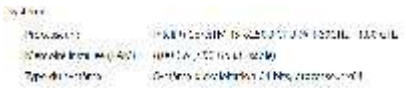

Figure 44: les performances de la machine qui lance les trois VM (manager, moteur de scan et la machine cible

4.3.2.1 Le temps d'exécution du code de scan :

Dans ce test, on lance un scan approfondi à partir du moteur de scan et on reporte le temps d'exécution :

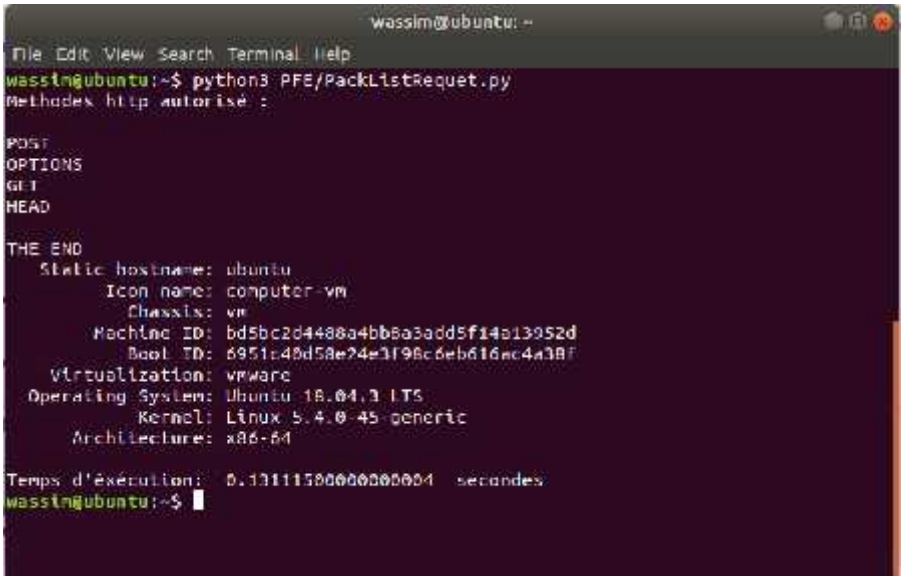

Figure 45: le temps d'exécution

4.3.2.2 Le temps de réponse total du scan :

On a ajouté un compteur qui se déclenche dès que le scan se lance à partir de l'interface de l'administrateur di site, et s'arrête au moment que le résultat du scan soit affiché. On a modifié l'interface, afin qu'elle affiche ce temps de réponse total :

Figure 46: le temps de l'exécution total

4.3.3 Tests du contrôle d'accès :

Dans ces tests nous allons tester le bon fonctionnement du module de contrôle d'accès. Pour effectuer les tests on aura besoin d'accéder à la base de données avec les 3 types d'administrateur et s'assurer que chacun d'eux n'a que les privilèges qui lui sont attribué selon la politique de « ELIT »

4.3.3.1 Administrateur principale :

Cet administrateur a tous les privilèges sauf la modification des résultats du scan, et les identifiants.

Pour tester on va donc lancer une requête qui demandera de changer le type de scan (à titre d'exemple), voici la requête :

Figure 47: requête non-permise pour l'administrateur principale

.

Résultat prédit :

Le SGBD n'exécutera pas la requête car l'administrateur principale n'a le droit de modifier le type du scan.

Résultat réel :

Figure 48: le résultat de la requête de l'administrateur principale

4.3.3.2 L'administrateur du site :

Cet administrateur n'a accès qu'aux machines, et scans des sites qu'il administre, et il n'a pas le droit lui aussi de modifier les résultats des scans.

Pour cela on lui a créer une vue, qui ne contienne que les champs qu'il a droit à lire, ou modifier, ou insérer, à savoir les informations des scans et des machines des sites qu'il administre

Pour effectuer le test, on prend l'exemple suivant :

L'administrateur du site veut afficher les adresses IP de toute les machines, y compris celle qu'il n'administre pas, il doit donc exécuter la requête suivante :

Figure 49: requête non-permise pour l'administrateur de site

Résultat prédit :

Le SGBD n'exécutera pas la requête car l'admin du site n'a le droit d'accéder qu'à la vue qui a été créer pour lui

Résultat réel :

Figure 50: résultat de la requête non-permise

Par contre, s'il veut afficher les adresses IP des machines qu'il administre, il doit exécuter la requête suivante :

Figure 51: requête permise pour l'administrateur de site

Résultat prédit :

Le SGBD devra exécuter la requête et retourner la liste des adresses IP.

Résultat réel :

Figure 52: résultats de la requête

Après les tests effectuer, nous concluons que le modèle RBAC fonctionne très bien.

4.3.4 Les tests de sécurité :

Dans cette phase on va essayer de pénétrer le système avec quelques attaques les plus fréquentes, on va prendre en considération que l'entreprise « ELIT » a un IDS et IPS dans son réseau interne, et des antiviraux robustes installés dans chaque machine, pour ces raisons on va éliminer les attaques réseau et les attaques de saturation des machines comme « DOS » et « DDOS », et car le réseau « ELIT » est presque isolé de l'internet, les attaques de l'extérieur sont théoriquement exclus, il reste alors les attaque qui viennent des administrateurs ou du personnel de l'entreprise.

En plus du devoir de l'entreprise de sensibiliser ses travailleurs, les failles liées aux interfaces utilisateurs doivent être éliminées, comme les injections SQL et l'accès par la force brute.

Et pour couvrir ses failles on a utilisé quelques protocoles de sécurité, ci-dessous on va appliquer des tests de pénétration afin d'assurer le succès de ces protocoles.

4.3.4.1 Les injection SQL :

Description de test : dans l'interface d'authentification en va essayer d'injecter des requêtes SQL, comme apparu dans la figure suivante :

Figure 53: le test de l'injection SQL

Et voici un deuxieme test :

Figure 54: le test de l'injection SQL (2)

4.3.4.2 L’accès par la force brute :

Après 5 tentatives successives d’identification échoué le compte de cet utilisateur sera bloqué, et seul l’administrateur principal pourrait alors le débloquer

Le résultat est le suivant :

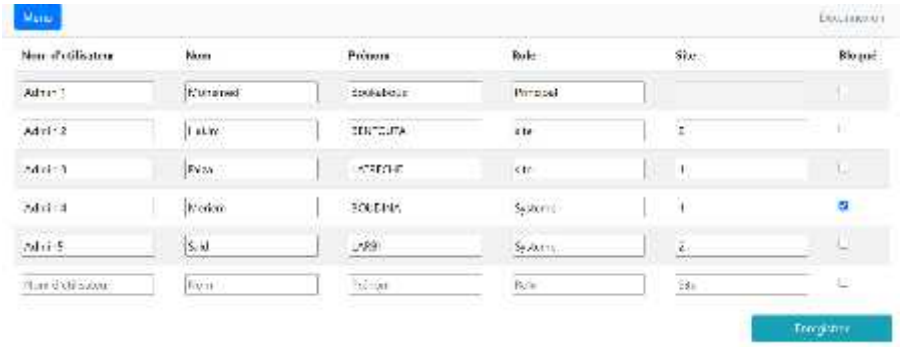

Figure 55: le résultat de l’accès par la force brute

4.4 Conclusion des tests effectué :

Le tableau suivant résume les résultats des tests effectués :

Le type de test	La partie tester		Nombre des machines utilisés dans le test	résultats
Test unitaire	Module de scan	Scan des CVE	1	Fonctionne correctement
		Scan des ports		Fonctionne correctement
		Scan des mots de passe par défaut		Fonctionne correctement
		Scan des méthodes http		Fonctionne correctement
	Module de gestion des données	Test de stockage	3	Fonctionne correctement
		Test de traitement et affichage		Fonctionne correctement
				Fonctionne correctement
Test d'intégrité	Le lancement du scan		3	Fonctionne correctement
	L'envoie des résultats		3	Fonctionne correctement
Test de validation	Temps de réponse d'exécution du code de scan par machine		1	
	Temps de réponse total de scan par machine		3	
Test de contrôle d'accès (RBAC)	Pour l'administrateur principale		1	Fonctionne correctement
	Pour l'administrateur de site			
Test de sécurité	Injection SQL		1	Protégé
	Accès par la force brute			Protégé

Tableau 6: conclusion des tests

CONCLUSION GENERALE :

Le système de gestion des vulnérabilités que nous avons réalisé se focalise principalement sur la détection des vulnérabilités des systèmes d'exploitation linux dans les machines d'une architecture réseau quelconque, il est vrai que nous nous sommes basés sur une conception compatible avec l'architecture et la politique de l'entreprise « ELIT », mais ce prototype reste fonctionnel dans d'autres architectures réseaux.

Nous avons proposé une conception complète qui représente une solution de sécurité pour l'entreprise « ELIT », elle remplace l'ancienne solution qui est très chère et suspecte, notre conception a couvert les besoins émergeants de la problématique, à travers le scan des machines sous plusieurs systèmes d'exploitation linux, jusqu'à la sécurité du gestionnaire lui-même, en passant par la gestion des résultats, des utilisateurs, des machines, des sites et des moteurs de scan.

Nous avons assuré aussi dans la partie de conception une source fiable des données des nouvelles CVE, en proposant une méthode pour l'atteindre qui respecte la politique et l'architecture réseau de l'entreprise.

Le prototype que nous avons réalisé prouve qu'il est apte à être utile, c'est la conclusion des tests effectués boite blanche et noire, fonctionnel et non-fonctionnel, ce prototype prouve aussi son immunité contre les attaques les plus fréquente.

Nous prévoyons dans un avenir proche de finaliser les points suivants :

- Le back-end de gestion des utilisateurs et des identifiants.
- La mise en place d'un gestionnaire des CVE qui assure la mise à jour quotidienne des CVE à partir de la source proposée et selon notre conception.
- Une extension qui permette de scanner les autres plateformes tel que (Windows et MacOs)

BIBLIOGRAPHIE :

(1) KARSPERSKY

(2) Source ITU X.1205

(3) https://www.securiteinfo.com/conseils/introsecu.shtml

(4) : ISO 27005

(5) : Institut national des normes et de la technologie (NIST) – Les états unis d'Amérique

(6) https://nvd.nist.gov/

(7) Vulnerability scanners (Author Johan Nilsson) Master of Science Thesis Stockholm May 2006

(8) https://searchsecurity.techtarget.com/feature/Tenable-Nessus-Vulnerability-Scanner-Product-overview

(10) https://www.rapid7.com/products/nexpose

(11) https://www.rapid7.com/info/introducing-insightvm

(12) https://www.codeflow.site/fr/article/how-to-use-vuls-as-a-vulnerability-scanner-on-ubuntu-18-04

(13) Vulnerability management tools for COTS software - A comparison – (S.M. Welberg Student MBI at University of Twente postbus 217 7500 AE Enschede)

(14)https://www.ibm.com/support/knowledgecenter/SS2TKN_9.0.0/com.ibm.tivoli.tem.doc_9.0/Security_and_Compliance/SCAP_Users_Guide

(15)https://csrc.nist.gov/publications/detail/conference-paper/1992/10/13/role-based-access-controls

(16) comment construire les tests d'un logiciel (P. Charpentier,2000).