Sujet N° A04-03

Les tests d'intrusion dans les réseaux Internet, l'outil Nessus

Dongé Laurent
laurent_donge@yahoo.fr

Président du jury : Mr. GRESSIER
CNAM Paris – Département informatique

Sommaire

Conventions

Les différentes typographies utilisées dans ce document sont les suivantes :

- une typographie ordinaire pour le texte,
- **Une mise en gras** pour les termes figurant dans le glossaire (Annexe A).

1. Introduction

Du fait de la démocratisation des moyens de connexion à l'Internet due à une pratique des prix de plus en plus attractifs par les différents fournisseurs d'accès, et d'une couverture géographique de plus en plus importante, le nombre d'internautes utilisant des connexions de type haut débit ne cesse de croître. Avec ces types de connexion, les internautes restent en ligne longtemps, ce qui les expose davantage à la convoitise de personnes mal intentionnées qui voient en eux des ressources à utiliser afin, par exemple, d'augmenter leur notoriété dans le monde des pirates. En effet, un pirate peut prendre le contrôle d'un tel poste afin d'attaquer une institution de l'Etat ou un acteur de l'Internet connu, tel qu'un portail ou un site de vente en ligne.

Les entreprises et les particuliers se voient donc confrontés de façon quotidienne à des vers, des virus, des attaques de tous types ou des tentatives d'intrusions. La sécurité est plus que jamais une problématique d'actualité et nous pouvons facilement le constater en parcourant les journaux de la presse spécialisée.

Un moyen rapide de connaître l'étendue de la fragilité de son environnement, vis à vis des attaques diverses et variées, est d'effectuer des tests d'intrusions qui permettent d'avoir une liste des failles de vulnérabilités potentielles.

L'une des étapes les plus importantes de la démarche utilisée dans les tests d'intrusion, est la récolte d'informations relatives aux systèmes et services présents sur les ordinateurs constituant le réseau. Ces informations permettent en effet, soit par recherche sur l'Internet, soit par l'emploi d'outils dédiés tels que les scanners de vulnérabilité, la détermination des failles de vulnérabilité.

Ce travail se propose de présenter les tests d'intrusion dans les réseaux Internet. En particulier nous présenterons l'outil **Nessus** qui permet de scanner des réseaux et de mettre en évidence un certain nombre de leurs failles de vulnérabilité.

Dans un premier temps, après une rapide présentation d'Internet en terme de topologie et de services proposés, nous aborderons le thème de la sécurité des réseaux informatiques qui devient un véritable enjeu pour les particuliers et les entreprises. Nous préciserons également la démarche généralement suivie lors des tests d'intrusion. Puis nous l'illustrerons en décrivant un certain nombre d'outils utilisés. Les outils d'audit seront ensuite plus particulièrement abordés, ainsi que la phase d'audit des réseaux dans les tests d'intrusion.

Une fois le contexte clairement posé, nous nous intéresserons plus spécialement au scanner **Nessus**. Que permet-il de faire ? Sur quelle architecture est il basé ? Comment le met-on on en place dans un réseau ? Quelles failles de vulnérabilité permet il de mettre à jour ? Comment sont restituées les informations récoltées et les failles de vulnérabilité découvertes ?

Suite à cette présentation de l'outil **Nessus**, nous l'utiliserons afin de faire des tests sur un réseau local **Ethernet**. La démarche suivie lors de ces tests sera également précisée.

Nous finirons par comparer **Nessus** avec différentes solutions du marché, gratuites ou commercialisées, afin de mieux le positionner par rapport à ce qui existe.

2. Les tests d'intrusion dans les réseaux Internet

2.1 Réseaux Internet

2.1.1 Présentation

Nous allons ici nous intéresser à deux aspects d'Internet. D'abord à l'Internet en tant qu'un ensemble de réseaux interconnectés, ensuite à ses nombreux services qui ne cessent d'évoluer.

2.1.1.1 Un ensemble de réseaux interconnectés

Les internautes se connectent pour des raisons diverses et variées. En effet, ils peuvent se connecter à partir de leur travail pour des raisons professionnelles ou à partir de chez eux pour des raisons personnelles ou ludiques.

Le public se connectant est très hétérogène. Les moyens de connexion sont multiples (**ADSL**, bas débit, mobile etc.) , ainsi que les systèmes d'exploitation utilisés (**Windows**, **Linux**, **Unix, Mac** etc.).

La figure suivante illustre la façon dont sont connectés des ordinateurs sur Internet.

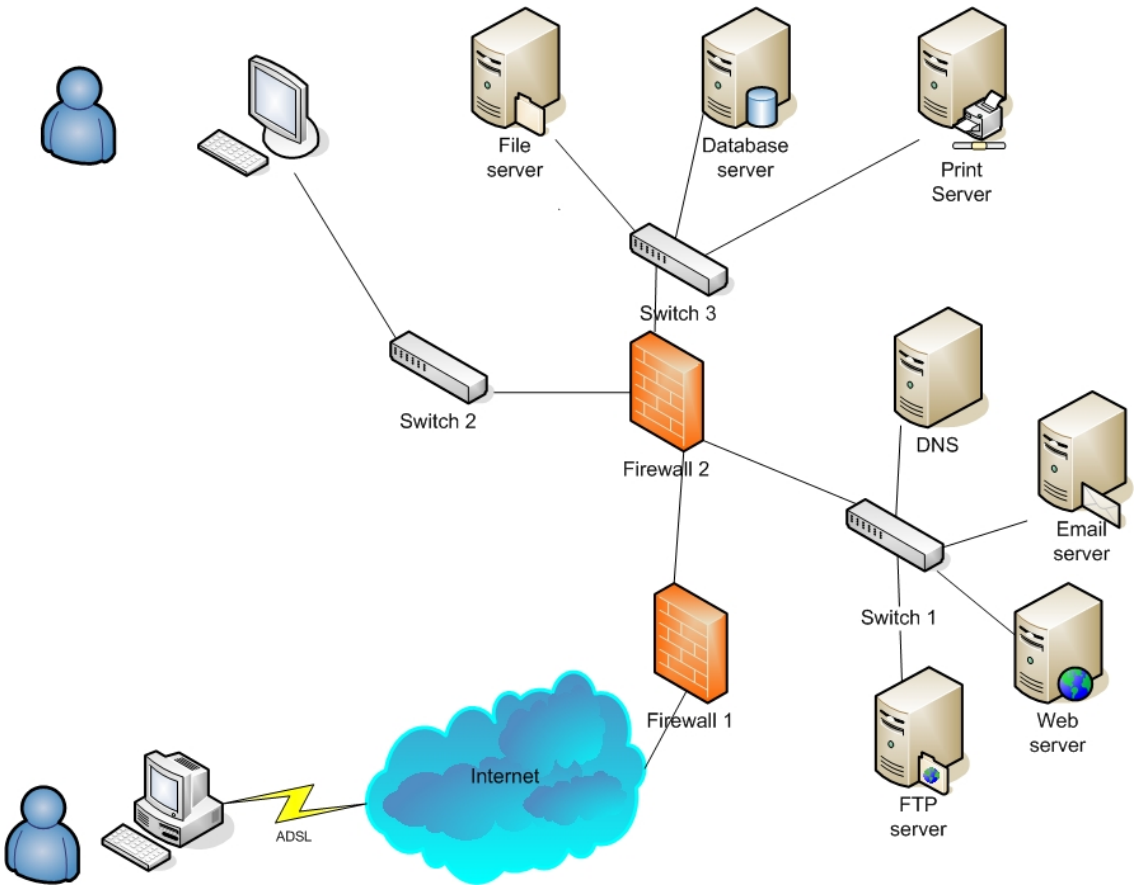


FIGURE 1 : INTERNET

2.1.1.2 De nombreux services proposés

Afin de mieux positionner les différents protocoles dont nous allons parler dans la suite, nous allons faire un bref rappel du modèle **OSI**. Le modèle **OSI** est un standard qui permet de faciliter l'étude des technologies impliquées dans les réseaux. Des protocoles définissent les divers services offerts par les différentes couches du modèle. En général, un utilisateur n'a connaissance que des services proposés par la couche d'application du modèle **OSI**, alors que, les **hackers et** les **crackers** possèdent davantage de connaissances sur les protocoles réseaux. En effet, l'intrusion dans un réseau nécessite une bonne compréhension de l'ensemble des couches, des plus hautes du modèle jusqu'à la couche réseau.

Le tableau ci dessous présente le modèle **OSI** *:*

<table>
<tr><th colspan="2">Modèle de référence</th><th colspan="2">Suite IP Internet Protocole</th><th></th></tr>
<tr><td>7</td><td>Application</td><td rowspan="3">FTP,Telnet SSH,SMTP http,NNTP</td><td>NFS</td><td rowspan="3">Protocoles de niveau application</td></tr>
<tr><td>6</td><td>Présentation</td><td>SMB</td></tr>
<tr><td>5</td><td>Session</td><td>RPC</td></tr>
<tr><td>4</td><td>Transport</td><td colspan="2">TCP,UDP</td><td rowspan="4">Protocoles de niveau réseau</td></tr>
<tr><td>3</td><td>Réseau</td><td colspan="2">IP (ICMP)</td></tr>
<tr><td>2</td><td>Liaison</td><td colspan="2">ARP</td></tr>
<tr><td>1</td><td>Physique</td><td colspan="2">Physique</td></tr>
</table>

Chaque fois qu'une machine a recours aux services proposés sur une autre machine, elle indique une destination en fournissant une adresse **Internet IP** et un protocole de transport tel que **UDP** ou **TCP** et spécifie un port qui correspond au service désiré. Un port est lié à une application. Lorsqu'une requête est faite sur ce port, l'application serveur correspondante répond au client qui a émis la requête. Les ports permettent donc à un ordinateur de faire plusieurs choses à la fois. Il est possible grâce à eux de faire transiter plusieurs types d'informations sur une même connexion.

Il existe de nombreux ports sur un serveur Internet moyen. La plupart d'entre eux sont inactifs. Un standard a été défini pour l'assignation des ports par l'**IANA**. La valeur d'un port est comprise entre 0 et 65535. Les ports se divisent en trois catégories. Les ports compris entre 1 à 1023 sont les «Well known » ports. Ces ports bien connus sont associés à un service de base (21 : **Telnet**, 80 : **HTTP**). Les ports compris entre 1024 et 49151 sont les registered ports qui ont été réservés pour un service donné (3306 : **MySQL**). Enfin, les ports compris entre 49152 et 65535 sont les ports dynamiques utilisés pour les **sessions**.

Le tableau suivant précise certains des ports les plus connus avec les applications qui leurs sont habituellement associées. Chaque port est associé à des protocoles de niveau application ou services visibles qu'un utilisateur peut utiliser directement.

Service ou application	**Port**
Hypertext Transfert Protocol (**HTTP**)	Port TCP 80
Domain Name System (**DNS**)	Port UDP et TCP 53
Telnet	Port TCP 23
File Transfert Protocol (**FTP**)	Port TCP 20 et 21
Simple Mail Transfer Protocol (**SMTP**)	Port TCP 25
Secure shell (**SSH**)	Port TCP 22
HTTP sur **SSL**/**TLS** (**HTTPS**)	Port TCP 443

Parmi les protocoles les plus connus, nous pouvons citer :

- ⇨ **HTTP**, c'est un protocole léger, donc rapide, qui sert à gérer l'information hypermédia. Il est utilisé pour présenter les données sur le **Web**.

- ⇨ **DNS** fournit le service de traduction des noms d'hôtes en **adresse IP**, et inversement. **DNS** traduit les adresses entre les couches réseaux et applications.

- ⇨ **FTP** est une méthode standard permettant de transférer des fichiers entre deux machines.

- ⇨ **SMTP** permet de transporter du courrier de façon fiable et efficace.

- ⇨ **ARP** fait correspondre les adresses Internet aux adresses physiques des machines.

L'utilitaire, **Telnet** permet à un utilisateur de se connecter à une machine distante et d'y effectuer des commandes.

Et l'interpréteur de commande **SSH** permet de se connecter sur un autre ordinateur d'un réseau, d'y exécuter des commandes à distance de la même façon qu'avec **Telnet.** A la différence près, qu'il propose des mécanismes d'authentification et une méthode de cryptage des communications.

2.1.2 La sécurité

2.1.2.1 Un enjeu

Le nombre de services disponibles sur l'Internet ne cesse de croître chaque jour, ainsi que le nombre de machines qui se connectent de façon permanente. Ceci a pour conséquence d'augmenter les risques liés au fait d'être présent sur Internet et de subir des attaques.

Les motivations des pirates ont diverses origines. Elles peuvent être liées à un goût du défi ou l'envie d'avoir plus de notoriété dans leur milieu. C'est ainsi que les organisations à grande visibilité comme les gouvernements ou les institutions financières sont de fréquentes cibles. L'appât du gain ou des avantages financiers relatifs à des activités comme le vol d'informations et l'espionnage industriel peuvent constituer une autre explication au désir de prendre contrôle de ressources informatiques que les pirates ne possèdent pas. De plus, la plupart du temps, les pirates se réfugient derrière un sentiment d'impunité et, dans la majorité des cas, ne se rendent pas réellement compte des risques qu'ils encourent.

Le piratage repose essentiellement sur les erreurs de conception des systèmes et sur des mauvais paramétrages lors des configurations de ces derniers, ainsi que sur des failles de sécurité présentes dans les différents services proposés. Nous assistons alors à la mise en place d'une compétition entre les pirates et les personnes en charge de la sécurisation des systèmes tels que les administrateurs réseaux ou les personnes en charge du développement des logiciels. Les premiers cherchent à exploiter par tous les moyens les trous de sécurité présents. Les seconds tentent de sécuriser le système d'information. La conséquence est que la sécurité est devenue un véritable enjeu pour les entreprises qui veulent protéger leur système d'information et leurs sites de commerce électronique . C'est aussi un enjeu, pour les particuliers, qui aimeraient bien utiliser leur connexion Internet en toute tranquillité.

2.1.2.2 Comment sécuriser un réseau ?

La problématique de sécurisation est liée à une démarche complexe qui revêt un caractère cyclique. En effet, l'évolution rapide des technologies et du parc informatique des entreprises fait que la question de la sécurité se pose de façon récurrente. Par exemple, l'apparition des réseaux sans fils (**WiFi**) a introduit de nouveaux types de vulnérabilités. Il en va de même lorsque nous ajoutons un nouveau poste dans un réseau d'entreprise. Si sa configuration n'est pas faite de façon correcte, ceci peut permettre des intrusions dans une partie du réseau.

La sécurisation d'un réseau n'est pas simple à réaliser. Le réseau est constitué d'un ensemble de systèmes hétérogènes. De nombreux services, qui ne cessent d'évoluer, sont disponibles. Les personnes en charge de la sécurité telles que les administrateurs réseau, ont à leur disposition toute une panoplie d'outils :

- ⇨ Des logiciels spécialisés dans la protection tels que les firewalls dont le rôle est de filtrer les paquets circulant entre le réseau et l'Internet, ou les logiciels **anti-virus** qui permettent de détecter et éradiquer les virus.

- ⇨ Des technologies dédiées permettant le cryptage des données circulant sur le réseau telles que les protocoles sécurisés.

- ⇨ Des outils de surveillance, des journaux de traces et des logiciels de détection d'intrusion **IDS**.

- ⇨ Pour finir, des scanners de vulnérabilités qui permettent de mettre en évidence les failles présentées par le réseau, que peuvent exploiter les pirates afin de corrompre le système. Ces scanners sont utilisés lors des tests d'intrusions effectués par les administrateurs réseaux pour anticiper les intrusions non désirées.

2.2 Tests d'intrusion

2.2.1 Définition

2.2.1.1 Qu'est ce qu'un test d'intrusion ?

Les tests d'intrusion constituent une tentative autorisée de simuler les activités d'un pirate qui veut s'approprier des ressources qui ne sont pas les siennes, ou nuire au bon fonctionnement d'un système d'informations, par exemple en le rendant indisponible.

Ces tests permettent d'avoir une image claire de la sécurité globale d'une entreprise ou d'un accès Internet chez un particulier. Ils correspondent à des attaques simulées d'un réseau. Ils permettent de tester la robustesse de la sécurité, d'apprécier l'efficacité des mécanismes mis en œuvre. Il est ainsi possible de savoir si les mécanismes mis en place permettent de stopper ou non un attaquant malintentionné.

Les tests d'intrusion ne peuvent pas se réduire à la simple utilisation d'un logiciel de détection automatique de vulnérabilités par balayage. Ils sont bien plus, en particulier ils nécessitent l'intervention d'une équipe de professionnels compétents qui eux vont identifier et

qualifier les failles de manière plus réfléchie et auront à l'esprit les conséquences des tests qu'ils effectueront. Néanmoins, les scanners de vulnérabilité présentent un certain intérêt dans leur caractère automatique mais ils ne suffisent pas à eux seuls à obtenir une bonne détermination des failles de vulnérabilité que présente un réseau.

2.2.1.2 Stratégie de tests

Il existe plusieurs stratégies de tests :

- ⇨ Les tests externes qui correspondent à un examen des services disponibles via Internet.

- ⇨ Les tests internes qui exploitent les failles de vulnérabilité qui pourraient être disponibles à un attaquant en provenance d'Internet ayant réussi à s'introduire dans le réseau ou à un employé malveillant.

Les méthodes et techniques utilisées dans les tests internes ou externes sont identiques. La seule différence notable est l'étendue des connaissances relatives au réseau, en possession des attaquants.

Pour simuler ce degré de connaissance du système, les tests d'intrusion peuvent se faire de plusieurs façons :

- ⇨ Test en aveugle : les équipes en charge du test ont un accès limité aux renseignements relatifs à la configuration du système d'information

- ⇨ Test en double aveugle : seule la personne qui est à l'initiative du test est au courant, la personne en charge de la sécurité ne l'est pas.

- ⇨ Test ciblé : l'équipe de sécurité est au courant et a des connaissances sur le réseau et sur la cible visée.

2.2.1.3 Types de tests

Il existe différents types de tests parmi lesquels nous pouvons noter ceux relatifs :

- ⇨ à la sécurité des applications Web.

Les points à évaluer alors sont ceux relatives à la confidentialité et l'intégrité des communications sur le réseau, à l'authentification des utilisateurs, à l'intégrité des **sessions** entre l'internaute et les applications, à la gestion des informations stockées sur les postes clients telles que les **cookies**…

- ⇨ au déni de service (**DoS**)

Les dénis de service sont des attaques dont le but est de rendre indisponibles des services ou de faire tomber un serveur proposant des services. Ceci peut poser des problèmes importants lorsque la disponibilité continue des services est impérative.

⇨ au scannage de numéros de téléphone (**War dialing**)

Cette technique consiste à identifier des connexions à l'Internet au sein d'une entreprise par appel systématique d'une série de numéros de téléphone. Une fois les modems ou autres dispositifs d'accès détectés, les techniques d'analyse et d'exploitation sont employées pour déterminer dans quelle mesure il est possible d'infiltrer le réseau de l'entreprise.

⇨ au réseau sans fil

Ce type de test a pour but de cerner les lacunes ou les faiblesses en matière de sécurité dans la conception, la mise en œuvre ou l'exploitation des réseaux sans fil. Par exemple, nous pouvons citer le war driving qui consiste à rechercher des réseaux sans fil non sécurisés à l'aide d'une voiture, d'un ordinateur portable, d'une carte **Wifi** et d'une antenne.

⇨ à l'**ingénierie sociale**

Ces techniques exploitent les interactions sociales, qui impliquent les utilisateurs d'un système d'information cible, afin de recueillir de l'information et de s'infiltrer dans le réseau de l'organisation. Tous les moyens sont bons, cela peut aller jusqu'à l'usurpation d'identité.

2.2.1.4 Leurs limites

Les tests d'intrusion peuvent échouer, ce qui ne signifie pas que le système ne présente pas de faille de vulnérabilité.

Il est difficile voire impossible de tester toutes les failles de vulnérabilité présentes dans un réseau. Les scanners de vulnérabilité par exemple ne simulent pas toutes les nouvelles failles. Certaines des vulnérabilités ne sont pas prises en compte par ces derniers, soit parce que la vulnérabilité est totalement inconnue, soit parce que la mise à jour du logiciel prenant en compte cette vulnérabilité n'est pas encore disponible.

De plus, il est nécessaire de répéter de façon régulière ces tests. Tout ajout de matériel, l'apparition de nouveaux outils de piratage ou de nouvelles technologies remettent en cause les résultats des tests d'intrusion.

2.2.1.5 La démarche utilisée dans les tests d'intrusion

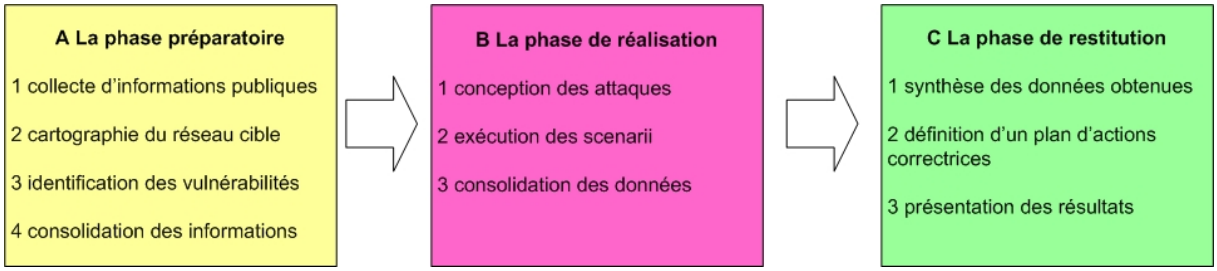


FIGURE 2 : LA DEMARCHE UTILISEE DANS LES TESTS D'INTRUSION

Nous nous intéressons ici à la description de la démarche employée dans les tests d'intrusion :

A. La phase préparatoire
    1. collection d'informations publiques (**DNS**, **WhoIs**, …)
    2. cartographie réseau de la cible (**ping, traceroute, nmap**)
    3. identification de vulnérabilité (**Nessus**)
    4. consolidation de données obtenues
B. La phase de réalisation
    1. conception des scénarii d'attaques à évaluer
    2. exécution des scénarii
    3. consolidation des données obtenues

Dans le cas des **hackers**, l'intrusion s'arrête ici. Ils nettoient généralement les traces laissées par leur passage.

Dans le cas d'un audit de sécurité, il faut alors fermer les brèches ouvertes, puis passer à l'étape suivante.

C. La phase de restitution
    1. synthèse des données obtenues lors des phases préparatoires puis de réalisation
    2. définition d'un plan d'actions correctrices
    3. présentation des résultats au commanditaire du test

Nous illustrerons dans la suite cette démarche en indiquant les différents outils utilisés agrémentés d'exemples.

2.2.2 L'audit de vulnérabilité

Nous parlons ici de l'audit en tant qu'étape de la phase préparatoire de la démarche utilisée dans les tests d'intrusion. Elle consiste à récupérer des informations relatives aux réseaux et aux systèmes présents sur ces derniers, dans le but d'identifier les failles de vulnérabilité. Les failles de vulnérabilité résultent en général de limites inhérentes à la conception des technologies ou découlent de mauvaises configurations ou utilisations. Les tests d'intrusions donnent des indications sur la facilité ou à l'inverse la difficulté d'accéder à l'information et au système d'informations en exploitant les vulnérabilités de sécurité. Les scanners de vulnérabilité correspondent à une façon automatisée de mise en évidence de ces failles. Ils indiquent la façon dont il est possible d'exploiter ces vulnérabilités et les méthodes permettant de résoudre les problèmes. Ils couvrent, en général, un large éventail de vulnérabilités connues. Tandis que les tests d'intrusion ciblent certaines vulnérabilités.

2.3 Outils liés aux tests d'intrusion

Nous allons reprendre dans cette partie les différentes étapes de la démarche utilisée dans les tests d'intrusion que nous avons présentées précédemment, dans lesquelles sont utilisés des outils spécifiques. Nous donnerons également des exemples afin de mieux illustrer la démarche.

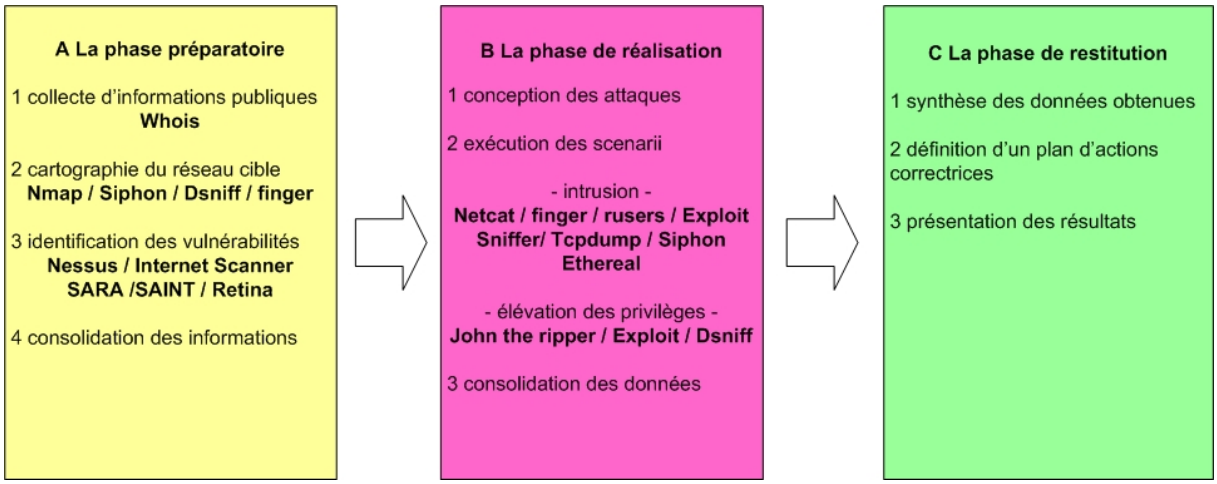


FIGURE 3 : EXEMPLES D'OUTILS UTILISES LORS D'UNE INTRUSION

2.3.1 Divers outils

2.3.1.1 Etape de collecte d'informations publiques (A.1.)

La commande **Whois** permet d'obtenir des informations publiques correspondant au réseau cible : nom du serveur **DNS**, nom du responsable, numéro de téléphone, adresse e-mail, description du réseau etc.

2.3.1.2 Etape de cartographie du réseau cible (A.2.)

Les outils utilisés dans cette étape permettent de récolter des informations relatives à la topologie du réseau afin de déterminer sa structuration. Parmi ces outils, nous pouvons trouver :

- **Nmap** qui est un scanner de réseau. Il permet de savoir quels sont les ports ouverts, fermés ou filtrés , ainsi que le système d'exploitation autorisé et sa version. Il permet par exemple de scanner un ensemble d'**adresses IP** en précisant la méthode de scan utilisée, les types de ports tels que les ports **UDP,** en tentant d'identifier la machine cible et en sauvegardant le résultat dans un fichier.

- **Siphon** permet de découvrir la topologie de la portion de réseau sur laquelle se trouve la machine où nous le lançons. Il indique les systèmes d'exploitation présents sur les machines , les ports ouverts , les machines qui ont le droit de se connecter au réseau. Il est ainsi possible de savoir pour quelle machine nous devons nous faire passer, afin de contourner les Firewalls.

- **Dsniff** permet de visualiser les paquets présents sur le réseau et ainsi de récupérer des clefs en sniffant (**sniffer**).

- **Finger** permet d'obtenir des comptes valides. En général, le **démon** correspondant est désactivé.

2.3.1.3 Etape d'identification des vulnérabilités (A.3.)

L'objectif est d'identifier les failles potentielles présentes sur le réseau en utilisant des scanners de vulnérabilité tels que **Nessus**. L'utilisation de ces logiciels n'est pas très discrète. En effet, étant donné que ces logiciels testent des failles bien connues des **NIDS**, ils sont facilement repérables. Les **NIDS** sont des systèmes de détection d'intrusion basés au niveau d'un réseau. Si nous souhaitons rester discrets, lors de l'audit de vulnérabilité, il est préférable de rechercher les vulnérabilités existantes sur Internet, sur des sites tels que **SecurityFocus** (www.securityfocus.com), **Bugtraq** (www.bugtraq.org). Les autres outils d'audit de vulnérabilité sont abordés un peu plus loin.

Une fois un certain nombre de vulnérabilités identifiées, il est alors nécessaire d'éliminer les failles non fondées. Pour cela nous utilisons des petits programmes appelés **Exploits.** Leur objectif est d'exploiter les vulnérabilités auxquelles ils correspondent. Ils sont programmés dans divers langages. Il n'y a pas d'automatisation qui permet l'utilisation ensembliste des **Exploits** mais il existe néanmoins des bibliothèques dédiées.

2.3.1.4 Etape d'exécution des scénarii (B.2.)

Nous pouvons distinguer deux ensembles de produits utilisés dans cette étape, ceux qui permettent de s'introduire sur un ordinateur ou un serveur et ceux qui permettent de se procurer des privilèges auxquels nous n'avons normalement pas accès.

a) <u>Outils d'intrusion</u>

Finger et **Rusers** permettent de trouver des comptes valides. Ce sont des commandes système de base d'**Unix**.

Exploits permettent d'exploiter des vulnérabilités afin d'avoir accès à un poste.

Netcat est un utilitaire multifonctions pour le réseau. Il fonctionne en client ou serveur en utilisant le protocole **TCP**. Il permet de simuler des services et d'écouter l'activité correspondant à un port donné.

Certains outils permettent la récupération de l'information qui circule sur le réseau. Nous pouvons citer par exemple : **Sniffer, TCPdump, Siphon, Ethereal.**

Ces outils permettent donc de s'introduire sur des serveurs. Par exemple, dans le cas où un **routeur** supporte la **source routing**, il est possible de recourir à **IP spoofing**. L'**IP spoofing** est une technique qui permet à une machine d'être authentifiée auprès d'une autre au moyen de paquets semblant émaner d'une adresse source habilitée. Cette technique peut être faite en deux étapes. Nous plaçons un alias sur l'interface réseau d'un poste, ensuite, nous utilisons une propriété du protocole IP qui est la possibilité de choisir la route à emprunter. **Netcat** permet alors d'établir une connexion **Telnet** en précisant la route à emprunter.

b) <u>Outils permettant l'élévation de privilèges</u>

John the ripper qui permet d'obtenir des mots de passe en clair à partir de mots de passe cryptés.

Exploits qui permettent par exemple de devenir administrateur.

Dsniff qui ne capture que les mots de passe. Il supporte divers protocoles (**SNMP**, **Netbios)** et peut être utilisé avec des services tel que **Telnet**.

2.3.2 Outils d'audit

Il existe de nombreux logiciels qui permettent d'automatiser la découverte de vulnérabilités, nous les appelons des scanners. Ils permettent d'évaluer les vulnérabilités présentes sur les réseaux. Ils se déclinent sous plusieurs formes et donnent des résultats avec des précisions variables.

2.3.2.1 Internet Scanner

Parmi ces logiciels, nous pouvons citer **Internet Scanner** de la société **ISS**. Il peut s'intégrer au produit **ISS** Décisions pour être utilisé avec d'autres produits de sécurité tels que les systèmes de détection d'intrusions et les **firewall** (pare-feu).

2.3.2.2 SATAN, SAINT, SARA

Vers le début des années 90, est apparu **SATAN** qui a remporté un énorme succès dans le domaine. Il avait la particularité d'être **open source**. Il n'est plus mis à jour depuis plusieurs années. Mais, il existe une myriade d'outils similaires tel que l'outil **open source SARA** qui est la troisième génération d'outil d'analyse basé sur **SATAN**, ou la solution commerciale **SAINT**.

2.3.2.3 Retina

Nous pouvons également citer le logiciel **Retina** de la société eEye, qui est rapidement devenu populaire. Il analyse le trafic sur chaque port afin de déterminer le service utilisé. Il existe une fonction nommée **CHAM** permettant de découvrir de nouvelles failles de vulnérabilité. Cette méthode repose sur un moteur d'intelligence artificiel. **Retina** est une solution commerciale.

2.3.2.4 Nessus, NeWT

Enfin, il existe une autre offre dont nous allons parler de façon plus approfondie dans ce document, l'outil **Nessus**, et sa version Windows appelée **NeWT**. **Nessus** est un outil **open source**. Un français, renaud Deraison, est l'auteur et l'animateur de se projet. La version Windows est disponible sur le site de la société Tenable Network Security en version d'évaluation. **Nessus** semble être l'un des outils les plus populaires du moment.

3. Nessus

3.1 L'outil

3.1.1 Objectif

Nessus permet d'auditer des réseaux possédant divers systèmes tels que les différentes versions de Windows et de nombreuses déclinaison d'**Unix** telles que **Linux**, **FreeBSD,** Sun Solaris, HP-UX, IBM AIX … **Nessus** permet de faire des tests d'intrusion aussi bien interne qu'externe. Les audits peuvent donc avoir lieu à l'intérieur d'une entreprise ou à l'extérieur à travers Internet à l'aide d'un poste connecté au **Web**.

Nessus balaye les ports d'un serveur et recherche puis identifie les failles de vulnérabilité présentes. Il indique les méthodes que peuvent utiliser les **hackers** pour s'introduire à l'intérieur du réseau audité.

Il analyse les protocoles utilisés sur chacun des ports du serveur afin d'identifier les services présents. Il est ainsi capable de détecter les services même si ces derniers n'utilisent pas les ports qui leurs sont attribués par défaut. Par exemple, il sera capable de détecter un service **FTP** disponible sur un port autre que le port 21. Il est également capable de détecter les services multiples d'un même serveur. En effet, si deux serveurs Web tournent sur des ports différents qui ne sont pas les ports attribués par défaut, **Nessus** les détectera tous les deux.

A la fin du balayage des ports, **Nessus** présente la liste des failles de vulnérabilités et dans la majorité des cas, indique également la façon d'y remédier .

Afin de permettre une recherche d'informations plus aisée, chaque faille de vulnérabilité est associée à des identifiants, ce qui permet aux administrateurs de trouver davantage d'informations sur les vulnérabilités publiques. Par exemple, **CVE (**Common Vulnerabilities and Exposures**)** propose une liste de noms standardisés pour les vulnérabilités et les autres expositions de sécurité informatique. L'objectif de **CVE** est de constituer un référentiel de noms standardisés pour les vulnérabilités publiquement connues et les révélations relatives à la sécurité. Ce type de base d'informations est également proposé par le **CERT** et l'**ICAT**. **Nessus** associe à la plupart des failles de vulnérabilité identifiées des identifiants **CVE**, qui permettent d'obtenir davantage d'informations sur le **Web**.

3.1.2 Architecture

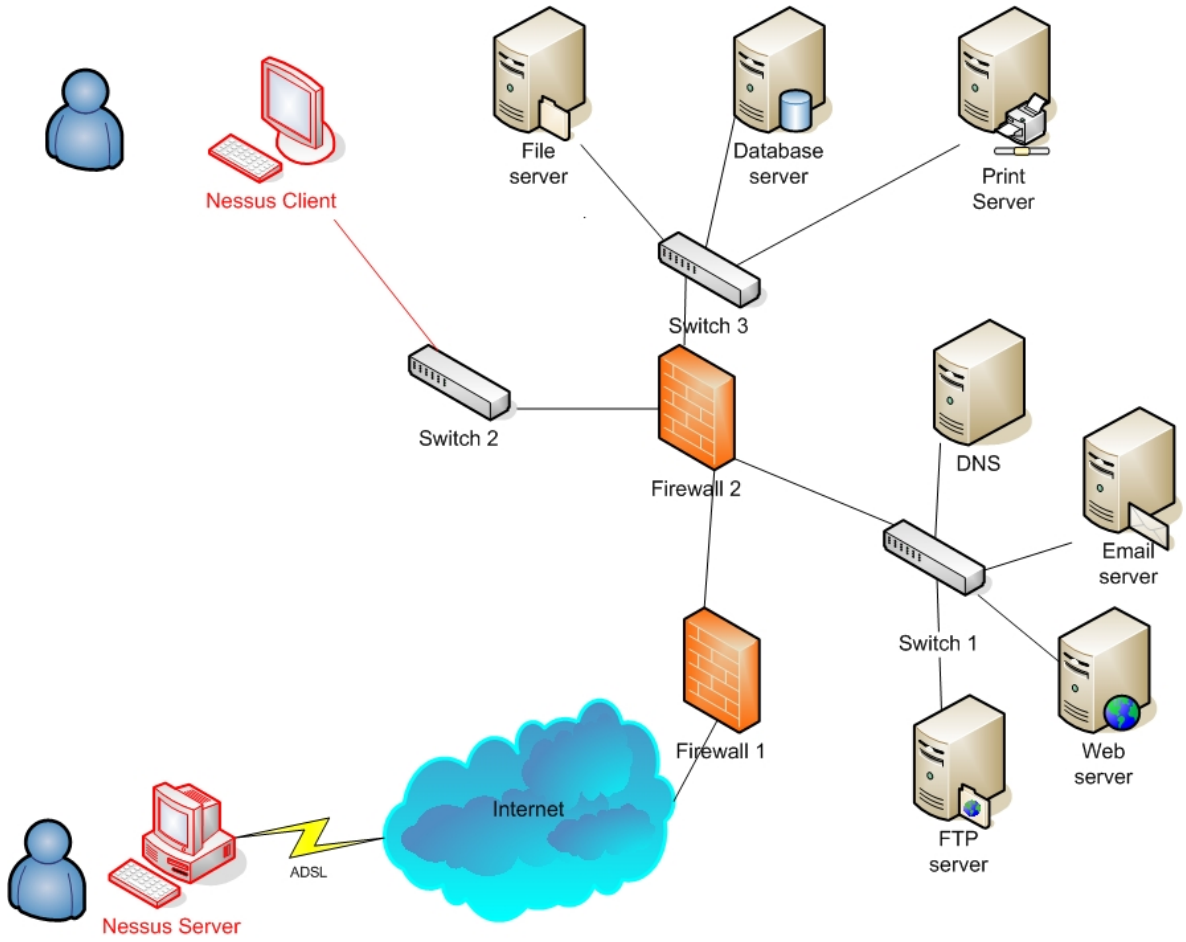


FIGURE 4 : TEST D'INTRUSION EXTERNE

Nessus est basé sur une architecture client / serveur qui permet de multiples configurations. En effet, nous pouvons placer le **démon** de **Nessus** à l'extérieur du réseau sur l'Internet afin d'effectuer des séries de tests externes. Le client lui est à l'intérieur du réseau. Il permet de contrôler le serveur et de configurer le serveur qui effectue l'attaque proprement dite de la machine cible. Il est ainsi possible d'avoir une vision claire des services effectivement vulnérables à partir d'Internet.

Nessus intègre d'importantes bases de connaissances relatives aux services proposés sur divers systèmes d'exploitation, aux failles de vulnérabilité et aux résolutions des problèmes crées par la présence des failles de vulnérabilité. La base de données a l'avantage d'être largement évolutive grâce au système de **plug-in**.

En effet, chaque test de sécurité se présente sous forme d'un **plug-in** extérieur. il est possible d'écrire ses propres tests sous forme de **plug-in** à l'aide d'un langage de script dédié **NASL**. De plus, des **plug-in** correspondant aux failles de sécurité les plus récentes sont disponibles sur Internet. Nous en dénombrons plus de 2000 à ce jour. Ces **plug-in,** sur un système d'exploitation de type **Linux**, sont placés dans le répertoire dédié /usr/lib/nessus/plugin/ . Dans l'interface cliente, il est possible de choisir les **plug-in** que nous voulons prendre en compte.

Grâce à la base de données, les tests peuvent coopérer entre eux. En effet, si un service **FTP** ne permet pas la connexion en anonyme, les tests associés à ce type de connexion ne seront pas effectués. La base d'informations permet de rendre plus efficace l'exécution des tests dans le sens où **Nessus** ne lance pas de façon systématique l'intégralité des tests existants. Il ne teste que ce qui est nécessaire par rapport aux services présents sur la cible. De plus, il existe une notion de dépendances entre les tests. Il est possible d'indiquer à un test de ne s'exécuter que si un ou plusieurs autres tests particuliers ont été réalisés. Si ce n'est pas le cas, le test ne s'effectue pas.

De plus, **Nessus** utilise des logiciels tiers s'ils sont disponibles : Le scanner de port **Nmap** qui fournit des fonctionnalités avancées dans le domaine du balayage de port ; Le logiciel **Nitko** ou **Whisker** qui permet de faire des tests et des attaques spécifiques sur les serveurs **Web** et les scripts **CGI** ; enfin l'outil **Hydra** qui fournit des attaques **brute-force** pour des services tels que **Telnet**, **IMAP** … Les attaques **brute-force** ont pour objectif de trouver un mot de passe valide. Ce sont des méthodes d'analyse de chiffrement où toutes les clefs possibles sont systématiquement essayées. Elles sont généralement basées sur un générateur ou un dictionnaire. Le fait que **Nessus** utilise des logiciels tiers tient du principe qu'il n'est pas nécessaire d'implémenter de nouveau ce qui existe déjà et répond parfaitement aux besoins.

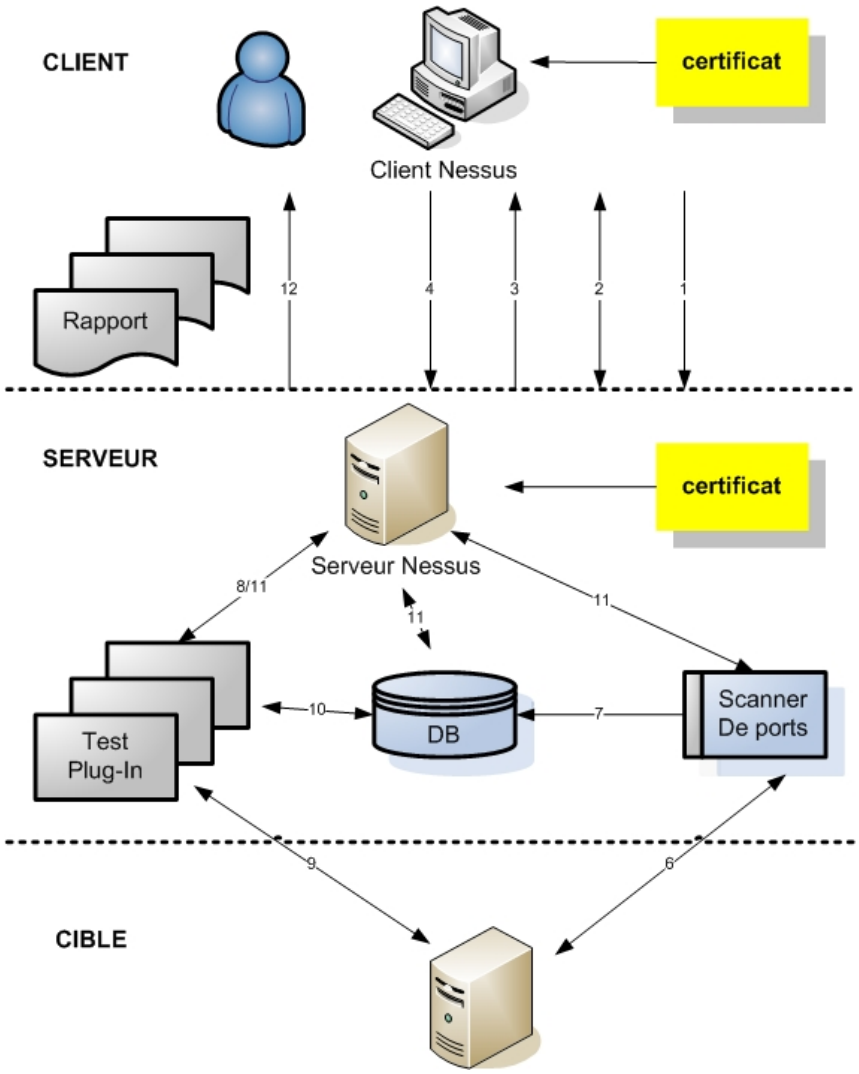


FIGURE 5 : LE FONCTIONNEMENT DE NESSUS

Le principe de fonctionnement de **Nessus** est le suivant :

1. Le client **Nessus** se connecte et s'identifie.

2. Le client et le serveur s'échangent leurs **certificats** afin de crypter les données et que le serveur authentifie le client. Les **certificats** sont des fichiers chiffrés qui permettent d'authentifier les différents intervenants lors de transactions sur Internet.

3. Le serveur informe le client des différents tests et options disponibles.

4. Le client envoie les différents paramétrages au serveur.

5. Le serveur **Nessus** effectue un balayage de la cible à l'aide des différents scanners de port à sa disposition. Le scanner de port employé peut être **Nmap**.

6. La réalisation du scan.

7. Les informations récoltées lors du scan sont enregistrées dans la base de données.

8. Le serveur **Nessus** effectue les tests correspondant aux données recueillies lors du balayage des ports. Par exemple si le port 23 est ouvert, les test correspondant à **Telnet** sont lancés.

9. Les **plug-in** de tests analysent la cible en se reposant sur la base de données.

10. Les **plug-in** enregistrent les informations relatives aux tests dans la base de données.

11. Toutes les informations sont envoyées au serveur **Nessus** lors de l'exécution des tests.

12. Les informations récoltées ainsi que leurs analyses sont mises à la disposition de l'utilisateur.

3.2 Mise en oeuvre

3.2.1 Installation

L'installation de **Nessus** est assez rapide. Il est possible de télécharger les sources sur le site de **Nessus** http://www.nessus.org. Il est important de noter qu'il existe un package d'installation automatique nommé nessus-installer.sh qui facilite grandement l'installation de **Nessus** sous **Linux**. Pour plus d'informations, vous pouvez vous reporter à l'Annexe D « **Nessus** : installation et configuration » de ce document.

3.2.2 Configuration

Après avoir installé le client et le serveur **Nessus**, il est nécessaire de créer un compte **Nessus**. Il est possible de définir des restrictions pour chaque utilisateur. Ainsi, **Nessus** serveur peut être utilisé par un ensemble d'administrateurs qui ne pourront tester que la partie

du réseau qui leur est attribuée. Dans les tests effectués par la suite, le profil utilisé ne comporte aucune restriction.

Ensuite, il est possible de préciser un certain nombre d'options du serveur à l'aide d'un fichier de configuration. Une fois ceci fait, nous pouvons lancer le serveur **Nessus**, puis le client **Nessus**.

Il est alors nécessaire de renseigner les différents onglets qui suivent :

- **Nessusd host :** permet de définir le serveur **Nessus** et de s'y connecter

- **Plug-ins :** il est possible ici de choisir la liste des **plug-in** que nous exécuterons lors de la détection des vulnérabilités.

- **Prefs :** cet onglet permet d'indiquer des informations complémentaires à **Nessus**, que des pirates sont susceptibles d'avoir en leurs possession. Cela peut être par exemple le nom et le mot de passe correspondant à un compte de type **FTP**, **SMB**, **IMAP** … etc. La connaissance de ces informations complémentaires peut permettre à **Nessus** de détecter davantage de vulnérabilités et rendre ainsi le scan plus complet.

- **Scan options :** c'est ici que nous indiquons la plage de ports que nous souhaitons scanner. Il est également possible d'activer l'option « safe checks » pour éviter de faire tomber le serveur ciblé. Il est également possible d'activer ou de désactiver de nombreuses autres options : le nombre d'ordinateurs à tester en même temps, choix des scanners de port utilisé …

- **Target selection :** nous effectuons le choix de la ou les machines visées. Nous pouvons spécifier une **adresse IP** ou une plage d'**adresse IP**. La notion **CIDR** est supportée.

- **User :** permet de spécifier des règles pour, par exemple, exclure une **adresse IP** sur laquelle nous ne désirons pas effectuer de tests.

Se référer à l'Annexe D « **Nessus** : installation et configuration », afin de voir les copies d'écran des différents onglets. Après tout ceci, il est possible de commencer le test de vulnérabilité.

3.3 Tests disponibles

Nous distinguons deux grands ensembles de tests : ceux qui correspondent à des attaques dangereuses pour la cible tels que les dénis de service qui peuvent avoir pour conséquence l'indisponibilité du système et ceux qui ne présentent pas de risques.

Ces deux grands ensembles de tests sont divisés en 24 familles :

- **Backdoors** : attaques et tests relatifs aux programmes qui détournent les fonctionnalités systèmes dans le but d'ouvrir des accès utiles aux pirates. Ils sont généralement contenus à l'intérieur de programmes inoffensifs
- **CGI** abuses : tests correspondants aux programmes écrits en script (**php**, **perl**…) utilisés sur les serveurs **Web**

- ⇨ **CISCO**: tests relatifs aux **routeurs CISCO**
- ⇨ Default Unix accounts : tests correspondant aux comptes définis par défaut
- ⇨ Denial of service **DoS** : tests d'attaque de type déni de service
- ⇨ **Finger** abuses : test détournant la commande **finger** qui permet d'obtenir des informations sur un utilisateur connecté à un réseau informatique
- ⇨ **Firewalls** : analyse relative aux logiciels permettant de contrôler le trafic
- ⇨ **FTP** : tests du protocole de transfert de fichier
- ⇨ Gain of shell remotely : tests relatifs à l'obtention d'un interpréteur de commande à distance tel que **SSH**
- ⇨ Gain root remotely : tests relatifs a l'obtention à distance de privilèges
- ⇨ General : tests liés aux informations générales relatives aux systèmes et aux logiciels
- ⇨ Misc : tests divers
- ⇨ **Netware** : tests liés au système d'exploitation développé par Novell corporation pour différents type de **LAN**
- ⇨ **NIS** : tests relatifs aux services d'informations sur le réseau
- ⇨ Peer-to-peer File sharing : tests relatifs aux partages de fichiers de type peer to peer
- ⇨ Port scanners : scanner de port utilisé par **Nessus**
- ⇨ Remote file access : tests d'accès à des fichiers distance
- ⇨ **RPC** : tests de détection de différents services proposés
- ⇨ Settings : **plug-in** relatif au paramétrage de **Nessus**
- ⇨ **SMTP** problèmes **:** tests relatifs aux problèmes relatifs au serveur mails.
- ⇨ **SNMP** : tests relatifs à ce protocole permettant d'administrer les réseaux **TCP/IP**
- ⇨ Useless services : tests relatifs aux services qui ne sont plus utiles mais qui peuvent être encore activés.
- ⇨ **Windows** : tests correspondant à des informations générales relatives aux systèmes et aux logiciels de type Windows
- ⇨ Windows - user management : tests touchant l'administration des utilisateurs.

3.4 Les rapports

3.4.1 Consultation

La consultation des résultats obtenus se fait au travers du client **Nessus** qui propose 5 parties distinctes dans la fenêtre d'affichage (Se reporter à l'Annexe D : « **Nessus** : installation et configuration » pour voir une copie d'écran de cette fenêtre).

- ⇨ Une première zone permet de sélectionner un sous réseau qui vient d'être testé.
- ⇨ Une seconde zone permet de choisir un ordinateur ou un serveur au moyen de son **adresse IP**.
- ⇨ Une troisième zone indique les ports qui ont été découverts avec une indication du niveau de gravité maximale associé à chaque port.
- ⇨ Une quatrième zone permet d'avoir la liste des failles découvertes sur un port avec leurs niveaux de gravité. C'est ici qu'est indiqué si le port a des alertes de sécurité, des notes de sécurités ou des trous de sécurités.

Lorsque nous sélectionnons un type de gravité, la liste exhaustive des vulnérabilités de ce type pour le port apparaît dans la cinquième zone. Les informations fournies dans cette dernière zone nous indiquent comment des pirates peuvent exploiter les failles, mais aussi comment nous pouvons les combler. Les références de type identifiant **CVE** et autres sont indiquées pour chaque faille afin que nous puissions aller chercher d'avantage d'informations sur le **Web**.

Les informations ainsi obtenues sont exportables sous divers formats. Nous pouvons citer par exemple les formats : **ASCII** text, **LaTeX**, **HTML** ou « Spiffy » **HTML**, c'est à dire des documents **HTML** qui comportent des graphiques et des graphes.

Il est à noter que l'interface cliente standard n'est pas la seule. Il existe d'autres interfaces, en particulier une interface Web qui est disponible sur le site officiel de **Nessus**.

3.4.2 Structuration

Nessus fournit des rapports complets. Ils ne nous indiquent pas simplement ce qui ne va pas avec notre environnement, mais la plupart du temps, nous donnent des conseils de mise à jour des services détectés comme étant vulnérables et nous donne le niveau de risque associé à chacun des problèmes trouvés en trois catégories : vulnerability, warning et informational. Si nous prenons un rapport de type **HTML**, **Nessus** va nous indiquer les informations suivantes :

- ⇨ Le nombre d'ordinateurs testés, le nombre total de trous de sécurité et de warning.
- ⇨ la liste des ordinateurs testés en indiquant leurs **adresses IP**
- ⇨ Par ordinateur, il indique les ports et services découverts ainsi que le niveau de gravité le plus élevé correspondant à ce port et à ce service. Ainsi que la liste des messages donnés par **Nessus** en indiquant le risque associé (vulnerability, Warning, Informationnal), le service et port correspondant et les problèmes et solutions possibles. Il fournit également les identifiants **CVE** et autres tels que **IAVA** ainsi qu'un identifiant propre à **Nessus** nommé **Nessus ID.**

3.5 Evolutions

Renaud Deraison et Ron Gula ont fondé la société **TNS** Tenable Network Security. Renaud Deraison est comme nous l'avons vu à l'origine du projet **Nessus**. Ron Gula, quant à lui, a travaillé à la réalisation de **DRAGON** qui est un outil de détection d'intrusion. Cette société propose une offre qui repose entre autre sur l'association de **Nessus** avec un système de détection d'intrusion (**IDS**). L'un des objectifs est d'avoir une interface qui centralise les résultats de plusieurs scanners **Nessus**, ainsi que des alertes d'**IDC**s placées sur le réseau, dans une unique console d'administration. Ceci permet de corréler les informations provenant des **IDC**s avec celles provenant des scanners. Il est ainsi possible de déterminer si une tentative d'intrusion est critique ou pas, par rapport aux failles de vulnérabilité présentes.

Un second objectif de cette offre est la couverture de grands réseaux tels que les réseaux de **classe B** tout en sollicitant moins l'infrastructure. Pour y parvenir, l'outil repose sur une architecture nommée Lightning, basée sur des Agents relais (**proxy**) qui peuvent piloter des ensembles de scanners.

L'évolution de **Nessus** et des autres outils de détection de vulnérabilité irait donc vers une meilleure intégration de ces derniers dans une solution globale de sécurisation des réseaux qui permettrait de centraliser davantage les informations des divers outils relatifs à la sécurité informatique afin de travailler plus efficacement.

4. Les tests réalisés avec Nessus

4.1 Protocole des tests

Nous allons dans cette partie effectuer une série de tests à l'aide de **Nessus**. Suite à des contraintes de type matériel, la stratégie adoptée est une stratégie de type interne. Le client et le serveur sont tous deux situés à l'intérieur d'un réseau local **Ethernet**. L'objectif est de mieux appréhender ce qu'est un scanner de vulnérabilité au travers de tests simples.

Test 1 : Audit avec Nessus sans utiliser d'attaques dangereuses

Nous effectuons des tests de vulnérabilité sur plusieurs ordinateurs qui possèdent des systèmes d'exploitation différents. Divers services et partages de répertoire ont été activés. Les tests effectués ne sont pas de type destructif. Des attaques dites dangereuses ne sont pas utilisées dans ce Test. Nous parlons d'attaques dangereuses dans la cas où elles peuvent rendre des services ou un serveur indisponibles. Les attaques de Déni de Service **DoS** sont un exemple d'attaque dangereuse. Une attaque **DoS** permet de faire tomber un service pour qu'il ne soit plus assuré.

Test 2 : Audit avec Nessus en utilisant des attaques dangereuses

Tous les types d'attaque sont utilisés dans ce test même celles de type destructif de type Déni de Service **DoS**. Ceci va permettre de savoir dans quelle mesure les attaques dangereuses influent sur les résultats obtenus.

Test 3 : Utilisation de Nessus avec Nmap et des attaques dangereuses

Nessus utilise dans ce test en plus de ses scanners de port par défaut le logiciel **Nmap** qui est réputé pour être l'un des meilleurs scanners de ports **open source**. Nous pouvons ainsi avoir une idée des apports de ce logiciel **Nmap** au niveau des audits effectués par **Nessus**.

4.2 Déploiement

4.2.1 Topologie

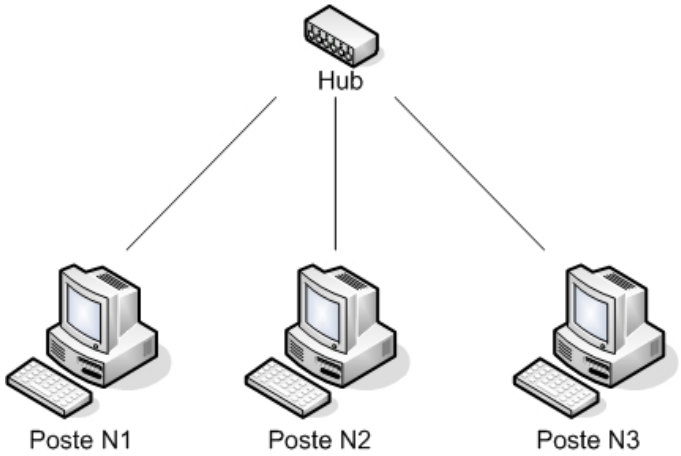


FIGURE 6 : LE RESEAU ETHERNET UTILISE POUR LES TESTS

La figure ci dessus représente le réseau **Ethernet** utilisé pour les tests.

Le réseau **Ethernet** utilisé pour les tests est constitué de trois ordinateurs reliés entre eux via un **HUB** 10 mégabits. Différents systèmes ont été installés sur ces postes. Les configurations de postes ont été indiquées dans le tableau suivant :

Ordinateur	**Poste N1-Linux**	**Poste N2-98**	**Poste N3-2000**
Adresse IP	168.192.1.2	168.192.1.3	168.192.1.4
Micro-processeur	Duron 1 Ghz	Celeron 650 Mhz	Duron 750
Mémoire vive	128 méga	128 méga	260 méga
OS	Linux Fedora core 1B	Windows 98 SE 2	Windows 2000 serveur
Scanner de vulnérabilité	Nessus 2.10.a.		
Sniffer		Ethereal 0.10.3*	

Des IP fixes ont été utilisées pour identifier les postes sur le réseau.

* **Ethereal** 0.10.3, un outil permettant le comptage des paquets qui transitent sur le réseau, a été installé sur le poste N2-98. Il va nous permettre de nous faire une idée du volume de paquets **TCP**, **UDP**, **ICMP**, **ARP** et autres types de paquets émis vers le poste N2-98. Ces outils sont appelés des **sniffers**. Ils permettent de renifler (dans le sens de capturer) les informations circulant sur le réseau

4.2.2 Paramétrage

Test 1 : Audit avec Nessus sans utiliser d'attaques dangereuses

Lors de l'installation, les paramétrages par défaut des systèmes d'exploitation ont été conservés. Les postes N1-Linux et N3-2000 sont protégés à l'aide de mot de passe. Le client et le serveur **Nessus** ont été installés sur le poste N1-Linux. Le scanner de port **Nmap** n'est pas utilisé pour ce test de vulnérabilité. La plage de port scanné est comprise entre 1 et 1500.

Test 2 : Audit avec Nessus en utilisant des attaques dangereuses

Les tests sont menés sur le poste N2-98 et N3-2000. Les tests correspondant aux attaques dangereuses sont activées dans ce test. Le scanner de port **Nmap** n'est pas utilisé pour ce test de vulnérabilité.

A. avec le poste N3-2000, sur une plage de ports comprise entre 1 et 1500.
B. avec le poste N3-2000, sur une plage de port comprise entre 1 et 65536.
C. avec le poste N2-98, sur une plage de ports comprise entre 1 et 1500.

Test 3 : Utilisation de Nessus avec Nmap et des attaques dangereuses

Les tests sont menés sur le poste N2-98 et N3-2000. Les tests correspondant aux attaques dangereuses sont activées dans ce test. On a indiqué à **Nessus** d'utiliser le scanner de port **Nmap** lors de l'audit.

A. sur le poste N2-98, sur une plage de ports comprise entre 1 et 1500.
B. sur le poste N3-2000, sur une plage de ports comprise entre 1 et 1500.

4.3 Exploitation

4.3.1 Résultats

Les tableaux utilisés dans cette partie correspondent à des comptages globaux des vulnérabilités afin de se faire une idée des failles de vulnérabilité présentes. Chacun des tests

réalisés ci-après à généré des rapports d'environ 16 pages. Les failles de vulnérabilité les plus importantes ont été mentionnées dans ce qui suit, ainsi que des informations relatives aux paquets utilisés lors des audits par **Nessus**.

Nessus donne trois types d'indications qui sont les suivantes :

- ⇨ Trou de sécurité : indique les failles de vulnérabilités présentent
- ⇨ Alerte de sécurité : indique des failles qui peuvent devenir des trous de sécurité
- ⇨ Message de sécurité : donne la possibilité à un attaquant de fournir des informations sur le poste

Test 1 : Nessus sans Nmap - sans attaque de type DoS

Ordinateur cible : Poste N1-Linux / Poste N2-98 / Poste N3-2000
Durée du test : 20 minutes
Plage de port : 1-150

Informations obtenues :

	Trou de sécurité	**Alerte de sécurité**	**Message de sécurité**
Total des indications	17	58	74

Nombre de ports concernés sur	**Trou de sécurité**	**Alerte de sécurité**	**Message de sécurité**
Poste N1-Linux	4	9	8
Poste N2-98	1	3	1
Poste N3-2000	6	16	13

Le total des indications représente le nombre total d'indications obtenues lors de l'audit. Le nombre de ports concernés indique le nombre de ports correspondant à chaque type d'indication pour chacun des postes cibles. Un port peut correspondre à plusieurs indications du même type. C'est pour cette raison que le total des indications ne correspond pas à la somme des ports concernés pour un type d'indication donné.

Services problématiques découverts sur		
Poste N1-Linux	**Poste N2-98**	**Poste N3-2000**
Ssh (22/tcp)	Netbios-ssn (139/tcp)	ftp (21/tcp)
Netbios-ssn (139/tcp)		http (80/tcp)
Unknown (1024/udp)		Unknown (135/tcp)
Unknown (665/tcp)		Unknown (135/udp)
		Netbios (139/tcp)
		snmp (161/tcp)

Nous n'indiquons dans les tableaux de ce type qui suivent que les ports auxquels sont associés des trous de sécurité. Unknown est indiqué lorsque le service présentant la vulnérabilité n'est pas très connu. Pour plus d'informations concernant les vulnérabilités, vous pouvez vous reporter à l'Annexe E « Failles de vulnérabilités détectées » de ce document.

Comptage par Ethereal des paquets envoyés sur poste N2-98					
Total	**TCP**	**UDP**	**ICMP**	**ARP**	**Other**
8302	7766	270	220	18	2
100%	93.8	3.3	2.7	0.2	0

Commentaire : Ce tableau nous donne une idée du type de paquets qui sont envoyés sur une cible lors d'un audit effectué par **Nessus**.

<u>Test 2 : Nessus sans Nmap - avec attaque de type DoS</u>

A. avec le poste N3-2000, sur une plage de ports comprise entre 1 et 1500.
On utilise des attaques de type **DoS**.
Ordinateur cible : Poste N3-2000
Durée du test : 23 minutes
Plage de port : 1-1500

Informations obtenues :

	Trou de sécurité	**Alerte de sécurité**	**Message de sécurité**
Total des indications	15	35	38

Nombre de ports concernés sur	**Trou de sécurité**	**Alerte de sécurité**	**Message de sécurité**
Poste N3-2000	8	14	11

Services problématiques découverts sur le poste N3-2000		
ftp (21/tcp)	Unknow (135/tcp)	Snmp (161/udp)
Smtp (25/tcp)	Unknow (135/udp)	General / tcp
http (80/tcp)	Netbios-ssn (139/tcp)	

Commentaire : Le fait d'utiliser des attaques de type **DoS** affine les résultats du test. Il y a davantage de failles qui sont considérées comme des trous de sécurité. La durée du test est légèrement supérieure au test précédent. **Nessus** fait uniquement les tests utiles par rapport au balayage de port qu'il a effectué.

B. avec le poste N3-2000 sur une plage de port comprise entre 1 et 65536.
Nous utilisons ici l'intégralité des attaques qu'elles soient dangereuses ou non. Il y a plus d'attaques dangereuses que dans le test précédent.
Ordinateur cible : Poste N3-2000
Durée du test : 2 heures
Plage de port : 1-65536

Informations obtenues :

	Trou de sécurité	**Alerte de sécurité**	**Message de sécurité**
Total des indications	4	27	35

Nombre de ports concernés sur	Trou de sécurité	Alerte de sécurité	Message de sécurité
Poste N3-2000	4	14	9

Services problématiques découverts sur le poste N3-2000			
ftp (21/tcp)	Telnet (23/tcp)	Netbios-ssn (139/tcp)	Snmp (161/udp)

Commentaire : Le balayage des ports prend beaucoup plus de temps lorsque nous choisissions une plage plus importante. Le temps du balayage semble proportionnel à la plage de ports disponibles. De plus, l'audit n'a révélé que la présence d'un service terminal serveur hors de la plage de ports comprise entre 1 et 1500.

Le fait d'avoir utilisé la panoplie complète des tests dangereux a fait disparaître un certain nombre de **faux positifs** . Les tests dangereux permettent de réduire les fausses alarmes mais ont pour inconvénient de faire tomber les services que proposent les diverses machines. C'est aussi une raison pour laquelle des failles de vulnérabilités disparaissent. Il est sans doute nécessaire d'employer une démarche plus ciblée lorsque nous utilisons des attaques dites dangereuses pour ne pas faire tomber les services en cours de test et les rendre ainsi indisponibles aux attaques qui suivent.

C. avec le poste N2-98, sur une plage de ports comprise entre 1 et 1500.
Ordinateur cible : Poste N2-98
Durée du test : 10 minutes
Plage de port : 1-1500

Informations obtenues :

	Trou de sécurité	Alerte de sécurité	Message de sécurité
Total des indications	5	9	3

Nombre de ports concernés sur	Trou de sécurité	Alerte de sécurité	Message de sécurité
Poste N2-98	1	3	1

Services problématiques découverts sur le Poste N2-98
Netbios-ssn (139/tcp)

Comptage par Ethereal des paquets envoyés sur poste N2-98					
Total	**TCP**	**UDP**	**ICMP**	**ARP**	**Other**
13310	8265	2147	2878	8	12
100	62.1	16.1	21.6	0.1	0.1

Commentaire : Les attaques de types **DoS** semblent utiliser davantage les protocoles de type **UDP** et **ICMP**. Le poste ne propose aucun service particulier, le résultat du test n'apporte rien de plus que le test 1. Cependant, bien que ce poste ne propose qu'un simple partage de fichier, il pourrait être exploité par un pirate ayant réussi à s'introduire sur le réseau.

Test 3 : Nessus avec Nmap - avec attaque de type DoS

A. sur le poste N2-98, sur une plage de ports comprise entre 1 et 1500.
Ordinateur cible : Poste N2-98
Durée du test : 10 minutes
Plage de port : 1-1500

Informations obtenues :

	Trou de sécurité	**Alerte de sécurité**	**Message de sécurité**
Total des indications	5	9	3

Nombre de ports concernés sur	**Trou de sécurité**	**Alerte de sécurité**	**Message de sécurité**
Poste N2-98	2	3	1

Services problématiques découverts sur le Poste N2-98
Netbios-ssn (139/tcp)

Comptage par Ethereal des paquets envoyés sur poste N2-98					
Total	**TCP**	**UDP**	**ICMP**	**ARP**	**Other**
39535	31358	2681	5476	8	12
100	79.3	6.8	13.9	0	0

Commentaire : Le fait que **Nessus** utilise **Nmap** pour scanner les ports de l'ordinateur cible a triplé le nombre de paquets qui transitent sur le réseau. Les informations obtenues sont légèrement différentes par rapport aux résultats que donnait **Nessus** sans **Nmap**. Par exemple, une vulnérabilité de type problème général **ICMP** est apparue. **Nmap** affine les résultats de la détection.

B. sur le poste N3-2000, sur une plage de ports comprise entre 1 et 1500.
Ordinateur cible : Poste N3-2000
Durée du test : 38 minutes
Plage de port : 1-1500

Informations obtenues :

	Trou de sécurité	**Alerte de sécurité**	**Message de sécurité**
Total des indications	16	35	38

Nombre de ports concernés sur	**Trou de sécurité**	**Alerte de sécurité**	**Message de sécurité**
Poste N3-2000	8	13	4

Services problématiques découverts sur le poste N3-2000		
ftp (21/tcp)	Unknow (135/tcp)	Snmp (161/udp)
Smtp (25/tcp)	Unknow (135/udp)	General / tcp
http (80/tcp)	Netbios-ssn (139/tcp)	

Comptage par Ethereal des paquets envoyés sur poste N3-2000					
total	**TCP**	**UDP**	**ICMP**	**ARP**	**Other**
134960	122610	3118	8192	28	12
100	90	2.3	6	0	0

Commentaire : Les résultats sont quasiment identiques au rapport obtenu lors de l'étape A du test 2. Il y a 4 messages supplémentaires et une alerte a disparu. Le nombre de paquets est élevé du fait de l'utilisation de **Nmap**, mais aussi parce que le poste N3-2000 sous Windows 2000 propose beaucoup plus de services que les autres postes. Plus le nombre de services est élevé, plus le nombre de tests effectués est important, ce qui explique le nombre élevé de paquets.

4.3.2 Analyse de l'audit

Les résultats des tests ont indiqué des failles de type **débordement de tampon** qui permettent à un attaquant d'exécuter du code à distance sur le poste, en particulier au niveau du protocole **SMTP**. D'autres failles permettent l'exécution de commandes, de codes et l'élévation des privilèges de l'attaquant. Nous n'allons pas analyser davantage ces failles détectées. **Nessus** a donc détecté un ensemble de failles qui rendent le réseau vulnérable à des attaques. Il indique dans les rapports pourquoi ces failles représentent des vulnérabilités, la façon d'y remédier soit en indiquant les patchs à télécharger qui résolvent le problème soit les nouvelles versions des produits qui ne présentent plus ces vulnérabilités. Il indique également des références **CVE** qui permettent aux administrateurs d'aller chercher davantage d'informations correspondant aux problèmes rencontrés.

Nous avons pu constater grâce aux tests effectués que les attaques dites dangereuses telles que les attaques de type **DoS** affinent les résultats obtenus. Ils permettent de réduire les **faux positifs** mais ils peuvent faire tomber les services qui ne seront plus disponibles pour la totalité du test. Dans le cas de l'utilisation d'attaques dangereuses, il est nécessaire de cibler ses choix sur des attaques particulières afin d'être sûr qu'elles puissent se faire dans des conditions normales. Il est sans doute nécessaire de prévoir des tests en plusieurs étapes en décomposant par exemple l'ensemble des attaques dangereuses en sous groupes.

Au vu des temps d'inspection des postes, **Nessus** ne semble faire que les tests utiles en fonction du balayage des ports qu'il a effectué. En effet, dans sa démarche de détection, **Nessus** commence par faire un balayage des ports en cherchant à identifier les protocoles utilisés. Puis, il effectue les tests sur ces ports correspondant aux protocoles découverts. Le fait de choisir des plages de balayage importantes augmente considérablement la durée de l'audit. Cette option s'avère très utile si nous voulons nous concentrer sur une plage de ports précise afin d'y découvrir un service atypique. La durée de l'audit est proportionnelle au nombre de ports ouverts et services présents, ce qui paraît tout à fait logique. Nous pouvons noter que la phase qui prend le plus de temps est celle correspondant au balayage des ports.
L'utilisation de **Nmap,** en plus des scanners de port, utilisé par défaut par **Nessus** affine légèrement les résultats obtenus. Mais, l'infrastructure réseau est davantage sollicitée.

Les rapports obtenus lors de cette série de test étant trop volumineux, ils n'ont pas été mis en annexe. Des exemples de rapport sont disponibles sur le site de **Nessus** http://www.nessus.org.

4.4 Limites d'utilisation

Les limites de l'utilisation de **Nessus** apparaissent avec des réseaux de tailles importantes. En effet, dans le cas des tests réalisés sur un petit réseau de trois postes qui ne propose pas énormément de services, les rapports d'audit atteignaient jusqu'à 16 pages. Dans un véritable réseau d'entreprise, où le nombre de machines présentes sur le réseau peut dépasser le millier, la question de l'exploitation convenable des résultats obtenus se pose. Des rapports volumineux peuvent vite devenir inexploitables. Une solution possible consisterait à faire appel à des personnes expérimentées dans le domaine de la sécurité des réseaux avant d'utiliser des outils tel que **Nessus**. Ce type d'outil ne remplace pas un audit de sécurité et encore moins une stratégie de sécurisation de réseaux informatique. **Nessus** peut être un des moyens de parvenir à sécuriser un réseau informatique mais pas une solution à elle seule, au même titre que les tests d'intrusion.

Dans un réseau important, il faut commencer par définir un plan d'actions afin de sécuriser le réseau. Ceci suppose un minimum de connaissances dans le domaine. Par exemple, il est possible de définir une politique de sécurisation par rapport à des parcs homogènes de machines tels que des postes utilisateur ou des serveurs d'applications. Il est donc nécessaire de faire un inventaire des services nécessaires au bon fonctionnement d'une entreprise. La sécurisation est une chose mais faut il encore donner la possibilité aux employés d'une entreprise d'avoir les moyens d'effectuer leur travail de façon correcte. Cette démarche s'accompagne en général d'une réflexion sur la façon dont est mis en place le réseau. Ensuite, nous pouvons utiliser des scanners de vulnérabilité afin de mettre en évidence les failles existantes.

Une autre limite des scanners en général est liée à la validité de la détection qu'ils effectuent. En effet, le scanner peut détecter des failles de vulnérabilités qui n'en sont pas. Elles sont appelées des **faux positifs**. La première étape qui suit un audit de vulnérabilité est l'élimination de ces **faux positifs** qui peuvent faire perdre beaucoup de temps pour rien.

D'autre part, les scanners ne détectent pas toutes les failles. En particulier, ils ne simulent pas toutes les nouvelles failles. Les nouvelles menaces ne sont pas prises en compte de façon immédiate. Leurs identifications et la mise au point des tests afin de les détecter nécessitent un certain temps pendant lequel des attaquants peuvent les exploiter.

5. Nessus par rapport aux autres scanners

5.1 Protocole des tests

Pour mieux situer l'offre de **Nessus** par rapport aux autres produits disponibles sur le marché, un certain nombre de tests simples ont été menés avec d'autres systèmes de détection de vulnérabilité. Ces logiciels ne sont autre que **Saint**, un logiciel commercial fonctionnant sur les plates-formes de type **UNIX**, et **Internet Scanner**, disponible sur les systèmes de type Windows NT / 2000. A la fin de ces tests, nous nous sommes enfin intéressés à **NeWT**, la version Windows de Nessus .

Test 4 : Audit de vulnérabilité en utilisant Saint

Nous allons effectuer des Tests de vulnérabilité à l'aide de **Saint**, dans un premier temps sans attaques dangereuses (Etape A) puis avec des attaques dangereuses de type **DoS** (Etape B).

Test 5 : Audit de vulnérabilité en utilisant Internet Scanner

Nous allons effectuer des Tests vulnérabilité à l'aide d'**Internet Scanner**, dans un premier temps sans attaques dangereuses (Etape A) puis avec des attaques dangereuses de type **DoS** (Etape B).

Test 6 : Audit de vulnérabilité en utilisant NeWT

Nous allons effectuer des Tests vulnérabilité à l'aide de la version Windows de **Nessus NeWT**, dans un premier temps sans attaques dangereuses (Etape A) puis avec des attaques dangereuses de type **DoS** (Etape B).

5.2 Déploiement

5.2.1 Topologie

Le réseau utilisé est le même que celui qui a permis d'effectuer les tests correspondant à **Nessus**. Les différents scanners ont été installés comme indiqué dans le tableau suivant :

Ordinateur	**Poste N1-Linux**	**Poste N2-98**	**Poste N3-2000**
Adresse IP	168.192.1.2	168.192.1.3	168.192.1.4
OS	Fedora core 1B	Windows 98 SE	Windows 2000 serveur
Scanner de vulnérabilité	**Nessus** 2.10.a. et Saint 5.3.2		Internet Scanner 7.0 NeWT 1.5
Sniffer		Ethereal 0.10.3	

5.2.2 Paramétrage

Test 4 : Audit de vulnérabilité en utilisant Saint

Dans ce test, **Saint** a été utilisé pour mener des tests sur les postes N2-98 et N3-2000. Cette version de **Saint** est une version d'évaluation du produit, elle n'autorise à faire des tests de vulnérabilité que sur deux IP distinctes que nous indiquons avant d'effectuer le téléchargement sur le site du fabricant. Nous allons effectuer ce test en deux étapes :
A. Audit avec **Saint** sans attaques dangereuses
B. Audit avec **Saint** avec attaques dangereuses (**DoS**)

Test 5 : Audit de vulnérabilité en utilisant Internet Scanner

Les tests de vulnérabilités à l'aide d'**Internet Scanner** ont été effectués sur le poste N3-2000. Cette version ne permet de faire un audit de vulnérabilité que sur le poste sur lequel elle est installée (mode loopback).
A. Audit avec **Internet Scanner** sans attaques dangereuses
B. Audit avec **Internet Scanner** avec attaques dangereuses (**DoS**)

Test 6 : Audit de vulnérabilité en utilisant NewT

Cette version Windows de **Nessus** a été installée sur le poste N3-2000 sur lequel ont été menés les tests de vulnérabilité.
A. Audit avec **NewT** sans attaques dangereuses
B. Audit avec **NewT** avec attaques dangereuses (**DoS**)

5.3 Exploitation

5.3.1 Résultats

Test 4 : Audit de vulnérabilité en utilisant Saint

A. Audit avec Saint sans attaques dangereuses
Ordinateur cible : Poste N2-98 / Poste N3-2000
Durée du test : 3 minutes

Saint utilise un système de couleur pour identifier les différents type de vulnérabilité.

- ⇨ Le rouge désigne les vulnérabilités de type *Critical* qui permettent d'obtenir des droits en lecture/écriture, d'exécuter des commandes sur la cible ou des attaques de type **DoS**.

- ⇨ Le jaune correspond aux vulnérabilités de *Area of concern* qui correspondent à l'élévation de privilèges et à l'obtention d'informations correspondantes aux différents postes.

- ⇨ Le marron est utilisé pour les vulnérabilités de type *Potential* qui demandent plus d'investigations.

- ⇨ Enfin, le vert correspond au nombre de *services*. C'est un comptage qui n'implique pas que le service soit vulnérable ou non.

Nous pouvons noter que ce regroupement différent légèrement de celui que présente **Nessus**.

Informations obtenues :

	Critical (rouge)	**Concern (jaune)**	**Potential (marron)**	**Services (vert)**
Total	3	1	12	53
Poste N2-98	1	0	0	
Poste N3-2000	2	1	12	

Services problématiques découverts sur	**Libellé**
Poste N2-98	Partage de fichier permettant des connexions non désirées
Poste N3-2000	Folder traversal **IIS** (possibilité de crash serveur)
Poste N3-2000	Attaque **DoS** sur le service SMTP ou d'utilisation du service

Comptage par Ethereal des paquets envoyés Sur poste N2-98					
total	**TCP**	**UDP**	**ICMP**	**ARP**	**Other**
66717	54698	6290	5721		
100	82	9.4	8.6		

Commentaire : Le temps pris par le balayage est beaucoup plus rapide qu'avec **Nessus**. **Saint** ne met cependant pas autant de vulnérabilités en évidence que **Nessus**. Il indique directement en clair un libellé correspondant à la vulnérabilité. On sait ainsi de quoi il retourne de façon plus rapide. **Nessus** lui indique le service et le port présentant l'anomalie, ce que ne fait pas **SAINT**. De plus, il y un plus grand nombre de références de type **CVE** et autres tel que **IAVA**. Les vulnérabilités ne sont pas qualifiées de la même façon. Nous aborderons le sujet un peu plus loin.

B. Audit avec Saint avec attaques dangereuses (*DoS*)

Ordinateur cible : Poste N2-98 / Poste N3-2000
Durée du test : 5 minutes

Informations obtenues :

	critical	**concern**	**potential**	**services**
Total	4	1	12	52
Poste N2-98	1	0	0	
Poste N3-2000	3	1	12	

Services problématiques découverts sur	**libellé**
Poste N2-98	Partage de fichier permettant des connexions non désirées
Poste N3-2000	Folder traversal **IIS** (possibilité de crash serveur)
Poste N3-2000	Attaque **DoS** sur le service SMTP ou d'utilisation du service
Poste N3-2000	**Buffer** owerflow dans **IIS** (possibilité de crash serveur)

Comptage par Ethereal des paquets envoyés Sur poste N2-98					
Total	**TCP**	**UDP**	**ICMP**	**ARP**	**Other**
69059	56684	6541	5826	8	0
100	82.1	9.5	8.4	0	0

Commentaire : Six fois plus de paquets ont été envoyés à la cible. **Saint** charge davantage le réseau que ne le fait **Nessus**. Les résultats obtenus sont sensiblement les même que précédemment. Les vulnérabilités que présente **IIS** ont été affinées. Une attaque de type critique a été identifiée. Nous constatons que les attaques de type dangereuse affinent les résultats de l'audit.

Test 5 : Audit de vulnérabilité en utilisant Internet Scanner

A. Audit avec Internet Scanner sans attaques dangereuses
Ordinateur cible : Poste N3-2000
Durée du test : 1 heure 15 minutes

Informations obtenues :

Risque	**fort**	**moyen**	**faible**
Vulnérabilités	40	74	74

Services problématiques découverts sur poste N3-2000		
ftp (21/tcp)	Http (80/tcp)	Microsoft-ds (445/tcp)
Smtp (25/tcp)	Snmp (161/udp)	

Commentaire : Le nombre de vulnérabilités indiqué par **Internet Scanner** est très élevé. Toutes les vulnérabilités ne renvoient pas à des liens **CVE**. Il semble y avoir beaucoup de redondance dans les informations indiquées. Il est assez difficile de s'y retrouver. **Nessus** est globalement plus convivial et intuitif que ne l'est **Internet Scanner**.

B. Audit avec Internet Scanner avec attaques dangereuses (*DoS*)

Ordinateur cible : Poste N3-2000
Durée du test : 1 heure 15 minutes

Informations obtenues :

Risque	**fort**	**moyen**	**faible**
Vulnérabilités	40	74	75

Services problématiques découverts sur poste N3-2000		
ftp 21/tcp	http 80/tcp	Microsoft-ds 445/tcp
Smtp 25/tcp		

Commentaire : Le service **SNMP** n'apparaît plus. Le temps d'exécution de l'audit est identique que nous utilisions ou pas les attaques dangereuses. Une détection de vulnérabilité prend beaucoup plus de temps avec **Internet Scanner** qu'avec **Nessus** et ne donne pas des résultats aussi clairs.

Test 6 : Audit de vulnérabilité en utilisant NewT

A. Audit avec NewT sans attaques dangereuses
Ordinateur cible : Poste N3-2000
Durée du test : 5 minutes

Informations obtenues :

Ports ouverts	Trous de sécurité	informations	notes
36	12	33	68

Services problématiques découverts sur poste N3-2000	
Smtp 25/tcp	**Epmap** 135/tcp
http 80/tcp	Snmp 161/udp

Commentaire : **NeWT** est moins paramétrable que **Nessus**. Il est beaucoup plus simple d'aspect que **Nessus**. Il donne les même résultats que **Nessus** au niveau de notre test avec davantage de messages d'information.

B. Audit avec NeWT avec attaques dangereuses (DoS)
Ordinateur cible : Poste N3-2000
Durée du test : 7 minutes
Informations obtenues :

Ports ouverts	Trous de sécurité	informations	notes
36	6	24	63

Services problématiques découverts sur poste N3-2000		
Smtp (25/tcp)	**Epmap** (135/tcp)	Microsoft-ds (445/tcp)
http (80/tcp)	Snmp (161/udp)	Unknown (7368/tcp)

Commentaire : Deux services supplémentaires apparaissent comme ayant des failles de vulnérabilité. Les attaques de type **DoS** affinent là encore les résultats.

5.3.2 Analyse de l'audit

Les résultats obtenus à l'aide des différents outils sont similaires aux résultats obtenus à l'aide de **Nessus** (voir la partie 4 « Les tests réalisés avec **Nessus** » de ce travail). Ils ne sont pas exactement les mêmes et il n'est pas aisé de comparer de façon minutieuse les différents rapports. Les points communs que nous pouvons prendre pour se repérer dans les rapports sont les services et numéros de ports associés ainsi que les références **CVE** si elles existent. Le nombre de vulnérabilités en lui même ne correspond pas à un critère auquel nous pouvons vraiment nous fier. En effet, la notion de vulnérabilité n'est pas tout à fait la même d'un outil

à un autre. Les outils qui identifient le plus de services présentant des vulnérabilités sont **Nessus** et **Internet Scanner**.

Saint révèle moins de failles mais propose des rapports plus conviviaux. Il utilise des couleurs pour identifier les différents types de vulnérabilité. Ces repères visuels permettent de se faire une idée plus rapide de la situation qu'avec **Nessus**. La compréhension des vulnérabilités est, de plus, améliorée par l'affichage de libellés en clair, là où **Nessus** indiquait un nom du service et un numéro de port. Dans le cas de **Nessus**, il est nécessaire d'aller voir les commentaires associés pour comprendre de quoi il en retourne. Cette démarche est certes moins rapide mais a l'avantage d'être plus complète et plus précise. Les rapports obtenus à l'aide de **Saint** sont plus riches et leurs présentations sont plus travaillées. Les références aux des articles dans lesquels il est possible d'obtenir davantage d'information sont beaucoup plus nombreuses. Les audits peuvent être programmés de façon automatique par l'utilisateur. Il est possible de suivre l'évolution des découvertes des failles en temps réel. Avant de lancer un audit, l'utilisateur doit choisir un ensemble de tests à exécuter. Cet ensemble de tests peut être personnalisé. Dans **Nessus**, cela correspond à faire un choix de tests dans les différentes familles proposées. **Saint** propose un ensemble particulier correspondant aux top 20 des vulnérabilités du **SANS.** Ces vulnérabilités sont les 20 menaces les plus critiques pour la sécurité Internet.

Internet Scanner propose les mêmes options. Les regroupements de tests s'appellent des politiques. La possibilité de suivre l'évolution de l'audit est également disponible. Un catalogue de vulnérabilités est présent. Il permet la recherche d'informations correspondantes aux diverses vulnérabilités. L'exploitation des informations qu'il fournit est assez difficile. Il n'est pas évident de s'y retrouver. La réalisation des rapports n'est pas très aisée. Au premier abord, cet outil paraît assez austère. Il dresse cependant une liste exhaustive des informations relatives aux failles de vulnérabilités détectées. Sur ces points, l'ergonomie proposée par **Nessus** est plus intuitive.

Enfin, **NeWT**, la version Windows de **Nessus**, propose une interface plus simple et plus conviviale que son homologue **Open Source**. Les diverses options restent similaires sans pour autant aller aussi loin. Les logiciels tiers tels que **Nitko** et **Hydra** n'apparaissent plus dans les paramétrages. Les rapports sont moins développés que ceux de **Nessus**. **NeWT** propose également une notion de politique de test. Ces différentes politiques de test correspondent à divers regroupements des familles de tests proposées par **Nessus**.

5.4 Synthèse comparative

Nous allons dans cette partie essayer de synthétiser le positionnement de l'outil **Nessus** par rapport à d'autres offres disponibles sur le marché des scanners de vulnérabilités. Nous ne cherchons pas à faire une comparaison au sens strict du terme, c'est à dire renseigner tous les critères de comparaison pour l'ensemble des produits choisis, de façon exhaustive. Le but de notre étude est de nous faire une idée des atouts et des faiblesses de **Nessus** par rapport aux autres produits sur la base d'un certain nombre de critères.

Nessus a l'avantage d'être un projet **Open source** très bien suivi avec un noyau de développeurs sérieux. La mise à disposition des tests permettant de déceler les nouvelles vulnérabilités est très rapide, ainsi que la mise à jour du produit. **Nessus** est donc un produit bien supporté qui augure d'une bonne continuité dans le développement. Il représente une forte alternative au produit commercial dans le cas de réseaux de petite taille.

*** pas convaincu ** passable *** correct **** bien ***** Excellent ▭ non renseigné**

Critères	Nessus	NeWT	Saint	Sara	Internet Scanner	Retina
Matériel et Système						
Les plate formes d'attaque	Unix (BSD, Linux, Solaris)	Windows NT et 2000	Unix	Unix	Windows NT 2000	Windows NT 2000 XP
Les systèmes testés / attaqués	Windows Unix		Windows Unix		Windows Unix	Windows Unix (Solaris, Linux, BSD)
Hardware	Ordinateur, serveur, routers, firewall	Ordinateur, serveur, routers, firewall	Ordinateur, serveur, routers, firewall		Ordinateur, serveur, routers, firewall	Ordinateur, serveur, routers, firewall
Réseau sans fil						oui
Vulnérabilité testés						
Temps du test	***	****	****		**	
Vulnérabilités testées	2000+	2000+			1300	
Vulnérabilités Détectées	****	****	***		****	
Outils						
L'outil	****	****	****		**	**
Portée des attaques au sens adressage	libre	Verrouillé	Verrouillé avec les licences d'utilisation	Libre	Verrouillé avec les licences d'utilisation	possible
La mise à jour des vulnérabilités	Par ajout de modules	Download direct sur le site	Changement de version	Changement de version	Par ajout de module ?	
La sélection des vulnérabilités	Simple clic		Simple clic	Difficilement exploitable	Simple clic	
Type d'outil	**Open source**		payant	**Open source**	payant	payant
CVE cross référence	Oui		oui	Oui	oui	
Correction automatique de vulnérabilité sélectionnée	Non		non	Non	non	Oui (permission sur fichiers, registry setting à distance)
Possibilité d'automatisation	Oui		Oui	Oui	oui	oui
Tests personnalisés	Oui (**NASL**)		oui	oui	non	Oui (**API**)
Logiciels tiers qui peuvent être utilisés	**Nmap Nitko Hydra**			**Nmap**		
Analyse des données et support SSL	Oui	oui				oui
Rapport	***	***	****		**	
Facilité d'installation	***	****	*****	*	***	*

Le fait que **Nessus** se base sur des logiciels tiers comme **Nmap**, **Hydra** et **Nitko** pour effectuer certaines tâches spécifiques et qu'il utilise un système de **plug-in** pour l'ajout des nouveaux tests de vulnérabilités font de **Nessus** un outil très évolutif et personnalisable. Ce caractère est renforcé par le fait de pouvoir programmer ses propres tests de vulnérabilité et la présence de nombreux paramétrages.

Saint semble plus simple. Il ne détecte pas autant de services présentant des failles de vulnérabilité. Mais il a le mérite d'avoir des rapports beaucoup plus développés que ceux de **Nessus**. C'est un produit commercial qui possède un homologue **Open source** nommé **Sara**. **Sara** est supporté par les auteurs de **Saint**. Ce sont des produits qui apportent de réels plus par rapport à **Nessus** en terme de présentation et d'ergonomie.

Internet Scanner repose quant à lui sur un fort groupe de professionnels. Il est moins ergonomique que **Nessus** mais permet de découvrir autant de failles de vulnérabilité.

Rétina se distingue un peu des autres solutions en proposant une méthode nommée **CHAM** qui permet de découvrir des vulnérabilités inconnues. La validité de cette démarche est assez difficile à qualifier sans faire une étude plus poussée de cette option.

La prise en main des problèmes de sécurité, à l'aide de scanners de vulnérabilité, peut se faire en deux temps. Dans un premier temps, nous pouvons utiliser des logiciels simples tels que **Saint** ou **Sara**. Ensuite nous pouvons affiner les résultats à l'aide de **Nessus** ou d'**Internet Scanner**.

Les scanners de vulnérabilités sont des outils qui évoluent très rapidement. Une comparaison des différentes solutions n'est valable que sur du court terme.

Les évolutions de ce type de produit se font dans la même direction. Ils recherchent tous à :

- ⇨ élargir le champs des machines testées, ainsi que le nombre de vulnérabilités
- ⇨ intégrer rapidement les nouvelles failles de vulnérabilités découvertes
- ⇨ améliorer la documentation associée aux vulnérabilités et aux actions correctrices à entreprendre pour pallier ces vulnérabilités.
- ⇨ indiquer les tests dangereux qu'il faut utiliser avec précaution.

Il est assez difficile de comparer ces produits au niveau du nombre de failles qu'ils détectent. L'information la plus significative est la liste des vulnérabilités qu'ils présentent. Un nombre de vulnérabilités en soit ne signifie pas grand chose s'il n'existe pas des identifiants uniques qui permettent de qualifier les différentes vulnérabilités. D'où l'intérêt énorme des références **CVE** qui permettent de savoir si nous parlons bien des mêmes vulnérabilités.

Généralement, nous attribuons aux produits des sociétés commerciales une meilleure pérennité que les logiciels **Open source**. **Nessus** est un parfait contre exemple.

Nessus ne propose pas la meilleur ergonomie au niveau de son interface cliente et de ses rapports, mais il fournit des détections complètes, avec des temps d'analyse très corrects par rapport aux autres produits du marché. Il est évolutif, ouvert et gratuit.

6. Conclusion

L'objectif de ce travail est d'appréhender les problématiques liées aux tests d'intrusion dans les réseaux Internet. Pour cela, des tests à l'aide d'outils d'audit de vulnérabilité, en particulier de l'outil **Nessus**, ont été réalisés. Une comparaison s'appuyant sur divers critères a été menée, afin de positionner **Nessus** par rapport aux autres offres du marché.

Cependant, une comparaison en terme de nombre de vulnérabilités détectables n'est pas aussi aisée. Cela suppose en effet que toutes les vulnérabilités soient identifiées de façon unique. Ce qui est difficilement réalisable en soit, du fait que nous ne connaissons pas toutes les vulnérabilités et que les différents scanners de vulnérabilité n'ont pas tous recours à ce genre de références, tels que les liens **CVE**, de manière systématique. Il est à noter qu'il existe plusieurs bases de connaissance qui recensent les vulnérabilités et que certains scanners utilisent des identifiants propres pour désigner les vulnérabilités. Dans le cas de **Nessus,** ces identifiants propres sont appelés les **Nessus Ids.**

Les scanners de vulnérabilités sont des moyens permettant de sécuriser un réseau. Ils sont analogues à une brique imparfaite d'un ensemble plus complexe que constitue l'ensemble des outils de sécurisation des réseaux. Ils ne suffisent pas à eux seuls. Ils ne sont par exemple pas adaptés pour l'analyse de la robustesse des mots de passe. Les mots de passe constituent pourtant l'une des vulnérabilités les plus importantes présentés par les réseaux. Ils sont impuissants devant les attaques de virus et de vers.

Les scanners sont utilisés dans les tests d'intrusion afin de récolter des informations sur les ordinateurs cibles. Ils ne constituent qu'une étape d'un certain type de tests d'intrusion. Il existe d'autres types de tests d'intrusion tels que le **dial up** et l'**ingénierie sociale** qui sont tout aussi efficaces.

Les tests d'intrusion simulent le comportement d'individus hostiles qui cherchent à exploiter, par tous les moyens, toute faille que présente un système d'information. Cependant, les personnes qui mènent ces tests ne sont pas animées des mêmes préoccupations. Ils ne peuvent pas risquer de rendre indisponibles les services partagés à travers le réseau. Les pirates quant à eux ne font preuve d'aucun scrupule. De plus, les pirates sont généralement à l'initiative d'exploitation des vulnérabilités non encore découvertes. Un test d'intrusion ne permet de découvrir que les failles de vulnérabilité connues à un instant donné.

Les tests d'intrusion ne sont pas non plus suffisants. Ils rentrent dans un processus plus global qui est celui de l'audit de sécurité. Le but d'un audit de sécurité est de faire un rapport des vulnérabilités du système et des actions à mener pour y remédier. L'audit de sécurité indique donc des stratégies à adopter pour résoudre les différents problèmes. Il suppose l'intervention de personnes spécialisées dans le domaine.

Les scanners font pour leurs parts un rapport des vulnérabilités et indiquent généralement les moyens d'y remédier de façon automatisée. Ceci ne remplace pas l'intervention de spécialistes qui peuvent avoir par exemple une meilleure visibilité sur les conséquences des tests qu'ils effectuent et des actions qu'ils mènent afin de résorber les différentes failles de vulnérabilités. Ils peuvent également combler dans une certaine mesure les limites des scanners que sont les **faux positifs** et les non détections.

La sécurisation d'un réseau demande un travail permanent. Il est nécessaire de faire des mises à jour régulières et fréquentes. L'audit de sécurité n'est qu'une évaluation des failles d'un système à un instant donné. La sécurisation d'un réseau revêt au contraire un aspect cyclique. C'est une tâche répétitive qui demande une surveillance de tous les instants. Elle nécessite la mise en place de nombreux outils qui couvrent chacun des fonctionnalités différentes.

Les pare-feu permettent par exemple de filtrer des paquets provenant d'Internet. De cette façon, une machine extérieure au réseau d'entreprise ne peut plus se faire passer pour une machine appartenant au réseau.

Des outils tels que les a**nti-virus** apportent une protection contre les effets des virus connus.

Divers outils permettent d'effectuer une surveillance du réseau, tels que les outils permettant de journaliser de façon centralisée les traces de plusieurs serveurs sur une même machine dédiée. Ainsi, même si les attaquants effacent leurs traces sur les machines qu'ils utilisent, le journal centralisé lui est encore intact. Les **IDS** permettent de détecter les tentatives d'intrusion. Il est à noter que les scanners de détection de vulnérabilités mettent en place des mécanismes afin de ne pas se faire détecter par les **IDS**.

Enfin, divers protocoles de cryptage permettent de sécuriser les communications. Ce qui permet en particulier d'empêcher un attaquant de s'approprier des mots de passe en utilisant un **sniffer**. Un **sniffer** permet de visualiser le contenu des paquets qui transitent sur le réseau et ainsi de récupérer des mots de passe lorsqu'ils apparaissent en clair dans certain protocole.

Dans le cas d'un grand réseau, la contrainte la plus difficile à gérer est le volume de l'information obtenue à l'aide d'outils comme les scanners de vulnérabilité et le temps nécessaire pour les obtenir. Il est alors quasi obligatoire de définir un plan d'actions afin de sécuriser le réseau. Ceci suppose d'avoir un minimum de connaissance dans le domaine et nécessite de faire un inventaire des services nécessaires à la bonne utilisation du réseau. D'autre part, les scanners ne permettent pas à eux seuls de récolter et de traiter les informations obtenues de façon rapide. Il est alors nécessaire de s'orienter vers des solutions de détection de faille de vulnérabilité reposant sur des architectures réparties qui sont plus robustes et mieux adaptées aux réseaux conséquents.

Annexe A Glossaire

ADSL: Asymetric Digital Subscriber Line. Technologie permettant d'accéder à Internet à grande vitesse à partir d'un réseau téléphonique.
API : Application Programming Interface. Interface de programmation d'applications, contenant un ensemble de fonctions courantes de bas niveau, bien documentées, permettant de programmer des applications**.**
ARP : Adress Resolution Protocol. Protocole de résolution d'adresse. Processus des réseaux IP permettant d'obtenir l'**adresse IP** à partir de l'**adresse MAC**.
ASCII : American Standard Code for Information Interchange. Code standard américain pour l'échange d'information qui traduit en nombre les caractères de l'alphabet et autres.
Adresse IP : identifie le réseau et la station sur un réseau **TCP/IP**. L'adresse se compose de 4 octets, séparés par un point. Selon, que l'adresse est de classe A, B ou C, 1,2,3 octets désigne le réseau et 3, 2, 1 octets désigne le poste.
Adresse MAC : Adresse identifiant un élément actif sur un réseau. L'identifiant est constitué de données relatives au fabricant.
Anti-virus : utilitaire capable de rechercher et d'éliminer les virus informatiques.

Backdoors : Portes dérobées. Programmes usurpateurs qui détournent les fonctionnalités systèmes dans le but d'ouvrir des accès utiles aux pirates pour contrôler à distance les machines ciblées. Ces programmes sont généralement installés par l'intermédiaire d'un autre programme.
BD : Base de données, Data Base. Outils permettant de stocker, gérer et consulter des informations.
Buffer : tampon. Aire de stockage intermédiaire associée aux Entrées/Sorties, fonctionnant comme une file d'attente et jouant le rôle d'un amortisseur entre deux éléments d'une machine qui tournent à des vitesses très différentes.
BugTraq : site spécialisé dans le domaine de la sécurité informatique
Brute-force : Méthode d'analyse de chiffrement dans laquelle toutes les clés possibles sont systématiquement essayées. Elle est aussi appelée « recherche exhaustive».

CIDR : Classless Inter-Domain Routing. Autorise la création de réseaux de petites tailles et le routage dynamique des données entre eux. La notation **CIDR** est formée d'un identifiant d'un ordinateur sur quatre octets / le nombre de bits du masque du réseau.
CERT : Centre for Emergency and Response Team. Institut de génie logiciel de l'Université de Carnegie Mellon (Pittsburgh/USA) spécialisé dans la sécurité informatique. Une mine de renseignements sur les dernières failles détectées et les solutions de protection.
CGI : Programme informatique, écrit en langage script, permettant de réaliser des pages dynamiques
CHAM : Common Hacker Attack Methods. Il s'agit d'un testeur de **débordement de tampon** générique sur les protocoles FTP, SMTP, HTTP.
CISCO : Fabricant de matériel réseau
CVE : Common Vulnerabilities and Exposures. Liste de noms standardisés pour les vulnérabilités.
Certificat : Fichier chiffré fourni par un organisme tiers permettant d'identifier formellement les différents acteurs impliqués dans une transaction.
Cookies : petits paquets de données qui renferment des renseignements sur l'ordinateur d'un utilisateur.

Classe B : correspond à la classe d'un réseau. 2 octets désignent le réseau. 2 octets désignent le poste

DB : Data Base, Base de données. Outils permettant de stocker, gérer et consulter des informations.
DNS : Domain Name Server / Domain Name System. Un serveur de noms de domaine a pour fonction de traduire les noms symboliques des ordinateurs connectés au réseau en adresses numériques "IP", de manière à permettre leur reconnaissance et donc l'établissement d'une communication.
DoS : attaques dont le but est de rendre indisponible des services ou de faire tomber le serveur proposant ces services.
Démon : Daemon. Disk And Execution MONitor. Programme réalisant des tâches de fond du système sous Linux par exemple.
Débordement de tampon : Buffer overflow. Technique d'attaque consistant à envoyer dans un **buffer** plus d'informations qu'il ne peut en contenir, occasionnant un dysfonctionnement qui conduit un système mal configuré à donner la main au pirate avec un maximum de droit.
Dial up : Connexion à un réseau par l'intermédiaire d'une ligne de téléphone
Download : Téléchargement d'un fichier depuis un serveur vers le poste.
Dragon : logiciel de détection d'intrusion
Dsniff : suite d'outils d'audit de réseau et de test de pénétration.

EPMAP : End Point MAPper. Un mapper est une représentation de la localisation d'un ensemble de données en vue de faciliter l'accès. correspond à un service.
Ethernet : Norme de protocole de réseau local relativement puissante et très répandue, inventée en 1970 au PARC de Xerox par Bob Metcalfe qui l'a décrite en 1974 dans sa thèse de Doctorat de Physique, puis repris par DEC, Intel et Xerox, normalisée par l'ISO et l'IEEE avec le numéro 802.3.
Ethereal : analyseur gratuit de protocole réseau pour Unix et Windows.
Exploits : Ce terme anglais désigne les trous de sécurité que les pirates cherchent à exploiter. Pour ce faire, les pirates utilisent souvent les nombreux outils développés par les plus "brillants" d'entre eux (librement téléchargeables) ou tirent profit des outils d'évaluation aux vulnérabilités proposés prioritairement aux administrateurs systèmes.

FTP : File Transfert Protocol. Ce Protocole de transfert de fichier est une méthode standard permettant de transférer des fichiers sur l'Internet par exemple.
Faux positifs : lorsqu'un scanner découvre une vulnérabilité alors qu'il n'y en a pas, on parle dans ce cas de Faux positifs.
Finger : Programme permettant d'avoir des informations sur des personnes travaillant sur un système Unix. Si ce système est accessible par l'Internet, ces informations peuvent y être disponibles. On trouve en général le nom, le temps de connexion de l'utilisateur, son terminal.
Firewall : Pare-feu. Système situé entre le réseau interne de l'entreprise et le réseau externe, dont la tâche est de contrôler aussi bien les communications entrantes que sortantes, dans le but de sécuriser les échanges d'informations. On confie principalement au **firewall** le soin de filtrer les paquets et traiter les services **proxy**.
FreeBSD : Système d'exploitation Unix libre pour PC.

GPL : General Public Licence. Il s'agit d'un modèle de licence pour logiciel libre proposé en 1991 par la Free Software Foundation.

HTTP : HyperText Transfert Protocol. Ce protocole est le mode de communication utilisé sur le Web entre le logiciel client (navigateur) et le serveur (celui qui fournit la page). A chaque fois qu'un utilisateur demande l'accès à une page, une requête http est envoyée au serveur qui renvoie le document correspondant.
HTTPS : HyperText Transfer Protocol Secure. Protocole de transmission issu de Netscape lié à une connexion par **socket** sécurisée.
Hackers : Individu possédant des connaissances très pointues sur le matériel informatique, les systèmes d'exploitation ou encore les réseaux. Contrairement au pirate informatique (voir "Cracker") auquel il est souvent assimilé, le Hacker n'a pas la volonté de nuire mais plutôt d'améliorer des technologies existantes, y compris dans le domaine de la sécurité.
Hub : Dispositif permettant de réunir les données de plusieurs lignes à faible débit pour les transmettre sur une seule ligne à haut débit ou, inversement, de scinder le trafic d'une grosse ligne sur plusieurs petites. C'est un type de concentrateur servant de Nœud d'un réseau **Ethernet** RJ45. Dans le cas du **hub**, toutes les informations sont envoyées vers tous les PC.
Hydra : logiciel spécialisé dans les attaques **brute-force**.

IANA : Internet Assigned Numbers Authority. Organisme qui a fixé les numéros des ports bien connus (Wellknown port) pour un service donné.
ICAT : Base de données de vulnérabilités.
ICMP : Internet Control Message Protocol. Protocole (public) de contrôle utilisé par les **routeurs** pour l'acheminement des paquets Internet. Les pirates l'emploient pour créer des problèmes de transmission qui, détectés et traités par les **routeurs**, aboutiront à une surcharge réseau parfois accompagnée de dénis de service.
IDS : Intrusion Detection System. Terme générique faisant référence aux équipements ou logiciels chargés de détecter des intrusions. On distingue deux types d'**IDS**: Les **NIDS** (Network Intrusion Detection Systems) pour les réseaux et les HIDS (Host-based Intrusion Detection Systems) pour les serveurs.
IIS : Internet Information Server. Serveur Web de Microsoft.
IMAP : Internet Mail Access Protocol. Protocole de gestion de messagerie permettant de stocker le courrier sur le serveur et pas sur le client.
IP : Internet Protocol. Protocole de transmission de l'Internet, décrit aussi les adresses du réseau.
ISS : Internet Security System
Ingénierie sociale : intrusion dans un système d'information exploitant les interactions sociales
Internet Scanner : scanner de vulnérabilité de la société **IIS**
IP spoofing : technique consistent à usurper une **adresse IP** afin de se faire passer pour un autre.
Internet : Réseau à échelle mondiale. Il est composé d'un grand nombre de réseaux internationaux, régionaux et locaux interconnectés entre eux. Tous utilisent les protocoles de transmissions de la famille **TCP/IP**.

John the ripper : permet de décrypter des mots de passe disponible sur diverse plate-formes.

LAN : Local Area Network. Réseaux locaux. Réseau situé dans une zone réduite ou dans un environnement commun, tels qu'un immeuble ou un bloc d'immeubles.
LATEX : Ensemble célèbre de macros pour le langage de formatage de texte TeX. TeX est un format de composition de texte très précis surtout utilisé pour les documents scientifiques.
Linux : Système d'exploitation Unix distribué sous forme de logiciel libre.

MySQL : Système de base de données entièrement **open source** compatible SQL. Installations GNU/Linux et Win32 disponibles.

NASL : Nessus Attack Scripting Language. Permet de programmer des **plug-in** utilisables par **Nessus**.
NFS : Network File System. C'est un SGF de réseau défini par un protocole sans connexion, présenté par **Sun** en 1985 pour ses stations sans disque. Un SGF est Système de Gestion de Fichiers qui définit, par exemple, la structure interne de l'arborescence, les formats d'enregistrements, le découpage des disques, les méta données sur les fichiers...
NIDS : Network Intrusion Detection System. Système de détection d'intrusion dans un réseau.
NIS : Network Information Services. Services d'information sur le réseau. Services donnant accès à des bases de données de réseau fournissant par exemples des **adresses IP**, **Ethernet** des mots de passe ou des noms de serveur.
NNTP : Network News Transfer Protocol. Protocole de transfert des News
Nessus : scanner vérifiant la sécurité d'un ensemble de machines.
Nessus ID : identifiant que **Nessus** associe aux différentes vulnérabilités.
Netbios : NETwork Basic Input-Output System. BIOS dédié aux réseaux conçus par IBM dans les années 1980 et devenu un standard.
Netcat : utilitaire simple qui permet de lire et écrire des données à travers une connexion réseau en utilisant TCP ou UDP.
Netware : Système d'exploitation de réseau de Novell, comprenant de nombreux utilitaires, et très utilisé sur les **LAN** de PC.
NeWT : version Windows de **Nessus**.
Nitko : scanner spécialisé dans les **CGI.**
Nmap : scanner de port.
Null : absence de valeur

OSI : Open Systems Interconnection. Interconnexion de système ouvert.
Open source : Famille regroupant des logiciels régis par une licence dite "libre".

PERL : Pratical Extraction and Report Language. Langage interprété optimisé pour le traitement du texte**.**
PHP : Pre Hypertext Processor. Un langage de programmation orienté Web et exécuté côté serveur. Il permet la génération de sorties HTML en fonction de requêtes effectuées par un ordinateur distant. Il s'interface facilement avec des bases de données telles que **MySQL**, PostgreSQL, etc. et bien entendu en module de logiciels serveurs tels que Apache, Xitami ou même **IIS**. **PHP** est un langage **open source**.
Ping : Packet INternet Groper. « Faire un **PING** » consiste à envoyer une requête **ICMP** à un serveur. S'il répond, c'est qu'on a des chances de pouvoir l'atteindre. Sinon, c'est qu'il est en panne ou inaccessible.
Plug-in : extension à une application qui vient se loger dans l'application elle-même. Une fois installée, on peut utiliser le **plug-in** de façon tout à fait transparente.
Protocole : Terme définissant les règles de communication entre plusieurs ordinateurs sur un réseau.
Port : Canal de communication.
Proxy : Service (au sens logiciel du terme) qui relaye la communication entre un poste client et un serveur. Il est placé entre le poste client et le firewall et entre le firewall et le serveur. Ainsi, le **proxy** est une sorte d'intermédiaire qui évite au serveur d'être directement "attaqué" par les requêtes du poste client.

RPC : Remote Procedure Call. Technique utilisée dans le modèle client-serveur. Le client appelle des procédures qui sont exécutées sur un ordinateur distant grâce à un serveur d'applications. Le protocole **RPC** gère les interactions entre le client et le serveur.
Registry : base de registres. La **registry** en elle-même est la base de donnée du système.
Retina : scanner de vulnérabilité de la société eEye.
Rusers : Commande qui permet de trouver le nom de compte valide.
Routeur : Equipement électronique dont le but est d'assurer la communication entre plusieurs réseaux. Il joue le rôle de serveur en gérant les échanges entre ces réseaux.

SANS : Institut qui rassemble les informations de plusieurs sources telles que **CERT, Bug Traq.**
SMB : Server Message Block. Protocole de Microsoft et d'Intel fonctionnant sur **NetBIOS** et permettant le partage de ressources (disques et imprimantes) à travers un réseau publié en 1987.
SMTP : Simple Mail Transfert Protocole. Protocole de la famille **TCP/IP** utilisé pour le transfert de courrier électronique.
SNMP : Simple Network Management Protocol. Il sert à administrer localement ou à distance les réseaux **TCP/IP**.
SSH : Secure SHell. Shell permettant de se connecter de façon sécurisée sur une machine distante et d'y exécuter des programmes, toujours de façon sécurisée
SSL : Secure Socket Layer. **sockets** sécurisées
SUN : Microsystems Computers Corp. Entreprise créée par des **hackers** et fabricants des stations de travail, une référence en matière de calcul mathématique et de réseau de micros.
SAINT : Security Administrator's Integrated Network Tool. Outil d'audit réseau se concentrant sur la sécurité du système. C'est un descendant de **SATAN**
SARA : Security Auditor Research Assistant. Scanner de vulnérabilité **Open source**.
SATAN : Security Administrator Tool for Analysing Network. Outils (mis au point par Dan Farmer) permettant de tester la sécurité d'un réseau en simulant une attaque de l'extérieur.
SecurityFocus : site traitant de la sécurité informatique
Session : période de temps ininterrompue au cours de laquelle un client est connecté à un serveur.
Siphon : permet de collecter de l'information sur la cartographie du réseau.
Sniffer : En français, sniffeur, déformation du nom anglais, autant dire renifleur. Sorte de sonde que l'on place sur un réseau pour l'écouter, et en particulier récupérer à la volée des informations sensibles, comme des mots de passe, sans que les utilisateurs ou les administrateurs du réseau ne s'en rendent compte.
Source routing : routage à la source. Permet de spécifier le chemin que vont emprunter les paquets sur le réseau.
Spiffy HTML : document HMTL comportant des camemberts et des graphes.
Switch : c'est un commutateur. A la différence d'un **Hub**, un **switch** reconnaît les différents PCs connectés sur le réseau. Il n'envoie l'information qu'au PC destinataire.
Socket : Norme de mode de communication sur réseaux, mise au point à Berkeley, qui permet à une application de dialoguer avec un protocole.

TCP/IP : Transmission Control Protocol / Internet Protocol. Les deux protocoles de communication qui forment les fondements de l'Internet
TLS : Transport Layer Security. Protocole destiné à assurer une meilleure confidentialité des communications Internet, il est issu d'un projet de l'IETF basé sur la version 3 de **SSL**.
TNS : Tenable Network Security

TCPdump : un **sniffer** classique pour le contrôle et l'acquisition de donnée.
Telnet : Utilitaire de connectivité **TCP/IP** permettant la connexion en mode terminal (ouverture de **session**) entre ordinateurs hétérogènes.
Traceroute : C'est un utilitaire permettant de déterminer le trajet emprunté par vos paquets IP sur Internet. Il est disponible sous Linux ou Windows avec, par exemple, la fonction sous **dos** "tracert".

UDP : User Datagram Protocol Protocole réseau niveau 4 (Transport) servant au transfert rapide de données. L'échange étant réalisé sans aucun contrôle, il ouvre la porte aux attaques de type spoofing.
Unix : De l'anglais "Uniplexed Information and Computer Service", Unix est un système d'exploitation multitâches et Multi-Utilisateurs très performant développé à la fin des années 60. Le dérivé le plus connu d'UNIX est Linux.

Vers : Petit programme qui se "ballade" dans la mémoire vive (RAM) des ordinateurs, détruisant tout sur son passage. Les vers se répandent en utilisant les réseaux. Sans qu'on puisse vraiment les classer comme des virus (leur attaque n'est pas ciblée), les vers occasionnent généralement "seulement" des dysfonctionnements et blocages des machines qu'ils parasitent.
Virus : Au sens large du terme, on qualifie généralement de virus tout programme capable de se reproduire (techniquement, se recopier) lui-même et d'endommager des données informatiques. On les classe en plusieurs catégories, principalement : parasite, compagnon, amorce, multiformes, résidant en mémoire ou non, furtifs, polymorphes, réseau et flibustier. Pour plus de détails reportez-vous aux définitions correspondantes.

Web : Le Web ou World Wide Web, (aussi appelé la "toile") est l'une des multiples dénominations de l'Internet.
Wifi : Contraction de « WIreless FIdelity ». utilisé pour désigner les réseaux sans fil
War dialing : consiste à identifier des numéros de téléphone appartenant à un système d'information permettant de s'introduire à l'intérieur du réseau
Whisker : scanner de vulnérabilité spécialisé dans les **CGI.**
Whois : Un Whois ("Qui est-ce ?") permet de connaître toutes les informations de propriété liées à un nom de domaine enregistré auprès du registre de l'Internic : contact administratif, contact technique, contact de facturation et les dates d'enregistrement du nom de domaine. Internic signifie INTERnet Network Information Center. C'est l'organisme se chargeant, entre autres, de distribuer les adresses Internet.
Windows: est un système d'exploitation. C'est à dire un programme assurant la gestion de l'ordinateur et de ses composants.

Annexe B Bibliographie / Référence Internet

[1] LesNouvelles. *Mariage surprise dans l'audit de vulnérabilités.* Disponible sur http://www.yacapa.com/article-993.html (consulté le 01 Avril 2004)

[2] Nessus.org. *Nessus.* Disponible sur http://www.nessus.org/doc/datasheet.pdf. (consulté le 23 mars 2004)

[3] GORIN Jean-denis. *Les tests d'intrusion* JDGorin@Computer.Org

[4] ANDERSON Harry. *Nessus, Part 3: Analysing Reports* . Disponible sur http://www.securityfocus.com/infocus/1741 (consulté le 01 Avril 2004)

[5] Anonyme. Sécurité maximale des systèmes et réseaux.4e édition. Paris, France : CampusPress, 2003. ISBN 2-7440-1549-0

[6] BOUTHERIN B., DELAUNAY B. *Sécuriser un réseau Linux.* Paris, France : Eyrolles, 2003, 155p. (collection Cahiers de l'Admin) ISBN 2-212-11245-9

[7] RENARDIAS Vincent. *Nessus.* redhat magazine, 2004, n°2, pp. 12-17.

[8] GomoR@gomor.org. *Test d'intrusion, méthodologie.* Disponible sur http://www.gomor.org/Securite/test_d_intrusion.html (consulté le 1 Avril 2004)

[9] The MITRE Corporation. *About **CVE***. Disponible sur http://cve.mitre.org/about/ (consulté le 29 Mars 2004)

[10] Comité consultatif sur les technologies de l'information de l'ICCA. *Test d'intrusion—Outil d'appréciation des risques pour la sécurité de l'information.* Canada : L'Institut Canadien des Comptables Agrées, 2003. Disponible sur www.icca.ca/ccti (consulté le 01 Avril 2004).

[11] Trustonme.net. *Nessus.* Disponible sur http://www.trustonme.net/didactels/?rub=201 (consulté le 24 mars 2004)

[12] E-Atlantide.com. *Scanner de vulnérabilités.* Disponible sur http://www.e-atlantide.com/securite/scanner_vulnerabilite/index.htm (consulté le 02 Avril 2004)

[13] Magistrat. *Dossier / News Blocus : Identifier des failles potentielles dans son réseau : méthode.* Disponible sur http://www.blocuszone.com/modules/news/article.php?storyid=589 (consulté le 02 Avril 2004)

[14] Marie-claude QUIDOC. *Comparatif des produits de simulation d'intrusions*. Disponible sur http://www.urec.cnrs.fr/securite/articles/si_comparatif.pdf (consulté le 08 Avril 2004)

[15] Eric UNER. *Simple Network Vulnerability Testing*. Disponible sur http://www.naspa.com/PDF/2003/0903/T0309001.pdf (consulté le 09 Avril 2004)

[16] SecurityWizardry UK. *Network Vulnerability Scanners.* Disponible sur http://www.networkintrusion.co.uk/N_scan.htm (consulté le 09 Avril 2004)

[17] Nikolai BEZROUKOV. *Ports scanning link.* Disponible sur http://www.softpanorama.org/security/port_scanners/links.shtml (consulté le 08 Avril 2004)

[18] Kevin NOVAK.*VA scanners Pinpoint your Weak Spots.* Disponible sur http://www.networkcomputing.com/1412/1412f2.html (consulté le 09 Avril 2004)

[19] eEye Digital Security. *RETINA Network Security Scanner, Superior Vulnerability Assessment & Remediation.* Disponible sur http://www.eEye.com (consulté le 9 Avril 2004)

[20] MISC. *Les tests d'intrusion.* Disponible sur http://www.miscmag.com/articles/index.php3?page=109 (consulté le 08 Avril 2004)

[21] http://www.tout-savoir.net (consulté le 14 Avril 2004)

[22] http://www.echu.org/portail/modules/glossaire (consulté le 14 Avril 2004)

[23] http://www.alaide.com (consulté le 18 Avril 2004)

Annexe C Table des illustrations

Annexe D Nessus : installation et configuration

La démarche d'installation suivante est celle indiquée sur le site officiel de Nessus à l'adresse suivante : http://www.nessus.org.

1. Installation de Nessus

Une façon d'installer Nessus simplement est d'utiliser le script *nessus-installer.sh* qui permet de faire une installation automatisée. Il suffit de taper la commande : sh nessus-installer.dh

Il est nécessaire d'avoir le package sharutils préalablement installé sur le poste. Nous pouvons le télécharger sous forme de rpm et l'installer en tapant par exemple les commandes suivantes, s'il se présente sous forme d'une source :
Rpm build --rebuild lenomdusource.rpm
Rpm –ivh lenomdubinaire.rpm

Pour installer Nessus sur un système de type Unix, dans le cas où nous ne voulons pas utiliser le script d'installation automatique, il faut télécharger les fichiers suivants :

- nessus-libraries-x.x.tar.gz
- libnasl-x.x.tar.gz
- nessus-core.x.x.tar.gz
- nessus-plugins.x.x.tar.gz

Il faut décompresser les fichiers et les compiler dans l'ordre suivant:

Installation de Nessus-libraries

Compiler Nessus-libraries en tapant :

```
cd nessus-libraries
./configure
make
```

puis, en étant root, exécuter la commande suivante :

```
make install
```

Installation de libnasl

Compiler Nessus-libraries en tapant :

```
cd libnasl
./configure
make
```

puis, en étant root, exécuter la commande suivante :

```
make install
```

Il faut alors répéter la même opération avec Nessus-core et Nessus-plugins.

2. créer un compte sur le démon Nessus

Le démon Nessusd a sa propre base de donnée, chaque utilisateur peut avoir un ensemble de limitations. Ceci permet de partager un unique démon pour un réseau complet et plusieurs administrateurs qui pourront tester uniquement la partie du réseau qui leur est attribuée.

L'utilitaire *Nessus-adduser* permet de configurer un profil utilisateur , il suffit pour cela de suivre la démarche suivante :

```
# nessus-adduser

Addition of a new nessusd user
------------------------------

Login : renaud
Authentication (pass/cert) [pass] : pass
Password : secret

User rules
----------
nessusd has a rules system which allows you to restrict
the hosts
that renaud2 has the right to test. For instance, you may
want
him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules
syntax

Enter the rules for this user, and hit ctrl-D once you
are done :
(the user can have an empty rules set)

deny 10.163.156.1
accept 10.163.156.0/24
default deny

Login            : renaud
Password         : secret
DN          :
Rules            :

deny 10.163.156.1
accept 10.163.156.0/24
default deny


Is that ok (y/n) ? [y] y

user added.
```

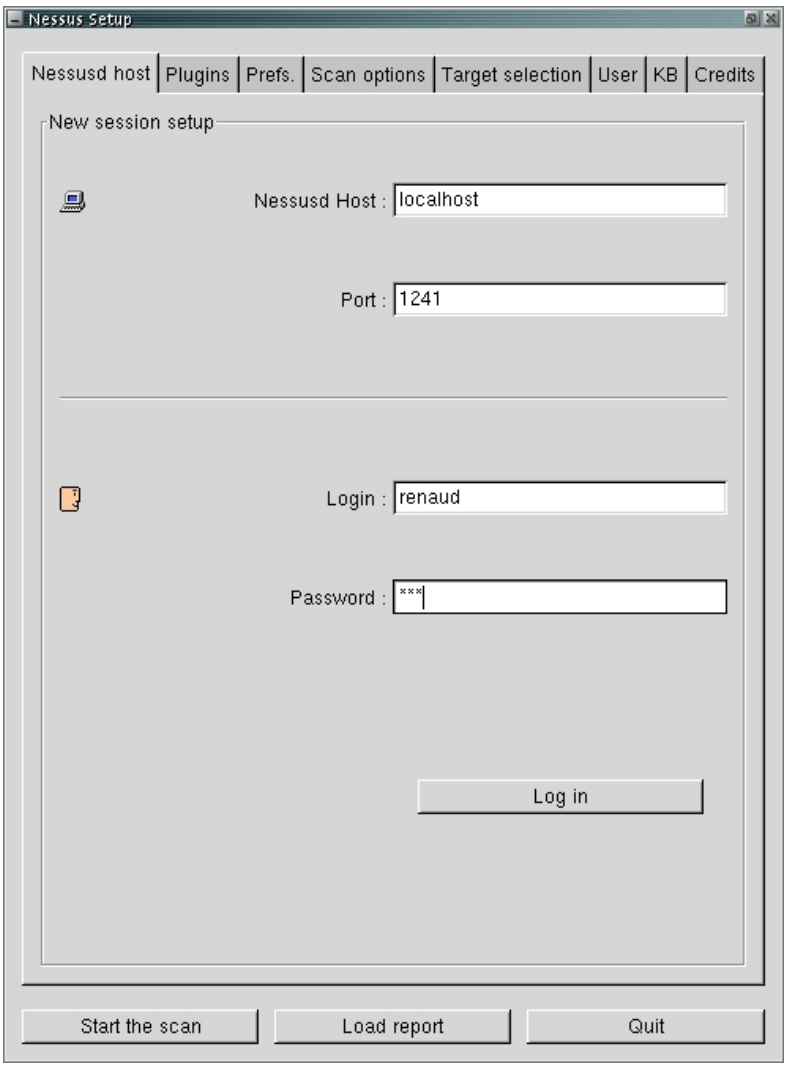
3. configurer le démon Nessus.

Si nous voulons changer les paramétrages par défaut du démon Nessus, il suffit de modifier le fichier suivant :
/usr/local/etc/nessus/nessusd.conf
Il est possible de spécifier dans ce fichier, les ressources que nous voulons que Nessus utilise, la vitesse à laquelle nous voulons effectuer la lecture des données …

Nous devons alors lancer le démon Nessus en tapant :
`nessusd -D`

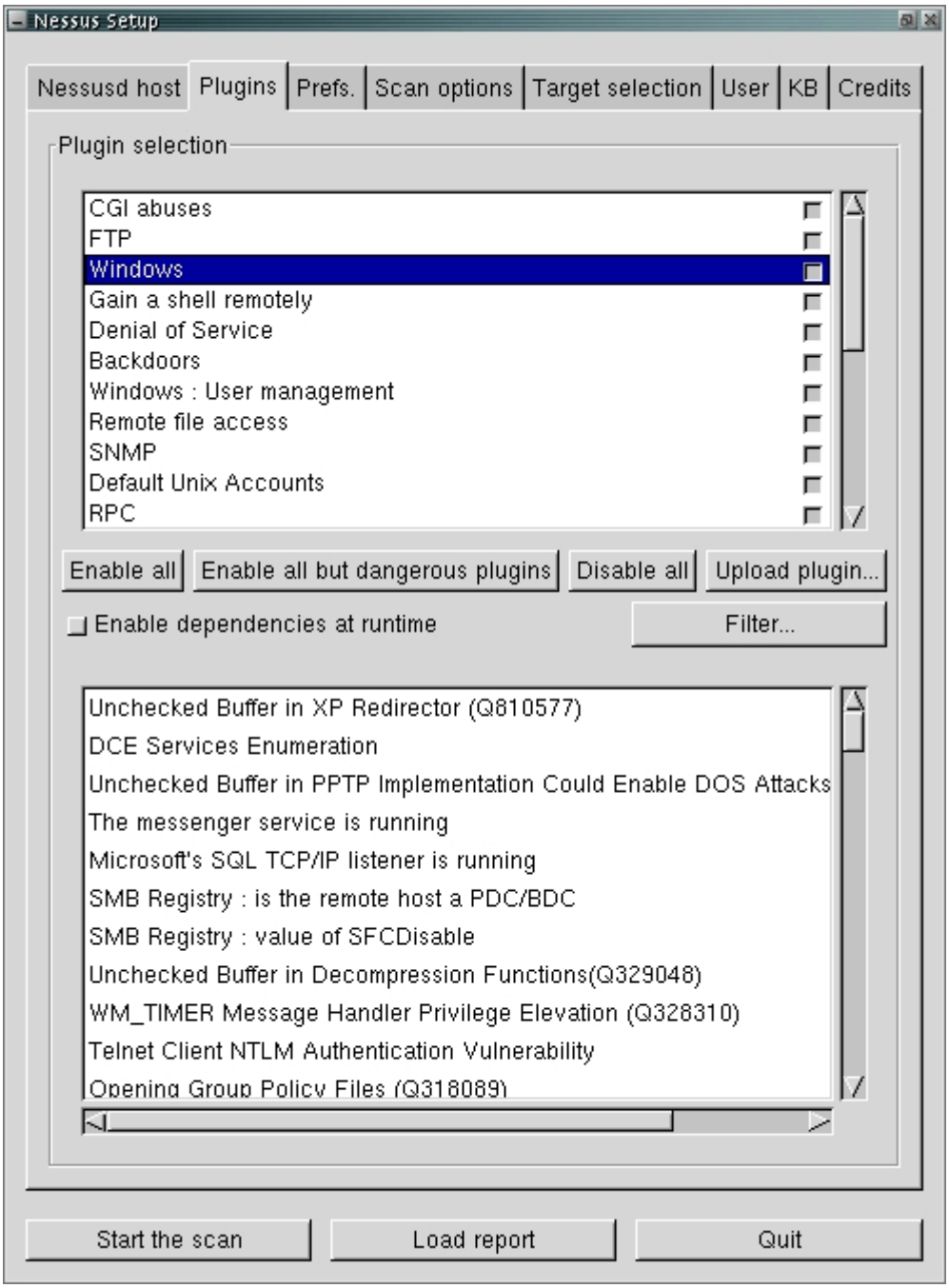
4. la configuration du client

- connexion au démon Nessus



Cette étape consiste à se connecter au démon Nessus.

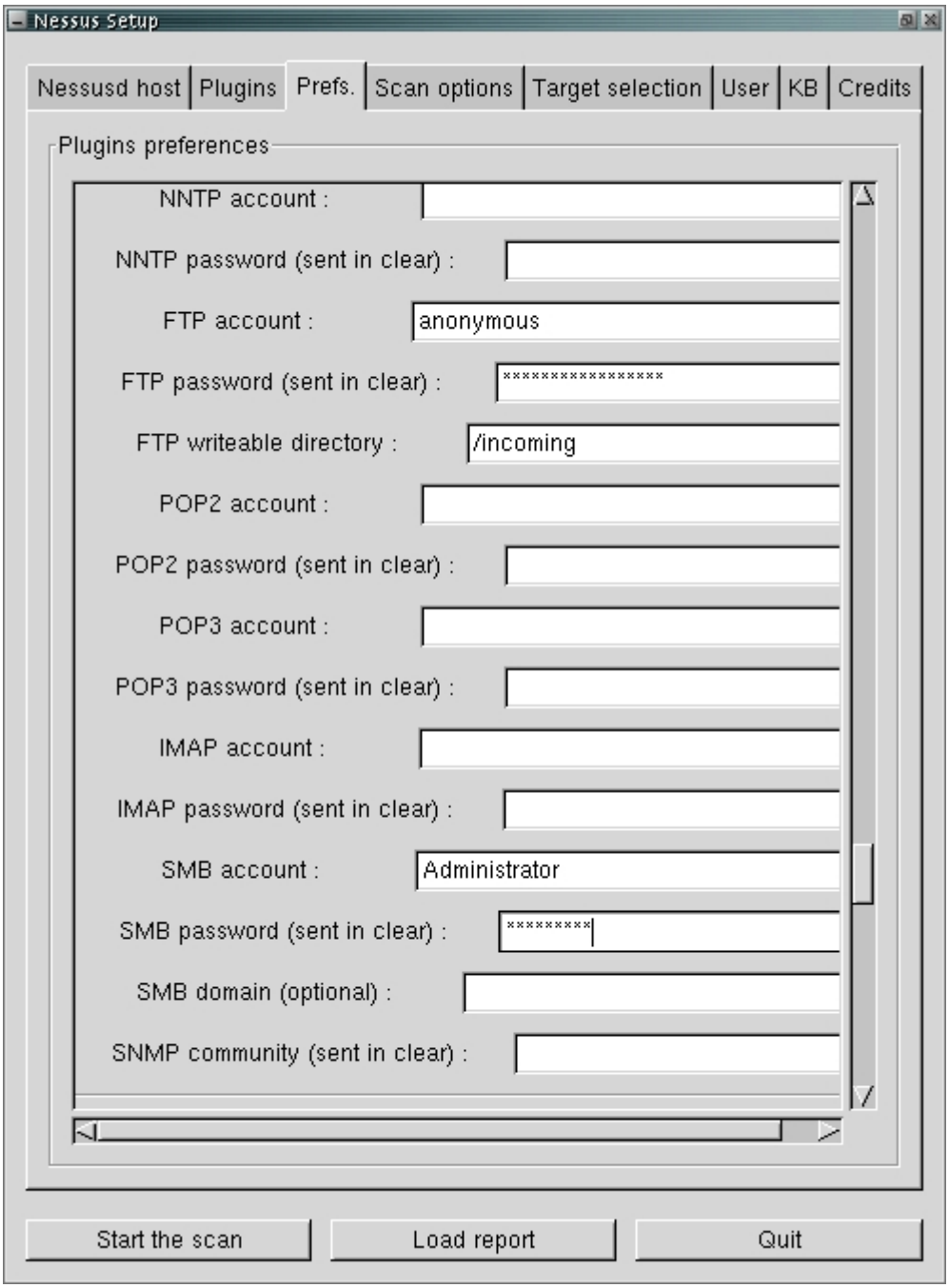
- configuration des tests de sécurité



il est possible de choisir quels **plug-in** on désire lancer au moyen de cet écran.

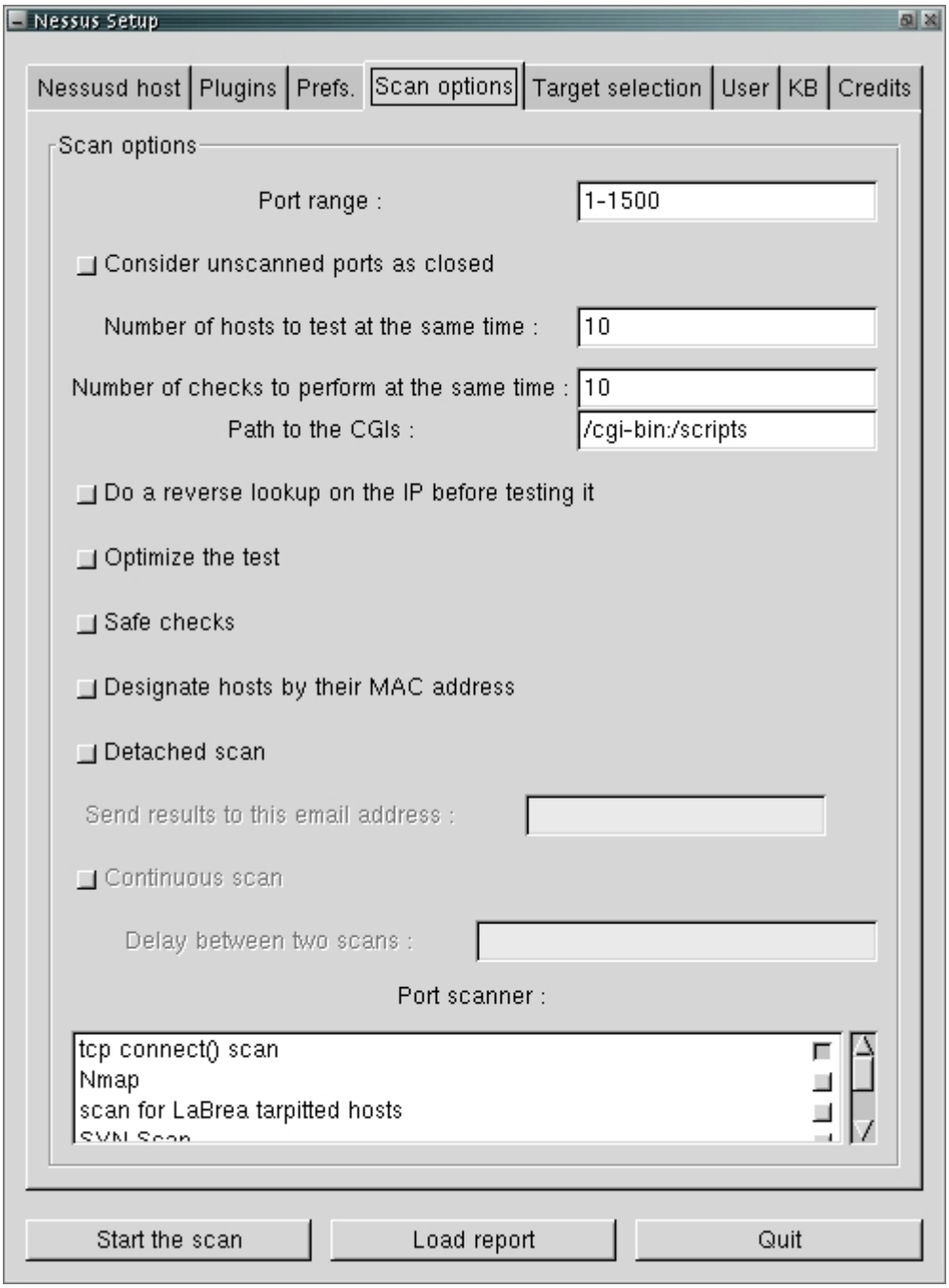
En cliquant su le nom d'un **plug-in**, une fenêtre expliquant ce que fait le **plug-in** s'affiche.

- préférences des **plug-in**



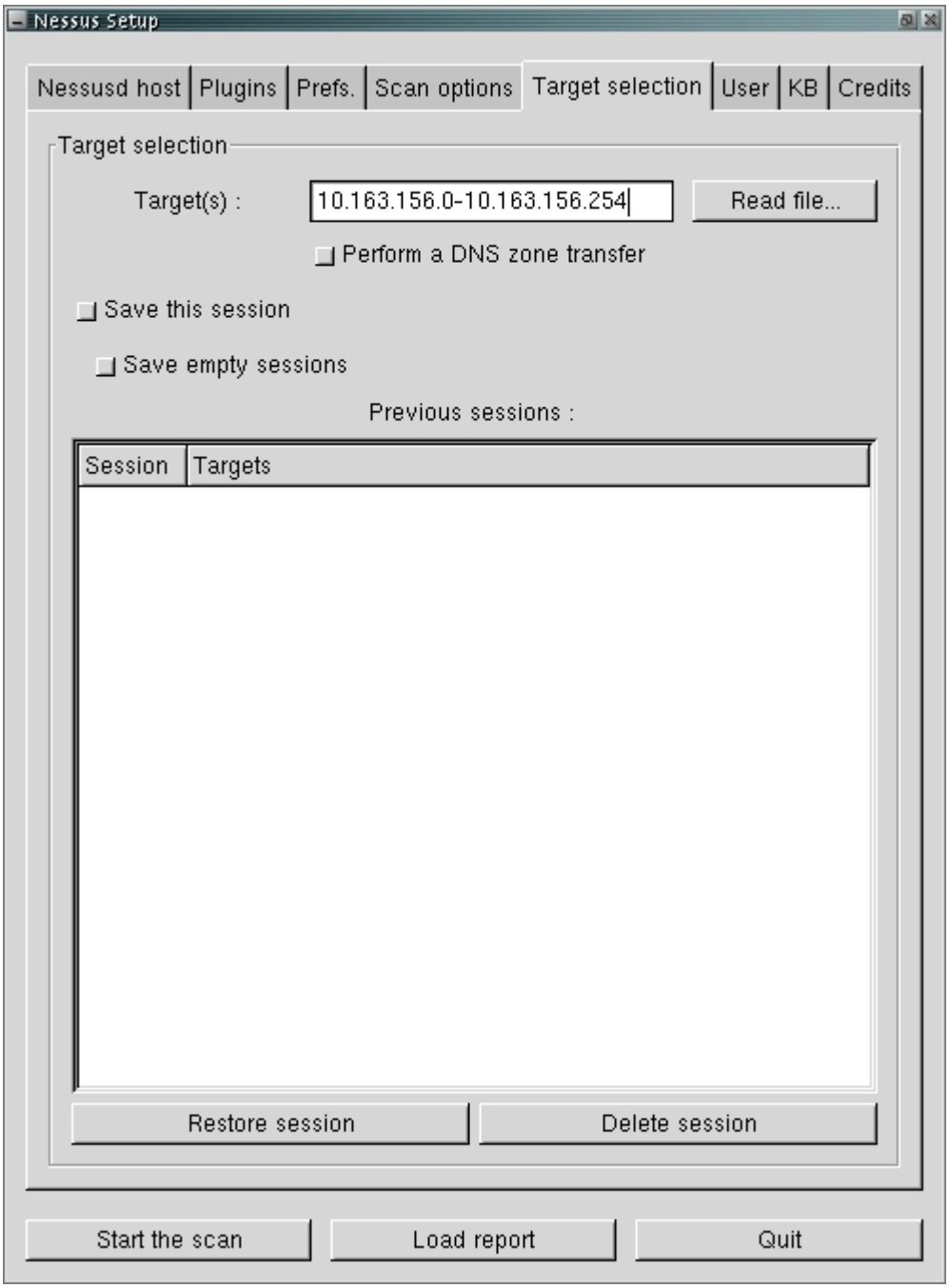
il est possible de fournir des informations complémentaires. L'audit sera alors plus complet.

- les options de scannage



Cette section permet de choisir le scanner de port que l'on veut utiliser, le nombre d'ordinateurs que l'on veut scanner en même temps et le nombre de **plug-in** que l'on veut exécuter en même temps sur la cible.

- définition de la cible :



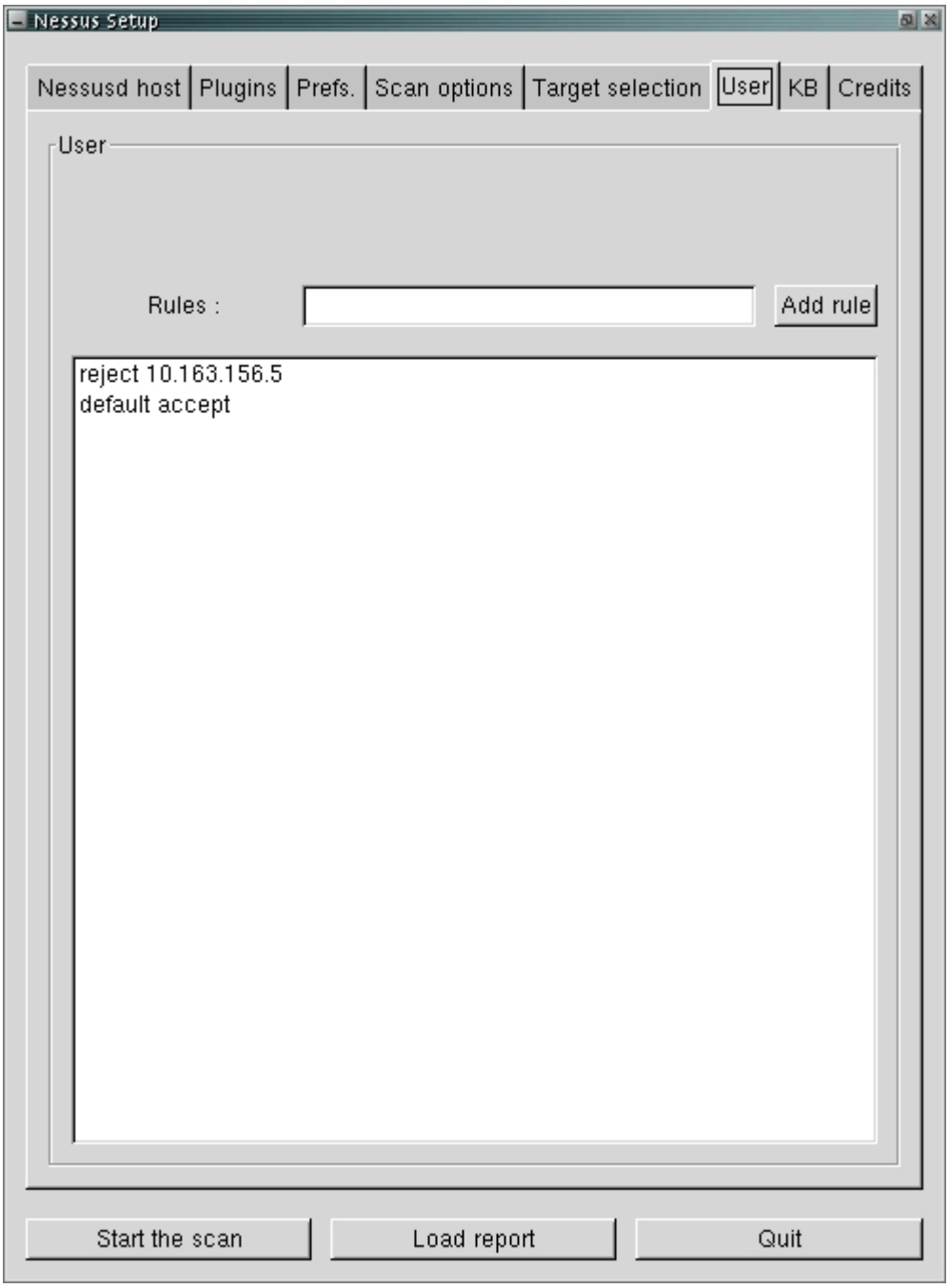
Il est possible d'indiquer la cible de diverses façons.

On peut utiliser une simple **adresse IP** ou une plage d'**adresse IP**.

La notation **CIDR** est supportée.

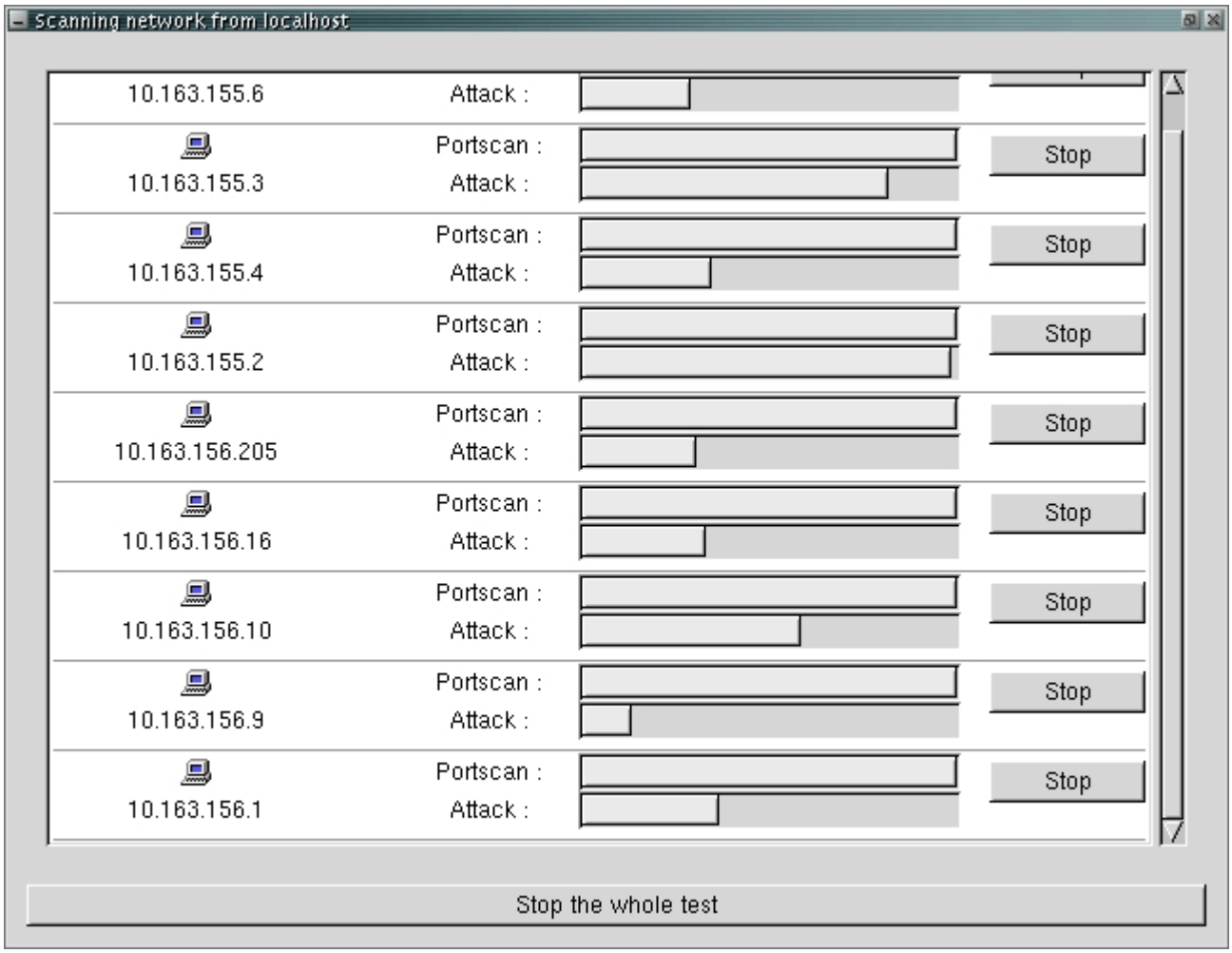
On peut également utiliser la qualification longue de la notation de domaine de nom.

- La section des régles :



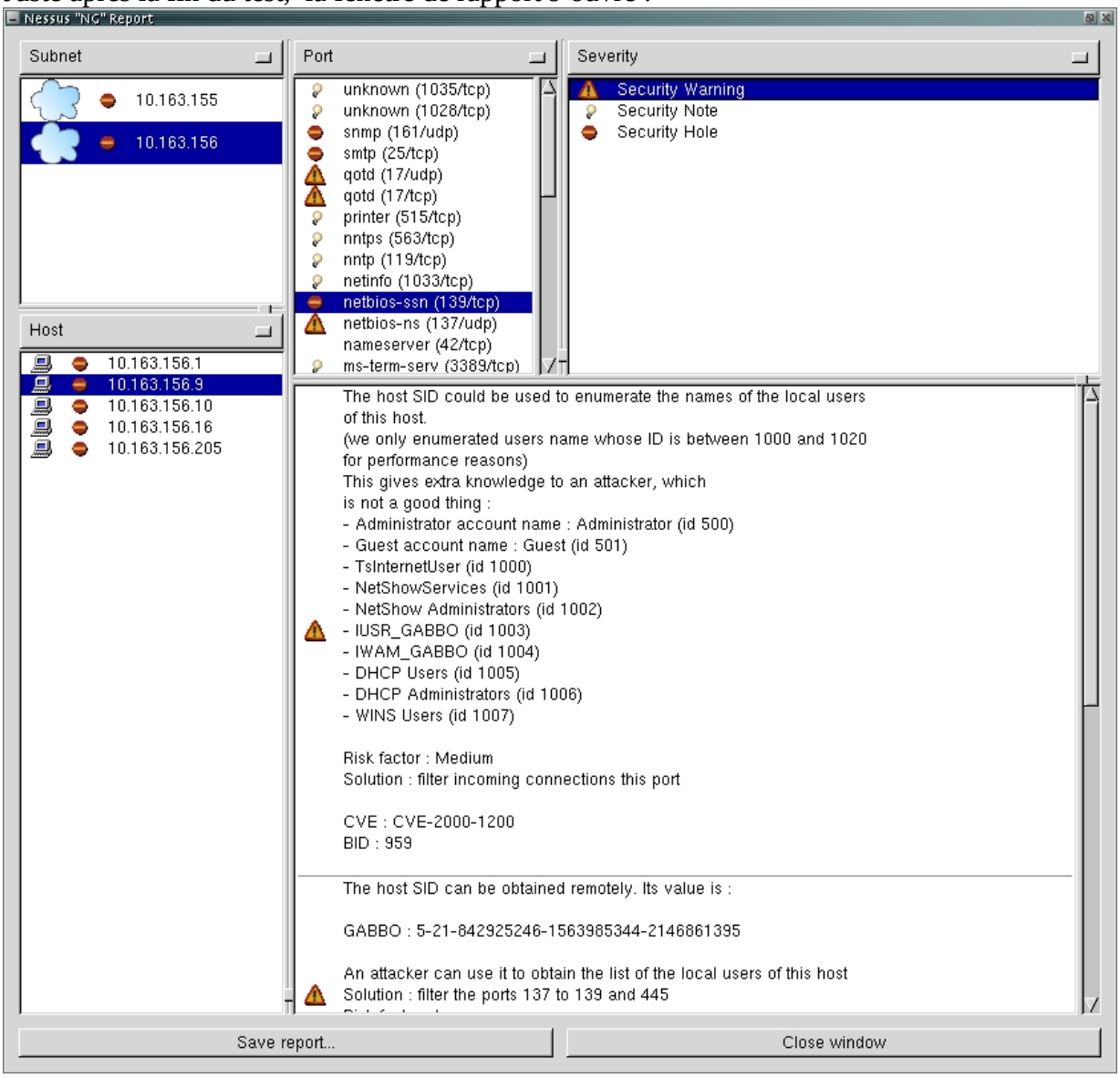
Les règles permettent aux utilisateurs de limiter leurs tests. Il est possible par exemple d'exclure une **adresse IP** que l'on ne veut pas tester.

Une fois que tout ceci est effectué, il est possible de commencer le test :



5 interprétation des rapports

Juste après la fin du test, la fenêtre de rapport s'ouvre :



Les rapports peuvent être sauvegardé sous différents formats.

- Au format NBE qui peut être lu par les clients Unix.
- Au format NSR
- Au format **spiffy HTML** qui est en fait du HTML avec des camemberts et des graphes
- Au format HTML
- Au format text **ASCII**
- Au format **LaTeX** (pdf)

Annexe E Failles de vulnérabilités détectées

Shh (22/tcp) : exécution arbitraire de commande sur le poste due à une faille présente au niveau de la gestion du **buffer**. La version de Shh est trop vieille.

Netbios-ssn (139/tcp) : possibilité de se connecter sur le poste en utilisant un utilisateur et un mot de passe **NULL**. Ce qui permet au pirate un accès en tant qu'invité.

Unknown (1024/udp) : exécution possible de code arbitraire due à un bug dans le démon correspondant à un service **RPC**.

Unknown (665/tcp) : possibilité d'obtenir un interpréteur en tant qu'administrateur sur le poste en exploitant une vulnérabilité du service **RPC** qui est ouvert sur ce port.

ftp (21/tcp) : possibilité de faire tomber le service **FTP** à l'aide d'une commande. Ce qui empêchera, par exemple, un site de fournir les services **FTP**.

http (80/tcp) : possibilité d'exécuter des commandes en tant qu'utilisateur système.

Unknown (135/tcp) : possibilité d'exécution de code arbitraire et d'obtenir des privilèges systèmes. gagner le contrôle de la machine.

Unknown (135/udp) : faille dans le service exploitable à l'aide d'une attaque **DoS**.

Snmp (161/udp) : l'agent **SNMP** répond à la communauté de nom public.

Smtp (25/tcp) : possibilité d'exploitation avec des attaques de type **DoS**.

General / tcp : possibilité de rendre indisponible les services **RPC** en envoyant une requête mal formée.

telnet (23/tcp) : le serveur telnet tombe lorsqu'il reçoit trop d'options.

Microsoft-ds (445/tcp) : exécution arbitraire de code.

Epmap (135/tcp) : gagner le contrôle de la machine.