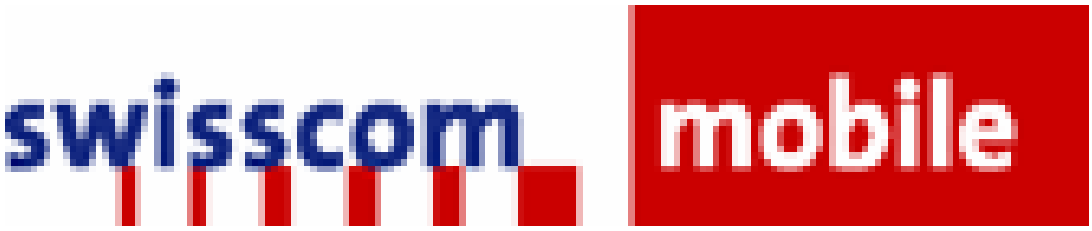


Travail de diplôme

Hybrid IDS

Réalisé par : Pierre Duc
Classe : ETR-6
Professeurs : Christophe Gabioud
: Stephan Robert

Hybrid IDS

Table des matières

Avant propos:

Ce travail de diplôme fait suite à une première étude sur le "security management" qui a été réalisée comme projet de semestre. Le travail de diplôme n'étant pas vraiment lié au travail de semestre, il ne sera pas fait mention de ce dernier autre part que dans cet avant propos. Les documents décrivant ce travail sont sur le CD, dans le répertoire **/ProjetDeSemestre**.

Par la suite et dans un but de simplification, nous allons utiliser les acronymes suivants:

-"CVE" pour indiquer une référence Common Vulnerability Exposure.
-"BID" pour indiquer une référence Bugtraq ID.
-"Nessus ID" pour indiquer une référence à un script NASL de Nessus.
-"SID" pour indiquer une référence à une règle Snort.
-"IDS" (= SDI en français) pour désigner un système de détection d'intrusion

1. Introduction

Avec toute cette agitation autour de l'insécurité informatique, les virus, les vers, les chevaux de Troie et la publicité autour des produits sensés nous protéger des hackers, il est souvent difficile de savoir si un outil supplémentaire tel un système de détection d'intrusion est utile, ou si cela est encore un produit superflu que l'on essaie de nous vendre. Un système de détection d'intrusions, appelé aussi IDS, est un appareil ou un logiciel capable de détecter des anomalies dans le trafic du réseau ou la modification de fichiers sur une machine. Il garde un fichier "log" de toutes les anomalies qu'il a relevées.

Le problème majeur des IDS est qu'ils ne sont pas parfaits et qu'ils lèvent une grande importante de fausses alarmes, provenant de deux facteurs principaux:

- Parce qu'un IDS est souvent un grand alarmiste et préfère lever de fausses alarmes que d'en perdre.

- L'IDS n'est pas fiable à 100%. Les règles ne sont pas parfaites et elles sont régulièrement mises à jour.

Une fausse alarme est appelée "false positive" dans le jargon informatique et une attaque non décelée une "false negative". Comme expliqué auparavant on préfère les "false positive" aux "false negative". C'est pour cela que les alertes sont plus nombreuses que les vraies attaques. Afin d'alléger le travail des responsables de la détection d'intrusions, il est nécessaire de trouver un moyen de réduire au maximum les "false positive". Pour atteindre ce but, on corrèle plusieurs sources d'informations différentes afin d'en extraire

les informations importantes. Ceci permet de séparer les vraies attaques de quelques fausses alertes. La corrélation peut être faite selon différents critères: adresses IP sources, adresses IP destinations, type d'attaques, etc.
Le but de ce travail de diplôme est de réaliser une corrélation entre un IDS et un scanneur de vulnérabilité, solution actuellement introuvable dans le monde Open Source.

1.1 Mandat

Le travail consiste à automatiser une partie de la détection d'intrusion dans un environnement restreint, grâce à une corrélation entre un IDS et un scanneur de vulnérabilités, puis d'automatiser cette corrélation le plus possible afin de soulager le travail des analystes de "logs" d'IDS.

1.2 Porté du projet

Ce projet n'a pas été étudié pour être mis en place sur un vaste réseau. On se limitera donc à une utilisation dans un sous réseau ou sur quelques machines.

1.3 Public cible

Ce travail se destine principalement aux responsables de la sécurité informatique et aux spécialistes de la détection d'intrusion.

2. Méthode et outils

2.1.1 Introduction

Une recherche sur l'Internet sur le fonctionnement des IDS actuels a été réalisée afin de trouver des informations sur la manière dont ces IDS fonctionnent et font leur corrélation, s'ils en font une. Au moment de cette recherche, aucun IDS Open Source ne faisait le type de corrélation que nous présentons dans cette étude. Seuls des produits commerciaux font une corrélation entre un IDS et un scanneur de vulnérabilité. Comme tout programme commercial qui se respecte, un minimum d'informations est donné pour se protéger des concurrents. On peut citer l'article qui a donné vie à ce travail de diplôme sur le site Securityfocus à l'adresse suivante:
http://www.securityfocus.com/infocus/1708

2.1.2 Les moyens de corrélation.

Afin de pouvoir faire une corrélation entre Nessus et Snort, il a fallu trouver un moyen de rapprocher les fichiers de sortie générés par ces deux programmes. Si nous y regardons de plus près, nous pouvons remarquer qu'il est possible de faire un lien sur ces fichiers de plusieurs manières.

Corrélation par les signatures
Une corrélation à partir des signatures des règles Snort et attaques Nessus aurait été envisageable. Néanmoins, il est impossible de récupérer directement le code de la signature à partir du rapport de Nessus et, en cas de modification de la signature, il ne serait pas aisé de déceler la mise à jour ou de faire cette mise à jour dans Snort à partir de cette signature. En effet, des modifications sont parfois faites au niveau des signatures afin de mieux cerner la vulnérabilité.

Corrélation par les noms
Comme chacune des règles Snort ou des vulnérabilités portent un nom, il aurait été envisageable de les corréler par leurs noms. Malheureusement, chaque programme utilise un nom différent pour les attaques que l'on ne retrouve pas dans les autres logiciels. Ceci se retrouve dans toute organisation prenant part à la standardisation des attaques: chacune donne un nom particulier à la vulnérabilité. Ceci n'aide en rien cette même standardisation.

```
[**] [1:598:10] RPC portmap listing TCP 111 sid598 [**]
[Classification: Decode of an RPC Query] [Priority: 2]
```

Figure 2-1 Nom de la vulnérabilité dans la règle snort

Name	CAN-1999-0632 (under review)
Description	The RPC portmapper service is running.
References	
Phase	Proposed (19990804)
Votes	ACCEPT(2) Baker, Wall REJECT(1) Northcutt
Comments	

Figure 2-2 Nom de la vulnérabilité dans la base de donnée CVE

Corrélation par les références
Il existe quelques organisations qui ont mis en place un système de références afin de pouvoir, à partir d'un nombre, retrouver avec certitude et sans ambiguïté la vulnérabilité. Chacune de ces organisations a cependant son propre système de numérotation. Les mises à jour ne sont pas synchronisées et la standardisation d'une organisation n'est pas reconnue par les autres. Chacun fait des choix différents, ce qui fait qu'il est rare d'avoir

une seule définition pour une même vulnérabilité. Il est aussi impossible de faire une conversion numérique entre les références.
Les trois grandes organisations actives dans le monde de la publication et de la standardisation des vulnérabilités sont:

- Le MITRE [1] avec leurs références CVE (Common Vulnerability and Exposure) http://www.cve.mitre.org/
- Security focus [2], racheté par Symantec avec leurs références bugtraq BID: http://www.securityfocus.com/bid
- Whitehats [3] avec leurs références arachNIDS: http://www.whitehats.com/cgi/arachNIDS/Search

Name	CAN-2003-0528 (under review)
Description	Heap-based buffer overflow in the Distributed Component Object Model (DCOM) interface in the RPCSS Service allows remote attackers to execute arbitrary code via a malformed RPC request with a long filename parameter, a different vulnerability than CAN-2003-0352 (Blaster/Nachi) and CAN-2003-0715.
References	• VULNWATCH:20030911 NSFOCUS SA2003-06 : Microsoft Windows RPC DCOM Interface Heap Overflow Vulnerability • URL:http://archives.neohapsis.com/archives/vulnwatch/2003-q3/0100.html • MISC:http://www.nsfocus.com/english/homepage/research/0306.htm • BUGTRAQ:20030920 The Analysis of RPC Long Filename Heap Overflow AND a Way to Write Universal Heap Overflow of Windows • URL:http://marc.theaimsgroup.com/?l=bugtraq&m=106407417011430&w=2 • MS:MS03-039 • URL:http://www.microsoft.com/technet/security/bulletin/MS03-039.asp • CERT:CA-2003-23 • URL:http://www.cert.org/advisories/CA-2003-23.html • CERT-VN:VU#254236 • URL:http://www.kb.cert.org/vuls/id/254236

Figure 2-3 Quelques références pour une même vulnérabilité

2.1.3 Solution de corrélation retenue

Introduction
Les références présentes dans les rapports Nessus ne sont pas toujours les mêmes. Cela dépend du format du rapport Nessus. Les vulnérabilités traitées par Nessus ont toutes au moins un numéro de référence CVE ou CAN.

[1] http://www.mitre.org/
[2] http://www.securityfocus.com/
[3] http://www.whitehats.com

Le problème est qu'il n'existe pour l'instant pas forcément une référence CVE pour chaque règle Snort. Il faudrait donc mettre à jour toutes les règles n'ayant pas de référence CVE à l'aide des autres références données.

```
Risk factor : Low
CVE : CAN-1999-0632, CVE-1999-0189
BID : 205
```

Figure 2-4 Version .txt du rapport Nessus

Risk factor : Low
CVE : CAN-1999-0632, CVE-1999-0189
BID : 205
Nessus ID : 10223

Figure 2-5 Version .html du rapport Nessus

Il est cependant possible de trouver les références CVE à partir des autres types de références car les sites webs de ceux-ci contiennent aussi une fonction de recherche pour d'autres références. Néanmoins, il serait très contraignant de devoir chercher toutes les références et de les mettre à jour dans les règles Snort. En mettant à jour les règles, il faudra aussi remettre à jour les références CVE.
Il n'a pas été trouvé de base de donnée contenant toutes les références selon les règles Snort ou CVE. Il existe des sites avec:

- un lapping BID→CVE
- un lapping arachNIDS→CVE et autres références

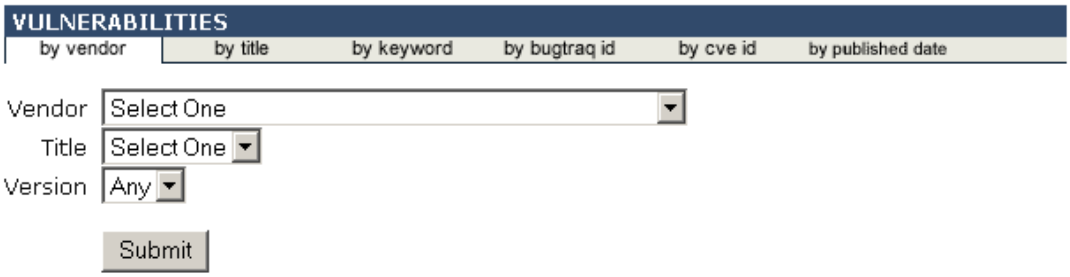


Figure 2-6 Recherche de références depuis www.securityfocus.com

Mais il n'existe pas de mapping SID→CVE sauf pour les règles déjà en place dans Snort. Ce qui est totalement inutile, car notre but est d'augmenter le nombre de références CVE dans Snort.

On peut donc remarquer qu'il existe une grande masse d'informations parsemées et que malheureusement il est impossible de corréler ces informations directement.
Le problème est identique pour Nessus. Il existe des références CVE, BID et quelques autres mais pas de références directes avec les règles Snort. Il faut donc se résoudre à trouver le type de référence le plus fréquent dans les deux programmes.

Le type de référence CVE est le plus présent dans les deux programmes. Mais il se peut qu'un NASL corresponde à plusieurs CVE. Si nous prenions les CVE comme référence de base, nous aurions plusieurs références pour Snort. Pour maximiser les chances d'avoir une corrélation exacte (une vulnérabilité = une règle Snort), il nous faut donc prendre les numéros NASL pour base. Pour atteindre ce but, il nous faudra par la suite ajouter les références NASL aux règles Snort.

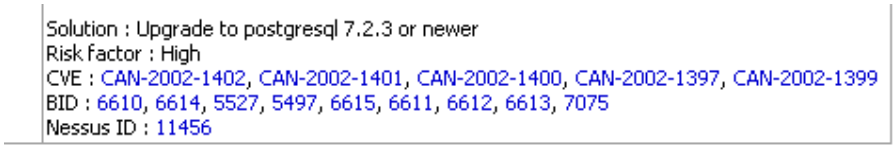


Figure 2-7 Rapport Nessus avec plusieurs références

Une corrélation moins stricte est toujours possible. Notre solution finale corrèle à partir de toutes les références présentes dans le rapport Nessus en version html.

Il serait aussi possible de faire le travail contraire: adapter le rapport Nessus avec les références SID des règles Snort. Mais il faut savoir que Nessus utilise un script particulier pour les références présentes dans son rapport et qu'il n'existe pas de script pour les références SID. Cela serait donc peut-être un travail inutile, si Nessus changeait son format de présentation.

The RPC portmapper is running on this port.

An attacker may use it to enumerate your list of RPC services. We recommend you filter traffic going to this port.

Risk factor : Low
reference SNORT: sid 598
CVE : CAN-1999-0632, CVE-1999-0189
BID : 205
Nessus ID : 10223

Figure 2-8 Rapport Nessus avec référence SID ajouté.

2.1.3.1 Les étapes

Mesure

Afin d'avoir des fichiers à analyser, des mesures on été effectuées. Pour ne pas avoir des problèmes de visibilités, les trois stations seront placées sur un hub. La mesure consiste à faire un scan complet avec tous les plugins Nessus activés d'une machine du réseau sur une autre. Un portable est placé sur le hub. Celui-ci fera office d'IDS. La machine attaquant a un système Debian avec Nessus 2.0.7. Le post attaqué a un système SuSe 8.0 et le portable une Debian avec Snort 2.0.2.

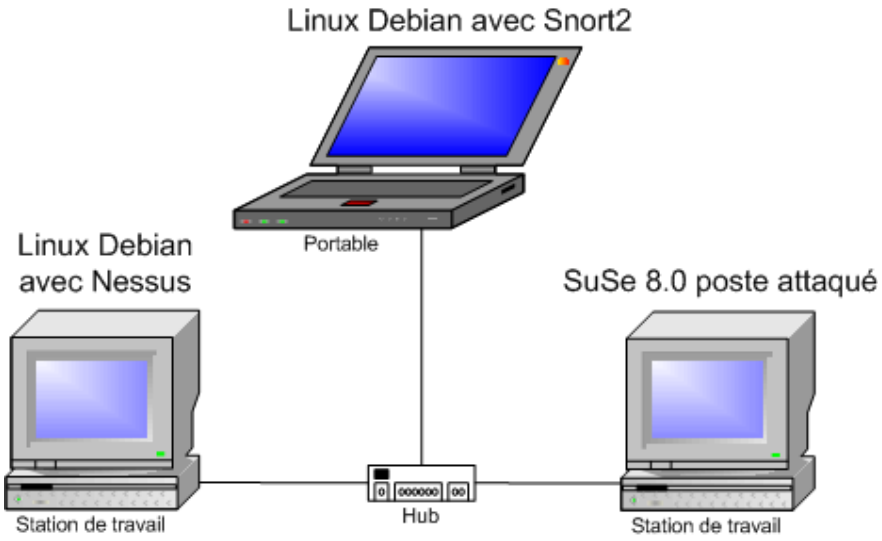


Figure 2-9 Schéma de mesure

Première approche
Une première solution a été réalisée en se basant sur les références CVE. Ceci pour deux raisons:

- Des références CVE existent dans les deux programmes.
- Un test de faisabilité devait être fait.

Un petit script Perl a été écrit sur un premier fichier; le fichier rapport de Nessus.
Ceci afin de récupérer les diverses références CVE de la version .txt.

Par la suite, la même chose a été faite sur le fichier alert de Snort. Comme la syntaxe n'est pas la même, il a fallut adapter le script Perl pour le second fichier. En fait, il a été écrit deux routines différentes écrivant chacune dans un fichier particulier les références CVE rencontrées dans les fichiers parsés. Ces deux routines forment le premier programme Perl qui crée trois fichiers:

- Un premier fichier outputfile contenant les références trouvées dans le rapport Nessus
- Un deuxième outputfile2 contenant les références trouvées dans le fichier alert de snort.
- Un troisième fichier **finaloutputfile** est crée à partir des deux premiers fichiers. Il contient les références communes. Ceci donne une première corrélation entre Nessus et Snort.

Deuxième approche
Après acceptation du procédé, il a été réalisé un programme plus conséquent en se basant cette fois-ci sur la version html du rapport Nessus, car cette version permet de récupérer les numéros NASL qui étonnement ne sont pas mentionnés dans la version txt. Le script a été modifié afin de chercher en fonction de plusieurs références possibles en même temps, en un seul passage. Cette version utilise toutes les références relevées sur le rapport pour la corrélation. Par souci d'efficacité, il a été utilisé le rapport html de Nessus comme base et les paquets ajoutés à la suite de la description de la vulnérabilité. Cette version a finalement été écartée car elle ne mettait pas assez en avant l'essentiel du travail, les paquets incriminés. Par souci de simplicité lors de la mise à jour, il serait préférable de faire la recherche que sur le Nessus id, car il suffirait de savoir quelle règle Snort correspond à la vulnérabilité et placer ce numéro dans la règle Snort.

The RPC portmapper is running on this port.

An attacker may use it to enumerate your list
of RPC services. We recommend you filter traffic
going to this port.

Risk factor : Low
ref!!! CVE : CAN-1999-0632, CVE-1999-0189
ref!!! BID : 205
Nessus ID : 10223

[**] [1:598:10] RPC portmap listing TCP 111 sid598 [**]
[Classification: Decode of an RPC Query] [Priority: 2]
10/15-08:43:52.652233 10.192.72.196:1293 -> 10.192.73.121:111
TCP TTL:64 TOS:0x0 ID:2480 IpLen:20 DgmLen:96 DF
AP Seq: 0x9DCB0852 Ack: 0xE5FD00BD Win: 0x3EBC TcpLen: 32
TCP Options (3) => NOP NOP TS: 76926 90004
[Xref => cve CAN-1999-0632][Xref => arachnids 428]

Figure 2-10 Extrait du rapport pour la deuxième approche.

Le paquet est inséré dans le rapport Nessus.

Troisième approche

Un dernier programme a été réalisé afin d'afficher les paquets suspects corrélés suivis de la description de la vulnérabilité. Cette version améliore grandement la lisibilité et met en évidence les paquets suspects de manière plus claire. Les paquets sont ici le centre d'intérêt. Cette version a aussi comme format l'html mais cette fois-ci, la mise en page est personnelle.

Figure 2-11 Rapport de corrélation final

Adaptation de la corrélation à l'environnement

Pour réduire la corrélation, il faut un certain "tuning" de l'IDS afin de lui enlever les règles de détection qui ne nous intéressent pas. En effet, si on lève une attaque smtp alors que le service n'existe pas sur le réseau ou la machine, cela ne fait qu'alourdir le travail de l'administrateur, et surtout inutile car même si l'attaque est réellement présente, elle n'a aucune chance d'aboutir. De plus, les alertes Snort n'étant pas des plus explicites, il est souvent difficile de savoir exactement ce que Snort a véritablement détecté.

Afin de clarifier encore plus les milliers de lignes que Snort aurait pu "logger", de réduire au maximum le rapport final, et de s'occuper des choses les plus importantes en premier lieu, il nous faut traiter et lever une alarme que si Snort a détecté une attaque correspondant à une vulnérabilité connue sur nos machines.
C'est ici que nous voyons l'utilité d'avoir testé les vulnérabilités avec Nessus. La première chose à faire est de "patcher" nos vulnérabilités, mais il n'est pas toujours possible de le faire sur des serveurs de production sur lesquels un arrêt ou un mauvais patch (Microsoft) signifierait encore plus d'embêtements que l'attaque elle-même. L'attaque est souvent plus vite mise à disposition que la solution. En ayant un rapport contenant toutes nos faiblesses et en faisant une corrélation avec les détections d'attaques que l'on a décelées, il nous est possible d'obtenir une liste des choses vraiment importantes et de se concentrer directement sur ces dernières en premier lieu, de se focaliser sur les actions importantes à prendre en cas de véritable problème.

2.1.4 Type

Plusieurs types d'outils sont disponibles dans le monde de la sécurité informatique. En voici un descriptif sommaire. Notons que les outils utilisés dans ce travail de diplôme auront un descriptif plus conséquent.

2.1.4.1 Les IDS

Les IDS, ou systèmes de détection d'intrusions, sont des systèmes software ou hardware conçus afin de pouvoir automatiser le monitoring d'événements survenant dans un réseau ou sur une machine particulière, et de pouvoir signaler à l'administrateur système, toute trace d'activité anormale sur ce dernier ou sur la machine surveillée. Ces systèmes de surveillance du réseau sont devenus pratiquement indispensables dû à l'incessant accroissement en nombre et en dangerosité des attaques réseaux depuis quelques années.

La détection d'intrusion est le processus de vigilance du réseau cherchant toute trace de tentative d'intrusion à ce même réseau. Une intrusion est définie comme toute tentative pouvant nuire à l'intégralité, la confidentialité ou la disponibilité dans le réseau ainsi que toute tentative visant à contourner les dispositifs de sécurité mis en place sur le réseau ou une machine. Ces intrusions sont faites par des gens malhonnêtes essayant d'acquérir des droits ou des privilèges (sur une machine ou sur le réseau) qu'ils ne possèdent pas ceci

depuis l'extérieur comme depuis l'intérieur du réseau. Ces tentatives d'intrusions peuvent être bénignes comme extrêmement dangereuses et préjudiciable pour l'entreprise.

Les HIDS
Les "Hosts Intrusion Detection Systems" sont des IDS ayant été crées dans le but de surveiller ce qui ce passe sur une machine particulière mais pas sur le réseau. Un HIDS à la tâche particulière de devoir surveiller prioritairement l'intégrité des fichiers situés sur la machine. De plus, il va "logger" tous les accès sur la machine afin de pouvoir déterminer qui a fait quoi en cas de problème. Il peut ainsi déceler des attaques de "logging" par "brute force" et indiquer quels sont les fichiers qui ont été modifiés[1]. Aux premières utilisations de l'HIDS sur une machine, il ne faut pas être surpris par le nombre de choses inattendues que l'on peut y voir. Beaucoup de manipulations interdites vont vous sauter aux yeux.

Les NIDS
Les "Network Intrusion Detection Systems" sont des IDS ayant été crées dans le but de surveiller la partie du réseau sur laquelle ils ont été placés. Un NIDS voit tous les paquets circulant dans son sous-réseau, mais il n'est pas capable de vérifier l'intégrité des fichiers sur les machines, contrairement à un HIDS. De même, il va "logger" tous les paquets suspects.

Les IDS hybrides
Un "hybrid" IDS est une sorte de tout en un, c'est un HIDS avec un NIDS. Ce nom peut aussi s'appliquer à une solution mêlant plusieurs IDS, ou des IDS particuliers. Un exemple d'"hybrid" IDS est donné par la suite, avec PRELUDE.

Les IPS
Un IPS (Intrusion Preventive System) est la toute dernière nouveauté en matière d'IDS, c'est le nouveau mot à la mode que l'on va entendre sortir de tout bon commercial qui se respecte. Un IPS n'est rien de nouveau au niveau technique, c'est seulement un nouveau terme technique décrivant la combinaison de l'utilisation d'un IDS de détection en temps réel avec la capacité d'un contrôle total sur ce qui va être "forwardé" ou pas à l'adresse de destination. Un IPS est capable de prévenir une attaque, car il est capable de détecter une attaque avant qu'elle atteigne sa destination.
La plupart des IDS actuels peuvent être modifiés de manière à devenir des IPS. Il suffit par exemple de prendre un NIDS tournant sur un système d'exploitation laissant la capacité à l'application de décider ce qui doit ou pas être "forwardé" d'une interface à une autre. Modifier l'IDS de manière à ce qu'il utilise la façon dont se fait ce genre de manipulations sur le système d'exploitation choisi. Décider si oui ou non le paquet doit être "forwardé" ou pas. Un IPS est une façon de rendre un IDS actif et non pas passif comme ils le sont actuellement. Malheureusement les IPS ne font que de diminuer le temps entre la détection d'une attaque et le moment où vous réagissez à celle-ci.

[1] Un HIDS est un fabuleux outil d'espionnage, même s'il faut rappeler que l'espionnage des employés est interdit par la loi sauf s'il y a une forte présomption de culpabilité et sous demande écrite de la direction !

Les honeypots
Les "honeypots" sont des machines piège placées sur le réseau comme leurre pour le hacker. Un "honeypot" simule le fonctionnement d'une machine du réseau tout en "loggant" toutes les choses qui s'y passent, afin d'apprendre comment s'y prend le hacker. Il est ainsi possible de remarquer qu'elles sont les faiblesses de nos machines et où il serait judicieux de renforcer la sécurité à moindre frais. Un honeypot est en fait un HIDS placé sur une machine leurre.

Les firewalls
Les firewalls sont des composants indispensables de la sécurité informatique. Un firewall est un filtre à paquets. Selon les règles qu'on lui impose le firewall filtre les paquets par type de contenu, adresse IP, type de messages etc. Il peut être couplé à un IDS afin que celui-ci modifie ses règles. Cette dernière fonctionnalité n'est pas toujours indiquée, car il est possible de créer des "denial of service" soi-même, dû aux "false positive" de l'IDS. Néanmoins, cela peut se révéler utile dans certains cas.

2.1.4.2 Les scanneurs de vulnérabilités

Pourquoi utiliserait-on un détecteur de vulnérabilités, à quoi cela sert-il ?
Un scanneur de vulnérabilités détecte les failles dans notre système avant de le mettre en environnement de travail. Cela permet aussi de déceler ce que le hacker voit de notre réseau. Il faut faire attention lors de son utilisation, car il est capable de réaliser un déni de service ou de planter des machines. Un scanneur de vulnérabilité est aussi utilisé pour tester la robustesse des machines (disponibilité).

Les hackers utilisent sciemment les outils de protection afin de savoir comment ils marchent, comment les contourner ou leur faire dire n'importe quoi [1]. Cela leur permet de contrer les protections mises en place contre eux. Les hackers s'inspirent beaucoup des outils de protections afin de développer leurs propres outils. Alors si les hackers utilisent, s'inspirent et apprivoisent nos outils, pourquoi ne nous leur rendrions pas la pareille ! Si nous sommes capables de voir les failles qu'ils voient, il nous sera plus facile de les colmatés. Si nous comprenons comment leurs outils fonctionnent, il nous sera plus facile de les contrer.
Le fait de passer le réseau aux détecteurs de vulnérabilités est une sorte de label de qualité. En effet, il est préférable d'être sûr de la qualité du produit avant de le mettre en vente, d'être sûr que le réseau se relèvera après un accident, qu'il n'y ait pas trop de dégâts. Il est préférable que le réseau crash dans un environnement de tests que lors de la production, surtout dans des environnements complexes où les pannes sont plus gênantes et difficiles à trouver.

[1] Les scanneurs de vulnérabilités sont avant tout des outils pour administrateur réseau. Ce n'est que par la suite, qu'ils ont été détournés de leur dessein par des hackers.

2.1.5 Conclusion sur les outils

On remarque qu'il existe une multitude d'outils et de combinaisons possibles. Le plus dur reste le choix des armes, leur placement et leur configuration.

2.1.5.1 Complémentarité des solutions

On peut combiner plusieurs types d'outils en même temps pour augmenter la sécurité du réseau. Les fabricants misent sur la diversité de mise en place de leur IDS. Nous pouvons en trouver dans les firewalls, les réseaux NIDS, les machines HIDS, des machines dédiées computer IDS, dans les "honeypots", des solutions réparties, avec un manager et des sensors, il y a aussi des IPS et des hybrid IDS faisant office d'HIDS et de NIDS.

2.1.6 Recommandations pour les IDS

2.1.6.1 Questions sur l'architecture des IDS.

Il nous faut examiner les problèmes auxquels seront confrontés les utilisateurs et les constructeurs de ces systèmes.

Placement de l'IDS
Un détecteur devrait être fourni avec deux cartes réseaux:

- Une carte en mode promiscuité (qui capture tout), mais sans adresse IP. Ce qui compliquera la tâche de l'attaquant pour trouver le détecteur.

- Une deuxième carte avec une adresse IP afin de communiquer avec le détecteur.

Les événements remarquables
Les événements remarquables sont un concept important car l'analyste peut obtenir de meilleurs résultats s'il sait ce qu'il doit chercher avec son IDS et qu'il peut le régler en conséquence. Il est important de savoir ce que l'on cherche et ce que l'on veut relever, car il n'est pas possible de tout enregistrer, collecter.

Problèmes d'évaluation
Tous les incidents ne sont pas équivalents et ne doivent donc pas être traités de la même manière.

Les événements non observables

Les événements sur un autre réseau
En effet, il n'est pas possible d'observer les connexions dérobées sur un autre réseau. Les réseaux commutés donnent donc plus de fil à retordre qu'un réseau "hubé" pour la gestion de la sécurité avec l'IDS.
Panne de l'IDS
Les pannes de l'IDS entraînent des pertes de données. Les disques durs trop plein aussi. Il est indiqué de relancer, rebooter l'IDS ou la machine de temps à autres, ceci constituant un remède de gourou aux pannes des IDS.

Protocoles non décodés
Si l'on n'est pas capable de décoder IPX ou NETBIOS, on ne sera pas capable de détecter des intrusions sur des réseaux NOVELL ou Windows. Il faut donc s'assurer que l'IDS a toutes les fonctions nécessaires au travail que l'on va lui assigner.

Limites du volume de données de l'IDS
Qu'elle est la capacité maximum de l'IDS que l'on utilise ? Il faut faire attention à ne pas dépasser sa limite de capacité, sinon nous serons en train de filtrer qu'une partie des paquets reçus, faute de possibilité de faire mieux. Nous allons donc perdre des données qui peuvent s'avérer cruciales. Les fabricants d'IDS ont tendance à exagérer les débits. Il est de mise de ne pas faire confiance aux chiffres donnés par le fabricant.

Limites liées au facteur humain
Il n'est pas toujours possible de tout relever, car il est probable qu'aucun filtre n'existe pas de filtre pour l'attaque en question.

Stratégies du pare-feu
La stratégie de filtrage standard consiste à ne rien laisser passer, c'est-à-dire tout refuser, sauf ce qui n'a pas été spécifiquement autorisé. Si le trafic ne correspond pas à une règle, alors il est supprimé par la règle par défaut. La possibilité d'être surpris par un trafic non autorisé ou un service oublié est nettement plus faible qu'avec une stratégie "tout autoriser". La stratégie "tout autoriser" peut être utile dans un cas où la liberté est à favoriser plutôt que la sécurité. Néanmoins cette approche comporte plus de risque, rend la gestion de la sécurité plus difficile, et complique la mise en place. La stratégie "tout autoriser" est donc à utiliser en dernier recours.

La connaissance de la stratégie du pare-feu permet à l'analyste, une meilleure connaissance de ce qui doit ou ne doit pas être considéré comme du trafic normal, et permet aussi d'augmenter l'efficacité de l'IDS si cette stratégie peut être intégrée dans les filtres. Une autre faiblesse des IDS est que la stratégie de site n'est pas implémentée. Cela implique que l'on délaisse de manière tacite une façon de pouvoir affiner le système d'analyse du trafic. Cela est peut être dû au manque de coordination ou de bonne volonté entre les diverses personnes en charge de la sécurité.

Pour pouvoir travailler de manière efficace, il faudrait que les personnes responsables du pare-feu et de l'IDS travaillent d'un commun accord.

Les IDS et les signatures
Les signatures n'en sont qu'à leur début. Les anti-virus comportent environ 20'000 de signatures de virus alors qu'un IDS en a environ que 2'000 à 3'000. De plus, un anti-virus est optimisé pour la recherche sur un fichier, alors qu'un IDS doit faire ses recherches au niveau des paquets (problème de la fragmentation, etc.). D'un point de vue fonctionnel, il n'y a pas grande différence entre un pare-feu personnel et un IDS. C'est pour ceci que l'on peut voit fleurir des solutions intégrant un pare-feu personnel avec un anti-virus comme la solution logicielle de Symantec: "Internet Security". On peut aussi regarder une partie de trafic particulier, comme le trafic web, sur un site. Ceci fait, l'analyste aura une meilleure vision de ce qu'est un trafic normal sur le site de l'entreprise. Il ne faut pas oublier que tout ce que l'on voit peut être vu par un hacker, les mots de passe en clair pour les logins http y compris (password sniffer). Les outils de vigilance réseau peuvent être retournés contre nous. Il est donc préférable de ne pas laisser ces outils à la première personne qui nous les demandent.

2.1.6.2 Recommandations finales sur les IDS

La solution de l'IDS ne résout pas le problème des accès illicites, ni de la corruption des données. Seul un outil de sauvegarde bien géré peut permettre de rétablir la situation. La partie la plus délicate en cas d'intrusion est l'éradication. Cette partie de la gestion des incidents demande le maximum de compétences.
Un IDS ne servira jamais à contrer une attaque DDOS[1] mais permettra d'identifier les hôtes compromis pour éviter une attaque depuis notre site.

2.1.7 Critères de choix

2.1.7.1 Choix de l'IDS et du scanneur de vulnérabilité

Les outils ont été choisis selon les critères suivants:

- Les outils doivent tourner sur Linux, car le but du projet est de protéger des machines Unix ou Linux principalement. Néanmoins, si les outils utilisés pouvaient gérer des machines Windows, cela serait intéressant pour une future utilisation à plus grande échelle de notre solution.

Les outils devaient provenir du monde Open Source pour les raisons suivantes:

- C'est gratuit. Nous ne sommes pas obligés de payer un produit pour faire une recherche qui n'est pas sûre d'aboutir.
- Les produits Open Source sont livrés avec le code. Rien n'est caché.

---

[1] Distributed Denial of Service ou attaque par déni de service distribuée. Attaque visant la destruction d'un service en employant plusieurs machines pour attaquer. Très dangereux.

- Il est possible de faire des modifications si nécessaire.
- La documentation aux grands projets Open Source est complète.
- Nous n'avons pas de formats propriétaires avec lesquels il faut s'adapter, car il n'est pas possible, ni autorisé de modifier un programme commercial (dans la plupart des cas).
- Les mises à jour sont régulières et corrigent les bugs de manière souvent bien plus rapide que les programmes commerciaux dont le but est plus de cacher leurs faiblesses que de les corriger.
- Les mises à jour sont aussi gratuites. Il ne sera pas nécessaire de payer pour mettre à jour les règles ou les vulnérabilités détectées par ces produits.

Les deux outils sélectionnés sont Snort et Nessus. Ces deux programmes sont des leaders dans leur domaine et répondent aux exigences ci-dessus. Il existe même des solutions commerciales basées sur ces produits. Snort et Nessus possèdent tout deux des versions Windows et Linux. Nessus exige au minimum une machine tournant sur Unix ou Linux pour sa partie serveur afin de pouvoir fonctionner. Les versions utilisées seront les versions Linux pour les raisons expliquées ci-dessus.

Une solution Nessus et Snort peut être mise à jour au niveau programme comme au niveau des règles sans autres soucis. De plus les règles sont personnalisables à volonté, seul le travail de maintenance de cette personnalisation pose un problème. Si nous arrivons à automatiser cette mise à jour, alors je pense que cette solution et plus viable économiquement et surtout plus personnalisable à l'environnement de travail. Bien sûr il faudra une personne pour cela, mais il est préférable d'avoir à payer un spécialiste ayant la connaissance de l'infrastructure que de laisser une entreprise externe avoir une vue sur le réseau interne. Dans le cas contraire cette personne ne serait pas efficace (IDS ou rapport incomplet mauvaise configuration des outils de travail dû au manque de renseignements sur le réseau).

2.1.7.2 Problèmes des outils commerciaux

Les IDS commerciaux ont été testés par une grande association qui en a conclu que la technologie utilisée par les fabricants n'était pas meilleure que celle des produits Open Source voire pire. En 2002, sur vingt produits différents, 19 étant des produits commerciaux de haute renommée, aucun de ces IDS n'a convaincu. Pire, ils ont nettement déçu, car certains ont planté ou se sont retrouvés totalement désorientés à lever de fausses alertes. Pire encore certains ont laissé passer de nombreuses attaques et se sont révélés être incapables de suivre le débit des paquets. Certains ne pouvaient plus écrire les paquets dans la base de données, d'autres manquaient des paquets. Les noms de ces IDS n'ont pas été dévoilés, mais il s'agissait des meilleurs produits commerciaux, des leaders du marché. Bien qu'une année soit passée, il n'est pas encore universellement reconnu que les produits commerciaux soient meilleurs que les produits Open Source, malgré les prix exorbitants que pratiquent leurs fabricants de solutions commerciales.

Un IDS commercial faisant une corrélation avec un testeur de vulnérabilités existe, mais n'utilise que leurs produits propriétaires qui sont impersonnalisables ou à un prix excessif. De plus, les mises à jour seront payantes et il est bien possible que l'on soit obligé d'acheter la version suivante du logiciel alors que seules les règles seraient à mettre à jour.

2.1.7.3 Choix du langage de programmation

Afin de trouver le moyen adéquat pour pouvoir parser les fichiers logs, une recherche sur "google" a été faite pour voir s'il n'existait pas un outil correspondant à nos besoins. Comme rien ne correspondait à nos attentes, une première approche a été tentée en utilisant les outils de base sous Linux (les outils de la console Linux). Finalement la recherche des chaînes de caractères sera faite avec grep. Comme grep retourne la ligne contenant la chaîne voulue, il a fallu encore trouver un moyen pour parser cette ligne, car il se peut qu'il y ait plus d'une référence par ligne.
Grep, awk, diff, patch ne semblaient pas pouvoir le faire simplement. Une autre solution aurait été Java, mais nous aurions besoin d'un fichier XML (saxEcho, DOM pour parser du XML). Nessus peut générer un fichier XML mais, il n'est pas bien généré ou lève des erreurs à l'ouverture. Quant à Snort, il ne le fait pas, ou alors peut-être avec un plugin. Il a été finalement décidé de faire ce travail avec Perl qui est un outil réputé pour le travail sur des chaînes de caractères, et de la génération de rapports. De plus, Perl est disponible pour Linux comme pour Windows. Nous avons donc une solution pour les deux environnements de travail.

2.2 Environnement de travail

2.2.1 Description des références utilisées

Références arachNIDS
Les références arachNIDS sont les plus anciennes et sont limitées en nombre (environ 555), elles souffrent aussi d'un moteur de recherche non exempt de défauts. La recherche donne plus souvent du tout ou rien, elle n'est pas efficace. Les mises à jour sont peu fréquentes, seul Snort utilise encore ces références. Elles ne sont pas présentes dans Nessus, ou alors très rarement.

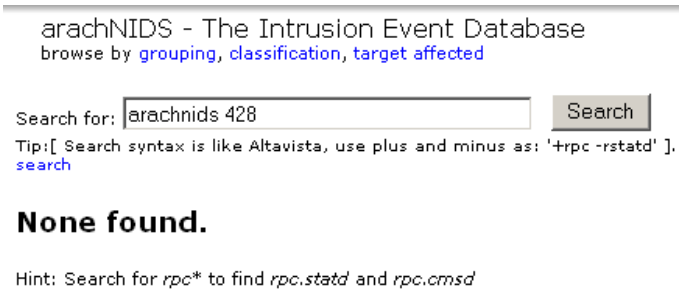


Figure 2-12 Recherche d'une référence arachNIDS sur le site www.whitehats.com

Références BID

Les références BID sont les références les plus rapides à apparaître. Les mises à jour sont quotidiennes sur le site de bugtraq.

Format

Les références BID sont constituées d'un seul nombre correspondant à la vulnérabilité. Il existe aussi des numéros "ButraqMailingRequest" qui correspondent aux CAN. Elles sont constituées d'une suite de nombres qui correspondent à la date de la découverte de la vulnérabilité et d'un autre nombre.

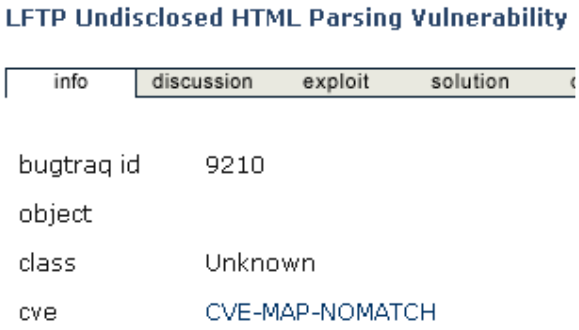


Figure 2-13 Recherche de références sur une vulnérabilité sur le site de Securityfocus

Références CVE

Les références CVE et CAN sont les références les plus complètes, malheureusement elles sont beaucoup plus lentes à apparaître que les BID. Il faut compter en moyenne 2 mois entre l'apparition d'un BID et celui d'un CAN. Un CAN est une référence pour une nouvelle vulnérabilité qui est **CAN**didate à devenir une référence CVE. Les CVE sont des vulnérabilités "confirmées et standardisées" par le Mitre.

Format des CVE
Les références CVE ont un format simple. Une référence CVE est constituée de trois blocs:

- Un premier bloc de trois lettres indiquant si c'est une référence candidate CAN (à l'essai, provisoire, pas standardisée) ou une référence standardisée. Dans ce dernier cas elle commencera par CVE.
- Un deuxième bloc de 4 chiffres indiquant l'année où est apparue la vulnérabilité.
- Un troisième bloc de 4 chiffres indiquant le numéro attribué à la vulnérabilité durant l'année.

Références dans Nessus
Nessus a aussi sa propre numérotation qui fait référence à ses scripts de test.
Ce sont les Nessus ID (ou NASL). Cette numérotation identifie chaque script par un nombre unique. Chose remarquable à relever, chaque script de Nessus correspond à une ou plusieurs références CVE. Il n'a pas été trouvé de script NASL n'ayant pas une référence CVE. Il n'y a pas forcément de correspondance un à un. Il peut par contre y trouver plusieurs références CVE pour un même script. De plus, il n'y a pas de script NASL pour tout CVE. Les rapports contiennent aussi d'autres références comme les BID, arachNIDS et d'autres références sur des pages webs.
Nessus donne parfois aussi des références BID néanmoins, ceci n'est pas systématique.
Les autres liens parfois proposés ne nous sont d'aucune utilité pour la corrélation.

Format
La numérotation NASL est une simple incrémentation du numéro actuel des scripts. Le numéro NASL est unique. En cas de mise à jour, c'est un numéro de version présent dans le descriptif de la vulnérabilité qui est incrémenté.

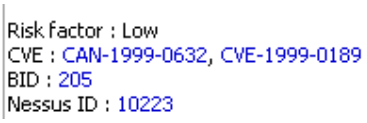


Figure 2-14 Exemple de références que l'on trouve dans un rapport Nessus. La dernière référence est toujours la référence au script NASL.

Références dans Snort

Snort numérote ses règles avec un numéro particulier appelé SID. Les règles Snort sont en général complétées avec des références supplémentaires comme les CVE, CAN, BID ou arachNIDS, mais ceci n'est pas systématique. Des fois, seul le SID est présent. Les références BID sont rares, les CVE, CAN et arachNIDS sont les plus fréquentes.

Format

Les règles Snort ont un numéro unique SID incrémenté depuis le numéro actuel. De plus, selon la catégorie, le numéro relevé dans l'en-tête de la règle correspond à un type particulier: attaque, scan de port, etc. Ce type sera complété par le SID seulement si la catégorie correspond à une attaque.

```
[**] [1:598:10] RPC portmap listing TCP 111 [**]
[Classification: Decode of an RPC Query] [Priority: 2]
10/15-08:43:52.652233 10.192.72.196:1293 -> 10.192.73.121:111
TCP TTL:64 TOS:0x0 ID:2480 IpLen:20 DgmLen:96 DF
***AP*** Seq: 0x9DCB0852  Ack: 0xE5FD00BD  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 76926 90004
[Xref => cve CAN-1999-0632][Xref => arachnids 428]
```

Figure 2-15 Exemple de log d'une règle Snort. La partie à gauche du nom entre crochets contient le SID qui désigne de manière unique la règle.

2.2.2 Snort

Snort, développé par M.Martin Roesch, doit être l'IDS Open Source le plus connu au monde. Il concurrence actuellement encore plusieurs produits commerciaux et il y a même certains produits qui se basent sur ce programme ou son moteur de recherche afin de construire leur solution par-dessus.
Un site complet sur cet IDS se trouve à l'adresse suivante: www.snort.org
Les règles sont quotidiennement mises à jour avec l'apparition de nouvelles attaques. Snort est disponible pour Unix comme pour Windows.
La version Linux est plus récente et plus utilisée car c'est sur cette plateforme qu'elle a été développée. La version que nous allons utiliser est la version Linux 2.0.2. Vous trouverez la version utilisée dans le CD-Rom d'accompagnement dans le répertoire /snort/linux [1].

Fonctionnement de Snort
Snort peu être utilisé en mode sniffer ou comme IDS. Nous l'utiliserons comme IDS. Snort se base sur deux fichiers de configuration principaux qui sont **Snort.conf** et **classification.config**. La détection se fait grâce aux règles qui se trouvent dans le répertoire **/rules**. Dans ce répertoire, on trouve des fichiers contenant plusieurs règles. Chaque fichier correspond en fait à une catégorie d'attaques ou d'événements.

[1] Vous trouverez aussi cette version et la version la plus récente à l'adresse suivante: http://www.snort.org/dl/ . Au moment où j'écris ce rapport, la version 2.0.5 était déjà en libre téléchargement.
La dernière version Windows appelée Winsnort est la 2.0.5 est présente sur le cd au même répertoire mais n'a jamais été utilisée dans ce projet.

Les règles Snort sont divisées en deux parties:

- l'action qui va être entreprise lors de la détection (description, alerte)
- la détection de l'attaque par signature (string à détecter, adresses IP, etc)

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 111 (msg:"RPC portmap listing TCP 111 ";
flow:to_server,established; content:"|00 00 00 00|"; offset:8; depth:4; content:"|00 01 86 A0|";
offset:16; depth:4; content:"|00 00 00 04|"; distance:4; within:4;
reference:arachnids,428;reference:cve,CAN-1999-0632;reference:NASL,10223;
classtype:rpc-portmap-decode; sid:598; rev:10;)
```

Figure 2-16 Exemple de règle Snort

2.2.3 Nessus

Nessus est un scanneur de vulnérabilités. Avec un outil comme Nessus, il est possible de scanner le réseau pour tester des failles connues sur l'ensemble du réseau à la fois, sur une ou plusieurs machines, cela est paramétrable. Couplé à un véritable scanner de ports comme Nmap, il devient possible de tester tous les ports de chaque machine afin de trouver des erreurs de configuration ou de déceler si des services tournent sur des machines alors qu'ils ne devraient pas. Nmap est un outil très puissant qui donne la possibilité de faire des scans de ports furtifs permettant de passer inaperçu aux yeux des IDS. Il faut faire attention avec ces outils, car il est possible de faire tomber une machine avec certains des plugins mis à disposition! Ces attaques dangereuses sont signalées par un signal d'attention (triangle rouge avec un point d'exclamation à l'intérieur).

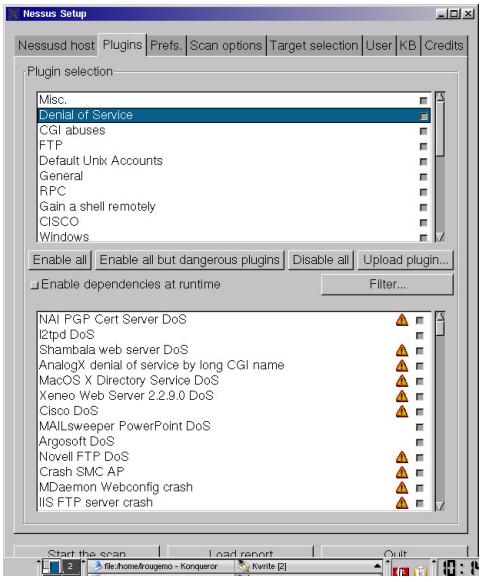


Figure 2-17 On peut remarquer les icônes indiquant les plugins dangereux

Il est possible de choisir les attaques une par une, ou de désactiver un ou plusieurs types d'attaque (cgi, root access, etc.)

On se sert de ce type d'outil afin de jouer au hacker. En effet, il est préférable d'utiliser les mêmes outils que les hackers sur notre réseau afin de voir par nous-mêmes les failles auxquelles nous pourrions être sensibles plutôt que d'attendre que quelqu'un de malveillant transperce nos défenses. Nessus est livré avec une grande panoplie d'attaques, d'attaques par force brute, et de plugins. On peut aussi aisément ajouter d'autres scanneurs de ports ou le coupler avec des outils de force brute, qui couplé avec des dictionnaires qu'il vous faudra ajouter, vous permettra de tester les mots de passe employés dans différents services. Cela permet de tester si un mot de passe est capable de résister au minimum requis, mais cela n'est pas le propos ici.

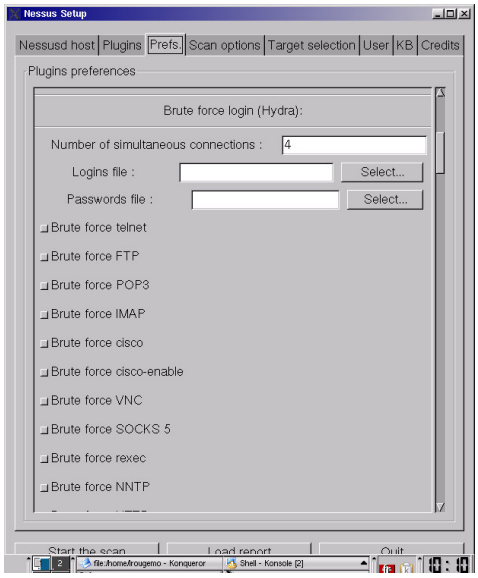


Figure 2-18 Les outils "interdits"

Nessus nous permet de faire un bilan pratiquement complet, un rapport sur toutes nos vulnérabilités actuelles si toutes les options et les plugins ont été cochés. Ce rapport devra par la suite être comparé avec le fichier alert généré par Snort.

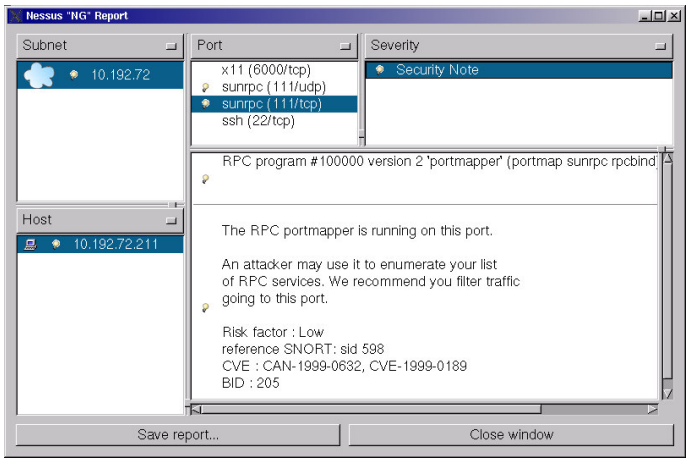


Figure 2-19 Un rapport Nessus

Les plugins

Nessus donne aussi la possibilité de créer ses propres plugins, afin de tester des choses propres à notre réseau. Ces plugins doivent être écrits dans le langage propre à Nessus le NASL. Le NASL est assez proche du langage C. Comme Nessus est un outil Open Source, il est possible de voir le code de chaque plugin afin de s'en inspirer ou de comprendre ce que le plugin fait exactement pour tester la faille en question. Un descriptif détaillé est donné pour tous les plugins sur le site www.nessus.org à la rubrique documentation.

Buffer Overflow in the Workstation Service (828749)	
This script is Copyright (C) 2003 Tenable Network Security	
View the source of this plugin here	
Family	Windows
Nessus plugin ID	11921
CVE ID	CAN-2003-0812

Figure 2-20 Descriptif d'un NASL. Le code y est directement accessible

Les scripts NASL sont constitués de deux parties:

- Une partie descriptive, qui contient les informations sur la vulnérabilité et les références.
- Une partie code qui contient le script testant la vulnérabilité.

Les scripts NASL sont situés dans le répertoire **/usr/local/lib/nessus/plugins** sur une Debian.

Optimisation des scans
Pour l'optimisation de Nessus, il faut avoir une bonne connaissance des applications et des services qui tournent sur les machines afin d'éliminer les vulnérabilités non associées aux programmes utilisés. Les vulnérabilités CGI sont souvent liées à des serveurs web bien particuliers. Il n'a y pas besoin de faire passer les tests pour toutes les applications si l'on utilise qu'un seul de ces serveurs. Une optimisation des scans Nessus a été tentée afin de supprimer toutes les vulnérabilités CGI tournant exclusivement sur Windows. Cela prend énormément de temps et n'est pas véritablement utile si l'on utilise Nessus dans un environnement restreint. Celle-ci devient utile lorsque l'on scanne plusieurs dizaines de machines ou plus. Le temps pour le scan devenant très lent dans ce dernier cas.

Les applications vedettes en matière de vulnérabilité après Microsoft sont, dans le désordre: PHP, Oracle, Perl, Apache. Plusieurs tests ont été réalisés et il en ressort que Nessus ne passe pas forcément du temps à essayer des vulnérabilités CGI si la machine ne dispose pas d'un serveur web. Il est donc inutile d'optimiser Nessus si l'on ne va pas faire tourner certains services dessus, le temps gagné, expérience faite est pratiquement nul. Un commentaire sur ce fait a même été trouvé sur le net.

2.2.4 Perl

Perl peut être un langage intéressant, mais il est plus adapté aux petits scripts qu'aux grands programmes, qui restent réalisables mais avec plus de difficultés.
Il manque un outil gratuit de l'acabit de netbeans ou jbuilder qui n'existe pas pour Perl. Le débogage avec Perl –w et l'ajout de "use strict" au début du fichier laisse le programmeur responsable de tous ses actes.

Il existe des outils pour avoir la coloration syntaxique de Perl sous Linux. En ajoutant quelques lignes sous VIM, il est possible d'avoir ce luxe. Ou alors en utilisant un outil comme Kwrite [1].

---

[1] Perl a été installé sur Windows afin d'utiliser un IDE gratuit: PerlIDE. www.sthomas.net/roberts-perl-tutorial.htm. Ce tutoriel est très bien fait et a permis la compréhension d'une partie des erreurs faites durant les premiers scripts Perl. Ce tutoriel a permis d'apprendre plus sur Perl en un jour que durant toute une semaine avec un livre.

Perl sous Windows
Pour faire fonctionner Perl sous Windows, il faut tout d'abord télécharger le pack d'installation Active Perl à l'adresse suivante, où il vous sera demandé de vous inscrire:
http://www.activestate.com/Products/Download/Register.plex?id=ActivePerl
Ce programme se trouve aussi sur le CD d'accompagnement ci-joint dans le répertoire **/IDE_Perl/activeperl**. Les deux versions, Windows et Linux Debian y sont présentes, bien que seule la version Windows nous intéresse. Pour l'installer, il suffit de suivre les instructions à l'écran.

Description du script Perl
Ce fichier est un programme Perl (à marquer donc comme exécutable avec chmod u+x). Ce programme permet de générer un fichier contenant les références CVE. Il faut donner à ce programme deux fichiers en paramètres. Le premier est le rapport de Nessus au format ASCII (.txt) et le deuxième fichier est le fichier alert généré par Snort. Il faut donner au script le chemin complet pour ces des deux fichiers en paramètres (il faut donc aussi que l'utilisateur ait les droits d'accès, et de lecture sur le répertoire). Le script va générer deux fichiers **outputfile** et **outputfile2** contenant toutes les références CVE des deux fichiers paramètres, puis un troisième fichier **finaloutputfile** contenant la liste des références CVE dont on trouve une occurrence dans chaque fichier. C'est-à-dire qu'il va chercher les correspondances CVE de chaque fichier.

Pour utiliser le script Perl il faut taper la ligne de commande suivante dans le répertoire contenant les trois fichiers:

perl <scriptperl> <rapport_121.txt> <alertcoupe.txt>

Un autre script présent, appelé reinitscrptperl, qui sert à effacer les fichiers de sortie générés par le premier programme. Si ces fichiers ne sont pas effacés avant l'exécution de scriptperl, ils ne seront pas regénérer à nouveau.

Deuxième programme
Le programme Perl est constitué de plusieurs sous-routines qui sont employées dans le programme principal. La sous-routine "ttabsimple" permet de ressortir un tableau de références (une référence par ligne) à partir du rapport de Nessus passé en paramètre. Ce rapport doit être la version en html, car elle contient des informations qui ne sont pas présentes dans la version texte du rapport Nessus. De plus, il est ainsi possible d'utiliser la structure du fichier html par la suite. La sous-routine va parcourir le fichier afin de trouver les motifs des références qui sont pour le moment les références CVE, BID et les numéros des scripts NASL: les Nessus ID. Pour chacun de ces types de références la sous-routine crée un tableau qui sera finalement parcouru afin de retourner un seul tableau au programme principal. Ce tableau permettra par la suite de trouver les règles Snort associées aux vulnérabilités trouvées sur la machine. Pour trouver les motifs, les expressions régulières (ou expressions rationnelles) ont été utilisées. Ce sont des outils puissants, mais pour lesquels il faut aussi être très méticuleux et vigilant.

Il suffit d'avoir mal déterminé les motifs pour nous retrouver séance tenante dans un débuggage "sportif". De plus, il est difficile de faire face aux caractères non imprimables que l'on peut rencontrer à des endroits inattendus dans les fichiers.

Pour utiliser le deuxième programme il faut taper la ligne de commande suivante dans le répertoire contenant les trois fichiers:

perl <essai1.pl> <rapport_121.txt> <alertcoupe.txt>

Ce programme génère un rapport de corrélation en html sur la base du rapport Nessus. Cette version contient encore des bugs (par exemple, les adresses IP ne sont pas vérifiées à la corrélation). Elle a été abandonnée par manque de clarté de l'affichage. De plus, il se peut que l'on ait un rapport Nessus complet sans paquets relevés. Cette version est donc trop lourde pour l'affichage.

Informational	sunrpc (111/tcp)	The RPC portmapper is running on this port. An attacker may use it to enumerate your list of RPC services. We recommend you filter traffic going to this port. Risk factor : Low ref!!! CVE : CAN-1999-0632, CVE-1999-0189 ref!!! BID : 205 Nessus ID : 10223 [**] [1:598:10] RPC portmap listing TCP 111 sid598 [**] [Classification: Decode of an RPC Query] [Priority: 2] 10/15-08:43:52.652233 10.192.72.196:1293 -> 10.192.73.121:111 TCP TTL:64 TOS:0x0 ID:2480 IpLen:20 DgmLen:96 DF ***AP*** Seq: 0x9DCB0852 Ack: 0xE5FD00BD Win: 0x3EBC TcpLen: 32 TCP Options (3) => NOP NOP TS: 76926 90004 [Xref => cve CAN-1999-0632][Xref => arachnids 428]
Informational	sunrpc (111/tcp)	RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is running on this port Nessus ID : 11111

Figure 2-21 Partie du rapport générée par le deuxième programme

2.3 Présentation d'un autre IDS: PRELUDE

Prelude est un IDS un peu particulier. En effet, ce n'est pas un HIDS, ni un NIDS mais les deux à la fois et même plus encore. Prelude est un projet d'IDS Open Source visant à construire un outil de détection d'intrusions modulaire. Prelude est passé d'un projet d'IDS à celui d'hybrid IDS.

Prelude est sensé apporter un plus par rapport à un IDS classique car il récolte les informations des deux IDS et d'autres systèmes de maintenance du réseau comme un firewall ou un routeur. Prelude est un centre de recueil d'informations sur le réseau, il fait donc office de central d'informations pour tous les éléments du réseau ce qui favorise la corrélation entre les événements et qui permet de voir selon différents points de vue,

les problèmes et aussi d'éviter de fausses alarmes avec Snort par exemple. Pour que les éléments puissent communiquer entre eux, il leur faut parler le même langage, un langage particulier le IDMEF[1]. Il faut aussi appliquer des patchs pour Nessus et Snort afin qu'ils deviennent "Prelude compliant".

Prelude utilise une librairie s'appelant LibPrelude (pour laquelle il faut avoir le paquetage libc6 et à jour pour pouvoir la compiler) et qui permet donc de loguer dans une base de données mysql, postgres, db2. Les diverses attaques ou autres informations sont glanées sur le réseau par les sensors. Ces sensors envoient donc les infos au manager qui peuvent ainsi envoyer des ordres aux "counter mesure agents" si demandé. Il existe aussi un front-end web pour pouvoir manager tout ceci de manière plus visuelle.

Prelude ne pourra pas faire mieux la corrélation de Nessus et de Snort. Par contre, il log de manière efficace et garde un historique de ce qui se passe sur le réseau comme par exemple le ferait une solution acid[3]. Le programmeur responsable du projet parle de détection préventive avec un IPS. Mais cela reste théorique. Il est vrai que chaque sensor travaille indépendamment et que l'on centralise les données sur le manager. On peut donc détecter une attaque sur un sous-réseau avant que le pirate n'y accède, s'il doit d'abord passer par plusieurs sous-réseaux.

Il est possible de faire une corrélation grâce à un script Perl se trouvant sur le site Prelude-IDS.org [2]. Ce script va chercher dans la base de données des correspondances avec le rapport Nessus et affiche les informations sur la vulnérabilité trouvée. Pour faire de la corrélation, il faut développer une infrastructure beaucoup plus conséquente que la simple utilisation de Snort et Nessus. En effet, il faut au minimum Prelude constitué d'un manager, un NIDS patché, une base de donnée mysql ou postgres, un serveur web supportant PHP4, les modules PHP4 pour le serveur web, les modules PHP4 pour la base de donnée et un browser. Néanmoins on peut considérer que si l'on doit "monitorer" un grand nombre de machines, réseaux et sous-réseaux, cette solution deviendrait plus intéressante par le fait qu'elle permet de centraliser toutes les données acquises par différents sensors (honeypots, HIDS, NIDS, scanneur de vulnérabilités).

2.3.1 Architecture de Prelude

Prelude est architecturé autour de cinq éléments:

LibPrelude
LibPrelude est une librairie qui permet de faire dialoguer les divers éléments composant Prelude avec un même langage standardisé, le IDMEF (Intrusion Detection Message Exchange Format).

---

[1] Intrusion Detection Message Exchange Format
[3] Analysis Console for Intrusion Database
[2] Un article est présenté sur cette corrélation à la page suivante: http://www.prelude-ids.org/article-forum.php3?id_article=21&id_forum=27

Sensors
Les "sensors" sont des éléments pouvant communiquer aux managers les informations relevées sur le terrain: ce sont les oreilles de Prelude. Les "sensors" sont capables d'envoyer des rapports détaillés sur les alertes survenues sur le réseau.

Managers
Le "central manager" est un processeur de données central pour les "sensors". Il regroupe les données envoyées par les "sensors". Un manager est capable d'envoyer à un manager central ou à un agent de contre mesure, toutes les données qu'il a reçues. Dans un environnement distribué le manager est capable de "forwarder" toutes les informations au central manager.

Counter mesure agents
L'agent de contre-mesures reçoit les informations sur les alertes de la part des managers et prend les contre-mesures associées à cette menace comme stopper la connexion, etc.

Frontend
Point central visuel pour gérer les attaques survenues sur le réseau.

Il est possible de patcher les règles Snort afin de pouvoir les utiliser avec Prelude. Ainsi, il enverra des messages IDMEF [1]. Le fait d'utiliser les règles Snort est un gage de qualité et d'assurance sur la mise à jour. Snort étant actuellement une référence en matière d'IDS. La même chose peut être faite avec Nessus, il sera ainsi possible d'avoir un rapport Nessus en IDMEF que l'on pourra utiliser pour essayer de corréler le tout avec la base de données. Le format des messages IDMEF est un format standardisé.

Les avantages d'une solution hybrid IDS peuvent être nombreux. On répartit la tâche entre plusieurs "sensors" que l'on peut mettre à différents endroits du réseau, afin de tester la vision depuis plusieurs sous-réseaux par exemple. Ces "sensors" peuvent avoir la même configuration sur plusieurs machines ou avoir une configuration plus spécifique en relation avec leur position dans le réseau, ceci afin de favoriser une prise d'information plus générale ou alors plus spécifique et ciblée sur la machine.
Cette configuration peut permettre d'alléger le travail du senseur, abaisser la bande passante utilisée, et minimiser les entrées dans la base de données. Le manager peut être configuré pour prendre des contre-mesures, même si cela n'est pas forcément le plus indiqué. La base de données est un outil efficace pour la recherche lors d'une corrélation ou de recherche de traces d'intrusions. Néanmoins, il faut dire que cette base de données risque de grossir énormément dû aux nombres de "sensors" présents dans le réseau. Déjà avec un IDS il y a une énormité de donnée dont la plupart sont des "false positive", il est donc regrettable de remplir la base avec une multitude d'informations inutiles à moyen et long terme.

[1] Intrusion Detection Message Exchange Format

L'impasse a été faite sur cette solution, car elle augmentait la complexité de la réalisation de notre solution alors que le temps était déjà compté. De plus, il était demandé une solution simple.

2.4 Autre solution testée

Mercredi 29 octobre a été trouvé sur le site www.whitehats.com un lien sur un projet similaire au nôtre. Ce projet appelé *wkr::ids alert verification* [1] est un patch à appliquer à Snort consistant à marquer ou taguer les "logs" de Snort grâce aux NASL. Le principe reste le même, c'est-à-dire une corrélation entre l'IDS et le scanneur de vulnérabilités. Seulement ici, la corrélation se fait de manière active: si le paquet ne correspond pas à une attaque il est supprimé, sinon il est "logué". Il y a donc une perte d'information.

Fonctionnement *d'alert verification*
L'idée est la suivante: avec l'utilisation de Snort, on obtient toujours des fichiers logs très longs. Afin de réduire ces logs, à la détection d'un paquet, le patch lance un script qui utilise Nessus afin de vérifier si la machine est vulnérable ou pas. Si oui, alors, il marque le paquet. La seule différence remarquée est que le fichier alarme contient des références CVE plus longues dans les deux cas (avec et sans marquage). Néanmoins, il est déjà remarquable que le programme renvoie des URL pour les paquets incriminés. Il est pourtant regrettable que chaque paquet de même type ait toujours la même référence CVE (tous les paquets snmp portaient les deux mêmes références CVE-2003-0012, CVE-2003-0013).
Ce projet utilise Snort et Nessus, néanmoins seul Snort est indispensable. Il existe deux versions de ce projet: une patchée et une autre intégrant Snort et le patch directement. Nessus n'est que partiellement installé, le projet n'ayant besoin que des NASL.
La version testée était la version 0.9.2 avec snort 2.0.2 [2]. Elle se trouve dans le répertoire **/programmesTest** du CD-Rom d'accompagnement [3].
Le patch permet d'utiliser le nouveau paramètre **–a** avec les options **mark ¦ suppress**, selon que l'on veuille marquer ou non les paquets. Les références ont été changées, il n'y a plus que les références CVE avec le lien direct sur la page web associée.

[1] Vous trouverez plus d'information à l'adresse suivante:
http://www.cs.ucsb.edu/~wkr/projects/ids_alert_verification/
[2] Vous pouvez aussi télécharger la dernière version à l'adresse suivante:
http://www.cs.ucsb.edu/~wkr/projects/ids_alert_verification/software.html
[3] La version 0.9.2 n'est je crois plus disponible sur le site.

```
[**] [1:615:4] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/29-14:10:14.949447 192.168.233.149:35224 -> 192.168.233.1:1080
TCP TTL:64 TOS:0x0 ID:59033 IpLen:20 DgmLen:60 DF
******S* Seq: 0xF9E7CE11  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 206984 0 NOP WS: 0
[Xref => http://help.undernet.org/proxyscan/]

[**] [105:1:1] spp_bo: Back Orifice Traffic detected (key: 31337) [**]
10/29-14:10:17.115375 192.168.233.149:32985 -> 192.168.233.1:31337
UDP TTL:64 TOS:0x0 ID:10593 IpLen:20 DgmLen:46 DF
Len: 18

[**] [1:1417:2] SNMP request udp [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/29-14:10:17.250547 192.168.233.3:32987 -> 192.168.233.1:161
UDP TTL:64 TOS:0x0 ID:59331 IpLen:20 DgmLen:70 DF
Len: 42
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0013][

[**] [1:1417:2] SNMP request udp [**]
[Classification: Attempted Information Leak] [Priority: 2]
10/29-14:10:17.250547 192.168.233.3:32988 -> 192.168.233.1:161
UDP TTL:64 TOS:0x0 ID:59331 IpLen:20 DgmLen:70 DF
Len: 42
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0013][
```

Figure 2-22 Partie de fichier log Snort avec le plugin. On peut remarquer que le deuxième paquet n'a pas été corrélé. Par la suite, on retrouve toujours les mêmes références. Est-ce un bug ?

Notes sur l'installation d'IDS alert verification:
La version patchée du programme a été installée avec**"./configure && make && make install**". Le fichier ./configure a demandé d'installer quelques fichiers supplémentaires qui appartiennent tous à Nessus 2.0.8. Après acceptation, bien que la version 2.0.7 soit sur la machine, le make a posé problème. En effet, la compilation ne se fait pas jusqu'au bout et lève deux erreurs sur une référence inexistante (execute_script_nasl). Une autre erreur est survenue sur un autre PC à l'installation. Ce fût un problème de librairie (libpcap) mise à jour sans succès (la version la plus récente étant présente sur le PC) ainsi que deux autres librairies associées. Ce n'est que sur une troisième machine standard utilisée par un collègue que l'installation a réussi sans aucun problème! Il semble que cela soit dû à un refus du programme de s'installer avec une version de Nessus installée. Veuillez donc désinstaller Nessus ou alors utiliser la version la plus récente qui a été corrigée entre-temps.

3. Résultats

Le troisième programme donne une page html affichant les paquets suspects relevés par Snort et corrélés avec Nessus dans un cadre. Après chaque type de paquets, la description de la vulnérabilité associée est aussi affichée dans un cadre. Le programme a été testé avec différents fichiers.

Une première fois avec le fichier original de la mesure faite en laboratoire. Le programme a bien affiché l'unique paquet corrélé, comme déjà fait avec la première approche. Il a été ensuite aussi testé avec des fichiers modifiés à la main, afin de déceler si la détection se faisait aussi avec les autres types de références. Les tests ont été passés sans problème.

```
[**] [111:9:1] (spp_stream4) STEALTH ACTIVITY (NULL scan) detection [**]
10/15-08:43:52.614370 10.192.72.196:53999 -> 10.192.72.211:22
TCP TTL:56 TOS:0x0 ID:9374 IpLen:20 DgmLen:60
******** Seq: 0x86938234  Ack: 0x0  Win: 0x400  TcpLen: 40
TCP Options (4) => WS: 10 NOP MSS: 265 TS: 1061109567 0
[Xref => NASL ,10114]

[**] [111:1:1] (spp_stream4) STEALTH ACTIVITY (unknown) detection [**]
10/15-08:43:52.614451 10.192.72.196:54000 -> 10.192.73.123:22
TCP TTL:56 TOS:0x0 ID:58395 IpLen:20 DgmLen:60
**U*P*SF Seq: 0x86938234  Ack: 0x0  Win: 0x400  TcpLen: 40  UrgPtr: 0x0
TCP Options (4) => WS: 10 NOP MSS: 265 TS: 1061109567 0
[Xref => bugtraq 205]

[**] [1:598:10] RPC portmap listing TCP 111  [**]
[Classification: Decode of an RPC Query] [Priority: 2]
10/15-08:43:52.652233 10.192.72.196:1293 -> 10.192.72.192:111
TCP TTL:64 TOS:0x0 ID:2480 IpLen:20 DgmLen:96 DF
***AP*** Seq: 0x9DCB0852  Ack: 0xE5FD00BD  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 76926 90004
[Xref => cve CAN-2003-0001]
```

Figure 3-1 Partie d'un fichier de test. Plusieurs types de références sont employées afin de vérifier leur affichage dans le rapport de corrélation

Nous avons un petit problème d'efficacité. Après avoir récupéré les BID pour les systèmes HP et pour Oracle sur les 6 derniers mois (plus exactement du mois de mai à début novembre). Il y a en tout 80 vulnérabilités pour HP et 24 pour oracle, certaines se recoupant. Le tableau ci-dessous résume les références trouvées. Cela donne environ 7,7% d'efficacité sur cette période critique de 6 mois, estimée comme utile. Ces chiffres semblent bas malgré le fait qu'ils soient évalués pour une période récente. Cela est au-dessous de nos espérances.

Pour la période du 1er mai au 15 novembre 2003

	Pour un environnement HP	Pour un environnement Oracle
Total des vulnérabilités décelées par bugtraq BID	**80**	**24**
Total des vulnérabilités ayant une référence CVE	**39**	**12**
Total des vulnérabilités ayant une référence NASL	**25**	**6**
Total des vulnérabilités ayant un SID	**7**	**1**

Pourcentage de vulnérabilités corrélées:

- au total: **7,7%**
- entre celles disponibles pour Nessus et Snort: 8 SID pour 31 NASL → **25,8%**

Le total des vulnérabilités corrélables correspond au minimum entre le total des vulnérabilités ayant une référence NASL et le total des vulnérabilités ayant une référence SID. Ce qui dans notre cas correspond au total des vulnérabilités ayant une référence SID dans les deux situations: HP et Oracle.

4. Discussion

La détection d'intrusion coûte chère, minimum un ou deux ordinateurs et si possible des ordinateurs puissants. Si nous devions choisir une solution commerciale alors, il faudra compter sur un minimum de 10'000$ rien que pour le logiciel de départ. Le réseau peut avoir besoin de subir quelques modifications et il faut aussi prendre en compte les dépenses liées au personnel. Il faut donc pouvoir justifier cet investissement à la direction, et montrer les bénéfices significatifs que l'on peut en retirer. Par exemple, l'IDS peut aider à mieux paramétrer le firewall en se basant, sur les attaques perpétrées sur le réseau et renforcer les autres défenses afin de les rendre aussi résistantes sur les attaques reçues dernièrement. Si l'on observe le fonctionnement de l'entreprise on risque de trouver des applications ou des situations dans lesquelles il est nécessaire que certaines vulnérabilités soient laissées sur des systèmes. Un exemple flagrant est lorsqu'il faut appliquer un patch de sécurité sur une machine. Cette opération bénigne peut rapidement tourner au cauchemar si cela entraîne par la suite des problèmes.

Il faut aussi pouvoir assurer une certaine dépense déterminée et prévisible afin d'éviter d'engendrer des surcoûts. L'entreprise peut être d'accord de dépenser une certaine somme, mais il faut pouvoir lui assurer qu'il n'y aura pas de surcoûts sur ces dépenses par la suite.

Configuration de Nessus et Snort
La configuration d'outils comme Nessus ou Snort pour l'environnement de travail est un travail de longue haleine qui, comme tout processus de sécurité, doit être répété de manière périodique afin d'affiner sa configuration, sa justesse, et s'adapter aux nouveautés introduites dans le réseau. Il faudrait donc aussi penser à un journal de configuration et de mises à jour strictes, minutieux et efficace. La pire chose qui puisse arriver est que l'on manque des attaques ("false negative").

Le prix à payer
Comme pour tout ce qui a trait à la sécurité, il faut savoir quel prix nous sommes prêts à payer en rapport à ce que cela peut nous apporter en retour. Il est possible, comme expliqué auparavant, de mettre un IDS dans tous les coins de la maison, afin d'être au courant de tout se qui se passe dans tous les recoins de notre réseau, mais ceci ne va pas forcément nous apporter une solution miracle. Au contraire, Il se peut que trop d'information tue "l'information".

Réseaux commutés
La difficulté est multipliée dans un réseau commuté, et par la vitesse auxquelles les informations transitent dans nos appareils. Car les IDS doivent en principe être à l'écoute d'absolument tous nos paquets, chose pratiquement impossible dans un grand réseau à cause du débit de données. C'est pour cela que l'on place plusieurs IDS dans un réseau voir un par machine.

Taille du réseau
La menace vient de l'extérieur comme de l'intérieur. Une étude du FBI a démontré que 80% des attaques étaient faites depuis l'intérieur du réseau. Il faut donc verrouiller depuis dedans comme depuis dehors. Pour cela il faut utiliser un HIDS sur les machines (protection d'intégrité du contenu et login de l'activité sur la machine) ainsi que d'un NIDS que l'on place à l'écoute du réseau ou du sous-réseau à surveiller. Ce qui augmente aussi les coûts.

L'option script Perl convient bien pour un petit réseau avec peu de machines, mais il manque un outil pour centraliser les données de tous les fichiers sortants. De plus, il y aura toujours un certain temps de latence entre le moment où l'on fait appel au script Perl et le moment où à lieu l'attaque. Nous aurons toujours un décalage dans le temps. Il faut préciser quel est le laps de temps pour lequel nous sommes d'accord de passer le scanneur de vulnérabilités, car cela prend énormément de temps dans un grand réseau. Il est envisageable de faire un script afin d'interroger dans une base de données **acid** ou Prelude.

Techniques d'évasions des IDS
Un IDS réalise sont travail en identifiant des signatures. Une signature est une chaîne de caractères fixe. Il est donc facile de reprendre une attaque et de modifier la signature de celle-ci. En réalisant ce travail, il est possible de passer à travers la détection des IDS. Il ne faut donc pas croire qu'en ayant une règle Snort pour une certaine vulnérabilité on soit totalement protégé contre celle-ci. Un IDS est un outil efficace pour la détection des attaques automatiques et non pas pour les attaques personnalisés.

Améliorations à apporter
La prochaine étape serait de pouvoir placer notre corrélation dans un environnement plus complexe, avec un outil comme Prelude ou acid. Le moment où cette corrélation sera intégrée dans une base de données, le travail sera fait de manière plus efficace et plus propice à de futurs changements. Il est plus facile de modifier une requête SQL que de changer tout un programme.

Il serait aussi possible d'adapter cette corrélation à plusieurs autres IDS ou scanneurs de vulnérabilités, voire à d'autres appareils faisant le même travail (honeypots, firewall, etc) Ainsi, en combinant l'utilisation de plusieurs programmes complémentaires, il serait peut-être possible d'augmenter le niveau des détections réalisées, et donc d'augmenter notre taux de corrélation.

Automatisation de la mise à jour
Afin d'avoir un outil véritablement automatisé, il manque encore un moyen de réaliser une mise à jour automatique. Pour cela, il faudrait avoir une base de données ou un fichier contenant toutes les références combinées pour ne pas avoir à les saisir à la main lors des mises à jour. Il faudrait aussi avoir un moyen sûr de recevoir les corrections et les dernières vulnérabilités.

Combler le vide
Comme les dernières vulnérabilités sont lentes à être implémentés dans les outils de détection, il faut trouver un moyen d'accélérer ce travail. Certaines entreprises vendent des règles faites par leurs soins ou opèrent des audits de sécurité. Il est possible de faire appel à ces spécialistes pour combler au plus vite les lacunes de notre environnement de travail. Cela devrait rester moins cher qu'une solution totalement commerciale.

5. Conclusion

L'automatisation de la détection d'intrusions n'en est qu'à ses débuts. Nous pouvons remarquer qu'il y a encore beaucoup de chemin à parcourir avant d'avoir un système fiable et complet. Pour l'instant, il n'est pas possible de se fier complètement aux IDS dont la base de données des signatures est encore trop restreinte et comporte encore des

faiblesses dont les "false positive", et "false négative" en sont la preuve. Même avec une corrélation il subsiste encore beaucoup de brèches qu'il sera difficile de combler.

Il est vrai que notre corrélation par les références donne des résultats médiocres. Néanmoins, malgré cette faiblesse, il est à relever q'une partie du travail peut être automatisée sans un effort surhumain, par une mise à jour régulière des règles Snort. Même si ce n'est que 7,7% des attaques spécifiques à une machine qui peut être automatisée, cela donne la possibilité d'écourter le travail de l'employé chargé de cette tâche et surtout, il permet de mettre en évidence les choses importantes tout de suite. Cela permet de réduire le temps entre l'attaque et la détection de l'intrusion. Il sera possible de répondre ou de parer aux attaques de manière plus rapide. Il faut ajouter que si l'on se place du côté de ce qui est détectable (mis actuellement à disposition) nous arrivons quand même à 25%.

Ce travail de diplôme m'aura permis de connaître le monde "merveilleux" des IDS et leur fonctionnement, comme une partie de la complexité de la détection des intrusions. La sécurité informatique reste un domaine aussi passionnant qu'extrêmement difficile à cerner. Ce travail m'aura aussi permis de découvrir Perl et de faire plus ample connaissance avec le html que je ne connaissais pratiquement pas. Linux est un OS remplit de surprises, bonnes comme moins bonnes. Il a été très intéressant de pratiquer quelques commandes **bash**, et de découvrir des moyens détournés pour faire fonctionner, installer des programmes.

6. Bibliographie

Livres:

Détection des intrusions réseaux Collection Référence CampusPress S.Northcutt, J.Novak, D.McLachlan

Sites webs d'intérêt:

Site officiel de Snort	www.snort.org/
Site official de Nessus	www.nessus.org
Distributed Intrusion Detection System	www.dshield.org/
LIDS Project - Secure Linux System	www.lIDS.org/
BlackICE PC Protection et Real Secure ISS	www.bvrp.fr
Firewall avec base de données pour logs	www.dshield.org/howto.php
Intrusion nIDS company	www.intrusion.com/
Intrusion detection FAQ	www.sans.org/resources/idfaq/
Talisker's Network Security Resource	www.networkintrusion.co.uk/
Emerald and nides projects	www.sdl.sri.com/projects/emerald/
Niksun IDS product	www.niksun.com/index.php?id=203
netalert firewall+nIDS	www.netalert.ch/
SPECTER Intrusion Detection System	www.specter.com/
NSWC SHADOW INDEX	www.nswc.navy.mil/ISSEC/CID/
MINDS - Minnesota Intrusion Detection System	www.cs.umn.edu/research/minds/
Lien sur des IDS et les honeypots	www.honeypots.net/

Bullseye IDS from compunet.com
www.compunet1.com/security/bullseye.asp

IDS part1 doc Linuxfocus
www.Linuxfocus.org/English/May2003/article292.shtml

Cisco Intrusion Detection
www.cisco.com/

www.cisco.com/warp/public/cc/pd/sqsw/sqIDSz/index.shtml
www.cisco.com/en/US/products/sw/secursw/ps2113/

FAQ: Network Intrusion Detection Systems
www.robertgraham.com/pubs/network-intrusion-detection.html/

Best practices IDS
www.lancope.com/XFRM.asp?RTN=Data/IDSbest&XML=products.xml&XSL=products.xsl

Landcope hybrid IDS product
www.lancope.com/XFRM.asp?RTN=Data/G1&XML=products.xml&XSL=products.xsl

Security metrics products & whites papers
www.securitymetrics.com/securitymetricsappliance.adp

Demarc IDS product Pure secure
www.demarc.com/products/puresecure/screenshots.html

dragon IDS suite
www.intrusion-detection-system-group.co.uk/index.htm/

Deploying an Effective Intrusion Detection System
www.networkmagazine.com/shared/article/showArticle.jhtml;jsessionid=L4543CHFMLQPYQSNDBCCKHQ?articleId=8702894&pgno=2

ISS real secure product
www.allstream.com/products/infrastructure/security/detection/realsecure.html
Test d'une vingtaine d'IDS
www.nwfusion.com/techinSIDer/2002/0624security/0624security.html

The Dragon Intrusion Detection Software System
www.securityware.co.uk/intrusion-detection/

Ciac.com documents
www.ciac.org/cgi-bin/index/documents

Securepoint products
www.securepoint.cc/en/products-IDS.html

Snare IDS www.intersectalliance.com/projects/

Autonomous agent for intrusion detection
www.cerias.purdue.edu/homes/aafid/

TriSentry, a Unix Intrusion Detection System
www.onlamp.com/pub/a/onlamp/2002/11/14/trisentry.html

Quarry Announces iQ-IDS Network-Based Intrusion Detection System
www.quarrytech.com/news/pr_20030825.shtml

Symantec Releases Decoy-Based Intrusion Detection System
www.symantec.com/press/2003/n030623b.html

Computer intrusion detection system with configurable components
www.globaltechnosCAN.com/19thDec-25thDec01/intrusion_detection_system.htm

IntruShield 4000 Intrusion Detection System Named a 2003 Winner by Network Computing For Well-Connected Award
www.networkassociates.com/us/about/press/sniffer_technologies/2003/20030501_152511.htm

7. Annexe A: Installation de Nessus

Sous Debian, il est préférable d'utiliser l'outil de mise à jour maison **apt.**
Pour utiliser **apt** faut avoir une connexion internet et avoir défini les liens dans le fichier sources.list. Si cette version d'installation ne fonctionne pas alors utiliser la méthode standard: **./configure && make && make install**.

Dans un terminal:

- tapez : **apt-get install nessus nessusd**.
- Déplacez-vous dans le répertoire d'installation de nessus, qui est normalement **/usr/bin** ou **/usr/local/bin**.[1]

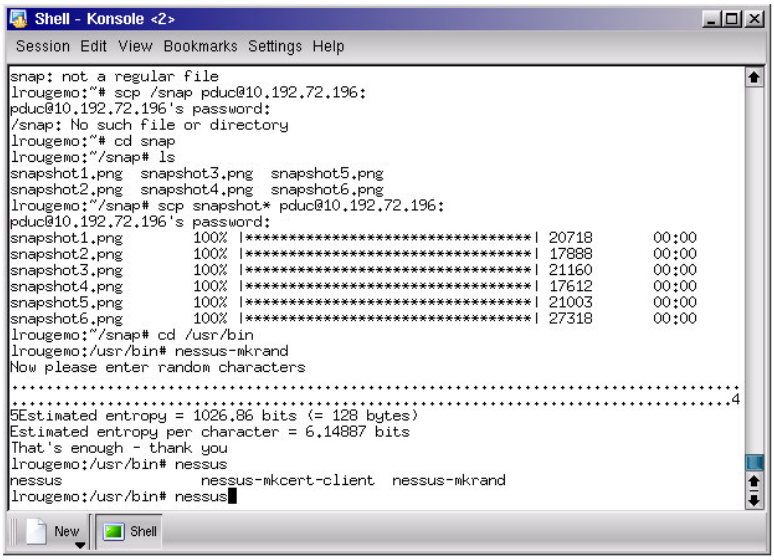


Figure 7-1 Génération du nombre aléatoire par nessus_mkrand

[1] Pour localiser un fichier vous pouvez tapez: locate *lenomdufichier* ou find / -name *lenomdufichier*

- Générer un nombre aléatoire en appelant **nessus-mkrand**. Il vous sera demandé de taper aléatoirement des caractères.

- Par la suite, il faut encore créer un utilisateur avec la commande **nessus-adduser**, et suivre les instructions à l'écran. Il est préférable de choisir l'option passphrase afin de ne pas avoir à faire de certificats. Notez bien le login et le mot de passe, ils vont vous servir tout de suite.

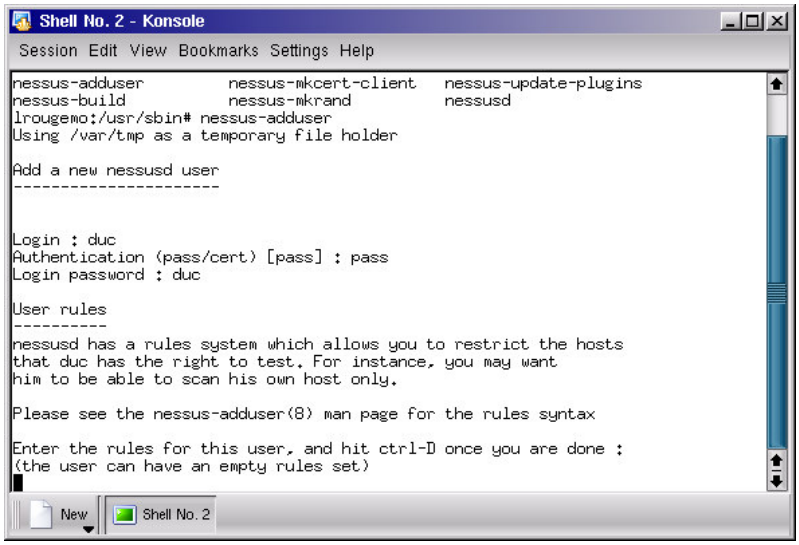


Figure 7-2 Ajout d'un utilisateur

- Lancer la partie serveur. Pour cela, il faut se placer dans le répertoire contenant nessusd. Et taper: **./nessusd &**
  Le programme nessusd est généralement placé dans /usr/sbin ou /usr/local/sbin.

- Lancer la partie client de Nessus en se plaçant dans le répertoire la contenant avec la commande: **./nessus &**
  Il vous sera demandé de vous "loger" avec le "login" et le mot de passe de l'utilisateur créé précédemment. Vous voilà prêt pour l'utilisation de nessus.

Recommandations
Après avoir modifié un NASL ou fait une mise à jour de celui-ci, il faut impérativement arrêter nessusd (**killall nessusd**) et le relancer. Sinon, les changements effectués ne seront pas pris en compte.

Si l'installation échouait, il reste la possibilité de télécharger les fichiers sur le site.
Par la suite:

- Déplacez-vous dans le répertoire où se trouvent les fichiers téléchargés
- Décompacter ceux-ci avec: **tar zxvf** *lenomdufichier*
- Utiliser les commandes: ./configure && make && make install

8. Annexe B: Installation de Snort version 2

Il ne faut pas actuellement utiliser **apt** pour l'installation de snort2 car la version stable de Debian utilise encore snort 1.8. Il faut donc:

- Télécharger les fichiers sur le site de Snort
- Se placer dans le répertoire contenant les fichiers téléchargés
- Décompacter avec: **tar zxvf** *lenomdufichier*
- Puis, taper: **./configure && make && make install**

Recommandations:

Pour utiliser snort, il faut:

- se positionner dans le répertoire le contenant: cd **/usr/local/bin**
- Tapez ./**snort** suivi des options choisies.

Pour obtenir les options tapez: **snort –help**

La ligne de commande complète utilisée durant nos mesures est la suivante:
./snort -a full -h 10.192.72.211/32 **-c** /usr/local/bin/snort.conf **-l** /usr/local/bin/log
Il faut adapter cette commande à votre environnement!

-h IPdelamachineAsurveiller/masqueDeReseau
-c /répertoireOuSeTrouveSnort
-l /répertoireOuOnVeutRetrouverLesLogs

Faites attention! Il faut impérativement supprimer les fichiers présents dans le répertoire des logs avant d'utiliser Snort. Pour la bonne raison que Snort n'écrase pas ces fichiers. Vous aurez donc toujours les mêmes fichiers dans le cas contraire.

Il faudra modifier le fichier **snort.conf** si:

- Il y a eu modification du chemin d'accès aux règles (répertoire **/rules**) et au fichier **classification.config**

- Snort ne trouve pas ces fichiers lors de son exécution.

9. Annexe C: Récupération d'informations sur les dernières vulnérabilités

1) Se rendre sur le site securityfocus à la page :
http://www.securityfocus.com/BID/vendor/

2) Sélectionner le vendeur "*vendor"* pour le matériel désiré:
Dans notre cas l'étude c'est portée sur la base de données Oracle et sur les machines HP-UX.[1]

3) Depuis ici, il faut sélectionner les vulnérabilités et prendre note des BID ou toute autre référence utile pour la suite. Dans notre cas, il est utile de récupérer les BID, CVE, CAN. Voici un lien menant directement à un exemple de vulnérabilité pour HP-UX
http://www.securityfocus.com/BID/5995

4) Passer sur le site de Nessus www.nessus.org à la page de recherche:
http://cgi.Nessus.org/plugins/search.html

5) Introduire le numéro BID ou CVE, CAN relevé sur le site de securityfocus afin d'obtenir le numéro du script NASL (le Nessus ID), et savoir s'il existe ou non d'autres références supplémentaires comme une référence CVE. Il est fortement conseillé d'essayer avec les références CVE que l'on ait ou pas trouvé quelque chose à partir des BID. On a quelques fois des surprises. En effet, il est possible de trouver des références supplémentaires grâce à cette recherche faites-là deux fois !

6) Lorsque l'on a les numéros BID, CVE et les Nessus ID on peut alors passer dans le répertoire des règles Snort.
A noter qu'il est aussi possible sur le site www.snort.org de faire une recherche à partir des SID ou des messages de vulnérabilités (le nom donné aux vulnérabilités). Le nom n'étant souvent pas le même entre Snort, Nessus CVE et BID il sera d'autant plus difficile de si retrouver de cette manière-là, c'est pourquoi il est recommandé de ne pas utiliser le nom des vulnérabilités

[1] Vous trouverez sur le cd les fichiers vulnérabilitésOracle.txt et vulnérabilitésHP.txt qui correspondent aux vulnérabilités décelées par bugtraq pour les 6 derniers mois. Le site garde les informations sur les vulnérabilités pour environ les 4 dernières années. Cela est intéressant pour faire une base de données mais inutile pour notre travail, car la vulnérabilité a été patché par le fabricant depuis longtemps (ou alors il faut vraiment changer de fabricant !)

De plus, il faut apporter une grande attention aux sites webs proposant des SID, car certains de ces sites proposent des références SID qui ne correspondent pas aux SID de Snort. En effet, ces références correspondent à leur propre logiciel et matériel. Il faut donc toujours bien vérifier quel est le type de SID, matériel, logiciel dont on parle!

7) Lorsque l'on a récupéré les références Snort et Nessus il suffit de les saisir à la main. Pour Snort, il faut ajouter : **reference:NASL,lenumerodescriptNessus;** à la suite des références déjà mises en place (normalement il y a toujours au moins une référence) afin de pouvoir savoir exactement de quelle vulnérabilité il s'agit.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 111 (msg:"RPC portmap listing TCP 111 ";
flow:to_server,established; content:"|00 00 00 00|"; offset:8; depth:4; content:"|00 01 86 A0|";
offset:16; depth:4; content:"|00 00 00 04|"; distance:4; within:4;
reference:arachnids,428;reference:cve,CAN-1999-0632;reference:NASL,10223;
classtype:rpc-portmap-decode; sid:598; rev:10;)
```

Figure 9-1 Règle Snort modifiée.

10. Annexe D: Recherche de règles Snort selon les références

Afin de pouvoir récupérer les SID de Snort grâce à un "**grep** uneReference" il faut se placer au-dessus du répertoire **/rules** (dans mon cas **/usr/local/lib/nessus**). Puis, il faut utiliser la ligne de commande ci-dessous:

grep -H -R CAN-2003-0386 ./plugins

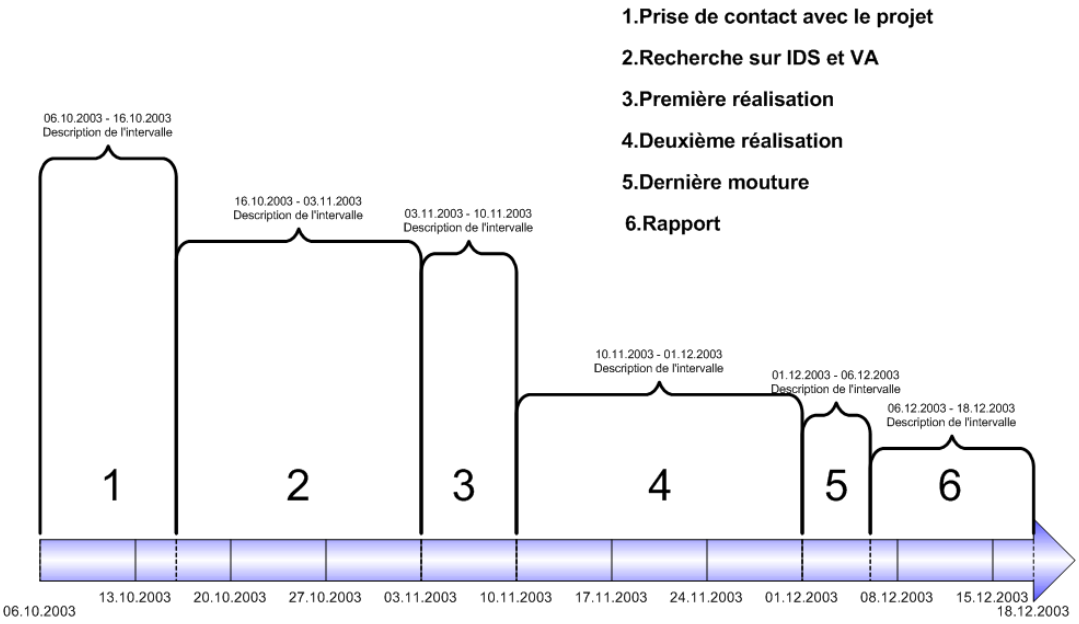
Cette commande permet de savoir s'il existe une règle Snort ayant la référence cherchée, ainsi que le fichier qui la contient.

Réponse à la commande ci-dessus:

./plugins/openssh_rev_dns_lookup_bypass.nasl: script_cve_id("CAN-2003-0386");

Recommandations
Il faut signaler que le format .txt des rapports Nessus ne propose pas les numéros des NASL (Nessus ID), le format html oui. Utilisez donc le format html des rapports Nessus pour garder un maximum des informations d'origine.

11. Annexe E: Planning



Yverdon, le 18 décembre 2003

Pierre Duc