Sécuriser son poste Linux

Olivier Hoarau (olivier.hoarau@funix.org)

V1.4 du 3.11.08

1 Historique du document

V1.4 3.11.08 gros toilettage

V1.3 06.01.04 Passage à sara 4.2.7, nmap 3.48, nessus 2.0.9, libpcap 0.8.1, ethereal 0.10.0a et snort 2.1.0

V1.2 16.03.03 Rajout d'un paragraphe sur Iplog
Changement de version (Sara 4.1.4b, Nessus 1.2.6, snort 1.9.1 , Ethereal 0.9.11)

V1.1 15.09.02 Rajout d'un paragraphe sur les outils ethereal et snort pour sniffer son réseau
Passage à Sara 4.0.1
Passage à nmap 3.00
Passage à nessus 1.2.5

V1.0 22.02.02 Création du document

2 Préambule

Ce document présente les moyens de sécuriser son poste Linux.
La dernière version de ce document est téléchargeable à l'URL http://www.funix.org. Ce document peut être reproduit et distribué librement dès lors qu'il n'est pas modifié et qu'il soit toujours fait mention de son origine et de son auteur, si vous avez l'intention de le modifier ou d'y apporter des rajouts, contactez l'auteur pour en faire profiter tout le monde.
Ce document ne peut pas être utilisé dans un but commercial sans le consentement de son auteur. Ce document vous est fourni "dans l'état" sans aucune garantie de toute sorte, l'auteur ne saurait être tenu responsable des quelconques misères qui pourraient vous arriver lors des manipulations décrites dans ce document.

3 Sécuriser son poste linux

3.1 Présentation

Le but de cette page est de présenter le moyen de sécuriser son poste linux, il n'a pour vocation de présenter une configuration sécurisée à 100% mais suffisamment protégée pour affronter le "chaos" d'internet.

3.2 Attaques possibles

3.2.1 Présentation

Avant de sécuriser votre poste, voici une liste non exhaustive des méthodes les plus classiques pour pénétrer un système, les détails pour exécuter ses attaques ne sont pas exposés, pour cela vous trouverez un tas de sites sur le net qui explique ça très bien.

3.2.2 Compte utilisateur

La première étape pour essayer de pénétrer sur un système est tout simplement de se loguer comme un simple utilisateur, pour cela il faut connaître un login, anciennement beaucoup de systèmes UNIX disposait de comptes génériques par défaut, du style "**guest**" pour invité, un hacker essayera toujours ses comptes par défaut.
Pour éviter cela, vérifiez votre fichier **/etc/passwd** et supprimer tous vos comptes génériques utilisateurs (pas les comptes systèmes !!).

Veillez à ce qu'aucun utilisateur n'ait pas de password et encore moins un password identique au login (très courant !!).

3.2.3 Les outils "r" (rsh, rlogin, ...)

Attention à ses outils, ils utilisent un système d'authentification relativement rudimentaire et sont connues pour constituer une faille de sécurité. Qui plus est avec ces outils on peut mettre en place des relations de "confiance" entre les machines, c'est à dire qu'on pourra accèder à une machine en utilisant les r-outils sans avoir à entrer de mot de passe, par conséquent si un système est hacké, cela signifie que tous les autres tomberont aussi. Sachez aussi que les r-outils ne disposent pour la plupart d'aucun mécanisme de loging, donc aucun moyen d'avoir l'historique de leur utilisateur. Les fichiers permettants d'établir les relations de confiance sont **.rhosts** ou **/etc/hosts.equiv**.
Les outils r principaux sont:
- **rsh** ouvre un shell à distance (pour lancer une commande par exemple), si vous voulez un outil équivalent sécurisé, tournez vous vers **SSH**.
- **rlogin**, équivalent à **telnet**, si un utilisateur place un fichier **.rhosts** dans sa homedirectory contenant **obelix**, n'importe qui sur la machine **obelix** pourra se connecter sur son compte sans avoir à donner de mot de passe. Si vous possèdez un **/etc/hosts.equiv** contenant un +, n'importe qui aura accès à votre machine.
- **rexecd**, c'est le serveur pour accepter des requêtes de **rexec**, permet de lancer des commandes à distance. Le serveur ne logue aucun échec de connexion, par conséquent vous pouvez essayer une tonne de mot de passe sans que l'administrateur de la machine visée sans rende compte.

3.2.4 FTP

Ne faites pas tourner un serveur **FTP** anonyme à moins que vous sachiez ce que vous faîtes, sachez que les daemons **FTP** sont connus pour présenter pas mal de problèmes de sécurité, un serveur **FTP** mal configuré peut très bien servir de passerelle à un hacker pour attaquer votre machine bien sûr mais aussi d'autres machines.

TFTP (Trivial File Transfer Protocol) est un service **FTP** simplifié, il est utilisé notamment pour le boot des terminaux X. Il ne demande aucune authentification, n'importe qui peut se connecter et lire ce qu'il veut.

Je vous conseille donc de désactiver tout ce qui tourne autour de **FTP**.

3.2.5 Outils de stats

Des outils comme **finger**, **systat**, **netstat**, **rusersd**, **rwhod**, ... servent à avoir des informations sur le système (sur les utilisateurs, stats de réseau, process qui tournent, ...). Pour la plupart ce sont des daemons qu'on peut interroger de l'extérieur, il devient alors très facile pour un hacker d'obtenir un max d'informations sur votre système, qui lui permettront de mieux cibler ses attaques.
Par exemple avec **systat** et **netstat** qui tournent sur votre système, un hacker peut visualiser vos process actifs ainsi que la configuration réseau. Des outils comme **rusersd** et **rstatd** permettent à un hacker de visualiser les gens qui sont logués à un moment donné. **Finger** permet de connaître les logins existants sur la machine.

3.2.6 NFS

Les versions précédentes de **NFS** comportaient des trous de sécurité, vous devez veiller à posséder la dernière version. Vous ne devez en aucun cas exporter vos systèmes de fichiers vers le monde entier, vous devez toujours restreindre l'accès à un nombre limité de machines et préféré toujours la lecture seule que la lecture/écriture. N'exportez pas plus que nécessaire, c'est à dire exporter **/usr/local/public** plutôt que / tout entier. Vous pouvez faire en sorte que le root de la machine distante qui monte votre répertoire puisse avoir les mêmes droits que celui de la machine locale, c'est à éviter (ce n'est pas le cas par défaut). Enfin **NFS** doit être utilisé strictement en interne sur un réseau local et en aucun cas entre des machines sur internet.

3.2.7 Serveur web

Faites très attention à vos scripts **cgi-bin**, un défaut dans l'un d'entre eux et c'est la porte ouverte à votre système. Si vous mettez un serveur web en place, supprimez donc tout ce qui se trouve sous le répertoire des **cgi-bin** et mettez y que vos scripts **CGI** dont vous soyez absolument sûrs. Ne faites pas confiance aux script **CGI** que vous récupérez sur internet.
En plus des scripts **CGI**, il y a bien d'autres failles dans un serveur **HTTP**, que ce soit au niveau d'une mauvaise configuration, des serveurs bogués, ... même les logs peuvent être un problème, s'ils sont visibles d'internet, notamment ceux contenant les erreurs (page 404), qui peuvent contenir les informations sensibles comme les login des utilisateurs.

3.2.8 Serveurs Mail

Attention aux serveurs **SMTP**, il y a eu par le passé pas mal d'attaques du à des failles dans **sendmail**, **smail** et d'autres, veillez à toujours faire tourner la dernière version et n'hésitez pas à appliquer les patchs. Désactivez tout serveur **SMTP** si vous ne vous en servez pas.
Pour les serveurs **POP/IMAP**, sachez qu'il y a eu sur ces serveurs un bogue (buffer overun) qui faisait qu'on pouvait lancer des commandes en tant que root à distance. Faites une mise à jour, ou désactivez les. Certaines serveurs **POP** ne loguent pas les erreurs de logins, donc

n'importe qui peut essayer une tonne de password sans que vous vous rendiez compte. Enfin **POP/IMAP** manipule les mots de passe en clair, on peut très bien sur le réseau intercepter une requête **POP** (par sniffing) et récupérer le login et le mot de passe. Préférez les serveurs **POP** sécurisés (APOP, ...).

3.2.9 Autres

Parmis les outils réputés potentiellement peu sûrs, même si les nouvelles versions tendent à corriger petit à petit les défauts, citons **DNS**, **Samba**, services **RPC**, **portmapper**, et j'en oublie.

3.3 Améliorer la sécurité

3.3.1 Installation de la distribution

La sécurisation du poste commence à l'install, je vous déconseille de choisir les configs standards proposés dans les distribs comme **RedHat** ou **Mandrake** avec **WorkStation** (Station de travail), **Server** (Serveur), le mieux est de choisir **Custom** (personnalisé) pour avoir un oeil sur les outils qui seront installés sur votre système. L'idée est d'installer le minimum d'outils, la règle la première si vous ne voyez pas à quoi peut servir un outil est de ne pas le choisir, il faut vous dire que de toute manière vous aurez toujours la possibililité de l'installer plus tard. En résumé moins vous avez d'outils, moins vous avez de trous potentiels de sécurité.
Au moment d'arriver au partitionnement du système, je vous conseille de créer plusieurs partitions pour bien séparer le système, des données, par exemple:

/ 150Mo
/usr 1Go au moins
/var 400Mo
/home 50Mo par utilisateurs au moins
swap 2*taille de la RAM au moins
/usr/local au moins 500Mo (pour stocker vos applis)

Une fois l'installation terminée, suivant votre distrib allez sur le site correspondant, dans la rubrique support vous devriez trouver les derniers packages corrigeant les trous de sécurité. Ces patchs sont indispensables pour sécuriser correctement un système, vous devez surveiller régulièrement leur parution. Sachez qu'il existee des mailing-listes à ce sujet qui vous préviendront automatiquement de la parution d'un nouveau patch.

Beaucoup de distributions dispose d'outils (**up2date** pour la RedHat, **urpmi** pour la Mandriva) qui permettent d'upgrader automatiquement le système.

Je vous conseille de tenir à jour un cahier d'administrateur où vous noterez chacune de vos manips systèmes, ça peut se révéler un brin écolier, voire fastidieux, mais ça peut se réléver extrèmement utile dans certains cas. Un simple fichier texte (droit 400 proprio root) avec un copier coller des commandes (et les résultats qui vont avec), et quelques annotations devrait faire l'affaire.

3.3.2 Eliminer les services inutiles

Maintenant que le système est installé et que vous avez patché les packages defectueux, passons maintenant à la sécurisation propre du système. Dans un premier temps on va désactiver tous les services inutiles qui tournent sur la machine.

Si vous utilisez inetd

On va d'abord travailler sur **inetd**, **inetd** est un "super daemon" c'est à dire qu'il permet à lui tout seul de lancer tout un tas d'autres daemons (ou services), il est configurable à partir du fichier **/etc/inetd.conf**. Par défaut il comprend un certain nombre de services activés dont vous n'avez pas forcément besoin, à l'inverse certains services désactivés peuvent vous être utiles. On décomment un service en lui mettant un # devant. Voici un exemple de fichier **inetd.conf** :

```
#
# inetd.conf    This file describes the services that will be available
#               through the INETD TCP/IP super server.  To re-configure
#               the running INETD process, edit this file, then send the
#               INETD process a SIGHUP signal.
#
# Version:      @(#)/etc/inetd.conf     3.10    05/27/93
#
# Authors:      Original taken from BSD UNIX 4.3/TAHOE.
#               Fred N. van Kempen, <waltje@uwalt.nl.mugnet.org>
#
# Modified for Debian Linux by Ian A. Murdock <imurdock@shell.portal.com>
#
# Modified for RHS Linux by Marc Ewing <marc@redhat.com>
#
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#
# Echo, discard, daytime, and chargen are used primarily for testing.
#
# To re-read this file after changes, just do a 'killall -HUP inetd'
#echo   stream  tcp     nowait  root    internal
#echo   dgram   udp     wait    root    internal
#discard        stream  tcp     nowait  root    internal
#discard        dgram   udp     wait    root    internal
#daytime        stream  tcp     nowait  root    internal
#daytime        dgram   udp     wait    root    internal
#chargen        stream  tcp     nowait  root    internal
#chargen        dgram   udp     wait    root    internal
#time   stream  tcp     nowait  root    internal
#time   dgram   udp     wait    root    internal
#
# These are standard services.
# Attention à ces deux services ce sont deux gros trous potentiels de sécurité
# j'ai désactivé le serveur ftp car j'en ai pas l'utilité
# ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

```
#
# Shell, login, exec, comsat and talk are BSD protocols.
# r-outils et autres à désactiver
#
#shell  stream  tcp    nowait  root    /usr/sbin/tcpd  in.rshd
#login  stream  tcp    nowait  root    /usr/sbin/tcpd  in.rlogind
#exec   stream  tcp    nowait  root    /usr/sbin/tcpd  in.rexecd
#comsat dgram   udp    wait    root    /usr/sbin/tcpd  in.comsat
#talk   dgram   udp    wait    root    /usr/sbin/tcpd  in.talkd
#ntalk  dgram   udp    wait    root    /usr/sbin/tcpd  in.ntalkd
#dtalk  stream  tcp    waut    nobody  /usr/sbin/tcpd  in.dtalkd
#
# Pop and imap mail services et al
# serveur pop3 pour réseau local activé, si poste isolé à désactiver
#pop-2  stream  tcp    nowait  root    /usr/sbin/tcpd ipop2d
pop-3   stream  tcp    nowait  root    /usr/sbin/tcpd  ipop3d
#imap   stream  tcp    nowait  root    /usr/sbin/tcpd imapd
#
# The Internet UUCP service.
#
#uucp   stream  tcp    nowait  uucp    /usr/sbin/tcpd  /usr/lib/uucp/uucico -l
#
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers." Do not uncomment
# this unless you *need* it.
#tftp   dgram   udp    wait    root    /usr/sbin/tcpd  in.tftpd
#bootps dgram   udp    wait    root    /usr/sbin/tcpd  bootpd
#
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers."  Many sites choose to disable
# some or all of these services to improve security.
# outils de stats réseau à désactiver
#
#finger stream  tcp    nowait  root    /usr/sbin/tcpd  in.fingerd
#cfinger stream tcp    nowait  root    /usr/sbin/tcpd  in.cfingerd
#systat stream  tcp    nowait  guest   /usr/sbin/tcpd  /bin/ps -auwwx
#netstat        stream  tcp    nowait  guest   /usr/sbin/tcpd  /bin/netstat -f inet
#
# Authentication
#
# auth   stream  tcp    nowait    nobody    /usr/sbin/in.identd in.identd -l -e -o
#
# linuxconf stream tcp wait root /bin/linuxconf linuxconf --http
#swat     stream  tcp    nowait.400      root /usr/sbin/swat swat
```

Si vous utilisez xinetd

Si vous utilisez **xinetd**, pour les services inutiles se trouvant sous **/etc/xinetd.d** il suffira de rajouter le paramètre **disable=yes**, exemple avec **daytime**

```
service daytime
{
   type    = INTERNAL
   id      = daytime-stream
   socket_type= stream
   protocol    = tcp
   user    =    root
   wait    =    no
   disable    =    yes
}
```

Relancez **xinetd**

/etc/rc.d/init.d/xinetd restart

3.3.3 Eliminer les scripts de lancement inutiles

Maintenant on va aller jeter un coup d'œil dans les répertoires **/etc/rc.d/rcX.d** qui contiennent des liens vers les scripts de lancement de services, on va désactiver les services inutiles. Si vous démarrez à l'état de marche 5, voici ce que vous pourriez trouver dans le répertoire **/etc/rc.d/rc5.d**

S05apmd Utile uniquement pour les portables, sert à gérer l'autonomie d'une batterie, à virer si vous avez un poste de bureau
S09pcmcia pour activer les services liés au **PCMCIA**, à virer si pas de **PCMCIA**
S10network active les interfaces réseau (**eth0**, ...)
S11portmap Utile si vous utilisez des services **RPC** comme **NFS** ou **NIS**
S15netfs lance le service **NFS** client, à désactiver si vous ne montez jamais de file systems d'un serveur **NFS**
S16ypserv pour lancer le serveur **NIS**, à désactiver si non utilisé
S20random ce n'est pas un daemon, mais un truc qui permet de générer bazar aléatoire, je vois pas trop son utilité mais vous pouvez le laisser, il n'y aucun risque niveau sécurité
S20rstatd service "r", à désactiver
S20rwhod idem, à désactiver
S20rusersd idem, à désactiver
S20bootparamd, idem **tftp**, sert pour les terminaux X ou autres clients "diskless", à désactiver
S30syslog permet de loguer l'activité des daemons lancés par (x)**inetd**, à conserver
S34yppasswd si vous êtes un serveur **NIS** pour pouvoir changer de mot de passe, à désactiver si non utilisé (très vulnérable)
S35dhcp daemon **DHCP** sert pour obtenir une adresse IP, sert pour les câblés entre autres, à désactiver si non nécessaire
S40atd utile pour le service **at** (similaire à **cron**), vous pouvez désactiver si vous vous en servez pas
S40crond pour lancer le daemon **cron** qui permet de programmer des tâches à lancer, à virer si vous ne l'utilisez pas
S50inet pour lancer le réseau,
S50snmpd pour lancer le daemon **SNMP**, permet de donner à des utilisateurs distants des informations détaillées sur votre système, à désactiver

S55named pour lancer le serveur **DNS**, à désactiver si vous ça ne sert à rien
S55routed pour router selon le protocole **RIP**, à désactiver
S57diald permet de lancer le daemon **diald** pour lancer une connexion internet automatiquement (du serveur ou d'un client du réseau local), à désactiver si vous vous en servez pas
S60lpd système d'impression
S60nfs pour lancer le serveur **NFS**, à désactiver si vous n'exportez pas vos systèmes de fichiers
S60mars-nwe pour lancer un serveur **Netware** de **Novell**, à désactiver si non nécessaire
S72autofs pour lancer l'automontage (peut s'appeller aussi **S72amd**), à virer si vous ne l'utilisez pas
S75keytable pour activer le clavier qui va bien (claver français azerty)
S75gated pour lancer d'autres protocoles de routage comme **OSPF**, à désactiver
S80sendmail pour lancer **sendmail**, à désactiver si vous ne l'utilisez pas
S85sound pour activer le son
S85gpm permet d'avoir la souris sur des applis textes comme **Midnight Commander**, à désactiver si inutile
S85http pour lancer le serveur Web (**Apache**), à désactiver si non utilisé
S86ypbind si vous êtes un client **NIS**, à désactiver sinon
S90squid pour lancer le proxy **squid**, pour partager la connexion internet, à désactiver si poste isolé
S90xfs pour activer le serveur de fonts X, nécessaire
S91smb pour lancer le serveur **samba**, si vous partager des file système ou des imprimantes vers des postes **Windows**, à désactiver sinon
S94ntpd pour lancer le serveur **NTP** (network time protocol) à désactiver (ancienne version **S94xntp**)
S95innd pour lancer le serveur de news **innd**, si non utilisé à désactiver
S99linuxconf permet à quelqu'un sur internet de faire de la maintenance sur votre système à travers une interface web, à désactiver
S99local c'est un lien vers **/etc/rc.d/rc.local**, où vous pouvez rajouter vos petits trucs

C'est une liste non exhaustive, d'un système à l'autre, le numéro de lancement peut changer. Pour désactiver le service **SNMP** il suffit de taper :

chkconfig --level 0123456 snmpd off

Un autre moyen est de supprimer les liens vers **/etc/rc.d/init.d/snmpd** se trouvant sous **/etc/rc.d/rcX.d** (remplacer X par 0, 1, 2, 3, 4, 5 et 6)

3.3.4 Sécuriser /etc/passwd

Tout d'abord on va mettre en place les **shadow password**, comme ça tous les mots de passe seront contenus dans un fichier distinct de **/etc/passwd**, ce fichier **/etc/shadow** est en lecture seule pour root (droit 400 proprio root). La première chose qu'un hacker cherche à faire est de lire /**etc/passwd**, le fait de mettre les password ailleurs apportent une protection supplémentaire.
Pour activer les **shadows password**, c'est très simple en tant que root, vous devez taper:

pwconv

Dans le champ password de **/etc/passwd** (le deuxième) vous devriez trouver un **x**, un fichier **/etc/shadow** a été créé contenant les mots de passe crypté.

Maintenant on va supprimer autant que possible tous les comptes systèmes inutiles, si vous en avez pas besoin, supprimez les, car ce sont autant de portes d'entrée pour les hackers. Les comptes systèmes suivants sont nécessaires:

root, **bin**, **daemon**, **adm**, **lp** (si vous avez un système d'impression), **mail** (si serveur mail), **news** (si serveur de news), **uucp** (si vous utilisez UUCP), **nobody**

Ceux-ci sont facultatifs:

games, **gopher**, **halt**, **sync**, **shutdown**, **operator**, **ftp** (si serveur FTP anonyme), **lists**, **xfs**.

3.3.5 Sécuriser FTP

Si vous voulez activer **FTP**, vous pouvez le faire avec dé commentant la ligne le concernant dans **/etc/inetd.conf**, on va faire cependant en sorte que le système logue tout ce qui concerne **FTP**. La ligne dans **/etc/inetd.conf** est:

ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -L -i -o

avec:

-l chaque session **FTP** est loguée
-L toutes les commandes utilisateurs sont loguées
-i chaque fichier récupéré est logué
-o chaque fichier envoyé est logué

En cas d'utilisation de **xinetd** voici un exemple de fichier **ftp** à placer sous **/etc/xinetd.d** définissant une plage horaire et une limitation à 4 personnes connectées simultanément

```
service ftp
{
   socket_type     = stream
   wait            = no
   protocol        = tcp
   user            = root
   server          = /usr/sbin/in.ftpd
   instances       = 4
```

```
    access_times = 7:00-12:00 14:00-17:00
    nice=15
}
```

Maintenant vous allez créer un fichier **/etc/ftpusers** qui va contenir la liste des utilisateurs qui n'ont pas le droit d'ouvrir une session **FTP** sur votre système, dans ce fichier vous devrez y mettre tous les utilisateurs systèmes, voici un exemple de **/etc/ftpusers**:

```
root
bin
daemon
adm
lp
mail
news
nobody
```

3.3.6 Sécuriser Telnet

On doit empêcher que root puisse accéder au système via **telnet**, ça force les utilisateurs à se loguer sur leur compte habituel puis de faire un **su** en local s'ils veulent devenir root. Pour cela le fichier **/etc/securetty** doit contenir uniquement des terminaux locaux (du style **ttyX**) et aucunement des pseudos terminaux (du style **ttypX**) qui permettent à un utilisateur distant de se loguer à distance en tant que root. Voici un exemple de **/etc/securetty** :

```
tty1
tty2
tty3
tty4
tty5
tty6
tty7
tty8
```

Maintenant quand vous vous connectez avec **telnet**, vous avez toujours un petit commentaire qui s'affiche, du style "bienvenu sur la machine untel", vous pouvez modifier cela en mettant le commentaire souhaité dans le fichier **/etc/issue**.

Attention sous une **Mandrake**, le fichier **/etc/issue** est regénéré à chaque boot, pour éviter cela, modifiez la ligne qui va bien dans **/etc/rc.d/rc.local**.

3.3.7 Les TCP Wrappers

Les **TCP Wrappers** rajoutent une protection supplémentaire aux services lancés par **inetd**, c'est en fait un contrôle d'accès, par exemple pour se connecter avec **telnet**, il faut d'abord passer le contrôle des **TCP Wrappers**, une fois passer ce contrôle on peut alors se connecter

à **telnet**, si on ne passe pas le contrôle des **TCP Wrappers**, la session **telnet** n'est même pas ouverte. Les **TCP Wrappers** permettent évidemment de loguer chaque connexion (réussie ou pas). Les fichiers de configuration des **TCP Wrappers** pour fixer les accès sont **/etc/hosts.allow** et **/etc/hosts.deny**. Ces fichiers déterminent qui peut ou ne peut pas accèder aux systèmes (ou du moins aux services lancés par **inetd**).
La règle est d'interdire tout à tout le monde, puis d'autoriser uniquement certains postes très limités à utiliser vos services. La syntaxe est la suivante:

service: source(adresse IP, réseau, ou nom): optionnel: ALLOW ou DENY

Par exemple **/etc/hosts.deny**:

ALL: ALL DENY

Pour **/etc/hosts.allow**

in.telnetd:192.168.13.10::ALLOW
in.ftpd:192.168.13.0/255.255.255.0::ALLOW

NOTES: - Pour le nom du service **in.telnetd** est le dernier champ de la ligne **telnet** du fichier **/etc/inetd.conf**

- Préférez les adresses IP, plutôt que le nom
- **192.168.13.10** est l'adresse IP d'un hôte autorisé
- **192.168.13.0/255.255.255.0** est un sous réseau complet

Les **TCP_WRAPPERS** concernent encore **xinetd** mais devraient disparaître dans le futur car **xinetd** offre des contrôles d'accès supérieurs à ce que peut offrir les T**CP_WRAPPERS**.

3.3.8 Root et utilisateurs privilégiés

Maintenant vous pouvez faire en sorte que seuls certains utilisateurs aient le droit d'utiliser certaines commande "puissantes" comme **su**. En limitant le nombre de personnes pouvant utiliser ces commandes vous améliorez la sécurité de votre site. Pour cela vous allez créer un groupe d'utilisateur privilégié, généralement il est appelé **wheel**, mais libre à vous de l'appeler comme vous voulez, choisissez quand même un nom discret, passe partout, pour ne pas éveiller les soupçons.
Maintenant pour que **su** soit lancé uniquement par les membres du groupe **wheel**, vous devrez taper:

chgrp wheel /bin/su
chmod 4750 /bin/su

Faites de même pour les autres commandes.

NOTES: - N'oubliez pas qu'un utilisateur peut très bien appartenir à deux groupes
- N'oubliez pas sudo pour donner des droits privilégiés à certains utilisateurs.

Pour ce qui concerne root, vous devez prendre quelques précautions:

- vous ne devez en aucun rajouter le **.** (répertoire courant) dans le PATH de root, car si par malheur quelqu'un crée un script avec droits exécutables dans **/tmp** qui s'appelle **rm** contenant:

```
#!/bin/bash
rm -Rf /
```

Si root a le malheur de taper **rm** alors qu'il se trouve dans **/tmp**, c'est le script qui peut être appelé (suivant l'ordre des chemins dans le PATH) et pouf ! plus de système.

3.3.9 Sécuriser les fichiers et systèmes de fichiers

3.3.9.1 SUID et GUID

Evitez d'avoir recours aux SUID et SGID, ils comportent certains risques au niveau de la sécurité, ils permettent à n'importe qui de lancer un programme avec les droits du propriétaire du programme. Les fichiers SUID sont une des principales cibles des hackers, à éviter donc, préférez amplement sudo.
Pour trouver les fichiers avec SUID ou SGID, tapez en tant que root:

```
find / -type f  \( -perm 04000 -o -perm 02000 \)
```

3.3.9.2 Fichiers .rhosts et hosts.equiv

On a vu plus haut qu'il fallait éviter d'avoir des fichiers **.rhosts** ou **hosts.equiv**, on va donc bloquer ces deux fichiers pour que personne ne puisse les recréer. Pour bloquer les fichiers, il suffit de taper les commandes:

```
touch /.rhosts /etc/hosts.equiv
chmod 0 /.rhosts /etc/hosts.equiv
```

Pour trouver les **.rhosts** dans les homedirectories des utilisateurs, taper:

```
find /home -name .rhosts -print
```

3.3.9.3 Umask

On peut définir les droits de création par défaut d'un fichier ou de répertoire avec la commande **umask**, on va faire en sorte d'éviter de créer des répertoires avec les droits 777.
Pour définir l'**umask** pour tous les utilisateurs du système, vous devez éditer le fichier **/etc/profile** et modifiez la ligne concernant **umask**, vous pouvez fixer 022, 027 ou même 077 qui est le masque le plus restrictif (voir mon cours unix pour voir comment marche les **umask**).

4 Auditer la sécurité de son réseau

4.1 Présentation

Le but de cette page est de vous présenter trois outils qui vous permettront de tester la sécurisation des machines de votre réseau, ils vous révéleront vos trous de sécurité et vous avertiront des problèmes potentiels, à partir de là libre à vous de "boucher" les trous en question et d'upgrader certains programmes présentant quelques déficiences de sécurité. Ces outils procèdent en scannant votre machine, en testant tous les ports ouverts notamment et en testant un grand nombre de trous de sécurité connus.
Les outils présentés sont **SARA** qui est dérivé du célèbre **SATAN** qui n'a pas été maintenu depuis un certain temps, **nmap** qui est un puissant "scanneur" et **nessus** basé entre autres sur **nmap** mais avec en plus une interface particulièrement conviviale. Des trois outils c'est le dernier que je juge le plus puissant et de surcroît facile d'utilisation.

4.2 AVERTISSEMENT

Je vous déconseille évidemment fortement de tenter de scanner une machine ne vous appartenant pas se trouvant sur le net, le scan sera considéré comme une attaque, vous vous exposez à des problèmes en proportion avec la machine visitée, mais bon je vous aurais prévenu.

4.3 Sara

4.3.1 Présentation

En 1995 est apparu un outil d'administration pour tester la sécurité d'un réseau, avec le nom évocateur de **SATAN** (Security Administrator Tool for Analyzing Networks). Il avait la particularité d'être Open Source, il est très vite devenu un standard et fut à la base d'une pléthore d'outils équivalents. Le problème est que maintenant **SATAN** commence fortement à dater, ça fait un paquet de temps qu'il n'a pas été remis à niveau, en d'autres termes, il est actuellement complètement dépassé. En conséquence une boîte (Advanced Research Corporation) commença le développement d'un outil similaire au goût du jour, ainsi naquit **SARA** (Security Auditor Research Assistant), comme **SATAN** , **SARA** est Open Source.
A noter qu'il existe une version commerciale de **SARA**, appelée **SARA-PRO**. A noter encore qu'un des auteurs de **SARA** est aussi à l'origine de **SAINT**, un autre SATAN-like.

4.3.2 Installation

On peut récupérer l'archive de **SARA** à savoir **sara-7.8.4.tgz** à l'URL www-arc.com/sara , qu'on décompressera en tapant :

tar xvfz sara-7.8.4.tgz

Après décompression, vous allez récupérer un répertoire **sara-7.8.4**. Installer d'abord le package **perl-devel** puis dans le répertoire ainsi obtenu, il suffit de taper successivement :

./configure

Puis en tant que root on tapera d'abord

ln -s /usr/lib/perl5/ /usr/perl
urpmi csh

puis

make

Lors de la compil, il peut se plaindre de l'absence de certaines commandes, mais ce n'est pas bien grave.

Looking for all the commands now...

AEEEIIII...!!! can't find mail

AEEEIIII...!!! can't find tftp

AEEEIIII...!!! can't find ypcat

AEEEIIII...!!! can't find finger

AEEEIIII...!!! can't find rusers

AEEEIIII...!!! can't find ypwhich

AEEEIIII...!!! can't find rlogin

AEEEIIII...!!! can't find rsh

Si tout à la fin vous obtenez

Now building CVE database

Now building FIFOs for SSS

puis on tape maintenant (à mon avis c'est inutile mais dans le doute...)

make install

4.3.3 Utilisation

En tant que root il suffit de taper

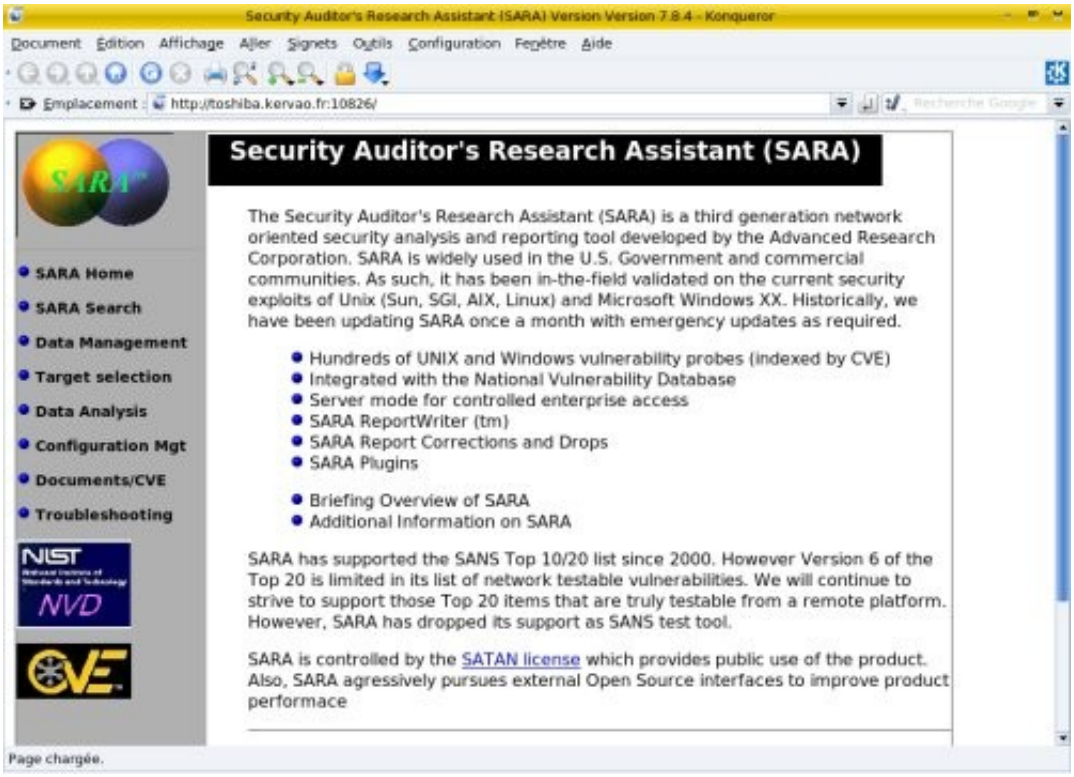
/usr/local/sara/sara

NOTE: Si vous disposez de **nmap** (voir le chapitre suivant), vous pouvez rajouter l'option **-n** pour lui indiquer, le scan sera d'autant meilleur.

Et là surprise, c'est le browser par défaut qui se lance. Avec **firefox** ça ne marche pas, alors du coup il faut lancer **konqueror** pour configurer le navigateur il faut alors éditer le fichier **/usr/local/sara/config/paths.pl** et modifier la ligne suivante ainsi

$MOSAIC="/usr/bin/konqueror";

Voilà maintenant dans le browser vous avez la page d'accueil de **Sara**



A noter que la partie **Documents/CVE** est très bien fournie ainsi que la section **Troubleshouting** se présentant sous forme de FAQ.

La première étape est de sélectionner la cible à analyser. on clique donc sur **Target Selection**, vous devez alors saisir:

Primary target selection: choix de la cible

Vous pouvez choisir une machine particulière (target host), un sous réseau complet (network), ou une plage d'adresse (range), la syntaxe est la suivante :

- un hôte simple hote.local.com
- plusieurs hôtes: hote1.local.com hote2.local.com
- un sous réseau: 192.168.13.0/24
- une plage d'adresse: 192.168.0.55-192.168.0.98

Scanning level selection choix du niveau de scan

Les choix sont :

- Light : léger (pour que la machine cible ne se rende pas compte)
- Normal : peut être détecté et inscrit dans les fichiers de log de la cible
- Heavy : lourd, des messages d'erreurs peuvent apparaître sur les consoles système de la cible
- Extreme : extrême, ça peut planter

- Custom : personnalisé (seulement scan SMB (samba), Web, Mail, Telnet, FTP)
- Custom : personnalisé (seulement scan TCP)
- Custom : personnalisé (seulement un scan de test)

Pour l'exercice, je choisis une machine sous Mandriva avec un scanning **heavy** sur mon réseau local.

Voilà maintenant on a un bouton **start the scan**, et c'est parti, là ça peut prendre un certain temps, voilà le résultat final sur la machine **obelix.armoric.bz** dans une page du navigateur.

Data collection in progress...

Adding a primary target
Add-primary: obelix.breizland.bz
Add-target: obelix.breizland.bz prox 0
policy: obelix.breizland.bz prox 0 level 2
Check-pulse: obelix.breizland.bz
==> running bin/timeout 180 bin/fping obelix.breizland.bz
process_targets: probe obelix.breizland.bz...
Prox: 0
AL : 2
Add-todo: obelix.breizland.bz|dns.sara|
Add-todo: obelix.breizland.bz|rpc.sara|
Add-todo: obelix.breizland.bz|finger.sara|
Add-todo: obelix.breizland.bz|ddosscan.sara|
Add-todo: obelix.breizland.bz|hosttype.sara|
Add-todo: obelix.breizland.bz|tcpscan.sara
1-1525,1527-5404,5406-5899,5901-7099,7101-8887,8889-9999,12345,16600,20034,27374,27665,31337,31785,65000|
Add-todo: obelix.breizland.bz|udpscan.sara
1-1760,1763-2050,31335,31337,27444,32767-33500|
==> running bin/timeout 20 bin/ddosscan.sara obelix.breizland.bz
==> running bin/timeout 180 bin/udpscan.sara
1-1760,1763-2050,31335,31337,27444,32767-33500 obelix.breizland.bz
Add-fact: obelix.breizland.bz|#|a|x||||offers #
Add-fact: obelix.breizland.bz|echo|a|x||||offers echo
Add-fact: obelix.breizland.bz|bootps|a|x||||offers bootps
Add-fact: obelix.breizland.bz|sunrpc|a|x||||offers sunrpc
Add-fact: obelix.breizland.bz|netbios-ns|a|x||||offers netbios-ns
Add-fact: obelix.breizland.bz|netbios-dgm|a|x||||offers netbios-dgm
==> running bin/timeout 20 bin/finger.sara obelix.breizland.bz
Add-fact: obelix.breizland.bz|finger.sara|u|||||program timed out
==> running bin/timeout 20 bin/rpc.sara obelix.breizland.bz
Add-fact: obelix.breizland.bz|nfs|a|x||||runs NFS
Add-fact: obelix.breizland.bz|mountd|a|x||||runs NFS
Add-fact: obelix.breizland.bz|statd|a|x||||runs statd
==> running bin/timeout 20 bin/dns.sara obelix.breizland.bz
==> running bin/timeout 180 bin/hosttype.sara obelix.breizland.bz
==> running bin/timeout 180 bin/tcpscan.sara
1-1525,1527-5404,5406-5899,5901-7099,7101-8887,8889-9999,12345,16600,20034,27374,27665,31337,31785,65000
obelix.breizland.bz
Add-fact: obelix.breizland.bz|ipp|a||||HTTP/1.0 400 Bad Request\r\nDate: Sat, 10 Mar 2001 09:01:04 GMT\r\nServer: CUPS/1.1\r\nContent-Type: text/html\r\nContent-

Length: 101\r\n\r\n
Bad Request
Bad Request\n|offers ipp
Add-fact: obelix.breizland.bz|masqdialer|a||||READY\n|offers masqdialer
Add-fact: obelix.breizland.bz|tproxy|a|||||offers tproxy
Add-fact: obelix.breizland.bz|nntp|a||||200 Leafnode NNTP Daemon, version 1.9.18 running at obelix.breizland.bz\r\n|offers nntp
Add-fact: obelix.breizland.bz|blackjack|a|||||offers blackjack
Add-fact: obelix.breizland.bz|smtp|a||||220 rennes-1-a7-21-114.dial.proxad.net ESMTP Sendmail 8.11.0/8.11.0/Olivier Hoarau-992911; Sat, 10 Mar 2001 10:01:11 +0100\r\n221 2.0.0 rennes-1-a7-21-114.dial.proxad.net closing connection\r\n|offers smtp
Add-fact: obelix.breizland.bz|submission|a||||220 rennes-1-a7-21-114.dial.proxad.net ESMTP Sendmail 8.11.0/8.11.0/Olivier Hoarau-992911; Sat, 10 Mar 2001 10:01:11 +0100\r\n221 2.0.0 rennes-1-a7-21-114.dial.proxad.net closing connection\r\n|offers submission
Add-fact: obelix.breizland.bz|telnet|a||||Welcome to obelix.breizland.bz\r\nLinux Mandrake release 7.2 (Odyssey) for i586\r\nKernel 2.2.17-21mdk on an i686\r\nQUIT\r\n\r\n|offers telnet
Add-fact: obelix.breizland.bz|ftp|a|||||offers ftp
Add-fact: obelix.breizland.bz|ssh|a|||||offers ssh
Add-fact: obelix.breizland.bz|http|a|||||offers http
Add-fact: obelix.breizland.bz|meregister|a|||||offers meregister
Add-fact: obelix.breizland.bz|pop3|a|||||offers pop3
Add-fact: obelix.breizland.bz|sunrpc|a|||||offers sunrpc
Add-fact: obelix.breizland.bz|1024:TCP|a|||||offers 1024:TCP
Add-fact: obelix.breizland.bz|1026:TCP|a|||||offers 1026:TCP
Add-fact: obelix.breizland.bz|netbios-ssn|a|||||offers netbios-ssn
Add-fact: obelix.breizland.bz|http-alt|a||||HTTP/1.0 500 WWWOFFLE Server Error\r\nServer: WWWOFFLE/2.6a\r\nDate: Sat, 10 Mar 2001 09:01:16 GMT\r\nContent-type: text/html\r\n\r\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\nWWWOFFLE - World Wide Web Offline Explorer\n
\n\n

\n\n\n\n
WWWOFFLE - Erreur du Serveur \n\n
\nLe serveur WWWOFFLE a rencontré une erreur fatale :\n
\nCannot parse the HTTP request\n
\nLe programme ne peut pas continuer à traiter cette requête.\n
\n\n
\nCeci est une erreur interne du serveur WWWOFFLE. Si elle persiste\nbien que le serveur soit correctement configuré et qu'il n'y ait pas\nd'autre problème sur l'ordinateur, alors signalez la (en\nanglais) à l'auteur de WWWOFFLE.\n\n<\n\n\n\n

\n\n
\nWWWOFFLE - [Page d'Accueil?FAQ] - WWWOFFLE\n
\n\n\n\n\n\n\n|offers http-alt
Add-fact: obelix.breizland.bz|X-0|a|||||offers X-0

Waiting for all processes to complete
Add-todo: obelix.breizland.bz|smb.sara|
Add-todo: obelix.breizland.bz|depends.sara|statd
Add-todo: obelix.breizland.bz|http.sara|http
Add-todo: obelix.breizland.bz|sample.sara.ext|http
Add-todo: obelix.breizland.bz|http.sara|http-alt
Add-todo: obelix.breizland.bz|sample.sara.ext|http-alt
Add-fact: obelix.breizland.bz|http-alt|a|||||offers http:http-alt
Add-todo: obelix.breizland.bz|depends.sara|nfs
Add-fact: obelix.breizland.bz|nfs|a|g||||runs NFS
Add-todo: obelix.breizland.bz|sendmail.sara|
Add-todo: obelix.breizland.bz|relay.sara|
Add-todo: obelix.breizland.bz|login.sara|-u root
Add-todo: obelix.breizland.bz|login.sara|-u guest
Add-todo: obelix.breizland.bz|depends.sara|telnet
Add-todo: obelix.breizland.bz|showmount.sara|
Add-todo: obelix.breizland.bz|nfs-chk.sara|-t 10
Add-fact: obelix.breizland.bz|mountd|a|g||||runs NFS
Add-todo: obelix.breizland.bz|ssh.sara|
Add-todo: obelix.breizland.bz|ftp.sara|
Add-todo: obelix.breizland.bz|xhost.sara|-d obelix.breizland.bz:0
Add-fact: obelix.breizland.bz|ipp|a|||||offers http:ipp
Add-todo: obelix.breizland.bz|http.sara|ipp
Add-todo: obelix.breizland.bz|sample.sara.ext|ipp
Add-fact: obelix.breizland.bz|http-alt|a|||||offers http
==> running bin/timeout 45 bin/login.sara -u root obelix.breizland.bz
Add-fact: obelix.breizland.bz|telnet|a|g||||
==> running bin/timeout 45 bin/login.sara -u guest obelix.breizland.bz
==> running bin/timeout 180 bin/http.sara ipp obelix.breizland.bz
Add-fact: obelix.breizland.bz|ipp|a|g||||offers http
==> running bin/timeout 700 bin/smb.sara obelix.breizland.bz
Add-fact:
obelix.breizland.bz|netbios-ssn|a|zwoi|ANY@obelix.breizland.bz|
ANY@obelix.breizland.bz|netbios
over the internet|Is your Netbios secure
==> running bin/timeout 20 bin/xhost.sara -d obelix.breizland.bz:0
obelix.breizland.bz
==> running bin/timeout 20 bin/ftp.sara obelix.breizland.bz
Add-fact: obelix.breizland.bz|ftp||||||offers ftp
==> running bin/timeout 20 bin/showmount.sara obelix.breizland.bz
Add-fact: obelix.breizland.bz|showmount|a|||||Not running showmount or other
error
==> running bin/timeout 20 bin/depends.sara telnet obelix.breizland.bz
==> running bin/timeout 180 bin/http.sara http-alt obelix.breizland.bz
Add-fact: obelix.breizland.bz|http-alt|a|g||||offers http
==> running bin/timeout 20 bin/sample.sara.ext http-alt obelix.breizland.bz
==> running bin/timeout 20 bin/depends.sara statd obelix.breizland.bz
Add-fact: obelix.breizland.bz|statd|a|rcio|ANY@ANY|ANY@ANY|rpc statd access|
rpc.statd on Linux is vulnerable if not patched
==> running bin/timeout 20 bin/sendmail.sara obelix.breizland.bz
Add-fact: obelix.breizland.bz|smtp|a|zcio|ANY@ANY|ANY@ANY|sendmail version|
sendmail VRFY command may provide hacker information

Add-fact: obelix.breizland.bz|smtp|a|zcio|ANY@ANY|ANY@ANY|sendmail version| sendmail EXPN command may provide hacker information
==> running bin/timeout 120 bin/nfs-chk.sara -t 10 obelix.breizland.bz
==> running bin/timeout 20 bin/sample.sara.ext http obelix.breizland.bz
==> running bin/timeout 700 bin/relay.sara obelix.breizland.bz
Add-fact:
obelix.breizland.bz|smtp|a|ycio|ANY@obelix.breizland.bz|ANY@obelix.breizland.bz| SMTP may be a mail relay|Probable smtp relay (spam)
==> running bin/timeout 180 bin/http.sara http obelix.breizland.bz
Add-fact: obelix.breizland.bz|http|a|g||||offers http
==> running bin/timeout 20 bin/depends.sara nfs obelix.breizland.bz
Add-fact: obelix.breizland.bz|nfsd|a|zcio|ANY@ANY|ANY@ANY|mountd vulnerabilities|nfsd version may be vulnerable to buffer overflow
==> running bin/timeout 20 bin/ssh.sara obelix.breizland.bz
Add-fact: obelix.breizland.bz|ssh|a|g||||offers ssh
==> running bin/timeout 20 bin/sample.sara.ext ipp obelix.breizland.bz
Waiting for all processes to complete

Data collection completed (1 host(s) visited).

Back to the SARA start page | Continue with report and analysis | View primary target results

ATTENTION Le scan peut provoquer le blocage de certains services de la machine cible.

Si vous choisissez **View primary target results**, vous obtiendrez un truc du style:

General host information:

Host type: unknown type
NFS server
NIS server
NNTP (Usenet news) server
SSH server
Telnet server
WWW server
Subnet 192.168.13
Scanning level: heavy
Last scan: Fri Mar 10 19:01:23 2001

Vulnerability information:

Is your Netbios secure
sendmail EXPN command may provide hacker information
sendmail VRFY command may provide hacker information
rpc.statd on Linux is vulnerable if not patched
Probable smtp relay (spam)
SMTP may be a mail relay
nfsd version may be vulnerable to buffer overflow

Vous constatez qu'il y a des liens un peu partout qui vous donne plus d'info.

Si par contre vous choisissez "**Continue with report and analysis** ", vous aurez alors le menu suivant:

Table of contents

Vulnerabilities

By Approximate Danger Level
By Type of Vulnerability
By Vulnerability Count

Host Information

By Class of Service
By System Type
By Internet Domain
By Subnet
By Host Name

Reporting

SARA ReportWriter

Chaque catégorie étant un lien vers un autre page avec davantages de détails et d'informations sur chaque vulnérabilité. Je ne vous les présenterai pas de manière exhaustive, car ça part vraiment dans tous les sens.

4.4 Nmap

4.4.1 Présentation

Nmap est un puissant outil qui permet de scanner les ports, il a pour but de tester la sécurisation des postes de vos réseau, vous devez vous en servir uniquement pour votre réseau, si vous tentez de scanner quelqu'un d'autres sur le réseau internet, sachez que ce sera considéré comme une attaque, c'est donc à vos risques et périls...
Pour info **nmap** se trouve maintenant intégré dans la mandrake, je vous présenterai l'installation avec le tarball avec la dernière version stable 3.76, l'installation avec les rpm de la Mandriva se réduit à sa plus simple expression. L'utilisation de **nmap** que vous l'ayez installé avec le rpm ou avec le tarball reste la même

4.4.2 Installation avec le tarball

Vous pouvez trouver **nmap** sur le site www.insecure.org/nmap La dernière version stable est la 4.76 qu'on peut récupérer sous forme de tarball.
Attention de ne pas installer **nmap** si vous disposez déjà de la version Mandrake, pour le savoir:

rpm -qa | grep -i nmap

Si vous obtenez

nmap-3.XX-Xmdk
nmap-frontend-3.XX-Xmdk

On va supprimer ces deux packages.

rpm -e nmap-3.XX-Xmdk
rpm -e nmap-frontend-3.XX-Xmdk

Pour décompresser le tarball rien de plus simple.

tar xvfj nmap-4.76.tar.bz2

Cela va créer dans le répertoire courant un répertoire **nmap-4.76**. Au préalable vous veillerez à installer si ce n'est déjà fait sur votre Mandrake les packages suivants: **byacc**, **glib-devel, gtk+-devel, flex** et **libpcap**

Puis la commande classique:

./configure

Tapez maintenant

make

Puis en tant que **root**

make install

Cela va installer les exécutables **nmap** et **nmapfe** sous **/usr/local/bin**, et d'autres fichiers sous **/usr/local/share/nmap**, si ça ne vous convient pas, vous pouvez changer les chemins en tapant **configure** avec les arguments qui vont bien (**./configure --help** pour la syntaxe).

4.4.3 Syntaxe

La syntaxe est classique:

nmap [options éventuelles] cible

Les options principales sont:

-sT scanning des ports TCP ouverts, on ouvre une connexion sur tous les ports ouverts, toutes les connexion sont visibles sur la machine cible (dans les fichiers de log notamment),
-sS scanning des ports TCP, on envoit un message SYN pour dire qu'on va ouvrir une connexion TCP puis on attend la réponse, après réponse on sait que le port est ouvert, l'avantage de cette option est que l'action n'est pas logué par la cible,
-sF,**-sX**,**-sN** Stealth FIN, Xmas, ou Null scan (marche que sous UNIX) scan des ports plus discrets (voir **man nmap** pour plus de détails)
-sP équivalent à **ping**, pour voir si la cible est "alive", ne fait pas de scan,
-sU scanning des ports UDP ouverts (plutôt lent),
-b <ftp_relay_host> ftp "bounce attack" port scan, en gros ça exploite un trou de sécurité sur certain proxy, je vous en dirais pas plus (voir le **man**).

D'autres options intéressantes (liste non exhaustive):

-O permet de connaître sur quel OS tourne la cible (fingerprinting en anglais),
-p <range> syntaxe pour avoir un ensemble de ports à scanner, exemple :
-p 23 on va regarder que le port 23
-p 20-30,63000- va scanner les ports de 20 à 30 et supérieur à 63000 (jusqu'à 65535 en fait)
par défaut il scrute uniquement de 1 à 1024 plus les ports se trouvant dans **/etc/services** ,

-F scan rapide, scanne uniquement les ports contenus dans **/etc/services**,
-I permet d'avoir plus d'info sur les ports TCP ouverts, notamment le proprio,

-o <logfile> log le résultat dans le fichier **<logfile>**,
-v Verbose, mode verbeux (recommandé),
-h help, pour avoir la liste complète des options,
-V pour avoir uniquement le numéro de version, pas la peine de rentrer de cible,
-e <devicename>, avec ça on peut spécifier une interface particulière pour envoyer les paquets (**eth0**, **ppp0**,etc.).

Maintenant la cible peut être identifié par son nom sur internet ou son adresse IP. On peut désigner un ensemble d'adresse aussi, ou des sous-masques de réseau particulier. Par exemple domaine.org/24 ou 192.88.209.5/24, 192.88.209.0-255 ou bien encore '128.88.209.*'.

4.4.4 Quelques exemples

L'utilisation la plus simple est celle-ci:

nmap -v tosh

voilà le résultat

Starting Nmap 4.76 (http://nmap.org) at 2008-11-01 20:56 CET
Initiating SYN Stealth Scan at 20:56
Scanning toshiba.kervao.fr (192.168.2.10) [1000 ports]
Discovered open port 80/tcp on 192.168.2.10
Discovered open port 5679/tcp on 192.168.2.10
Discovered open port 111/tcp on 192.168.2.10
Discovered open port 6000/tcp on 192.168.2.10
Discovered open port 990/tcp on 192.168.2.10
Discovered open port 631/tcp on 192.168.2.10
Completed SYN Stealth Scan at 20:56, 0.18s elapsed (1000 total ports)
Host toshiba.kervao.fr (192.168.2.10) appears to be up ... good.
Interesting ports on toshiba.kervao.fr (192.168.2.10):
Not shown: 994 closed ports
PORT STATE SERVICE
80/tcp open http
111/tcp open rpcbind
631/tcp open ipp
990/tcp open ftps
5679/tcp open activesync
6000/tcp open X11

Read data files from: /usr/local/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds
Raw packets sent: 1000 (44.000KB) | Rcvd: 2006 (84.264KB)

La commande suivante va permettre de scanner tous les ports TCP réservés :

nmap -sS -O tosh

voilà le résultat

Starting Nmap 4.76 (http://nmap.org) at 2008-11-01 20:57 CET
Interesting ports on toshiba.kervao.fr (192.168.2.10):

Not shown: 994 closed ports
PORT STATE SERVICE
80/tcp open http
111/tcp open rpcbind
631/tcp open ipp
990/tcp open ftps
5679/tcp open activesync
6000/tcp open X11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.25
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.47 seconds

Maintenant on va tester si **sshd**, **DNS**, **pop3d**, **imapd** (ports respectifs 22, 53, 110 et 143) tournent sur le système et si le port 4564 est utilisé:

nmap -sX -p 22,53,110,143,4564 tosh

Starting Nmap 4.76 (http://nmap.org) at 2008-11-01 20:57 CET
Interesting ports on toshiba.kervao.fr (192.168.2.10):
Not shown: 994 closed ports
PORT STATE SERVICE
80/tcp open http
111/tcp open rpcbind
631/tcp open ipp
990/tcp open ftps
5679/tcp open activesync
6000/tcp open X11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.25
Network Distance: 0 hops

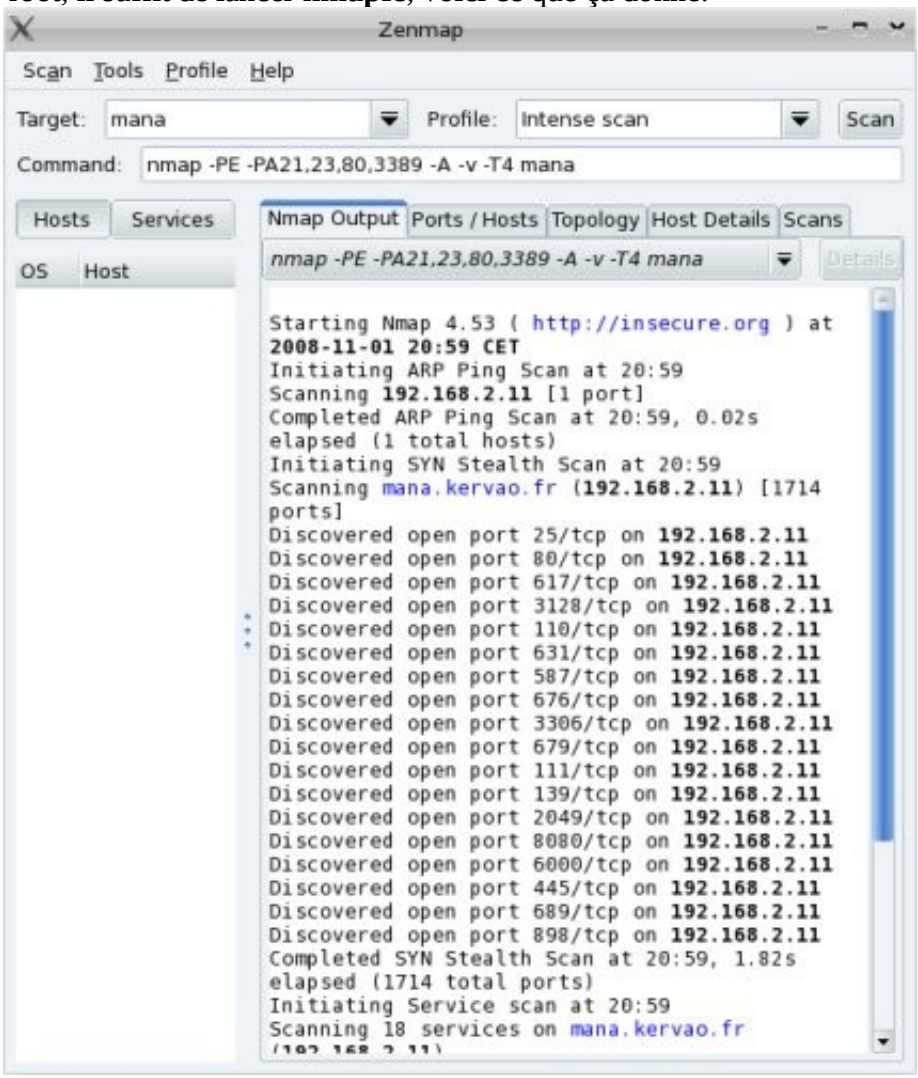
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.47 seconds
[root@toshiba olivier]#
[root@toshiba olivier]# /usr/local/bin/nmap -sX -p 22,53,110,143,4564 toshiba

Starting Nmap 4.76 (http://nmap.org) at 2008-11-01 20:58 CET
Interesting ports on toshiba.kervao.fr (192.168.2.10):
PORT STATE SERVICE
22/tcp closed ssh
53/tcp closed domain
110/tcp closed pop3
143/tcp closed imap
4564/tcp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds

4.4.5 Le front end de nmap

En tant que root, il suffit de lancer **nmapfe**, voici ce que ça donne:



Vous voyez qu'il devient très simple d'utiliser **nmap**.

4.5 Nessus

4.5.1 Présentation

Nessus est un outil qui permet de tester la sécurisation des postes de votre réseau, il se base en autres sur **nmap**, mais en plus de tester vos ports, il va scruter les composants logiciels de votre machine, pour déterminer les trous de sécurité et vous avertir sur certaines faiblesses.
Nessus est constitué d'une partie serveur et cliente, les deux peuvent très bien fonctionner sur la même machine.

J'ai choisi d'installer la version 2 qui est disponible en source, la version 3 n'est disponible qu'en binaire.

4.5.2 Installation

Tout d'abord vous devez vous procurez les packages de **nessus** que vous trouverez à l'URL suivante www.nessus.org. Dézippez l'un après l'autre les packages dans un même répertoire
tar xvfz nessus-core-2.2.11.tar.gz
tar xvfz nessus-libraries-2.2.11.tar.gz
tar xvfz nessus-plugins-2.2.11.tar.gz
tar xvfz libnasl-2.2.11.tar.gz

Cela va donner les répertoires **nessus-core**, **nessus-libraries**, **nessus-plugins** et **libnasl**. Il faut d'abord installer le package **byacc** et **bison**, puis dans le répertoire **nessus-libraries** on tape

./configure

Un message vous indique de taper **uninstall-nessus** si vous voulez effacer une ancienne installation de **nessus**, on tape maintenant :

make

on passe en root puis on tape

make install

Les librairies sont installées sous **/usr/local/lib**, maintenant vous devez rajouter ce chemin dans **/etc/ld.so.conf** et tapez

ldconfig

Pour la suite des opérations les répertoires **/usr/local/bin** et **/usr/local/sbin** doivent être dans votre **PATH** (**env | grep PATH** pour vérifier) si ce n'est pas le cas, dans le fichier **.bashrc** de votre homedirectory rajoutez :

PATH=$PATH:/usr/local/bin:/usr/local/sbin
export PATH

Et relancez un shell pour prendre en compte la modif (ou alors **source ~/.bashrc**). Dans le répertoire **libnasl**, on tape :

./configure

Puis

make

Et enfin en tant que **root**

make install

suivi de

ldconfig

On tape les mêmes commandes (**./configure**, **make** et en tant que **root make install**) dans l'ordre dans le répertoire **nessus-core** puis **nessus-plugins**. Le daemon **nessusd** est installé sous **/usr/local/sbin**, les fichiers nécessaires sous **/usr/local/lib/nessus**

Maintenant on va créer un utilisateur **lambda** pour pouvoir utiliser **nessus**. En tant que root on tape :

nessus-adduser

Voilà les commentaires

Add a new nessusd user

Login : lambda

Méthode d'authentification (mot de passe ou certificat), je choisis la méthode par défaut password (**pass**),

Authentication (pass/cert) [pass] : pass

Mot de passe (en clair),

Login password :
Login password (again) :

A ce niveau on peut créer des critères pour limiter **lambda** dans ses actions de scan, faire un **man nessus-adduser** pour plus de détail

User rules
----
nessusd has a rules system which allows you to restrict the hosts
that lambda 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)

On tape CTRL-D pour stopper la saisie, par défaut les règles peuvent être vides s'il y en a pas.

On récapitule

Login : lamba
Password : ************
DN :
Rules :

Is that ok ? (y/n) [y]
user added.

Pour lancer le serveur on doit préalablement créer les certificats d'authentification, tapez en tant que root

nessus-mkcert

Vous pouvez taper **enter** à chaque ligne, si vous voulez aller vite

----
Creation of the Nessus SSL Certificate

This script will now ask you the relevant information to create the SSL
certificate of Nessus. Note that this information will *NOT* be sent to
anybody (everything stays local), but anyone with the ability to connect to your
Nessus daemon will be able to retrieve this information.

CA certificate life time in days [1460]:1460

Server certificate life time in days [365]:365
Your country (two letter code) [FR]:FR
Your state or province name [none]:none
Your location (e.g. town) [Paris]: Papeete
Your organization [Nessus Users United]: Tahiti Connection

Congratulations. Your server certificate was properly created.

/usr/local/etc/nessus/nessusd.conf updated

The following files were created :

. Certification authority :
Certificate = /usr/local/com/nessus/CA/cacert.pem
Private key = /usr/local/var/nessus/CA/cakey.pem

. Nessus Server :
Certificate = /usr/local/com/nessus/CA/servercert.pem
Private key = /usr/local/var/nessus/CA/serverkey.pem

Press [ENTER] to exit

Ce ne sera plus nécessaire de taper cette commande par la suite. Maintenant on tape

nessusd &

A noter que j'ai eu anciennement l'erreur suivante

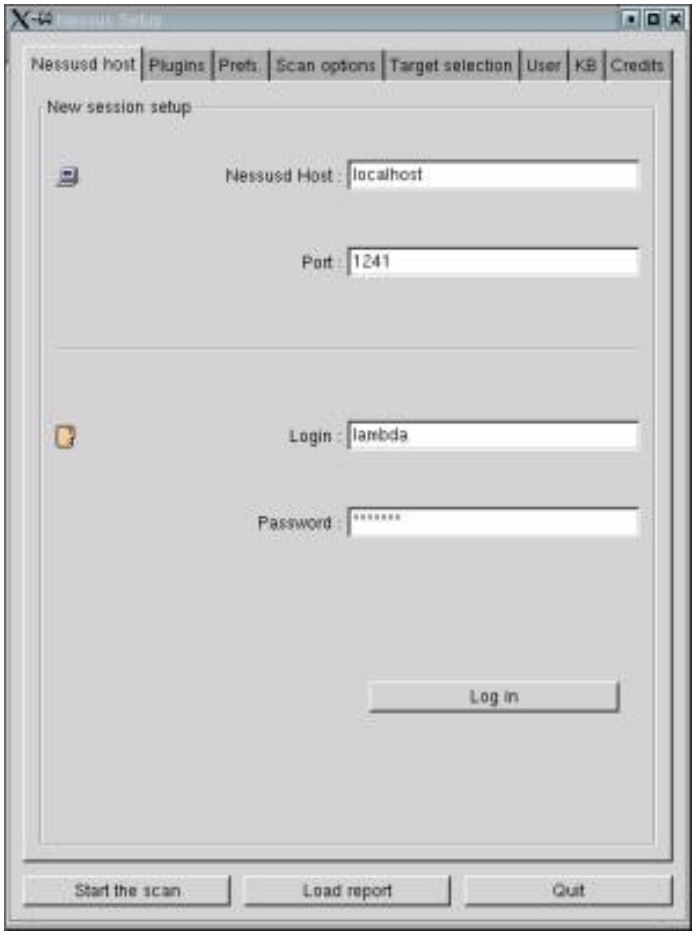
Access rights problem with /usr/local/var/nessus/nessusd.messages (File has path segment with wrong owner)-- aborting[

Cela venait du fait que le répertoire **/usr/local/var** appartenait à **mysql** Il faut qu'il appartienne à **root**

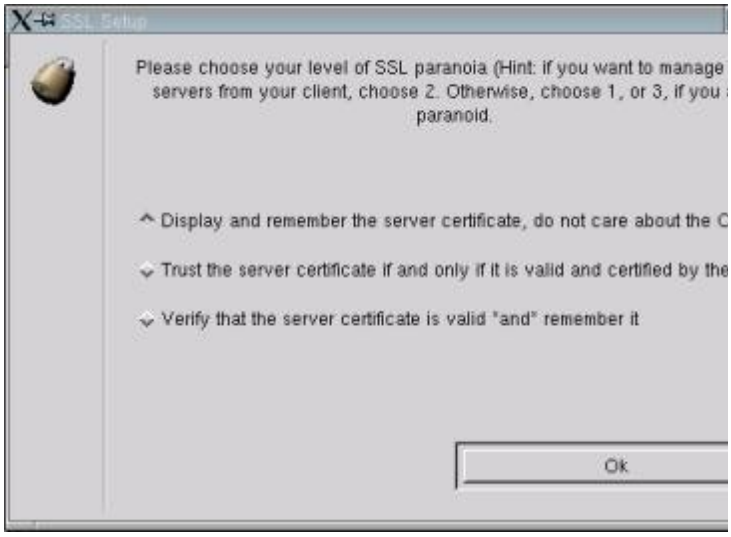
4.5.3 Utilisation

En supposant que le daemon **nessusd** est lancé, il suffit en tant qu'utilisateur quelconque de taper **nessus** pour lancer le client.

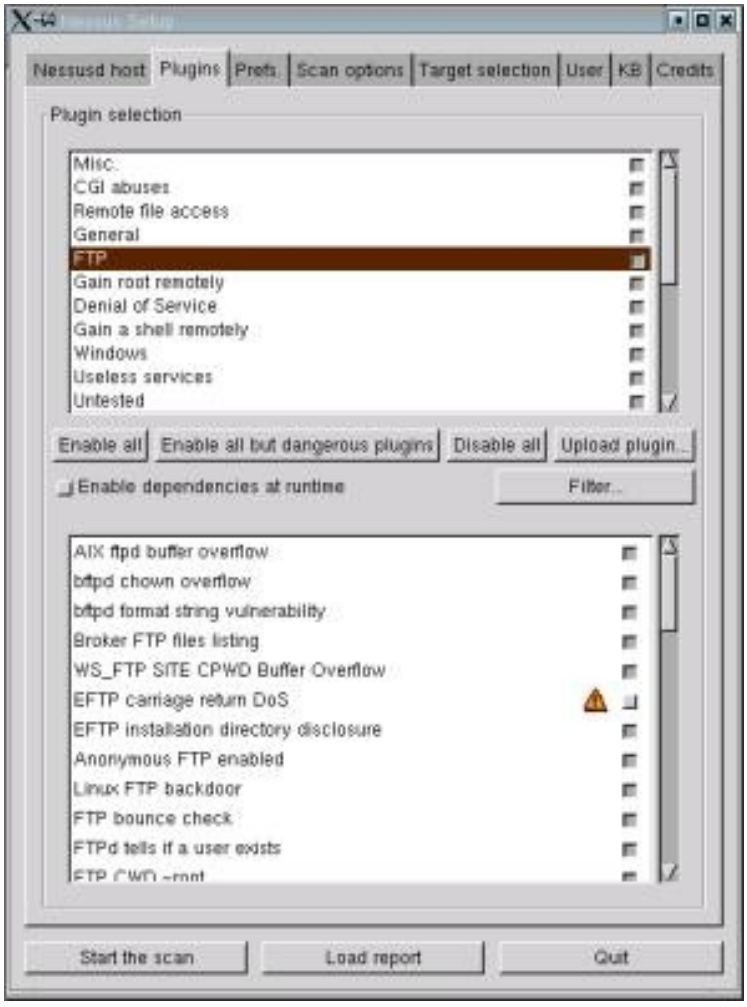
La fenêtre suivante (à droite) apparaît, vous devez saisir le nom de la machine où tourne le serveur, si la machine et le client tourne sur la même machine, vous pouvez laisser **localhost**. Laissez le port par défaut, c'est celui utilisé par **nessusd**.
Vous devez au niveau du champ **Login** saisir l'utilisateur **nessus** que vous avez créé auparavant (**lambda** dans notre cas). Saisissez le mot de passe dans le champ **Password**.
Maintenant vous pouvez vous connecter au serveur en appuyant sur le bouton **Log in**. Vous voyez alors que le champ du bouton **Log in** se change en **Log out**, c'est bon vous êtes connecté.



La fenêtre à droite apparaît, on clique OK. Une autre fenêtre affiche le certificat, on doit l'accepter ou non.
Une autre fenêtre nous indique que les plugins qui peuvent planter un service ou un hôte distant ont été désactivés. Il faut les réactiver pour avoir un scan exhaustif

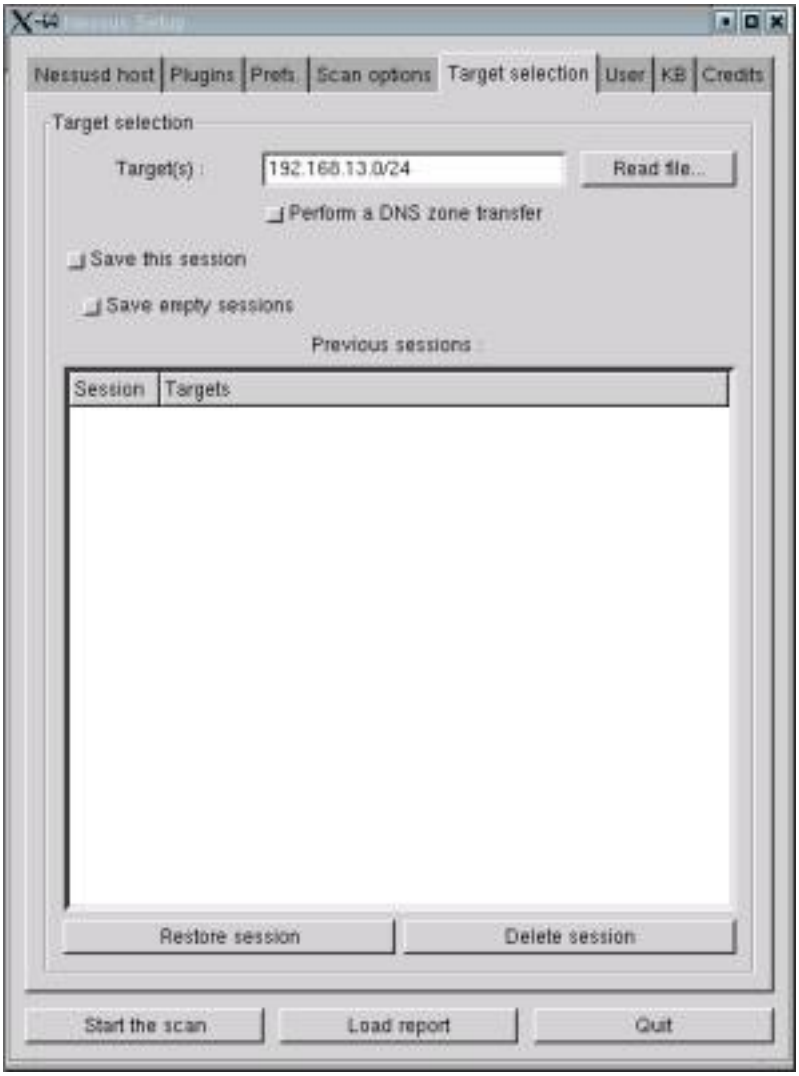


Une fois connecté dans le champ **plugins**, vous verrez la liste des points à vérifier (liste du haut), la liste du bas contenant les détails du point sélectionné dans celle du haut, vous pouvez éventuellement invalider certains points à vérifier. A noter que vous avez des infos bulles fort riches quand vous passez la souris sur un champ. Par défaut tout est sélectionné. Sur le site de **nessus**, vous pouvez trouver d'autres plugins au gré de la découverte de nouveaux trous de sécurité, on vous explique aussi comment les installer, ou éventuellement carrément en créer d'autres tout seul dans son coin.
NOTE Par défaut les plugins qui peuvent faire planter la machine sont désactivés (comme le Denial of Service).



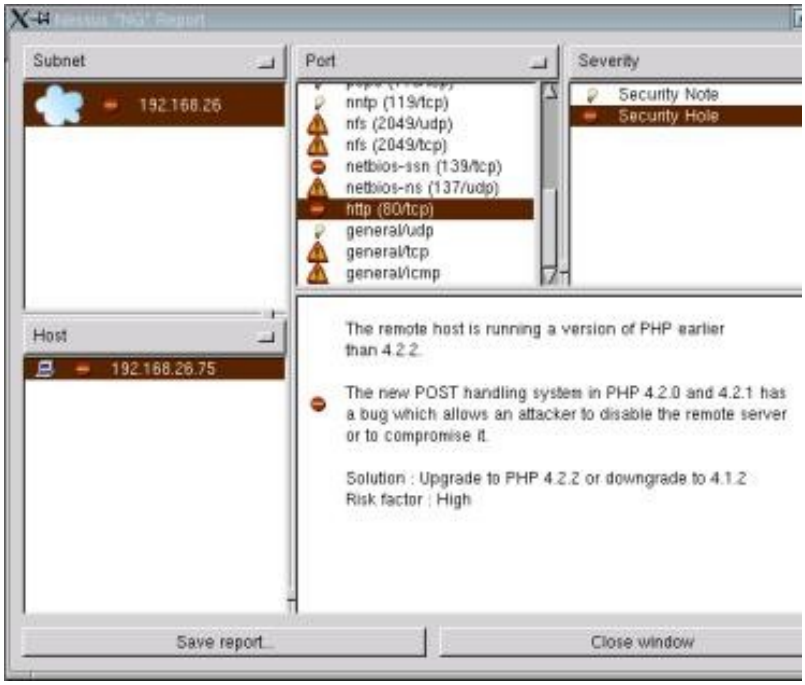
Dans l'onglet **Prefs**, vous pouvez saisir d'autres informations, notamment un nom d'utilisateur valide et son mot de passe si votre machine est aussi serveur POP ou SMB (samba).
Dans l'onglet **Scan options**, vous avez encore d'autres options pour la vérification, je n'y ai pas touché.
Dans l'onglet **User**, vous devez indiquer l'adresse email de la personne destinataire du rapport (**root@locahost** par défaut).
L'onglet **KB** permet de tout sauvegarder ou non dans une base de données

Dans l'onglet **Target Selection**, vous devez indiquer la machine cible dont on va auditer la sécurité en rentrant son adresse IP, éventuellement si vous voulez auditer toutes les machines du réseau 192.168.13.X, saisissez comme dans le screenshot 192.168.13.0/24.



Voilà, on peut lancer maintenant le scan en appuyant sur le bouton "**Start the scan**", une fenêtre apparaît avec une barre d'avancement. Le scan peut être relativement long, ça dépend évidemment de la puissance de votre machine, du coup soyez patient et ne l'interrompez pas même s'il parait bloqué.

Une fois le scan terminé, une fenêtre "**Report**" apparaît avec le détail des trous de sécurité et des warnings, vous pouvez éventuellement sauvegarder le rapport au format désiré (html, txt, latex, ...) (bouton du milieu en bas). On a même une option de sauvegarde en html avec des graphes.
On voit ici que j'ai un trou de sécurité pour PHP, je dois passer à la version 4.2.2.



5 Détecter les attaques en temps réel

5.1 Présentation

Le but de cette page est de vous présenter des outils permettant de détecter en temps réel les attaques lorsque vous êtes connectés à internet, dans cette catégorie vous trouverez des outils de détection passifs se contentant de vous avertir d'attaques, libre à vous de mener les actions en conséquence, vous trouverez ensuite des outils de détection actifs, c'est à dire qu'ils vont eux mêmes automatiquement faire des manips systèmes pour bloquer les attaques le plus rapidement possibles.
Ce type d'outils de détection d'intrusion réseau sont des programmes qui permettent d'analyser le trafic réseau et de remonter les alarmes quand ça devient suspect. On distingue deux grandes familles
- les NIDS (Network Based Intrusion Detection System) qui vérifient la sécurité au niveau du réseau
- les HIDS (Host Based Intrusion Detection System) qui vérifient la sécurité au niveau des machines

Un NIDS est généralement un système dédidé qui vérifie les paquets circulant sur ou plusieurs liens réseau, pour cela il dispose souvent de plusieurs cartes réseau qui fonctionnent en mode promiscuité (promiscuous mode) ce qui permet de capture les paquets qui ne lui sont pas destinés.

Le HIDS est un daemon qui tourne sur une machine particulière qui analyse les journaux de logs et capture également les trames réseau qui entrent ou sortent de l'hôte afin de déceler les intrusions.

Snort présentée plus loin dans ce document est un NIDS, je présente dans ce paragraphe un outil hybride à la fois HIDS et NIDS et qui se nomme **Prelude**.

Par la suite j'aurais tendance à mélanger capteur, sonde ou senseur mais c'est la même chose !

5.2 Prelude

5.2.1 Présentation

Prelude est un IDS qui se décompose en :
- des capteurs (sensors en anglais) ou bien encore sondes qui sont chargés de détecter les évènements et de remonter les alarmes à un manager
- le manager **prelude** est chargé de collecter les évènements rapportés par les différents capteurs. Il archive les informations et permet en outre de définir une action pour contrer une attaque (contre mesure)
- les agents de contre mesure sont chargés de contrer les attaques
- le frontend **prewikka** est une interface d'administration accessible via un navigateur qui lui permet d'avoir en un coup d'oeil les alertes et statistiques associés
La communication entre tous ses outils se fait grâce au format IDMEF (Intrusion Detection Message Exchange Format) basé sur XML (RFC4765).

Les outils concernés sont:
- la bibliothèque **Prelude** (**libprelude**) sur laquelle se base tous les autres outils, elle gère notamment tout le dialogue entre les composants (capteurs et manager), le dialogue comprend la connexion réseau jusqu'à l'authentification.
- la bibliothèque **PreludeDB** fournit des outils de base pour tous les composants pour pouvoir accéder facilement à la base de données sans se soucier de son type (MySQL, SQLlite, et j'en passe)
- le capteur/sonde **Prelude-LML** permet la collecte et l'analyse des informations issues de tous types d'applications émettant des événements sous forme de logs (journaux système, messages syslog, etc.) afin de détecter des activités suspectes et de les transformer en alerte vers le Manager. C'est une sonde qui est locale à une machine.
- le manager **Prelude** est le serveur chargé de recevoir les messages de l'ensemble des sondes. Il enregistre les évènements et les transforme en alerte.
- **prewikka** est l'interface web qui fournit une console d'analyse graphique
- **Prelude-Correlator** est un outil puissant basé sur un langage de programmation qui permet grâce à de multiples informations de diverses sondes et sources d'être corrélées à partir de certaines règles à écrire pour pouvoir générer telle ou telle action.

vous découvrirez sur le site de **prelude** qu'il existe un certain nombre de sondes externes qui ne sont pas présentées dans cette page dont des sondes qui tournent sous windows. On retient que **snort** en fait partie.

A noter qu'il peut y avoir plusieurs manager, dans ce cas il y a souvent un manager principal et des manageurs secondaires qui alimentent le manager principal comme le fait un capteur/sonde. On parle alors de relais (relaying dans la litterature anglo saxonne). Ce type d'architecture est utilisé dans le cas d'un réseau étendu avec plusieurs sous réseau distants (type réseau d'entreprise).

Pour mémoire **Prelude** sait analyser les logs issus de diverses origines dont:

Firewall, Routers & VPN	BIG-IP, Check Point, CISCO ASA, CISCO IOS, CISCO Router, CISCO VPN, D-Link, Ipchains, IpFw, Juniper Networks NetScreen,

	Linksys WAP11, ModSecurity v2, Netfilter, SonicGuard SonicWall
Switchs	CISCO CSS
IDS	CISCO IPS, Portsentry, Shadow, Tripwire
Monitoring	APC-EMU, ArpWatch, Dell OpenManage, Nagios
AntiVirus/AntiSpam	ClamAV, P3Scan, SpamAssassin
Database	Microsoft SQL Server, Oracle
SMTP/POP Server	Exim, Postfix, Qpopper, Sendmail, Vpopmail
FTP Server	ProFTPD, WU-FTPD
Web Server	Apache
Vulnerability Scanner	Nessus
Honeypots	Honeyd, Honeytrap, Kojoney
Authentication	OpenSSH, Su
Applications	Asterisk, Cacti, Libsafe, Shadow Utils, Squid, Sudo
OS (security tools)	GrSecurity, PaX, SELinux
Miscellaneous	Unix specific logs, Webmin, Windows Server, Arbor, Linux bonding, Microsoft Cluster Service, NetApp ONTAP, NTSyslog, OpenHostAPD, Rishi, Suhosin

5.2.2 Installation

Il faudra préalablement avoir installé une base de données, pour ma part j'utilise **MySQL** tel que décrit dans cette page.
La page principale de Prelude est http://www.prelude-ids.org/ on récupérera un certain nombre de tarball qu'on installera dans l'ordre suivant:

5.2.2.1 installation de libprelude

tout d'abord **libprelude**, on décompresse l'archive en tapant

tar xvfz libprelude-0.9.21.2.tar.gz

Cela donne le **libprelude-0.9.21.2** il faudra préalablement installer le package suivant

urpmi libgnutls-devel

on tape ensuite

./configure

cela donne (à la fin)

***** Dumping configuration *****
- Generate documentation : no
- Libtool dynamic loader : System

- LUA binding : no
- Perl binding : yes
- Python binding : yes
- Ruby binding : no
- Easy bindings : no

on tape ensuite **make** puis en tant que root

make install

5.2.2.2 Installation de libpreludedb

on passe maintenant à **libpreludedb**, on décompresse l'archive en tapant

tar xvfz libpreludedb-0.9.15.1.tar.gz

cela donne le répertoire **libpreludedb-0.9.15.1** dans lequel on tape

./configure

voilà le résultat de la commande

***** Dumping configuration *****
- Generate documentation : no
- Enable MySQL plugin : yes
- Enable PostgreSQL plugin : no
- Enable SQLite3 plugin : no
- Perl binding : yes
- Python binding : yes

on tape maintenant **make** puis en tant que root

make install

on rajoute dans le fichier **/etc/ld.so.conf** les lignes suivantes

/usr/local/lib/libpreludedb/plugins/formats
/usr/local/lib/libpreludedb/plugins/sql

puis on tape **ldconfig**

on crée maintenant la base de données correspondante

mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.67 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>CREATE database prelude;

Query OK, 1 row affected (0,02 sec)

on donne les droits d'accès à la base à un utilisateur particulier

mysql>GRANT ALL PRIVILEGES ON prelude.* TO olivier@'localhost' IDENTIFIED BY 'mot-de-passe';
Query OK, 0 rows affected (0,00 sec)

mysql>quit

création des tables de la base

mysql -u olivier prelude -p < /usr/local/share/libpreludedb/classic/mysql.sql

5.2.2.3 Installation de prelude-lml

on passe à **prelude-lml** on décompresse l'archive en tapant

tar xvfz prelude-lml-0.9.14.tar.gz

Cela donne le répertoire **prelude-lml-0.9.14** dans lequel on tape

./configure

voilà le résultat de la commande

***** Dumping configuration *****
- Enable unsupported rulesets: : yes

on tape ensuite **make** puis en tant que root

make install

on édite le fichier **/etc/ld.so.conf** on rajoute la ligne suivante

/usr/local/lib/prelude-lml

puis en tant que root on tape **ldconfig**

Le fichier de configuration est **/usr/local/etc/prelude-lml/prelude-lml.conf** voilà un exemple pour prendre en compte **syslog**

[format=syslog]
time-format = "%b %d %H:%M:%S"
prefix-regex = "^(?P<timestamp>.{15}) (?P<hostname>\S+) (?:(?P<process>\S+?)(?:\[(?P<pid>[0-9]+)\])?:)?"
file = /var/log/messages

pour **apache**

[format=apache]
time-format = "%d/%b/%Y:%H:%M:%S"

prefix-regex = "(?P<hostname>\S+) \S+ \S+ \[(?P<timestamp>.{20}) [+-].{4}\] "
file = /usr/local/apache2/logs/access_log
file = /usr/local/apache2/logs/online-access_log

[format=apache-error]
time-format = "%a %b %d %H:%M:%S %Y"
prefix-regex = "^\[(?P<timestamp>.{24})\] \S+ (\[client (?P<hostname>\S+)\])?"
file = /usr/local/apache2/logs/error_log
file = /usr/local/apache2/logs/apache2/online-error_log

5.2.2.4 Installation de prelude-manager

on passe à **prelude-manager** on décompresse l'archive en tapant

tar xvfz prelude-manager-0.9.14.2.tar.gz

cela donne le répertoire **prelude-manager-0.9.14.2** dans lequel on tape

./configure

voilà le résultat de la commande

***** Dumping configuration *****
- TCP wrapper support : no
- XML plugin support : yes
- Database plugin support: yes

on tape ensuite **make** puis en tant que root

make install

on rajoute les lignes suivantes dans le fichier **/etc/ld.so.conf**

/usr/local/lib/prelude-manager/decodes
/usr/local/lib/prelude-manager/filters
/usr/local/lib/prelude-manager/reports

on tape ensuite **ldconfig**

A présent on va modifier le fichier de configuration **/usr/local/etc/prelude-manager/prelude-manager.conf** voilà les modifs que j'ai faites

tout d'abord la liste des interfaces à écouter

listen = 127.0.0.1
listen = 192.168.2.10

puis des infos sur la base de données

[db]

 # The type of database: mysql, pgsql or sqlite3.

type = mysql

 # Host the database is listening on.
host = localhost

 # Port the database is listening on.
port = 3306

 # Name of the database.
name = prelude

 # Username to be used to connect the database.
user = olivier

 # Password used to connect the database.
pass = mot-de-passe

5.2.2.5 Installation de prewikka

on passe à **prewikka**, on décompresse l'archive en tapant

tar xvfz prewikka-0.9.14.tar.gz

cela donne le répertoire **prewikka-0.9.14** dans lequel on tape en tant que root

urpmi python-cheetah
python setup.py install

on crée maintenant la base de données de **prewikka** en tapant

mysql -u root -p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.0.67 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> CREATE database prewikka;
Query OK, 1 row affected (0,01 sec)

on donne ensuite les droits sur la base sur un utilisateur qui va bien

mysql> GRANT ALL PRIVILEGES ON prewikka.* TO olivier@'localhost' IDENTIFIED BY 'mot-de-passe';
Query OK, 0 rows affected (0,00 sec)

on crée les tables de la base **prewikka**

mysql -u olivier prewikka -p < /usr/share/prewikka/database/mysql.sql

maintenant on va créer un hôte virtuel avec **apache**, on modifie le fichier **httpd.conf** pour y rajouter

```
<VirtualHost 192.168.2.10>
  ServerName ids.kervao.fr
  Alias /prewikka/ /usr/share/prewikka/htdocs/
  ScriptAlias / /usr/share/prewikka/cgi-bin/prewikka.cgi
</VirtualHost>

<Directory "/usr/share/prewikka">
   AllowOverride FileInfo AuthConfig Limit Indexes
   Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
   <Limit GET POST OPTIONS>
      Order allow,deny
      Allow from all
   </Limit>
   <LimitExcept GET POST OPTIONS>
      Order deny,allow
      Deny from all
   </LimitExcept>
</Directory>
```

je vous laisse bien entendu adapter à votre adresse IP et au nom du serveur. On modifie maintenant le fichier de configuration de **prewikka /etc/prewikka/prewikka.conf** voilà les lignes que j'ai modifiées

```
# Default locale to use (default is English):
default_locale: fr

type: mysql
host: localhost
user: olivier
pass: mot-de-passe
name: prelude

[database]
type: mysql
host: localhost
user: olivier
pass: mot-de-passe
name: prewikka
```

5.2.2.6 Installation de prelude-correlator

On passe à **prelude-correlator**, on décompresse l'archive en tapant

tar xfvz prelude-correlator-0.9.0-beta3.tar.gz

Cela donne le répertoire **prelude-correlator-0.9.0-beta3**, préalablement j'ai du installer le package suivant

urpmi lua-devel

on revient dans le répertoire où l'on tape

./configure

puis **make** j'ai droit à l'erreur suivante

prelude-correlator.c:48: erreur: expected '=', ',', ';', 'asm' or '__attribute__' before 'got_signal'

pour la corriger il suffit d'éditer le fichier **./src/prelude-correlator.c** et de rajouter au niveau des includes

#include <signal.h>

on retape **make** et enfin en tant que root

make install

on rajoute maintenant la ligne suivante

/usr/local/lib/prelude-correlator

dans le fichier **/etc/ld.so.conf** et on tape **ldconfig**

5.2.3 Engistrement d'une sonde

5.2.3.1 Présentation

Pour pouvoir utiliser une sonde il faut d'abord l'enregistrer. auprès du manager **Prelude**. Pour cela il faut donner un identifiant unique à la sonde puis l'enregistrer avec les permissions qui vont bien. Dans la litterature anglo saxonne l'identifiant est souvent connu sous le terme de "profile".

Un profil est donc l'identifiant pour une sonde, quand celle-ci est lancée elle va chercher auprès du manager **Prelude** un fichier profil fixant sa configuration.

Pour enregistrer une sonde et donc créer un profil il faudra taper la commande suivante

prelude-admin register (nom du profil) (permissions) (adresse du manager) --uid (uid) --gid (gid)

avec

(nom du profil) le nom de votre sonde

(permissions) il existe plusieurs types de permission, idmef et admin qui peuvent avoir des droits en lecture (read r) ou écriture (w). Généralement une sonde a besoin d'un accès en écriture à la base (**idmef:w**) et en accès en lecture aux fichiers profils (**admin:r**)

(uid gid) on doit indiquer l'uid et le gid qui permettent d'utiliser la sonde et d'accéder aux informations sur le profil

(adresse du manager) c'est l'adresse IP du manager **Prelude**, si c'est une installation locale on peut mettre localhost.

A noter que la première fois qu'une sonde est enregistrée **prelude-admin** crée une clé privée.

La première chose à faire est de créer un profil pour le manager **Prelude** lui même en tapant

prelude-admin add "prelude-manager" --uid 0 --gid 0
Generating 1024 bits RSA private key... This might take a very long time.
[Increasing system activity will speed-up the process].
Generation in progress... X..+++++O.+++++O

Created profile 'prelude-manager' with analyzerID '3897808600812191'.

C'est fait et plus à faire. On peut lancer maintenant le manager en tapant

prelude-manager

voilà le résultat

02 Nov 16:55:41 (process:12957) INFO: Subscribing Normalize to active decoding plugins.
02 Nov 16:55:41 (process:12957) INFO: server started (listening on 127.0.0.1 port 4690).
02 Nov 16:55:41 (process:12957) INFO: server started (listening on 192.168.2.10 port 4690).
02 Nov 16:55:41 (process:12957) INFO: Generating 1024 bits Diffie-Hellman key for TLS...
02 Nov 17:25:49 (process:15259) INFO: Subscribing db[default] to active reporting plugins.

5.2.3.2 Engistrement de la sonde prelude-lml

Pour enregistrer la sonde **prelude-lml** on tapera

prelude-admin register prelude-lml "idmef:w admin:r" localhost

voilà le résultat

*** WARNING: no --uid or --gid command line options were provided.**
*****
*** The profile will be created under the current UID (0) and GID (0). The**
*** created profile should be available for writing to the program that will**
*** be using it.**
*****
*** Your sensor WILL NOT START without sufficient permission to load the profile.**
*** [Please press enter if this is what you intend to do]**

Generating 1024 bits RSA private key... This might take a very long time.
[Increasing system activity will speed-up the process].
Generation in progress... .+++++O.+++++O

You now need to start "prelude-admin" registration-server on localhost:

example: "prelude-admin registration-server prelude-manager"

Enter the one-shot password provided on localhost:

Quelques commentaires, je n'ai pas mis d'uid et de gid par défaut il a mis ceux de root. Par ailleurs j'ai indiqué que le manager tournait localement (**localhost**), vous pouvez évidemment indiquer une adresse distante. Comme il nous l'indique dans le commentaire je tape maintenant sur la machine où tourne le manager (qui est la même machine me concernant)

prelude-admin registration-server prelude-manager

voilà le résultat

The "svgdj6t6" password will be requested by "prelude-admin register"
in order to connect. Please remove the quotes before using it.

Generating 1024 bits Diffie-Hellman key for anonymous authentication...
Waiting for peers install request on :::5553...
Waiting for peers install request on 0.0.0.0:5553...

je reviens sur la machine **prelude-lml** et je tape le mot de passe qui est indiqué

Enter the one-shot password provided on localhost:
Confirm the one-shot password provided on localhost:

Connecting to registration server (localhost:5553)... Authentication succeeded.

sur la machine manager on voit s'afficher ensuite sur la console

Connection from ::ffff:127.0.0.1:38635...
Registration request for analyzerID="46154944204286" permission="idmef:w admin:r".
Approve registration? [y/n]: y

::ffff:127.0.0.1:38635 successfully registered.

voilà **prelude-lml** est enregistré, on peut le lancer en tapant simplemant

prelude-lml

voilà les traces dans la console

02 Nov 20:05:42 (process:17778) INFO: PCRE plugin loaded 443 rules.
02 Nov 20:05:42 (process:17778) INFO: Connecting to 127.0.0.1:4690 prelude Manager server.
02 Nov 20:05:42 (process:17778) INFO: TLS authentication succeed with Prelude Manager.
02 Nov 20:05:42 (process:24978) INFO: /usr/local/apache2/logs/online-error_log: No metadata available, starting from tail.
02 Nov 20:05:42 (process:24978) INFO: /usr/local/apache2/logs/error_log: No metadata available, starting from tail.
02 Nov 20:05:42 (process:24978) INFO: /usr/local/apache2/logs/online-access_log: No

metadata available, starting from tail.
02 Nov 20:05:42 (process:24978) INFO: /usr/local/apache2/logs/access_log: No metadata available, starting from tail.
02 Nov 20:05:42 (process:17778) WARNING: /var/log/everything/current does not exist.
02 Nov 20:05:42 (process:17778) INFO: /var/log/messages: No metadata available, starting from tail.

5.2.3.3 Engistrement de la sonde réseau snort

De la même manière que plus haut, sur le poste où tourne **snort** je tape

prelude-admin register snort "idmef:w admin:r" localhost

*** WARNING: no --uid or --gid command line options were provided.**
*****
*** The profile will be created under the current UID (0) and GID (0). The**
*** created profile should be available for writing to the program that will**
*** be using it.**
*****
*** Your sensor WILL NOT START without sufficient permission to load the profile.**
*** [Please press enter if this is what you intend to do]**

Generating 1024 bits RSA private key... This might take a very long time.
[Increasing system activity will speed-up the process].
Generation in progress... .+++++O+++++O

You now need to start "prelude-admin" registration-server on localhost:
example: "prelude-admin registration-server prelude-manager"

Enter the one-shot password provided on localhost:

sur la machine serveur

prelude-admin registration-server prelude-manager
The "z5l59zd6" password will be requested by "prelude-admin register"
in order to connect. Please remove the quotes before using it.

Generating 1024 bits Diffie-Hellman key for anonymous authentication...
Waiting for peers install request on :::5553...
Waiting for peers install request on 0.0.0.0:5553...

sur la machine où tourne snort je mets le mot de passe indiqué plus haut

Enter the one-shot password provided on localhost:
Confirm the one-shot password provided on localhost:

Connecting to registration server (localhost:5553)... Authentication succeeded.

et voilà ce qu'on retrouve sur le serveur

Connection from ::ffff:127.0.0.1:47647...

Registration request for analyzerID="495528782460288" permission="idmef:w admin:r".
Approve registration? [y/n]: y
::ffff:127.0.0.1:47647 successfully registered.

maintenant je lance **snort** en tapant

snort -c /etc/snort.conf -i eth1

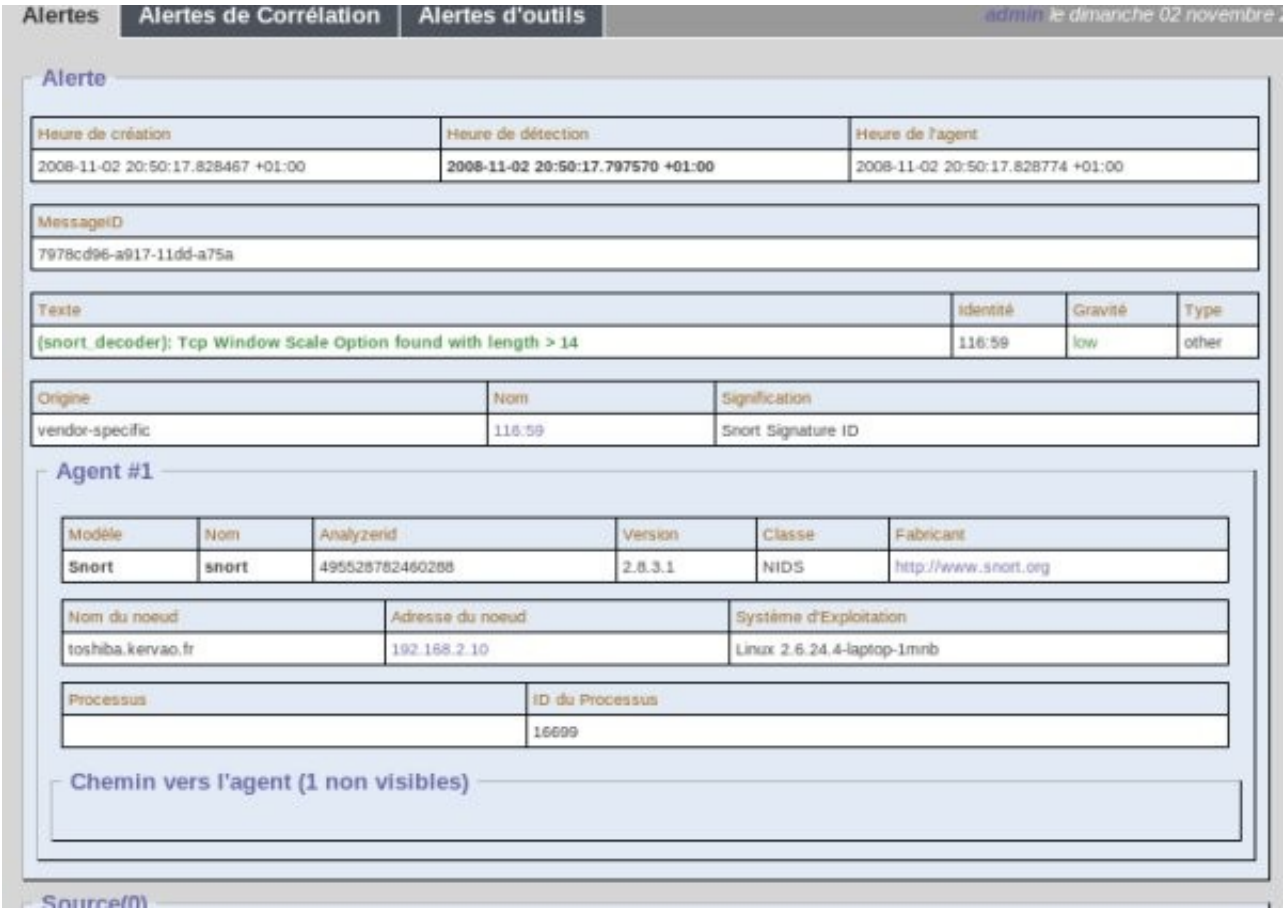
vous pouvez indiquer l'interface qui vous plait. Préalablement il faudra éventuellement créer le répertoire **/var/log/snort**

5.2.4 Le frontend prewikka

Comme indiqué plus haut l'adresse pour accéder au frontend est **http://ids.kervao.fr** le login est par défaut **admin** avec le mode de passe **admin** (qu'on pourra changer). Voilà quelques screenshots

sur ce screenshot on peut voir les deux sondes qui ont été enregistrées ainsi que le manager **Prelude**. Et voilà la page d'alerte

j'ai déclenché un scan avec **nmap** pour générer l'alerte, voilà quand on clique sur l'alerte en question



il y a des infos supplémentaires en descendant plus bas dans la fenêtre.

Pour aller plus loin le manuel d'administration de **Prelude** se trouve par ici
https://trac.prelude-ids.org/wiki/ManualPreludeAdmin

6 "Sniffer" son réseau avec WireShark et Snort

6.1 Présentation

Wireshark est un analyseur de trafic réseau ou "sniffer". Il utilise une interface graphique basée sur **GTK+**, il est basé sur la bibliothèque **libpcap**, qui fournit des outils pour capturer les paquets réseau. Il a pris la suite d'ethereal qui n'est plus maintenu.

Contrairement à **wireshark**, **snort** permet de sniffer le réseau en temps réel alors que le premier permet l'analyse uniquement après une période de capture.

6.2 Libpcap

6.2.1 Présentation

libpcap est une bibliothèque d'outils permettant de faire la capture des paquets qui circulent sur le réseau, on peut ainsi faire des stats, de la surveillance de réseau, du débugage et bien d'autres choses.

6.2.2 Installation

Vous avez le choix entre installer le package fourni avec votre distribution ou d'installer le tarball qu'on trouvera sur le site www.tcpdump.org. L'archive est le tarball **libpcap-1.0.0.tar.gz** qu'on décompressera en tapant :

tar xvfz libpcap-1.0.0.tar.gz

Cela va nous donner le répertoire **libpcap-1.0.0**, dans ce répertoire on tapera pour créer le **Makefile**:

./configure

A présent tapons :

make

NOTE : Les packages suivants sont nécessaires **byacc** et **flex**

Puis en tant que **root**

make install

Si vous avez choisi d'installer avec le package RPM, pour la suite des opérations il vous faudra aussi installer le package de développement **libpcap0-devel**

6.3 Wireshark

6.3.1 Présentation

wireshark a été développé par les développeurs d'**Ethereal**, d'ailleurs il en est issu (c'est un fork), il n'y a que le nom qui a changé. Tout laisse penser qu'**ethereal** a été abandonné au profit de **wireshark**.

6.3.2 Installation

Le site officiel est http://www.wireshark.org/ on y récupère l'archive qu'on décompresse en tapant

tar xvfz wireshark-1.0.4.tar.gz

cela donne le répertoire **wireshark-1.0.4** dans lequel on tape

./configure

voilà le résultat

The Wireshark package has been configured with the following options.

Build wireshark : yes
Build tshark : yes
Build capinfos : yes
Build editcap : yes
Build dumpcap : yes
Build mergecap : yes
Build text2pcap : yes
Build idl2wrs : yes
Build randpkt : yes
Build dftest : yes
Build rawshark : yes

Install dumpcap setuid : no
Use plugins : yes
Build lua plugin : no
Build rtp_player : no
Use GTK+ v2 library : yes
Use threads : no
Build profile binaries : no
Use pcap library : yes
Use zlib library : yes
Use pcre library : yes
Use kerberos library : no
Use GNU ADNS library : no
Use SMI MIB library : no
Use GNU crypto library : no
Use SSL crypto library : no
Use IPv6 name resolution : yes
Use gnutls library : no
Use libcap library : no

puis on tape

make

et en tant que root

make install

on rajoutera dans le fichier **/etc/ld.so.conf** la ligne suivante

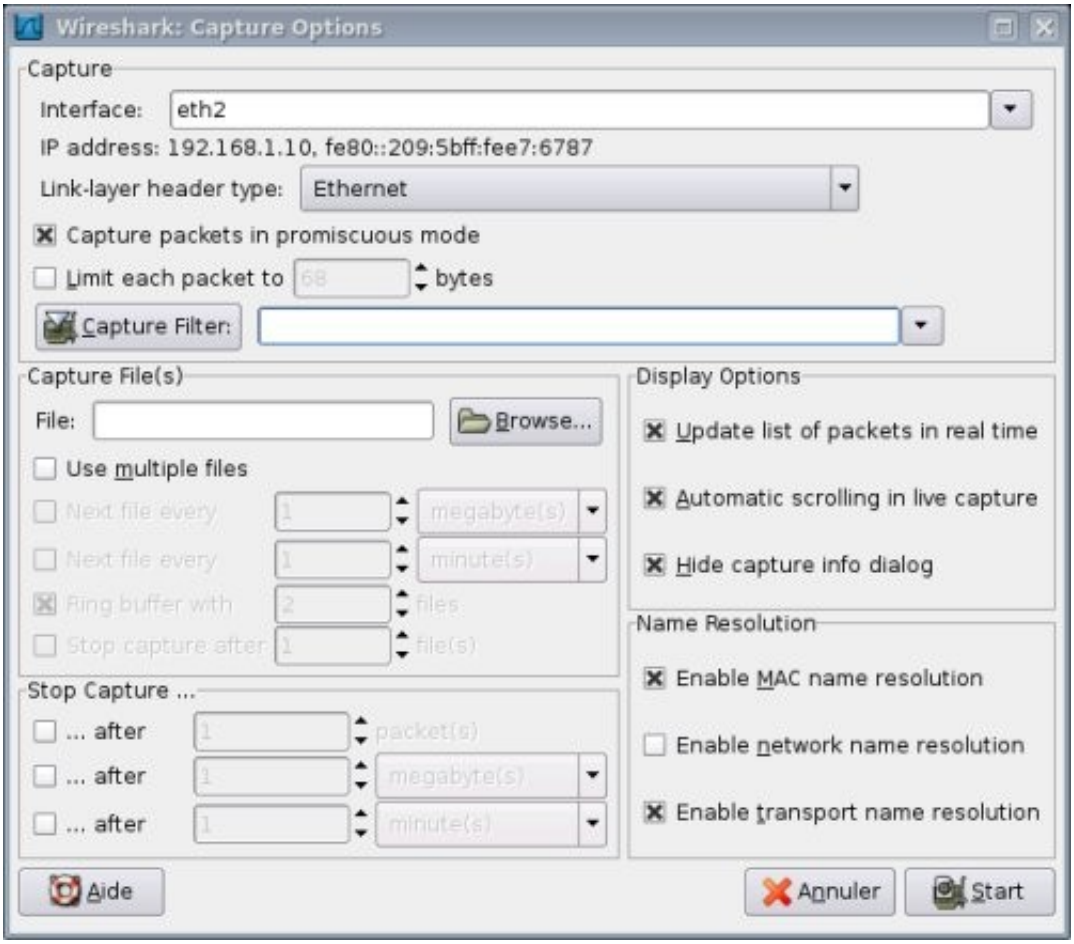
/usr/local/lib/wireshark/plugins/1.0.4

puis on tape **ldconfig**

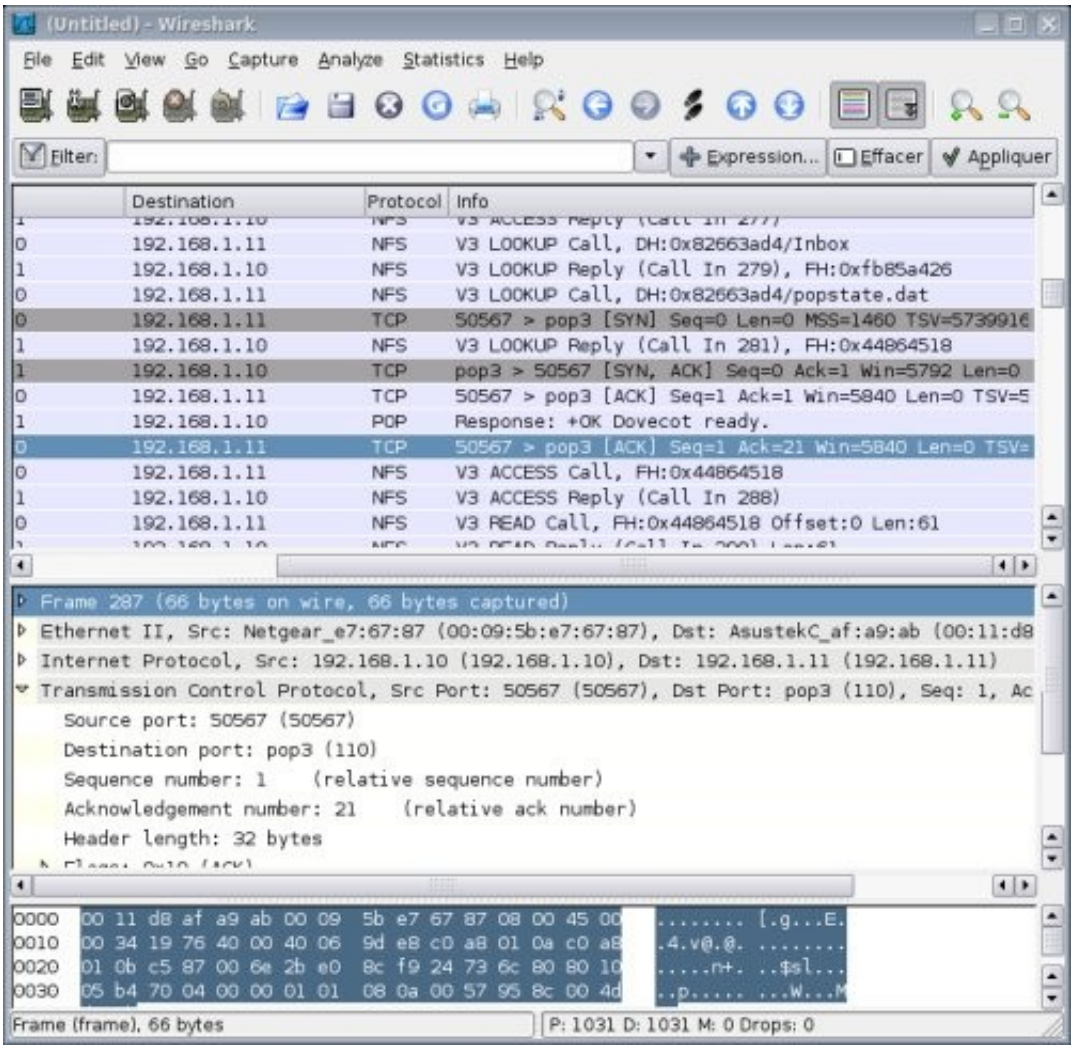
6.3.3 Utilisation

Il suffit de taper **wireshark** en tant que root, dans la barre de menu on choisit **Capture** puis **Interfaces** on voit les interfaces réseau présentes, pour ce qui me concerne ppp0 connexion avec le PDA et eth2 connexion wifi vers Freebox.

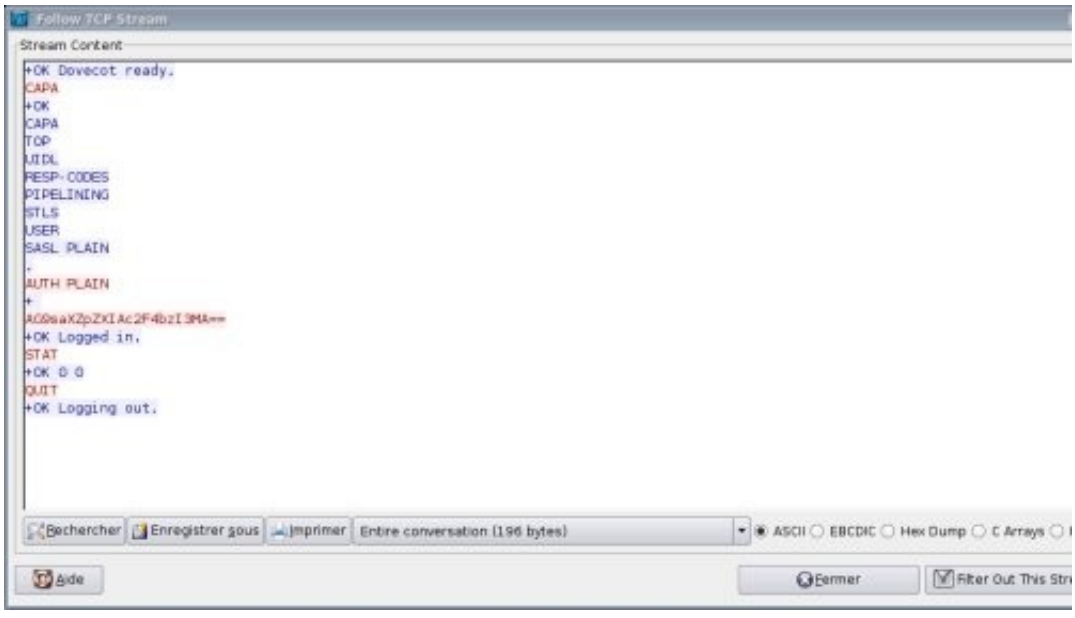
On choisit son interface et on clique sur **Options**



J'ai laissé les options par défaut, puis on commence la capture en cliquant sur **Start**, pour stopper la capture il faut ensuite cliquez sur l'icône correspondante (4 eme à partir de la gauche).



On peut voir une session de récupération de mail (**pop**), on sélectionne la ligne correspondante et avec le bouton droit de la souris on sélectionne dans le menu déroulant **Follow TCP Stream** et voilà le résultat de la session **POP**.



6.4 snort

6.4.1 Présentation

Snort est aussi un sniffer de réseau, à la différence de **WireShark**, il agit en temps réel. Le site officiel de snort est http://www.snort.org.

6.4.2 Installation

On doit préalablement installer **pcre** qui est une bibliothèque fournissant des outils gérant des expressions régulières compatibles avec Perl. L'URL officiel http://www.pcre.org/ on y récupère l'archive qu'on décompresse en tapant:

tar xvfz pcre-7.8.tar.gz

Cela donne le répertoire **pcre-7.8** dans lequel on tape successivement

./configure
make

Puis en tant que root

make install
ldconfig

Ensuite on décompresse l'archive de **snort** en tapant

tar xvfz snort-2.8.3.1.tar.gz

Cela va nous créer un répertoire **snort-2.8.3.1**. Avant d'aller plus loin, vous devez vous assurer que la bibliothèque libpcap est bien installée. On tape d'abord :

./configure --enable-prelude

l'option **enable-prelude** permet de pouvoir intégrer snort à **Prelude** ce dernier doit être prélablement installé. On tape maintenant

make

NOTE Les logs peuvent être archivés dans une base de données, par défaut si MySQL est installé sur votre système et que vous ne voulez pas bénéficier de cette option rajoutez l'option **--without-mysql** à **configure**

Puis en tant que **root**

make install

pour terminer l'intégration avec **Prelude** on crée le fichier **/etc/snort.conf** en y rajoutant la ligne suivante

output alert_prelude: profile=snort

Les exécutables seront placés sous **/usr/local/bin**

on rajoute dans le fichier **/etc/ld.so.conf** la ligne suivante

/usr/local/lib/snort_dynamicengine
/usr/local/lib/snort_dynamicrules
/usr/local/lib/snort_dynamicpreprocessor

puis on tape **ldconfig**

6.4.3 Syntaxe

La syntaxe de la commande est la suivante :

snort -options expression

Les options disponibles sont les suivantes:

-b les paquets sont logués dans un fichier au format **tcpdump** appelé **snort.log**. C'est l'option qu'il faut prendre si on ne veut pas que **snort** perdre du temps à faire la conversion binaire->ASCII pour ne rater aucun paquets, l'option **-r** permet de relire ces fichiers en temps différé.

-c <cf> on utilise le fichier de règles **<cf>**, ce fichier indique ce que le système doit loguer.

- h <hn> on définit ici l'adresse du réseau local, cela sert uniquement pour le formattage du texte pour mettre la flèche qui va bien en fonction du sens du trafic par rapport au réseau (entrant ou sortant)

-i <if> on utilise l'interface **<if>**, par défaut eth0

-s Les alertes sont archivées au travers de **syslog** dans **/var/log/secure**.

-l log-dir pour définir le répertoire d'archivage pour les logs. Par défaut le répertoire est **/var/log/snort**

-d Pour extraire uniquement la couche transport (layer) du paquet

-v mode verbeux, les paquets apparaissent dans la console ou le shell à partir duquel a été lancée la commande, contrairement à l'option **-b** ça ralentit considéralement le fonctionnement de **snort** du coup on peut perdre des paquets, à déconseiller donc, si vous ne voulez pas perdre une miette des échanges de paquets

L'expression fixe les critères pour les paquets qui seront logués. Si aucune expression n'est donnée, tous les paquets seront logués. Une expression consiste en un ou plusieurs primitives,

une primitive consiste en une identité (nom ou adresse) précédée par un ou plusieurs qualificateurs, dont il existe trois types:

- type: pour définir le type précis de l'identité, on a le choix entre **host** pour une machine, **net** pour un réseau et **port** pour un port, par défaut on utilise le type **host**.
Exemple: **host www.breizland.bz** pour l'hôte du même nom, ou encore **net 192.168.13** pour un réseau du type 192.168.13.X

- dir: pour définir la direction à partir ou vers l'identité, les directions possibles sont **src**, pour en provenance de, et **dst**, pour à destination de. Par défaut on prend les paquets dans les deux sens (**src** et **dst**).
Exemple: **src www.breizland.bz** les paquets provenant de l'hôte **www.breizland.bz**, ou encore **dst port 20** les paquets à destination du ports 20 (**ftp**).

- proto: pour restreindre les paquets utilisant un protocole particulier, les protocoles possibles sont : **ether**, **fddi**, **ip**, **arp**, **rarp**, **decnet**, **lat**, **sca**, **moprc**, **mopdl**, **tcp** et **udp**. Si aucun protocole n'est spécifié on capture les paquets quel que soit le protocole utilisé.
Exemple: **ether src www.breizland.bz** les paquets utilisant **ethernet** provenant de **www.breizand.bz**, ou encore **tcp port 21**, les paquets allant ou venant du port 21 et utilisant **TCP**.

6.4.4 Utilisation

Commençons par la base, on va afficher à la console tous les entêtes de paquets :
snort -v

voilà le résultat

```
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Verifying Preprocessor Configurations!

Initializing Network Interface eth1
Decoding Ethernet on interface eth1

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.8.3.1 (Build 17)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
           (C) Copyright 1998-2008 Sourcefire Inc., et al.
           Using PCRE version: 7.8 2008-09-05

Not Using PCAP_FRAMES
04/01-14:19:39.648482 ARP who-has 192.168.1.10 tell 192.168.1.1

04/01-14:19:39.648946 ARP reply 192.168.1.10 is-at 0:9:5B:E7:67:87
```

04/01-14:19:40.461793 192.168.1.10:32862 -> 212.27.54.252:53
UDP TTL:64 TOS:0x0 ID:46579 IpLen:20 DgmLen:61 DF
Len: 33
=+=

04/01-14:19:40.515700 212.27.54.252:53 -> 192.168.1.10:32862
UDP TTL:57 TOS:0x0 ID:0 IpLen:20 DgmLen:77 DF
Len: 49
=+=

(...)

04/01-14:19:40.516710 192.168.1.10 -> 152.46.7.80
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:2562 Seq:1 ECHO
=+=

En faisant un CTRL-C pour stopper la commande, on obtient le bilan suivant sur **eth2** (mon interface wifi connectée à internet via routeur)

***** Caught Int-Signal**

===
Snort received 181 packets
Analyzed: 179(98.895%)
Dropped: 0(0.000%)
Outstanding: 2(1.105%)
===
Breakdown by protocol:
TCP: 161 (89.944%)
UDP: 14 (7.821%)
ICMP: 4 (2.235%)
ARP: 0 (0.000%)
EAPOL: 0 (0.000%)
IPv6: 0 (0.000%)
ETHLOOP: 0 (0.000%)
IPX: 0 (0.000%)
FRAG: 0 (0.000%)
OTHER: 0 (0.000%)
DISCARD: 0 (0.000%)
===
ALERTS: 0
LOGGED: 0
PASSED: 0
===
Snort exiting

Si vous voulez voir le contenu du paquet en plus de l'entête, vous pouvez taper:

snort -vd

voilà le résultat

```
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Verifying Preprocessor Configurations!

Initializing Network Interface eth1
Decoding Ethernet on interface eth1

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.8.3.1 (Build 17)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
           (C) Copyright 1998-2008 Sourcefire Inc., et al.
           Using PCRE version: 7.8 2008-09-05

Not Using PCAP_FRAMES

03/19-10:19:24.566553 192.168.26.100:43631 -> 192.168.26.50:23
TCP TTL:64 TOS:0x10 ID:15513 IpLen:20 DgmLen:133 DF
***AP*** Seq: 0xA6CF7331  Ack: 0xDB1B136  Win: 0x5B4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 144002676 5915219
FF FA 20 00 33 38 34 30 30 2C 33 38 34 30 30 FF  .. .38400,38400.
F0 FF FA 23 00 6D 61 6E 61 2E 6B 65 72 76 61 6F  ...#.mana.kervao
2E 66 72 3A 30 FF F0 FF FA 27 00 00 44 49 53 50  .fr:0....'..DISP
4C 41 59 01 6D 61 6E 61 2E 6B 65 72 76 61 6F 2E  LAY.mana.kervao.
66 72 3A 30 FF F0 FF FA 18 00 58 54 45 52 4D FF  fr:0......XTERM.
F0                                               .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/19-10:19:24.572556 192.168.26.50:23 -> 192.168.26.100:43631
TCP TTL:64 TOS:0x0 ID:62121 IpLen:20 DgmLen:55 DF
***AP*** Seq: 0xDB1B136  Ack: 0xA6CF7382  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 5915226 144002676
FF FD 01                                         ...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
03/19-10:19:24.580523 192.168.26.100:43631 -> 192.168.26.50:23
TCP TTL:64 TOS:0x10 ID:15514 IpLen:20 DgmLen:55 DF
***AP*** Seq: 0xA6CF7382  Ack: 0xDB1B139  Win: 0x5B4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 144002690 5915226
FF FC 01                                         ...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

03/19-10:19:24.580797 192.168.26.50:23 -> 192.168.26.100:43631
TCP TTL:64 TOS:0x0 ID:62122 IpLen:20 DgmLen:70 DF
***AP*** Seq: 0xDB1B139  Ack: 0xA6CF7385  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 5915235 144002690
FF FB 01 54 65 6C 6E 65 74 20 53 65 72 76 65 72  ...Telnet Server
```

```
0D 0A                               ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

03/19-10:19:24.580936 192.168.26.100:43631 -> 192.168.26.50:23
TCP TTL:64 TOS:0x10 ID:15515 IpLen:20 DgmLen:55 DF
***AP*** Seq: 0xA6CF7385  Ack: 0xDB1B14B  Win: 0x5B4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 144002690 5915235
FF FD 01                            ...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

03/19-10:19:24.581011 192.168.26.50:23 -> 192.168.26.100:43631
TCP TTL:64 TOS:0x0 ID:62123 IpLen:20 DgmLen:59 DF
***AP*** Seq: 0xDB1B14B  Ack: 0xA6CF7388  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 5915235 144002690
6C 6F 67 69 6E 3A 20                login:

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

03/19-10:19:24.620312 192.168.26.100:43631 -> 192.168.26.50:23
TCP TTL:64 TOS:0x10 ID:15516 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xA6CF7388  Ack: 0xDB1B152  Win: 0x5B4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 144002730 5915235

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/19-10:19:24.650669 192.168.26.50:33167 -> 213.157.170.39:8400
TCP TTL:64 TOS:0x0 ID:61232 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xE32D07B4  Ack: 0xF80538C9  Win: 0xB50  TcpLen: 32
TCP Options (3) => NOP NOP TS: 5915305 821134246

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

03/19-10:19:24.743261 213.157.170.39:8400 -> 192.168.26.50:33167
TCP TTL:48 TOS:0x0 ID:29714 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0xF8053321  Ack: 0xE32D07B4  Win: 0x16A0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 821134371 5913865
06 70 EA 5C 27 08 7F 01 C1 10 FE 2E AF E2 DE 2F  .p.\'........../
D0 C6 70 28 08 3A 2A 4F 97 5E 2C 31 5F A2 9F FF  ..p(.:*O.^,1_...
67 55 66 32 07 10 19 20 73 79 43 0F 88 CC 9F 2D  gUf2... syC....-
5D 59 C0 0B 8C ED 14 EE 55 65 26 0B D6 28 19 5B  ]Y......Ue&..(.[

(...)

84 2F A2 EE 23 AB 49 FE 9A 64 E3 60 9B E0 9F 77  ./..#.I..d.`...w
C6 92 ED F4 86 DD 28 7A C5 FB B4 F3 36 48 80 22  ......(z....6H."
07 EA 0F 28 B1 E0 2B 4C 4B 22 59 77 F5 1D 3D 54  ...(..+LK"Yw..=T
FF D0 FF F3 30 C4 ED 11 D9 8A AC 00 C3 0A 94 B5  ....0...........
69 43 CA AF F0 CB 66 59 05 64 4F F9 C1 42 EF 4D  iC....fY.dO..B.M

Snort received 270 packets
```

```
   Analyzed: 270(100.000%)
   Dropped: 0(0.000%)
===================================================
    TCP: 143       (52.963%)
    UDP: 88        (32.593%)
   ICMP: 0         (0.000%)
    ARP: 0         (0.000%)
  EAPOL: 0         (0.000%)
   IPv6: 0         (0.000%)
    IPX: 0         (0.000%)
  OTHER: 0         (0.000%)
DISCARD: 0         (0.000%)
===================================================
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
===================================================
Fragmentation Stats:
Fragmented IP Packets: 39       (14.444%)
   Fragment Trackers: 0
   Rebuilt IP Packets: 0
   Frag elements used: 0
Discarded(incomplete): 0
   Discarded(timeout): 0
  Frag2 memory faults: 0
===================================================
```

Vous avez ici un extrait d'une session **telnet**. Si vous voulez afficher les aussi l'entête des paquets **ethernet**, vous pouvez taper :

snort -vde

Maintenant on va loguer les paquets en tapant

snort -vde -l /var/log/snort

Attention le répertoire **/var/log/snort** doit être créé préalablement. Ainsi en tapant cette commande pour la machine d'adresse 192.168.13.15 après avoir fait un **ping** et un **telnet** sur la machine 192.168.13.11, on trouvera dans le répertoire**/var/log/snort**, les répertoires suivants 192.168.13.11 et 192.168.13.15, le premier contenant le fichier **ICMP_ECHO_REPLY** et le deuxième **ICMP_ECHO**, TCP:1025-23. Ces fichiers contenant les paquets résultants du **ping** et du **telnet**.
Pour spécifier un fichier de règles vous taperez :

snort -vde -l /var/log/snort -c snort-lib

Vous trouverez un exemple de fichier de règles sous le répertoire de **snort**, et un certain nombre sur le site même de **snort**.
Maintenant si vous voulez exploiter en temps différé un fichier binaire du format **tcpdump**, vous taperez:

snort -vde -l /var/log/snort -r tcpdump_file

Pour afficher les alertes dans **/var/log/secure** au moyen de **syslog** en utilisant un fichier de règles on tapera:

snort -vde -l /var/log/snort -c snort-lib -s

Pour regarder maintenant un peu les paquets qui circulent lors d'une connexion **PPP**, on peut taper (l'adresse devant **host** correspond à l'adresse IP attribuée par le FAI) :

snort -h 192.168.13.0/24 -d -v host 213.228.15.14 -i ppp0

Voilà un extrait d'une session **POP**

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/16-21:31:09.048632 194.158.97.244:110 -> 213.36.44.26:1292
TCP TTL:59 TOS:0x0 ID:62404  DF
*****PA* Seq: 0xDCC4E1BD   Ack: 0xF1084599   Win: 0x8218
TCP Options => NOP NOP TS: 240412927 876595
2B 4F 4B 20 50 4F 50 33 20 73 65 72 76 65 72 20  +OK POP3 server
4D 65 64 69 61 6E 65 74 2F 31 2E 31 33 20 3C 32  Medianet/1.13 <2
32 37 38 34 2E 39 37 31 37 32 34 36 36 37 40 6D  2784.971724667@m
65 64 69 61 6E 65 74 2D 31 76 2E 67 72 6F 6C 69  edianet-1v.groli
65 72 2E 66 72 3E 0D 0A                          er.fr>..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/16-21:31:09.058729 213.36.44.26:1292 -> 194.158.97.244:110
TCP TTL:64 TOS:0x0 ID:4594  DF
******A* Seq: 0xF1084599   Ack: 0xDCC4E205   Win: 0x7F40
TCP Options => NOP NOP TS: 876617 240412927

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/16-21:31:09.058988 213.36.44.26:1292 -> 194.158.97.244:110
TCP TTL:64 TOS:0x0 ID:4595  DF
*****PA* Seq: 0xF1084599   Ack: 0xDCC4E205   Win: 0x7F88
TCP Options => NOP NOP TS: 876617 240412927
55 53 45 52 20 6F 6C 69 76 69 65 72 2E 68 6F 61  USER mon-login

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/16-21:31:09.208661 194.158.97.244:110 -> 213.36.44.26:1292
TCP TTL:59 TOS:0x0 ID:62405  DF
******A* Seq: 0xDCC4E205   Ack: 0xF10845AE   Win: 0x8218
TCP Options => NOP NOP TS: 240412944 876617

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/16-21:31:09.218671 194.158.97.244:110 -> 213.36.44.26:1292
TCP TTL:59 TOS:0x0 ID:62406  DF
*****PA* Seq: 0xDCC4E205   Ack: 0xF10845AE   Win: 0x8218
TCP Options => NOP NOP TS: 240412944 876617
2B 4F 4B 20 6F 6C 69 76 69 65 72 2E 68 6F 61 72   +OK mon-login
61 75 20 70 6C 65 61 73 65 20 65 6E 74 65 72 20   please enter
70 61 73 73 77 6F 72 64 0D 0A                      password..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/16-21:31:09.218898 213.36.44.26:1292 -> 194.158.97.244:110
TCP TTL:64 TOS:0x0 ID:4596  DF
*****PA* Seq: 0xF10845AE   Ack: 0xDCC4E22F   Win: 0x7F88
TCP Options => NOP NOP TS: 876633 240412944
50 41 53 53 20 73 61 78 6F 32 37 30 0D 0A       PASS mot-de-passe-en-clair..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/16-21:31:09.428660 194.158.97.244:110 -> 213.36.44.26:1292
TCP TTL:59 TOS:0x0 ID:62407  DF
```

********A* Seq: 0xDCC4E22F   Ack: 0xF10845BC   Win: 0x8218**
TCP Options => NOP NOP TS: 240412969 876633

=+
10/16-21:31:09.638641 194.158.97.244:110 -> 213.36.44.26:1292
TCP TTL:59 TOS:0x0 ID:62408 DF
*******PA* Seq: 0xDCC4E22F   Ack: 0xF10845BC   Win: 0x8218**
TCP Options => NOP NOP TS: 240412987 876633
2B 4F 4B 20 6F 6C 69 76 69 65 72 2E 68 6F 61 72 +OK mon-login
61 75 20 68 61 73 20 30 20 6D 65 73 73 61 67 65 has 0 message
73 20 28 30 20 6F 63 74 65 74 73 29 0D 0A   s (0 octets)..