Bernard **Boutherin**
Benoit **Delaunay**

Cahiers de l'Admin

Linux

Sécuriser un réseau

3e édition

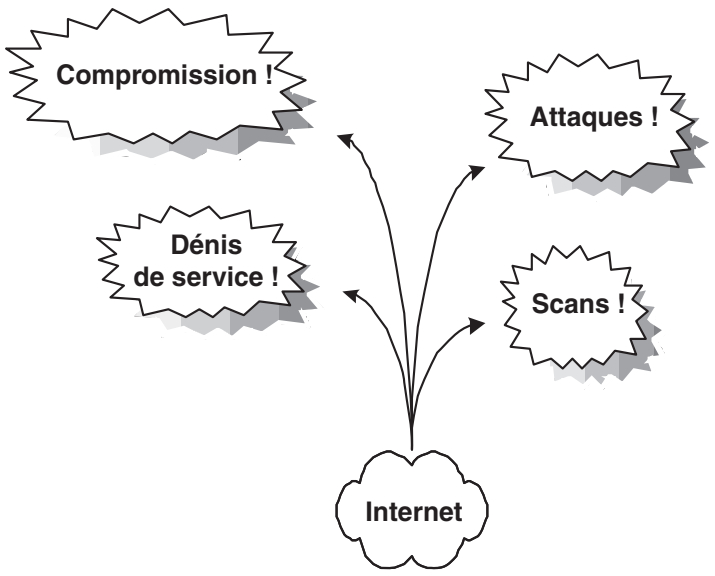
Collection dirigée par Nat **Makarévitch**

ISBN : 2-212-11960-7, ISBN 13 : 978-2-212-11960-2

EYROLLES

Dépôt légal : novembre 2006
N° d'éditeur : 7538
Imprimé en France

chapitre 3



Attaques et compromissions des machines

L'attaque survenue à Tamalo.com offre l'occasion d'analyser les différentes étapes de la compromission d'une machine ainsi que les contre-mesures adéquates. Là encore, il ne faut pas négliger le profil, les motivations et les techniques des pirates pour concevoir un niveau de protection adapté.

SOMMAIRE

MOTS-CLÉS

- kiddies, hackers, crackers
- warez, rebond
- DDOS, buffer overflow
- exploit, scan
- compromission
- rootkit, t0rn, sniffer
- Ethereal
- backdoor
- promiscous
- OSI, MAC
- Logs, core, Whois, CERT, abuse

Le but de ce chapitre est d'alarmer le lecteur par la description d'une intrusion informatique et des outils mis en œuvre par les pirates. Les administrateurs voire les utilisateurs qui ont vécu une telle intrusion deviennent souvent les meilleurs défenseurs du développement de la sécurité informatique.

À partir du cas concret survenu à Tamalo.com (figure 3-1), il s'agit de connaître le mieux possible les différentes étapes de la compromission d'une machine afin d'être à même de contrer efficacement une attaque.

Nous décrirons également comment réagir face à une intrusion dans un système informatique, comment analyser les systèmes compromis, quelles organisations peuvent être utiles dans un tel cas.

Pour commencer, une connaissance du profil, des motivations et des techniques de ceux qui nous attaquent permettra de bien évaluer le risque et d'adapter le niveau de protection de nos systèmes.

Carder, phreaker, hacker, cracker, script kiddies

Le *carder* est impliqué dans la réalisation de fausses cartes bancaires.

Le *phreaker*, est spécialisé dans le vol d'unités téléphoniques dans les autocommutateurs.

Le *hacker* est un expert des systèmes d'exploitation. Il cherche à mettre en évidence les points faibles des systèmes mais s'interdit leur exploitation malveillante.

Beaucoup moins scrupuleux que les *hackers*, les *crackers* n'hésitent pas à utiliser les points faibles des systèmes à des fins nuisibles.

Dépourvus de compétences techniques les *script kiddies* ou plus simplement les *kiddies* utilisent, sans les comprendre, des scripts qui leur permettent de prendre le contrôle de leur cible.

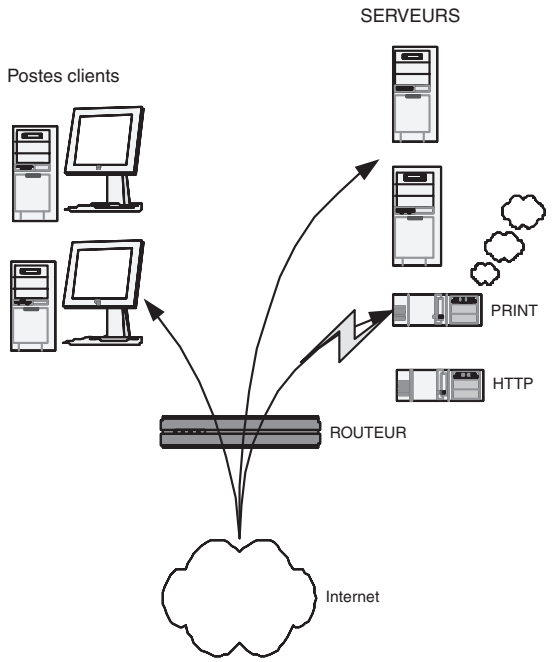


Figure 3–1
Vulnérabilité du serveur d'impression de Tamalo.com

Kiddies, warez et rebonds

Les pirates informatiques se répartissent principalement en deux catégories qui ont chacune une clientèle, des moyens et des objectifs différents.

Le plus grand nombre d'entre eux est constitué par les *kiddies* (« marmots ») ou *script kiddies*, parfois des adolescents, qui épatent leurs amis en prenant la

main sur tel ou tel site plus ou moins connu. Il faut savoir qu'à l'heure d'Internet, il n'est pas nécessaire d'être un *gourou* des systèmes et des réseaux pour être un pirate informatique. Il faut juste une petite dose de curiosité ajoutée à la méconnaissance des risques encourus. Du point de vue pratique, tous les outils sont disponibles sur le Web. Avec quelques mots-clés et un bon moteur de recherche, le jeune pirate se trouvera en quelques minutes en possession d'une panoplie d'outils permettant de prendre le contrôle d'une machine, quelque part dans le monde.

À l'autre extrême, un petit nombre de pirates est issu de la communauté des *crackers*, à ne pas confondre avec les *hackers*. Ces derniers, dans le monde de la sécurité, sont des développeurs système extrêmement pointus dans leur domaine qui éprouvent la sécurité des systèmes dans le but de la renforcer. Les crackers, eux, étudient les failles des systèmes et écrivent des programmes permettant d'en prendre le contrôle. Ils publient ces programmes qui sont alors mis en œuvre par des *script kiddies*.

Les motivations des pirates pour attaquer un site peuvent être de plusieurs ordres. Tout d'abord, la prise de contrôle de machines dans le but d'en utiliser les ressources, par exemple pour y installer un site warez, un robot IRC (Internet Relay Chat) ou un scanner.

Autre motivation possible, l'utilisation de la machine comme rebond, dans le but d'en attaquer une autre. Cette technique est fréquemment utilisée car c'est une garantie d'impunité pour le pirate. En effet, pour remonter la chaîne des machines compromises, il faut contacter les sites correspondants, qui, tour à tour, vont mettre un certain temps à trouver la cause de l'attaque, puis protester auprès du site attaquant. Dans l'hypothèse favorable, tous les administrateurs des sites concernés réagissent et essaient de trouver la cause du problème. Pourtant, après seulement quelques rebonds, le pirate peut être assuré que plusieurs semaines vont s'écouler pour remonter sa piste. Ainsi les traces les plus flagrantes contenues dans les routeurs ne seront plus disponibles le jour où il aurait été possible d'identifier sa machine.

Enfin, dans certains cas, des machines sont compromises afin de constituer un pool de machines sous le contrôle d'un pirate. Le moment venu, ce dernier pourra lancer, à partir de l'ensemble de ces machines compromises, une attaque en déni de service distribué, en anglais DDOS (Distributed Deny Of Service), vers la cible de son choix. Ce type d'attaque peut mettre en jeu plusieurs centaines de machines ! Une telle attaque a été dirigée le 21 octobre 2002 contre les serveurs racines du DNS (Domain Name System) mondial. Ces 13 serveurs racines, root servers, sont à la base de la résolution nom – adresse IP pour toute communication sur Internet. L'attaque a rendu inopérants 9 des 13 serveurs racines, ce qui aurait pu avoir comme conséquence un blocage complet de l'Internet mondial !

B.A.-BA Mettre à jour pour sécuriser

La publication des techniques d'exploitation des failles mène à la correction des sources des programmes vulnérables. Ainsi, il devient de plus en plus difficile pour les pirates d'y découvrir de nouvelles défaillances. L'administrateur avisé ne manquera pas de mettre à jour fréquemment ses programmes grâce à une source sûre et se défiera des codes immatures.

Qu'est-ce que c'est ? Warez

Un site warez est constitué d'une machine sous le contrôle d'un pirate, dotée le plus souvent d'un espace disque important, ainsi que d'un bon accès réseau.
Les pirates y déposent des logiciels, des films ou des fichiers qu'ils veulent distribuer. L'adresse du site piraté est le plus souvent diffusée par l'intermédiaire de canaux IRC.
Une machine warez génère toujours une charge réseau considérable, capable de saturer un lien ADSL de quelques centaines de Kbits/s (trafic montant), aussi bien qu'une liaison spécialisée de plusieurs dizaines de Mbits/s ! C'est parfois même la présence de ce débit anormal qui éveille l'attention de l'administrateur réseau !

Scénario de l'attaque du réseau de Tamalo.com

Une faille dans le système

À l'heure actuelle, les compromissions de machines exploitent pour la plupart une faille du système ou une erreur dans la configuration de ce dernier. La faille la plus classique utilisée pour prendre le contrôle d'un service est le débordement de mémoire-tampon ou *buffer overflow*, en anglais.

B.A.-BA Débordement de mémoire et exécution de code arbitraire

En voici le principe : un service, par exemple wu-ftpd sous Linux, reçoit des arguments depuis l'application cliente. Ces arguments transitent par une zone de mémoire allouée par le service.
Si le service ne vérifie pas correctement les arguments qu'il reçoit (taille…), un débordement de mémoire est possible.
Ainsi, l'application cliente, parfois modifiée par un cracker, envoie au serveur un argument beaucoup trop long. Une partie de cet argument écrase donc le contenu de la mémoire qui suit immédiatement la zone allouée par le service. À cet endroit étaient stockées des instructions qui vont être prochainement exécutées par le processus serveur.
En écrasant ces instructions, le pirate va donc pouvoir faire exécuter le code de son choix à la machine, sous l'identité du service !
Si ce dernier tourne en mode privilégié, il est probable que le client aura bientôt un accès sur le compte administrateur `root` de la machine…

L'exploitation de la faille (« exploit »)

La publication d'une nouvelle faille entraîne une course contre la montre entre les crackers et les développeurs des systèmes. Les premiers travaillent à l'écriture d'un « exploit » : il s'agit d'un programme permettant d'exploiter la faille, c'est-à-dire de prendre le contrôle de toute machine employant la version vulnérable. Les autres travaillent à l'écriture d'une nouvelle version ou d'un correctif, nommé *patch* en anglais.

À ce jeu-là, les pirates ont malheureusement toujours l'avantage, car même si dans la plupart des cas le patch arrive avant l'exploit, celui-ci n'est jamais déployé à temps sur l'ensemble du parc immense que constitue Internet.

Le cracker met son exploit à la disposition de la communauté des pirates par l'intermédiaire d'un certain nombre de sites Internet connus dans ce milieu. De leur côté, les développeurs diffusent le patch de sécurité.

Utilité des scans réseau

Si vous avez un jour la curiosité d'analyser le trafic à la frontière de votre réseau, vous aurez la surprise de voir que les scans y sont permanents. Cela signifie qu'il y a toujours quelqu'un, quelque part, en train d'analyser votre parc informatique pour voir si une machine de votre réseau n'offre pas, par hasard, une faille connue.

Si vous êtes en charge de ce réseau, ne négligez pas toute cette surveillance. En effet, une compromission est le plus souvent précédée d'un scan. C'est l'élément qui permet au pirate d'avoir la liste des machines et des services de votre réseau qui présentent une vulnérabilité.

Par exemple, un attaquant pourra scanner le port 80 de l'ensemble de votre réseau afin d'établir la liste des serveurs web accessibles depuis l'extérieur.
Le scanner n'est pas un outil réservé aux pirates. Un administrateur réseau se doit d'en posséder un pour établir la liste des services offerts sur son réseau. Le scanner le plus réputé disponible sous Linux s'appelle Nmap ; il est téléchargeable à partir de l'URL suivante :

- http://www.insecure.org/nmap/nmap_download.html

Outils Scanner réseau

Un scanner est un programme qui balaye une plage de ports sur un ensemble de machines, afin d'établir la liste des couples machine/service ouverts.

- Le scan horizontal consiste à scanner un port sur un ensemble de machines.
- Le scan vertical consiste à scanner une plage de ports sur une même machine.

La compromission

Les pirates ont le plus souvent une connaissance minimale du site choisi pour cible. Ils organisent une attaque pendant une période où la vigilance de l'administrateur est relâchée. Ainsi, les attaques ont souvent lieu la nuit, le week-end, les jours fériés ou pendant les périodes de vacances.

C'est souvent une accumulation de détails plus ou moins anodins qui laisse suspecter la compromission d'une machine.

C'est ainsi qu'au retour des vacances de Noël, un développeur de Tamalo.com se plaint du comportement anormal de la commande `ps` : elle rend un `segmentation fault` en lieu et place de la liste des processus attendue.

Quelques commandes permettent rapidement de cerner le problème :

1. Déterminer où se trouve le programme `ps`.

```
which ps
/bin/ps
```

2. Examiner la date de dernière modification puis de création du fichier.

```
ls -l /bin/ps
ls -lc /bin/ps
```

1. Examiner la nature du fichier `ps`...

```
file /bin/ps
ELF 32-bits LSB executable, blabla.., not stripped ?????
```

L'attribut `not stripped` est étrange pour une commande système. En général, les commandes système sont déployées pour économiser de l'espace disque et la table des symboles n'y figure pas. Cet attribut ne devrait donc pas apparaître, ce qui laisse redouter que le programme ne provient pas de la distribution.

De plus, si le résultat de `ls -l /bin/ps` paraît acceptable, `ls -lc /bin/ps` révèle que le fichier a été créé le 26 décembre dernier ; un administrateur aurait-il installé un nouveau `ps` pendant les vacances ?

Par ailleurs, les utilisateurs de la machine concernée se plaignent de ralentissements et d'accès disque permanents, ce que ne confirme pas la commande `ps`… sauf si cette commande a été modifiée par un pirate afin de dissimuler son activité.

Une seule conclusion s'impose : il est urgent de débrancher la machine du réseau afin de faire des vérifications plus approfondies. Ce n'est pas sans conséquence pour Tamalo.com car ce serveur contient le référentiel CVS qui gère les versions de logiciels.

Le temps de ressortir les CD-Rom de la distribution Red Hat, de recompiler de manière statique quelques commandes originales (`ps`, `ls`, `netstat`) et il va être possible de mesurer l'ampleur des dégâts.

ATTENTION

Dans certains cas, le programme au comportement anormal qu'on soupçonne d'avoir été installé par les pirates (ici `ps`) peut contenir une bombe qui se déclenchera après un certain nombre d'invocations. L'existence même d'un doute nous contraint à douter de l'ensemble des programmes installés. On comprendra que la première chose à faire est de réamorcer sur un système sûr et de sauvegarder afin de préserver les données des utilisateurs mais aussi les programmes du système pour une analyse ultérieure. Il faut donc invoquer le moins possible de programmes sur la machine potentiellement compromise.

B.A.-BA **En cas de compromission…**

Il est impossible d'affirmer que l'analyse d'une machine compromise permettra de détecter toutes les modifications qu'elle a pu subir. C'est pourquoi la compromission d'un système doit conduire à sa réinstallation complète. Le système de fichiers doit être entièrement reformaté. Cette règle est essentielle pour garantir le retour à un système d'exploitation sain après piratage.

C'était le premier contact pratique de Tamalo.com avec un piratage informatique. Dans ce qui suit, nous allons analyser plus en détail les traces laissées par les pirates au cours de cette compromission afin de mieux connaître les outils utilisés et les failles de nos systèmes qui ont été exploitées.

Analyse de la machine compromise

Traces visibles sur le système avant réinitialisation

En cas de compromission d'une machine, il est utile de récupérer quelques indicateurs de l'état du système d'exploitation avant tout redémarrage. En effet, les pirates disposent parfois d'outils permettant d'effacer leurs traces après un redémarrage.

Il faut en premier lieu s'assurer que les commandes utilisées sont saines. Certaines commandes peuvent avoir été modifiées par les outils dont dispose le pirate, par exemple afin de masquer sa présence. Ces modifications ont pu être faites à différents niveaux :

- Modification des commandes : `ps`, `ls`, `netstat`, `find`, `du`, `passwd`, etc. Dans ce cas, la parade consiste simplement à recopier la commande d'origine sur le système.
- Modification des bibliothèques dynamiques : les fonctions incluses dans les bibliothèques dynamiques sont utilisées lors de l'exécution d'une commande (la commande `ldd /bin/ps` fournira la liste des bibliothèques utilisées par la commande `/bin/ps`). Pour y remédier, il suffit de compiler (bien entendu pas sur la machine compromise) les commandes en mode statique afin qu'elles n'en dépendent plus. Pour cela, utilisez l'option de compilation `-static` de gcc. L'exécutable résultant contiendra le code de toutes les fonctions utilisées, il ne chargera aucune bibliothèque au moment de son exécution.
- Modification des modules du noyau : si les modules du noyau sont modifiés, on considèrera que l'analyse à chaud ne peut pas apporter de résultat fiable et on passera directement, après sauvegarde, à la réinstallation complète.

Dans notre exemple, l'analyse à chaud dévoile quelques anomalies qui seront confirmées par la suite.

La commande `netstat -tupan` sur le système fait apparaître un service en écoute sur le port 15 000, invisible auparavant.

La commande `ps` modifiée nous cachait quelques processus, dont le programme : `/usr/sbin/nscd`, qui est ici une *backdoor* SSH en écoute sur le port 15 000. Grâce à cette porte dérobée, le pirate pouvait revenir se connecter

Investigation Forensique

Forensique, en anglais *forensic*, est synonyme de criminalistique. Une investigation forensique a pour objectif de prouver l'existence d'un crime et de déterminer l'identité de l'auteur ainsi que son mode opératoire.

L'analyse forensique d'une machine compromise est donc du ressort des autorités de police judiciaire plutôt que le travail courant d'un administrateur système ! Pour autant, il n'est pas inutile de connaître et d'avoir mis en œuvre les outils et méthodes d'une telle analyse, et ce pour deux raisons. D'une part, en cas de compromission grave, cela permettra d'éviter les fausses manipulations qui risqueraient d'avoir pour conséquence l'effacement de certaines traces du pirate… des traces qui seront indispensables si un dépôt de plainte est envisagé ! D'autre part, la connaissance du mode opératoire des pirates permet d'être mieux armés pour se protéger contre de futures attaques.

OUTILS **« rescue » et « live CD »**

Il est nécessaire de réamorcer le système afin de bénéficier, lors de l'exploration du contexte, d'un environnement logiciel de provenance sûre, autonome donc non dépendant des éléments installés et non exposé aux modifications intempestives de programmes déployés par le pirate. La disquette « rescue » produite durant l'installation exploite d'ordinaire les programmes placés sur le disque dur. On lui préférera donc les distributions de Linux disponibles sous forme « live CD », c'est-à-dire utilisables grâce à leur seul CD-Rom et sans installation. Certaines furent conçues et réalisées en fonction de cet objectif, d'autres sont si riches qu'elles intègrent cela.

sur notre machine de façon discrète. Les connexions du pirate sont chiffrées. Elles ne sont pas journalisées par le système ; le pirate n'est pas détectable par les commandes `who` ou `w`.

Enfin, la commande `ifconfig` indique que l'interface réseau est en mode `PROMISCUOUS`, ce qui laisse penser qu'un *sniffer* réseau a été installé sur la machine.

ATTENTION **nscd**

`nscd` est aussi le nom d'un démon tout à fait honorable (le Name Server Cache Daemon). La présence de `nscd` n'implique pas que la machine est piratée ! Notons que le risque de confusion est délibéré de la part du pirate.

Sauvegarde du système compromis

Chaque partition est sauvée sur un autre système à l'aide des commandes `dd` pour le dump et `nc` (netcat) pour le transfert réseau :

```
machine-saine> nc -l -p 10101 > fich-hda1
machine-compromise> dd if=/dev/hda1 | nc machine-saine 10101
```

`nc` écoute sur le port 10101

BON SENS **Choix des noms de fichier**

Dans le cas où plusieurs machines sont compromises, faites apparaître le nom de la machine compromise dans le nom du fichier : par exemple `tamalo1-hda1` plutôt que `fich-hda1`.

Analyse fine de l'image du disque piraté

L'analyse à froid du système sera faite en poursuivant différents objectifs :

1. Déterminer la date précise de la compromission initiale. La connaissance de celle-ci permettra un certain nombre de corrélations avec les fichiers de journalisation du routeur d'entrée et des machines du réseau.
2. Déterminer la faille exploitée pour prendre le contrôle du système. Il sera alors possible de mettre à jour le service correspondant.
3. Déterminer la nature des outils installés par le pirate et en identifier les fichiers de traces et les programmes afin de rechercher sur d'autres machines du site des signes éventuels de compromission.
4. Connaître la source de l'attaque afin de la contacter pour avoir des explications (attention : il est très probable que la machine attaquante soit elle-même sous le contrôle du pirate).
5. Déterminer jusqu'à quel point l'intrusion a réussi, savoir si les mots de passe du réseau ont pu être compromis.

OUTILS **Application client/serveur avec netcat**

Le logiciel netcat, fourni en standard avec Linux, permet de réaliser très simplement une application client/serveur.

Le serveur est lancé sur la machine `host1.tamalo.com` pour écouter sur le port `99999` avec la commande :

```
nc -l -p 99999
```

Le client est lancé sur la machine `host2.tamalo.com` pour se connecter sur `host1.tamalo.com:99999` avec la commande :

```
nc host1.tamalo.com 99999
```

Tout message envoyé sur l'entrée standard `<sdtin>` du client `nc` qui s'exécute sur `host1` – c'est-à-dire saisi au clavier de `host1` – apparaîtra sur la sortie standard `<stdout>` du serveur `nc` qui tourne sur `host2` – c'est-à-dire à l'écran de `host2`.

Cette commande est très utile pour analyser le fonctionnement de certaines applications serveurs, comme on le verra au chapitre 7 pour l'analyse du fonctionnement de FTP actif.

▸ http://netcat.sourceforge.net

Montage pour l'analyse

Pour l'analyse, il reste à monter (en loopback) le système de fichiers concerné.

```
mount -o loop,ro,noexec,nodev fich-hda1 /root/
host_compromis_hda1
```

Il peut être utile d'utiliser l'option `ro` (read only) pour ne pas altérer les traces sur le système compromis. De plus, pour éviter d'exécuter par erreur des commandes sur la machine compromise, on utilisera l'option `noexec`. Enfin, on pourra ajouter l'option `nodev` pour ignorer les fichiers de type `device`, qui sont des points d'entrée vers les périphériques.

Étude des fichiers de démarrage et configuration

Pour déceler les traces sur l'image du disque d'une machine piratée, le plus simple est de commencer par l'étude des fichiers de démarrage, très souvent modifiés par les pirates, afin de :

- masquer certaines traces en cas de redémarrage de la machine ;
- relancer un certain nombre de processus : backdoor, scanner, sniffer, à chaque redémarrage du système.

Sous Linux, il faut s'intéresser aux fichiers et aux répertoires suivants :

- `/etc/inittab`
- `/etc/init.d/`
- `/etc/rc.sysinit`
- `/etc/sysconfig/`
- `/etc/rc.d/`
- `/etc/inetd.conf` ou `/etc/xinetd.conf` et `/etc/xinetd.d/`
- `/etc/crontab`
- `/etc/cron.daily`, `/etc/cron.hourly`...

Étude des fichiers créés lors du piratage

Bon sens

Aucune démarche de recherche n'est éternellement valide car les attaquants disposent d'outils de plus en plus évolués.

Il convient aussi de rechercher les fichiers créés le jour du piratage, à l'aide d'une simple commande `find`. Il faut préférer une recherche basée sur la date de création du fichier `CTIME`, qui n'est modifiée que par le noyau, plutôt que sur la date de modification `MTIME`. En effet, la date de modification peut-être altérée facilement par le pirate à l'aide de la commande `touch`.

Analyse avec The Coroner toolkit

The Coroner Toolkit, ou TCT est une panoplie d'outils forensiques destinés à l'analyse d'une machine compromise. TCT fournit des outils très performants pour analyser une machine compromise. Dans ce qui suit, nous allons l'utiliser pour affiner notre analyse et retrouver certaines traces moins évidentes laissées par le pirate.

TCT appuie sa démarche de recherche sur le recoupement des événements temporels. Pour cela, il introduit la notion de *MAC time*, MAC étant l'acronyme de Modification Access Creation. En effet, la connaissance de ces trois attributs d'un fichier peut fournir des informations décisives sur l'acti-

vité du pirate. Par rapport à une simple commande `find`, TCT ajoute la détermination de l'*access time* qui est impossible en passant par les appels système standards. Pour que cette détermination soit possible, le système de fichiers doit avoir été sauvegardé par une copie des partitions, comme le permet `dd`, et non par une commande d'archivage ou de copie de fichiers qui altérerait l'*access time*.

La commande `grave-robber` capture les informations utiles dans l'image du système de fichiers et renseigne la base de données de TCT. Notez que cette même commande capture également les informations concernant les processus et les connexions réseau actives sur un système vivant.

```
grave-robber –c /host_compromis_hda1 –o LINUX2 –m -i
Le fichier /root/tct-1.15/data/tamalo1_01_23_17\:36\:37+0100/
body contient la base de données de TCT.
```

Il est possible de compléter les informations fournies par `grave-robber` avec des informations sur les fichiers effacés, grâce à la commande `fls` issue de la boîte à outils *sleuthkit*. Les formats étant compatibles, il suffit de rediriger la sortie de `fls` pour compléter la base de données de TCT comme indiqué ci-dessous :

```
fls –f linux-ext2 –r –m /host_compromis_hda1 fich-hda1 >> /
root/tct-1.15/data/tamalo1_01_23_17\:36\:37+0100/body
```

La commande `mactime` ci-dessous génère, à partir de la base de données de TCT, la liste chronologique des modifications du système de fichiers.

```
mactime –p /host_compromis_hda1/etc/passwd –g /
host_compromis_hda1/etc/group 1/1/1971 > mactime-tamalo1.out
-p indique le chemin du fichier passwd utilisé pour résoudre les
noms d'utilisateurs
-g indique le chemin du fichier de groupes
Sont considérés tous les événements postérieurs au 1/1/1971, on
n'en rejette donc aucun.
```

La figure 3-2 montre les répercussions sur le système de fichiers de l'activité du pirate pendant la configuration du rootkit t0rn (voir section suivante). On voit assez clairement le déroulement des opérations, qui commence par une lecture attentive de la documentation de SSHD (nul n'est parfait) ! Notez la présence des attributs `m`, `a`, `c`, ainsi que des références à des fichiers effacés marqués `(deleted)`. Dans ce cas précis, les références à ces fichiers ne sont pas d'une grande utilité car le pirate ne s'est pas donné la peine d'effacer ses programmes sources.

Dans certains cas au contraire, on pourra être très motivé pour récupérer un fichier effacé afin d'en analyser le fonctionnement, s'il s'agit par exemple du code source d'un exploit.

Référence The Coroner Toolkit

Les outils du Coroner toolkit sont disponibles à l'adresse http://www.porcupine.org/forensics/
Pour ceux qui veulent en savoir plus sur l'analyse forensique, le livre (en anglais) de Dan Farmer et Wietse Venema, *Forensic Discovery*, est disponible en libre téléchargement sur ce même site.

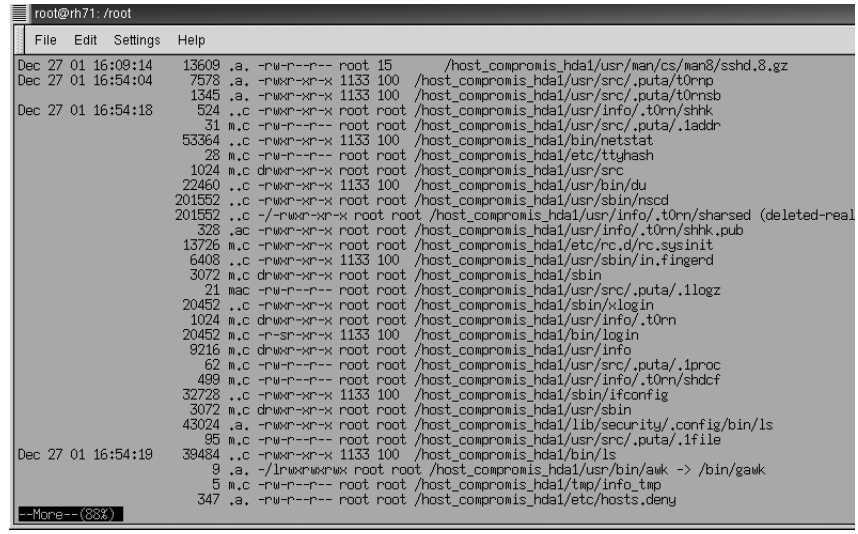


Figure 3–2
Fichiers lus et modifiés par le pirate au moment de la compromission

Sachez que TCT fournit la méthode et les outils nécessaires à une telle récupération. Pour cela, on s'appuiera sur le fait que le système d'exploitation incrémente linéairement les *inodes* des fichiers créés dans un même répertoire comme le montre la figure 3-3.

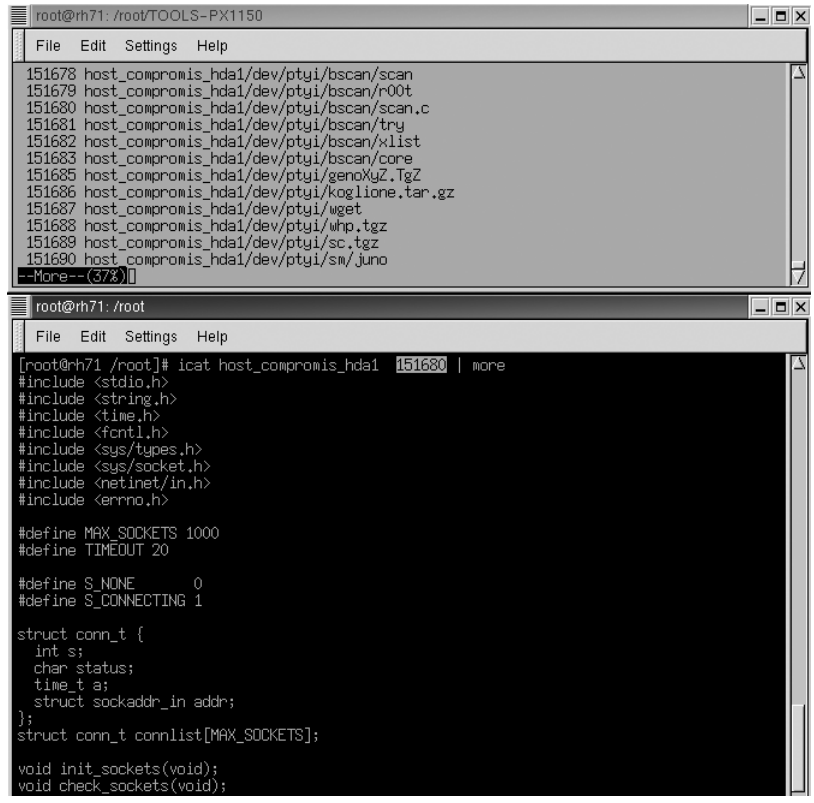


Figure 3–3
Recherche d'un fichier et visualisation à partir de son inode

Ainsi pour récupérer le code source d'un exploit, on recherchera des fichiers, toujours présents sur le disque, créés immédiatement avant et après le fichier perdu. Un fichier effacé sera caractérisé par un trou dans la séquence des inodes du répertoire. Si les blocs occupés par le fichier manquant n'ont pas été réaffectés, il sera possible de le visualiser. La figure 3-3 montre comment voir le contenu du fichier `scan.c` dont l'inode est 151 680, à l'aide de la commande `icat` fournie par TCT.

Trousse à outils du pirate : le rootkit t0rn

Un rootkit est défini par l'Agence nationale de sécurité américaine (National Security Agency) comme une panoplie de logiciels utilisés par des pirates. Cette panoplie fournit des outils pour :

- capturer le trafic réseau et les mots de passe ;
- créer des portes dérobées (backdoors) dans le système ;
- collecter sur le réseau des informations sur d'autres systèmes (scanner) ;
- dissimuler que le système est compromis.

Dans notre exemple, le rootkit t0rn a été installé par le pirate. Il s'agit d'un classique du genre. À l'exception du scanner, il implémente toutes les fonctionnalités prévues par la définition.

Sur notre machine, nous avons trouvé deux répertoires utilisés par le rootkit : `/usr/src/.puta` et `/usr/info/.t0rn`.

Sniffer réseau d'un rootkit

L'exécutable `/usr/src/.puta/t0rnp` est un sniffer réseau, c'est-à-dire un programme qui écoute le réseau dans le but de récolter les éventuels mots de passe qui transitent en clair. Il faut savoir que les applications que nous utilisons couramment, comme TELNET, FTP, IMAP ou HTTP, n'effectuent en général aucun chiffrement du mot de passe au moment où ce dernier est envoyé sur le réseau.

Des logiciels tels que tcpdump, disponibles en standard sous Linux, permettent de constater combien il est facile d'écouter sur le réseau. Notons aussi Ethereal et dsniff qui sont faciles à installer.

La figure 3-6 retrace les échanges de paquets au cours de la phase d'authentification dans une session FTP. Cet exemple est facile à reproduire avec un PC Linux. Il met en évidence le risque lié à l'utilisation des applications non chiffrées.

1. Lancer Ethereal à partir du compte `root`.
2. Sélectionner le menu *capture start* et indiquer `port ftp` dans la rubrique *filter* (voir figure 3-4).

OUTILS dsniff, Ethereal et tcpdump

Ces performants outils d'analyse rendent de grands services aux administrateurs réseau. Leur utilisation est très simple et constitue une aide importante pour comprendre le fonctionnement des applications client/serveur.

- dsniff : http://monkey.org/~dugsong/dsniff/
- ethereal : http://www.ethereal.com
- tcpdump : http://www.tcpdump.org

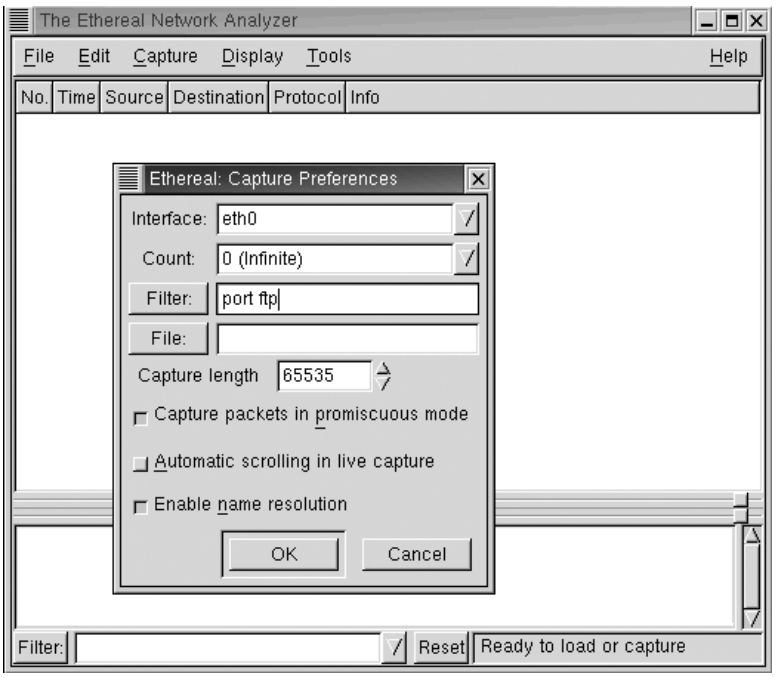


Figure 3–4
Lancement de Ethereal

Ethereal est démarré. Il écoute le réseau en ne conservant que les paquets correspondant au protocole FTP.

3 Ouvrir une connexion FTP vers un serveur quelconque pour analyser le trafic correspondant, comme indiqué à la figure 3-5.

Figure 3–5 Ouverture d'une connexion FTP

OUTILS Sniffer et analyse réseau

Une différence essentielle entre un sniffer et un outil d'analyse de réseau est que le premier est écrit dans l'unique but d'extraire des couples : identification/mot de passe, tandis que le second permet d'analyser l'ensemble du trafic qui circule sur la couche de transport sur laquelle la sonde est posée.

La figure 3-6 montre que les informations circulent en clair sur le réseau. Ainsi, le paquet numéro 13 contient le nom de l'utilisateur `bernard`, dont nous avons écouté la connexion, tandis que le paquet numéro 17 nous renseigne sur son mot de passe : `u1arpl7` !

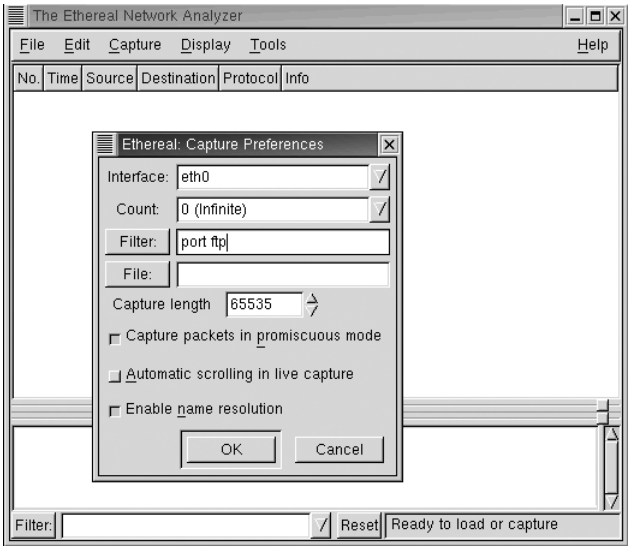


Figure 3–6 Écoute d'une session FTP avec Ethereal

Afin d'écouter sur le réseau, le pirate doit faire passer la carte Ethernet en mode promiscuous. Sur une machine Linux, et sur les systèmes Unix en général, cela nécessite un accès privilégié (compte `root`).

Le mode promiscuous

Pour comprendre ce qu'est le mode promiscuous, il est nécessaire de faire référence au modèle OSI.

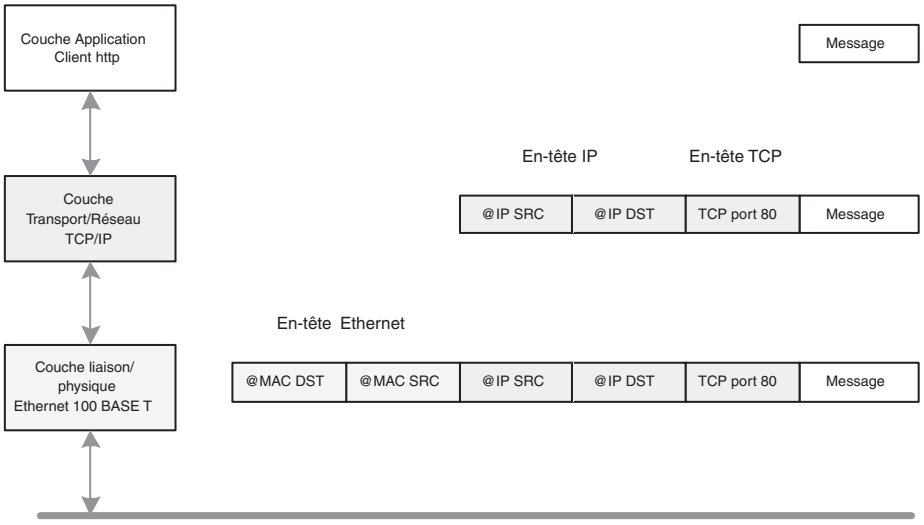


Figure 3–7 Modèle OSI : encapsulation des messages

RAPPEL Adresse MAC

Une adresse MAC (Media Access Control) est constituée de 2 champs de 3 octets chacun :

`06:60:B0:59:DE:B3`

Les trois premiers octets constituent le champ fournisseur tandis que les trois autres constituent un numéro de série. Deux interfaces Ethernet ne peuvent pas avoir la même adresse MAC – une même machine pouvant avoir plusieurs interfaces.

Notez que l'adresse MAC de destination est placée au début de la trame, ce qui permet à la carte Ethernet de déterminer tout de suite si elle doit garder la trame ou non.

La figure 3-7 montre une trame Ethernet arrivant à destination d'une machine. Elle est traitée par les couches basses du pilote réseau : les couches 1 (physique) et 2 (liaison) du modèle OSI (Open Systems Interconnection). Ces couches sont généralement implémentées dans le microcode de la carte Ethernet. Les couches supérieures, 3 et suivantes, sont quant à elles implémentées dans le noyau de Linux.

Une fonction importante des couches basses est de vérifier si la trame Ethernet est destinée ou non à la machine considérée. Cette opération est effectuée en comparant l'adresse MAC de destination contenue dans la trame avec celle de l'interface. Si les adresses coïncident, la trame est déshabillée de son en-tête Ethernet et le paquet est reconstitué pour être transmis aux couches supérieures implémentées dans le noyau de Linux. Dans le cas contraire, il ne tient pas compte de la trame. Cela ne fait pas l'affaire des mécanismes d'écoute du réseau, quelles qu'en soient les motivations. La mise en mode promiscuous de la carte Ethernet remédie à ce problème en obligeant cette dernière à transmettre toutes les trames au noyau qui se chargera de faire le tri.

Sur le système compromis, le fichier `/usr/src/.puta/system` contient les couples « nom de compte – mot de passe » qui ont été enregistrés par le sniffer lors de l'écoute frauduleuse. L'examen de ce fichier montre que le sniffer a capturé l'identifiant et le mot de passe de deux comptes appartenant à Tamalo.com, sur les machines `dial025.mon-fai.com` et `www.diffusion-tamalo.com.it`, comme le montre la figure 3-8. Les deux connexions qui ont pu être sniffées sont des sessions FTP non chiffrées.

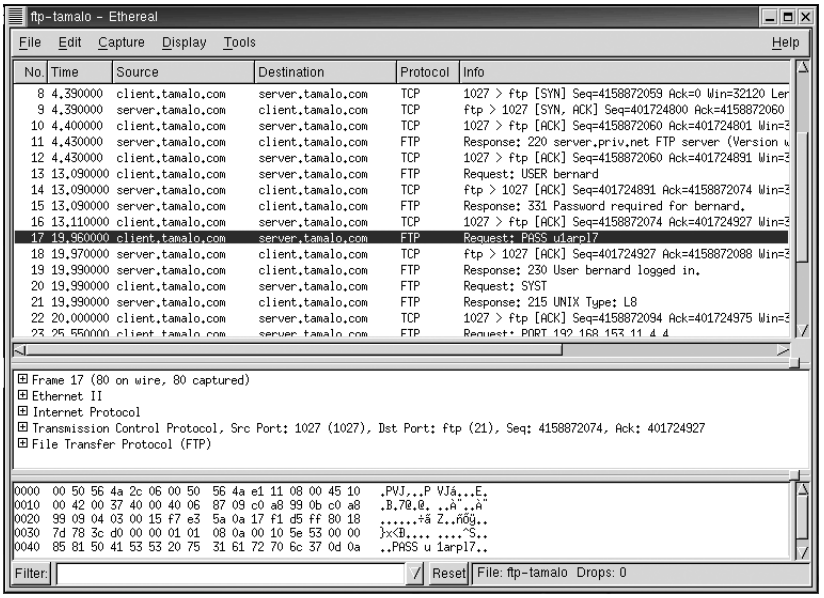


Figure 3–8 Fichier de sortie du sniffer réseau

Action

Quelques mesures doivent être mises en place sans délai :

1. Le changement des mots de passe interne et externe de tous les utilisateurs du réseau.
2. La vérification de l'historique de toutes les connexions sur des sites distants afin de savoir si des connexions frauduleuses ont eu lieu.
3. L'installation d'applications client/serveur mettant en œuvre du chiffrement pour se protéger des écoutes sur le réseau.

Rootkit : effacer les traces et masquer la présence du pirate

Le programme `/usr/src/.puta/t0rnsb` est un nettoyeur de fichiers de journalisation (logs). Son rôle est d'effacer les traces de passage du pirate.

En effet, les services de Linux sont généralement conçus pour enregistrer par l'intermédiaire d'un processus appelé `syslogd` un certain nombre de traces dans des fichiers journaux, configurés grâce au contenu du fichier `/etc/syslogd.conf` (ou `/etc/syslog-ng/`). Le plus souvent, ces fichiers sont situés dans le répertoire `/var/log`. Par défaut, les deux fichiers les plus importants du point de vue de la sécurité sont `/var/log/messages` et `/var/log/secure`.

Exemple de log

```
Jan 12 17:21:26 gw sshd[14168]:
Accepted password for root from
192.168.40.176 port 1034 ssh2
```

Cette ligne de log est produite par le serveur sshd. Elle indique une connexion SSH depuis la machine dont l'adresse IP est `192.168.40.176` sur la machine `gw`. L'heure de la connexion et l'utilisateur, `root`, sont précisés.

Un autre type de trace est enregistré dans le fichier `/var/log/wtmp`. Ce fichier stocke sous forme binaire une trace de chaque connexion sur la machine considérée. Il est possible de visualiser ces connexions avec la commande `last`. Le rootkit fournit donc au pirate des outils pour faire disparaître les traces qui le concernent afin d'éviter que l'adresse de la machine à partir de laquelle il nous a attaqués n'apparaisse dans ces fichiers.

Les outils de nettoyage des logs sont plus ou moins évolués, se situant entre la mise à zéro pure et simple du fichier, pour les rootkits primitifs, et la suppression des lignes concernant la période de présence du pirate pour les plus évolués.

Dans certains cas, l'observation des fichiers de log, même nettoyés par le rootkit, fournira des informations précieuses à l'administrateur de la machine. Par exemple, l'absence totale de logs pendant une période donnée nous indique que le pirate était probablement connecté pendant ce créneau. Si nous disposons des logs du routeur d'entrée pour cette période, ils seront riches d'informations.

À RETENIR

Il faut bien comprendre que tous les programmes, même développés localement, s'ils sont « linkés » dynamiquement, dissimuleront tout ce qui relève du piratage en cours (fichiers…) si le pirate est allé jusqu'à instrumenter la libC.

Le fichier `/usr/src/.puta/.1file` contient la liste des fichiers cachés à l'utilisateur de la machine par les commandes `ls` et `find` modifiées. Il con-

tient en particulier les fichiers du rootkit lui-même : les répertoires `.puta` et `.t0rn`, les fichiers `.1file`, `.1addr`...

Le fichier `/usr/src/.puta/.1addr` contient le début des adresses IP cachées à l'utilisateur de la machine par la commande `netstat` modifiée. Seul le début de l'adresse est donné (`192.168` par exemple) pour que la découverte de ce fichier ne trahisse pas le nom de la machine originaire de l'attaque.

Le fichier `/usr/src/.puta/.1proc` contient la liste des processus cachés à l'utilisateur de la machine par la commande `ps` modifiée. Le sniffer t0rnp est dans la liste, ainsi que la porte dérobée : `nscd`.

Le fichier `/usr/src/.puta/.1logz` contient la liste des adresses IP filtrées par le rootkit et qui n'apparaîtront pas dans les logs.

Rootkit : la porte dérobée (backdoor)

RAPPEL Porte dérobée (backdoor)

Une porte dérobée est un programme qui ménage un accès privilégié, discret et direct à celui qui sait l'employer.
De nombreux ports sont connus pour être utilisés comme portes dérobées. Certains pirates, plus fainéants que les autres, effectuent directement un *scan* de ces ports espérant y trouver une *backdoor* ouverte par un autre !
Le site http://ports.tantalo.net/index.php donne une liste des utilisations des ports, officielles ou comme portes dérobées.

Les rootkits permettent généralement au pirate d'ouvrir une porte dérobée qui écoute sur un port de son choix, supérieur à 1 024 la plupart du temps.

Dans le cas de t0rn, la porte dérobée est constituée par un serveur sshd qui écoute sur le port 15 000. Le démon `ssh` est appelé `/usr/sbin/nscd` (prétendument *Name Server Cache Daemon*).

Notez que la connexion de l'attaquant est chiffrée, ce qui lui évite d'être lui-même sniffé par ses propres outils ou par tout autre outil d'analyse réseau !

Les fichiers de configuration de ce service sont dans le répertoire `/usr/info/.t0rn` :

- `/usr/info/.t0rn/shhk.pub` contient la clé publique du serveur sshd.
- `/usr/info/.t0rn/shhk` contient la clé privée du serveur sshd.
- `/usr/info/.t0rn/shdcf` est le fichier de configuration de sshd. On y découvre qu'il est lancé sur le port 15 000.

En cas de redémarrage du système, ce « `nscd` » est relancé par le script `/etc/rc.sysinit`.

Le processus `nscd` est caché à l'exécution de `ps` grâce au fichier `/usr/src/.puta/.1proc`.

Rootkit t0rn : conclusion

Une analyse rapide du rootkit`t0rn` montre combien la découverte des fichiers de configuration de ce dernier peut être précieuse pour nous. Par exemple, en recoupant les plages d'adresses IP cachées avec les logs des routeurs d'entrée, nous sommes à même de déterminer l'adresse IP de la machine qui nous a attaqués.

Pour cette raison, des rootkits plus évolués fournissent parfois des outils de chiffrement de leurs propres fichiers de configuration. Il existe par exemple des rootkits possédant un fichier de configuration unique chiffré par un « ou

exclusif » de chaque octet avec 255. Heureusement, le décryptage de ces fichiers constitue un défi qui motive de nombreux amateurs, et l'algorithme de chiffrement de ces fichiers ne tarde jamais à être diffusé sur Internet.

Encore une fois, le fait que de nombreux rootkits soient identifiés et leur fonctionnement bien connu ne dispense en aucun cas du reformatage du système de fichiers ni de la réinstallation complète d'une machine compromise.

Détecter la compromission à partir des logs

Depuis quelques semaines, nous avions mis en place chez Tamalo.com une centralisation systématique des logs des machines Linux vers une machine de collecte n'offrant aucun autre service à l'extérieur. Cette centralisation va nous aider à retrouver les logs qui ont étés effacés par le rootkit sur notre serveur.

Copie de log retrouvée sur la machine de collecte : une attaque en deux phases

```
Dec 26 19:02:22 Tamalo1 inetd[158]: printer[5336] from 192.168.20.29 1623 (1)
Dec 26 19:02:22 Tamalo2 inetd[175]: printer[5227] from 192.168.20.29 1627
Dec 26 19:03:55 Tamalo1 SERVER[2320]: Dispatch_input: bad request line (2)
'BBÜóÿ¿Ýóÿ¿Þóÿ¿ßóÿ¿XXXXXXXXXXXXXXXXXX0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000480000000
00001073835088security0000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000006????????????????????????????????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????????????????????????????????????????????
?????????????1Û1É1À°FÍ??å1Ò²f?Ð1É?ÈC?]øC?]ôK?Mü?MôÍ?1É?EôCf?]ìfÇEî^O'?Mð?Eì?EøÆEü^P?Ð?MôÍ??ÐCCÍ??ÐC
Í??Ã1É²??ÐÍ??ÐAÍ?ë^X^?u^H1À^F^G?E^L°^K?ó?M^H?U^LÍ?èãÿÿÿ/bin/sh'
```

L'analyse de la copie des logs sur la machine de collecte met clairement en évidence les deux étapes de l'attaque. Dans un premier temps, à 19:02:22 (1), un scan horizontal de notre réseau sur le port d'impression 515 identifie les machines qui présentent une faille. Dans un deuxième temps, à 19:03:55 (2), cette faille est exploitée sur Tamalo1. L'enchaînement entre le scan et l'exploit est extrêmement rapide (1 minute 30), ce qui suppose que le pirate a utilisé un script qui lance automatiquement l'opération en fonction du résultat du scan.

Il apparaît que nous avons été la victime d'une attaque sur le service d'impression `lprng`. Remarquez dans la ligne de log précédente la taille importante de l'argument fourni à `lpr`, ainsi que la présence de la chaîne de caractères `/bin/sh` à l'intérieur de cet argument.

Cet argument est passé par `lprng` au `syslogd` par l'intermédiaire de la fonction `use_syslog()`. Cette dernière présente une faille : elle utilise l'argument reçu par le service pour déterminer le format de la chaîne de caractères à transmettre. Un format inattendu contenant un appel à `/bin/sh` sera donc exécuté par le service `lprng` autorisant le pirate à exécuter le code de son choix.

Cette vulnérabilité du service d'impression `lprng` de Linux était parfaitement décrite dans un avis de sécurité de la société Red Hat qui nous était parvenu quelques temps auparavant.

Avis de sécurité de Red Hat sur la vulnérabilité du service lprng

```
---------------------------------------------------------------------
-----
Red Hat, Inc. Security Adviso ry
Synopsis: LPRng contains a critical string format bug
Advisory ID: RHSA-2000:065-04
Issue date: 2000-09-26
Updated on: 2000-10-04
Product: Red Hat Linux
Keywords: LPRng security lpd printing lpr syslog
Cross references: N/A
---------------------------------------------------------------------
-----
1. Topic:
LPRng has a string format bug in the use_syslog function which
could
lead to root compromise.

2. Relevant releases/architectures:
Red Hat Linux 7.0 - i386
3. Problem description:
LPRng has a string format bug in the use_syslog function. This
function returns user input in a string that is passed to the
syslog()
function as the format string. It is possible to corrupt the
print
daemon's execution with unexpected format specifiers, thus
gaining root
access to the computer. The vulnerability is theoretically
exploitable
both locally and remotely.
```

Origine de l'attaque

Une négligence du pirate donnera de façon non ambiguë l'adresse de la machine à partir de laquelle il nous a attaqués. Dans le répertoire `/dev/ptyi` utilisé par le pirate pour déposer quelques outils, nous avons découvert un fichier `core`.

Une simple commande `strings` appliquée à ce fichier extrait l'ensemble des chaînes de caractères contenues dans la mémoire. Elle nous dévoile l'environnement complet dans lequel travaillait le pirate au moment de l'incident !

B.A.-BA Fichier core

Un fichier `core` est créé lorsque le noyau Linux interrompt sans sommation le déroulement d'un processus tentant de commettre une action interdite, par exemple accéder à une portion de la mémoire ne lui appartenant pas. Ce fichier est une image de la mémoire occupée par le processus au moment du problème. Un fichier `core` peut être ouvert avec un débogueur afin de connaître l'endroit exact du plantage, ainsi que l'environnement complet du programme au moment de l'incident.

Chaînes de caractères extraites du fichier core

```
HOME=/root
USER=root
LOGNAME=root
PATH=/usr/sbin:/sbin:/usr/bin:/bin:/usr/X11R6/bin
MAIL=/var/spool/mail/root
SHELL=/bin/tcsh
SSH_CLIENT=192.168.16.58 1029 15000
SSH_TTY=/dev/pts/2 TERM=xterm HOSTTYPE=i386-linux VENDOR=intel
OSTYPE=linux MACHTYPE=i386
```

Pour des raisons de confidentialité, les deux premiers octets de l'adresse source de l'attaque ont été remplacés par `192.168`.

La variable `SSH_CLIENT` indique très clairement que la machine dont l'adresse IP est `192.168.16.58` était connectée sur le port 15 000 de notre serveur, ce qui correspond à la porte dérobée.

Une interrogation des bases *whois* détermine rapidement la provenance de l'attaque.

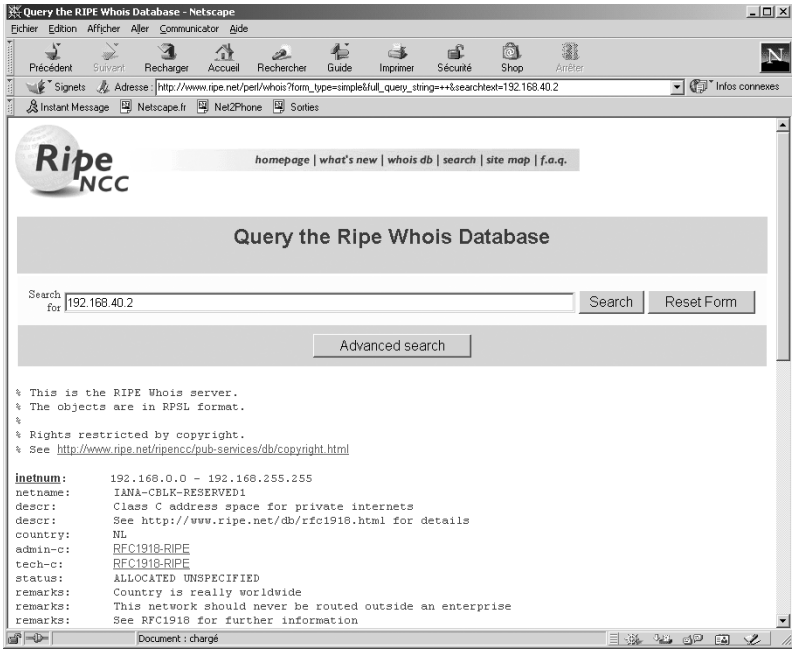


Figure 3–9 Les bases whois : Ripe

L'organisation qui gère l'adresse de notre pirate n'étant pas localisée en France, nous décidons de contacter notre CERT afin qu'il transmette nos récriminations à son homologue.

En parallèle, nous tentons une protestation par courrier électronique à l'adresse `abuse` du réseau source de l'attaque, ainsi qu'avec le contact technique indiqué dans la base whois. Nous obtenons très rapidement une

ORGANISMES Les bases Whois

Les bases whois déterminent à quel organisme a été affecté un domaine IP. Elles fournissent les coordonnées du responsable fonctionnel des adresses attribuées et de la personne à contacter en cas de problème. Ces bases sont au nombre de trois, RIPE pour l'Europe et l'Afrique, ARIN pour les États-Unis et APNIC pour l'Asie. Elles peuvent être interrogées à partir de leur site Web.

- http://www.ripe.net/perl/whois
- http://www.arin.net/whois/index.html
- http://www.apnic.net/

réponse à notre courrier, l'administrateur de la machine concernée nous indiquant qu'il venait de découvrir que sa machine était également compromise.

En résumé…

Les attaques des systèmes informatiques sont de plus en plus automatisées. Leur scénario est assez reproductible : découverte d'une faille dans un service, publication d'un « exploit », scan réseau et tentative de compromission.

Les pirates utilisent des panoplies d'outils qui cachent leur présence, capturent les mots de passe circulant sur le réseau, installent des portes dérobées ou enfin scannent un autre réseau.

L'analyse d'une machine compromise fait apparaître les outils mis en œuvre par le pirate. Elle révèle comment le réseau a été pénétré et si d'autres machines présentent les mêmes failles. Elle permet souvent d'identifier l'origine de l'attaque, mais rarement de remonter jusqu'au pirate qui, en général, se protège par de nombreux rebonds.

Pour se protéger contre ces attaques, deux types d'actions seront décrits dans les chapitres qui suivent : le recours à des applications client/serveur mettant en œuvre du chiffrement pour interdire l'écoute réseau, le filtrage des services vulnérables par la mise en place de pare-feu et la segmentation du réseau.

ORGANISMES Les CERT : Computer Emergency Resource Team

Un CERT est une organisation qui travaille sur les problèmes de sécurité informatique pour une communauté donnée. En France, il y en a quatre. Le plus ancien est le CERT Renater. Il concerne la communauté université-recherche. Chaque CERT dispose de moyens techniques et humains propres. Dans certaines affaires, les CERT peuvent travailler en relation avec les autorités judiciaires.

Au niveau mondial, les CERT sont en relation entre eux par le biais d'un forum appelé FIRST (Forum of Incident Response and Security Team). Les CERT échangent ainsi des informations sur les failles nouvelles et les incidents de sécurité courants.

Les CERT effectuent une veille technologique par rapport aux failles des logiciels pouvant donner lieu à une attaque. Ils diffusent des avis de sécurité à leurs correspondants et les avertissent par des messages d'alerte lorsque certaines attaques prennent des proportions très importantes.

- http://www.cert.org
- http://www.certa.ssi.gouv.fr

CONVENTION L'adresse abuse

Il est recommandé à l'administrateur d'un domaine nommé `nom.de.domaine` de créer l'adresse abuse électronique correspondante (`abuse@nom.de.domaine`) qui est redirigée vers la sienne.

Une personne qui aurait à se plaindre d'un comportement anormal d'une des machines de `nom.de.domaine` pourrait ainsi le signaler à l'administrateur dudit domaine par l'envoi d'un simple courrier électronique.

L'adresse abuse s'avère également utile pour l'administrateur du réseau concerné. Par exemple, c'est grâce à cette adresse qu'il sera informé en cas de problème avec des machines de son domaine.