


MANDIANT[®]




Kris Harms
Peter Silberman

INTRUSION RESPONSE REALITY CHECK

MANDIANT

2 **USA vs Slovenia**

- Score Update



MANDIANT

3 MANDIANT

- APT and CDT experts
- VISA Qualified Incident Response Assessor (QIRA)
- Located in
 - Washington
 - New York
 - Los Angeles
 - San Francisco
- Services, software, and education



MANDIANT

4 MANDIANT Intelligent Response (MIR)

- Collect indicators from thousands of agents
- Index and search the results
- Live IR on thousands of systems at once
- From disk images to registry keys to live memory forensics



MANDIANT

5

Introductions



Kris Harms

– IR Engagement Lead, Instructor



Peter Silberman

– Researcher / Engineer, Co- Author of
Memoryze and Audit Viewer, Malware
Analysis Team



6

Important note

**All information is derived
from MANDIANT observations
in non-classified environments.**

**Some information has been sanitized
to protect our clients' interests.**



7

Agenda

- Why Most Defenders Lose
- A Few Malware Samples and Attacker Techniques
- How to Win
- A Few Investigation Techniques That Work Today



8

Why Defenders Lose



VS



9

Why Defenders Lose

COMPLIANCE \neq SECURITY

MANDIANT

10

Why Defenders Lose



MANDIANT

15

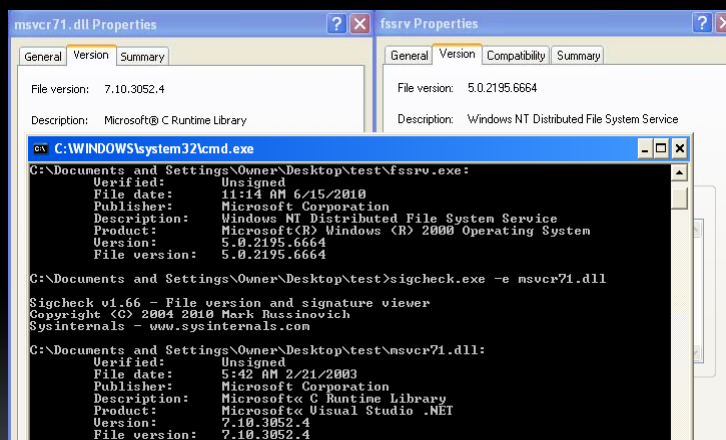
Well...It Depends

- Sample A
 - Obfuscated shellcode
 - Built in
 - Keylogger functionality
 - Ability to download functionality
- Unknown functionality
 - Compromised accounts?
 - Exploit component?
 - Pivot component?



16

Hiding in plain sight



The screenshot shows two windows: 'msvcr71.dll Properties' and 'fssrv Properties'. The 'msvcr71.dll Properties' window shows 'File version: 7.10.3052.4' and 'Description: Microsoft® C Runtime Library'. The 'fssrv Properties' window shows 'File version: 5.0.2195.6664' and 'Description: Windows NT Distributed File System Service'. Below these is a command prompt window titled 'C:\WINDOWS\system32\cmd.exe' showing the output of the command 'sigcheck.exe -e msvcr71.dll'. The output shows that the file is unsigned and has a file version of 7.10.3052.4.

```

msvcr71.dll Properties
-----
General  Version  Summary
File version: 7.10.3052.4
Description: Microsoft® C Runtime Library

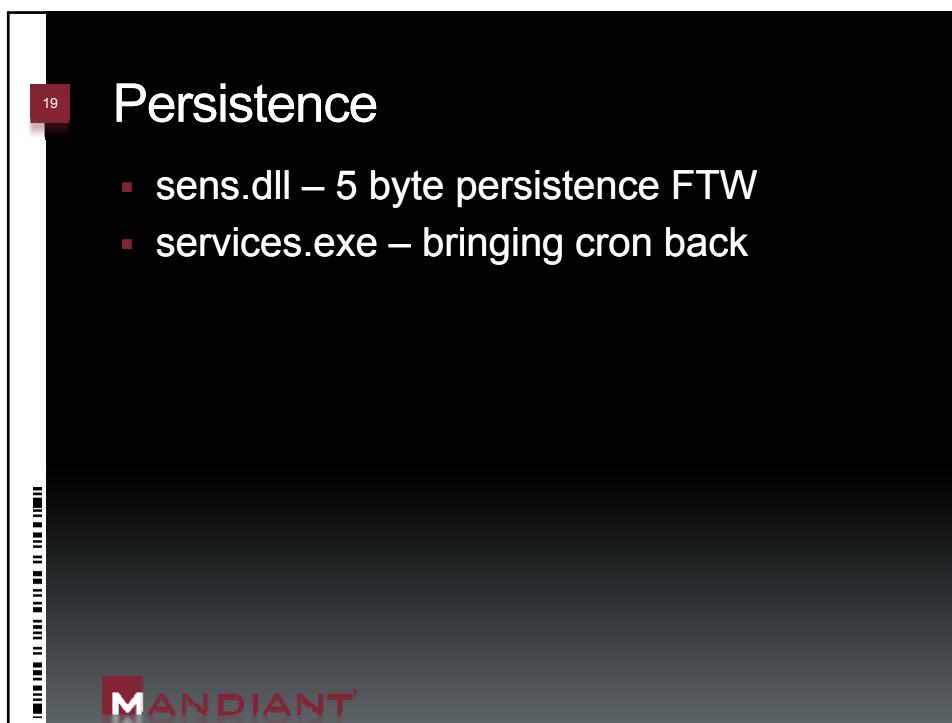
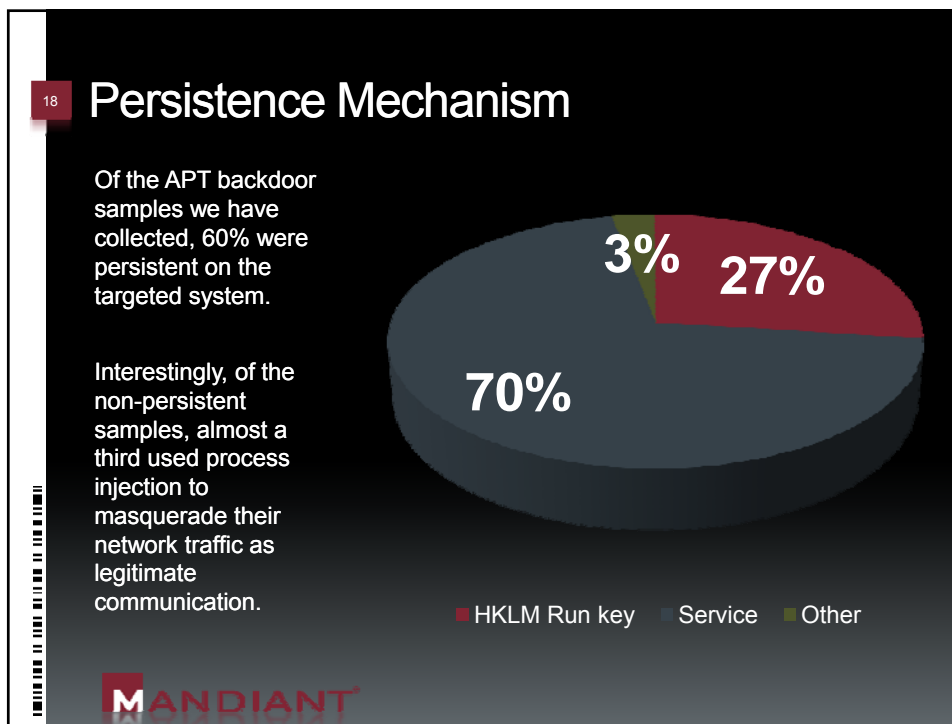
fssrv Properties
-----
General  Version  Compatibility  Summary
File version: 5.0.2195.6664
Description: Windows NT Distributed File System Service

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Owner\Desktop\test\fsrv.exe:
Verified: Unsigned
File date: 11:14 AM 6/15/2010
Publisher: Microsoft Corporation
Description: Windows NT Distributed File System Service
Product: Microsoft® Windows® 2000 Operating System
Version: 5.0.2195.6664
File version: 5.0.2195.6664

C:\Documents and Settings\Owner\Desktop\test>sigcheck.exe -e msvcr71.dll
Sigcheck v1.66 - File version and signature viewer
Copyright (C) 2004-2010 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Documents and Settings\Owner\Desktop\test>msvcr71.dll:
Verified: Unsigned
File date: 5:42 AM 2/21/2003
Publisher: Microsoft Corporation
Description: Microsoft® C Runtime Library
Product: Microsoft® Visual Studio .NET
Version: 7.10.3052.4
File version: 7.10.3052.4
  
```





20

The Legitimate DllMain() Function

- The code in the `DllMain()` function calls two library functions: `DisableThreadLibraryCalls()` and `GetProcessHeap()`

```

722D12B9 ; int __stdcall DllMain(struct HINSTANCE__ *, unsigned long, void *)
722D12B9         mov     edi, edi
722D12BB         push   ebp
722D12BC         mov     ebp, esp
722D12BE         mov     eax, [ebp+fdwReason]
722D12C1         dec     eax
722D12C2         jnz     short loc_722D12D8
722D12C4         push   [ebp+hLibModule]
722D12C7         call   ds:__imp_DisableThreadLibraryCalls@4
722D12CD         call   ds:__imp_GetProcessHeap@0
722D12D3         mov     ?ghSensHeap@@@3PAXA, eax
722D12D8 loc_722D12D8:
722D12D8         xor     eax, eax
722D12DA         inc     eax
722D12DB         pop     ebp
722D12DC         retn   0Ch
722D12DC DllEntryPoint endp

```

21

The Trojanized DllMain() Function

- Now code in the `DllMain()` only `GetProcessHeap()` gets called.
- The Call to `DisableThreadLibraryCalls()` has been replaced by a mysterious `jmp` instruction.

```

722D12B9 ; int __stdcall DllMain(struct HINSTANCE__ *, unsigned long, void *)
722D12B9         mov     edi, edi
722D12BB         push   ebp
722D12BC         mov     ebp, esp
722D12BE         mov     eax, [ebp+fdwReason]
722D12C1         dec     eax
722D12C2         jnz     short loc_722D12D8
722D12C4         push   [ebp+hinstDLL]
722D12C7         jmp    loc_722D822D
722D12C7 ; -----
722D12C7         db 88h
722D12CC ; -----
722D12CD loc_722D12CD:
722D12CD         call   ds:__imp_GetProcessHeap@0
722D12D3         mov     ?ghSensHeap@@@3PAXA, eax
722D12D8 loc_722D12D8:
722D12D8         xor     eax, eax
722D12DA         inc     eax
722D12DB         pop     ebp
722D12DC         retn   0Ch
722D12DC DllEntryPoint endp

```

22

Would you know its bad?

Entry Location	Description	Publisher	Image Path
HKLM\System\CurrentControlSet\Services	VPatch	(Not verified) Internet Security Systems, Inc.	c:\program files\iss\proventia\desktop\vpatch.exe
HKLM\System\CurrentControlSet\Services	MakoNT	(Not verified) Internet Security Systems, Inc.	c:\windows\system32\drivers\makont.sys
HKLM\System\CurrentControlSet\Services	rap	(Not verified) Internet Security Systems, Inc.	c:\windows\system32\drivers\rapdrv.sys
HKLM\System\CurrentControlSet\Services	SENS	(Not verified) Microsoft Corporation	c:\windows\system32\sens.dll
HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved	Directory Service Find	(Verified) Microsoft Windows Publisher	c:\windows\system32\dsquery.dll
HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved	Directory Service Common UI	(Verified) Microsoft Windows Publisher	c:\windows\system32\dsuiext.dll
HKLM\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved	Directory Service Common UI	(Verified) Microsoft Windows Publisher	c:\windows\system32\dsuiext.dll

23

Abusing services.exe

UNMODIFIED

```

push    28h
push    offset stru_100C080
call    __SEH_prolog
xor     edi, edi
push    edi                ; lpModuleName
call    ds:GetModuleHandleA
cmp     word ptr [eax], 5A4Dh
jnz     loc_100C076
mov     ecx, [eax+3Ch]
add     ecx, eax
cmp     dword ptr [ecx], 4550h
jnz     loc_100C076
movzx   eax, word ptr [ecx+18h]
cmp     eax, 10Bh
jnz     loc_100F8DD
cmp     dword ptr [ecx+74h], 0Eh
jbe     loc_100C076
xor     eax, eax
cmp     [ecx+0E8h], edi

; CODE XREF: st
setnz   al
mov     [ebp+var_1C], eax

; CODE XREF: st
mov     [ebp+ms_exc.disabled], edi
push    1
call    ds:_set_app_type

```

MODIFIED

```

push    ebp
mov     ebp, esp
sub     esp, 174h
mov     edi, ebp
add     [edi-3Ch], edx
add     edi, [ebp+var_C]
sub     ecx, esi
or      edi, ecx
imul   ecx, edi
mov     edx, ebp
mov     dword ptr [edx-24h], 0CB07FE30h
ror     esi, 45h
mov     ebx, edx
or      edx, esi
dec     edx
and     esi, 0CD0823B8h
imul   edx, esi
and     esi, edi
mov     ebx, ebp
ror     dword ptr [ebx-0Ch], 7Fh
xor     esi, eax
mov     eax, ebp
sub     [eax-34h], esi
mov     edi, [ebp+var_18]
xor     eax, ebx
mov     eax, ebp
or      [eax-8], ebx
mov     esi, ebp

```

MANDIANT

24

services.exe

- Automatic installer
- services.exe loads malicious DLL
- DLL implements cron like functionality

The Mandiant logo, consisting of a red square with a white 'M' followed by the word 'MANDIANT' in red capital letters.

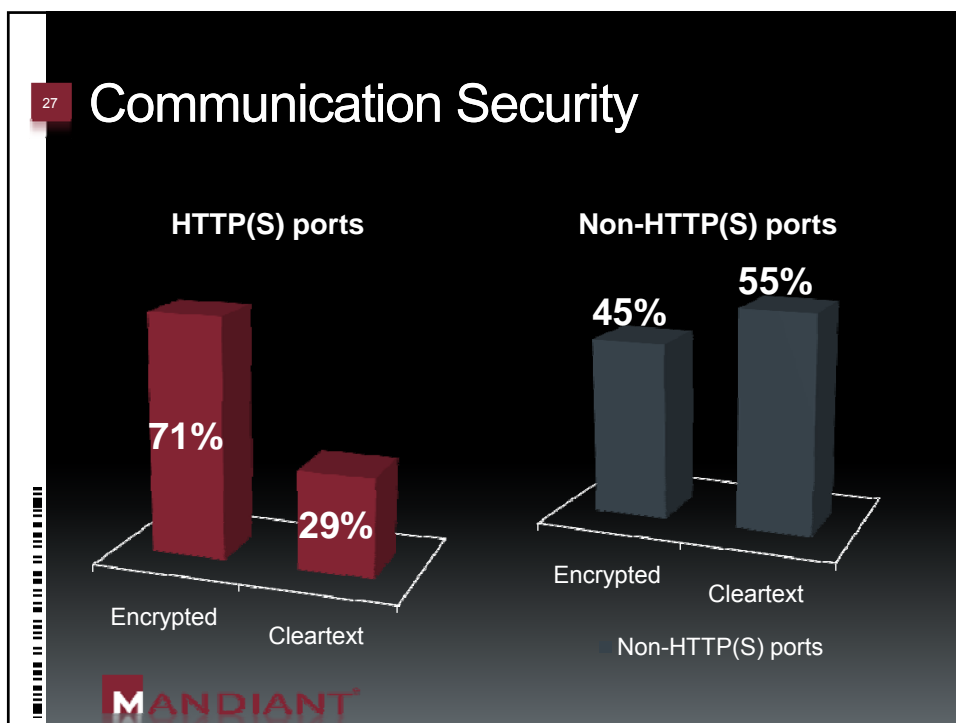
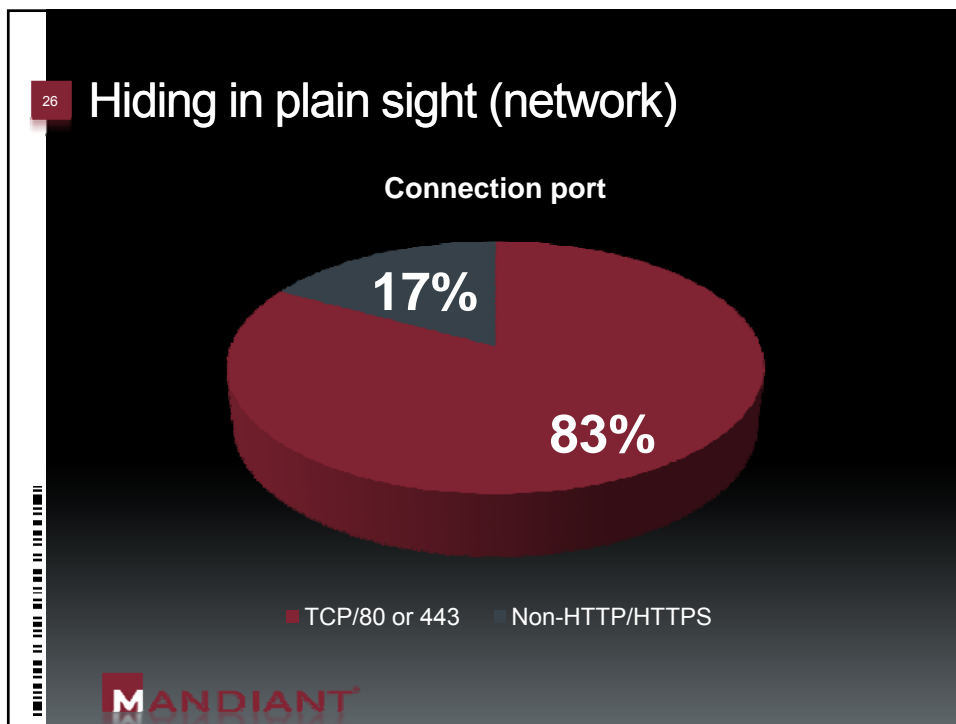
25

Hiding in plain sight (network)

- Used for Command and Control
 - Communicate, control the target
 - Gather information
 - Attackers want this to be covert
- HTTP/S is commonly encrypted
 - But it's not always SSL!
 - Encrypted HTML comments

```
<!--aHR0cAAXXXXXX -->
```

The Mandiant logo, consisting of a red square with a white 'M' followed by the word 'MANDIANT' in red capital letters.



28

Access management

- Attackers track your assets
 - Backdoors
- Need to know:
 - IP/Hostname
 - May know:
 - OS / SP Level
 - MAC
 - RAM
- When one goes away they need to re-up their inventory x 2
 - New malware

The MANDIANT logo is located at the bottom left of the slide. It consists of a red square with a white 'M' inside, followed by the word 'MANDIANT' in a bold, red, sans-serif font.

29

Sample beacon


- GET
`/search(#)####?h1=#&h2=#&h3=#&h4=FMFEFEFHA
EBIBKFOFEAGFGFC`
 - (#) – random number
 - h1 = OS
 - h2 = proxied
 - h3 = malware version
 - h4 = encoded mac address

The MANDIANT logo is located at the bottom left of the slide. It consists of a red square with a white 'M' inside, followed by the word 'MANDIANT' in a bold, red, sans-serif font.

30

USA vs Slovenia

- Score Update



MANDIANT

MANDIANT

The beatings will continue until security improves

HOW TO WIN

MANDIANT

32

Step 1: Redefine Winning

- Goals Are Customized Per Organization, But Can Include:
 - Improve Detection Capability
 - Centralize Logs
 - Acquire Outside Intelligence
 - Improve Response Capability
 - Remove Political Hurdles
 - Iron Processes Out
 - Practice Remediation
 - Raise the Cost of the Theft to Equal Development
 - Staff Management

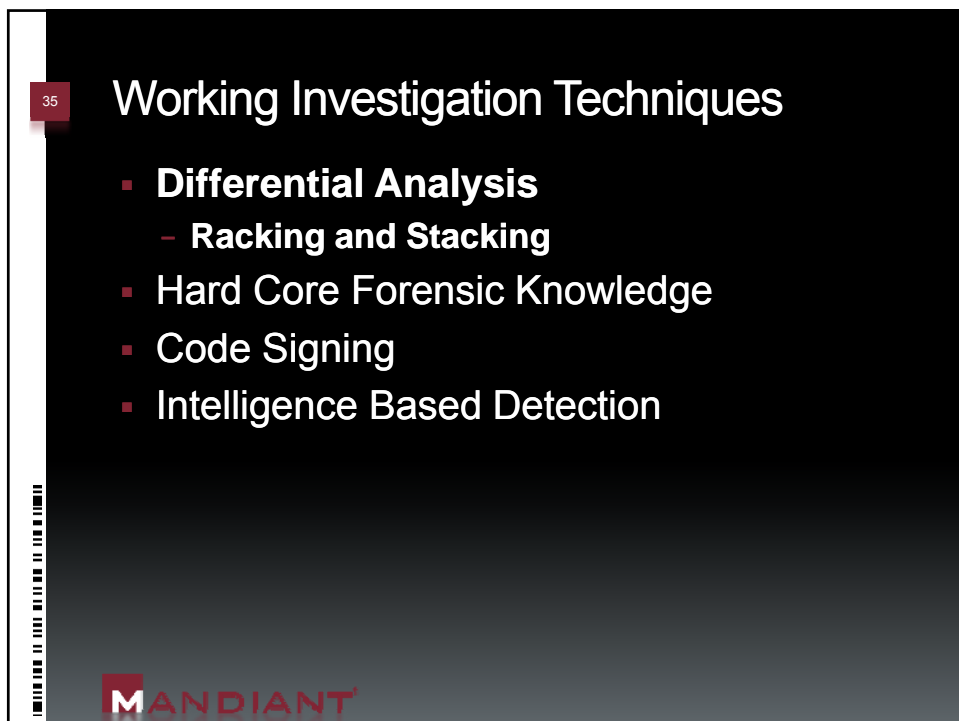
MANDIANT

33

Practical Advice

- Detect and Respond is what is working today.

**MANDIANT**



36

They Dare You to Notice

Service Name	Path	Service DLL
Seclogon	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\System32\seclogon.dll
Seclogon	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\System32\selogon.dll
NWCworkstation	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\System32\nwks.dll
NWCworkstation	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\System32\nwks.dll
iprip	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\System32\iprip.dll
iprip	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\System32\iprip.dll
iprip	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\System32\iprip32.dll
wuauerv	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\system32\wuauerv.dll
wuauerv	C:\WINDOWS\System32\svchost.exe	%SystemRoot%\System32\wuauerv.dll

What's bad?

MANDIANT

38

Working Investigation Techniques

- Differential Analysis
 - Racking and Stacking
- **Hard Core Forensic Knowledge**
- Code Signing
- Intelligence Based Detection

MANDIANT

39

File System Review

	Name	File Created	Last Written	Last Accessed	Logical Size	Hash Value
39553	undrvfm.dll	08/23/01 08:00:CCAM	08/23/01 08:00:CCAM	03/05/09 2:33:41AM	17,312	ac2ac9ac452a0a8d926fb
39554	undrvfm.exe	08/23/01 08:00:CCAM	08/23/01 08:00:CCAM	08/27/09 0:15:17PM	4,096	03691572 9e1401753103
39555	ureg.dll	08/23/01 08:00:CCAM	08/23/01 08:00:CCAM	03/05/09 2:33:45AM	17,920	a1e48073:1daa8:29b434
39556	user.exe	08/23/01 08:00:CCAM	08/23/01 08:00:CCAM	08/20/09 0:15:18PM	47,872	00c376c8416c211417ed
39557	hpojcan.ir	08/23/01 08:00:CCAM	08/23/01 08:00:CCAM	04/07/05 04:04:04AM	5,945	3db106e8:211a0b7a8717
39550	hoscrc.rf	00/23/01 00:00:CCAM	00/23/01 00:00:CCAM	04/07/05 04:04:04AM	20,714	ff4c07061155f51b+2099
39559	ibmscap.rf	08/23/01 08:00:CCAM	08/23/01 08:00:CCAM	04/07/05 04:04:04AM	2,805	f527e4eebds5dde8b684
39500	icam0.rf	00/23/01 00:00:CCAM	00/23/01 00:00:CCAM	04/07/05 04:04:04AM	7,000	f6f4ce7f6de09050e9f1:5
39561	unmrcn.tsp	08/23/01 08:00:CCAM	08/23/01 02:56:56AM	08/24/09 09:42:10AM	206,898	1dfdbecca0fedd14ef3a12
39562	netgcd.dll	08/23/01 08:00:CCAM	04/16/07 11:52:ESAM	08/20/09 02:07:11PM	15,360	a77bf115307f8872ccc5
39563	util01.rll	08/23/01 08:00:CCAM	08/23/01 08:00:CCAM	03/05/09 2:33:51AM	25,900	9d3949e07c 80 27252a1
39564	icam0sup.rf	08/23/01 08:00:CCAM	08/23/01 08:00:CCAM	04/07/05 04:04:04AM	11,341	72320c8c8a8b677ce8e6

40

MFT Parsing

In-Depth Analysis of the Master File Table

Mandiant identified a discrepancy in the timestamps applied to the malware "netgcd.dll". Therefore Mandiant parsed each Master File Table (MFT) record in order to compare all the embedded file time metadata for inconsistencies. Specifically, we compared the \$STANDARD_INFORMATION creation time with the \$FILENAME attribute creation time for a mismatch. Below is an example of this sort of comparison:

Filename #1	\$STANDARD INFORMATION Creation Date	\$FILENAME Creation Date	\$FILENAME Modify Date	\$FILENAME Last Access Date	\$FILENAME Entry Modified Date
netgcd.dll	8/23/2001 12:00	6/18/2008 18:31	6/18/2008 18:31	6/18/2008 18:31	6/18/2008 18:31

Figure 2: MFT Record for "netgcd.dll" Showing Timestamp Manipulation

In Figure 2 the \$STANDARD_INFORMATION Creation Date for "netgcd.dll" differs from the \$FILENAME Creation Date. This difference illustrated that the malware manipulated its timestamps at runtime. The \$FILENAME attribute accurately reflects the "netgcd.dll" file was created on 6/18/2008 at 18:31.

41

Working Investigation Techniques

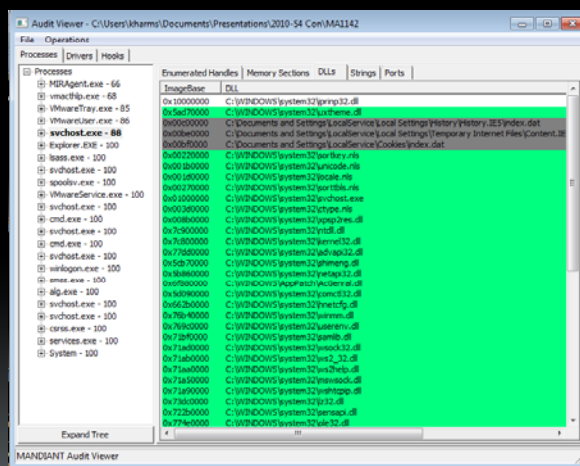
- Differential Analysis
 - Racking and Stacking
- Hard Core Forensic Knowledge
- **Code Signing**
- Intelligence Based Detection



42

Digital Signature Checking

- Audit Viewer and Memoryze with MRI Intelligence




43

Working Investigation Techniques

- Differential Analysis
 - Racking and Stacking
- Hard Core Forensic Knowledge
- Code Signing
- **Intelligence Based Detection**



Generate a Compromise Profile

There is an ongoing APT-related incident. At least 35 systems with APT backdoors have been discovered. One of the backdoors installs itself as a Windows service named "ersvc" with a service DLL of "%systemroot%\system32\ersvr.dll". The file size is 23,040 bytes and the MD5 hash is 906b5626b779eb90b4f403c3b4503b46. In all cases, the modification date of the backdoor file was 2009-03-21 10:06 AM.

The backdoor connects to a remote site via standard HTTP protocol, and downloads a Web page that contains a specially formatted HTML comment. The HTML comment contains instructions for the backdoor, and starts with "<- - #!#obot". The backdoor will use the user-agent string "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0; obot)". The backdoor uses a mutex called ")!VoqA.I4 ". In some cases, the backdoor has been installed laterally using the credentials of a user named "lazydg".

Your boss would really like you to clean up the network.

Identify Content You Can Use to Identify This Attacker in your Network



Generate a Compromise Profile

There is an ongoing APT-related incident. At least 35 systems with APT backdoors have been discovered. One of the backdoors installs itself as a Windows service named "ersvc" with a service DLL of "%systemroot%\system32\ersvr.dll". The file size is 23,040 bytes and the MD5 hash is 906b5626b779eb90b4f403c3b4503b46. In all cases, the modification date of the backdoor file was 2009-03-21 10:06 AM.

The backdoor connects to a remote site via standard HTTP protocol, and downloads a Web page that contains a specially formatted HTML comment. The HTML comment contains instructions for the backdoor, and starts with "<- - #!#obot". The backdoor will use the user-agent string "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0; obot)". The backdoor uses a mutex called ")!VoqA.I4 ". In some cases, the backdoor has been installed laterally using the credentials of a user named "lazydg".

Your boss would really like you to clean up the network.

Cheap to Change = No Coding Necessary

More Costly To Change = Original Author / Source Code Available

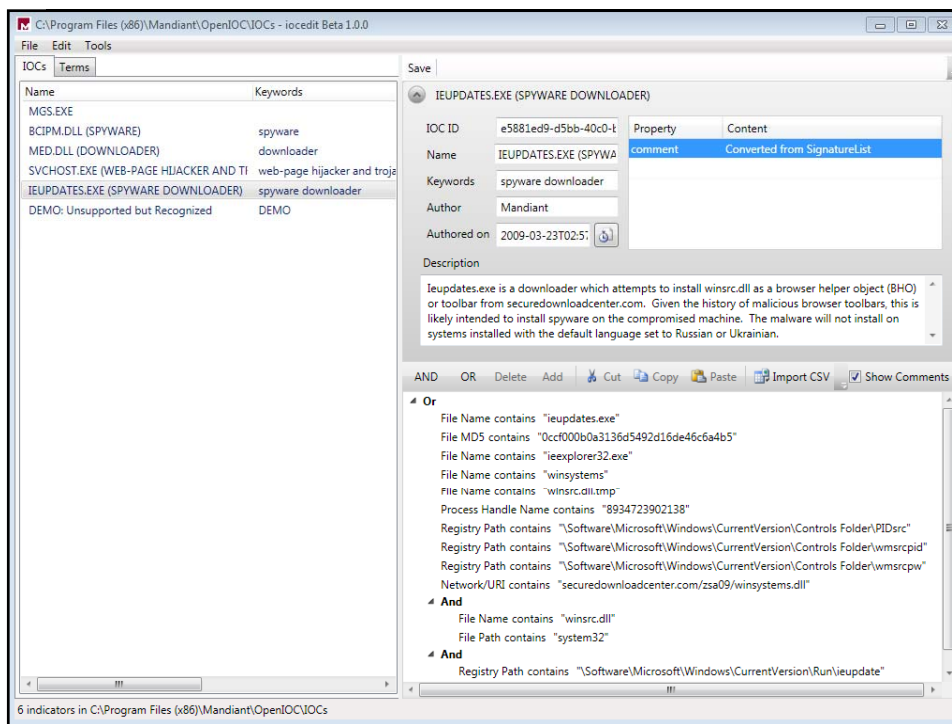
MANDIANT

46

Intelligence Based Detection

- OpenIOC (Open Indicator of Compromise Language)
- Developed by Mandiant in conjunction with Industry
- Designed to Facilitate Sharing of Actionable Intelligence
- Free OpenIOC Editor on the Mandiant Website to Create and Manage Indicators

MANDIANT



48

Truths To Date:

- No Organization Has:
 - Been Prepared to Defend Their Network Against A Nation State Sponsored Attacking Capability
- There is no industry or government solution to protect our commercial companies right now

MANDIANT

49

USA vs Slovenia

- Score Update



MANDIANT

50

Questions?

RESOURCES

- M-Trends – MANDIANT website
- M-Union Blog (blog.mandiant.com)
- Mandiant is Hiring! Help us Out!
Recruiting@mandiant.com
- Web Historian 2.0 Release Yesterday at FIRST

MANDIANT