USA vs Slovenia
- Score Update
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 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
Introductions

- Kris Harms
  - IR Engagement Lead, Instructor

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

Important note

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

Some information has been sanitized to protect our clients’ interests.
Agenda

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

Why Defenders Lose

VS

Your Company Logo Here
Why Defenders Lose

COMPLIANCE ≠ SECURITY

Grades
ABCD/F
Why Defenders Lose

GENERAL WINDOWS KNOWLEDGE VS YOUR NETWORK CONFIGURATION

Choose Your Theater

Porn Sites
(sorry no pictures)

facebook

Just say no

DEAD END
YOU'VE REACHED THE END OF THE INTERNET
Well... It Depends

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

Hiding in plain sight
Persistence Mechanism

Of the APT backdoor samples we have collected, 60% were persistent on the targeted system.

Interestingly, of the non-persistent samples, almost a third used process injection to masquerade their network traffic as legitimate communication.

Persistence

- sens.dll – 5 byte persistence FTW
- services.exe – bringing cron back
The Legitimate DllMain() Function

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

```assembly
722D12B9 int __stdcall DllMain(struct HINSTANCE__, unsigned long, void *)
722D12B9 mov edi, edi
722D12B9 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
```

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.

```assembly
722D12B9 int __stdcall DllMain(struct HINSTANCE__, unsigned long, void *)
722D12B9 mov edi, edi
722D12B9 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
722D12CC db 88h
722D12CD loc_722D12CD:
722D12CD call ds:__imp__GetProcessHeap@@0
722D12D3 mov ?ghSensHeap@@3PAXA, eax
722D12D8 loc_722D12D8:
722D12DA inc eax
722D12DB pop ebp
722D12DC retn 0Ch
722D12DC DllEntryPoint endp
```
Would you know it's bad?

<table>
<thead>
<tr>
<th>Entry Location</th>
<th>Description</th>
<th>Publisher</th>
<th>Image Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKLM\System\CurrentControlSet\Services</td>
<td>VPatch</td>
<td>Security Systems, Inc.</td>
<td>c:\program files\iss\proventia\Desktop\vpatch.exe</td>
</tr>
<tr>
<td>HKLM\System\CurrentControlSet\Services</td>
<td>MakoNT</td>
<td>Security Systems, Inc.</td>
<td>c:\windows\system32\drivers\makont.sys</td>
</tr>
<tr>
<td>HKLM\System\CurrentControlSet\Services</td>
<td>rap</td>
<td>Security Systems, Inc.</td>
<td>c:\windows\system32\drivers\rapdrv.sys</td>
</tr>
<tr>
<td>HKLM\System\CurrentControlSet\Services</td>
<td>SENS</td>
<td>Microsoft Corporation</td>
<td>c:\windows\system32\drivers\makont.sys</td>
</tr>
</tbody>
</table>

Abusing services.exe

UNMODIFIED

```assembly
push 0
push offset stru_100C000
call _SSS_prolong
mov edi, edi
push edi
mov ebp, [ebp+4]
```

MOODIFIED

```assembly
push ebp
mov ebp, esp
sub esp, 12h
mov edi, ebp
add [esp-10h], edi
add edi, [ebp+var_10]
```

```
rop push [ecx+7Ah], esp
```

```
```

services.exe

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

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
    
<!aHR0cAXXXXXX>
Hiding in plain sight (network)

Connection port

- TCP/80 or 443
- Non-HTTP/HTTPS

Communication Security

HTTP(S) ports
- Encrypted: 71%
- Cleartext: 29%

Non-HTTP(S) ports
- Encrypted: 45%
- Cleartext: 55%
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

Sample beacon

- GET
  /search(#)##?h1=#&h2=&h3=#&h4=FMFEFEFHAEBIBKFOFEAGFGFC
  - (#) – random number
  - h1 = OS
  - h2 = proxied
  - h3 = malware version
  - h4 = encoded mac address
USA vs Slovenia

- Score Update

HOW TO WIN

The beatings will continue until security improves
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

Practical Advice

- Detect and Respond is what is working today.
Game Changers of Today

INVESTIGATION TECHNIQUES

Working Investigation Techniques

- Differential Analysis
  - Racking and Stacking
- Hard Core Forensic Knowledge
- Code Signing
- Intelligence Based Detection
They Dare You to Notice

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Path</th>
<th>Service DLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seclogon</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\seclogon.dll</td>
<td></td>
</tr>
<tr>
<td>Seclogon</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\seclogon.dll</td>
<td></td>
</tr>
<tr>
<td>NWWorkstation</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\nwks.dll</td>
<td></td>
</tr>
<tr>
<td>NWWorkstation</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\nwks.dll</td>
<td></td>
</tr>
<tr>
<td>iprip</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\iprip.dll</td>
<td></td>
</tr>
<tr>
<td>iprip</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\iprip.dll</td>
<td></td>
</tr>
<tr>
<td>iprip</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\iprip32.dll</td>
<td></td>
</tr>
<tr>
<td>wuauserv</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\wuauserv.dll</td>
<td></td>
</tr>
<tr>
<td>wuauserv</td>
<td>C:\WINDOWS\System32\svchost.exe %SystemRoot%\System32\wuauserv.dll</td>
<td></td>
</tr>
</tbody>
</table>

What’s bad?

Working Investigation Techniques

- Differential Analysis
  - Racking and Stacking
- Hard Core Forensic Knowledge
- Code Signing
- Intelligence Based Detection
File System Review

MFT Parsing

In-Depth Analysis of the Master File Table

Mandiant identified a discrepancy in the timestamps applied to the malware "netgdc.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:

<table>
<thead>
<tr>
<th>Filename #1</th>
<th>$STANDARD_INFORMATION Creation Date</th>
<th>$FILENAME Creation Date</th>
<th>$FILENAME Last Access Date</th>
<th>$FILENAME Last Write Date</th>
<th>$FILENAME Entry Modified Date</th>
</tr>
</thead>
</table>

Figure 1: MFT Record for "netgdc.dll" Showing Timestamp Manipulation

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

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

Digital Signature Checking

- Audit Viewer and Memoryze with MRI Intelligence
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 "<!-- # ! ##!#obotobot". The backdoor will use the user-agent string "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0; obotobot)". 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 906b5626b779eb90b4f403c3b403b46. 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 “<!--”. The backdoor will use the user-agent string “Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0; bob)”. The backdoor uses a mutex called “lazydog”.

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

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
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
USA vs Slovenia

- Score Update

Questions?

RESOURCES

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