MALWARE ANALYSIS REPORT

Googost.A – The Hello Proxy

Alcatel·Lucent

TABLE OF CONTENTS

SUMMARY

Googost is a bot infection that provides the attacker with a Transmission Control Protocol (TCP) proxy on the infected host. The amount of network traffic generated by a single infected computer can be enormous. We observed a single infected computer making over 800,000 web connections from a mobile network in a 24-hour period, consuming over 3GB of bandwidth.

The infected Googost computer contacts a control and command (C&C) server in Europe and establishes a TCP connection. When the C&C server has work, it sends the domain name and an HTTP GET request indicating what web page should be retrieved. The infected computer contacts the indicated web server, issues the HTTP GET request and retrieves the results. These results are sent immediately to the C&C server using the pre-established TCP session. Subsequently, the C&C server closes the TCP connection to the infected machine to indicate that the request is complete. Then the infected computer reopens the TCP connection and repeats the process.

The proxy can be used for a variety of purposes:

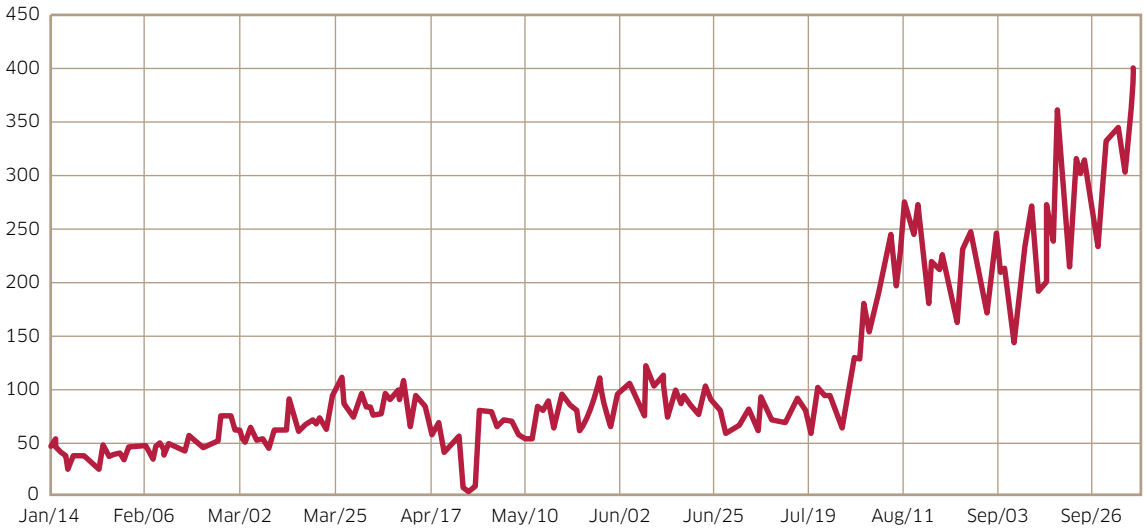
- Anonymous browsing services
- Access to restricted foreign content
- Web site optimization fraud
- Ad-click fraud
- Internal network probe and data exfiltration (Enterprise advanced persistent threat [APT])

The traffic observed from our lab tests leaned primarily toward web site optimization or ad-click fraud, unless the attacker felt access to Canadian gourmet cooking sites required anonymization.

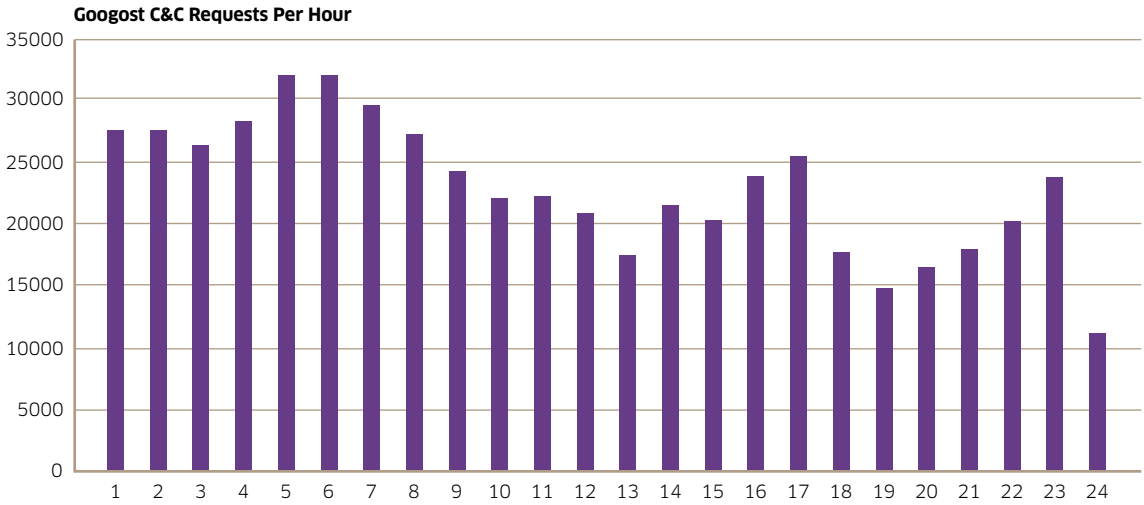
We use the name “Googost” from the Snort® detection signature that detects the C&C traffic. The signature originally came from Telus Security Labs. Most antivirus vendors will report a different name (for example, Alureon or Kazy) that relates more to how the malware is packaged than by what it does. A better name would be “Hello Proxy,” which is derived from what the malware actually does.

BACKGROUND

In September 2013 we noticed the infection rate for Googost.A had significantly increased.



We took a closer look at the network traffic associated with this infection and discovered that it was huge. In one instance, a single infected computer was polling a C&C web site over 400 times per minute. An hourly C&C activity graph for this user is shown below. Each event is an HTTP connection to the Googost. A command and control server involving a 10-packet exchange and about 500 bytes of data. Thus, this single malware infection instance consumed about 10MB per hour of bandwidth 24/7 for just C&C traffic.



In September 2013 about 0.4 percent of homes in our field deployments (1 in 250) were infected with Googost. Not all were as active as the case described above, but many were so we decided to have a closer look at what it was doing.

MALWARE SAMPLE

We have a few hundred malware samples in our library that trigger the Googost C&C detection signature so we chose the most recent one for an in-depth analysis.

Name: Win32.Trojan.Googost.A
MD5: d9521421bfb58374e53f2045b31749a5
Size: 238080 bytes
File type: PE32 executable for MS Windows (GUI) Intel 80386 32-bit
Sample collected: 2013-10-19 20:45:10

VirusTotal provided a mixed bag of detection names for this sample and for others that also triggered the Googost detection rule. This is likely because the infection mechanism uses elements commonly associated with Alureon, Kazy and many other infections.

Data provided by VirusTotal ® on 2013-11-05.

AhnLab-V3	Trojan/Win32.PMax	**Avast**	Win32:Rootkit-gen [Rtk]
Comodo	UnclassifiedMalware	**DrWeb**	Trojan.Packed.22496
K7AntiVirus	Trojan	**Panda**	Generic Malware
Symantec	Trojan.Gen.2	**VIPRE**	Trojan.Win32.Generic!SB.0
Ikarus	Backdoor.Win32.PMax	**Sophos**	Mal/Generic-S
VBA32	Backdoor.PMax	**K7GW**	Trojan
Microsoft	Trojan:Win32/Alureon.GQ	**Antiy-AVL**	Backdoor/Win32.PMax
ESET-NOD32	a variant of Win32/Kryptik.BMNY	**McAfee**	RDN/Generic BackDoor!tz
McAfee-GW-Edition	RDN/Generic BackDoor!tz	**AntiVir**	TR/Alureon.GQ.219
Emsisoft	Gen:Variant.Kazy.267830 (B)	**Kaspersky**	Backdoor.Win32.PMax.asok
F-Secure	Gen:Variant.Kazy.267830	**GData**	Gen:Variant.Kazy.267830
Kingsoft	Win32.Hack.PMax.as.(kcloud)	**Bkav**	W32.Clodb7c.Trojan.16f6
Baidu-International	Trojan.Win32.Kryptik.BMNY	**TrendMicro-HouseCall**	TROJ_SPNV.05JR13
TrendMicro	TROJ_SPNV.05JR13	**Fortinet**	W32/PMax.ASOK!tr.bdr
Jiangmin	Backdoor/PMax.dlv	**NANO-Antivirus**	Trojan.Win32.PMax.cixgst
AVG	BackDoor.Generic17.BVDQ	**Norman**	Troj_Generic.QNBFF
Agnitum	Backdoor.PMax!qjkyvDarq6s	**BitDefender**	Gen:Variant.Kazy.267830

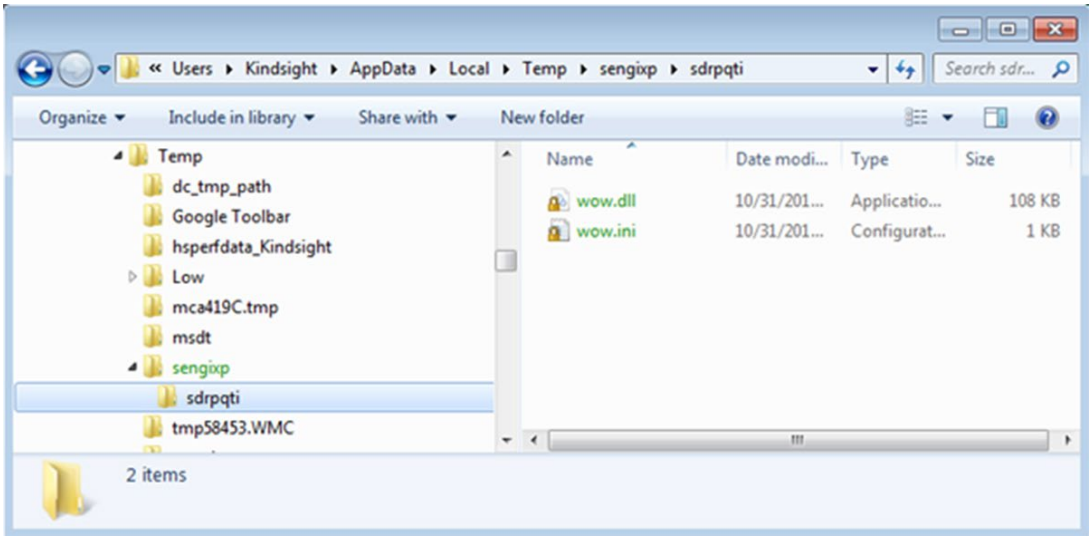
Update

INFECTION

Based on the various detection names provided by VirusTotal, this malware uses a common toolkit to infect the host, install the main malware payload, gain a permanent foothold, and then protect itself from discovery. It uses elements commonly found with Alureon, Kazy and other malware families. The audio-visual products on VirusTotal are detecting these elements rather than the actual Googost payload, which is packed and encrypted within them.

The sample was executed on a Windows 7 virtual machine and resulted in the following activity:

1. The executable created a hidden system directory as shown below. The files *wow.dll* and *wow.ini* are written there. The location chosen to drop the files will vary from one malware sample to another.



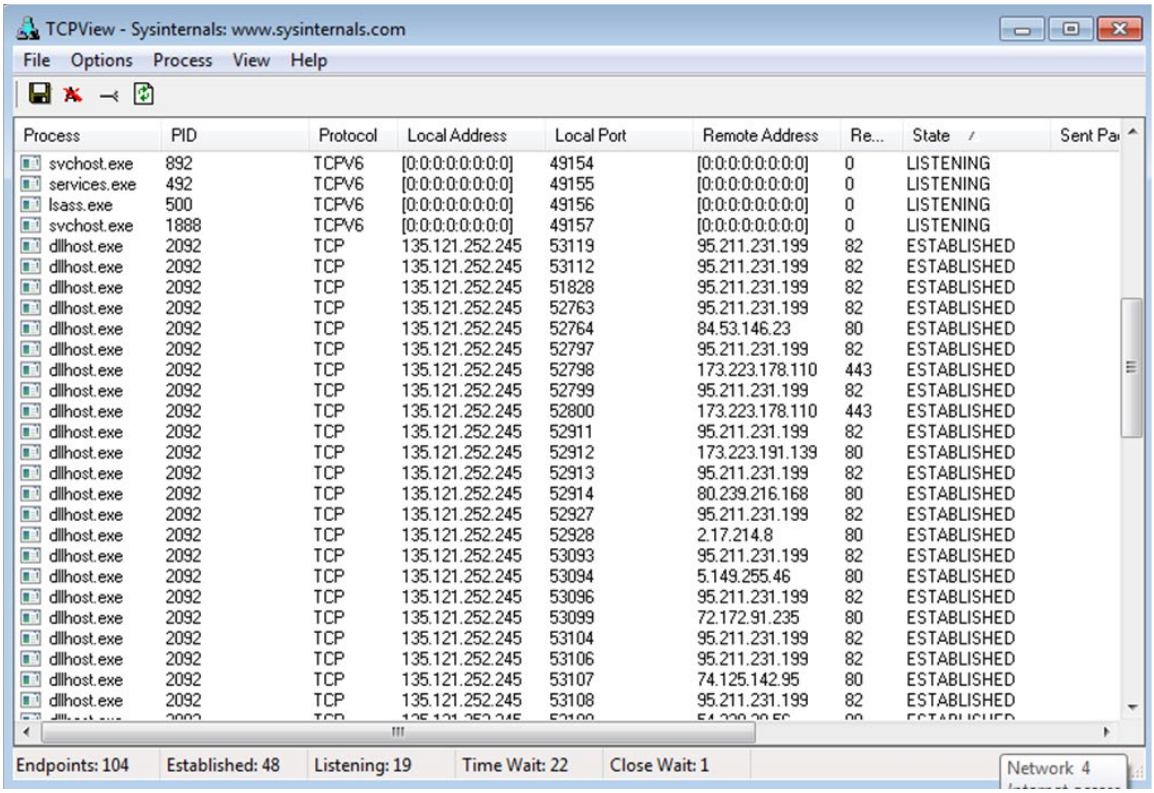
   The *dll* file contains the malware payload. The *ini* file contains the configuration for the malware. These location drops will vary depending on the sample.

2. A user specific "inprocserver32" registry key is associated with the CLSID "fbeb8a05-beee-4442-804e-409d6c4515e9." This association will cause the system to load *wow.dll* instead of *shell32.dll* during system initialization. It is the manner in which the malware gets restarted when the system boots up.

```
HKEY_CURRENT_USER\software\classes\clsid\\{fbeb8a05-beee-4442-804e-409d6c4515e9}\inprocserver32\
Set too ===>>>
C:\Users\Kindsight\AppData\Local\Temp\sengixp\sdrpqti\wow.dll
```

   Some sort of rootkit must be in use to prevent security tools from detecting this change because it is not visible when regedit is used to inspect the registry.

3. When *wow.dll* is loaded it unpacks the Googost payload and runs it under the Windows DLLHOST COM surrogate process. All subsequent network activity can be traced to this process.



4. The malware then deletes the original executable and begins operating as a proxy.

PROTECTION

The malware takes several precautions to protect itself. It looks like a watchdog process is monitoring the malware operation and providing some cloaking. As mentioned previously, the registry changes that start the malware at bootup time are not visible through regedit, so it looks like it hooked that API. When the DLLHOST process is killed, it automatically restarts in a few seconds. When we attempted to attach a debugger to the DLLHOST process, it terminated immediately and we were unable to trace what the code was doing. In addition, large sections of memory appeared to be overwritten with garbage.

OPERATION

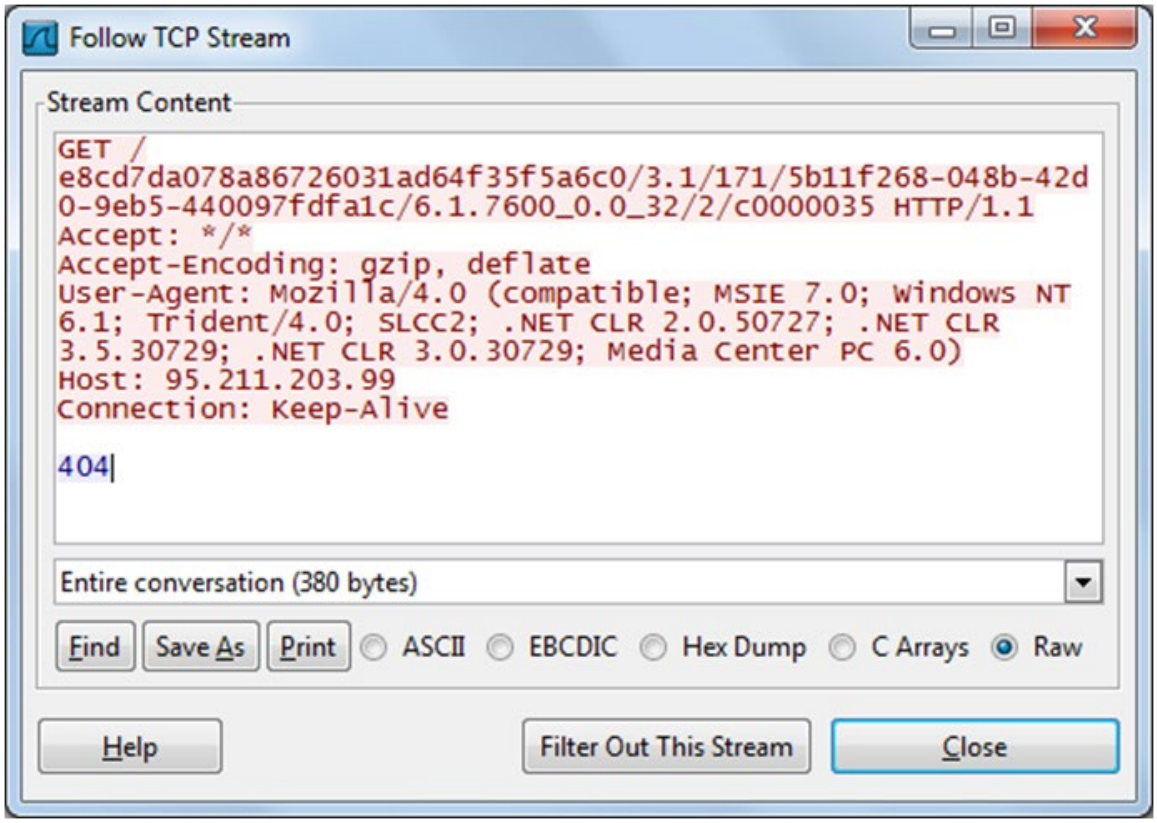
The operation begins when the wow.dll is loaded during system initialization. This download runs the main malware process that provides the proxy services under the DLLHOST COM surrogate. The operation of the malware is controlled by the *wow.ini* configuration file.

```
[main]
version=3.1
aid=171
servers=newagelimp.com:80;newfogfrom.com:80;95.211.203.99:80;
knock=95.211.203.99
```

File: wow.ini

The infected host first makes a TCP connection to google.com to verify Internet connectivity, which is probably the source of the original name (Googost).

The host then contacts the IP address defined by the "knock" parameter from the *wow.ini* file on Port 80 and issues an HTTP GET request.



This step appears to be a preliminary check-in occurring after the initial infection, but does not seem to reoccur later on. The parameters on the GET request identify the infected host and the malware version and are discussed later in greater detail. The data that is returned (404) is just raw text and not an actual HTTP error code.

The malware then issues a domain name system (DNS) request for the servers named in the configuration file. These names vary from one malware sample to another. If the malware is unable to resolve a name, it will likely be inoperable. In the case above, the "knock" – the IP address – is also provided in the server's list, presumably as a backup. In any event, the *wow.ini* file contains the information required to sinkhole a particular instance of the bot.

The port numbers for the servers can also vary. We have seen 80, 81, 8080, 8000, 443 used in various samples.

The bot contacts one of the servers on the list and issues its "hello" command.

```
hello/3.1/171/5b11f268-048b-42d0-9eb5-440097fdfa1c/6.1.7600_0.0_32/1/00000000
```

This contains the following fields separated by slashes:

- The text “hello”
- Version (from *wow.ini* file)
- Aid (from *wow.ini* file)
- Infected host CLSID identifier (from HKLM\SOFTWARE\Microsoft\Cryptography)
- Operating system identifier
- 1
- 00000000

The last two fields were always observed as 1 and 00000000, respectively. The remote server was observed to respond in one of two ways – either it redirected the infected host to another server or the server asked to use the proxy service. This redirection indicates that in addition to the botnet of infected proxies, there may also be a botnet of computers that use the proxies. However, it is not known if these user machines are infected with malware or owned by the bot operators.

REDIRECTION CASE

The packets below show the redirection operation. The infected host sends the “hello” packet and the server responds with 0xfe054f41cc0051. Here, the first byte (0xfe) indicates that it is a redirection command. The next four bytes are the IP address of the server that it is being redirected to and the last two bytes are the port number. In the example above, the redirection is to Port 81 on 5.79.65.204. The infected host will then connect to the new server and send it a “hello” command. This process may be repeated for two or three iterations.

```
00000000  68 65 6c 6c 6f 2f 33 2e  31 2f 31 37 31 2f 35 62 hello/3. 1/171/5b
00000010  31 31 66 32 36 38 2d 30  34 38 62 2d 34 32 64 30 11f268-0 48b-42d0
00000020  2d 39 65 62 35 2d 34 34  30 30 39 37 66 64 66 61 -9eb5-44 0097fdfa
00000030  31 63 2f 36 2e 31 2e 37  36 30 30 5f 30 2e 30 5f 1c/6.1.7 600_0.0_
00000040  33 32 2f 31 2f 30 30 30  30 30 30 30 30 30      32/1/000 00000
    00000000  fe 05 4f 41 cc 00 51                          ..OA..Q
0000004D  ff                                               .
```

PROXY REQUEST CASE

After a number of redirections the remote server will eventually respond with a proxy request. This is shown below.

```
00000000  68 65 6c 6c 6f 2f 33 2e  31 2f 31 37 31 2f 35 62 hello/3. 1/171/5b
00000010  31 31 66 32 36 38 2d 30  34 38 62 2d 34 32 64 30 11f268-0 48b-42d0
00000020  2d 39 65 62 35 2d 34 34  30 30 39 37 66 64 66 61 -9eb5-44 0097fdfa
00000030  31 63 2f 36 2e 31 2e 37  36 30 30 5f 30 2e 30 5f 1c/6.1.7 600_0.0_
00000040  33 32 2f 31 2f 30 30 30  30 30 30 30 30 30      32/1/000 00000
    00000000  05 01 00 03 0e 38 38 2e  32 31 34 2e 31 39 33 2e .....88. 214.193.
    00000010  32 31 32 00 50                               212.P
0000004D  05 00 00 01 00 00 00 00  00 00                   ........ ..
```

The server responds to the “hello” request with the 0x05 command indicating that this is a proxy request. This consists of 0x05010003 followed by a one byte length (in this case, 0x0e – 14), a character string containing the domain name or IP address of the target for the proxy, and a two-byte port number (in this case, 0x0050 – 80). The infected host responds with 0x5000000100000000, which appears to be an acknowledgement that it will execute the proxy request. The server then provides the data that the proxy should send to the target. In most cases, this was an HTTP GET request, but in some instances it was HTTPS traffic. The proxy does not process the data in any way; it simply moves the data provided by the endpoint and is capable of handling any type of TCP session. The infected computer opens a connection to the target on the specified port and sends the data that was provided by the requesting server. Any response from the target is sent back to the requesting server on the TCP session established with the “hello” command.

A complete example is shown below. The requesting server may ask for multiple requests to the same target. After it has completed all proxy transactions with the target, the requesting server closes the TCP connection with the infected proxy host. At that point, the infected host will establish a new TCP connection with the requesting server and send a new "hello" command to begin the process again. This sequence is then repeated as long as the DLLHOST process is running.

```
hello/3.1/171/5b11f268-048b-42d0-9eb5-440097fdfa1c/6.1.7600_0.0_32/1/00000000....
68.67.185.221.P.........GET /tt?id=1732325 HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Referer: http://quotefountain.com/Modules/Oceaneddy.OpenX/Scripts/iframe.html
Accept-Language: en-US
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/6.0)
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
Host: advs.adgorithms.com
Connection: Keep-Alive

HTTP/1.1 302 Found
Cache-Control: no-store, no-cache, private
Pragma: no-cache
Expires: Sat, 15 Nov 2008 16:00:00 GMT
P3P: policyref="http://cdn.adnxs.com/w3c/policy/p3p.xml", CP="NOI DSP COR ADM
PSAo PSDo OURo SAMo UNRo OTRo BUS COM NAV DEM STA PRE"
X-XSS-Protection: 0
Location: http://ib.adnxs.com/tt?id=1732325
Date: Thu, 31 Oct 2013 19:06:01 GMT
Content-Length: 0
Content-Type: text/html; charset=ISO-8859-1
```

HTTPS

In many cases, HTTPS sessions were observed in which the infected host was operating as a pure TCP proxy moving bytes on the wire. The end-to-end secure sockets layer (SSL) communications is between the requesting server and the target web server. The proxy is capable of handling any TCP connection.

MULTI-THREADING

The infected process does not limit itself to one connection at a time with the requesting server, but was observed to open 20 or 30 simultaneous connections. These connections can then be used to proxy multiple simultaneous connections.

USES

There are a number of uses for this type of proxy service.

ANONYMOUS WEB BROWSING SERVICE

The most obvious use of this malware is to provide an anonymous web browsing service. People will pay for this type of service to conceal their browsing activities for a variety of reasons, particularly if the browsing activity is criminal in nature. Users of the service would look like their web browsing is originating from the infected computers that are running the proxy service. Consequently, a person in Europe or China would look like they are browsing the Web from Canada or the United States.

PROVIDING ACCESS TO RESTRICTED FOREIGN CONTENT

Content such as movies is frequently restricted to distribution within a specific geographic region. For example, Canadians do not have access to the U.S. version of Netflix. This proxy could be used to provide illegal access to this type of restricted foreign content by making the user "look" like they are within the correct geographic zone.

AD-CLICK FRAUD

Ad-click fraud detection mechanisms often use an IP address geo-location to verify that the ad-clicks are coming from a reasonable location. For example, it would be very suspicious if several Russian IP addresses were clicking on Canadian advertisements. This type of proxy could be used to ensure that the fake ad-click locations are realistic.

WEB SITE OPTIMIZATION FRAUD

Web site optimization companies execute campaigns to increase the number of visitors to their customers' web sites and often target specific demographics and locations. It is feasible they could enlist the services of this proxy service to generate web visits that look like they are coming from the target locations.

APT PROBING AND EXFILTRATION

This type of proxy could be a key component in an APT scenario. If a device inside a corporate network is infected, the attacker can use the proxy to connect to computers and services inside the corporate network and exfiltrate the data.

SO WHAT WAS IT ACTUALLY USED FOR?

The most common use of this type of proxy is to allow anonymous web browsing that cannot be traced back to the origin. However, in the traffic we observed there were no sites involved that would require anonymity. Also, there were no sites that were hosting regionally restricted content. The network traffic that we observed from our test laboratory located in Canada was to a fairly ordinary mix of Canadian and U.S. web sites with a reasonable amount of advertising traffic included. For example, in one session we saw a lot of traffic to gourmet cooking sites. Because it is unlikely that these types of sites would require anonymity, the most likely explanation is that this is some sort of web site optimization or ad-click fraud scheme. We would have to look at much more traffic to determine exactly how it works.

NETWORK IMPACT

When the requesting server is active, the network impact of the proxy server can be quite significant, but there were also times when activity was low for extended periods. More extensive testing is required to measure the long-term network impact of this infection, but the following cases were observed.

FIELD RESULTS FROM A DETECTION SIGNATURE

The incident that initially triggered our interest in this malware was the very high rate of detection events from the field. The detection signature will trigger on the "hello" message for each proxy session. We saw a single infected computer triggering this message over 400 times per minute for 36 hours. Each of these represented a minimum of 10 packets and about 500 bytes of data. Thus, this single malware infection is consuming about 10MB per hour for just the C&C traffic. Actual proxy traffic will increase the MB significantly.

LAB RESULTS

As illustrated in the following table, the lab results varied considerably. For example, the second entry shows that the impact can be significant even in short bursts.

TEST DURATION	TOTAL BANDWIDTH	MB/HOUR	SESSIONS/HOUR
1 h	20M	20	969
5 min	229M	2748	30,504
15 min	2.5M	10	704
2 h 20 min	116M	50	746

FIELD RESULT WITH BANDWIDTH CONSUMPTION MEASUREMENT

In one mobile field trial, we noticed an infected user was active for a 24-hour period with over 500,000 detection events. In this case, we were able to monitor the total bandwidth consumption. This user consumed over 3GB of bandwidth, mostly from web browsing, and had more than 800,000 TCP connections during that period. Almost all of this activity was due to the malware.

CONCLUSION

In September 2013 about 0.4 percent of homes in our malware detection deployments (1 in 250) were infected with the “Hello Proxy” (Googost).

Many of these infected users were consuming large amounts of bandwidth acting as TCP proxies for web browsing activities originating from servers in Europe, mostly in Germany and The Netherlands as illustrated below. These infected proxies are likely being used as part of a web site optimization or ad-click fraud scheme, but could also be used for anonymous browsing and access to restricted content.

Kindsight™

www.alcatel-lucent.com/solutions/kindsight-security

Alcatel·Lucent