# RATicate: an attacker’s waves of information-stealing malware

news.sophos.com/en-us/2020/05/14/raticate/

Markel Picado

May 14, 2020

In a series of malspam campaigns dating back to November of 2019, an unidentified group sent out waves of installers that drop remote administration tool (RAT) and information stealing malware on victims’ computers.

We’ve identified five separate campaigns between November, 2019 and January, 2020 in which the payloads used similar packing code and pointed to the same command and control (C&C) infrastructure. The campaigns targeted industrial companies in Europe, the Middle East, and the Republic of Korea. This leads us to believe that they are all the work of the same actors—a group we’ve dubbed **RATicate**.

A new campaign we believe connected to the same actors leverages concern about the global COVID-19 pandemic to convince victims to open the payloads. This is a shift in tactics, but we suspect that this group constantly changes the way they deploy malware—and that the group has conducted campaigns prior to this past November.

In this post, we’ll focus on the initial wave of campaigns, which all used Nullsoft Scriptable Install System (NSIS) installers. NSIS is an open source tool for creating Windows installers, designed for Internet-based software distribution. But it has also been abused for a long time to disguise and deploy malware. (We’ll discuss newer campaigns using other installers, and the group’s shift in phishing tactics, in an upcoming follow-up report.)

## Plugged in for malware

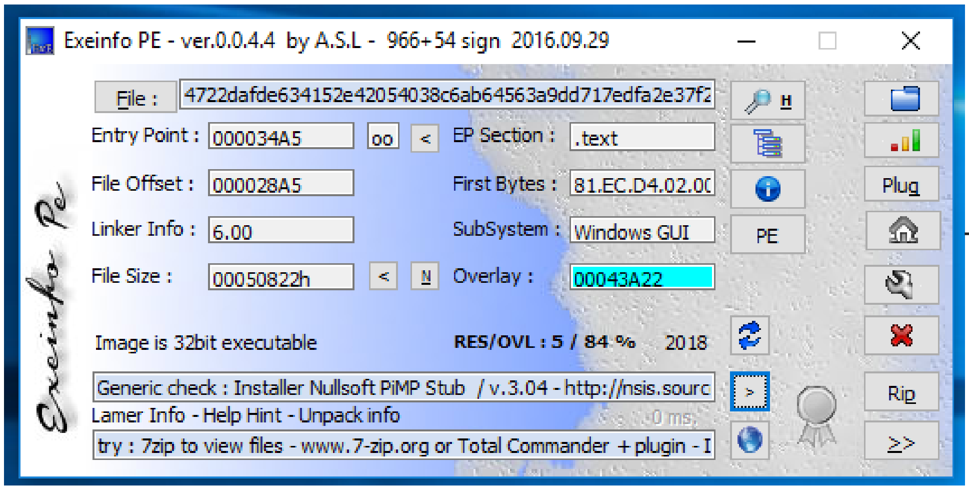
One of the interesting features of NSIS installers is their plug-in architecture, which allow installers to communicate with other software components—including components of the Windows operating system. (A list of available plug-ins can be found here.) These plug-ins are deployed as Windows DLL files. If selected during the installer build, they will be automatically added to the final compiled NSIS installer's packaged files inside the "$PLUGINS" folder.

Some of the capabilities these plugins can provide include:

The installers we looked at caught our attention because they all drop the same set of "junk files" (files that are never used by the installed malware) across the initial sample set. We've seen the tactic of packing NSIS installers with garbage files to conceal malware in the past; the junk files are intended to confuse analysts and create "noise" during sandbox analysis. So this behavior caught our attention, and we started to analyze it in more detail.

We found that all the samples use the **System.dll** plugin, which allows you to load a DLL and call its exported functions. The DLL called by these malicious installers injects a payload into memory (in most cases by using *cmd.exe*).

For purposes of illustration, this report focuses primarily on the analysis of one sample NSIS installer from the first group we discovered:

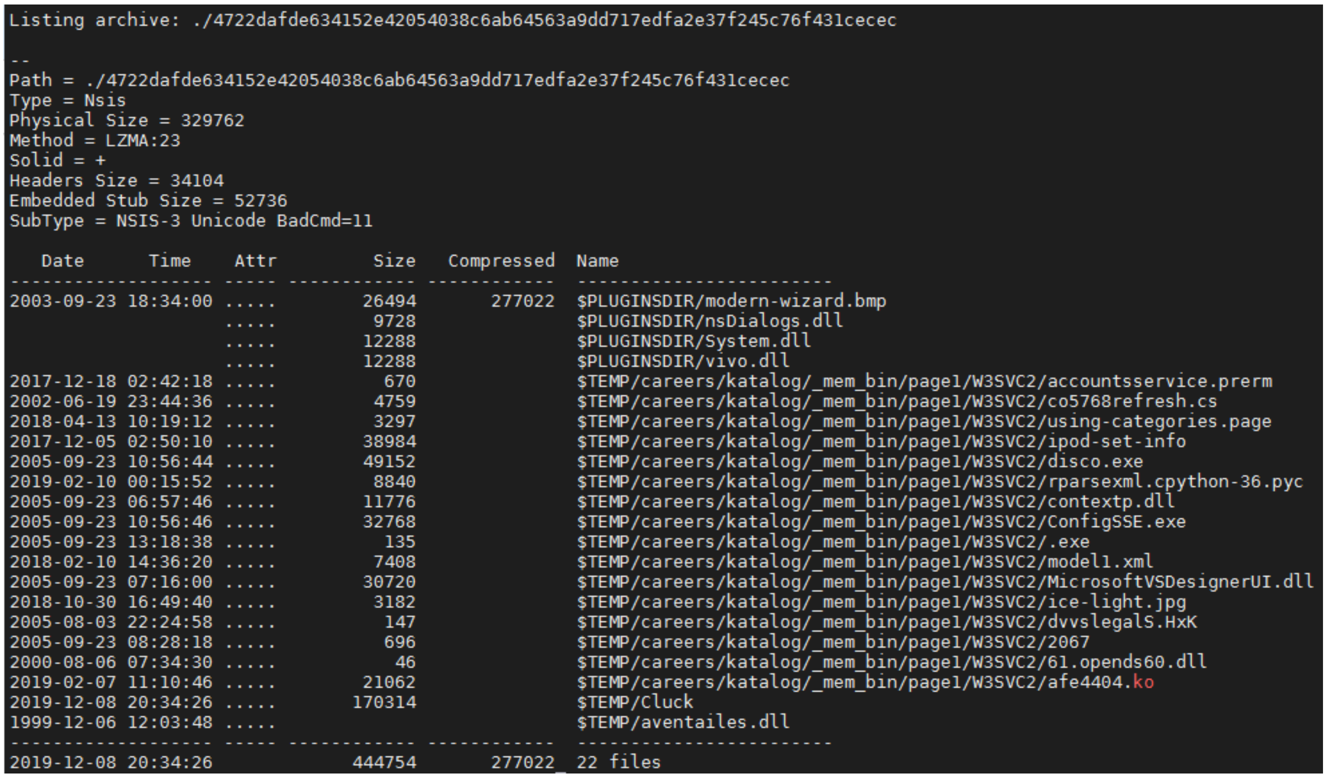


Exeinfo PE - ver.0.0.4.4 by A.S.L - 966+54 sign 2016.09.29

File : 4722dafde634152e42054038c6ab64563a9dd717edfa2e37f2
Entry Point : 000034A5
EP Section : .text
File Offset : 000028A5
First Bytes : 81.EC.D4.02.00
Linker Info : 6.00
SubSystem : Windows GUI
File Size : 00050822h
Overlay : 00043A22
Image is 32bit executable
RES/OVL : 5 / 84 % 2018
Generic check : Installer Nullsoft PiMP Stub / v.3.04 - http://nsis.sourc
Lamer Info - Help Hint - Unpack info
try : 7zip to view files - www.7-zip.org or Total Commander + plugin - I

The output of the Exeinfo PE tool identifies the sample as an NSIS installer

NSIS installers contain compressed components, including executable code, which can be loaded into memory by the installers. These components can be extracted using file decompression tools, such as 7zip.



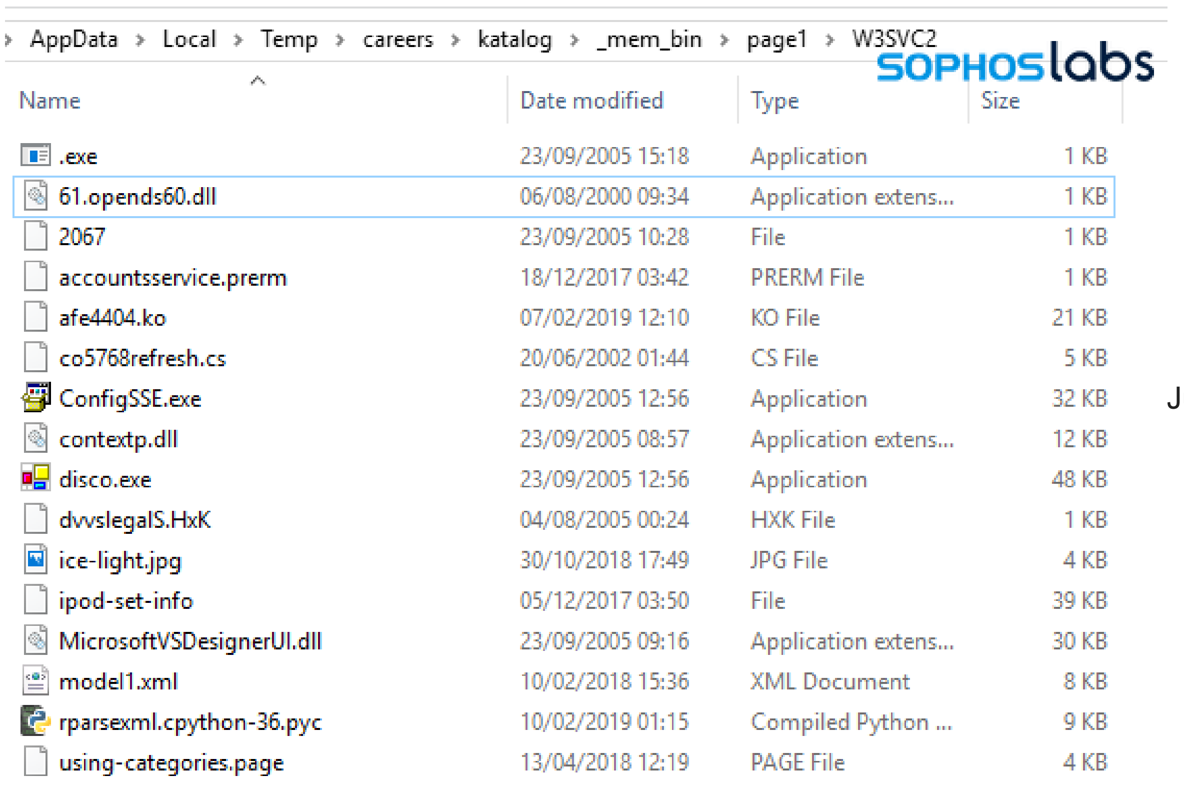
```
Listing archive: ./4722dafde634152e42054038c6ab64563a9dd717edfa2e37f245c76f431cecec

--
Path = ./4722dafde634152e42054038c6ab64563a9dd717edfa2e37f245c76f431cecec
Type = Nsis
Physical Size = 329762
Method = LZMA:23
Solid = +
Headers Size = 34104
Embedded Stub Size = 52736
SubType = NSIS-3 Unicode BadCmd=11

   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
2003-09-23 18:34:00 .....        26494       277022  $PLUGINSDIR/modern-wizard.bmp
                    .....         9728               $PLUGINSDIR/nsDialogs.dll
                    .....        12288               $PLUGINSDIR/System.dll
                    .....        12288               $PLUGINSDIR/vivo.dll
2017-12-18 02:42:18 .....          670               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/accountsservice.prerm
2002-06-19 23:44:36 .....         4759               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/co5768refresh.cs
2018-04-13 10:19:12 .....         3297               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/using-categories.page
2017-12-05 02:50:10 .....        38984               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/ipod-set-info
2005-09-23 10:56:44 .....        49152               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/disco.exe
2019-02-10 00:15:52 .....         8840               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/rparsexml.cpython-36.pyc
2005-09-23 06:57:46 .....        11776               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/contextp.dll
2005-09-23 10:56:46 .....        32768               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/ConfigSSE.exe
2005-09-23 13:18:38 .....          135               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/.exe
2018-02-10 14:36:20 .....         7408               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/model1.xml
2005-09-23 07:16:00 .....        30720               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/MicrosoftVSDesignerUI.dll
2018-10-30 16:49:40 .....         3182               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/ice-light.jpg
2005-08-03 22:24:58 .....          147               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/dvvslegalS.HxK
2005-09-23 08:28:18 .....          696               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/2067
2000-08-06 07:34:30 .....           46               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/61.opends60.dll
2019-02-07 11:10:46 .....        21062               $TEMP/careers/katalog/_mem_bin/page1/W3SVC2/afe4404.ko
2019-12-08 20:34:26 .....       170314               $TEMP/Cluck
1999-12-06 12:03:48 .....                            $TEMP/aventailes.dll
------------------- ----- ------------ ------------  ------------------------
2019-12-08 20:34:26             444754       277022  22 files
```

Output of 7zip after list the files contained on the analyzed sample

The files dropped by this sample included the following types:

- ASCII text
- C source files, in ASCII text
- data
- Executable and Linkable Format (ELF) 64-bit
- GIF image data
- JPEG image data
- PC bitmap, Windows 3.x format, 164 x 314 x 4
- PE32 executable (DLL)
- PE32 executable (GUI)
- POSIX shell script, ASCII text executable
- Python 3.6 byte-compiled
- XML 1.0 document

The installer drops the junk files into the *%TEMP%/careers/katalog/_mem_bin/page1/W3SVC2* folder.



> AppData > Local > Temp > careers > katalog > _mem_bin > page1 > W3SVC2

| Name | Date modified | Type | Size |
|---|---|---|---|
| .exe | 23/09/2005 15:18 | Application | 1 KB |
| 61.opends60.dll | 06/08/2000 09:34 | Application extens... | 1 KB |
| 2067 | 23/09/2005 10:28 | File | 1 KB |
| accountsservice.prerm | 18/12/2017 03:42 | PRERM File | 1 KB |
| afe4404.ko | 07/02/2019 12:10 | KO File | 21 KB |
| co5768refresh.cs | 20/06/2002 01:44 | CS File | 5 KB |
| ConfigSSE.exe | 23/09/2005 12:56 | Application | 32 KB |
| contextp.dll | 23/09/2005 08:57 | Application extens... | 12 KB |
| disco.exe | 23/09/2005 12:56 | Application | 48 KB |
| dvvslegalS.HxK | 04/08/2005 00:24 | HXK File | 1 KB |
| ice-light.jpg | 30/10/2018 17:49 | JPG File | 4 KB |
| ipod-set-info | 05/12/2017 03:50 | File | 39 KB |
| MicrosoftVSDesignerUI.dll | 23/09/2005 09:16 | Application extens... | 30 KB |
| model1.xml | 10/02/2018 15:36 | XML Document | 8 KB |
| rparsexml.cpython-36.pyc | 10/02/2019 01:15 | Compiled Python ... | 9 KB |
| using-categories.page | 13/04/2018 12:19 | PAGE File | 4 KB |

Junk files created by analyzed sample.

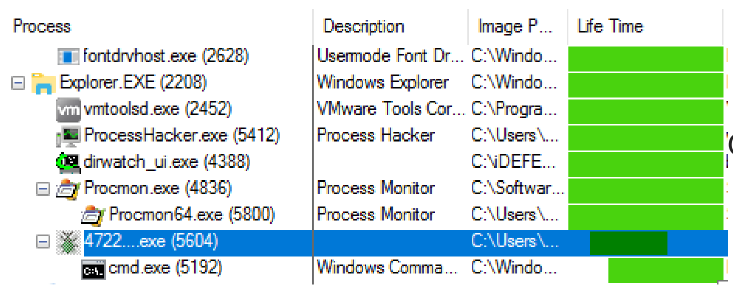
There are only two components dropped by the installer that are important to the malware installation, which are dropped into the *$TEMP* folder. In the case of the NSIS installer we analyzed for this report, these two components are:

- aventailes.dll (the Initial Loader)
- Cluck (Encrypted data)

The payloads of the installers we examined vary. During analysis of the samples we collected—conducted both manually and with the aid of sandboxing tools—we found several different families of RATs and infostealers. These included Lokibot, Betabot, Formbook, and AgentTesla. But all of them followed the same multi-stage unpacking process when executed.

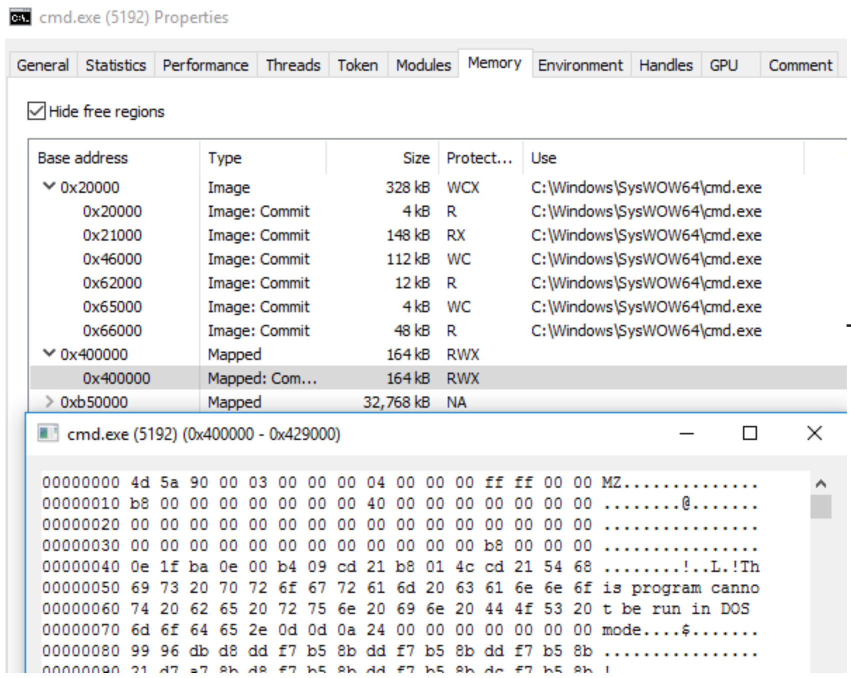
## First stage: initial loader and shellcode

In the first stage, the installer deploys the initial loader, a malicious DLL. The DLL is then used to begin decryption of the malicious payload, and then finally to inject malicious payload into memory while the NSIS layer drops the junk files. The following images show how the analyzed sample creates a cmd.exe process, which is used to inject the Final Payload.



| Process | Description | Image P... | Life Time |
|---|---|---|---|
| fontdrvhost.exe (2628) | Usermode Font Dr... | C:\Windo... | 🟩 |
| Explorer.EXE (2208) | Windows Explorer | C:\Windo... | 🟩 |
| vmtoolsd.exe (2452) | VMware Tools Cor... | C:\Progra... | 🟩 |
| ProcessHacker.exe (5412) | Process Hacker | C:\Users\... | 🟩 |
| dirwatch_ui.exe (4388) |  | C:\DEFE... | 🟩 |
| Procmon.exe (4836) | Process Monitor | C:\Softwar... | 🟩 |
| Procmon64.exe (5800) | Process Monitor | C:\Users\... | 🟩 |
| 4722....exe (5604) |  | C:\Users\... | 🟩 |
| cmd.exe (5192) | Windows Comma... | C:\Windo... | 🟩 |

Output of Procmon which shows how the analyzed sample creates a child process



cmd.exe (5192) Properties

☑ Hide free regions

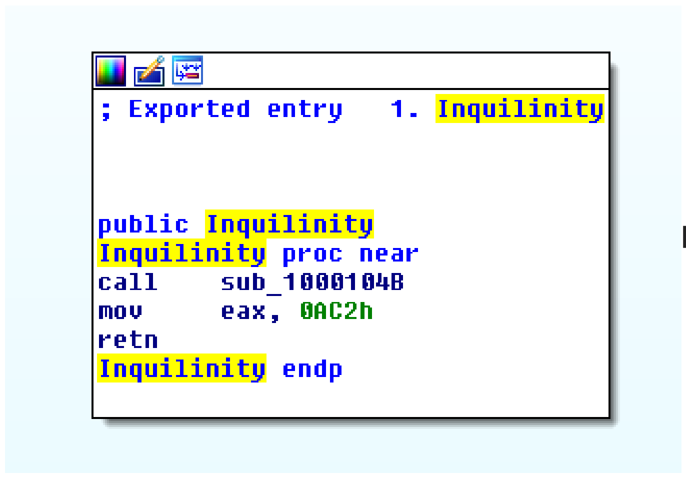
| Base address | Type | Size | Protect... | Use |
|---|---|---|---|---|
| 0x20000 | Image | 328 kB | WCX | C:\Windows\SysWOW64\cmd.exe |
| 0x20000 | Image: Commit | 4 kB | R | C:\Windows\SysWOW64\cmd.exe |
| 0x21000 | Image: Commit | 148 kB | RX | C:\Windows\SysWOW64\cmd.exe |
| 0x46000 | Image: Commit | 112 kB | WC | C:\Windows\SysWOW64\cmd.exe |
| 0x62000 | Image: Commit | 12 kB | R | C:\Windows\SysWOW64\cmd.exe |
| 0x65000 | Image: Commit | 4 kB | WC | C:\Windows\SysWOW64\cmd.exe |
| 0x66000 | Image: Commit | 48 kB | R | C:\Windows\SysWOW64\cmd.exe |
| 0x400000 | Mapped | 164 kB | RWX |  |
| 0x400000 | Mapped: Com... | 164 kB | RWX |  |
| 0xb50000 | Mapped | 32,768 kB | NA |  |

cmd.exe (5192) (0x400000 - 0x429000)

```
00000000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ..............
00000010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ........@.......
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000030 00 00 00 00 00 00 00 00 00 00 00 00 b8 00 00 00 ................
00000040 0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68 ........!..L.!Th
00000050 69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f is program canno
00000060 74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20 t be run in DOS 
00000070 6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00 mode....$.......
00000080 99 96 db d8 dd f7 b5 8b dd f7 b5 8b dd f7 b5 8b ................
00000090 21 d7 a7 8b d8 f7 b5 8b dd f7 b5 8b dc f7 b5 8b !
```

The memory of created child process by the analyzed sample. The final payload is loaded at the address 0x400000.

The malicious DLL deployed with the RATicate installers (in this case, **aventailes.dll**) is a custom loader, likely developed by the threat actor, stored in the *$TEMP* folder of the file package. All of the analyzed initial loaders are DLL files with only one export, though the name of the loader and the export function vary across the samples. In this case, the export was named Inquilinity.

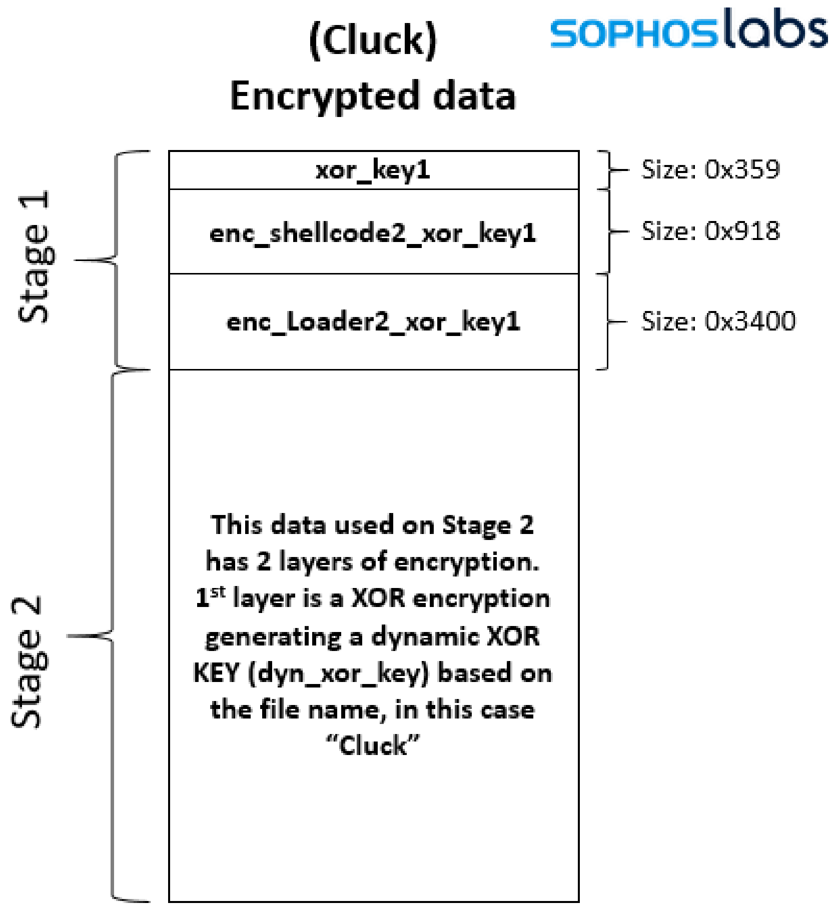


```
; Exported entry   1. Inquilinity


public Inquilinity
Inquilinity proc near
call     sub_1000104B
mov      eax, 0AC2h
retn
Inquilinity endp
```

Export of Initial Loader

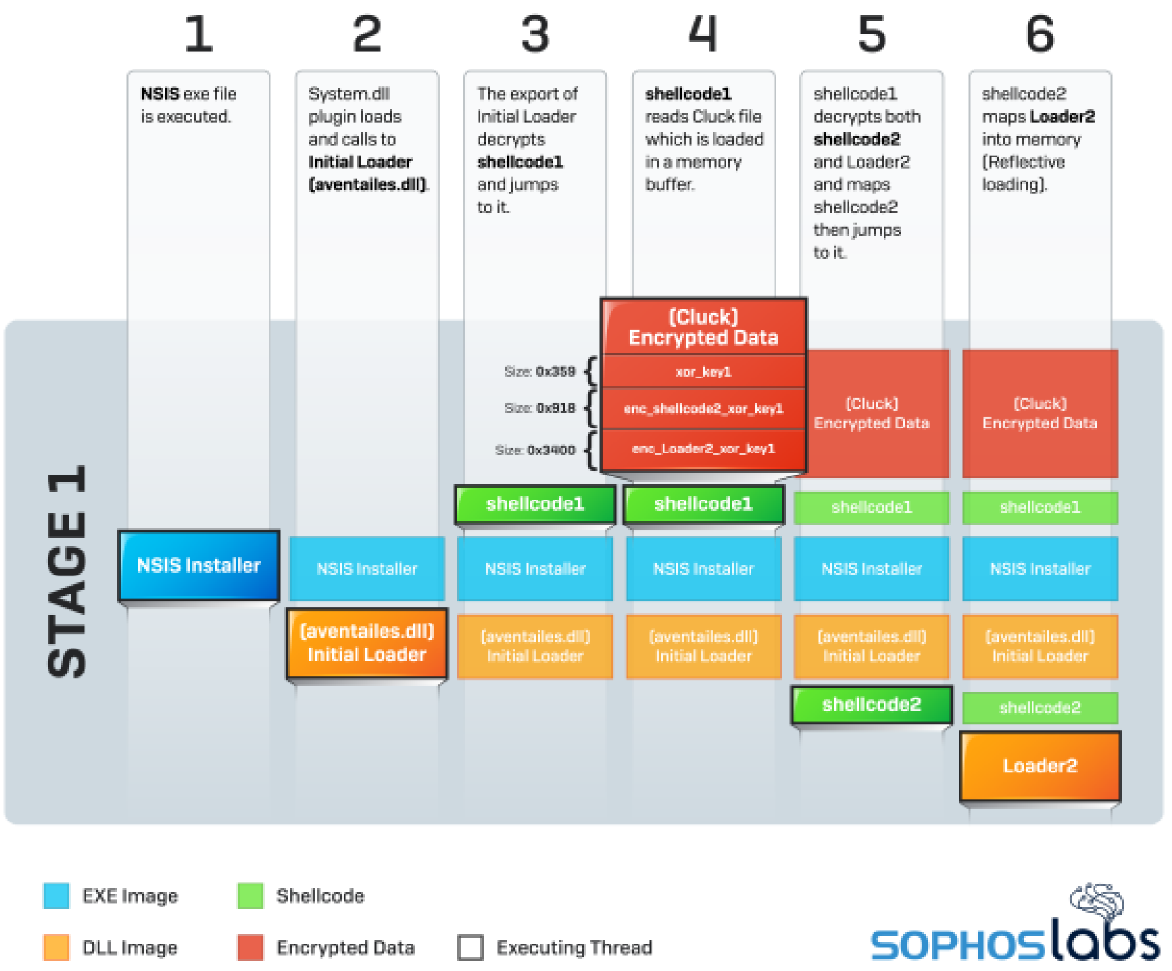
This export is called using the NSIS System plugin as explained previously. The export loads and executes a shellcode, located in the initial loader's .rdata section. The shellcode is initially encrypted using a basic arithmetic operation. This operation varies across the initial loaders we analyzed.

The shellcode dropped by the initial loader then reads the Encrypted data (Cluck file) where other loaders and payloads are stored. These PE files and shellcodes are decrypted on demand during the next two stages of malware deployment. In the first stage of the decryption, done by the shellcode called by initial loader, contains an xor key, a second shellcode (shellcode 2), and a PE file (Loader 2).


(Cluck)
Encrypted data
SOPHOSlabs
Stage 1
Stage 2
xor_key1
enc_shellcode2_xor_key1
enc_Loader2_xor_key1
Size: 0x359
Size: 0x918
Size: 0x3400
This data used on Stage 2 has 2 layers of encryption. 1st layer is a XOR encryption generating a dynamic XOR KEY (dyn_xor_key) based on the file name, in this case "Cluck"

The xor key is used to decrypt shellcode2 and Loader 2.

Here's how the workflow of Stage 1 breaks down in depth:

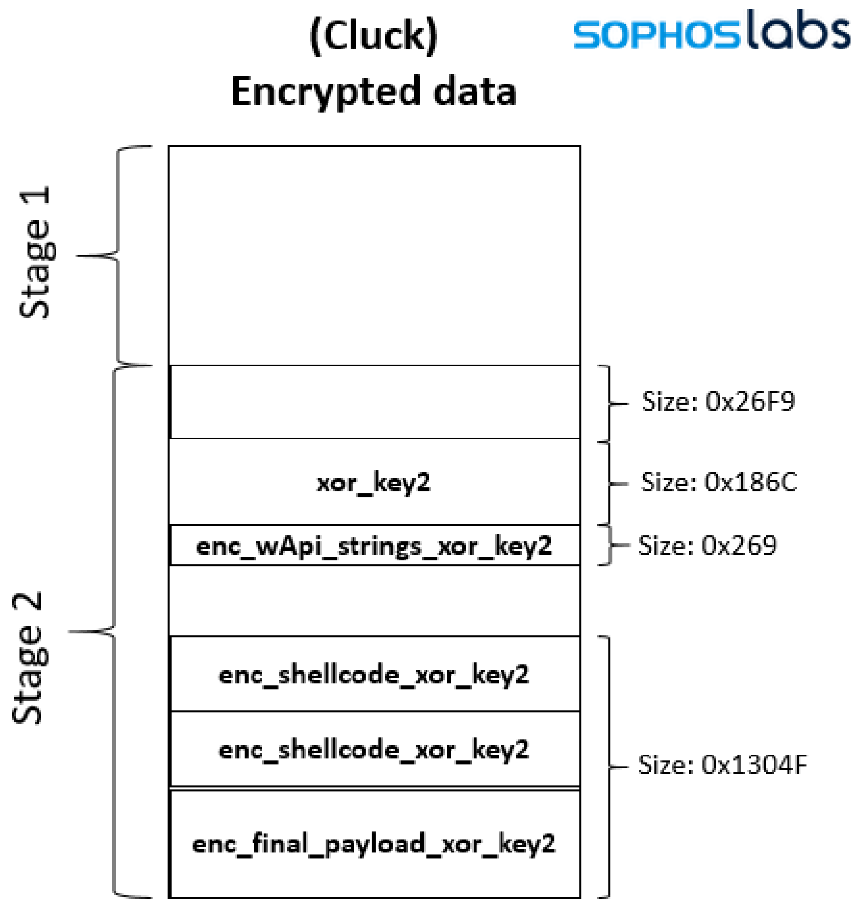

1
NSIS exe file is executed.
2
System.dll plugin loads and calls to Initial Loader (aventailes.dll).
3
The export of Initial Loader decrypts shellcode1 and jumps to it.
4
shellcode1 reads Cluck file which is loaded in a memory buffer.
5
shellcode1 decrypts both shellcode2 and Loader2 and maps shellcode2 then jumps to it.
6
shellcode2 maps Loader2 into memory (Reflective loading).
STAGE 1
NSIS Installer
(aventailes.dll) Initial Loader
shellcode1
(Cluck) Encrypted Data
xor_key1
enc_shellcode2_xor_key1
enc_Loader2_xor_key1
Size: 0x359
Size: 0x918
Size: 0x3400
shellcode2
Loader2
EXE Image
DLL Image
Shellcode
Encrypted Data
Executing Thread
SOPHOSlabs

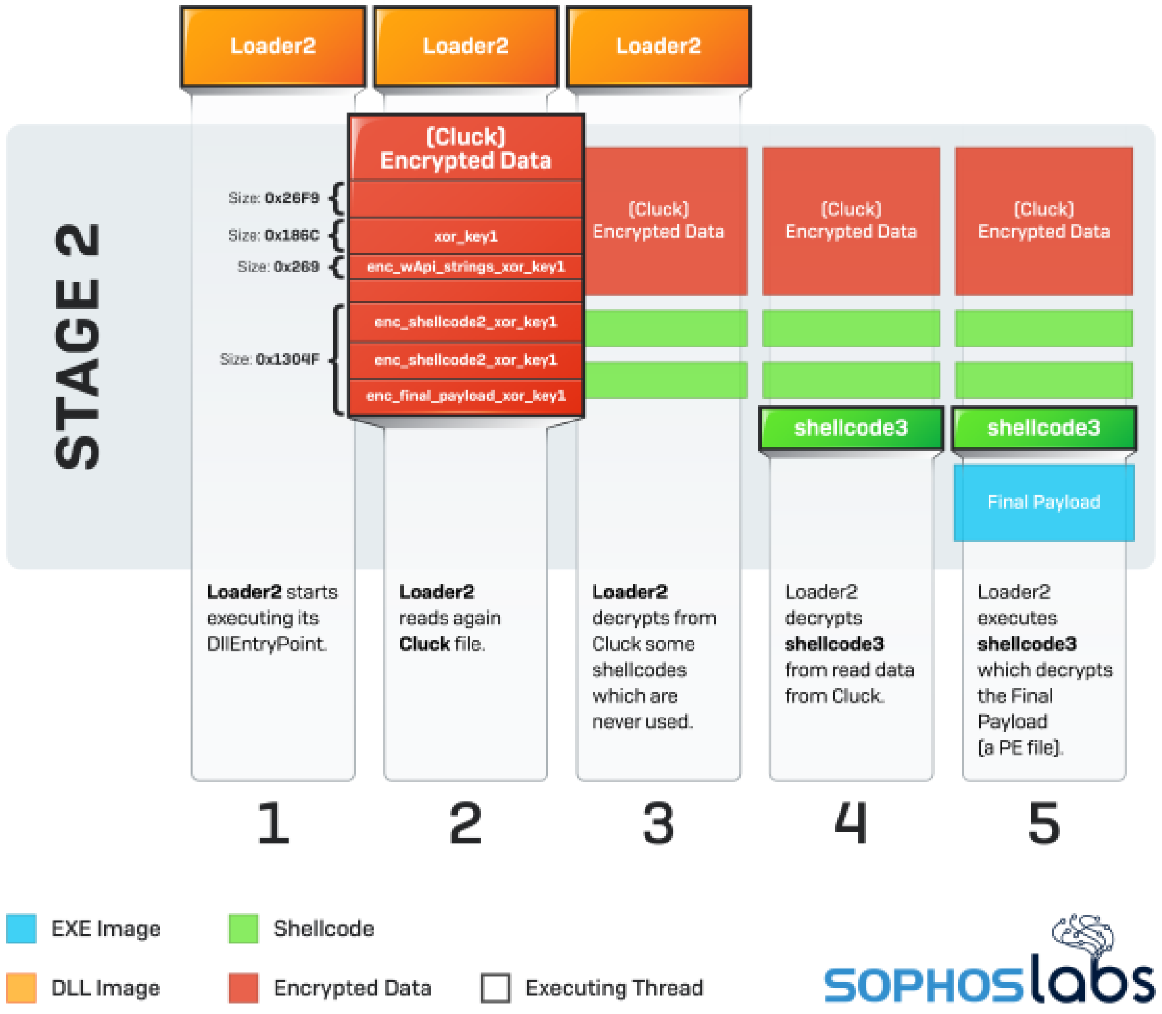
Stage 1 workflow:

1. NSIS exe file is executed.
2. System.dll plugin loads and calls to Initial Loader (aventailes.dll)
3. The export of Initial Loader decrypts shellcode1 and jumps to it.
4. shellcode1 reads Cluck file which is loaded in a memory buffer.
5. shellcode1 decrypts both shellcode2 and Loader2 and maps shellcode2 then jumps to it.
6. shellcode2 maps Loader2 into memory (Reflective loading).

## Second stage: second shellcode and loader DLL

The second stage of decryption begins when Loader 2 is loaded in memory by shellcode2. Loader 2 reads the Cluck file in order to decrypt more artifacts. The data for this stage is decrypted with a dynamically generated xor key based on the name of the file which contains the encrypted data (which in this case is *Cluck*). As shown below, after this xor is applied, there is another xor key (xor_key2) stored in the second part of the file, which is used to decrypt different artifacts like strings, shellcodes, and PE files.


(Cluck)
Encrypted data
sophoslabs
Stage 1
Stage 2
Size: 0x26F9
xor_key2
Size: 0x186C
enc_wApi_strings_xor_key2
Size: 0x269
enc_shellcode_xor_key2
enc_shellcode_xor_key2
enc_final_payload_xor_key2
Size: 0x1304F



Stage 2 workflow

1. Loader2 starts executing its DllEntryPoint.
2. Loader2 reads again Cluck file.
3. Loader2 decrypts from Cluck some shellcodes which are never used.
4. Loader2 decrypts shellcode3 from read data from Cluck.
5. Loader2 executes shellcode3, which decrypts the Final Payload (a PE file).

## Third stage: injection

After the decryption, shellcode3 injects the final payload in a child process. It accomplishes this using cmd.exe with the NtCreateSection + NtMapViewOfSection code injection technique.

These are the extracted artifacts during the analysis.

| ARTIFACT | HASH |
|---|---|
| Loader 2 | c2cdb371d3394ff71918ac2422a84408644fa603f1b45e3fb1a438dbce9dcad0 |

| | |
|---|---|
| Final Payload | 46c6fa90acdf651e99620c257ae4e9ed9d1cfcb31fd676dc9b570bb3f9720ac8 |

## Hints of a single actor

We found 38 NSIS installer samples in total that shared very similar characteristics:

Identical junk files. Not only their name, but also their content. When generating the installer from NSIS Script, the actor who is packing the payload would have to have all these random files in their possession on their hard drive.

The loader is the same: All the loaders across analyzed NSIS installers are the same, not in terms of their hash value but in terms of their functionality.

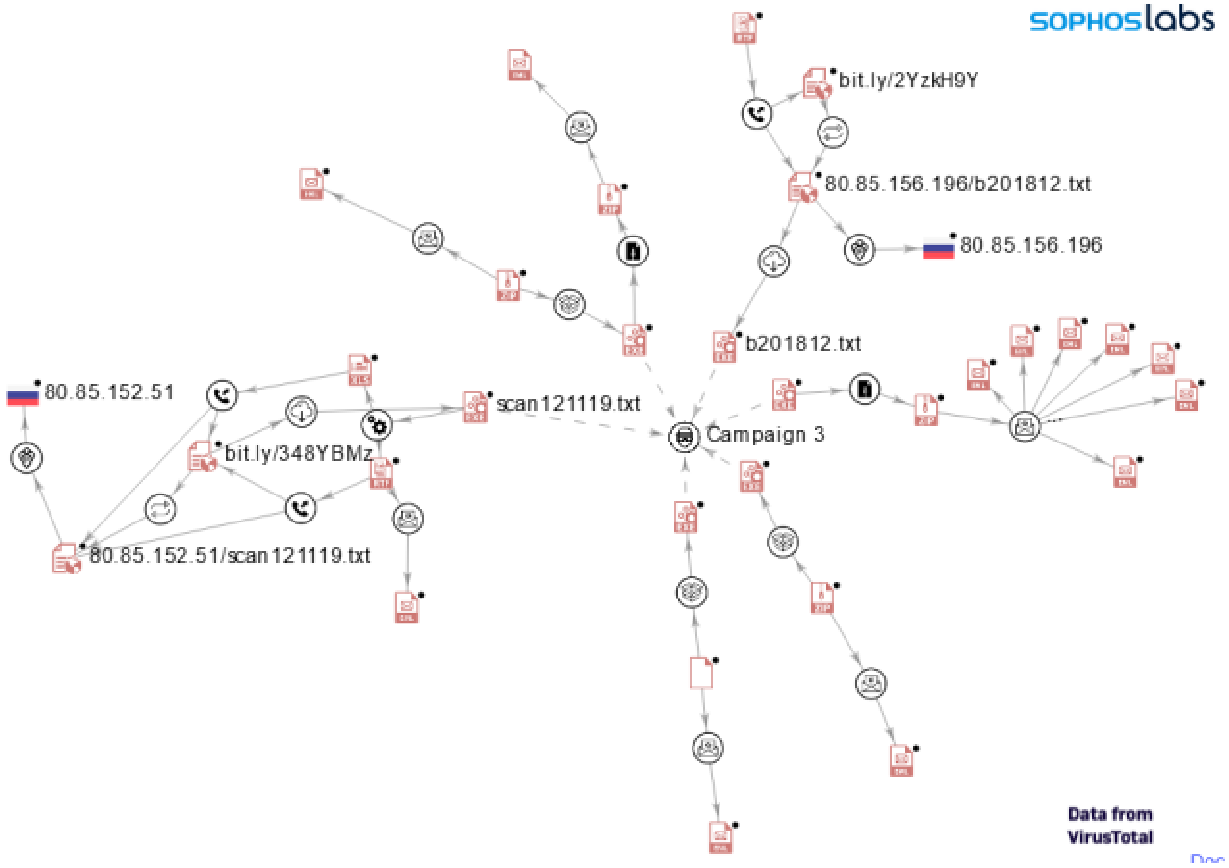
- All initial loaders have just one export, which is called by the NSIS installer
- The Initial Loader reads from Encrypted Data in order to decrypt a shellcode which loads the Loader 2.
- Loader 2 across all samples extracts and decrypts shellcode 3 from Encrypted Data.
- Shellcode 3, responsible for decrypting the final payload and injecting it into a remote process, is binary-equal between all analyzed samples.

However, each NSIS installer we looked at dropped different malware payloads. We considered two possible scenarios: either the malicious NSIS package is a generic packer sold on dark forums; or, the same threat actor is using a custom loader to deploy different payloads in a variety of their attacks.

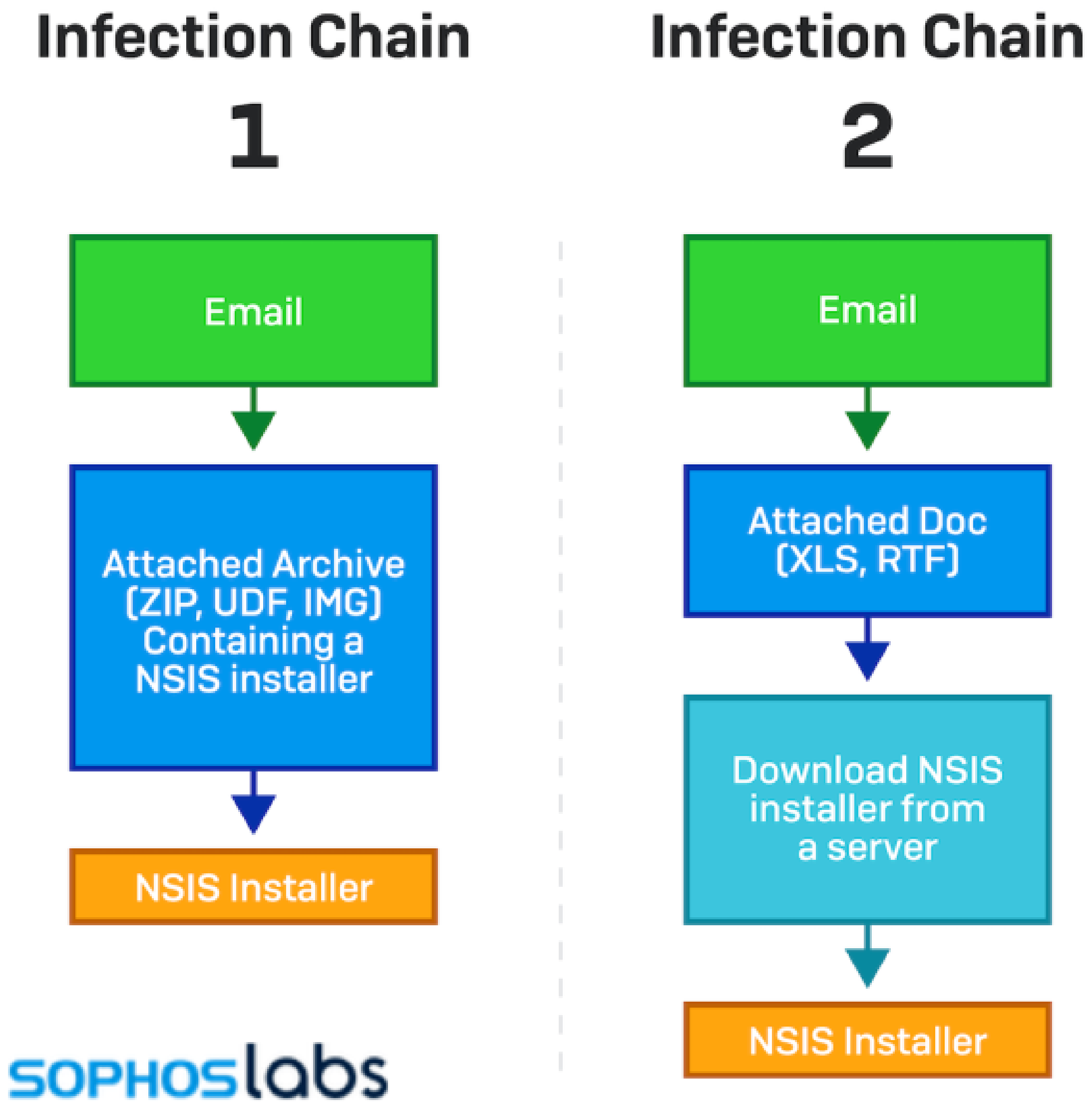
While there are many packers sold in dark forums, we found this scenario unlikely, as one should expect the junk files to change along with the payloads, if different actors were using the same generic packer. So, we continued our investigation with the hypothesis the attacks come from the same actor.

Given the evidence we have in hand, we can't prove that a single actor was responsible for all of them, but we at least knew from the identical packing strategy and artifacts that we could find a way to connect all of them. We performed further analysis in search of a definitive link, turning to the infection chain that delivered them.

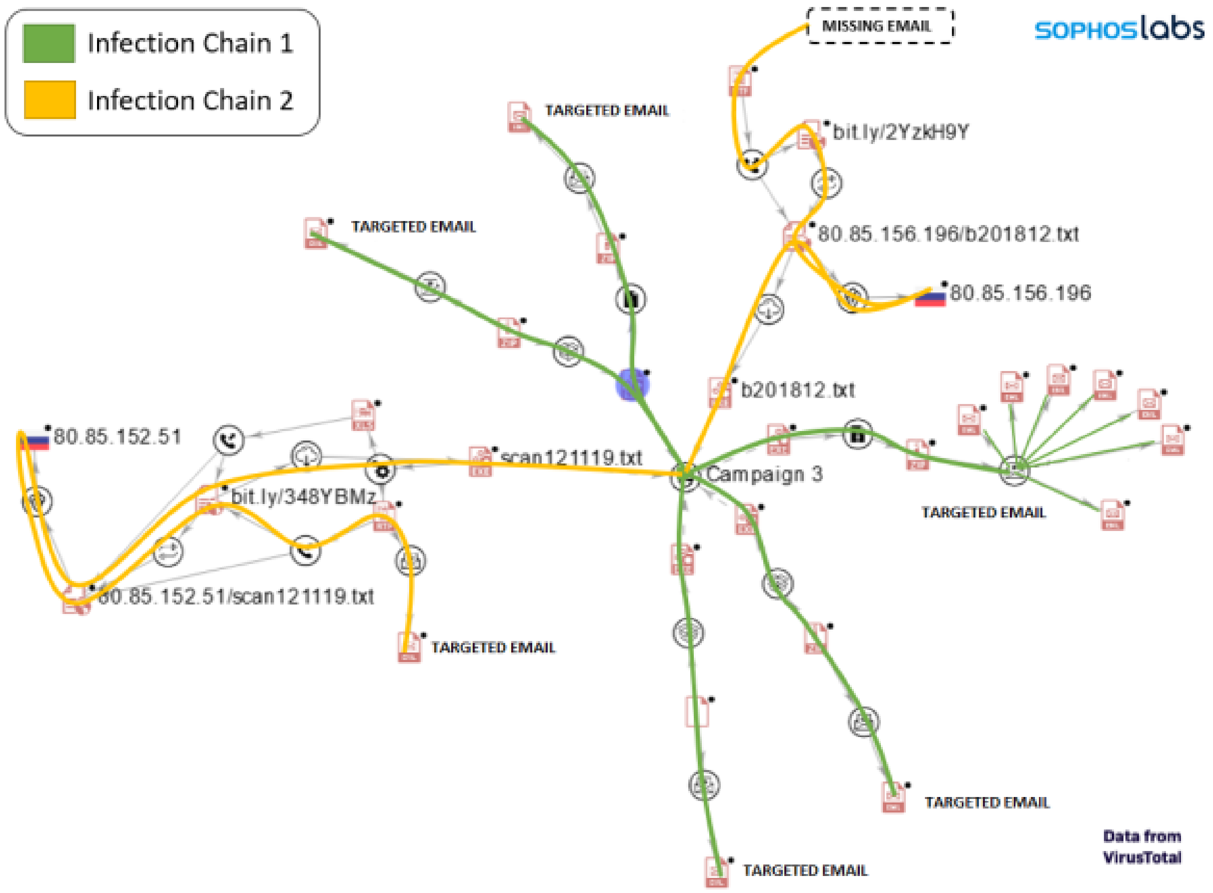
Based on Sophos telemetry, we found a set of NSIS installers dropping these same junk files as part of an email campaign seen between December 8 and December 13, 2019. (We later designated this wave Campaign 3, after discovering other sets of NSIS installers, discussed later.) In the email attacks we observed, the targets appeared to all be critical infrastructure providers (or businesses related to critical infrastructure). We analyzed the observed attacks using VirusTotal's graphing feature, gathering open-source information about other victims.



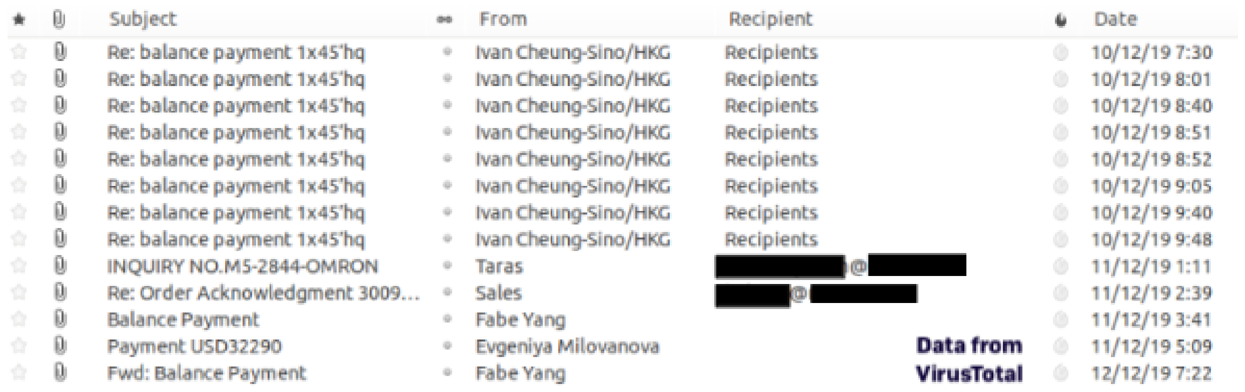
The graph above shows the infection chain for some of the analyzed NSIS installers. It reveals two common patterns used to infect a victim:



Superimposing the distinct infection chains over the graph shows that both chains were used for the same target company revealed by VT data. It is likely the same approach is taken for any targeted company.

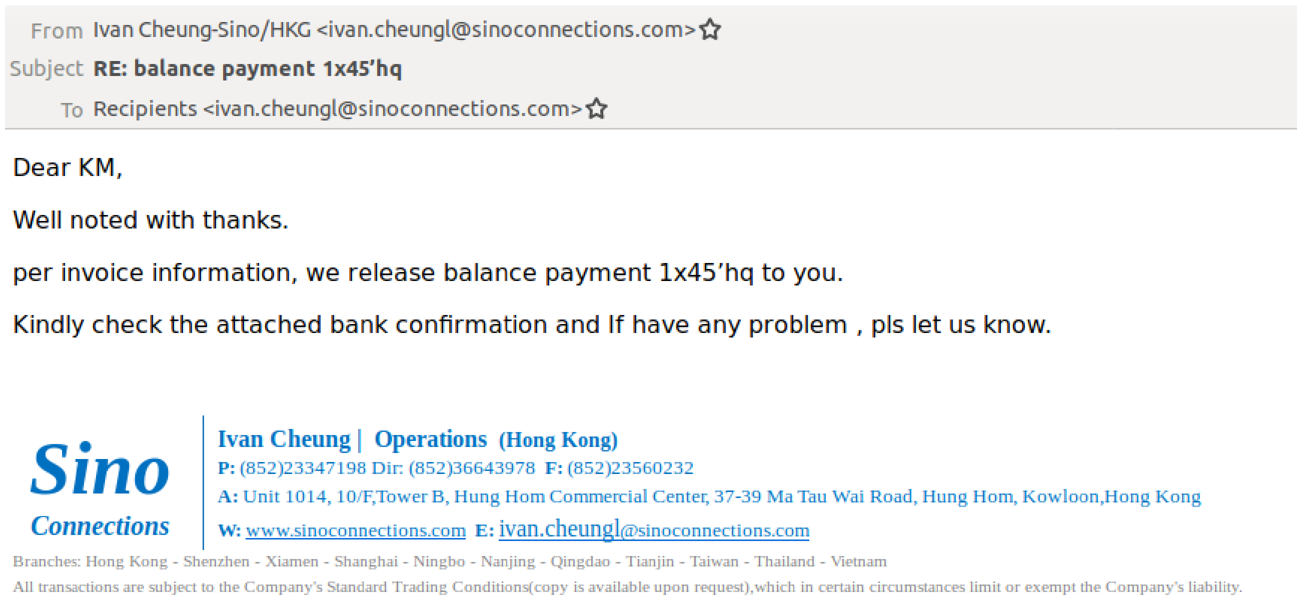


We were able to retrieve some of the emails associated with this campaign from VT. With these emails, we were able to identify some of the installers' targets.



| Subject | From | Recipient | Date |
|---|---|---|---|
| Re: balance payment 1x45'hq | Ivan Cheung-Sino/HKG | Recipients | 10/12/19 7:30 |
| Re: balance payment 1x45'hq | Ivan Cheung-Sino/HKG | Recipients | 10/12/19 8:01 |
| Re: balance payment 1x45'hq | Ivan Cheung-Sino/HKG | Recipients | 10/12/19 8:40 |
| Re: balance payment 1x45'hq | Ivan Cheung-Sino/HKG | Recipients | 10/12/19 8:51 |
| Re: balance payment 1x45'hq | Ivan Cheung-Sino/HKG | Recipients | 10/12/19 8:52 |
| Re: balance payment 1x45'hq | Ivan Cheung-Sino/HKG | Recipients | 10/12/19 9:05 |
| Re: balance payment 1x45'hq | Ivan Cheung-Sino/HKG | Recipients | 10/12/19 9:40 |
| Re: balance payment 1x45'hq | Ivan Cheung-Sino/HKG | Recipients | 10/12/19 9:48 |
| INQUIRY NO.M5-2844-OMRON | Taras | [redacted]@[redacted] | 11/12/19 1:11 |
| Re: Order Acknowledgment 3009… | Sales | [redacted]@[redacted] | 11/12/19 2:39 |
| Balance Payment | Fabe Yang | | 11/12/19 3:41 |
| Payment USD32290 | Evgeniya Milovanova | | 11/12/19 5:09 |
| Fwd: Balance Payment | Fabe Yang | | 12/12/19 7:22 |

Data from VirusTotal


From Ivan Cheung-Sino/HKG <ivan.cheungl@sinoconnections.com>
Subject **RE: balance payment 1x45'hq**
To Recipients <ivan.cheungl@sinoconnections.com>

Dear KM,

Well noted with thanks.

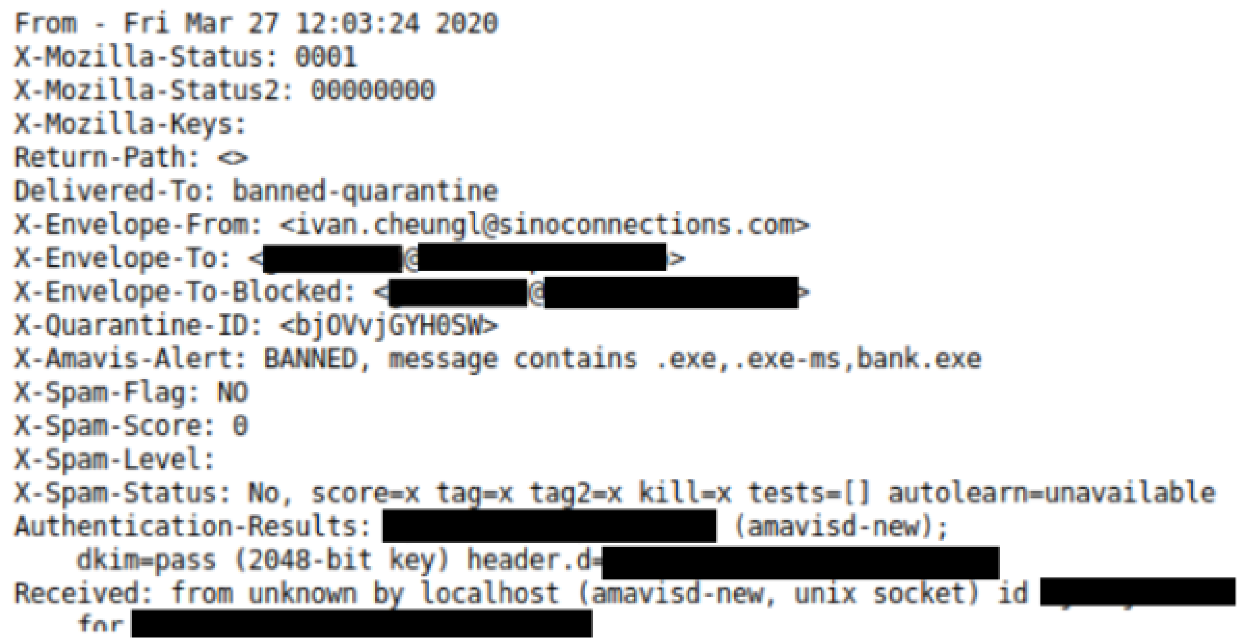
per invoice information, we release balance payment 1x45'hq to you.

Kindly check the attached bank confirmation and If have any problem , pls let us know.

Sino Connections

Ivan Cheung | Operations (Hong Kong)
P: (852)23347198 Dir: (852)36643978 F: (852)23560232
A: Unit 1014, 10/F,Tower B, Hung Hom Commercial Center, 37-39 Ma Tau Wai Road, Hung Hom, Kowloon,Hong Kong
W: www.sinoconnections.com E: ivan.cheungl@sinoconnections.com
Branches: Hong Kong - Shenzhen - Xiamen - Shanghai - Ningbo - Nanjing - Qingdao - Tianjin - Taiwan - Thailand - Vietnam
All transactions are subject to the Company's Standard Trading Conditions(copy is available upon request),which in certain circumstances limit or exempt the Company's liability.

One of the Campaign 3 emails, presenting the installer as a "banking confirmation."
Many of the the emails we found in VirusTotal data did not show recipients' addresses, or the "To" address was filled with the same email address that appeared in the "From" field. In these cases, we analyzed the email headers—since the headers hold more information related to the email, like the original recipients.


```
From - Fri Mar 27 12:03:24 2020
X-Mozilla-Status: 0001
X-Mozilla-Status2: 00000000
X-Mozilla-Keys:
Return-Path: <>
Delivered-To: banned-quarantine
X-Envelope-From: <ivan.cheungl@sinoconnections.com>
X-Envelope-To: <[redacted]@[redacted]>
X-Envelope-To-Blocked: <[redacted]@[redacted]>
X-Quarantine-ID: <bjOVvjGYH0SW>
X-Amavis-Alert: BANNED, message contains .exe,.exe-ms,bank.exe
X-Spam-Flag: NO
X-Spam-Score: 0
X-Spam-Level:
X-Spam-Status: No, score=x tag=x tag2=x kill=x tests=[] autolearn=unavailable
Authentication-Results: [redacted] (amavisd-new);
    dkim=pass (2048-bit key) header.d=[redacted]
Received: from unknown by localhost (amavisd-new, unix socket) id [redacted]
    for [redacted]
```

During the analysis of the NSIS installers we found with identical junk files to our initial sample, we identified at least 5 different malware families used as final payload—all of them InfoStealer or RAT malware:

- ForeIT/Lokibot
- BetaBot
- Formbook
- AgentTesla
- Netwire

We then looked at the Command and Control (C&C) infrastructure used for these payloads, to check for any relationship between them and to see if the C&Cs were used to send the stolen data points to same or similar servers.

These are some of the families identified in this campaign and their C&Cs:

| TYPE | OBSERVED PAYLOADS | OBSERVED PAYLOAD C&C DOMAINS |
|---|---|---|
| Info Stealer | Betabot | stngpetty.ga<br>allenservice.ga<br>gelcursot.top |
| Info Stealer | Formbook | ef-oh.com/c208<br>odoyo.net/c208<br>hearee.com/c208<br>binzom.com/c208<br>pizzans.com/c208<br>phochain.com/sa<br>rdrfi.com/sa/<br>skylod.com/sa<br>hsctsu.com/sa |
| Info Stealer | Lokibot | gelcursot.top |
| RAT | Netwire | 79.134.225.97:2556 |
| RAT | AgentTesla | mail.newmedicacare.com<br>mail.jrdigitalstore.com<br>mail.koyo.com.my<br>mail.qoa.com.my<br>mail.sedirectory.com.my |

Almost all of the malware samples of each type connected to the campaign share the same C&C. And in some cases, even different families—such as Lokibot and Betabot—share same domain for their C&C.

## Identifying more campaigns

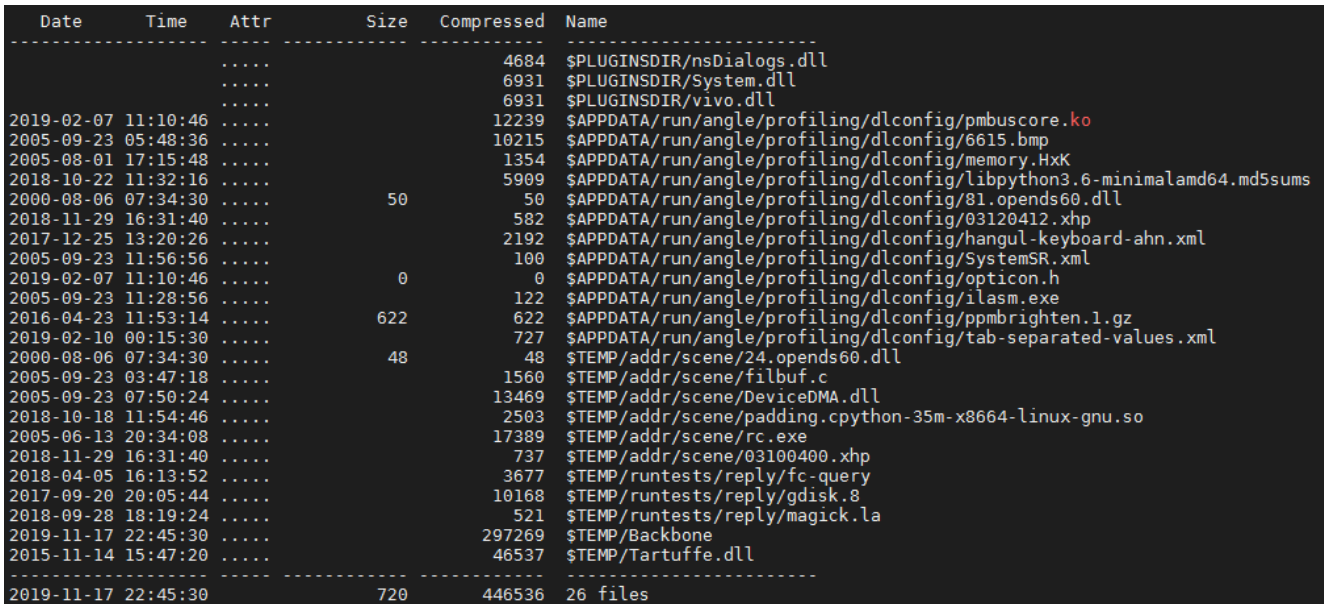
Following this pattern—looking for other groups of NSIS installers which drop identical junk files during the same range of dates—we were able to identify 5 distinct NSIS campaigns that took place between November 16, 2019 and January 8, 2020. While the junk files for each of these campaigns were different from our first samples, their behavior was identical (or at least similar) to those observed in Campaign 3.

| NAME | DATES |
|---|---|
| Campaign 1 | 2019-11-16/2019-11-20 |
| Campaign 2 | 2019-11-25/2019-11-26 |

| | |
|---|---|
| Campaign 3 | 2019-12-08/2019-12-13 |
| Campaign 4 | 2019-12-20/2019-12-31 |
| Campaign 5 | 2020-01-03/2020-01-08 |

## Campaign 1 (November 16-20, 2019)

These are the dropped junk files for all NSIS installers that belong to Campaign 1:



```
   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
                    .....                    4684  $PLUGINSDIR/nsDialogs.dll
                    .....                    6931  $PLUGINSDIR/System.dll
                    .....                    6931  $PLUGINSDIR/vivo.dll
2019-02-07 11:10:46 .....                   12239  $APPDATA/run/angle/profiling/dlconfig/pmbuscore.ko
2005-09-23 05:48:36 .....                   10215  $APPDATA/run/angle/profiling/dlconfig/6615.bmp
2005-08-01 17:15:48 .....                    1354  $APPDATA/run/angle/profiling/dlconfig/memory.HxK
2018-10-22 11:32:16 .....                    5909  $APPDATA/run/angle/profiling/dlconfig/libpython3.6-minimalamd64.md5sums
2000-08-06 07:34:30 .....           50         50  $APPDATA/run/angle/profiling/dlconfig/81.opends60.dll
2018-11-29 16:31:40 .....                     582  $APPDATA/run/angle/profiling/dlconfig/03120412.xhp
2017-12-25 13:20:26 .....                    2192  $APPDATA/run/angle/profiling/dlconfig/hangul-keyboard-ahn.xml
2005-09-23 11:56:56 .....                     100  $APPDATA/run/angle/profiling/dlconfig/SystemSR.xml
2019-02-07 11:10:46 .....            0          0  $APPDATA/run/angle/profiling/dlconfig/opticon.h
2005-09-23 11:28:56 .....                     122  $APPDATA/run/angle/profiling/dlconfig/ilasm.exe
2016-04-23 11:53:14 .....          622        622  $APPDATA/run/angle/profiling/dlconfig/ppmbrighten.1.gz
2019-02-10 00:15:30 .....                     727  $APPDATA/run/angle/profiling/dlconfig/tab-separated-values.xml
2000-08-06 07:34:30 .....           48         48  $TEMP/addr/scene/24.opends60.dll
2005-09-23 03:47:18 .....                    1560  $TEMP/addr/scene/filbuf.c
2005-09-23 07:50:24 .....                   13469  $TEMP/addr/scene/DeviceDMA.dll
2018-10-18 11:54:46 .....                    2503  $TEMP/addr/scene/padding.cpython-35m-x8664-linux-gnu.so
2005-06-13 20:34:08 .....                   17389  $TEMP/addr/scene/rc.exe
2018-11-29 16:31:40 .....                     737  $TEMP/addr/scene/03100400.xhp
2018-04-05 16:13:52 .....                    3677  $TEMP/runtests/reply/fc-query
2017-09-20 20:05:44 .....                   10168  $TEMP/runtests/reply/gdisk.8
2018-09-28 18:19:24 .....                     521  $TEMP/runtests/reply/magick.la
2019-11-17 22:45:30 .....                  297269  $TEMP/Backbone
2015-11-14 15:47:20 .....                   46537  $TEMP/Tartuffe.dll
------------------- ----- ------------ ------------  ------------------------
2019-11-17 22:45:30                720     446536  26 files
```

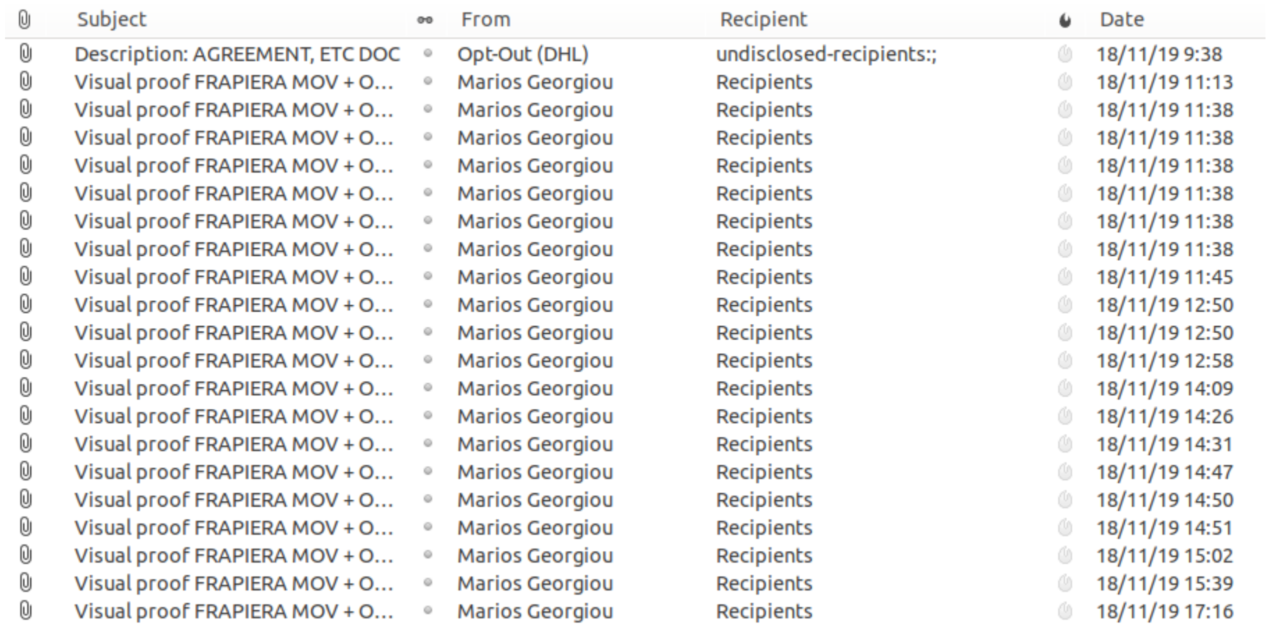
Output of 7z after list the files contained on a sample that belongs to campaign 1

These are some of the payloads identified for Campaign 1 on a first triage of the installers.

| TYPE | OBSERVED PAYLOADS | OBSERVED PAYLOAD C&C DOMAINS |
|---|---|---|
| Info Stealer | Betabot | negrodesigns.ga<br>gelcursot.top<br>webxpo.ga |
| Info Stealer | Lokibot | gelcursot.top |

| | | |
|---|---|---|
| Info Stealer | Formbook | cbespania.info/c206<br>conrak.net/c206<br>coxemen.com/c206<br>dachfix.com/c206<br>hypnose-beziers.com/c206<br>jevmod.com/c206<br>lighthouse-campus24.com/c206<br>oleum.gmbh/c206<br>pupilfy.com/c206<br>tellpizzqhut.com/c206<br>terenium.com/c206<br>vibe.restaurant/c206<br>yamatobb.com/c206<br>yncits89.com/p0x<br>ratokasutka.com/p0x<br>miscov.com/p0x |
| RAT | Netwire | 79.134.225.11:1199 |

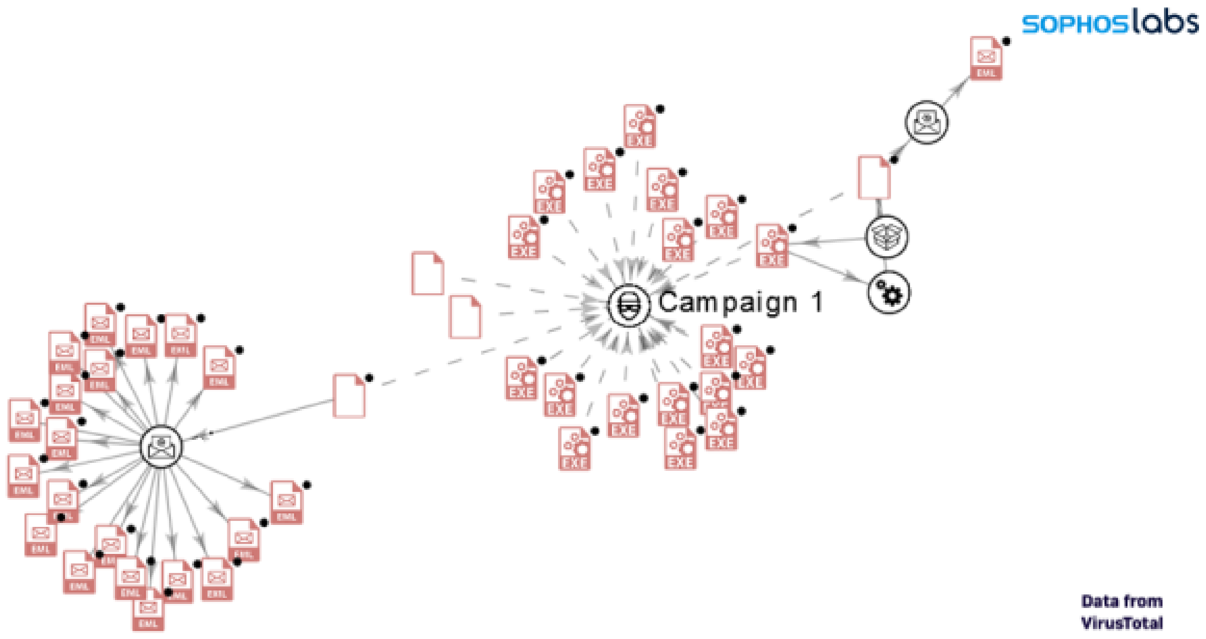
Here is a sample of the emails we collected from VirusTotal connected to Campaign 1:



| Subject | From | Recipient | Date |
|---|---|---|---|
| Description: AGREEMENT, ETC DOC | Opt-Out (DHL) | undisclosed-recipients:; | 18/11/19 9:38 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 11:13 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 11:38 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 11:38 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 11:38 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 11:38 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 11:38 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 11:38 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 11:45 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 12:50 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 12:50 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 12:58 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 14:09 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 14:26 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 14:31 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 14:47 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 14:50 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 14:51 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 15:02 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 15:39 |
| Visual proof FRAPIERA MOV + O… | Marios Georgiou | Recipients | 18/11/19 17:16 |

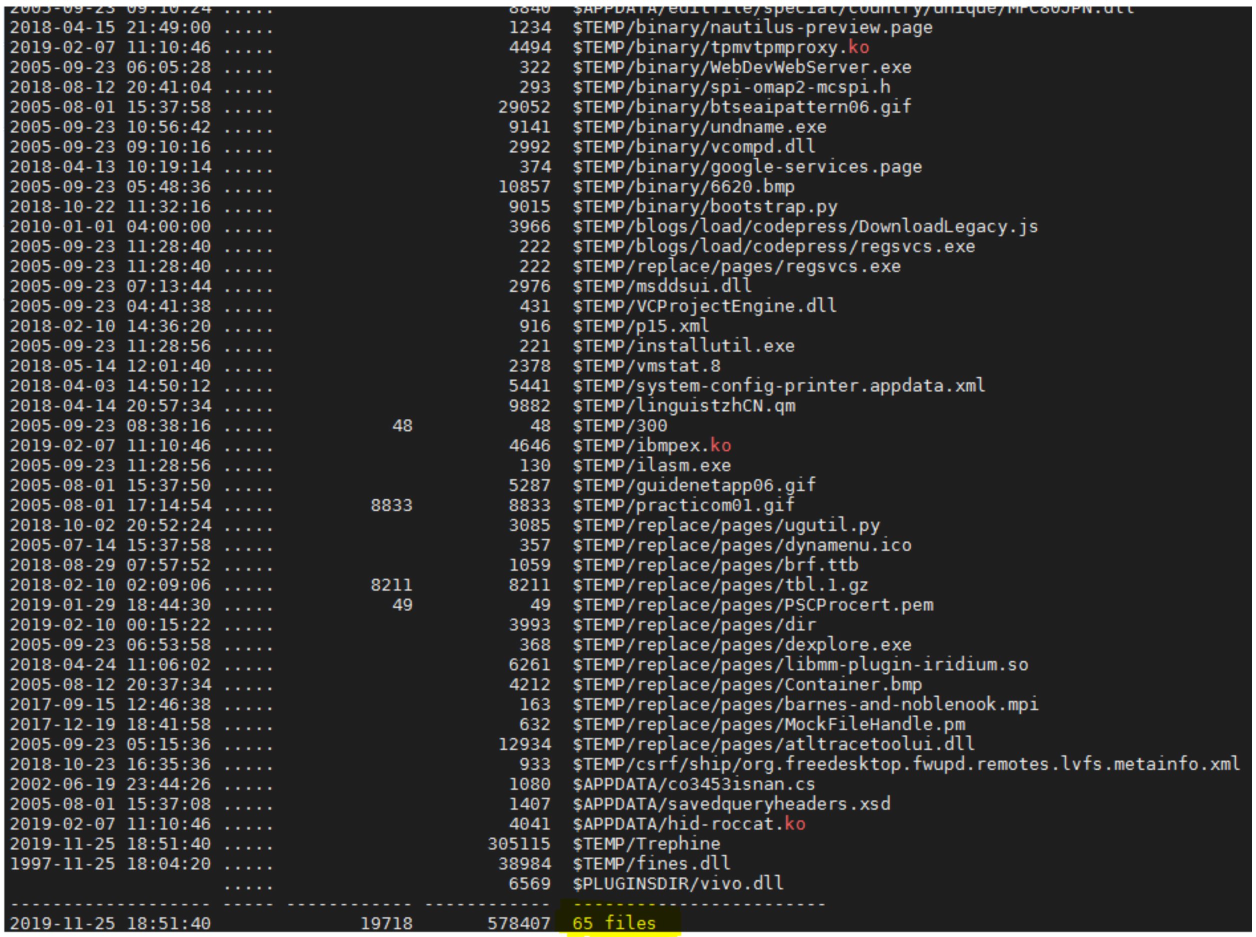
Used emails on Campaign 1

The following graph shows the relation and infection chain for campaign 1 (based on available data on VT)



## Campaign 2 (November 25, 2019 to November 26, 2019)

These are the dropped junk files for all NSIS installers that belong to campaign 2:

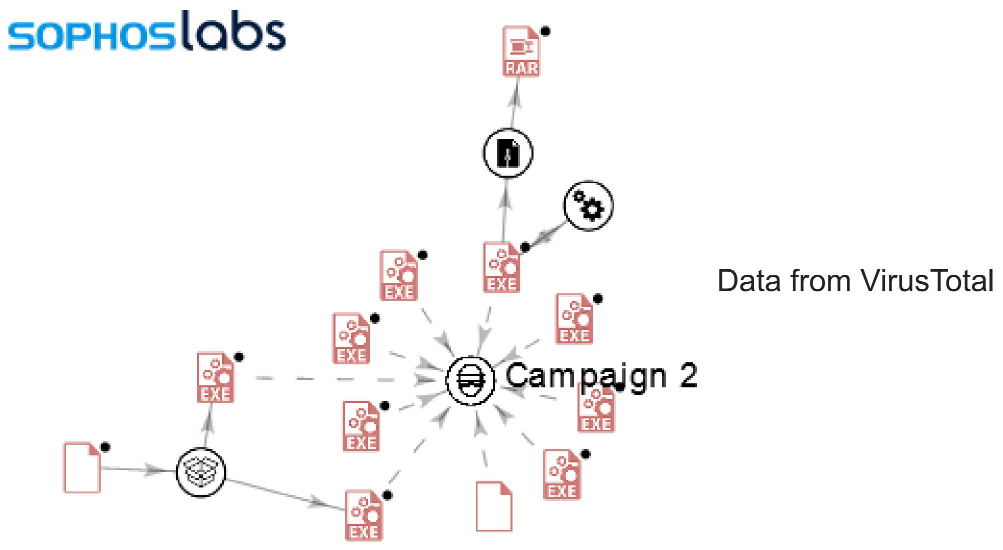

```
2005-09-23 09:10:24 .....                     8840  $APPDATA/editfile/special/country/unique/MFC80JPN.dll
2018-04-15 21:49:00 .....                     1234  $TEMP/binary/nautilus-preview.page
2019-02-07 11:10:46 .....                     4494  $TEMP/binary/tpmvtpmproxy.ko
2005-09-23 06:05:28 .....                      322  $TEMP/binary/WebDevWebServer.exe
2018-08-12 20:41:04 .....                      293  $TEMP/binary/spi-omap2-mcspi.h
2005-08-01 15:37:58 .....                    29052  $TEMP/binary/btseaipattern06.gif
2005-09-23 10:56:42 .....                     9141  $TEMP/binary/undname.exe
2005-09-23 09:10:16 .....                     2992  $TEMP/binary/vcompd.dll
2018-04-13 10:19:14 .....                      374  $TEMP/binary/google-services.page
2005-09-23 05:48:36 .....                    10857  $TEMP/binary/6620.bmp
2018-10-22 11:32:16 .....                     9015  $TEMP/binary/bootstrap.py
2010-01-01 04:00:00 .....                     3966  $TEMP/blogs/load/codepress/DownloadLegacy.js
2005-09-23 11:28:40 .....                      222  $TEMP/blogs/load/codepress/regsvcs.exe
2005-09-23 11:28:40 .....                      222  $TEMP/replace/pages/regsvcs.exe
2005-09-23 07:13:44 .....                     2976  $TEMP/msddsui.dll
2005-09-23 04:41:38 .....                      431  $TEMP/VCProjectEngine.dll
2018-02-10 14:36:20 .....                      916  $TEMP/p15.xml
2005-09-23 11:28:56 .....                      221  $TEMP/installutil.exe
2018-05-14 12:01:40 .....                     2378  $TEMP/vmstat.8
2018-04-03 14:50:12 .....                     5441  $TEMP/system-config-printer.appdata.xml
2018-04-14 20:57:34 .....                     9882  $TEMP/linguistzhCN.qm
2005-09-23 08:38:16 .....           48           48  $TEMP/300
2019-02-07 11:10:46 .....                     4646  $TEMP/ibmpex.ko
2005-09-23 11:28:56 .....                      130  $TEMP/ilasm.exe
2005-08-01 15:37:50 .....                     5287  $TEMP/guidenetapp06.gif
2005-08-01 17:14:54 .....         8833         8833  $TEMP/practicom01.gif
2018-10-02 20:52:24 .....                     3085  $TEMP/replace/pages/ugutil.py
2005-07-14 15:37:58 .....                      357  $TEMP/replace/pages/dynamenu.ico
2018-08-29 07:57:52 .....                     1059  $TEMP/replace/pages/brf.ttb
2018-02-10 02:09:06 .....         8211         8211  $TEMP/replace/pages/tbl.1.gz
2019-01-29 18:44:30 .....           49           49  $TEMP/replace/pages/PSCProcert.pem
2019-02-10 00:15:22 .....                     3993  $TEMP/replace/pages/dir
2005-09-23 06:53:58 .....                      368  $TEMP/replace/pages/dexplore.exe
2018-04-24 11:06:02 .....                     6261  $TEMP/replace/pages/libmm-plugin-iridium.so
2005-08-12 20:37:34 .....                     4212  $TEMP/replace/pages/Container.bmp
2017-09-15 12:46:38 .....                      163  $TEMP/replace/pages/barnes-and-noblenook.mpi
2017-12-19 18:41:58 .....                      632  $TEMP/replace/pages/MockFileHandle.pm
2005-09-23 05:15:36 .....                    12934  $TEMP/replace/pages/atltracetoolui.dll
2018-10-23 16:35:36 .....                      933  $TEMP/csrf/ship/org.freedesktop.fwupd.remotes.lvfs.metainfo.xml
2002-06-19 23:44:26 .....                     1080  $APPDATA/co3453isnan.cs
2005-08-01 15:37:08 .....                     1407  $APPDATA/savedqueryheaders.xsd
2019-02-07 11:10:46 .....                     4041  $APPDATA/hid-roccat.ko
2019-11-25 18:51:40 .....                   305115  $TEMP/Trephine
1997-11-25 18:04:20 .....                    38984  $TEMP/fines.dll
                    .....                     6569  $PLUGINSDIR/vivo.dll
------------------- ----- ------------ ------------  ------------------------
2019-11-25 18:51:40              19718       578407  65 files
```

Output of 7z after list the files contained on a sample that belongs to campaign 2

Some of the payloads identified for campaign 2 on a first triage included the following:

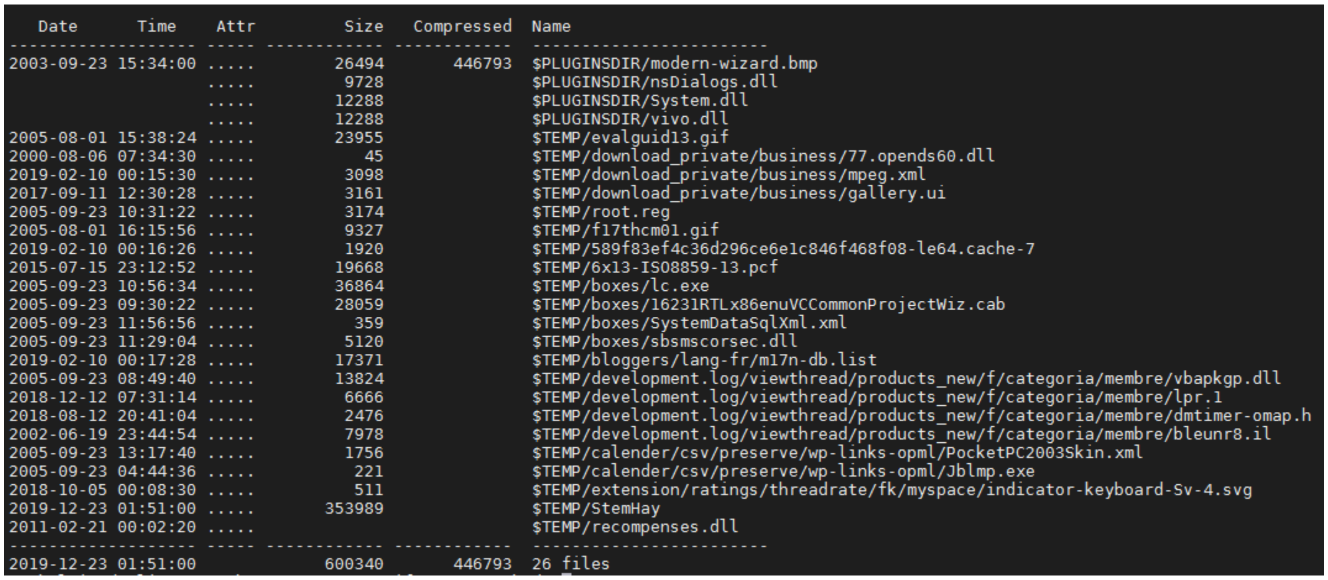
| TYPE | OBSERVED PAYLOADS | OBSERVED PAYLOADS C&C DOMAINS |
|---|---|---|
| Info Stealer | Betabot | negrodesigns.ga |
| Info Stealer | Formbook | czxpkj.com/c206<br>pupilfy.com<br>cbespania.info/c206<br>jevmod.com/c206 |
| RAT | Bladabindi | tucson1989.duckdns.org<br>pedrobedoya201904.duckdns.org |
| RAT | Blackrat | 79.134.225.97:1982 |
| RAT | Remcos | cashout2018.ddns.de |

We found no emails for this campaign, so we were unable to map its intended targets. The graph below shows the relationship between the similar payloads.



**Campaign 4 (December 20, 2019 to December 31, 2019)**

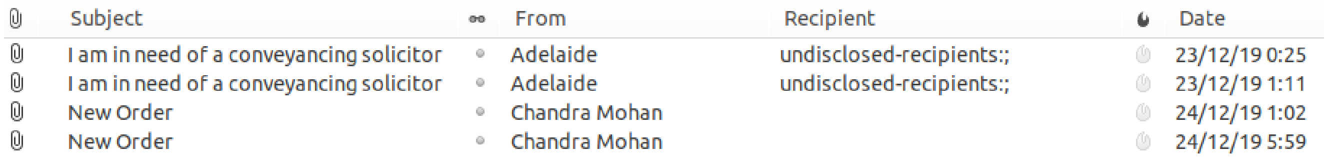
These are the dropped junk files for all NSIS installers that belong to campaign 4:



```
   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
2003-09-23 15:34:00 .....        26494       446793  $PLUGINSDIR/modern-wizard.bmp
                    .....         9728               $PLUGINSDIR/nsDialogs.dll
                    .....        12288               $PLUGINSDIR/System.dll
                    .....        12288               $PLUGINSDIR/vivo.dll
2005-08-01 15:38:24 .....        23955               $TEMP/evalguid13.gif
2000-08-06 07:34:30 .....           45               $TEMP/download_private/business/77.opends60.dll
2019-02-10 00:15:30 .....         3098               $TEMP/download_private/business/mpeg.xml
2017-09-11 12:30:28 .....         3161               $TEMP/download_private/business/gallery.ui
2005-09-23 10:31:22 .....         3174               $TEMP/root.reg
2005-08-01 16:15:56 .....         9327               $TEMP/f17thcm01.gif
2019-02-10 00:16:26 .....         1920               $TEMP/589f83ef4c36d296ce6e1c846f468f08-le64.cache-7
2015-07-15 23:12:52 .....        19668               $TEMP/6x13-ISO8859-13.pcf
2005-09-23 10:56:34 .....        36864               $TEMP/boxes/lc.exe
2005-09-23 09:30:22 .....        28059               $TEMP/boxes/16231RTLx86enuVCCommonProjectWiz.cab
2005-09-23 11:56:56 .....          359               $TEMP/boxes/SystemDataSqlXml.xml
2005-09-23 11:29:04 .....         5120               $TEMP/boxes/sbsmscorsec.dll
2019-02-10 00:17:28 .....        17371               $TEMP/bloggers/lang-fr/m17n-db.list
2005-09-23 08:49:40 .....        13824               $TEMP/development.log/viewthread/products_new/f/categoria/membre/vbapkgp.dll
2018-12-12 07:31:14 .....         6666               $TEMP/development.log/viewthread/products_new/f/categoria/membre/lpr.1
2018-08-12 20:41:04 .....         2476               $TEMP/development.log/viewthread/products_new/f/categoria/membre/dmtimer-omap.h
2002-06-19 23:44:54 .....         7978               $TEMP/development.log/viewthread/products_new/f/categoria/membre/bleunr8.il
2005-09-23 13:17:40 .....         1756               $TEMP/calender/csv/preserve/wp-links-opml/PocketPC2003Skin.xml
2005-09-23 04:44:36 .....          221               $TEMP/calender/csv/preserve/wp-links-opml/Jblmp.exe
2018-10-05 00:08:30 .....          511               $TEMP/extension/ratings/threadrate/fk/myspace/indicator-keyboard-Sv-4.svg
2019-12-23 01:51:00 .....       353989               $TEMP/StemHay
2011-02-21 00:02:20 .....                            $TEMP/recompenses.dll
------------------- ----- ------------ ------------  ------------------------
2019-12-23 01:51:00             600340       446793  26 files
```

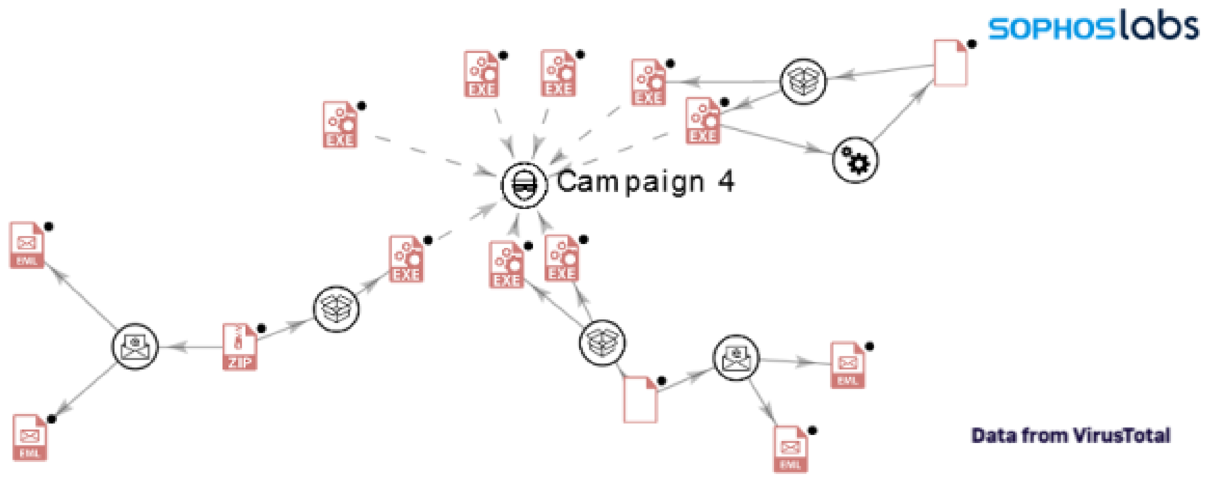
Some of the payloads observed associated with campaign 4 included:

| TYPE | OBSERVED PAYLOADS | OBSERVED PAYLOADS C&C DOMAINS |
|---|---|---|
| Info Stealer | Betabot | pitchstak.ga |
| Info Stealer | Lokibot | pitchstak.ga |
| Info Stealer | Formbook | slashoff.com/c208<br>sofisleep.com/c208<br>jinshasoft.com/c208<br>binzom.com/c208 |
| RAT | Netwire | 79.134.225.97:2556 |
| RAT | AgentTesla | mail.newmedicacare.com |



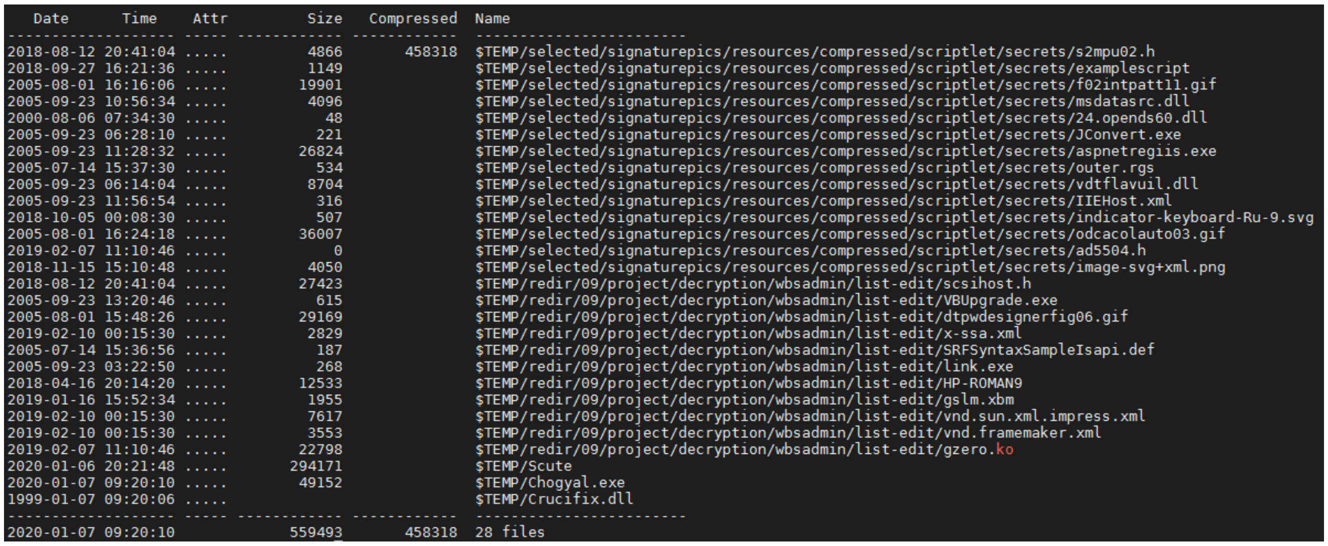
| Subject | From | Recipient | Date |
|---|---|---|---|
| I am in need of a conveyancing solicitor | Adelaide | undisclosed-recipients:; | 23/12/19 0:25 |
| I am in need of a conveyancing solicitor | Adelaide | undisclosed-recipients:; | 23/12/19 1:11 |
| New Order | Chandra Mohan | | 24/12/19 1:02 |
| New Order | Chandra Mohan | | 24/12/19 5:59 |

Emails collected from VirusTotal tied to campaign 4.



## Campaign 5 (January 3, 2020 to January 8, 2020)

These are the dropped junk files for all NSIS installers that belong to campaign 5:



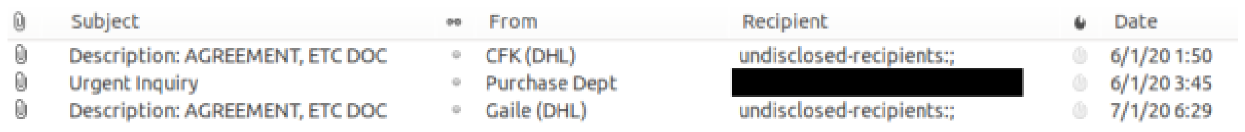
```
   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
2018-08-12 20:41:04 .....         4866       458318  $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/s2mpu02.h
2018-09-27 16:21:36 .....         1149               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/examplescript
2005-08-01 16:16:06 .....        19901               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/f02intpatt11.gif
2005-09-23 10:56:34 .....         4096               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/msdatasrc.dll
2000-08-06 07:34:30 .....           48               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/24.opends60.dll
2005-09-23 06:28:10 .....          221               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/JConvert.exe
2005-09-23 11:28:32 .....        26824               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/aspnetregiis.exe
2005-07-14 15:37:30 .....          534               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/outer.rgs
2005-09-23 06:14:04 .....         8704               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/vdtflavuil.dll
2005-09-23 11:56:54 .....          316               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/IIEHost.xml
2018-10-05 00:08:30 .....          507               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/indicator-keyboard-Ru-9.svg
2005-08-01 16:24:18 .....        36007               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/odcacolauto03.gif
2019-02-07 11:10:46 .....            0               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/ad5504.h
2018-11-15 15:10:48 .....         4050               $TEMP/selected/signaturepics/resources/compressed/scriptlet/secrets/image-svg+xml.png
2018-08-12 20:41:04 .....        27423               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/scsihost.h
2005-09-23 13:20:46 .....          615               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/VBUpgrade.exe
2005-08-01 15:48:26 .....        29169               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/dtpwdesignerfig06.gif
2019-02-10 00:15:30 .....         2829               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/x-ssa.xml
2005-07-14 15:36:56 .....          187               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/SRFSyntaxSampleIsapi.def
2005-09-23 03:22:50 .....          268               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/link.exe
2018-04-16 20:14:20 .....        12533               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/HP-ROMAN9
2019-01-16 15:52:34 .....         1955               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/gslm.xbm
2019-02-10 00:15:30 .....         7617               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/vnd.sun.xml.impress.xml
2019-02-10 00:15:30 .....         3553               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/vnd.framemaker.xml
2019-02-07 11:10:46 .....        22798               $TEMP/redir/09/project/decryption/wbsadmin/list-edit/gzero.ko
2020-01-06 20:21:48 .....       294171               $TEMP/Scute
2020-01-07 09:20:10 .....        49152               $TEMP/Chogyal.exe
1999-01-07 09:20:06 .....                            $TEMP/Crucifix.dll
------------------- ----- ------------ ------------  ------------------------
2020-01-07 09:20:10             559493       458318  28 files
```

Output of 7z after list the files contained on a sample that belongs to campaign 5

Some of the payloads of campaign 5:

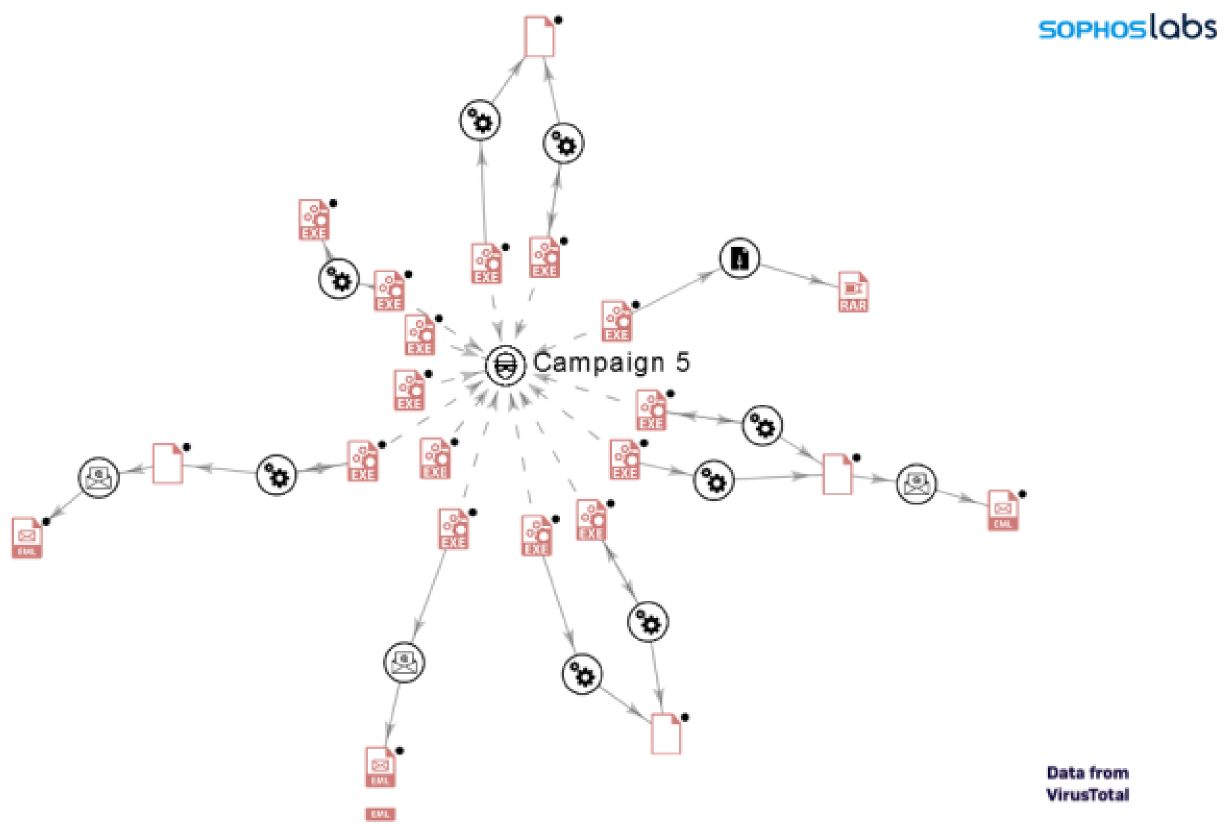
| TYPE | OBSERVED PAYLOADS | OBSERVED PAYLOADS C&C DOMAINS |
|---|---|---|
| Info Stealer | Betabot | pitchstak.ga |
| Info Stealer | Lokibot | pitchstak.ga |
| Info Stealer | Formbook | binzom.com/c208<br>bywebhost.com/c208<br>jinshasoft.com/c208 |
| RAT | Netwire | 79.134.225.97:2556 |
| RAT | AgentTesla | mail.arkazo.com<br>mail.alhilaly-group.com |

Sample emails we collected tied to campaign 5:



| Subject | From | Recipient | Date |
|---|---|---|---|
| Description: AGREEMENT, ETC DOC | CFK (DHL) | undisclosed-recipients:; | 6/1/20 1:50 |
| Urgent Inquiry | Purchase Dept | [redacted] | 6/1/20 3:45 |
| Description: AGREEMENT, ETC DOC | Gaile (DHL) | undisclosed-recipients:; | 7/1/20 6:29 |

The following graph shows the relation and infection chain for campaign 5 (based on available data on VT)



## Profiling the threat actor

Looking across all the campaigns we discovered during this analysis, we saw frequent duplications in C&C infrastructure, as shown in the table summarizing the campaigns below:

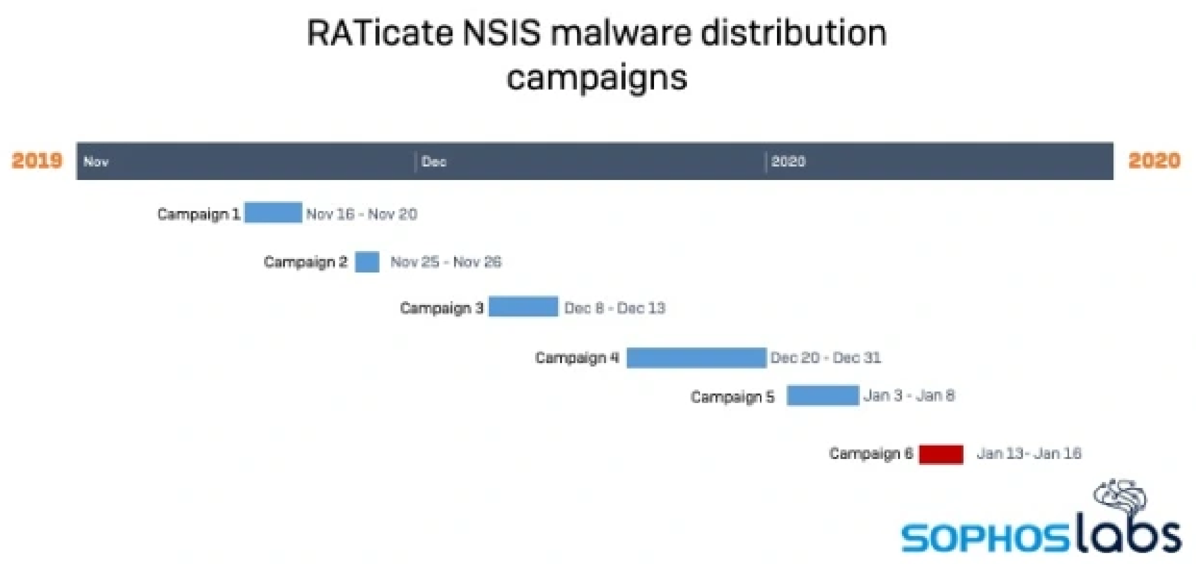
| CAMPAIGN | DATES | TYPE | OBSERVED PAYLOADS | OBSERVED PAYLOADS C&C DOMAINS |
|---|---|---|---|---|
| 1 | 2019-11-16/2019-11-20 | Info Stealer | Betabot | negrodesigns.ga gelcursot.top webxpo.ga |
| Info Stealer | Lokibot | gelcursot.top | | |

| | | | | |
|---|---|---|---|---|
| Info Stealer | Formbook | cbespania.info/c206<br>conrak.net/c206<br>coxemen.com/c206<br>dachfix.com/c206<br>hypnose-beziers.com/c206<br>jevmod.com/c206<br>lighthouse-campus24.com/c206<br>oleum.gmbh/c206<br>pupilfy.com/c206<br>tellpizzqhut.com/c206<br>terenium.com/c206<br>vibe.restaurant/c206<br>yamatobb.com/c206<br>yncits89.com/p0x<br>ratokasutka.com/p0x<br>miscov.com/p0x | | |
| RAT | Netwire | 79.134.225.11:1199 | | |
| 2 | 2019-11-25/2019-11-26 | Info Stealer | Betabot | negrodesigns.ga |
| Info Stealer | Formbook | czxpkj.com/c206<br>pupilfy.com<br>cbespania.info/c206<br>jevmod.com/c206 | | |
| RAT | Bladabindi | tucson1989.duckdns.org<br>pedrobedoya201904.duckdns.org | | |
| RAT | Blackrat | 79.134.225.97:1982 | | |
| RAT | Remcos | cashout2018.ddns.de | | |
| 3 | 2019-12-08/2019-12-13 | Info Stealer | Betabot | stngpetty.ga<br>allenservice.ga<br>gelcursot.top |
| Info Stealer | Formbook | ef-oh.com/c208<br>odoyo.net/c208<br>hearee.com/c208<br>binzom.com/c208<br>pizzans.com/c208<br>phochain.com/sa<br>rdrfi.com/sa/<br>skylod.com/sa<br>hsctsu.com/sa | | |
| Info Stealer | Lokibot | gelcursot.top | | |
| RAT | Netwire | 79.134.225.97:2556 | | |

| | | | | |
|---|---|---|---|---|
| RAT | AgentTesla | mail.newmedicacare.com<br>mail.jrdigitalstore.com<br>mail.koyo.com.my<br>mail.qoa.com.my<br>mail.sedirectory.com.my | | |
| 4 | 2019-12-20/2019-12-31 | Info Stealer | Betabot | pitchstak.ga |
| Info Stealer | Lokibot | pitchstak.ga | | |
| Info Stealer | Formbook | slashoff.com/c208<br>sofisleep.com/c208<br>jinshasoft.com/c208<br>binzom.com/c208 | | |
| RAT | Netwire | 79.134.225.97:2556 | | |
| RAT | AgentTesla | | | |
| 5 | 2020-01-03/2020-01-08 | Info Stealer | Betabot | pitchstak.ga |
| Info Stealer | Lokibot | pitchstak.ga | | |
| Info Stealer | Formbook | binzom.com/c208<br>bywebhost.com/c208<br>jinshasoft.com/c208 | | |
| RAT | Netwire | 79.134.225.97:2556 | | |
| RAT | AgentTesla | mail.arkazo.com<br>mail.alhilaly-group.com | | |

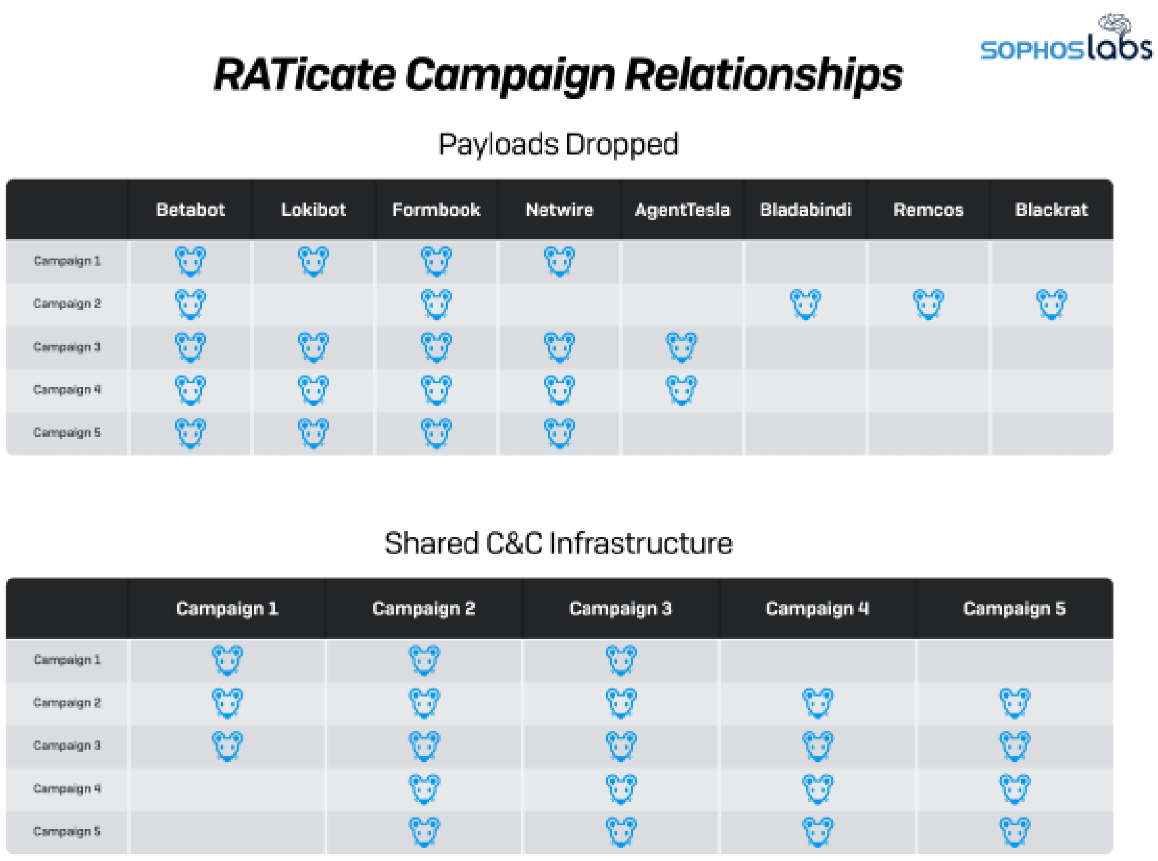
We also found that some of the different payloads from each campaign (mostly Betabot, Lokibot, AgentTesla and Formbook) shared the same C&C. This suggests that the same actor/group was managing the web panels behind these malware campaigns.

There was also a distinct clustering of the campaign timelines—there was never any overlap between them, suggesting that they were operated serially by the same threat actors (including a sixth campaign we observed, to be covered in our next report):



These campaigns didn't just share command and control infrastructure across different payloads within the same campaign. Some of the infrastructure was also shared across multiple campaigns, which also suggests the same actor was involved across all of them.

The following tables show some interesting relations between campaigns.



## RATicate Campaign Relationships

SOPHOSlabs

### Payloads Dropped

| | Betabot | Lokibot | Formbook | Netwire | AgentTesla | Bladabindi | Remcos | Blackrat |
|---|---|---|---|---|---|---|---|---|
| Campaign 1 | ✓ | ✓ | ✓ | ✓ | | | | |
| Campaign 2 | ✓ | | ✓ | | | ✓ | ✓ | ✓ |
| Campaign 3 | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| Campaign 4 | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| Campaign 5 | ✓ | ✓ | ✓ | ✓ | | | | |

### Shared C&C Infrastructure

| | Campaign 1 | Campaign 2 | Campaign 3 | Campaign 4 | Campaign 5 |
|---|---|---|---|---|---|
| Campaign 1 | ✓ | ✓ | ✓ | | |
| Campaign 2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| Campaign 3 | ✓ | ✓ | ✓ | ✓ | ✓ |
| Campaign 4 | | ✓ | ✓ | ✓ | ✓ |
| Campaign 5 | | ✓ | ✓ | ✓ | ✓ |

## Targeting and motivation

Based on the payloads used by RATicate, it's clear that the campaigns run by the group are intended to gain access to and control of computers on the targeted companies' networks. The targets identified from the collected emails sent by these campaigns include:

- An electrical equipment manufacturer in Romania;
- A Kuwaiti construction services and engineering company;
- A Korean internet company;
- A Korean investment firm;
- A British building supply manufacturer;
- A Korean medical news publication;
- A Korean telecommunications and electrical cable manufacturer;
- A Swiss publishing equipment manufacturer;
- A Japanese courier and transportation company.

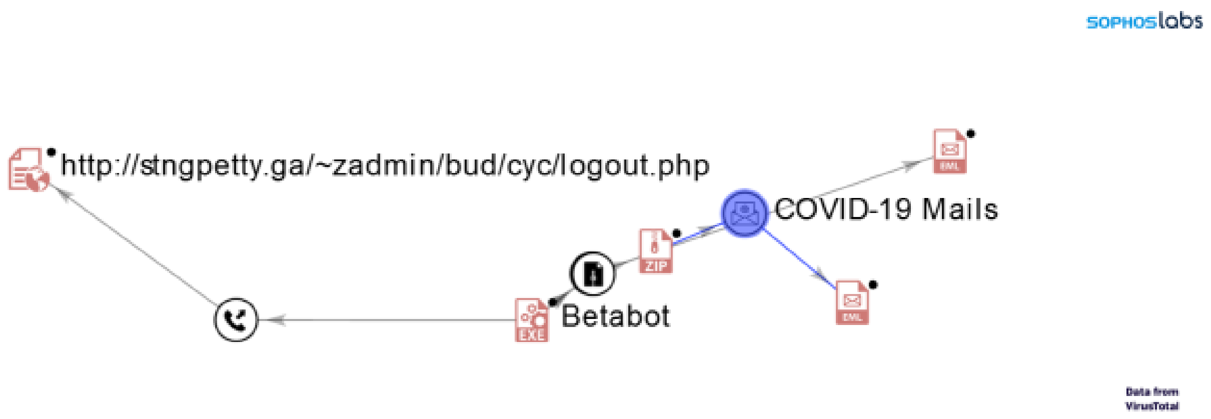
We know that the targets overlapped on at least two campaigns: Campaign 1 and 2 both targeted the electrical equipment manufacturer. There are likely more targets that were common across multiple campaigns (we looked only at publicly-available data from VirusTotal, and have not explored non-public databases). And many (but not all) of the companies that have been targeted-up are related to critical infrastructure.

We've detected one more recent campaign using these NSIS installers (from January 13-16). However, as we've continued to research this actor group, we've been studying other campaigns that we believe are being run by the the same actor—and we believe that since January, the actor has moved to using other loaders and packers.

One of those campaigns is an email campaign we detected in March that uses the COVID-19 global pandemic as a lure to get victims to open the payload. The most recent detected samples are delivered with a variety of Visual Basic loaders —including the Guloader malware dropper discovered by Proofpoint on December 2019.

We believe these campaigns are run by the same actor fro a number of reasons:

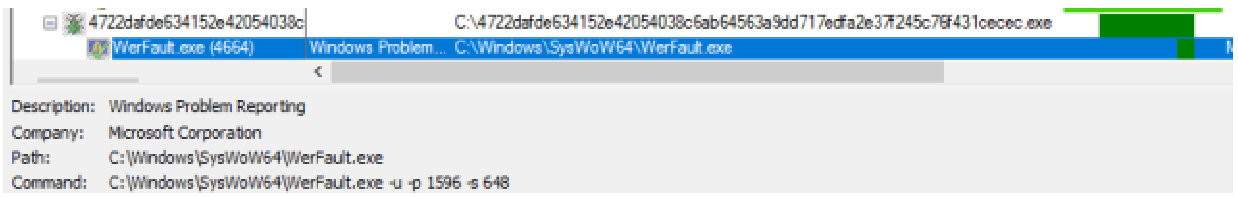
- The email targets the same companies seen in previous campaigns.
- Some of the detected payloads are Betabot and Lokibot, families observed in previous campaigns.
- This Betabot's C&C are similar to observed in these previous campaigns—it uses same domain as Campaign 3 for Betabot (**stngpetty[.]ga**) and uses a similar path (/~zadmin/{NAME1}/{NAME2}/logout.php).



Based on their behavior, we're unsure of whether the RATicate group is focused on corporate espionage or is simply acting as a malware-as-a-service provider to other actors. It could simply be that they are dropping malware on targeted companies in order to provide paid access to others, or are using InfoStealer and RAT malware as part of a larger malware distribution effort. We continue to analyze the new attacks and hope to get deeper insight into their motivations.

## Anti-sandboxing by dumb luck

During our analysis of the first RATicate sample, we discovered that the Shellcode3 dropped by the installer uses a number of interesting techniques to make it difficult to analyze API calls, as well as a number of anti-debugging tricks to further hinder analysis. But we also found a strange behavior in these samples: if the sample is executed with its SHA256 hash as its filename, the program will crash.

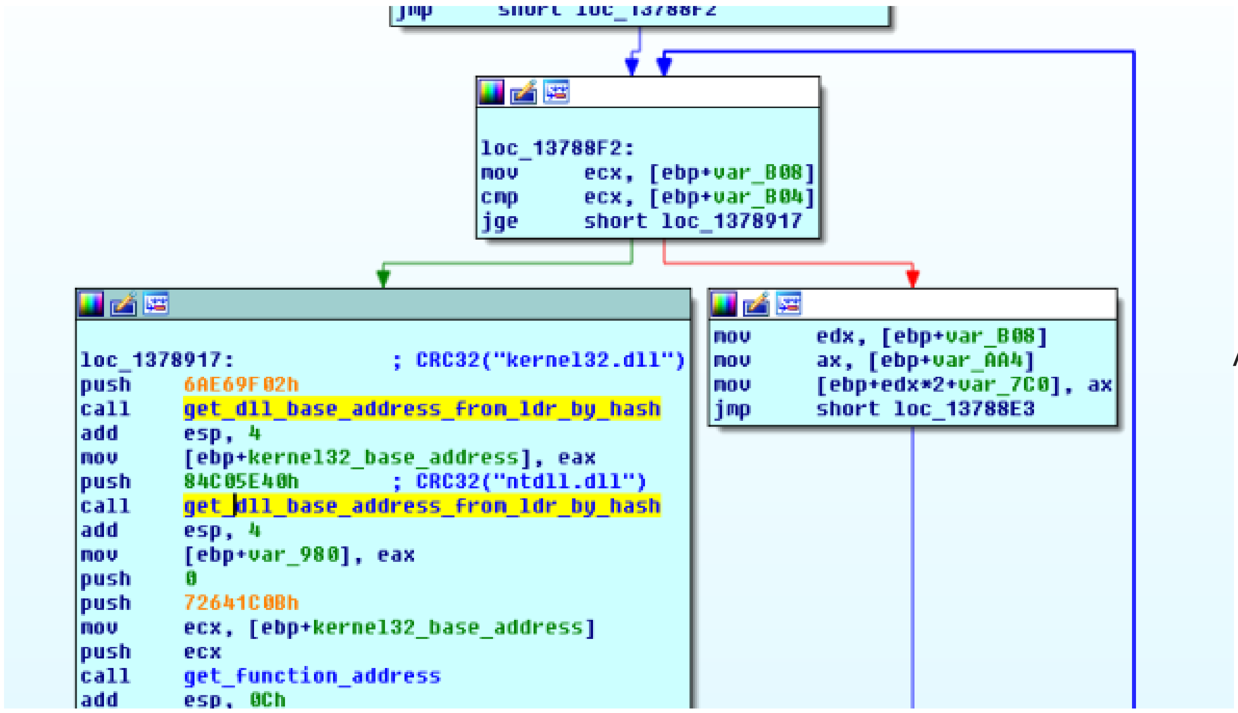


```
4722dafde634152e42054038c          C:\4722dafde634152e42054038c6ab64563a9dd717edfa2e37f245c76f431cecec.exe
  WerFault.exe (4664)  Windows Problem...  C:\Windows\SysWoW64\WerFault.exe

Description:  Windows Problem Reporting
Company:      Microsoft Corporation
Path:         C:\Windows\SysWoW64\WerFault.exe
Command:      C:\Windows\SysWoW64\WerFault.exe -u -p 1596 -s 648
```

Analyzed sample crashing when the file name size is its SHA256 hash.

This sort of behavior might be seen as an anti-analysis trick. Since sandboxes usually run the samples with their hash as a filename, this technique could avoid the execution of the payload in sandbox environments. But in this case, the behavior is actually because of a bug in the code.

The error occurs during the execution of shellcode 3.

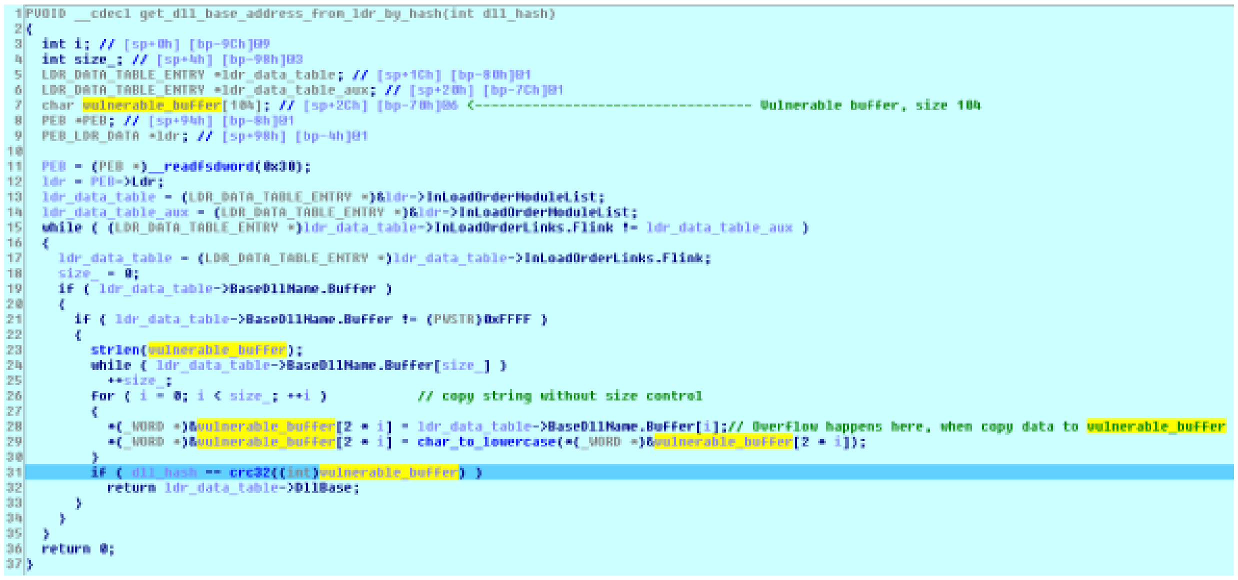


```
jmp     short loc_13788F2

loc_13788F2:
mov     ecx, [ebp+var_B08]
cmp     ecx, [ebp+var_B04]
jge     short loc_1378917

loc_1378917:            ; CRC32("kernel32.dll")
push    6AE69F02h
call    get_dll_base_address_from_ldr_by_hash
add     esp, 4
mov     [ebp+kernel32_base_address], eax
push    84C05E40h       ; CRC32("ntdll.dll")
call    get_dll_base_address_from_ldr_by_hash
add     esp, 4
mov     [ebp+var_980], eax
push    0
push    72641C0Bh
mov     ecx, [ebp+kernel32_base_address]
push    ecx
call    get_function_address
add     esp, 0Ch

mov     edx, [ebp+var_B08]
mov     ax, [ebp+var_AA4]
mov     [ebp+edx*2+var_7C0], ax
jmp     short loc_13788E3
```

A

snippet of shellcode 3's code as viewed in IDA Pro.

Shellcode3 uses a known technique to get the address of loaded modules (such as libraries and the executable's image itself) by searching against the LDR_DATA_TABLE_ENTRY data structure within the Windows operating system's Process Environment Block (PEB). The LDR structure contains information that includes the names and addresses of loaded modules. The shell code checks this structure against hashes of the desired function names, providing a silent way to dynamically resolve the memory address of a function to be called.



```
PVOID __cdecl get_dll_base_address_from_ldr_by_hash(int dll_hash)
{
  int i; // [sp+0h] [bp-9Ch]@9
  int size_; // [sp+4h] [bp-98h]@3
  LDR_DATA_TABLE_ENTRY *ldr_data_table; // [sp+1Ch] [bp-80h]@1
  LDR_DATA_TABLE_ENTRY *ldr_data_table_aux; // [sp+20h] [bp-7Ch]@1
  char vulnerable_buffer[104]; // [sp+2Ch] [bp-70h]@6 <------------------------------------ Vulnerable buffer, size 104
  PEB *PEB; // [sp+94h] [bp-8h]@1
  PEB_LDR_DATA *ldr; // [sp+98h] [bp-4h]@1

  PEB = (PEB *)__readfsdword(0x30);
  ldr = PEB->Ldr;
  ldr_data_table = (LDR_DATA_TABLE_ENTRY *)&ldr->InLoadOrderModuleList;
  ldr_data_table_aux = (LDR_DATA_TABLE_ENTRY *)&ldr->InLoadOrderModuleList;
  while ( (LDR_DATA_TABLE_ENTRY *)ldr_data_table->InLoadOrderLinks.Flink != ldr_data_table_aux )
  {
    ldr_data_table = (LDR_DATA_TABLE_ENTRY *)ldr_data_table->InLoadOrderLinks.Flink;
    size_ = 0;
    if ( ldr_data_table->BaseDllName.Buffer )
    {
      if ( ldr_data_table->BaseDllName.Buffer != (PWSTR)0xFFFF )
      {
        strlen(vulnerable_buffer);
        while ( ldr_data_table->BaseDllName.Buffer[size_] )
          ++size_;
        for ( i = 0; i < size_; ++i )               // copy string without size control
        {
          *(_WORD *)&vulnerable_buffer[2 * i] = ldr_data_table->BaseDllName.Buffer[i];// Overflow happens here, when copy data to vulnerable_buffer
          *(_WORD *)&vulnerable_buffer[2 * i] = char_to_lowercase(*(_WORD *)&vulnerable_buffer[2 * i]);
        }
        if ( dll_hash == crc32((int)vulnerable_buffer) )
          return ldr_data_table->DllBase;
      }
    }
  }
  return 0;
}
```

Shellcode 3 function to get module base addresses based on LDR_DATA_TABLE, which contains a bug that causes the sample to crash.

This feature is implemented in the code's *get_dll_base_addres_from_ldr_by_hash(dll_hash)* function, which is where the crash happens. The function walks through the LDR data structure, hashing the names of loaded modules in order to try to match the hash passed as argument.

The function puts the contents of ldr_data_table->BaseDllName.Buffer into vulnerable_buffer in order to convert the ANSI string to a UNICODE string.

But since the size of the vulnerable_buffer string is 104 and it’s storing a Unicode string, which means its size limit is really just 52 ANSI characters. The consequences of that are if the filename has a length of 53 or more characters, a buffer overflow will occur. To make the program crash, you simply need to give the sample a 57-character-long filename (such as “this_is_57_length_filename_in_order_to_do_a_crash_PoC.exe”).

Once analyzed, we determined this was a programming error, rather than an anti-sandbox technique.

## Indicators of Compromise (IOCs)

Hashes for the files associated with the RATicate campaigns can be found on SophosLabs’ GitHub here.