IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF GEORGIA
ATLANTA DIVISION

MICROSOFT CORPORATION, a Washington corporation, FS-ISAC, INC., a Delaware corporation and HEALTH-ISAC, INC., a Florida corporation, Plaintiffs, v. DENIS MALIKOV AND JOHN DOES 1-7, Defendants.	)	Case No.: 1:22-cv-1328-MHC

DECLARATION OF TOMAS PROCHAZKA IN SUPPORT OF APPLICATION FOR TEMPORARY RESTRAINING ORDER AND ORDER TO SHOW CAUSE RE PRELIMINARY INJUNCTION

I, Tomas Prochazka, declare as follows:

1. I work as a Senior Detection Engineer for ESET spol. s.r.o. ("ESET"). I make this declaration in support of Application or Temporary Restraining Order And Order To Show Cause Re Preliminary Injunction. I make this declaration of my own personal knowledge, and, if called as a witness, I could and would testify competently to the truth of the matters set forth herein.

2. ESET is an Internet security company that investigates malicious

threat actors, including the present threat known as ZLoader, and offers anti-virus products. In my role at ESET, I investigate security threats, including botnets, track their activity and build a defense to protect millions of ESET product end users against these threats.

3. My role at ESET has provided me an in-depth insight into how malware authors deploy and utilize online threats for their monetary gain. Prior to joining ESET, I was working in similar positions for anti-virus vendors AVG Technologies and AVAST Software.

Overview of ZLoader malware

4. I have carried out research regarding the ZLoader botnet. During the investigation I reverse engineered the ZLoader malware and monitored threat activity associated with the ZLoader botnet. From that work, I am familiar with the operation and structure of ZLoader and conclude that the core functionalities of ZLoader are directed at stealing sensitive data from the infected machine. As of February 2022, the main components of this threat are primarily detected by ESET products as Win32/Zbot.ADI and Win64/Zbot.Q.

5. My investigation has confirmed that ZLoader has a modular architecture that enables it to carry out a variety of illegal activities, including:

a. Stealing of sensitive data from the infected machine by accessing its data or through a Man-in-the-browser attack to websites such as,

Microsoft's login page, various financial institutions and other online platforms;

b. Remote control of the compromised system; and

c. Delivering and executing of other malware.

6. ZLoader causes harm to victim end-users who are targeted by the ZLoader operators and whose computers are infected with the ZLoader malware. The ZLoader malware enables the perpetrators to infect victim computers, steal sensitive credentials and identifying information from the users of those computers, and potentially use the stolen credentials to illegally transfer money from the victims' financial accounts.

7. Through my experience with remediation of malware attacks, I am aware that users whose computers are infected with ZLoader face complex challenges determining how to remove the ZLoader malware from their computers. In my experience, users having to manage this process experience frustration and confusion, and must invest time in attempting to mitigate the injury. Many will not realize that they are infected, whilst the consequences of being infected are severe as the malware is amongst others designed to steal information, including login credentials for online banking services. The perpetrators can also use ZLoader to install additional malware and find ways to spread more malware to other machines connected to the same local network.

Technical Analysis of ZLoader

Infecting victims with ZLoader

8. ZLoader operators used number of techniques to deliver malware to victims' systems. My investigation revealed that the most common technique used lately was luring victims into visiting bogus sites pretending to be either download sites for legitimate software or an adult site requesting to download update for Java browser plug-in. In both cases, malicious component is downloaded instead of the expected legitimate software. This malicious component is then responsible to deliver ZLoader malware to the user's system.

9. Once ZLoader's initial component is executed, it runs the Microsoft process "msiexec.exe", a software used to unpack and run MSI files, and injects ZLoader's code into this process by modifying memory of the msiexec.exe process. ZLoader does not use msiexec.exe for unpacking MSI files and makes no use of any of the functionality of msiexec.exe. Instead ZLoader uses the misiexec.exe to disguise ZLoader's functionality such that users believe genuine Microsoft software described as "Windows installer" is running instead of ZLoader as shown in figure 1. ZLoader's functionality includes, but is not limited to:

a. Communicating with ZLoader's Command and Control (C2) servers.

b. Collecting personal data from victim's system and sending these to ZLoader's C2 servers.

c. Downloading additional ZLoader's components and possibly other malware

d. Setting up a proxy server to support Man-in-the-Browser attacks (as defined in the declarations of Christopher Coy and Rodel Finones filed concurrently).

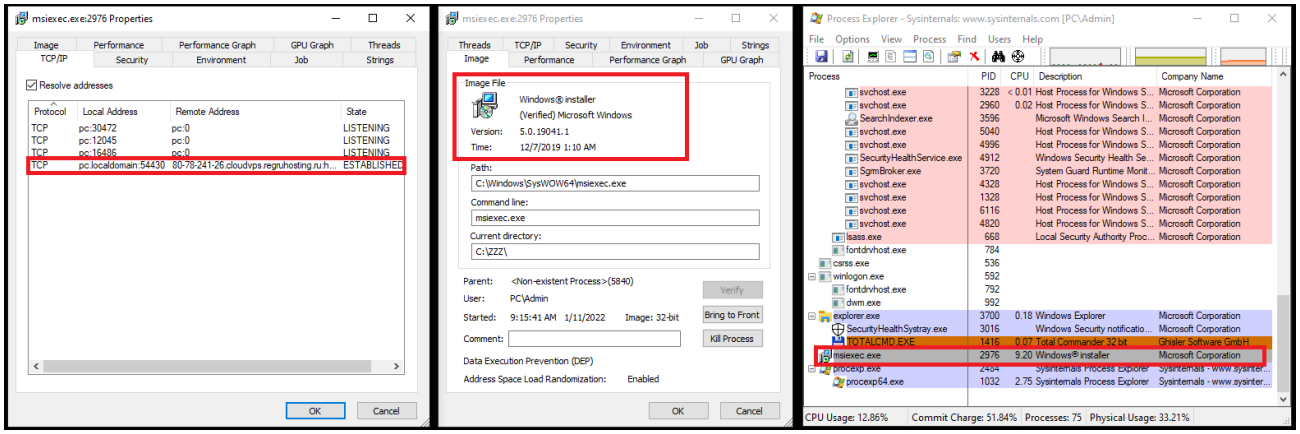

Figure 1: Legitimate msiexec.exe process injected by ZLoader.

10. Figure 1 shows legitimate "msiexec.exe" process which acts as a cover for the malicious ZLoader's code and communicating with ZLoader's command and control servers. Due to the nature of a code injection into this process, the Windows operating system (and its core processes and functionalities) continues to appear to the victim as a legitimate Microsoft process and cannot be distinguished from a regular harmless process without thorough analysis of the process behavior or inspection of its memory.

11. During infection, ZLoader creates several files and registry keys on the victim's system:

a. Number of randomly named folders and files in %APPDATA% folder, which hold various ZLoader's components, including the copy of the initial executable file.

b. Registry entries in HKEY_CURRENT_USER\SOFTWARE\Microsoft, which are used to hold various settings for ZLoader instance.

c. Registry entry in HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run, which is used to achieve persistence and contains record pointing to the copy of the initial malicious executable file.

Command & Control Server Communication

12. ZLoader malware communicates with its Command & Control (C2) server over HTTP/HTTPS protocol in order to exchange information. Packets are encrypted using RC4 cypher to hide their content and some of them are also additionally encrypted using XOR based algorithm known as "Visual encrypt".

13. ZLoader includes a list of C2 servers which it communicates to and is part of the configuration present in ZLoader's binary file. This list of C2 servers, identified through an internet address using various domain names the malware is programmed to communicate with, is present at the time of the infection of the victim's machine and can moreover be updated with additional C2 addresses.

14. In March 2020, ZLoader implemented Domain Generation Algorithm (DGA) into its code base to help malware operators maintain a connection to their botnet, should all the static C2 servers stop working. Every day, list of 32 new domains is generated. URLs generated by this DGA have following format:

https://<20 random lowercase letters>.com/post.php

15. ZLoader supports several commands to be executed when received from C2 server. Apart from standard commands supporting ZLoader's basic operations, following extensive commands can be received through dynamic configuration from C2 server:

a. user_execute

b. bot_uninstall

c. user_cookies_get

d. user_cookies_remove

e. user_passwords_get

f. user_files_get

g. user_url_block

h. user_url_unblock

i. user_execute_shell

j. user_execute_cmd

Out of all these commands, we have seen only user_execute,

user_cookies_get and user_passwords_get being used by ZLoader during our tracking.

16. user_execute command is used to download and execute arbitrary files on the victim's system. Recently, we have seen following URLs being used by this command to distribute Raccoon infostealer malware:

a. https://braves[.]fun/racoon.exe

b. https://endoftheendi[.]com/12.exe

17. ZLoader is a modular malware which has an initial loader component and additional modules. ZLoader can download following modules from C2 server and then use to support its malicious tasks:

Malicious modules	
Main bot component	The main bot component that controls all the of ZLoader's functionalities for either 32-bit or 64-bit Windows processes.
Hidden VNC module	Remote control module, Virtual Network Computing software, which allows the perpetrators to have remote access to the ZLoader infected machine similar to sitting at the machine and using the machine's keyboard and mouse.

Additional modules through misused legitimate tools	
zlib1.dll	Data compression tool used by ZLoader to compress and decompress data.

Libssl.dll	Used by ZLoader to encrypt and decrypt data using SSL certificates.
Certutil.exe + necessary DLL files	Used by ZLoader to install its own SSL certificate in the Firefox browser and make the copy of the webpage in ZLoader's proxy look as if it is an encrypted channel directly to the genuine webserver.
Sqlite3.dll	Small relational database program used to process browser data.

Man in the Browser Attacks

18. One of the main features of ZLoader is its ability to steal victims' credentials to various online services, such as banks, e-commerce sites, and other online platforms such as Microsoft's login page. ZLoader implements Formgrabbing and Webinjection techniques to do so. Formgrabbing allows ZLoader to gather data entered into selected websites as well as taking screenshots of these websites, while Webinjection allows alteration of the selected websites directly in the victim's browser. To support these techniques, ZLoader implements a proxy server, as a Man-in-the-browser attack, which will intercept website traffic between the user's browser and websites. By intercepting and capturing web traffic between a browser and a website through a man-in-the-browser technique, removing the encryption in ZLoader's main bot proxy component, the web browser's content can be read, modified, and re-encrypted again as a new web page. The new webpage with a different SSL certificate than the one from the authentic webserver is made to look like the authentic webpage, including using

logos and trademarks, but it will be located in ZLoader's proxy function and served to the user's browser from there while made to look like the authentic webpage from the authentic website. ZLoader through this process can access all information between a webserver and the browser even as the user believes it is encrypted secure communications. Formgrabbing and Webinject features are controlled by dynamic configuration pushed by the C2 server.

19. Figure 2 shows Formgrabbing related portion of the configuration as received from C2 server. Screenshots and data entered by victims into their browser, when visiting sites matching expressions in this list, will be collected by ZLoader and sent to its C2 server as displayed in .

```
@https://*commerzbank.de*
@https://*.de/*/entry*
@https://*.de/banking-*/portal?*
@https://*.de/banking-*/portal;*
@https://*.de/portal/portal*
@https://*.de/privatkunden/*
@https://*.de*abmelden*
@https://*.de/de/home*
@https://*.de/en/home*
@https://*.de/fi/home*
@https://*banking.sparda.de*
@https://*banking.sparda-*
@https://*banking.sparda.de/wps/loggedout.jsp
@https://*meine.deutsche-bank.de/trxm/db*
@https://*banking.berliner-bank.de/trxm*
@https://*meine.norisbank.de/trxm/noris*
@https://*targobank.de*
@https://banking4.anz.com/IBAU/BANKAWAY*
@https://banking.westpac.com.au/*
@https://www1.my.commbank.com.au/netbank/Portfolio/Home/*
@https://ibanking.stgeorge.com.au/ibank/*
@https://ibanking.banksa.com.au/ibank/*
@https://ibanking.bankofmelbourne.com.au/ibank/*
@https://online.macquarie.com.au/*
@https://ob.cua.com.au/ib/*
@https://banking.bendigobank.com.au/banking*
@https://internetbanking.suncorpbank.com.au/*
@https://www.ing.com.au/securebanking/*
@https://ib.nab.com.au/*
@https://online.beyondbank.com.au/*
@https://ib.greater.com.au*
@www.independentreserve.com*
@www.coinspot.com.au*
@https://auth.btcmarkets.net/*
```

Figure 2. Formgrabbing part of the dynamic configuration.:

```
Url: https://meine.deutsche-bank.de/trxm/db/gvo/login/login.do
Host: meine.deutsche-bank.de
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 239
Origin: https://meine.deutsche-bank.de
Connection: keep-alive
Referer: https://meine.deutsche-bank.de/trxm/db/
Cookie: TRXMSETTINGS=en|dbag-desktop|00|PIN|DisplayFinancialOverview|; TRXMSECURITY=769305096546l!1573982802-
-!8z7eyQKxM9kws7TILFGYr2Sl_HC7ZSgrpB2fTbRCtzR_fMEzqIJd; csrfToken=GpnDG2hxHFrOJsBcwsawwl; font-sizer=%7B%22font-size%22%3A%22fs-
-small%22%7D; JSESSIONID=dJIqzEMf_RztCRbTf2BprgSZ7CH_lS2rYGFLIT7swk9MxKQye7z8!-2082893751; at_check=true; SameSite=None;
-ADRUM_BTa=R:36|g:df04a9c9-84e4-400e-9e07-8a907a72057b|n:customer1_1871e7bb-1b11-448d-aa99-6eabf9a44312;
-ADRUM_BT1=R:36|i:301416|e:119|d:3; ADRUM_BT2=R:36|i:301416|e:119
Upgrade-Insecure-Requests: 1
Data:
-gvo=DisplayFinancialOverview&process=&wknOrIsin=&quantity=&fingerprintToken=&fingerprintTokenVersion=&updateFingerprintToken=false
-&javascriptEnabled=true&branch=214&account=2343244&subaccount=00&pin=32553&quickLink=DisplayFinancialOverview&
```

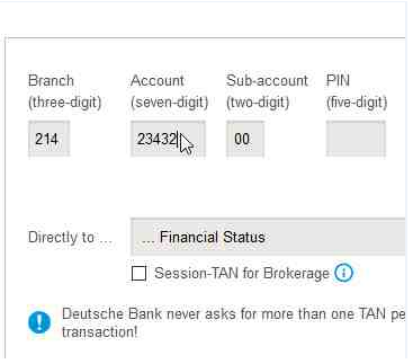

Figure 3:
Data and one of the screenshots collected by ZLoader and sent to its C2 server.

20. Webinjects allow ZLoader to modify the content of a webpage in ZLoader's proxy server of ZLoader so it can be used for criminal activity compared to the authentic webpage which is on the server of the bank or other online service targeted by ZLoader. Figure 4 shows portion of the Webinject configuration.

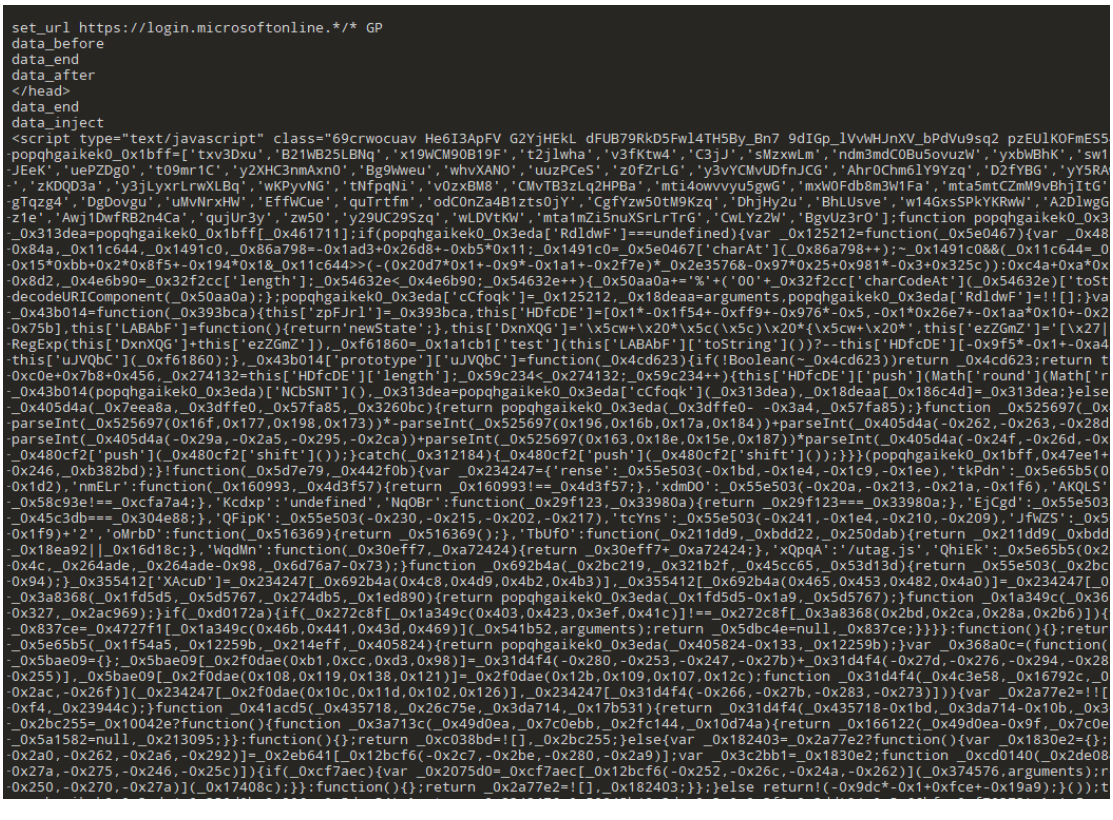

Figure 4: Part of the webinjects configuration.

21. The instructions for how to make changes to the copies of webpages provided by ZLoader's proxy function are sent to the infected machine when the the ZLoader malware communicates to the ZLoader's C2 servers. Webinjects configuration follows the following format:

set_url

data_before

data_end

data_after

data_end

data_inject

data_end

a. set_url contains expression, that is to be matched in URL to perform code injection, followed by an expected HTTP request method. POST (P) and GET (G) methods are supported.

b. Expression between data_before/data_end and data_after/data_end is used to mark position, where the code is going to be injected.

c. Code between data_inject/data_end encloses javascript code to be injected into the web page loaded by browser on victim's machine. This code is usually heavily obfuscated and alters a behavior of the loaded webpage.

22. Figure 5, below, shows loading of additional malicious scripts into the copy of the webpage, located in ZLoader's proxy server, for Microsoft's online sign in page https://login.microsoftonline.com. This means that the webpage served from ZLoader's proxy to the user's browser may look identical to the authentic webpage from Microsoft's authentic webserver, but the copy in ZLoader's proxy server in reality has a very different function. It has been crafted with the purpose to steal information while the copy of the webpage looks identical

to the authentic one, including logos and trademarks. In this particular case, visual representation of the webpage in ZLoader looks visually similar to the authentic webpage on the authentic webserver even if the fake webpage has an information stealing and modifying functionality and thus difficult for users to reveal that the website they requested was altered before rendering in their browser.

23. To support this functionality, ZLoader runs a local proxy server

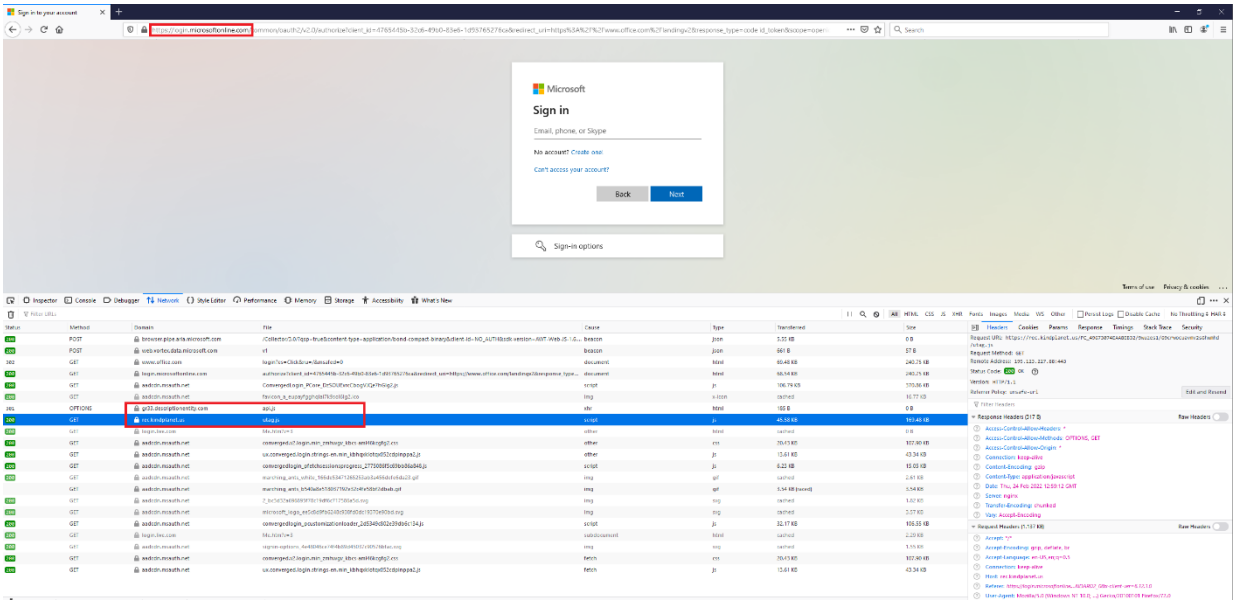

Figure 5:
Additional malicious scripts loaded by ZLoader into legitimate website

operated by the aforementioned injected msiexec.exe process so that ZLoader looks like it is Microsoft code because appears to the user as “Windows® installer”. Additionally, ZLoader hooks certain functions in a browser process to intercept browser traffic and redirect this traffic to ZLoader’s local proxy server. ZLoader may then collect private data being sent by the browser or modify a copy

of the received webpage and send it back to the browser.

24. Based on this research and observations, I can declare that users of Microsoft Corporation and Deutsche Bank were targeted among many others companies by ZLoader malware. This may lead to loss of personal data and possible financial loss for customers of these companies.

Executed this 5th day of April, 2022 at Brno, Czech Republic.

X Tomas Prochazka