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.: **FILED UNDER SEAL**

DECLARATION OF RODELIO G. FIÑONES IN SUPPORT OF PLAINTIFFS' APPLICATION FOR AN EMERGENCY *EX PARTE* TEMPORARY RESTRAINING ORDER AND ORDER TO SHOW CAUSE RE PRELIMINARY INJUNCTION

I, Rodelio G. Fiñones, declare as follows:

1. I am a Senior Security Software Engineer & Malware Researcher in Microsoft Corporation's Digital Crimes Unit ("DCU"). I make this declaration in support of Microsoft's Application for An Emergency Temporary Restraining Order and Order To Show Cause Re Preliminary Injunction. I make this declaration of my own personal knowledge or on information and belief where

noted. If called as a witness, I could and would testify competently to the truth of the matters set forth herein.

I. INTRODUCTION

2. I have been employed by Microsoft since June 2009. In my role at Microsoft, I assess technological security threats to Microsoft and the impact of such threats on Microsoft's business and customers. I work with a team of investigators that focuses in part on researching different categories of malware, including botnets. My team and I research emerging malware threats through analysis of submitted samples, reverse engineering, forensic examination, data stream analysis, and development of tools to track botnet development. I am the team lead in developing malware prevention and eradication tools. Prior to joining Microsoft, I worked from 2004-2009 for Fortinet Technologies (Canada), Inc. as a Principal Software Developer/Researcher (2007-2009) and Senior Antivirus Analyst (2004-2007). My job included research and analysis of complex malware and the development of tools to detect and eradicate malware. From 1999-2004, I worked for Trend Micro, Inc. as a Senior Anti-Virus Researcher and Anti-Virus Engine Developer. During my professional career, I have received advanced, specialized training and extensive practical experience in investigating malware and botnets and in devising technical countermeasures to detect and disable them.

A. Overview Of My Investigation Into ZLoader And My Top Conclusions

3. My declaration concerns a botnet referred to collectively as "ZLoader." A "botnet" is a network of computing devices comprised of Internet-connected computing devices that have been infected with some form of malicious software ("malware"). The malware provides control over the infected computers to the individual criminals who operate the botnet. Botnets vary in size and complexity and may be comprised of only a few hundred up to many millions of infected computers. The Defendants in this case have directed such malware over the Internet to many victim computing devices and have created and control the ZLoader botnet.

4. ZLoader is a very scaled botnet and Microsoft has observed over 200,000 infected computing devices around the world, leveraging victim machines to steal online banking credentials with a variety of tools, including Man in the Browser ("MITB") capabilities. My investigation leads me to believe that a central group of botnet developers wrote the code for ZLoader and then chose to commercialize it by offering other criminals use of ZLoader on a "malware-as-a-service" basis. ZLoader is disseminated via malicious advertisements, exploits, malicious spam email and spearphishing campaigns, among other methods. The spam email and spearphishing campaigns send unsolicited messages that deceive

targeted victims into downloading the ZLoader malware from malicious websites or through malicious attachments, such as those designed to look like legitimate Microsoft Word or Excel files. ZLoader is also dropped as a secondary payload by other malware.

5. For example, once installed, beyond its own financial theft functionality, ZLoader can further deliver the Ransomware families such as Ryuk, Egregor, Nefilim, and DarkSide ransomware to the victim's machine. ZLoader can also install other tools for malicious purposes, such as CobaltStrike, which is used to assist with lateral movement and ransomware deployment, and Darkside, which is used to target multiple large, high-revenue organizations resulting in the encryption and theft of sensitive data and threats to make it publicly available if the ransom demand is not paid.

6. Once ZLoader infects a new victim computing device, it contacts a command and control computer over the Internet from which it begins to receive instructions and additional malware modules. This effectively places the infected computer under the command of the operators of the botnet.

7. ZLoader's design is "modular," which means that it is designed to allow other types of malware to plug into it and perform different tasks. This allows the ZLoader malware running on an infected computing device to serve as a

general platform for other types of malware. Consequently, ZLoader-infected computing devices are subject to a wide range of secondary malware infections, which victimize the infected user in a wide variety of criminal schemes. I have witnessed or have read reports of ZLoader's modules performing tasks for carry out various tertiary tasks that normally involve credential theft, system and network profiling, email and data harvesting, and further propagation of the malware. These ZLoader components contain software code that interacts with and makes changes to Microsoft's operating system and application software during the infection process. We reverse-engineered the ZLoader malware to determine how it operates.

B. Description of Investigation into ZLoader

8. Investigators at Microsoft's Digital Crimes Unit and Microsoft Defender Advanced Threat Protection have been researching ZLoader recently. One of the other investigators with whom I have worked on this investigation, Christopher Coy, is a co-declarant in this matter, and I refer the Court to his declaration for further information on other aspects of ZLoader, including ZLoader's command and control structure, function, and operation, ZLoader's harm to Microsoft customers, and the plan for disrupting its operations. I have reviewed his declarations and concur in his conclusions.

9. In the course of Microsoft's investigation into ZLoader, I, and other Microsoft investigators, purposely infected several investigator-controlled computers with the malware that the ZLoader botnet deploys. This placed the computers under the control of the cybercriminals operating the botnet to enable me and other Microsoft investigators to monitor the telemetry of the ZLoader infrastructure.

10. Among other things, we carefully analyzed ZLoader's core malware, additional modules that are part of the malware and associated configuration files. These ZLoader components contain software code that interacts with and makes changes to Microsoft's operating system and application software during the infection process. We reverse-engineered the ZLoader malware to determine how it operates.

11. In order to reverse engineer ZLoader's code, we allowed the malware samples to execute in an isolated and controlled environment in order to take a code snapshot (called a "dump") containing decrypted and unpacked malware code from computer memory. These dumps are then loaded to a decompiler called IDA Pro, for comprehensive analysis and understanding of the behavior of the ZLoader sample.

C. ZLoader's Man-in-the-Browser Capabilities

12. While ZLoader is used to distribute other malicious software, including ransomware known as Ryuk, ZLoader itself is a malware device that is designed to steal the end-user's banking credentials and other personal identifying information in order to access the end-user's bank accounts and siphon funds to the Defendants or other criminal organizations. The primary process by which ZLoader steals this information is through client-side web injection and formgrabbing. Web injection allows the attacker to alter content of the websites displayed to the victim while formgrabbing capture credentials from the browser windows.

13. To accomplish those actions, the malware implements Man-in-the-browser (MITB) attacks. ZLoader steals account credentials as follows. This technique is designed to monitor the victim's activity, identify and exfiltrate cookies and credentials from browsers and Microsoft Outlook, and detects when the victim is navigating via their browser to the online portals of a wide variety of financial institutions, including banks, brokerage firms and credit card companies. For example, the malware has the following script commands that enable it to obtain highly confidential information from the victim device:

a. **User_cookies_get**: this command is responsible for searching databases where cookies of particular browsers are stored, opening them, and extracting content by SQLite queries. The following queries are used:

i. select `host`, `name`, `value`, `path`, `expiry`, `isSecure`, `isHttpOnly`,
`sameSite` from `moz_cookies`

b. **User_passwords_get**: Execution of this command triggers stealing passwords saved in the attacked browsers. The following query are executed:
   i. select `origin_url`, `username_value`, `password_value` FROM logins

c. **User_files_get**: Execution of this command triggers the operation of searching and uploading important documents such as databases and crypto wallets and credentials from the victim device.

14. **Figure 1** below is a code snapshot of ZLoader's functionalities designed to exfiltrate cookies and cache usernames and passwords from major browsers like Internet Explorer, Firefox, Chrome, and Microsoft Edge.

```
int __userpurge Thread_StealCreds@<eax>(int *a1@<ebx>, int a2)
{
  void (__stdcall *SetEvent)(_DWORD); // esi
  char *CoreInfo; // eax
  int v4; // eax
  int v5; // eax

  SetEvent = (void (__stdcall *)(_DWORD))GetApiByHash(0, SetEvent_);
  CoreInfo = GetCoreInfo();
  SetEvent(*((_DWORD *)CoreInfo + 292));
  sub_2629300();
  if ( (unsigned __int8)DownloadSqlite((int)a1) )
  {
    if ( !(unsigned __int8)ItemExist(20021, (int)a1) )
    {
      StealChromeCreds();
      ResetItem(20021, (int)a1);
    }
    v4 = xor(0xF9298AF);
    if ( !(unsigned __int8)ItemExist(v4, (int)a1) )
    {
      StealCookies(a1);
      ResetItem(20020, (int)a1);
    }
    v5 = xor(0xF9298AC);
    if ( !(unsigned __int8)ItemExist(v5, (int)a1) )
    {
      StealOutlookCreds();
      ResetItem(20023, (int)a1);
    }
  }
  return 0;
}
```

Figure 1: Code snapshot for stealing functionalities

15. ZLoader Defendants use a common technique known as "hook browser processes" to support its "man-in-the-browser" functionality. Hooking is a process by which an application intercepts an application program interface call between two other applications. In effect, the hooking process effectively passes

control of the API calls to the control of the malware. This process allows ZLoader to intercept victim data, evade detection mechanisms, and maintain persistence over victim machines. For example, one known hook is "TranslateMessage." This hook intercepts API calls responsible for keylogging and making screenshots. This functionality allows the malware proceeds to record what is happening on the screen: by capturing the title of the active window, recording the keyboard state, and, eventually making a screenshot showing the performed activity.

16. In addition, certain hooking functionalities enable the ZLoader malware to install fake browser certificates on the local network. For example, whenever a victim accesses a website, that website will register a browser certificate that confirms the website is secure. However, in the case of ZLoader, the Defendants install their own certificates to exploit on the victim's browser. This prevents the victim user from disseminating information to the recipient website through secure HTTPS protocol. Instead, when the victim launches a website, the malware will intercept that request and create an independent connection to the server through the fake certificate. This is achieved by ZLoader by implementing certain Application Programming Interfaces ("API") that are designed to specifically track the victim's browser usage and redirect certain

network activity. For example, the following two API functions enable ZLoader to validate any browser certificate:

- ntdll.dll – *ZwDeviceIoControlFile*
- crypt32.dll – *CertGetCertificateChain, CertVerifyCertificateChainPolicy*

17. Once this connection is made, the victim thinks they are communicating with a legitimate recipient, but in reality, any further communications from the victim will go directly to the Defendants. At first, the victim believes the communication is secure, typically indicated by a lock to the right of the browser field. However, upon further review, the "details of the connection" show that Mozilla Firefox does not recognize this certificate issuer.

18. ZLoader's main process msiexec.exe spawns several threads running at the same time to achieve different malicious tasks. Each of these threads communicate to one another using shared data store in the global memory, system registry and encrypted files. In other words, ZLoader's main process's thread sends communications that execute certain functions within Windows registry and folder paths, modifies the system processes that contain the "Microsoft" and "Windows" trademarks.

19. This executable process, msiexec.exe executes functions to install fake certificate and run a local proxy while other thread is injected and executing

inside the loaded browser process and responsible for redirecting traffic via proxy.

Figure 2 shows the local proxy servers listening on local address (127.0.0.1) on random high ports from 10000 to 20000.

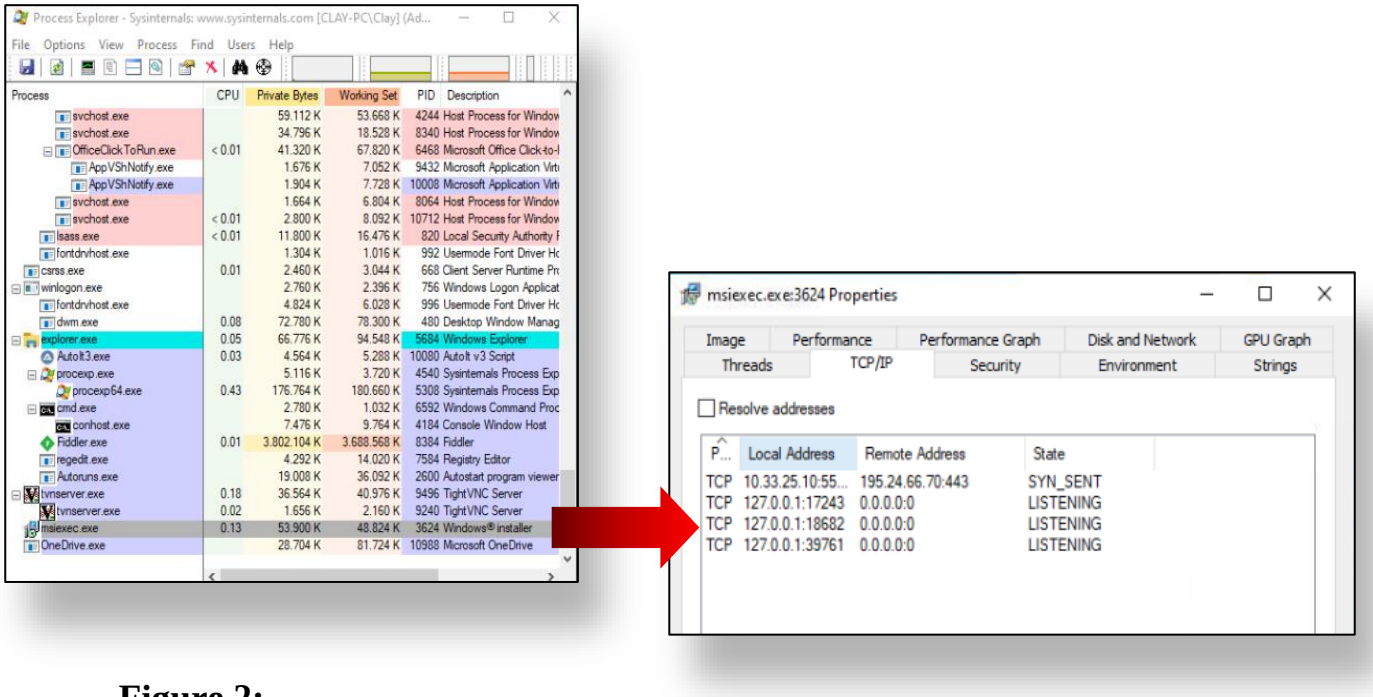

Figure 2:
Msiexe.exe running local proxy servers

20. Each web inject instruction maintains the following format, which is further depicted by **Figure 3** below:

set_url <REGEX_URL> [OPTIONS]
data_before
<STRING_BEFORE>
data_end

data_after
< STRING_AFTER>
data_end

data_inject
<JAVASCRIPT_CODES>
data_end

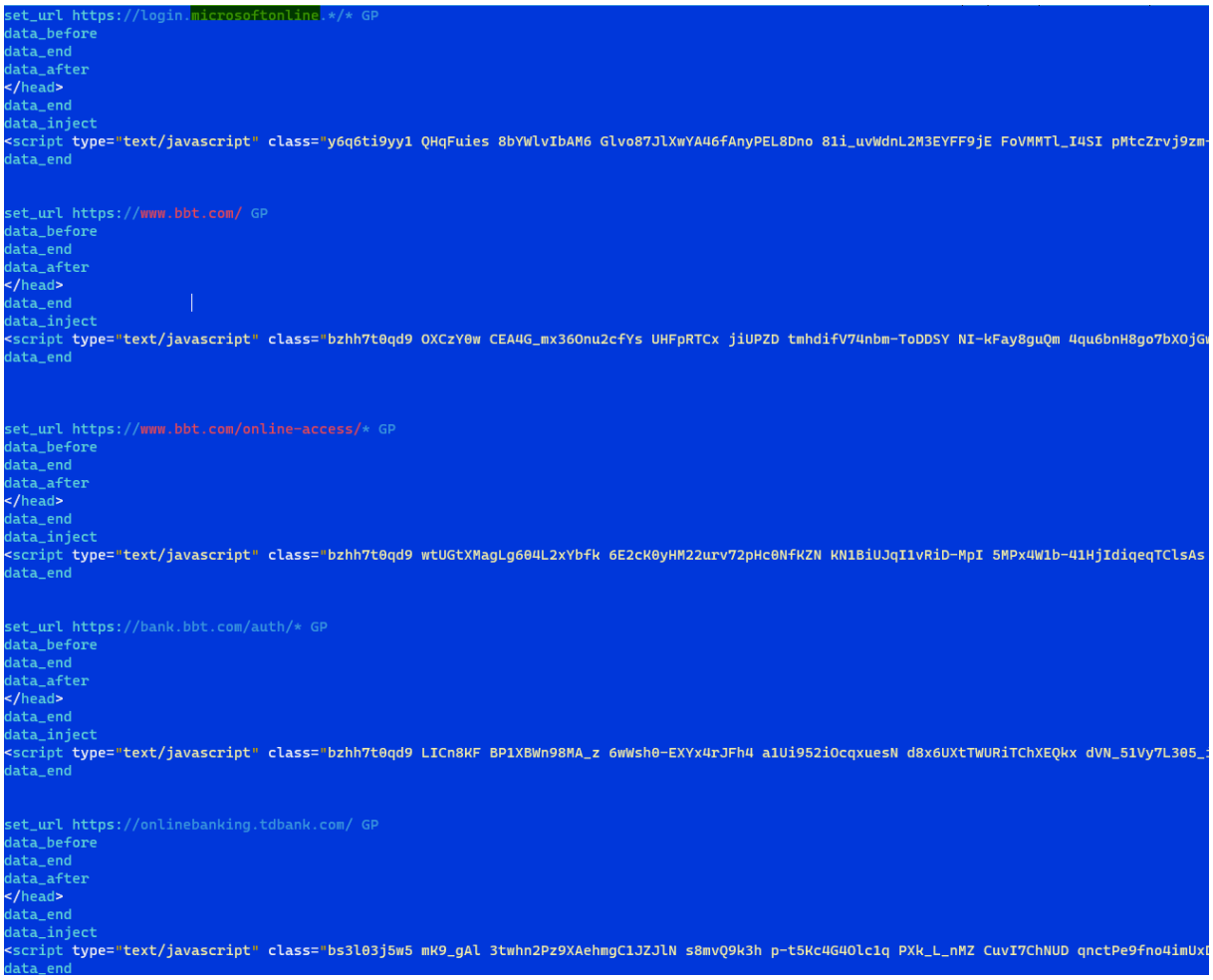

Figure 3: Webinject Format

21. Once ZLoader infiltrates a victim device, its initial process is to navigate through core functionalities within the Window's system. For example, ZLoader runs processes that enable the operators to inject codes to target browser processes discovered. **Figure 4** code snapshot shows the code responsible for such

behavior. As can be seen in **Figure 4**, ZLoader targets the following browser processes.

- iexplore.exe
- firefox.exe
- chrome.exe
- msedge.exe (Microsoft Edge)

```
CreateToolhelp32Snapshot = (int (__stdcall *)(int, _DWORD))GetApiByHash(0, CreateToolhelp32Snapshot_);
v1 = xor(261281433);
result = CreateToolhelp32Snapshot(v1, 0);
if ( result != -1 )
{
  v3 = result;
  ClearMem_0((int)v45, 556);
  v45[0] = xor(261280951);
  Process32FirstW = (int (__stdcall *)(int, int *))GetApiByHash(0, Process32FirstW_);
  v5 = Process32FirstW(v3, v45);
  if ( (is_eq(v5, 0) & 1) == 0 )
  {
    v60 = v3;
    do
    {
      v6 = DecryptUString(aJwOaQ, v50);       // b'iexplore.exe'
      LODWORD(v57) = tolower___0(v47, v6);
      v7 = DecryptUString(aE55eaO, v53);      // b'firefox.exe'
      HIDWORD(v57) = tolower___0(v47, v7);
      v8 = DecryptUString(L"@Z5?NKx$\bQw", v55);// b'chrome.exe'
      v58 = tolower___0(v47, v8);
      v9 = DecryptUString(L"NA\"4DKx$\bQw", v54);// b'msedge.exe'
      v10 = tolower___0(v47, v9);
      v11 = DecryptUString(aFj7L33Q, v49);    // b'explorer.exe'
      v12 = tolower___0(v47, v11);
```

Figure 4: ZLoader browser process injection

22. The configuration files downloaded to the infected computer may contain templates that contain the website addresses for online banking sites of major financial institutions, as well as other websites and web services. The configuration files contain instructions designed to modify the appearance of the genuine banking websites. The configuration files may contain templates that

mimic the websites of major financial institutions and other websites and web services.

23. Once installed on an end-user computer, the malicious software detects when the user of that computer navigates to an online banking website (or any other website specified in the configuration files). When a user visits their online banking website, the malicious software may do one of the following:

a. Access the real banking website, but unknown to the user, execute instructions that modify or extend the website. In particular, the ZLoader Botnets may cause the website to contain extra fields into which users are instructed to type additional sensitive information that is not requested at the legitimate website. For example, the fake versions of the websites may seek information such as ATM "PIN," social security number, mother's maiden name, addresses, birthdates and similar information.

b. Intercept the request from the user's web browser and present the user with a fake website, based on the template, which appears to be the legitimate website.

24. During my investigation, I have seen ZLoader target Microsoft's Outlook application. The ZLoader malware contains functions that queries several entries in Window's registry to extract information about email related credentials

such POP3, IMAP, SMTP and HTTP servers, email accounts, and cache passwords. Depending on the versions of Outlook applications, these encrypted passwords can be recovered with some tools available on the internet. Below is the list of relevant registry entries being queried.

- *HKCU\Software\Microsoft\Internet Account Manager\Accounts\Identities*
- *HKCU\Software\Microsoft\Internet Account Manager\Accounts\Outlook*
- *HKCU\Software\Microsoft\Office\Outlook\OMI Account Manager\Accounts*
- *HKCU\ Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Microsoft Outlook Internet Settings*
- *HKCU\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\Outlook*
- *HKCU\Software\Microsoft\Office\15.0\Outlook\Profiles\Outlook*

25. Beyond the financial institutions, ZLoader's Man-in-the-Browser functionalities targets webpages associated with Microsoft. Specifically, ZLoader targets *http://login.microsoftonline[.]com*. **Figure 5** is a depiction of a malicious Microsoft sign-in screen with an injected Javascript code:

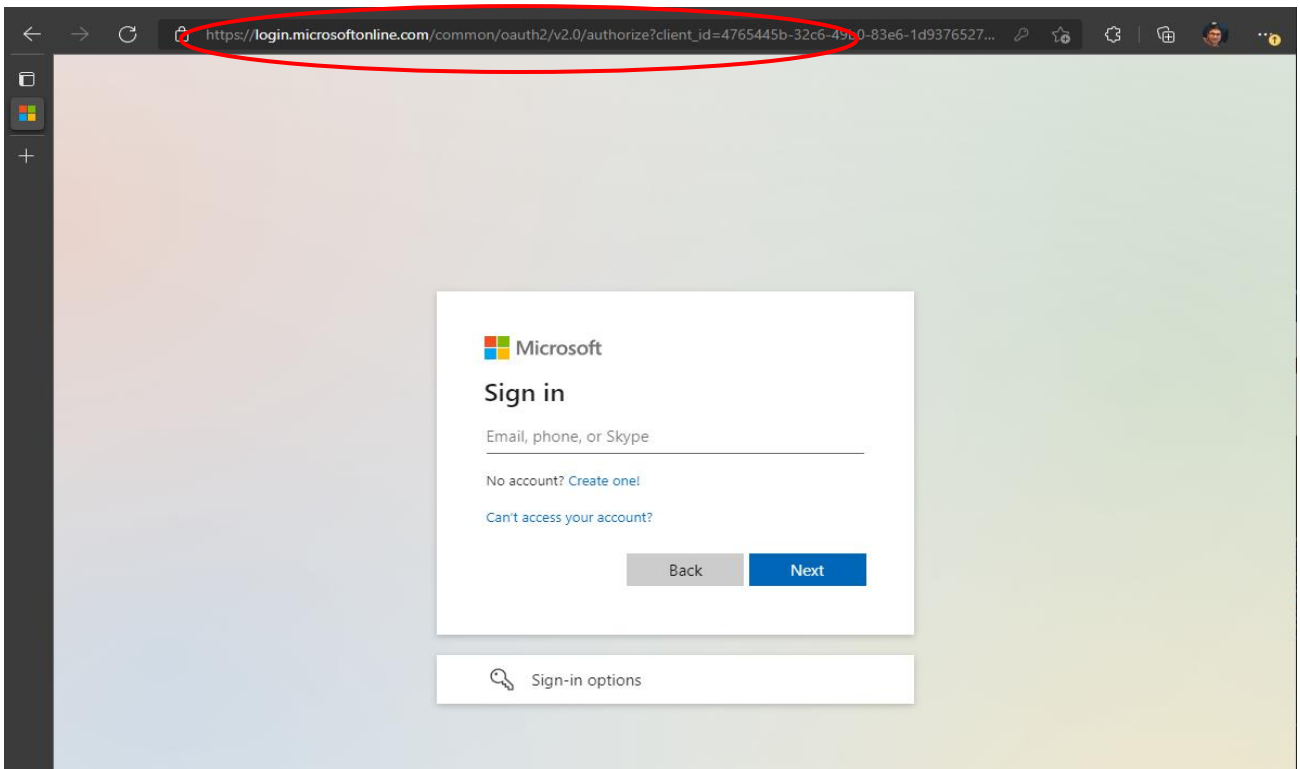

Figure 5: Snapshot of the edge browser with injected JavaScript

26. Microsoft's login pages for several of its main products, such as office.com, automatically will redirect the user to the aforementioned Microsoft online page when the user tries to log in to their Microsoft account. When the user loads their favorite web browser such as Microsoft Edge and the users visits and tries to log into their Microsoft account, ZLoader will match the URL to the list of targets. In this case it will match to the first one above and perform the web

```
<!-- Copyright (C) Microsoft Corporation. All rights reserved. -->
<!DOCTYPE html>
<html dir="ltr" class="" lang="en">
<head>
    <title>Sign in to your account</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=2.0, user-scalable=yes">
    <meta http-equiv="Pragma" content="no-cache">
    <meta http-equiv="Expires" content="-1">
    <link rel="preconnect" href="https://aadcdn.msftauth.net" crossorigin>
<meta http-equiv="x-dns-prefetch-control" content="on">
<link rel="dns-prefetch" href="//aadcdn.msftauth.net">
<link rel="dns-prefetch" href="//aadcdn.msauth.net">

    <meta name="PageID" content="ConvergedSignIn" />
    <meta name="SiteID" content="" />
    <meta name="ReqLC" content="1033" />
    <meta name="LocLC" content="en-US" />


        <meta name="format-detection" content="telephone=no" />

    <noscript>
        <meta http-equiv="Refresh" content="0; URL=https://login.microsoftonline.com/jsdisabled" />
    </noscript>



<meta name="robots" content="none" />
```

injection by inserting the code specify as JAVASCRIPT_CODES parameter after the string "</head>" and then rendering to the browser application. **Figures 6 and 7** show the snapshot of the codes injected into the Edge browser.

Figure 6: Snapshot of the JavaScript codes injected into Edge browser

Figure 7: Snapshot of the obfuscated JavaScript codes

27. When ZLoader detects that the user is visiting a financial institution website or a Microsoft sign-in website, the malware utilizes the webinject that alter

or replace content or display additional fields in the website as it appears to the victim in their browser. In this way, the victim believes that they are at the legitimate online financial website, when in fact they are seeing either an entirely fake version of the website to which the ZLoader module has diverted them, or a version of the website that has been manipulated by Defendants. When the user types their login credentials into the website or types additional information into fraudulent fields injected by the Defendants (such as pin codes, answers to security questions or other personal information), the Defendants are able to intercept that information and use it to log into the user's online accounts. This captured information is encrypted and sent to the main bot and then to C2 server. The Defendants can then initiate funds transfers, resulting in theft of the victim's money.

28. In each of these cases, the website presented to the user is a fake or modified version, which appears very similar to the legitimate website and misuses the trademarks and website content of financial institutions and of Microsoft. During my investigation, I have observed that the ZLoader Botnets create fraudulent, extended versions of websites of an array of financial institutions and payment services targeting mostly US, Canadian, Australian, and selected German banks. The complete list of targeted banks, financial institutions and other targeted

online service providers is as follows.

- AOL
- Abt Electronics
- Amazon
- Amerant
- American Express
- Ameritrade
- Associated Bank
- BT Banking
- BancorpSouth
- Bank of America
- Bank of Melbourne
- Bank of Montreal
- Bank of Queensland
- Barclays
- Best Buy
- Blockchain
- Build with Ferguson
- CIBC
- Cabelas
- CapitalOne Bank
- Centennial Bank
- Charles Schwab
- Chase
- Cigars International
- Citigroup
- Citizens
- Columbia Bank
- Comerica
- CommBank
- Commerzbank
- Costco
- Designer Shoe Warehouse
- Desjardins
- Deutsche Bank
- Deutsche Bank (aka norisbank)
- Discover
- Duluth Trading
- E*Trade
- Eastern Bank
- Ebanking Services
- Ebay
- Elan Financial Services
- FIS
- FWRD
- Fidelity
- Fifth Third
- First National Bank of Omaha
- Fiserv
- Frost
- Fulton Bank
- Funds Xpress
- Gilt
- Global Industrial
- GoToMyCard
- Google
- Groupon
- HSBC
- HSN (Home Shopping Network)
- HawaiiUSA FCU
- Heritage Bank
- Home Depot
- Huntington
- Interactive Brokers
- Investors Bank
- J.Crew
- JP Morgan
- Jerry's Artarama
- KeyBank
- LLBean
- LexisNexis
- M&T Bank
- Merrill
- Microsoft
- Mouser Electronics
- NAB
- NFM
- Navy Federal
- Neiman Marcus
- NewEgg
- Office Supply
- Overstock
- PNC Financial
- Paypal
- Prosperity Bank (aka Legacy Texas Bank)
- QVC
- R.S. Hughes
- Regions
- Robinhood
- Royal Bank of Canada
- Scotia Bank
- SendSpace
- Sephora
- Southwest
- Sparda
- Staples
- State Farm
- Summit Racing Equipment
- Sutherlands
- Synchrony
- Synovus
- T.RowePrice
- TD Ameritrade
- TD Bank
- TIAA
- Targo Bank
- Timberland
- Truist (aka BB & T)
- Truist (aka SunTrust)
- UMB
- US Bank
- USAA
- Union Bank
- Vanguard
- Walmart
- Washington Trust Bank
- Wayfair
- Wayfair (aka All Modern)
- Wayfair (aka Birch Lane)
- Wayfair (aka Joss & Main)
- Wells Fargo
- Westpac
- Westpac (aka bankSA)
- Yahoo
- Zappos (aka 6pm)
- ZoomInfo

29. The Defendants then use the data exfiltrated during the Man-in-the-

Browser attacks from victim computers to the command and control servers to access victims' online financial accounts and steal money from those accounts.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct to the best of my knowledge.

Executed on April 4, 2022.

/s/ Rodelio G. Fiñones
Rodelio G. Fiñones