G DATA
Whitepaper 2018 – paper 4

Analysis of

Script.Trojan-Downloader.Fodevepdf.A

Analysis by: https://twitter.com/RansomBleed

G DATA Software AG | Malware analysis | June 2018

Contents

1

1. Introduction

The downloader *Script.Trojan-Downloader.Fodevepdf.A*[1] was discovered by the Twitter user @malwrologist. He tweeted about this here. The special feature of this downloader is its fileless UAC bypassing technique which we will have a deeper look on in this report.

2. Initial resume.js file

The actual malicious code in JScript malware is often obfuscated in order to bypass antivirus systems. This time it's not different. The malicious code is stored in the variable *ivlckqku.* Inside a while loop a number between 1 and 109 is generated. This generated number replaces all the 'o' characters inside the *ivlckqku* string. Then the function *nfdqnn* is called which does the actual deobfuscation of the modified *ivlckqku* string.

This happens inside the while loop until the first function can be found in the deobfuscated string. This is a check to see if the deobfuscation has been done correctly in order to not throw any errors in the next step. If the keyword 'function' is found, the deobfuscated JScript is now executed.

The procedure reminds me of the hyperion PE-crypter[2] which lets the final payload brute-force its own AES key to then launch the decrypted program in memory.

3.1 Deobfuscated downloader called without parameters

In this part the actual bypassing technique comes into play. If no parameters are called with the downloader, it checks for the current windows version by using the query "SELECT * FROM Win32_OperatingSystem" to the WMI(windows management instrumentation).

For windows 7: The downloader places the registry key *HKEY_CURRENT_USER\SOFTWARE\Classes\mscfile\shell\open\command* with the value *"C:\Windows\System32\wscript.exe* '"+ Location of the script + "'*ohwoebccm"*. Then it executes *eventvwr.exe*(Windows system file). The interesting part is that eventvwr.exe is first looking for *HKEY_CURRENT_USER\Software\Classes\mscfile\shell\open\command* before it looks into *HKEY_CLASSES_ROOT\mscfile\shell\open\command* to execute the value "*mmc.exe"*(Also a legit windows system file) of the key. Because the downloader updated the value of *HKEY_CURRENT_USER* the *eventvwr.exe* is executing the updated value elevated instead of the *mmc.exe.*

After two seconds, the downloader deletes the created registry key to hide its actions.

For windows 10: The UAC bypassing technique is very similar for windows 10 systems. The difference is that the registry key *HKEY_CURRENT_USER\Software\Classes\ms-settings\shell\open\command* is used to update the instructions and an additional registry key *HKEY_CURRENT_USER\Software\Classes\ms-settings\shell\open\command\DelegateExecute* is created. Then the *fodhelper.exe* is called instead of the *eventvwr.exe.* The created keys are deleted as well.

After the UAC bypass: If the file *C:\Users\USERNAME\Documents\SERIALNR* doesn't exist, the trojan executes itself with the parameter *"ohwoebccm"*, sleeps 1 second, creates the file above and deletes itself.

3.2 Deobfuscated downloader called with parameters

If the downloader is called with parameters, it either means that it's running with elevated privileges because the windows 7 / 10 UAC bypass was successful or that the UAC bypass was unsuccessful and that the user still needs to interact with the upcoming download later on. Firstly, the file *C:\Users\USERNAME\Documents\SERIALNR* is created. After that the downloader deletes itself if it's not already deleted.

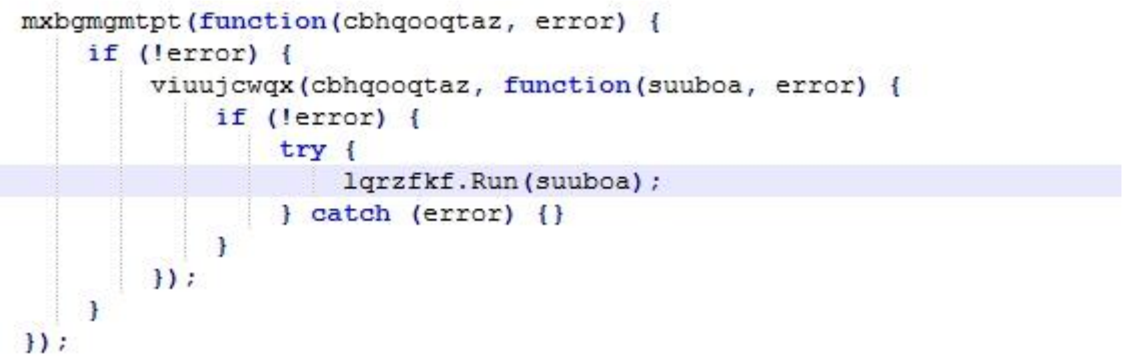
Download and execute routine: The function *mxbgmgmtpt* is called with a new function as parameter. Inside *mxbgmgmtpt* there are 3 calls to the function *fxzgyeg* which are executed seperatly if the previous call has failed. *Fxzgyeg* is called with a download URL and another new function. The download links are listed below:

- http://xn--tor573cjye2rebtnlwvxkd.com/update.php
- http://xperjeans.com/update.php
- http://yiceo.com/update.php

The function *fxzgyeg* connects to one of those URLs and then returns the function call *zcvfplr(lfpwiwtqu.ResponseBody, false)* if the connection was successful. *Lfpwiwtqu* is a *MSXML2.XMLHTTP* ActiveXObject.

If this whole "functionception" is successful, the function *viuujcwqx(cbhqooqtaz, function(suuboa, error)* is called. Due to the structure of the function calls the variable *cbhqooqtaz* contains the downloaded data from one of the download links. This function creates an *ADOB.Stream* ActiveXObject which is a way to read and write binary files with JScript. After that the content of *cbhqooqtaz* is written to *C:\Users\USERNAME\AppData\Local\Temp.*

Once this operation is successful, the final file which is a GandCrab[3] version is executed. In figure 1 below you can see the main execution structure to get a visual understanding of the structure.



```
mxbgmgmtpt(function(cbhqooqtaz, error) {
    if (!error) {
        viuujcwqx(cbhqooqtaz, function(suuboa, error) {
            if (!error) {
                try {
                    lqrzfkf.Run(suuboa);
                } catch (error) {}
            }
        });
    }
});
```

Figure 1. deobfuscated downloader main routine.

4 File hashes and resources

[1] SHA-256 bd42d38e6146672ff14e4bd6f3eb8bae3d11f71f68d2beb4ee7c91ba9337feb3

[2] http://nulnullsecurity.net/tools/binary.html

[3] SHA-256 a7316264628f8a8e586fe250925b546dc3172a59837ae26c0fbc442d9d746413

If you want to stay updated about malware, be sure to follow these accounts:

RansomBleed - My personal twitter account about the latest malware reports.

GDataSoftwareAG – G DATAs twitter company account.

Blog – The G DATA blog about all kinds of security-related news.