Practical Malware Analysis & Triage

Malware Analysis Report

Unknown.exe

May 2023 | CybErlich | Dor Erlich

Table of Contents

Executive Summary

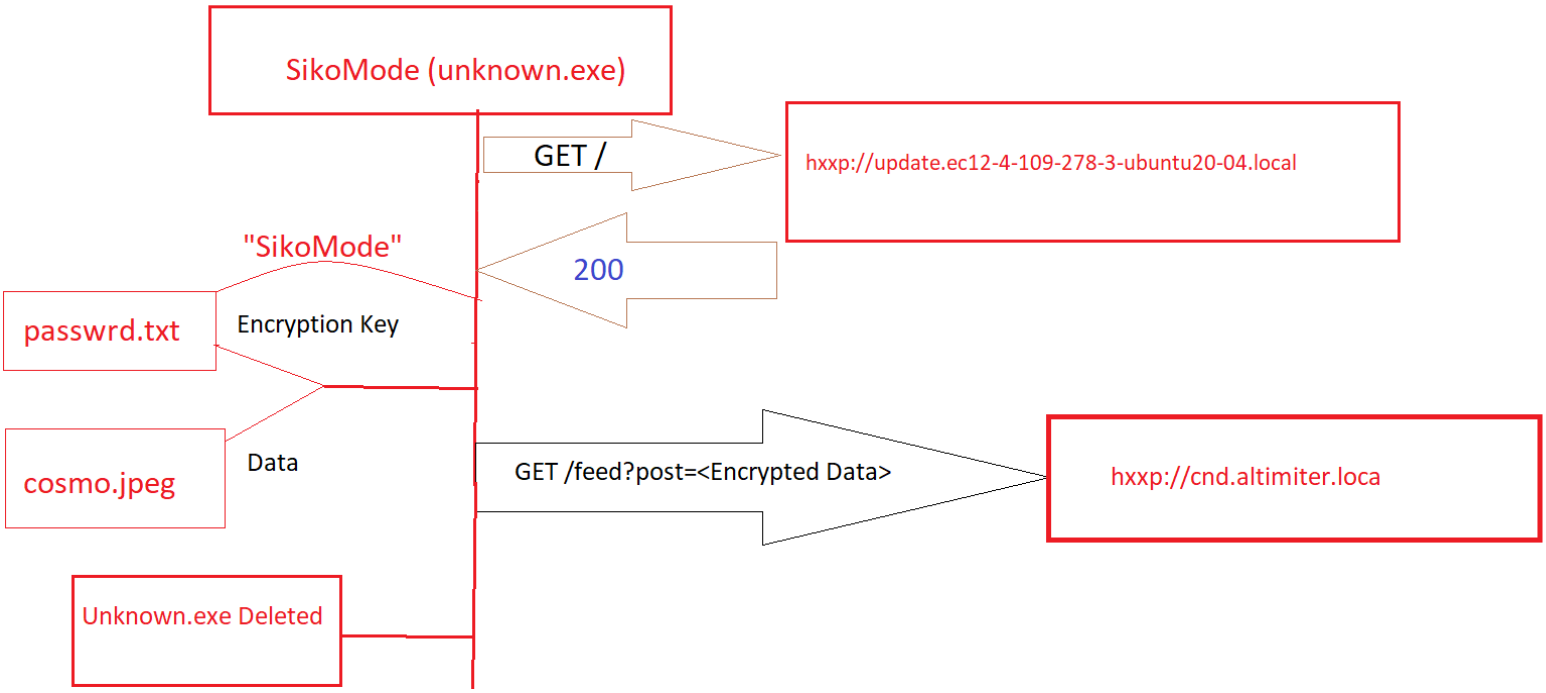
SHA256 hash	3aca2a08cf296f1845d6171958ef0ffd1c8bdfc3e48bdd34a605cb1f7468213e

SikoMode is a Num compiled Exfiltration-Trojan first identified on Jan 11th 2021. It was discovered on a Windows 10 machine. The malware was found to be communicating with two URLs listed in Appendix and was able to exfiltrate data from the infected system. This sample targets and exfiltrates a specific Jpeg file by its name, however, it can be an exfiltration stage of a larger attack and also can be modified in the feature and spread in other versions.

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

High-Level Technical Summary

SikoMode is an exfiltration Trojan. Once executed, the malware sends a Get request to the domain (hxxp://update.ec12-4-109-278-3-ubuntu20-04.local) creates a text file write an encryption key in it, and then reads, encrypt and exfiltrates data from a file (cosmo.jpeg), to the remote server (hxxp://cnd.altimiter.loca). The malware is capable of stealing information. To evade detection, the malware deletes itself on the end of the execution.



Malware Composition

File Name	SHA256 Hash
unknown.exe	3aca2a08cf296f1845d6171958ef0ffd1c8bdfc3e48bdd34a605cb1f7468213e
password.txt	1eebfcf7b68b2b4ffe17696800740e199acf207afb5514bc51298c2fe7584410

SikoMode consists of the following components:
The malware is composed of a single executable file named "unknown.exe”

The malware creates a text file named “passwrd.txt” in the “User\Public” directory.

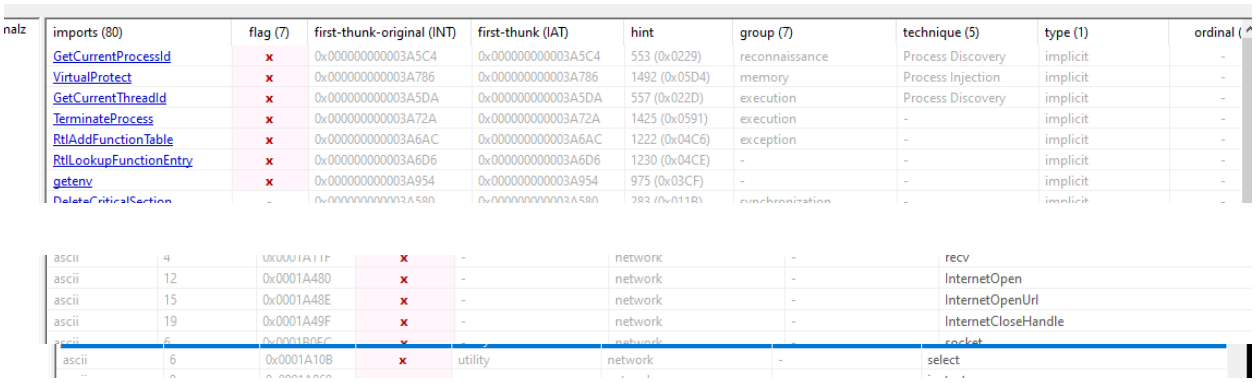
The malware uses Data encryption to obfuscate its functionality, making it difficult to analyze

Basic Static Analysis

{Screenshots and description about basic static artifacts and methods}

Hash: sha256 - 3aca2a08cf296f1845d6171958ef0ffd1c8bdfc3e48bdd34a605cb1f7468213e

CPU:64-bit
File-type:executable



nalz	imports (80)	flag (7)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (7)	technique (5)	type (1)	ordinal (
	GetCurrentProcessId	x	0x000000000003A5C4	0x000000000003A5C4	553 (0x0229)	reconnaissance	Process Discovery	implicit	-
	VirtualProtect	x	0x000000000003A786	0x000000000003A786	1492 (0x05D4)	memory	Process Injection	implicit	-
	GetCurrentThreadId	x	0x000000000003A5DA	0x000000000003A5DA	557 (0x022D)	execution	Process Discovery	implicit	-
	TerminateProcess	x	0x000000000003A72A	0x000000000003A72A	1425 (0x0591)	execution	-	implicit	-
	RtlAddFunctionTable	x	0x000000000003A6AC	0x000000000003A6AC	1222 (0x04C6)	exception	-	implicit	-
	RtlLookupFunctionEntry	x	0x000000000003A6D6	0x000000000003A6D6	1230 (0x04CE)	-	-	implicit	-
	getenv	x	0x000000000003A954	0x000000000003A954	975 (0x03CF)	-	-	implicit	-
	DeleteCriticalSection	-	0x000000000003A580	0x000000000003A580	283 (0x011B)	synchronization	-	implicit	-

ascii	4	0x0001A11F	x	-	network	-	recv
ascii	12	0x0001A480	x	-	network	-	InternetOpen
ascii	15	0x0001A48E	x	-	network	-	InternetOpenUrl
ascii	19	0x0001A49F	x	-	network	-	InternetCloseHandle
ascii	6	0x0001B0EC	x	-	network	-	socket
ascii	6	0x0001A10B	x	utility	network	-	select

Interesting strings:
@Mozilla/5.0
@C:\Users\Public\passwrd.txt
@http://cdn.altimiter.local/feed?post=
@Nim httpclient/1.6.2
@Desktop\cosmo.jpeg
@SikoMode
toRC4

VirusTotal results:

38 out of 69 engines

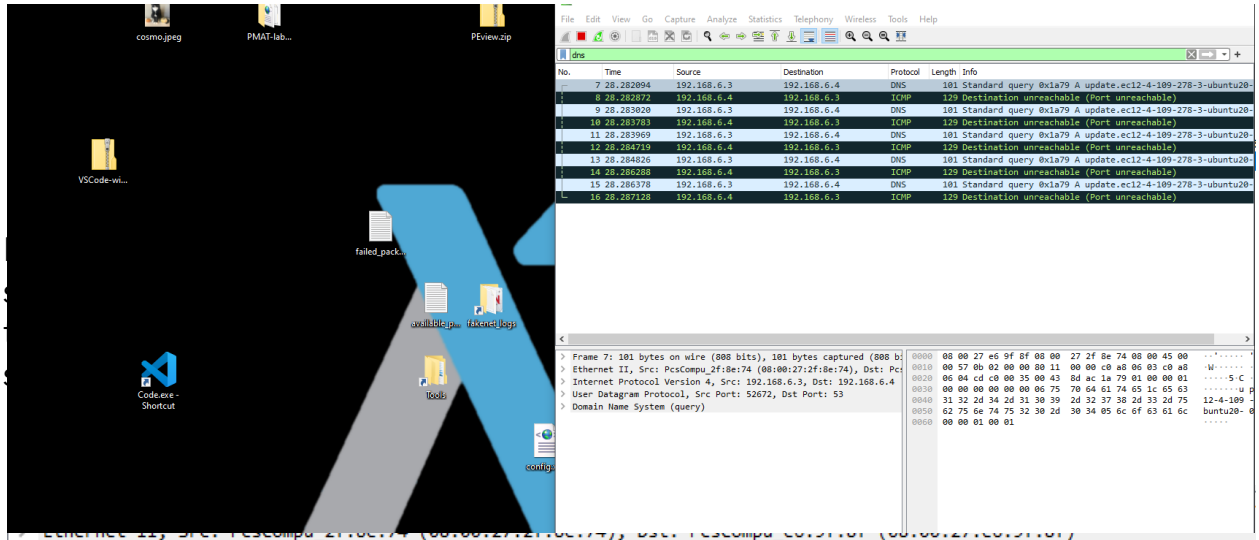
Threat Category: Trojan

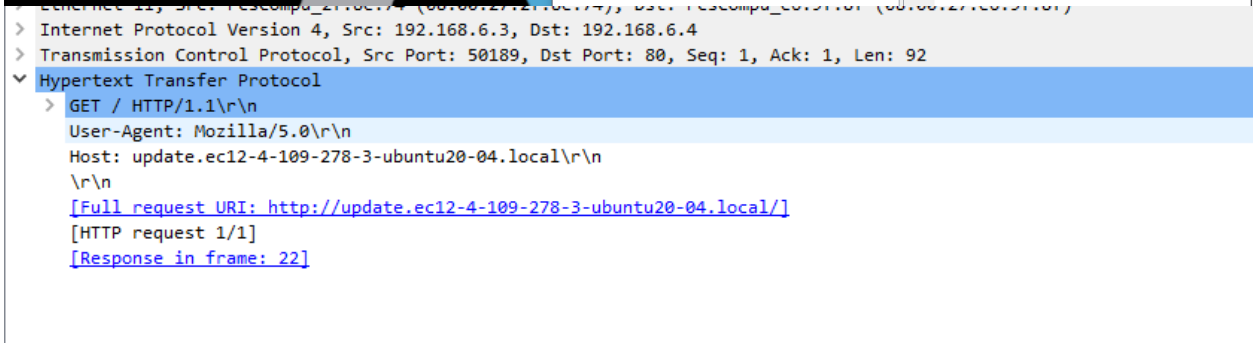
Popular threat label: trojan.pmax/tedy

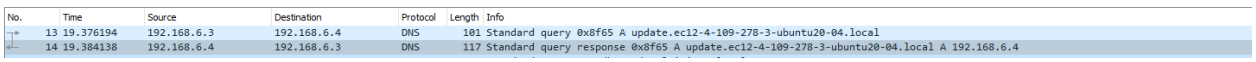
Basic Dynamic Analysis

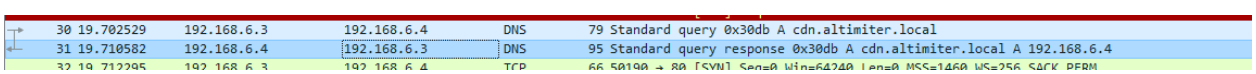
DNS queries: update.ec12-4-109-278-3-ubuntu20-04.local,cnd.altimiter.loca

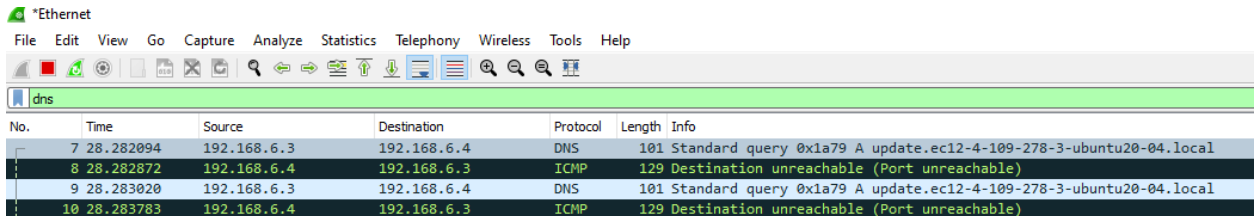
If there is no answer for the first DNS query, then the file deletes itself, otherwise, the deletion occurs at the end of the exfiltration or when one of the other phases goes wrong.









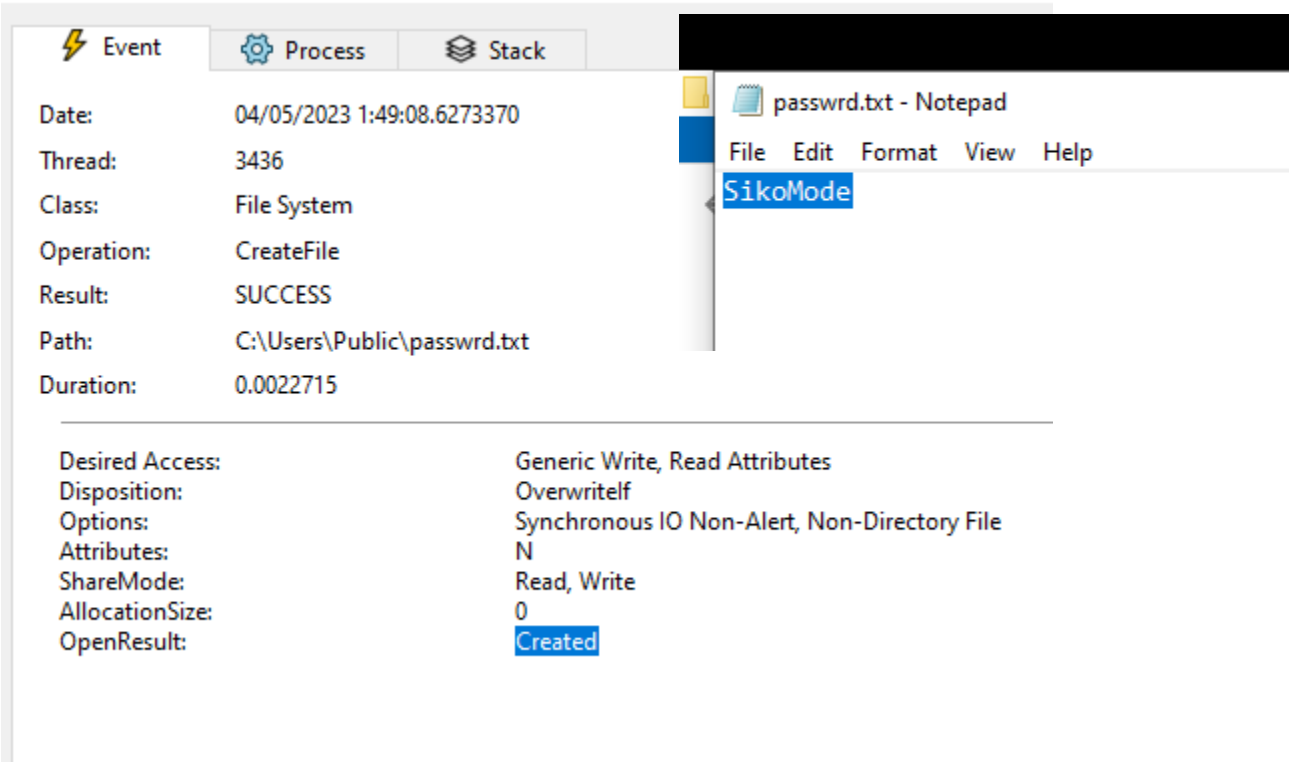


From the process point of view, we can see that it creates the file “passwrd.txt” on the “Public” directory, and then it generates the WriteFile on passwrd.txt, then it accesses cosmo.jpeg and keeps on writing data in passwrd.txt

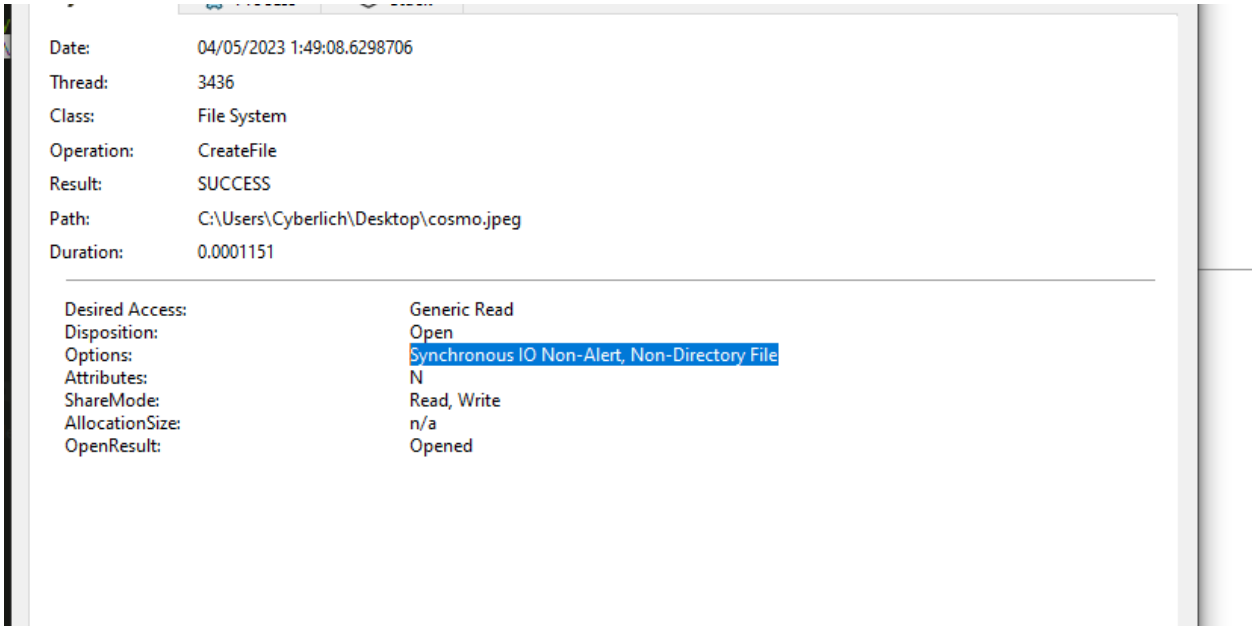
When we check the passwrd.txt file, we see that it contains the phrase “SikoMode”

If cosmo.jpeg doesn’t exist it deletes itself as well

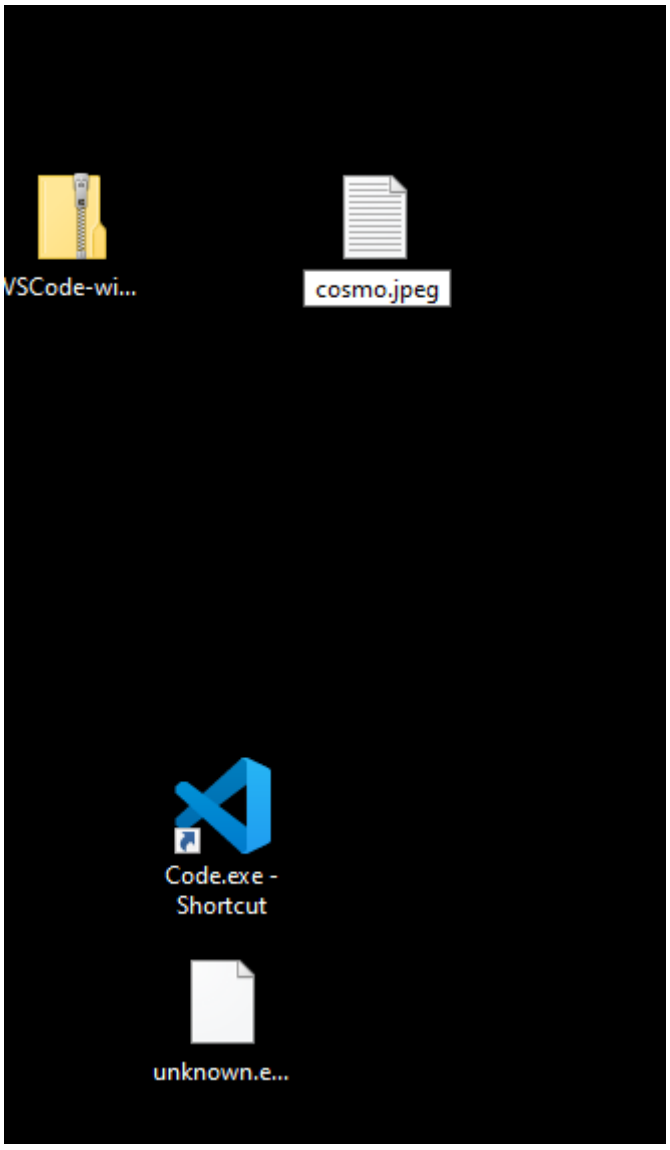
This malware doesn't have child processes or persistence methods.

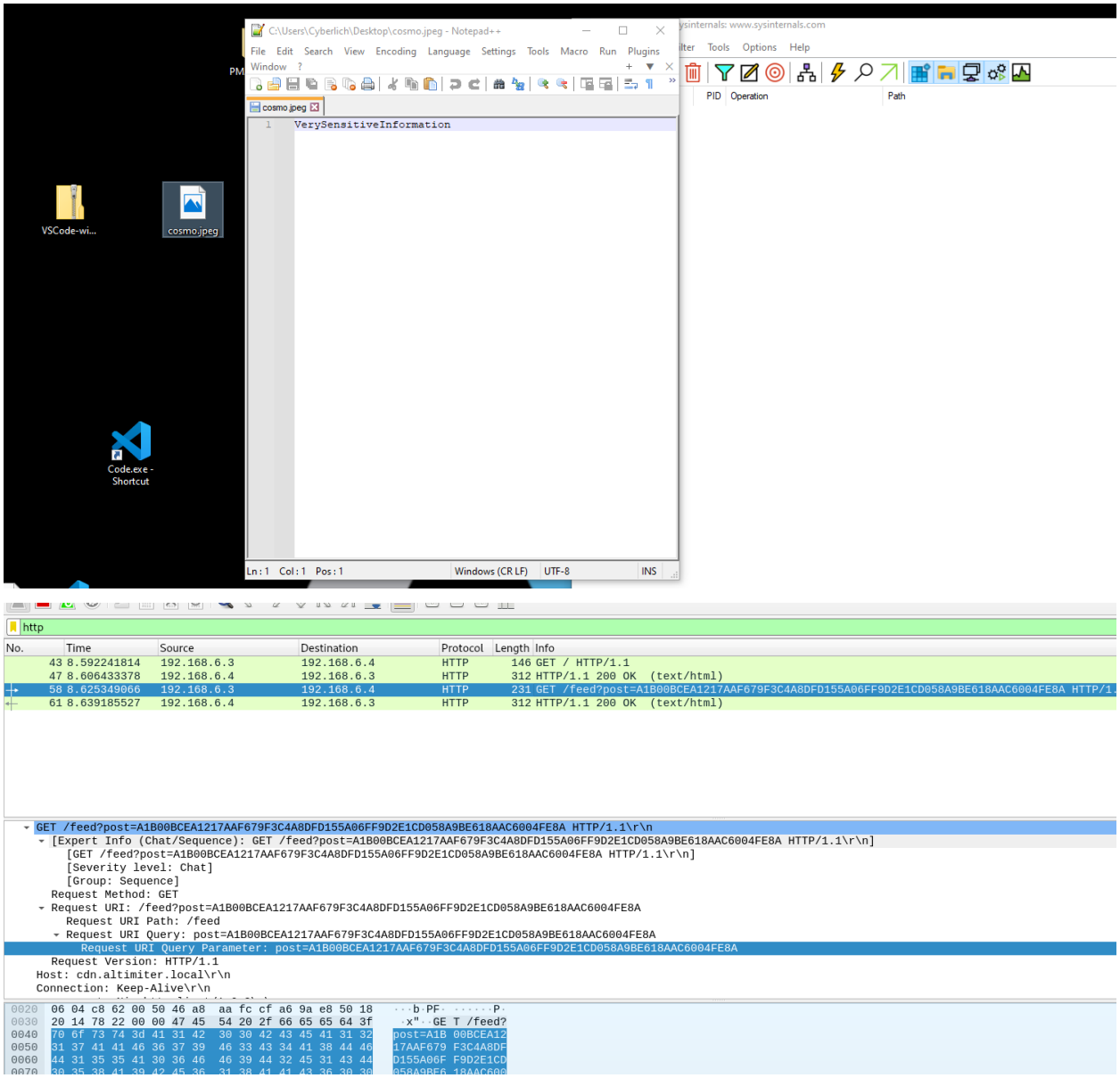


992	CreateFile	C:\Users\Cyberlich\AppData\Local\Microsoft\Windows\INet...	SUCCESS	Des
992	QueryAttributeTagFile	C:\Users\Cyberlich\AppData\Local\Microsoft\Windows\INet...	SUCCESS	Attri
992	CloseFile	C:\Users\Cyberlich\AppData\Local\Microsoft\Windows\INet...	SUCCESS	
992	CreateFile	C:\Users\Public\passwrd.txt	SUCCESS	Des
992	ReadFile	C:\$Secure:$SDH:$INDEX_ALLOCATION	SUCCESS	Offs
992	WriteFile	C:\Users\Public\passwrd.txt	SUCCESS	Offs
992	CloseFile	C:\Users\Public\passwrd.txt	SUCCESS	
992	CreateFile	C:\Users\Cyberlich\Desktop\cosmo.jpeg	SUCCESS	Des
992	QueryStandardInformationFile	C:\Users\Cyberlich\Desktop\cosmo.jpeg	SUCCESS	Allo
992	ReadFile	C:\Users\Cyberlich\Desktop\cosmo.jpeg	SUCCESS	Offs
992	ReadFile	C:\Users\Cyberlich\Desktop\cosmo.jpeg	SUCCESS	Offs
992	ReadFile	C:\Users\Cyberlich\Desktop\cosmo.jpeg	END OF FILE	Offs
992	CloseFile	C:\Users\Cyberlich\Desktop\cosmo.jpeg	SUCCESS	
992	ReadFile	C:\Users\Cyberlich\Desktop\unknown.exe	SUCCESS	Offs
992	ReadFile	C:\Users\Cyberlich\Desktop\unknown.exe	SUCCESS	Offs
992	CreateFile	C:\Users\Public\passwrd.txt	SUCCESS	Des
992	QueryStandardInformationFile	C:\Users\Public\passwrd.txt	SUCCESS	Allo
992	ReadFile	C:\Users\Public\passwrd.txt	SUCCESS	Offs
992	ReadFile	C:\Users\Public\passwrd.txt	END OF FILE	Offs
992	CloseFile	C:\Users\Public\passwrd.txt	SUCCESS	
992	ReadFile	C:\Users\Cyberlich\Desktop\unknown.exe	SUCCESS	Offs


Date: 04/05/2023 1:49:08.6298706
Thread: 3436
Class: File System
Operation: CreateFile
Result: SUCCESS
Path: C:\Users\Cyberlich\Desktop\cosmo.jpeg
Duration: 0.0001151
Desired Access: Generic Read
Disposition: Open
Options: Synchronous IO Non-Alert, Non-Directory File
Attributes: N
ShareMode: Read, Write
AllocationSize: n/a
OpenResult: Opened

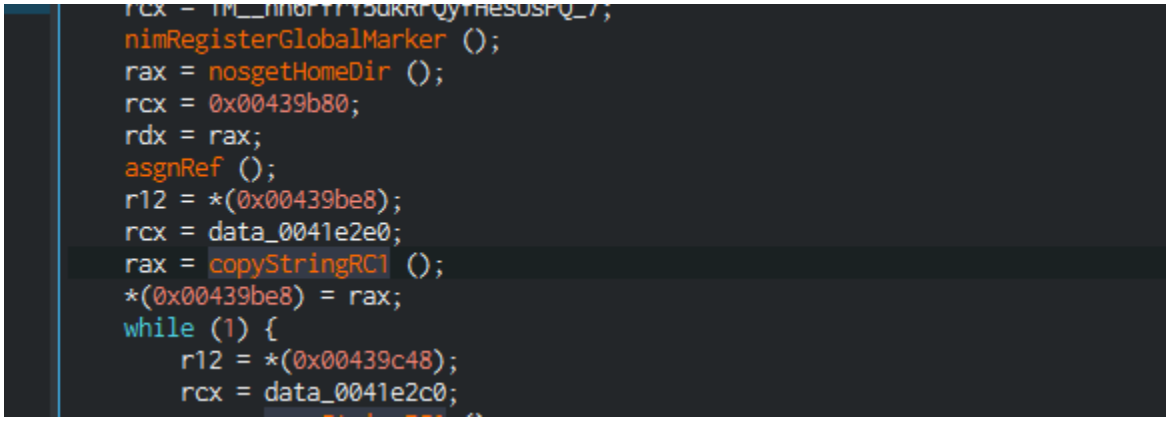
After some checking, it turns out that “cosmo.jpeg” doesn’t have to be a real JPEG file, it can be anything and be changed to “cosmo.jpeg” before the detonation, so this malware can be used as the exfiltration phase of a larger attack where the sensitive data was already written into a file called “cosmo.jpeg” in an earlier phase.



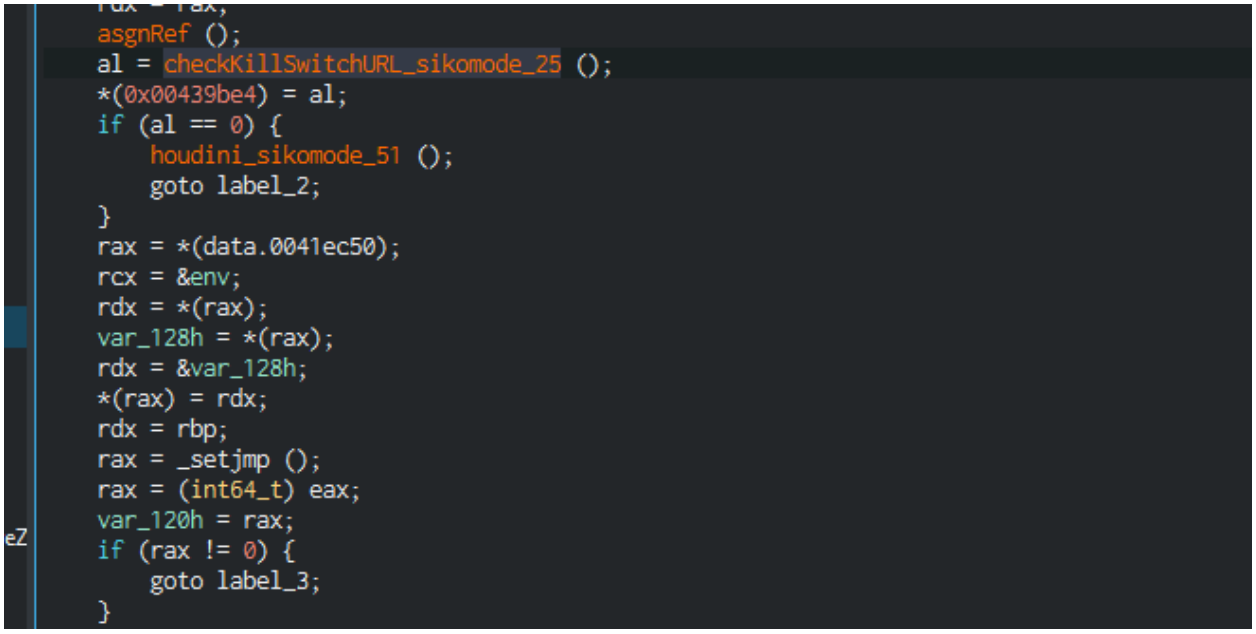

C:\Users\Cyberlich\Desktop\cosmo.jpeg - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins
Window ?
cosmo.jpeg
1 VerySensitiveInformation
Ln : 1 Col : 1 Pos : 1 Windows (CR LF) UTF-8 INS
sysinternals: www.sysinternals.com
Filter Tools Options Help
PID Operation Path
VSCode-wi...
cosmo.jpeg
Code.exe - Shortcut
http
No. Time Source Destination Protocol Length Info
43 8.592241814 192.168.6.3 192.168.6.4 HTTP 146 GET / HTTP/1.1
47 8.606433378 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
58 8.625349066 192.168.6.3 192.168.6.4 HTTP 231 GET /feed?post=A1B00BCEA1217AAF679F3C4A8DFD155A06FF9D2E1CD058A9BE618AAC6004FE8A HTTP/1.
61 8.639185527 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
GET /feed?post=A1B00BCEA1217AAF679F3C4A8DFD155A06FF9D2E1CD058A9BE618AAC6004FE8A HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /feed?post=A1B00BCEA1217AAF679F3C4A8DFD155A06FF9D2E1CD058A9BE618AAC6004FE8A HTTP/1.1\r\n]
[GET /feed?post=A1B00BCEA1217AAF679F3C4A8DFD155A06FF9D2E1CD058A9BE618AAC6004FE8A HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /feed?post=A1B00BCEA1217AAF679F3C4A8DFD155A06FF9D2E1CD058A9BE618AAC6004FE8A
Request URI Path: /feed
Request URI Query: post=A1B00BCEA1217AAF679F3C4A8DFD155A06FF9D2E1CD058A9BE618AAC6004FE8A
Request URI Query Parameter: post=A1B00BCEA1217AAF679F3C4A8DFD155A06FF9D2E1CD058A9BE618AAC6004FE8A
Request Version: HTTP/1.1
Host: cdn.altimiter.local\r\n
Connection: Keep-Alive\r\n
0020 06 04 c8 62 00 50 46 a8 aa fc cf a6 9a e8 50 18 ···b·PF· ·····P·
0030 20 14 78 22 00 00 47 45 54 20 2f 66 65 65 64 3f ·x"··GE T /feed?
0040 70 6f 73 74 3d 41 31 42 30 30 42 43 45 41 31 32 post=A1B 00BCEA12
0050 31 37 41 41 46 36 37 39 46 33 43 34 41 38 44 46 17AAF679 F3C4A8DF
0060 44 31 35 35 41 30 36 46 46 39 44 32 45 31 43 44 D155A06F F9D2E1CD
0070 30 35 38 41 39 42 45 36 31 38 41 41 43 36 30 30 058A9BE6 18AAC600

Advanced Static Analysis

In the Decompiler we can see that there is a copyStringRC1 function, which is probably related to the malware's encryption feature.

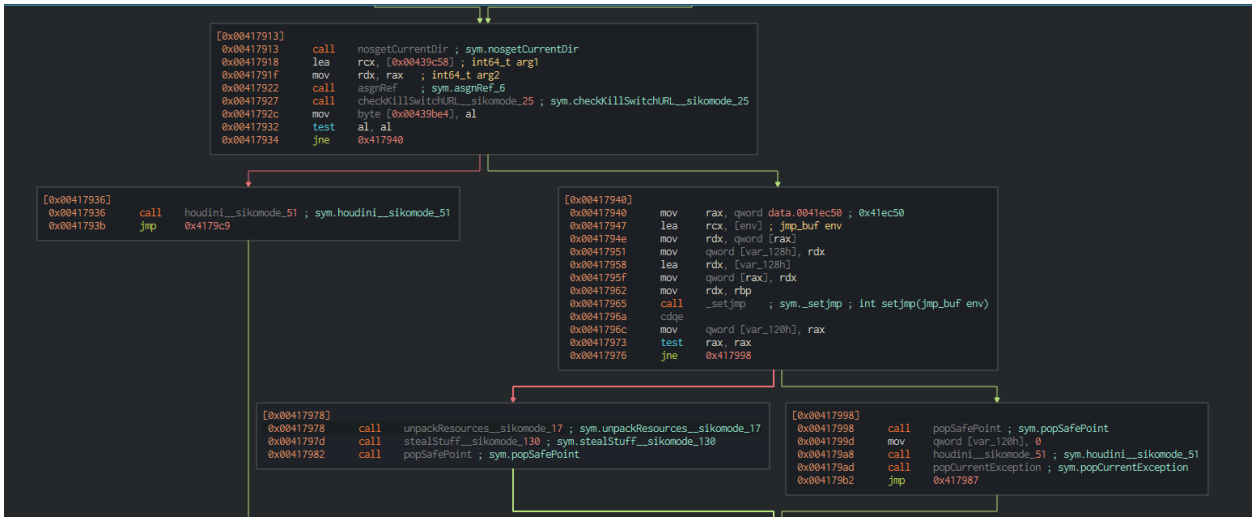


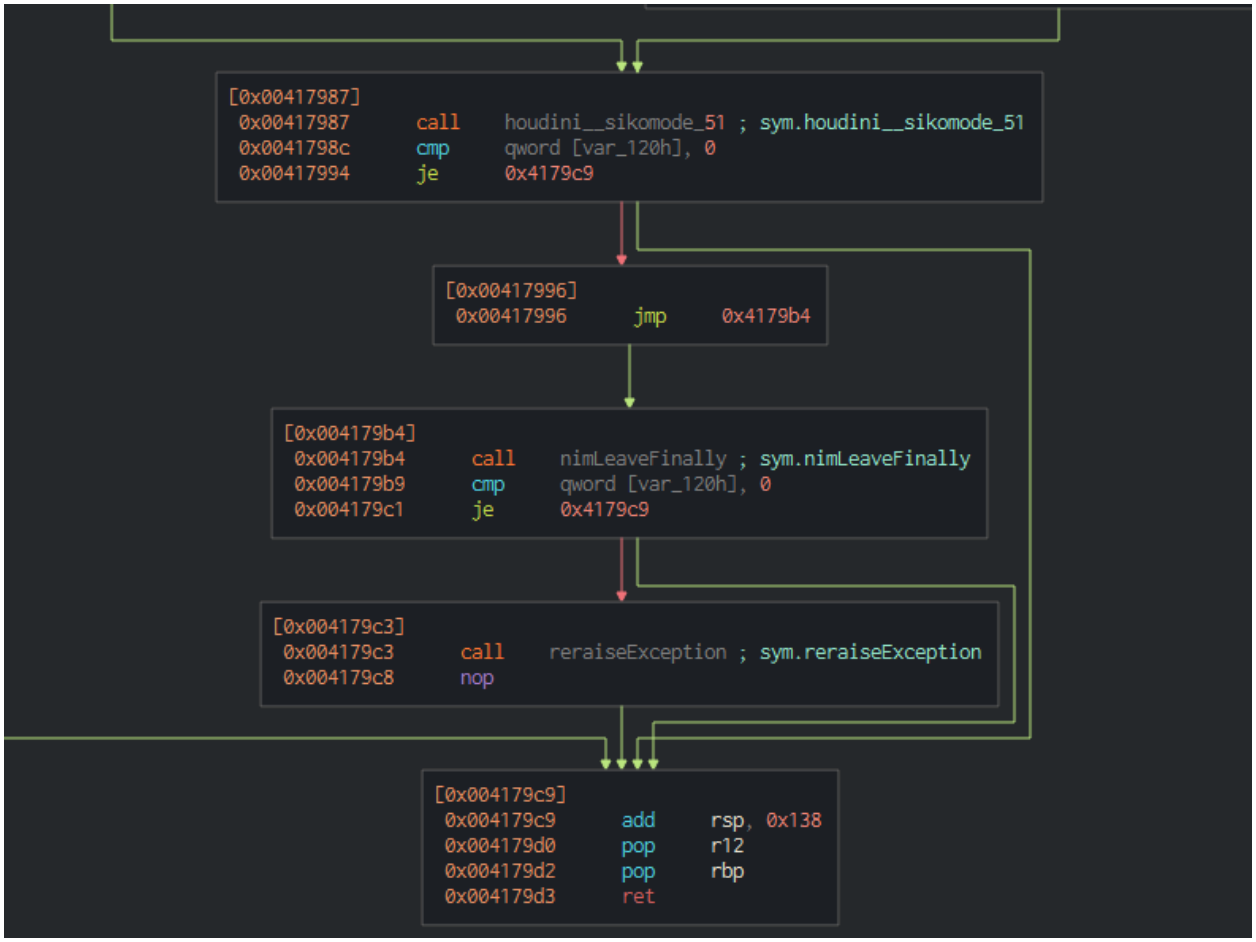
we can also see the CheckKillSwitchURL_sikemode_25 functnion which probably related to the first DNS query,

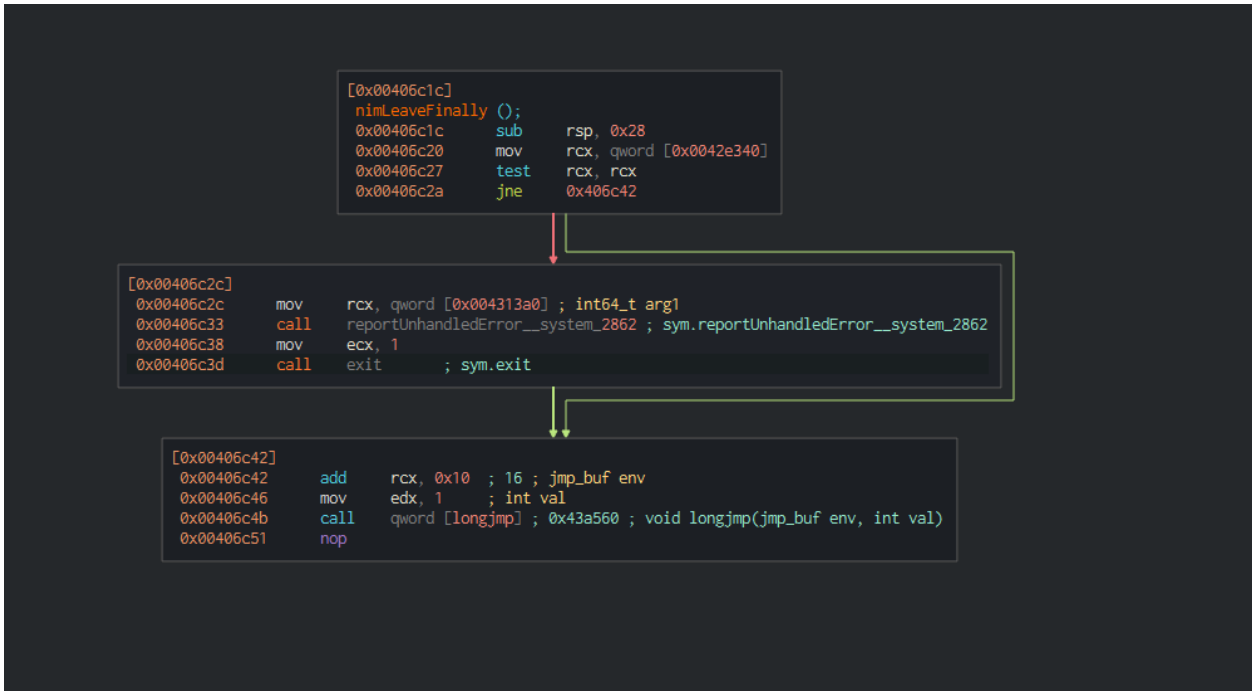


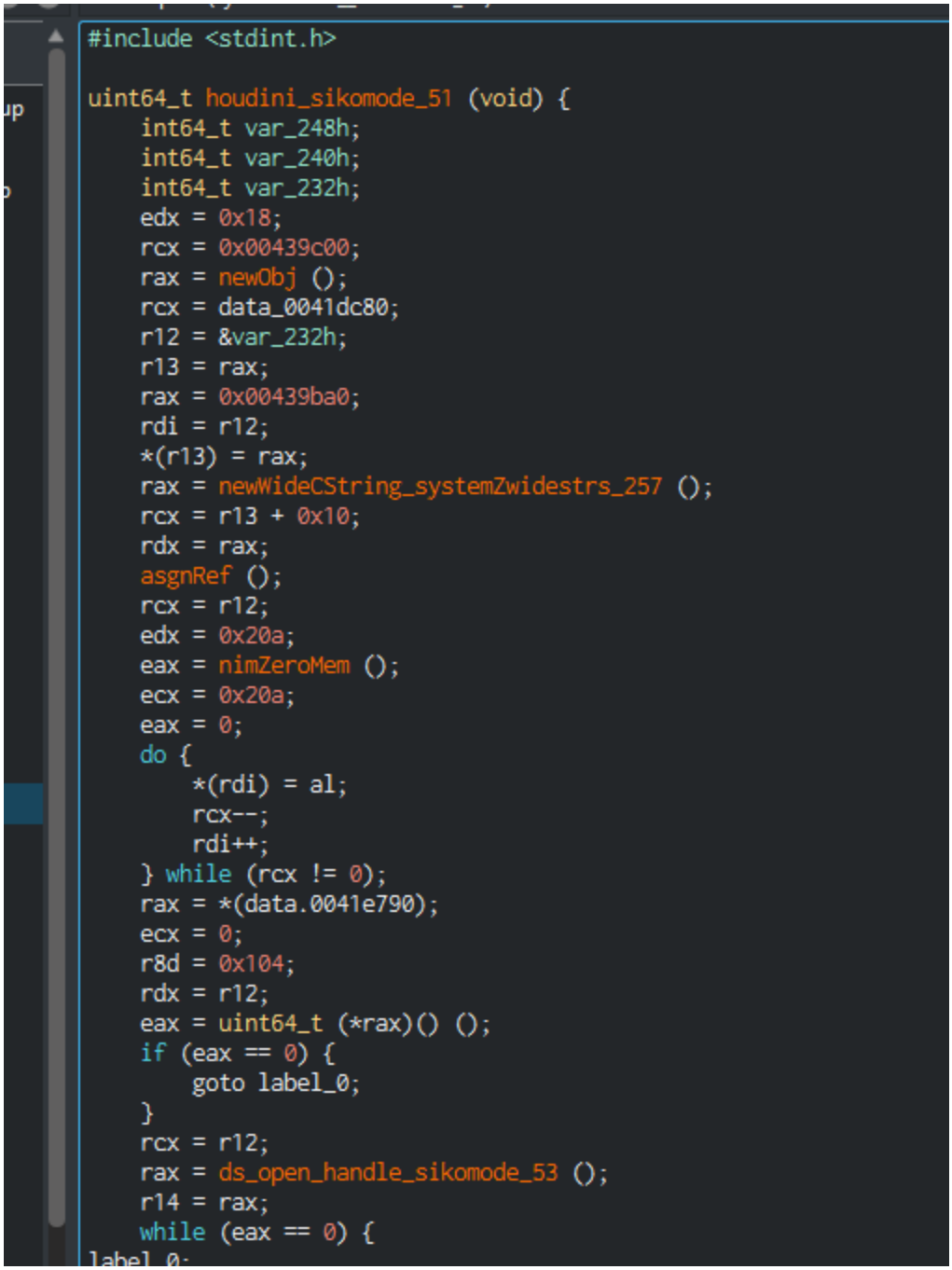
We can see the functions unpackResources and stealStuff which sound pretty interesting.

We can also see the “houdini_sikemode_51” function, which seems to be the function that closes the program




[0x00417987]
0x00417987 call houdini__sikomode_51 ; sym.houdini__sikomode_51
0x0041798c cmp qword [var_120h], 0
0x00417994 je 0x4179c9
[0x00417996]
0x00417996 jmp 0x4179b4
[0x004179b4]
0x004179b4 call nimLeaveFinally ; sym.nimLeaveFinally
0x004179b9 cmp qword [var_120h], 0
0x004179c1 je 0x4179c9
[0x004179c3]
0x004179c3 call reraiseException ; sym.reraiseException
0x004179c8 nop
[0x004179c9]
0x004179c9 add rsp, 0x138
0x004179d0 pop r12
0x004179d2 pop rbp
0x004179d3 ret


[0x00406c1c]
nimLeaveFinally ();
0x00406c1c sub rsp, 0x28
0x00406c20 mov rcx, qword [0x0042e340]
0x00406c27 test rcx, rcx
0x00406c2a jne 0x406c42
[0x00406c2c]
0x00406c2c mov rcx, qword [0x004313a0] ; int64_t arg1
0x00406c33 call reportUnhandledError__system_2862 ; sym.reportUnhandledError__system_2862
0x00406c38 mov ecx, 1
0x00406c3d call exit ; sym.exit
[0x00406c42]
0x00406c42 add rcx, 0x10 ; 16 ; jmp_buf env
0x00406c46 mov edx, 1 ; int val
0x00406c4b call qword [longjmp] ; 0x43a560 ; void longjmp(jmp_buf env, int val)
0x00406c51 nop



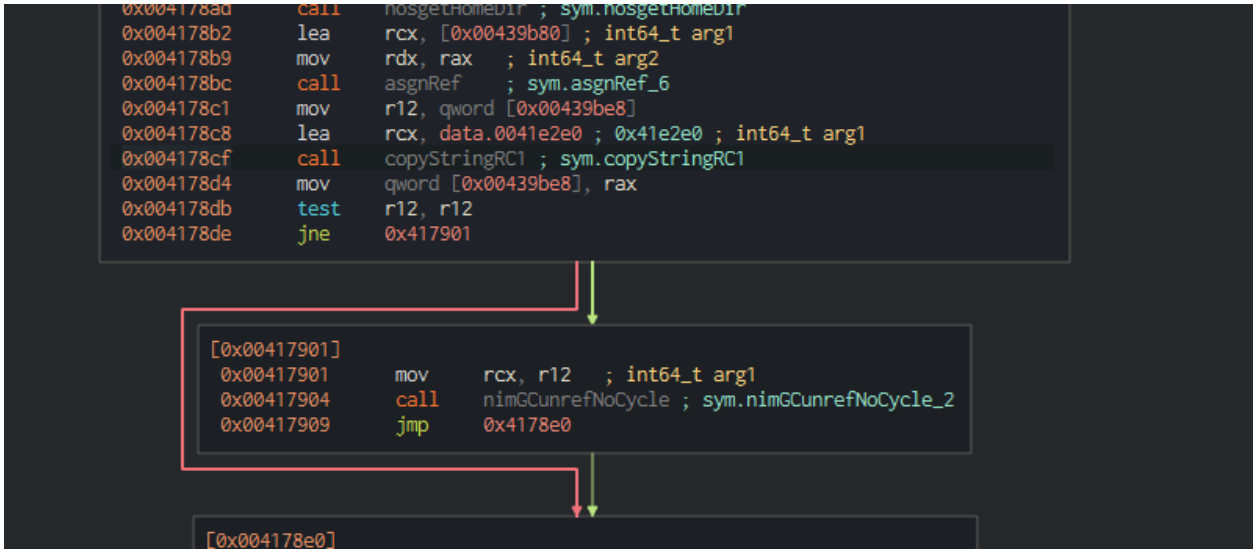
```c
#include <stdint.h>

uint64_t houdini_sikomode_51 (void) {
    int64_t var_248h;
    int64_t var_240h;
    int64_t var_232h;
    edx = 0x18;
    rcx = 0x00439c00;
    rax = newObj ();
    rcx = data_0041dc80;
    r12 = &var_232h;
    r13 = rax;
    rax = 0x00439ba0;
    rdi = r12;
    *(r13) = rax;
    rax = newWideCString_systemZwidestrs_257 ();
    rcx = r13 + 0x10;
    rdx = rax;
    asgnRef ();
    rcx = r12;
    edx = 0x20a;
    eax = nimZeroMem ();
    ecx = 0x20a;
    eax = 0;
    do {
        *(rdi) = al;
        rcx--;
        rdi++;
    } while (rcx != 0);
    rax = *(data.0041e790);
    ecx = 0;
    r8d = 0x104;
    rdx = r12;
    eax = uint64_t (*rax)() ();
    if (eax == 0) {
        goto label_0;
    }
    rcx = r12;
    rax = ds_open_handle_sikomode_53 ();
    r14 = rax;
    while (eax == 0) {
label_0:
```

Advanced Dynamic Analysis

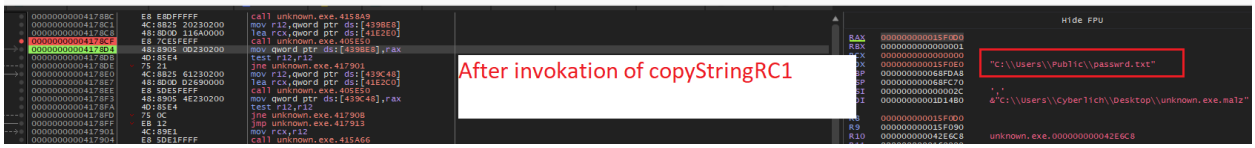
After examining the main function of the code, I copied the addresses of the important functions from Cutter and put breakpoints to see them in action.

Looking into the main function structure on Cutter, we see the “copyStringCR1” function

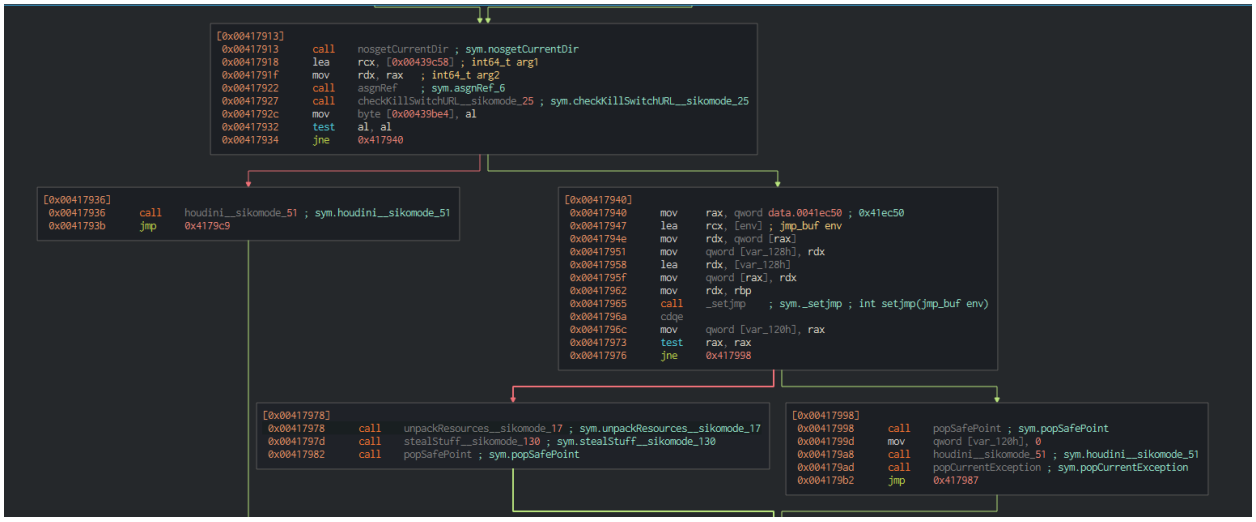
I set a breakpoint on that as well and found out that the function uses the “passwrd.txt” file, so the string for the RC4 encryption key will be the content of passwrd.txt which is “SikoMode”.



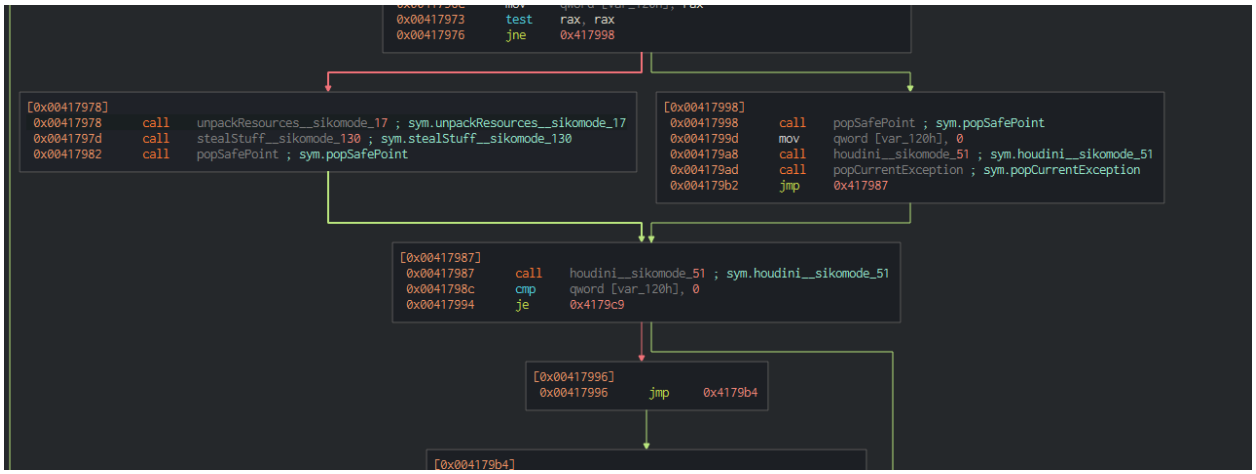
```
[0x00405dfe]
 copyStringRC1.part.0 (int64_t arg1);
 ; arg int64_t arg1 @ rcx
 0x00405dfe      push    rdi
 0x00405dff      push    rsi
 0x00405e00      push    rbx
 0x00405e01      sub     rsp, 0x20
 0x00405e05      mov     eax, 7
 0x00405e0a      mov     rbx, qword [rcx] ; arg1
 0x00405e0d      cmp     rbx, 7     ; 7
 0x00405e11      cmovl   rbx, rax
 0x00405e15      mov     rsi, rcx   ; arg1
 0x00405e18      lea     rcx, data.0041a020 ; 0x41a020 ; int64_t arg1
 0x00405e1f      lea     rdx, [rbx + 0x11] ; int64_t arg2
 0x00405e23      add     rsi, 0x10  ; 16
 0x00405e27      call    newObjRC1  ; sym.newObjRC1
 0x00405e2c      mov     rdx, qword [rsi - 0x10]
 0x00405e30      mov     qword [rax + 8], rbx
 0x00405e34      mov     qword [rax], rdx
 0x00405e37      mov     rdi, qword [rsi - 0x10]
 0x00405e3b      lea     rdx, [rax + 0x10]
 0x00405e3f      lea     rcx, [rdi + 1]
 0x00405e43      mov     rdi, rdx
 0x00405e46      rep     movsb byte [rdi], byte ptr [rsi]
 0x00405e48      add     rsp, 0x20
 0x00405e4c      pop     rbx
 0x00405e4d      pop     rsi
 0x00405e4e      pop     rdi
 0x00405e4f      ret
```

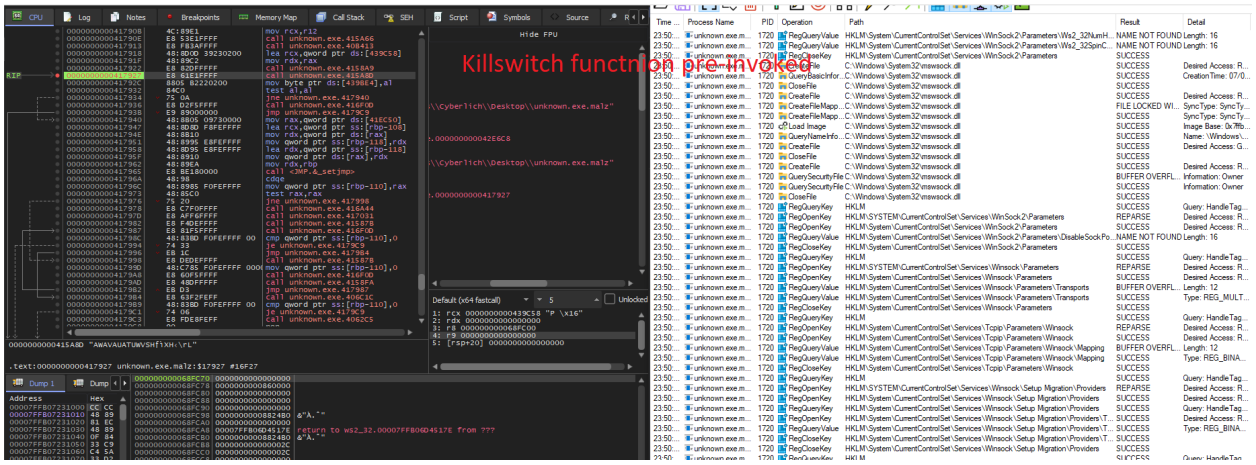


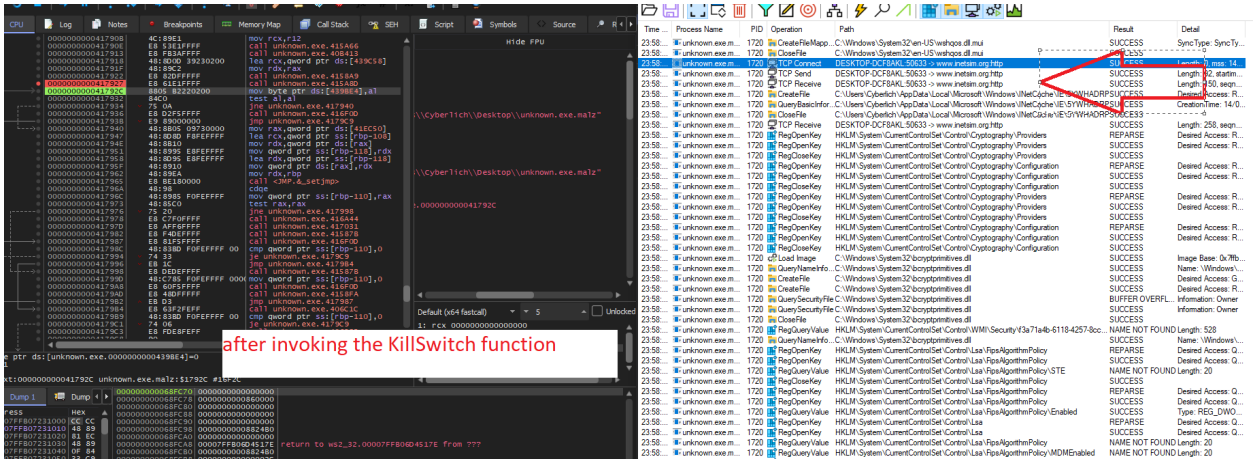
Then we see the Killswitch function, which connects to the first URL “update.ec12-4-109-278-3-ubuntu20-04.local” and if it gets answered it then proceeds to the “unpackResources” function, otherwise it jumps to the “houdini” function to terminate the program

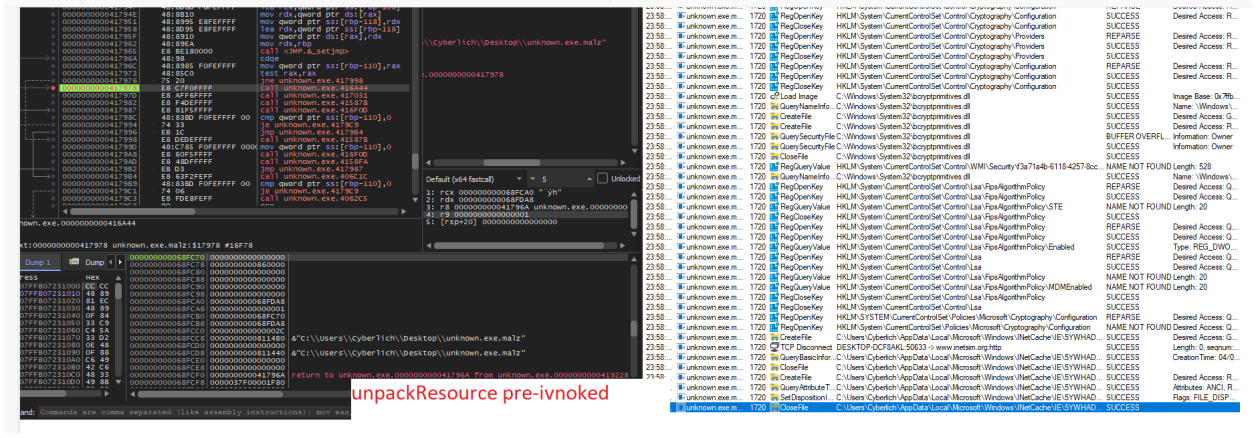


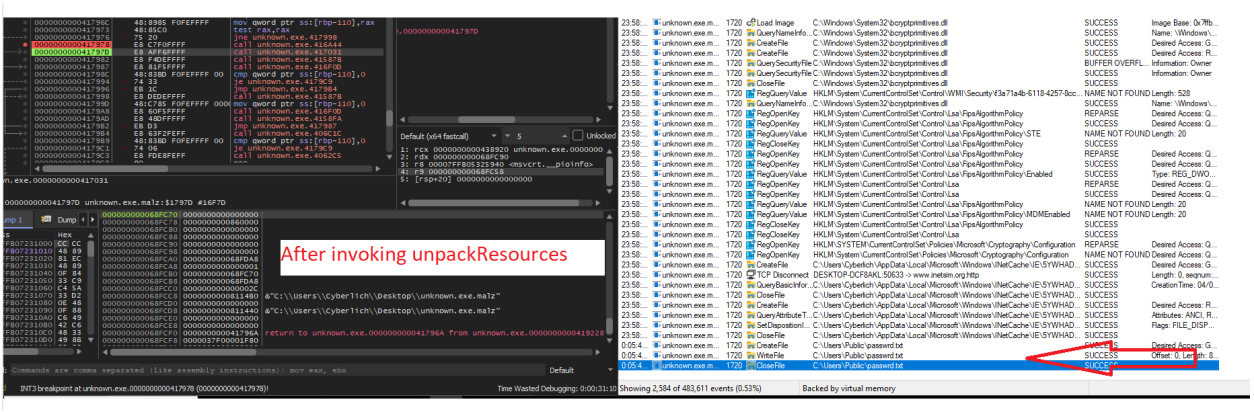
We can see the the “unpackResources” function is followed by the “stealstuff” function and then the popSafePoint that make the jump to the houdini function




Killswitch function pre-invoked

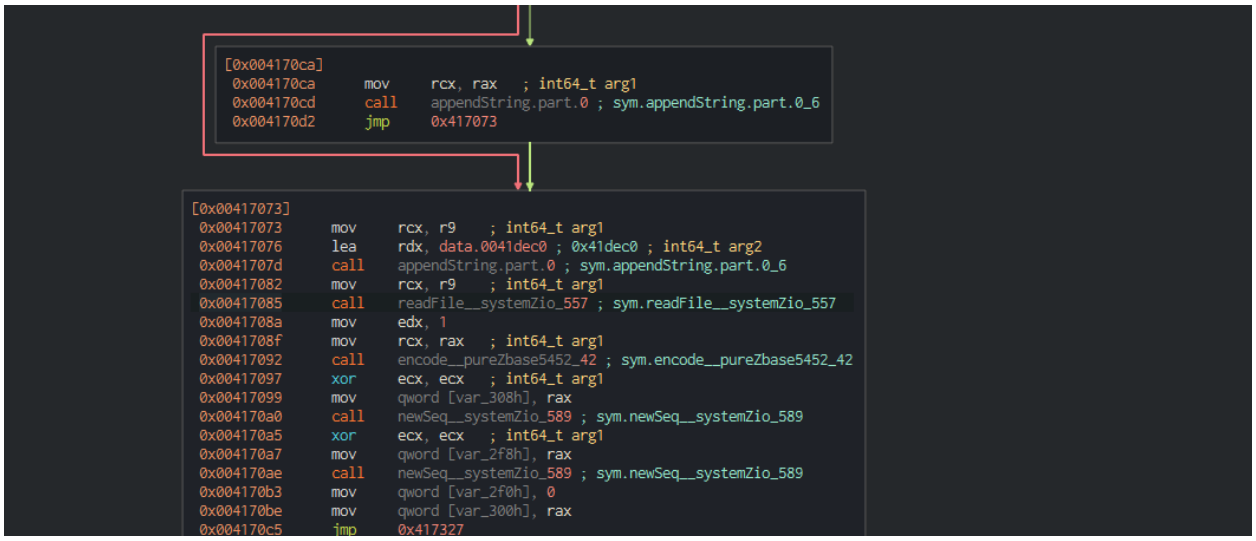

after invoking the KillSwitch function

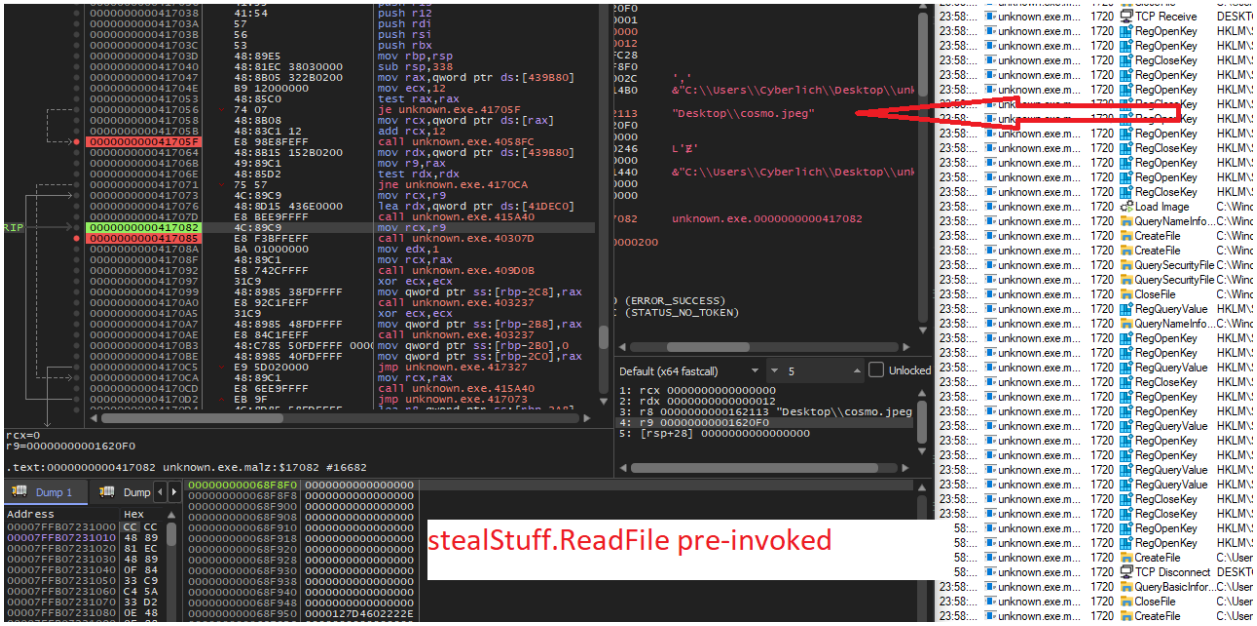

unpackResource pre-ivnoked



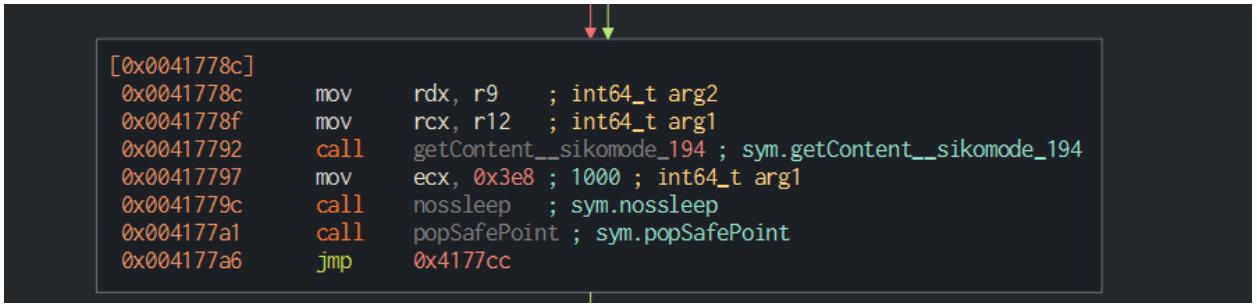
The stealStuff function is pretty complex, we can see that it has different kinds of functions and conditions inside it, there are some interesting functions that we'd like to follow

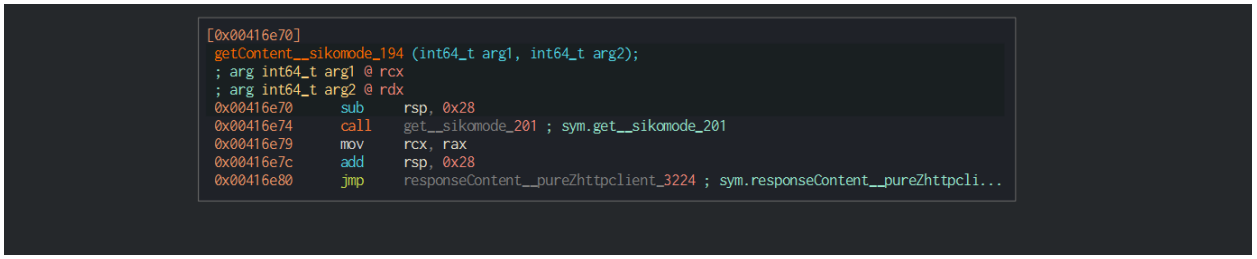
ReadFile - we can see that it sets the values for the exfiltration, it sets the "cosmo.jpeg" string for later use and writes data into the "passwrd.txt" file

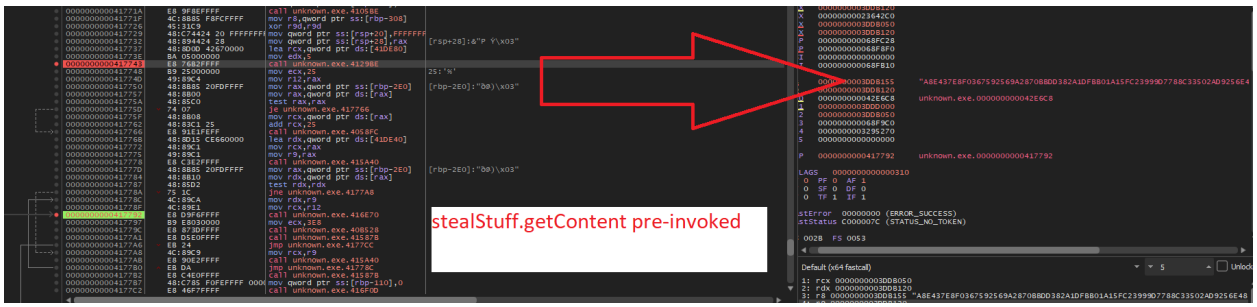


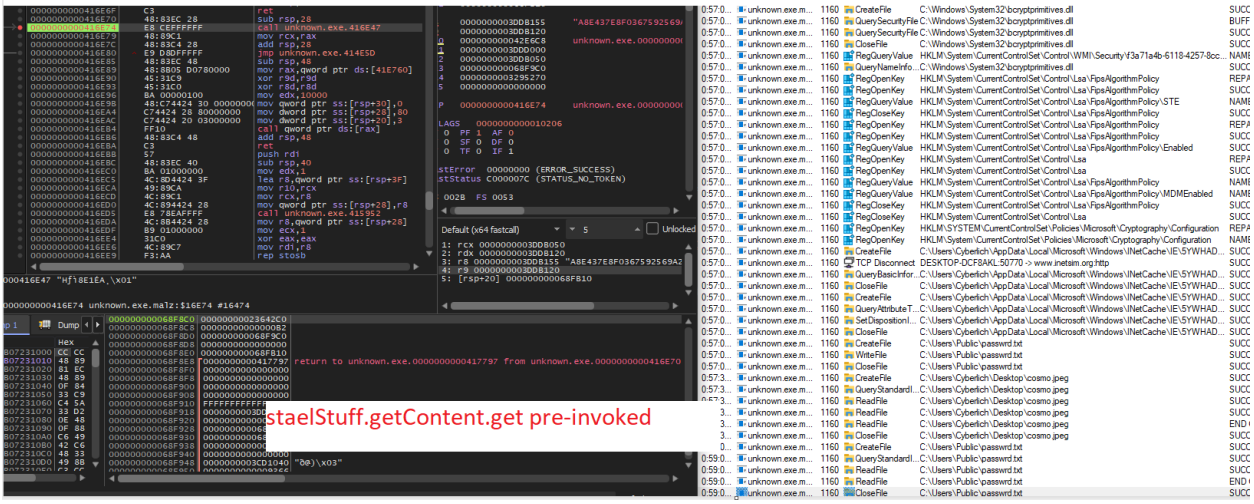


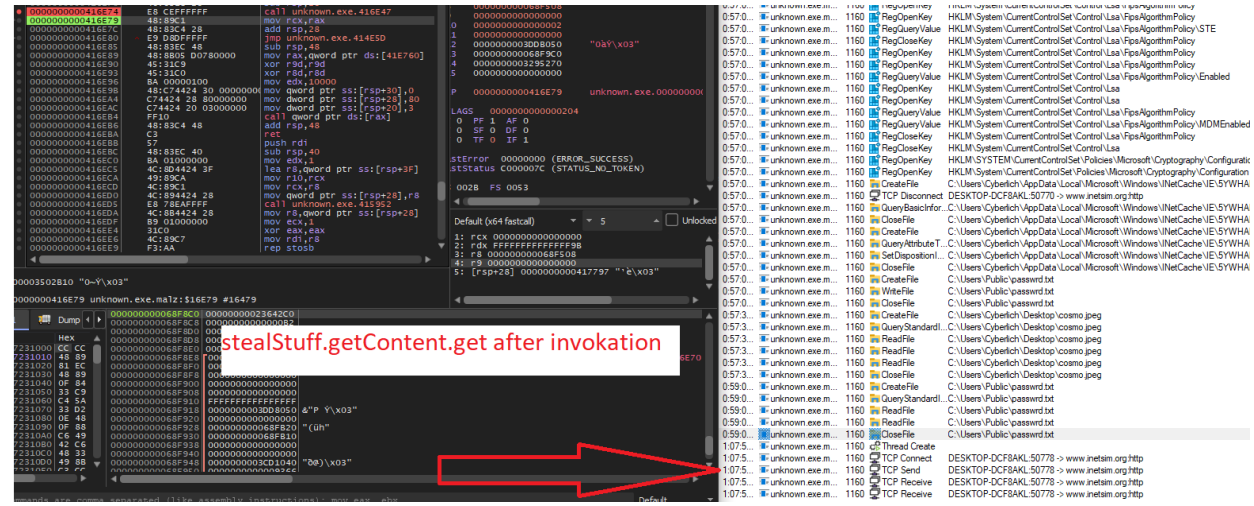
Another interesting part is the getContent function, we can see that once we get there we see the content of the first GET request “post” value, it has two parts inside it, the “get” function and a jump to the responseContent function

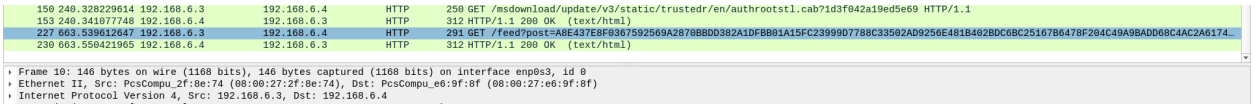




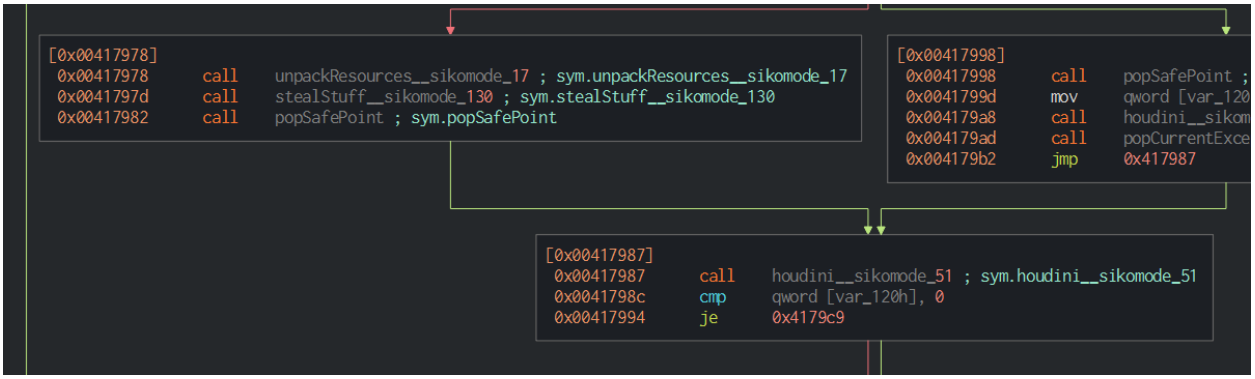


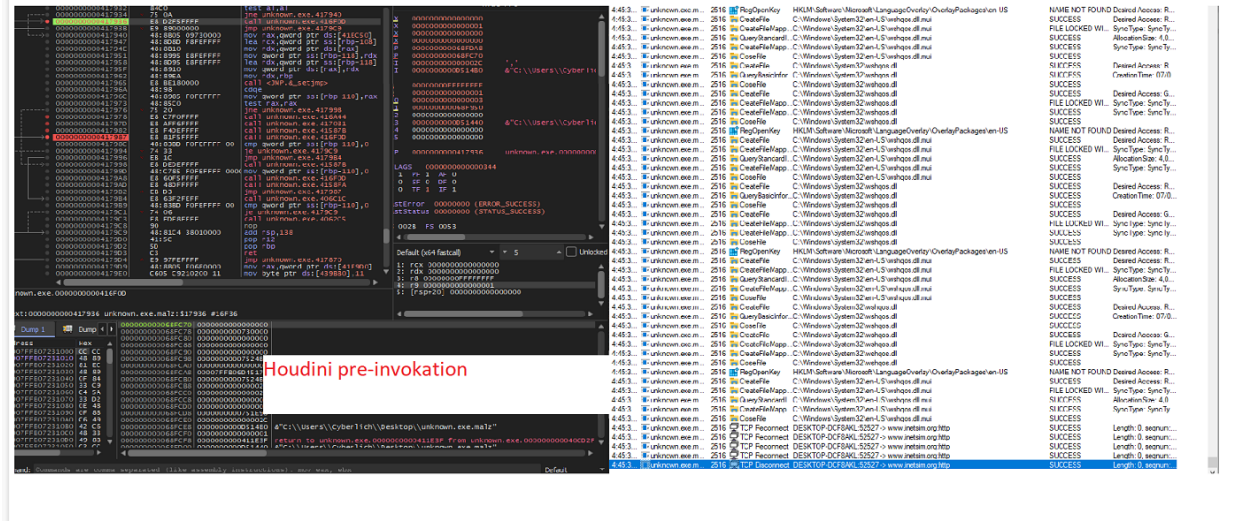


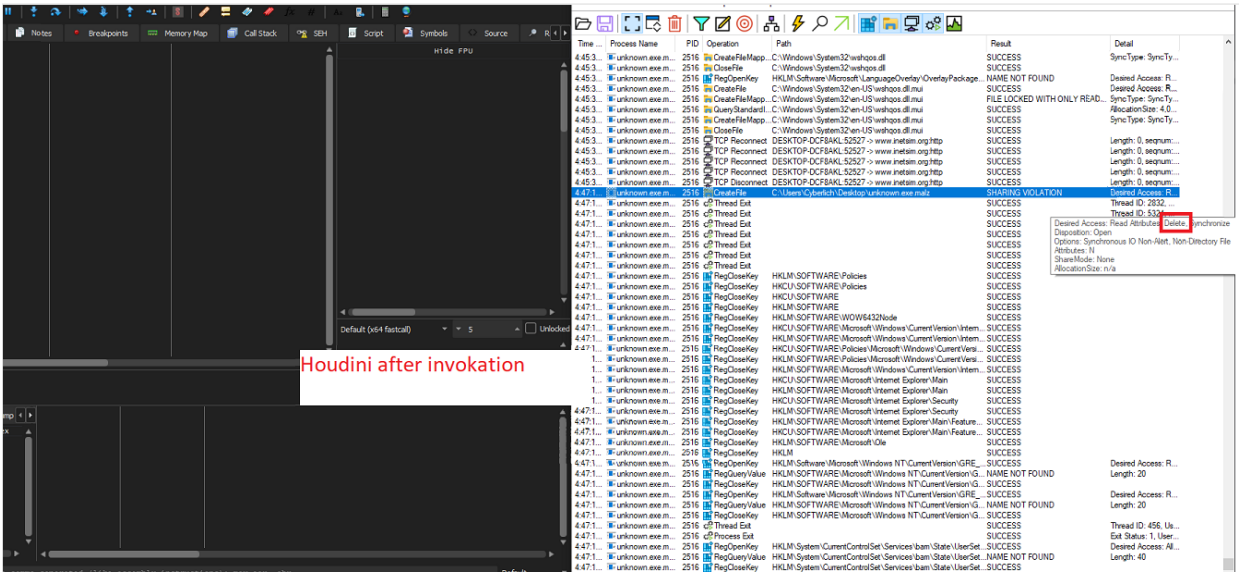




Then, after the exfiltration we have the popSafePoint function and then the “houdini” function, where the file is attempting to delete itself (this time it didn’t manage to do so since the detonation took place by the debugger)




Houdini pre-invokation


Houdini after invokation

Indicators of Compromise

Network Indicators

DNS queries: update.ec12-4-109-278-3-ubuntu20-04.local (Fig 1), cdn.altimiter.local (Fig 2)

Http communication with the first domain, a simple GET request with the “Mozilla/5.0” user-agent (Fig 3)

Http communication with the second domain, with the user-agent “Nim http-client” , sending GET requests with the parameter “post” with a value of 128 hexadecimal characters string to the “/feed” page. (Fig 4)

Fig 1

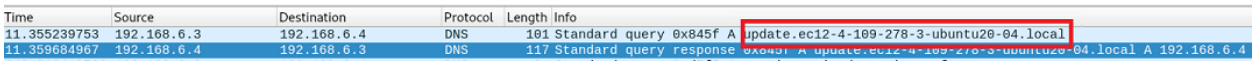


Fig 2

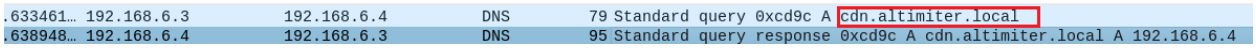


Fig 3

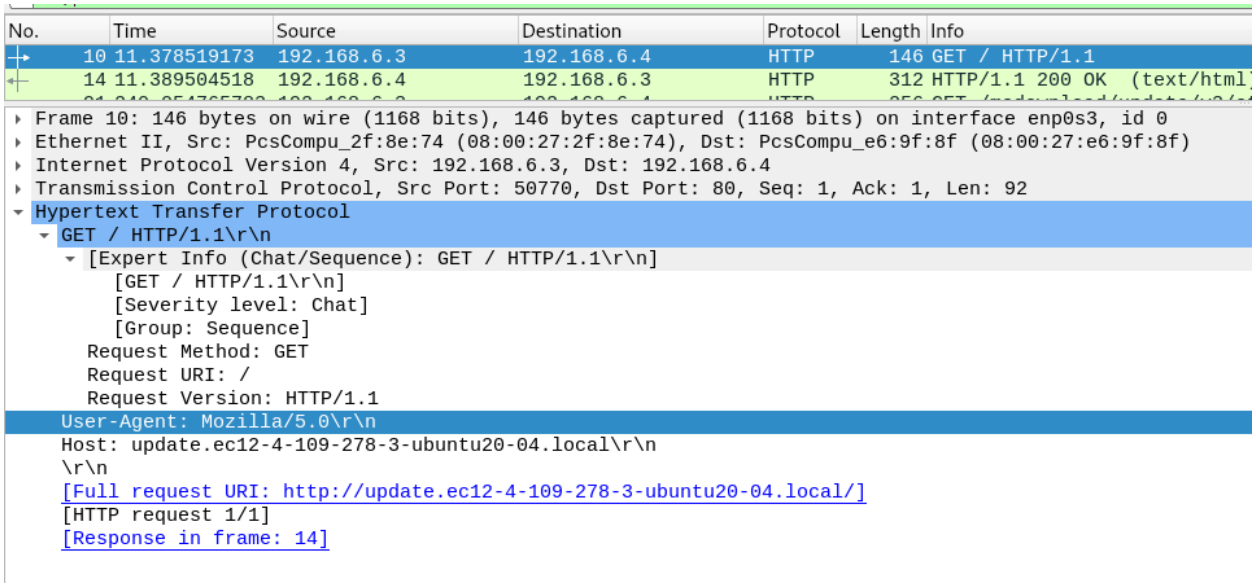
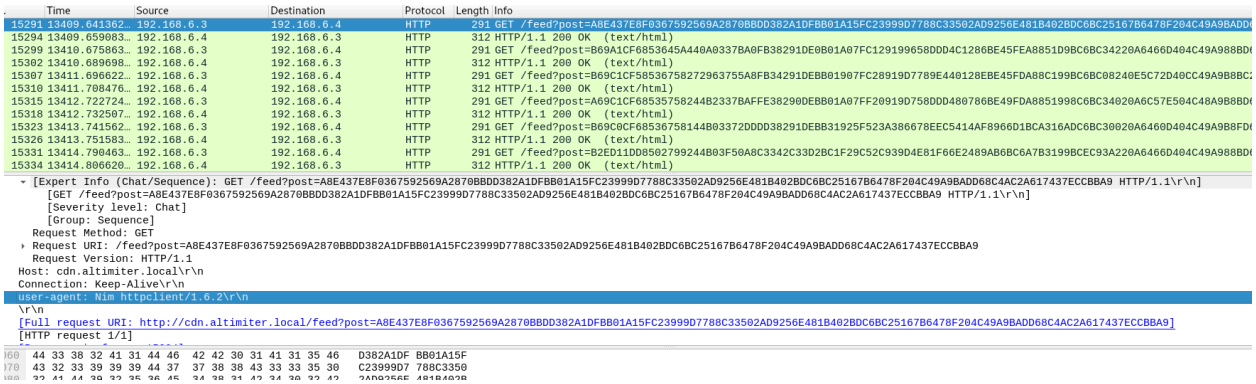

No. Time Source Destination Protocol Length Info
10 11.378519173 192.168.6.3 192.168.6.4 HTTP 146 GET / HTTP/1.1
14 11.389504518 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
Frame 10: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_2f:8e:74 (08:00:27:2f:8e:74), Dst: PcsCompu_e6:9f:8f (08:00:27:e6:9f:8f)
Internet Protocol Version 4, Src: 192.168.6.3, Dst: 192.168.6.4
Transmission Control Protocol, Src Port: 50770, Dst Port: 80, Seq: 1, Ack: 1, Len: 92
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
[GET / HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
User-Agent: Mozilla/5.0\r\n
Host: update.ec12-4-109-278-3-ubuntu20-04.local\r\n
\r\n
[Full request URI: http://update.ec12-4-109-278-3-ubuntu20-04.local/]
[HTTP request 1/1]
[Response in frame: 14]

Fig 4


Time Source Destination Protocol Length Info
15291 13409.641362… 192.168.6.3 192.168.6.4 HTTP 291 GET /feed?post=A8E437E8F0367592569A2870BBDD382A1DFBB01A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD
15294 13409.659083… 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
15299 13410.675863… 192.168.6.3 192.168.6.4 HTTP 291 GET /feed?post=B69A1CF68536645A440A0337BA0FB38291DE0B01A07FC129199658DDD4C1286BE45FEA8851D9BC6BC34220A6466D404C49A988BD
15302 13410.689698… 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
15307 13411.696622… 192.168.6.3 192.168.6.4 HTTP 291 GET /feed?post=B69C1CF585367582729637 55A8FB34291DEBB01907FC28919D7789E440128EBE45FDA88C199BC6BC08240E5C72D40CC49A9B8BC
15310 13411.708476… 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
15315 13412.722724… 192.168.6.3 192.168.6.4 HTTP 291 GET /feed?post=A69C1CF68535758244B2337BAFFE38290DEBB01A07FF20919D758DDD480786BE49FDA8851998C6BC34020A6C57E504C48A9B8BD
15318 13412.732507… 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
15323 13413.741562… 192.168.6.3 192.168.6.4 HTTP 291 GET /feed?post=B69C0CF68536758144B03372DDDD38291DEBB31925F523A386678EEC5414AF8966D1BCA316ADC6BC30020A6460D404C49A9B8FD
15326 13413.751583… 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
15331 13414.790463… 192.168.6.3 192.168.6.4 HTTP 291 GET /feed?post=B2ED11DD8502799244B03F50A8C3342C33D2BC1F29C52C939D4E81F66E2489AB6BC6A7B3199BCEC93A220A6466D404C49A988BD
15334 13414.806620… 192.168.6.4 192.168.6.3 HTTP 312 HTTP/1.1 200 OK (text/html)
[Expert Info (Chat/Sequence): GET /feed?post=A8E437E8F0367592569A2870BBDD382A1DFBB01A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9 HTTP/1.1\r\n]
[GET /feed?post=A8E437E8F0367592569A2870BBDD382A1DFBB01A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9 HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /feed?post=A8E437E8F0367592569A2870BBDD382A1DFBB01A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9
Request Version: HTTP/1.1
Host: cdn.altimiter.local\r\n
Connection: Keep-Alive\r\n
user-agent: Nim httpclient/1.6.2\r\n
\r\n
[Full request URI: http://cdn.altimiter.local/feed?post=A8E437E8F0367592569A2870BBDD382A1DFBB01A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9]
[HTTP request 1/1]
44 33 38 32 41 31 44 46 42 42 30 31 41 31 35 46 D382A1DF BB01A15F
43 32 33 39 39 44 37 37 38 38 43 33 33 35 30 C23999D7 788C3350

Host-based Indicators

Existence of unknown.exe sha 256 -6c8f50040545d8cd9af4b51564de654266e592e3

File creation named password.txt
sha256 - 1eebfcf7b68b2b4ffe17696800740e199acf207afb5514bc51298c2fe7584410 (Fig 1)

Searching and Reading a file named cosmo.jpeg sha256 - 2b43cd921a96b83fb73ea8fdfd645443d58573b1a5ff31d5531ec29cb3366d7 (Fig 2)

File deletes itself (Fig 3)

Fig 1

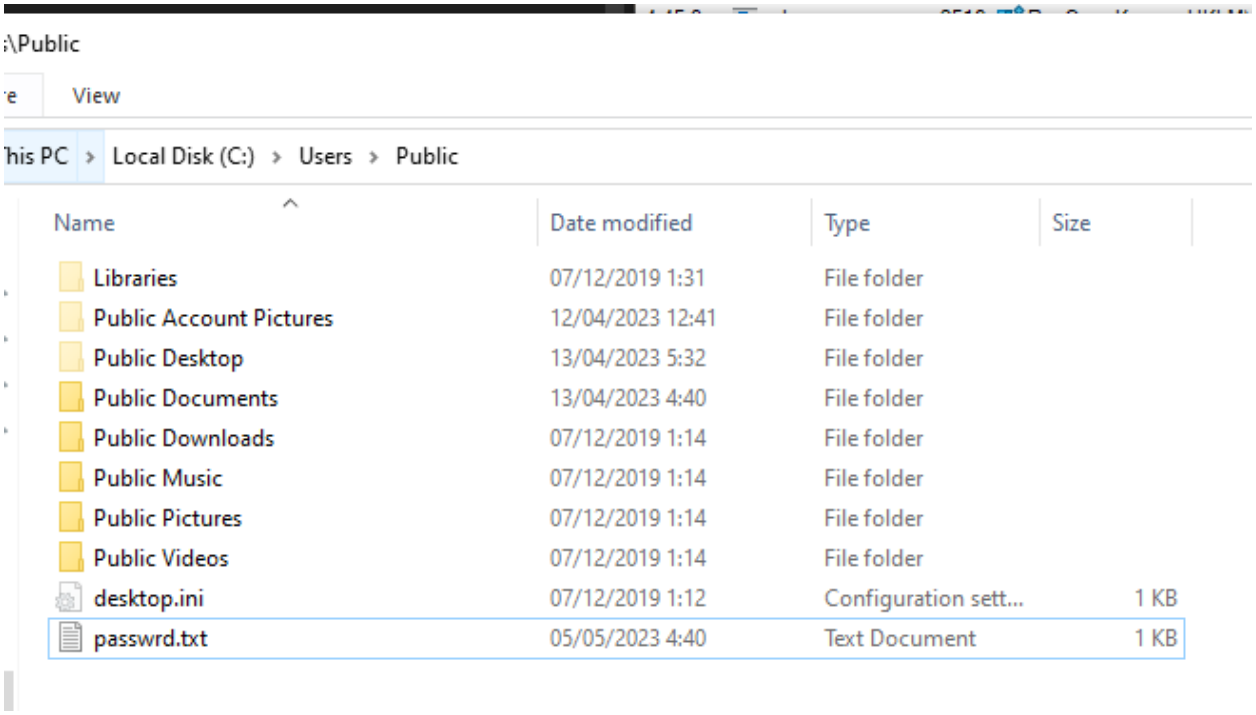


Fig 2

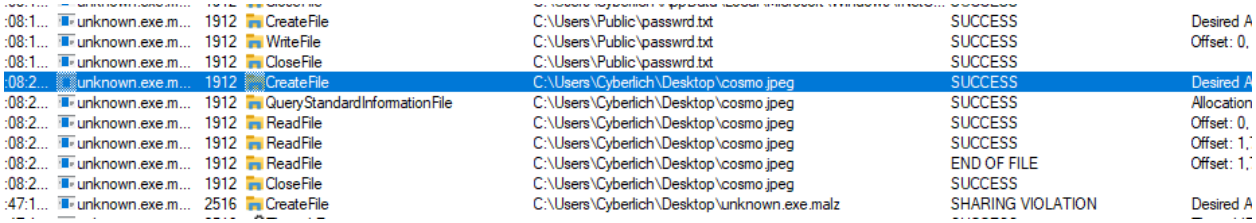
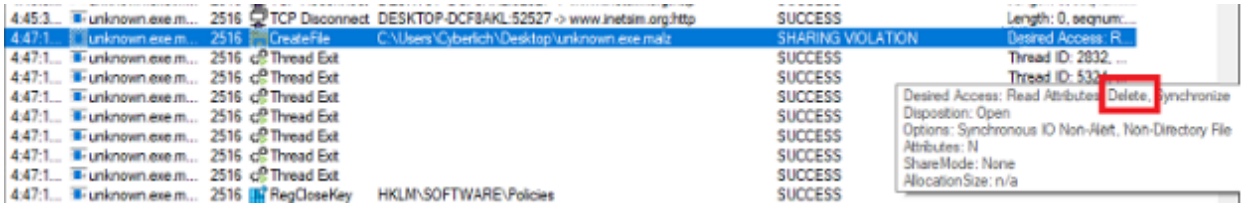


Fig 3


4:45:3... unknown.exe m... 2516 TCP Disconnect DESKTOP-DCF8AKL:52527 -> www.inetsim.org:http SUCCESS Length: 0, seqnum:...
4:47:1... unknown.exe m... 2516 CreateFile C:\Users\Cyberlich\Desktop\unknown.exe.malz SHARING VIOLATION Desired Access: R...
4:47:1... unknown.exe m... 2516 Thread Exit SUCCESS Thread ID: 2832, ...
4:47:1... unknown.exe m... 2516 Thread Exit SUCCESS Thread ID: 533...
4:47:1... unknown.exe m... 2516 Thread Exit SUCCESS
4:47:1... unknown.exe m... 2516 Thread Exit SUCCESS
4:47:1... unknown.exe m... 2516 Thread Exit SUCCESS
4:47:1... unknown.exe m... 2516 Thread Exit SUCCESS
4:47:1... unknown.exe m... 2516 Thread Exit SUCCESS
4:47:1... unknown.exe m... 2516 RegCloseKey HKLM\SOFTWARE\Policies SUCCESS
Desired Access: Read Attributes, Delete, Synchronize
Disposition: Open
Options: Synchronous IO Non-Alert, Non-Directory File
Attributes: N
ShareMode: None
AllocationSize: n/a

Rules & Signatures

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

Appendices

A. Yara Rules

```
rule SikoMode {

    meta:
        last_updated = "2023-05-05"
        author = "Dor Erlich - CybErlich"
        description = "SikoMode malware detection rule"

    strings:

        $cosmo = "cosmo.jpeg"
        $passwrd_file = "passwrd.txt"
        $RC = "toRC4"
        $dns_2 = "cdn.altimiter.local"
        $nim_client = "Nim httpclient/1.6.2"
        $houdini_func = "houdini"


    condition:
// the second condition set in case that the malware will have a variant with
different files
        ($passwrd_file and $RC and $dns_2 and $houdini_func and $cosmo and
$nim_client) or

        ($nim_client and $RC and $dns_2 and $houdini_func)


}
```

B. Callback URLs

0

Domain	Port
hxxp://update.ec12-4-109-278-3-ubuntu20-04.local	80
hxxp://cnd.altimiter.loca	80