

# Practical Malware Analysis & Triage

## Malware Analysis Report

### SikoMode Malware

July 2023| Analysis by BRDB Labs | v1.0

# Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>Executive Summary</b>	<b>3</b>
<b>High-Level Technical Summary</b>	<b>4</b>
<b>Malware Composition</b>	<b>5</b>
<b>Basic Static Analysis</b>	<b>6</b>
<b>Basic Dynamic Analysis</b>	<b>7</b>
<b>Advanced Static Analysis</b>	<b>9</b>
<b>Indicators of Compromise</b>	<b>10</b>
Network Indicators	10
Host-based Indicators	10
<b>Appendices</b>	<b>11</b>
A. Yara Rules	11
B. Callback URLs	11
D. Decompiled Code Snippets	12

# Executive Summary

SHA256 hash	3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E
-------------	--

SikoMode is a malware sample (unknown.exe) first identified during the PMAT class on July 14<sup>th</sup>, 2023. It is a Nim-compiled executable that runs on the x64 Windows operating system. Its aim is to breakdown a file on a victim's machine, encrypt and encode chunks of that file and exfiltrate each piece to an external domain where it can be captured within the logs of the domain that it is sent to and reconstructed back together.

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 consists of two parts: a call back to the below URL to determine if an internet connection is available, and then an exfiltration of data through the second URL by breaking down a file and sending chunks of the data through the post parameter of the URL.

1. It first attempts to contact its callback URL (<http://update.ec12-4-109-278-3-ubuntu20-04.local/>). If this URL can not be reached it deletes itself.
2. If this URL can be reached it will continue to execute and exfiltrate data to a second URL: ([http://cdn.altimiter.local/feed?post=\[data\]](http://cdn.altimiter.local/feed?post=[data])).

[data] is different each time to illustrate different chunks of the file. RC4 encryption and base64 encoding of the [data] is used in this case.

Once exfiltration is complete it will delete itself from disk. It will also delete itself if the exfiltration process is stopped before completion.

# Malware Composition

SikoMode consists of the following components:

File Name	SHA256 Hash
<b>unknown.exe</b>	3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E

# Basic Static Analysis

Static analysis was performed using Floss whereby strings of note were extracted from the unknown.exe sample which include:

- hxxp://cdn.altimiter.local/feed?post=
- Nim httpclient/1.6.2
- C:\Users\Public\passwrд.txt

```
@:houdini
@Authorization
@Host
@httpclient.nim(1144, 15) `false`
@Transfer-Encoding
@Content-Type
@Content-Length
@httpclient.nim(1082, 13) `not url.contains({'\r', '\n'})` url shouldn't contain any newline characters
@http://cdn.altimiter.local/feed?post=
@Nim httpclient/1.6.2
@Desktop\cosmo.jpeg
@SikoMode
@iterators.nim(240, 11) `len(a) == L` the length of the seq changed while iterating over it
@ccc
@Mozilla/5.0
@C:\Users\Public\passwrд.txt
```

*Fig 1: Snippet of key strings within the unknown.exe sample.*

This suggests the program was written in Nim, and a possible URL is used for an outbound connection.

# Basic Dynamic Analysis

When the unknown.exe sample is executed we see the initial call back address of: `hxxp://update.ec12-4-109-278-3-ubuntu20-04.local/` through DNS as captured by Wireshark below.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.5	10.0.0.4	DNS	101	Standard query 0xf23c A update.ec12-4-109-278-3-ubuntu20-04.local
2	0.002961846	10.0.0.4	10.0.0.5	DNS	117	Standard query response 0xf23c A update.ec12-4-109-278-3-ubuntu20-04.local A 10.0.0.4

Frame 1: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface enp0s3, id 0  
Ethernet II, Src: PcsCompu\_1a:b0:c3 (08:00:27:1a:b0:c3), Dst: PcsCompu\_d6:c0:17 (08:00:27:d6:c0:17)  
Internet Protocol Version 4, Src: 10.0.0.5, Dst: 10.0.0.4  
User Datagram Protocol, Src Port: 56023, Dst Port: 53  
Domain Name System (query)  
Transaction ID: 0xf23c  
Flags: 0x0100 Standard query  
Questions: 1  
Answer RRs: 0  
Authority RRs: 0  
Additional RRs: 0  
Queries  
update.ec12-4-109-278-3-ubuntu20-04.local: type A, class IN  
[Response In: 2]

Fig 2: WireShark Packet Capture of initial beacon check-in

With an http response of 200 as a connection is made.

Time ...	Process Name	PID	Operation	Path	Result	Detail
2:57:3...	unknown.exe	5680	TCP Connect	DESKTOP-CLE6HT5:1087 -> www.inetsim.org/http	SUCCESS	Length: 0, mss: 14...
2:57:3...	unknown.exe	5680	TCP Send	DESKTOP-CLE6HT5:1087 -> www.inetsim.org/http	SUCCESS	Length: 92, startm...
2:57:3...	unknown.exe	5680	TCP Receive	DESKTOP-CLE6HT5:1087 -> www.inetsim.org/http	SUCCESS	Length: 150, seqn...
2:57:3...	unknown.exe	5680	TCP Receive	DESKTOP-CLE6HT5:1087 -> www.inetsim.org/http	SUCCESS	Length: 258, seqn...
2:57:3...	unknown.exe	5680	TCP Disconnect	DESKTOP-CLE6HT5:1087 -> www.inetsim.org/http	SUCCESS	Length: 0, seqnum:...

Fig 3: TCPView capture of outbound http connection by unknown.exe

Further analysis through Procmon using the “CreateFile” filter reveals that the file **C:\Users\Public\passwd.txt** is created on disk, and the malware checks for the **C:\Users\brdb\Desktop\cosmos.jpg** file.

3:28:1...	unknown.exe	3032	CreateFile	C:\Users\Public\passwd.txt	SUCCESS	Desired Access: G...
3:28:1...	unknown.exe	3032	CreateFile	C:\Users\Public\passwd.txt	SUCCESS	Desired Access: G...
3:28:1...	unknown.exe	3032	CreateFile	C:\Users\brdb\Desktop\cosmo.jpeg	SUCCESS	Desired Access: G...
3:28:1...	unknown.exe	3032	CreateFile	C:\Users\Public\passwd.txt	SUCCESS	Desired Access: G...

Fig 4: ProcMon filter view of host indicators

The data within the file cosmos.jpg is encrypted using the password within the password.txt file. It is then encoded, and transmitted out in chunks within the exfiltration URL mentioned above.

If the callback URL is reachable, then the cosmos file is exfiltrated to the: `hxxp://cdn.altimeter.local/feed?post=[data]` address. A wireshark capture illustrates the connection to this address and the different [data] parameters sent.

Source	Destination	Protocol	Length	Info
10.0.0.4	10.0.0.5	TCP	56	443 → 1288 [ACK] Seq=1 Ack=198 Win=64128 Len=0
10.0.0.5	10.0.0.4	TCP	62	1288 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
10.0.0.4	10.0.0.5	TCP	68	443 → 1288 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
10.0.0.5	10.0.0.4	TCP	68	1288 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.0.0.4	10.0.0.5	HTTP	314	HTTP/1.1 200 OK (text/html)
10.0.0.5	10.0.0.4	HTTP	293	GET /feed?post=B2ED11008592799244B03F50A8C3342C3D2BC1F29C52C939D4E81F66E2489AB6BC6A7B31998CEC93A220A6466D404C49A988BD68958ECBF1D6676CCFAFA9 HTTP/1.1
10.0.0.4	10.0.0.5	HTTP	314	HTTP/1.1 200 OK (text/html)
10.0.0.5	10.0.0.4	HTTP	293	GET /feed?post=B69C0CF68536758144B03372D0DD38291DEBB31925F523A386678EEC5414AF8966D1BCA316ADC68C30020A6466D404C49A988BD6895AC5BF174376CCB8BC HTTP/1.1
10.0.0.4	10.0.0.5	HTTP	314	HTTP/1.1 200 OK (text/html)
10.0.0.5	10.0.0.4	HTTP	293	GET /feed?post=A69C1CF68536758244B2337BAFFE38296DEBB01A07FF209190758DD0480786BE49FDA8851998C6BC34020A6C57E504C48A988BD68959C6B7174302E29084 HTTP/1.1
10.0.0.4	10.0.0.5	HTTP	314	HTTP/1.1 200 OK (text/html)
10.0.0.5	10.0.0.4	HTTP	293	GET /feed?post=B69C1CF58536758272963755A8FB34291DEBB01907FC2891907789E440128EBE45FDA88C1998C6BC08240E5C72D40CC49A988BD6895AC6B7666571CEBBA9 HTTP/1.1
10.0.0.4	10.0.0.5	HTTP	314	HTTP/1.1 200 OK (text/html)
10.0.0.5	10.0.0.4	HTTP	293	GET /feed?post=B69A1CF6853645A44A0337BA9FB38291DEBB01A07FC129199658DD04C1286BE45FEA8851D98C6BC34220A6466D404C49A988BD6895AF291136076CCFAFA9 HTTP/1.1
10.0.0.4	10.0.0.5	HTTP	314	HTTP/1.1 200 OK (text/html)
10.0.0.5	10.0.0.4	HTTP	293	GET /feed?post=A8E437E8F0367592569A2870BBD0382A1DFBB01A15FC239990778BC33502AD9256E481B402BDC6BC25167B6478F204C49A98ADD68C4AC2A617437ECCBBA9 HTTP/1.1
10.0.0.4	10.0.0.5	HTTP	314	HTTP/1.1 200 OK (text/html)
10.0.0.5	10.0.0.4	HTTP	148	GET / HTTP/1.1
10.0.0.4	10.0.0.5	HTTP	314	HTTP/1.1 200 OK (text/html)
10.0.0.5	10.0.0.4	HTTP	258	GET /msdownload/update/v3/static/trusted/en/disallowedcertst1.cab?d170b06d999753ac HTTP/1.1

Fig 5: Wireshark capture of file exfiltration through multiple GET requests.

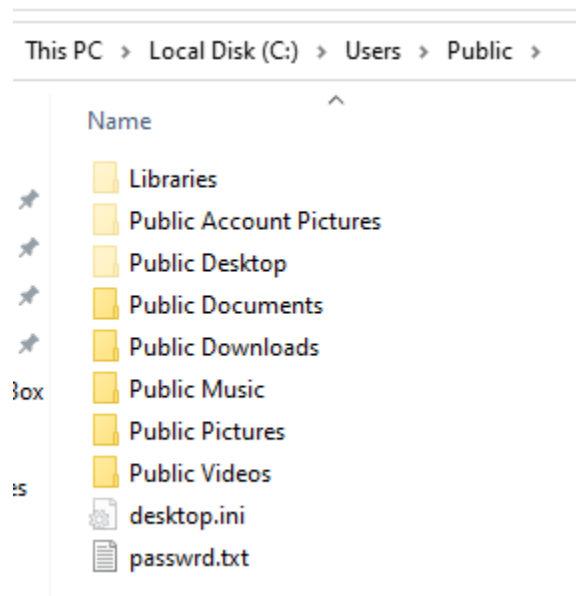


# Advanced Static Analysis

String analysis revealed reference to the “toRC4” string within the sample.

Inside of the binary we can search for the “toRC4” method call which is invoked from the **sym.stealStuff** method. This method will then encrypt the information that it brings into it.

The password within the passwrд.txt file (C:\Users\Public\passwrд.txt) created by unknown.exe is the password (key) used for the RC4 encryption routine as shown in the C:\Users\Public folder.



*Fig 6: Presence of password.txt file after unknown.exe is run*

Within a static string analysis, a reference to the string “houdini” is mentioned.

Houdini refers to the method call that deletes the binary from disk (after the Kill Switch URL has been checked and fails). This is the mechanism where if an internet connection is available the payload will execute. However, if the callback domain cannot be reached (i.e. there is no internet connection or the connection is interrupted) then Houdini will action the deletion of the binary off disk.

# Indicators of Compromise

## Network Indicators

- `hxxp://update.ec12-4-109-278-3-ubuntu20-04.local`
- `hxxp://cdn.altimiter.local/feed?post=[data]`

## Host-based Indicators

- `C:\Users\Public\passwd.txt`
- `C:\Users\brdb\Desktop\cosmos.jpg`

# Appendices

## A. Yara Rules

```
rule Nim_Malware_Unknown {
  meta:
    description = "Yara rule for detecting the unknown.exe Nim malware"
    author = "BRDB"
    date = "2023-07-17"
  strings:
    $string1 = "http://cdn.altimiter.local"
    $string2 = "cosmo.jpeg"
    $string3 = "C:\\Users\\Public\\passwrд.txt"
  condition:
    all of them
}
```

*Fig 8: Initial Yara rule for detection*

## B. Callback URLs

Domain	Port
<b>hxxp://update.ec12-4-109-278-3-ubu</b>	53

## C. Exfiltration URL

Domain	Port
<b>hxxp://cdn.altimiter.local/feed?post=[data]</b>	80

## D. Decompiled Code Snippets

```
[0x0041761d]
0x0041761d    mov     rcx, r12     ; int64_t arg2
0x00417620    call    raiseIndexError2 ; sym.raiseIndexError2
0x00417625    jmp     0x417547

[0x00417547]
0x00417547    mov     rax, qword [var_2f8h]
0x0041754e    mov     rcx, rbx     ; int64_t arg1
0x00417551    mov     rdx, qword [rax + r12*8 + 0x10] ; int64_t arg2
0x00417556    call    toRC4_00Z00Z00Z00Z00Z0nimbleZpkgsZ8267524548049048Z826752_51 ; sym.toRC4...
0x0041755b    mov     rdx, qword data.0041e9f0 ; 0x41e9f0 ; int64_t arg2
0x00417562    mov     rcx, qword [var_300h] ; int64_t arg1
0x00417569    mov     r14, rax
0x0041756c    call    incrSeqV3     ; sym.incrSeqV3
0x00417571    mov     rcx, r14     ; int64_t arg1
0x00417574    mov     qword [var_300h], rax
0x0041757b    mov     rax, qword [rax]
0x0041757e    mov     rdi, qword [var_300h]
0x00417585    lea     rdx, [rax + 1]
0x00417589    mov     qword [rdi], rdx
0x0041758c    lea     rdi, [rdi + rax*8]
0x00417590    mov     r15, qword [rdi + 0x10]
0x00417594    call    copyStringRC1 ; sym.copyStringRC1
0x00417599    mov     qword [rdi + 0x10], rax
0x0041759d    test    r15, r15
0x004175a0    jne     0x41762a
```

Fig 9: stealStuff method invokes the toRC4 method to encrypt the data.