Practical Malware Analysis & Triage

Malware Analysis Report

SikoMode Exfiltration Malware

January 2024 | Analysis by 2shy | v1.0

PRACTICAL MALWARE ANALYSIS & TRIAGE

Table of Contents

PRACTICAL MALWARE ANALYSIS & TRIAGE

Executive Summary

SHA256 hash	3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E

SikoMode (unknown.exe) is an exfiltration-type malware sample first identified on January 30th, 2024. It is a Nim-compiled x64 binary that runs on the Windows operating system. This malware attempts to communicate with two callback URLs - see Appendix B. Symptoms of infection include callouts to the URLs listed in Appendix B, and the creation of a file in the C:/Users/Public/ directory named "passwrd.txt". SikoMode's purpose is to locate and exfiltrate a file by the name of cosmo.jpeg.

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

PRACTICAL MALWARE ANALYSIS & TRIAGE

High-Level Technical Summary

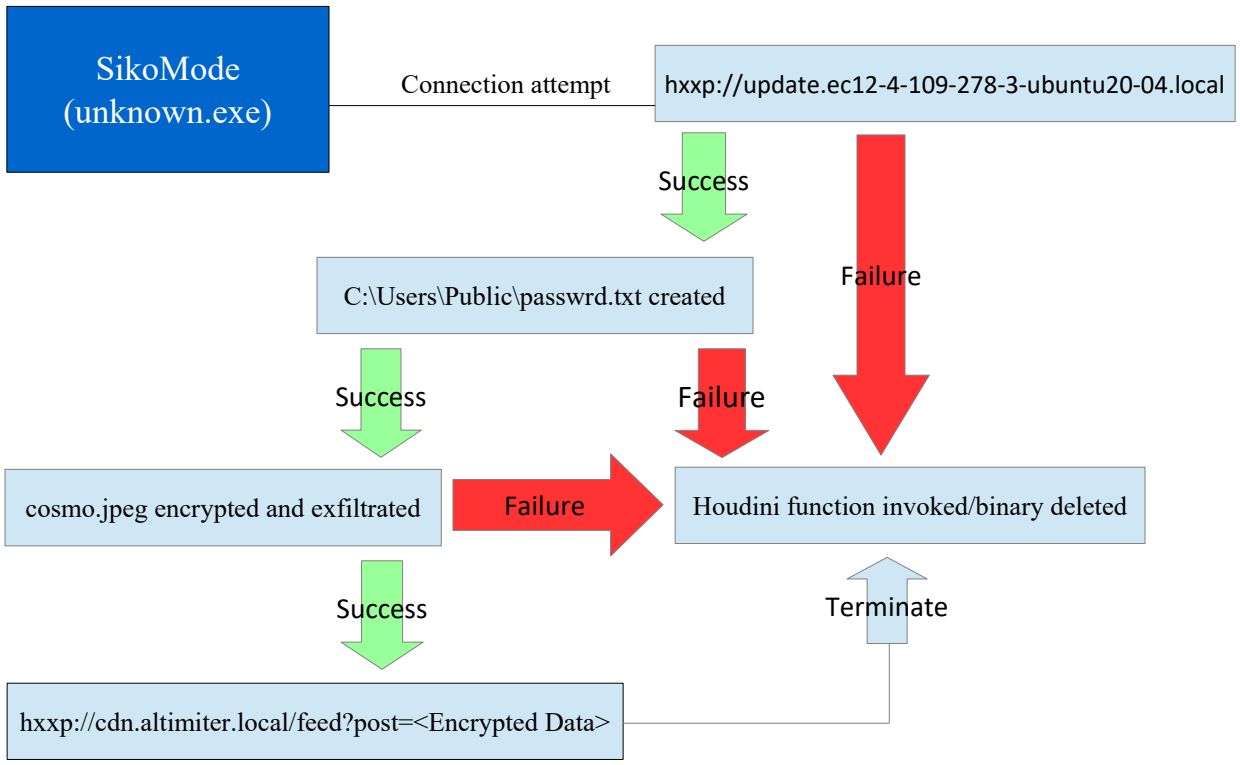
When detonated, SikoMode first sends out a beacon to the initial callback URL, “hxxp://update.ec12-4-109-278-3-ubuntu20-04.local/” If this attempt is successful, it creates a file called “passwrd.txt” in the C:/Users/Public/ directory, and will subsequently attempt to locate a file named cosmo.jpeg. SikoMode then exfiltrates the contents of cosmo.jpeg back to its second callback URL via a POST parameter, “hxxp://cdn.altimiter.local/feed?post=<Encrypted Data>”. The contents of the POST parameter are obfuscated via RC4 encryption, and its value changes with each request.

If SikoMode's attempt to reach its first callback domain fails, there is a kill switch function in place that invokes a function called Houdini. Houdini's purpose is simply to delete the binary from disk. Houdini will also be invoked if its callback attempts are interrupted, or it can not locate the cosmo.jpeg file.

The malware does not spawn any child processes, and does not employ any persistence mechanisms.

PRACTICAL MALWARE ANALYSIS & TRIAGE

SikoMode Execution Flow Chart:



Malware Composition

SikoMode consists of the following components:

File Name	SHA256 Hash
unknown.exe	3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E
passwrd.txt	1eebfcf7b68b2b4ffe17696800740e199acf207afb5514bc51298c2fe7584410

unknown.exe

This is the main executable file.

passwrd.txt

This is a file that is created after successfully detonating the malware. It is created in the C:/Users/Public/ directory and contains the RC4 encryption key that is used to encrypt the exfiltrated cosmo.jpeg contents.

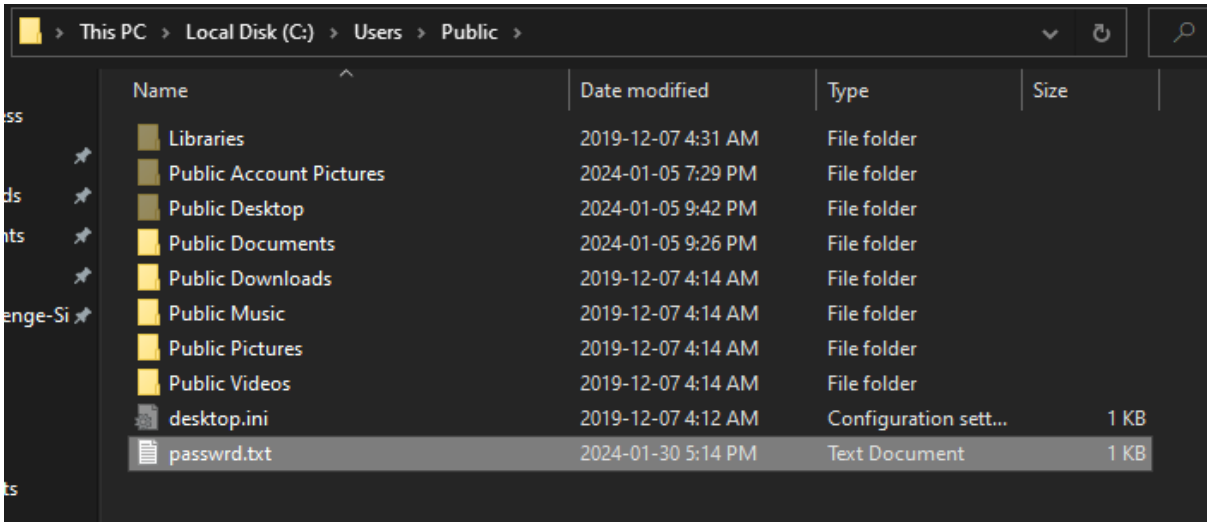


Fig 1: The passwrd.txt file has been created in the C:/Users/Public/ directory.

Fig 2: Contents of passwrd.txt containing the encryption key.

Static Analysis

Basic static analysis using PEStudio determined that this is an unpacked 64-bit Windows binary.

file-type	executable
cpu	64-bit

Fig 1: Snippet taken from PEStudio reveals a 64-bit executable.

When we compare the binary's raw size vs its virtual size, we can see that the bytes are similar in size indicating that this is an unpacked binary.

raw-address (end)	0x00019000
raw-size (451584 bytes)	0x00018A00 (100864 bytes)
virtual-address	0x00001000
virtual-size (536320 bytes)	0x00018818 (100376 bytes)

Fig 2: Snippet taken from PEStudio comparing raw and virtual sizes.

PEStudio flagged several API calls as potentially malicious. These API calls perform reconnaissance/process injection/execution functions, and can potentially be associated with malware.

GetCurrentProcessId	x	0x000000000003A5C4	0x000000000003A5C4	553 (0x0229)	reconnaissance	T1057 | Process Discovery
VirtualAlloc	x	0x000000000003A768	0x000000000003A768	1486 (0x05CE)	memory	T1055 | Process Injection
VirtualProtect	x	0x000000000003A786	0x000000000003A786	1492 (0x05D4)	memory	T1055 | Process Injection
GetCurrentProcess	x	0x000000000003A5B0	0x000000000003A5B0	552 (0x0228)	execution	T1057 | Process Discovery
GetCurrentThreadId	x	0x000000000003A5DA	0x000000000003A5DA	557 (0x022D)	execution	T1057 | Process Discovery
RtlAddFunctionTable	x	0x000000000003A6AC	0x000000000003A6AC	1222 (0x04C6)	execution	-
RtlLookupFunctionEntry	x	0x000000000003A6D6	0x000000000003A6D6	1230 (0x04CE)	execution	-
TerminateProcess	x	0x000000000003A72A	0x000000000003A72A	1425 (0x0591)	execution	-

Fig 3: Snippet taken from PEStudio displaying potentially malicous API calls.

Running the sample through FLOSS revealed several strings of interest. These extracted strings suggest that this binary was written in Nim, and that there may be a potential callback URL present and a potential encryption scheme in place. Two potentially interesting files were also referenced, passwrd.txt and cosmo.jpeg Also of note was the string “stealStuff__sikomode_130”.

Fig 4: FLOSS output showing various strings of interest.

When examining the malware with Cutter, we can see the existence of a “checkKillSwitchURL__sikomode_25” function which likely relates to the initial DNS beacon.

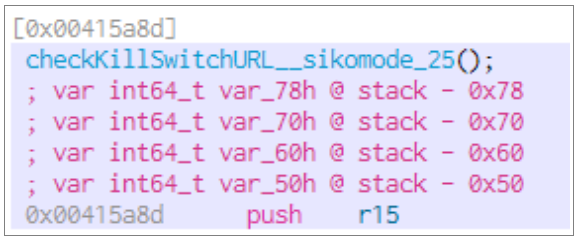


Fig 5: Snippet from Cutter showing the presence of a kill switch function.

We can also see that there is a “houdini__sikomode_51” function that the strings pulled from Floss eluded to.

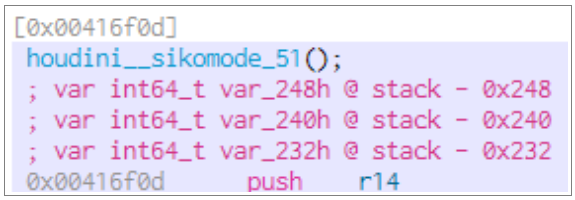


Fig 6: Snippet from Cutter showing the presence of a houdini function.

There is also a “toRC4__OOZOOZOOZOOZOOZOnimbleZpkgsZ8267524548O49O48Z826752_51” function present which is likely SikoMode's method of encryption.

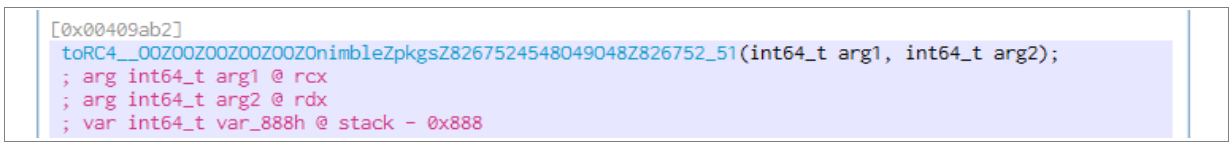


Fig 7: Snippet from Cutter showing the presence of the probable encryption mechanism.

Cutter also shows the presence of another interesting function, “stealStuff__sikomode_130”.

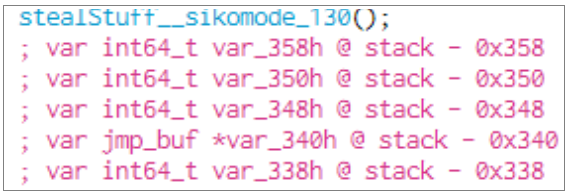


Fig 8: Snippet from Cutter showing an interesting fnction.

Dynamic Analysis

Viewing unknown.exe's activities in Procmon64 shows that the file passwrd.txt is created in the C:\Users\Public\ directory, and that the program has successfully located the cosmo.jpeg file.

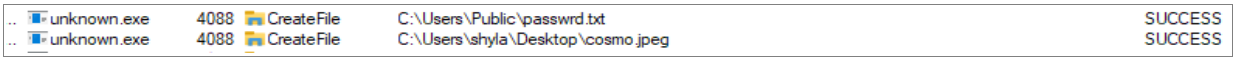


Fig 1: Procmon64 output showing the creation of passwrd.txt.

Deploying Wireshark to analyze the network traffic generated by SikoMode shows that the malware is attempting to contact two separate callback URLs in succession. The program first sends a beacon out to its first callback URL, hxxp://update.ec12-4-109-278-3-ubuntu20-04.local/. If contact is successful, the second callback URL is contacted, and the encrypted cosmo.jpeg contents are sent via a POST parameter to hxxp://cdn.altimiter.local/feed?post=<Encrypted Data>. The POST parameter contents are encrypted and display a different output each time it is sent.

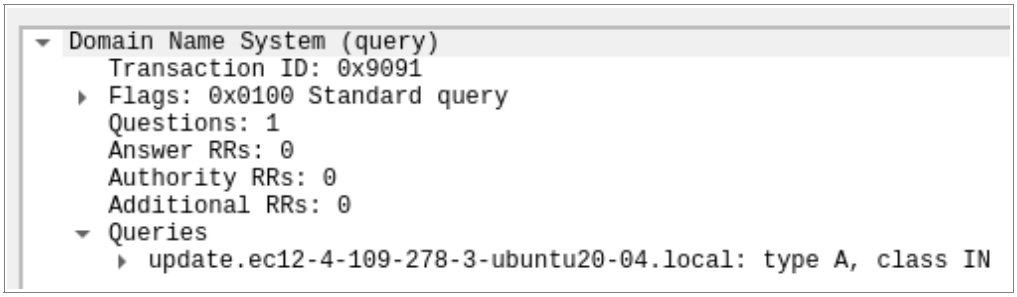


Fig 2: Wireshark Packet Capture of a DNS request to the initial callback URL.

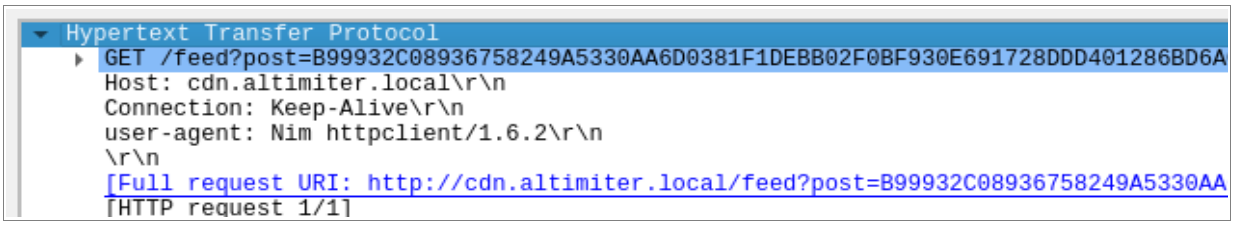


Fig 3: Wireshark Packet Capture of the exfiltrated file being sent to the second callback URL.

Indicators of Compromise

Network Indicators

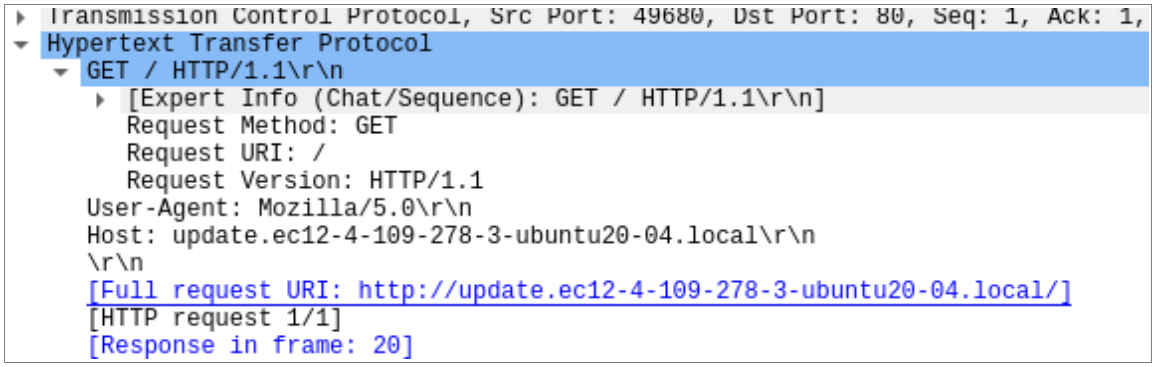


Fig 1: Wireshark Packet Capture of initial beacon check-in.

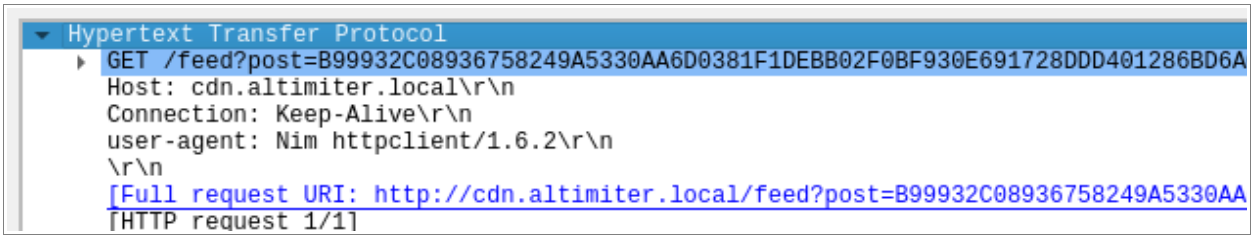


Fig 2: Wireshark Packet Capture of the exfiltration URL.

Host-based Indicators

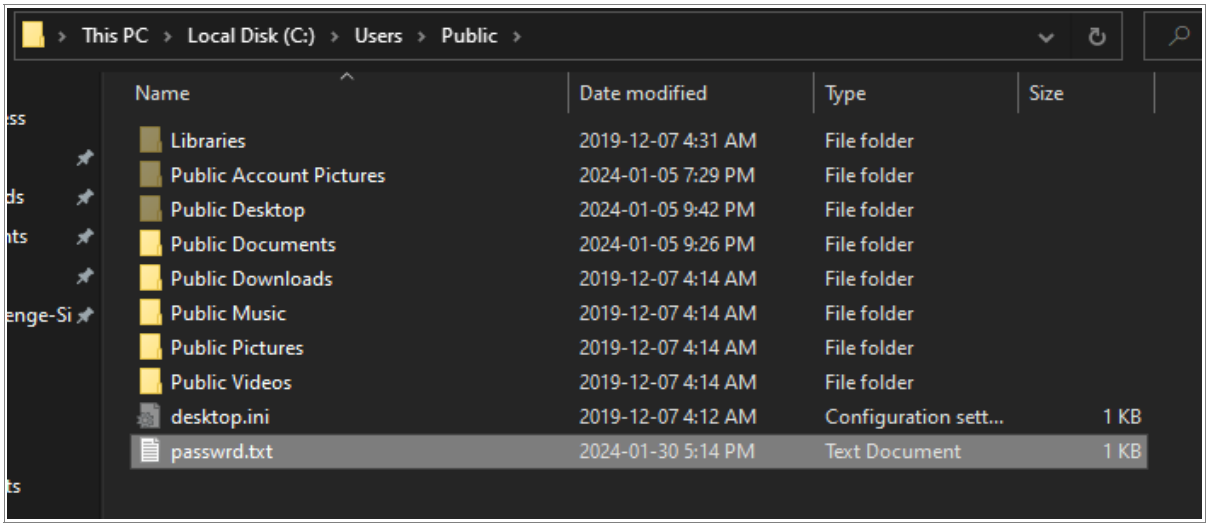


Fig 3: Presence of passwrd.txt in the C:/Users/Public directory.

Appendices

A. Yara Rules

```
rule SikoMode {

    meta:
        last_updated = "2021-10-15"
        author = "PMAT"
        description = "SikoMode YARA Rules"

    strings:
        $String1 = "http://cdn.altimiter.local/feed?post=" ascii
        $String2 = "nim"
        $String3 = "C:\Users\Public\passwrd.txt" ascii
        $String4 = "Desktop\cosmo.jpeg" ascii
        $PE_magic_byte = "MZ"

    condition:
        $PE_magic_byte at 0 and
        ($String1 and $String2 and $String3 and $String4)
}
```

B. Callback URLs

Domain	Port
hxxp://update.ec12-4-109-278-3-ubuntu20-04.local/	80
hxxp://cdn.altimiter.local/feed?post=<Encrypted Data>	80

C. Decompiled Code Snippets

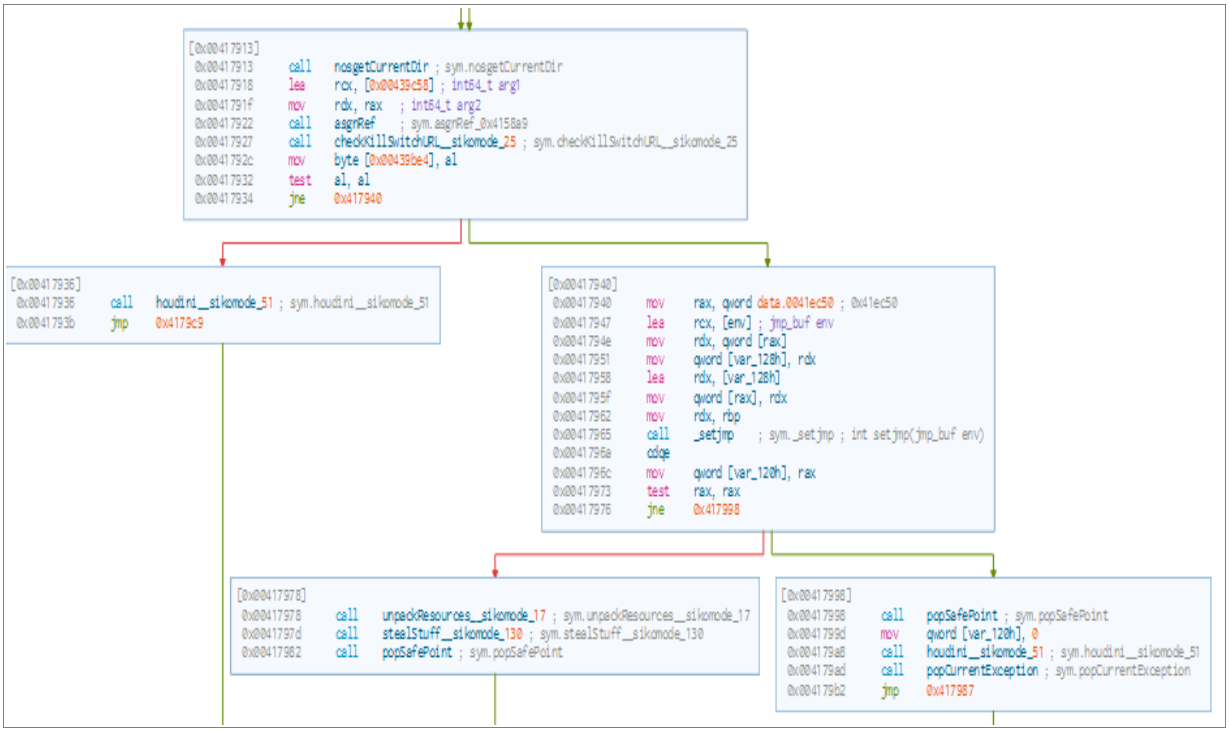


Fig 1: Snippet from Cutter showing a flow graph containing many key functions.