Practical Malware Analysis & Triage

Malware Analysis Report

WannaHusky

Oct 2022 | Vien Amor V | v1.0

Table of Contents

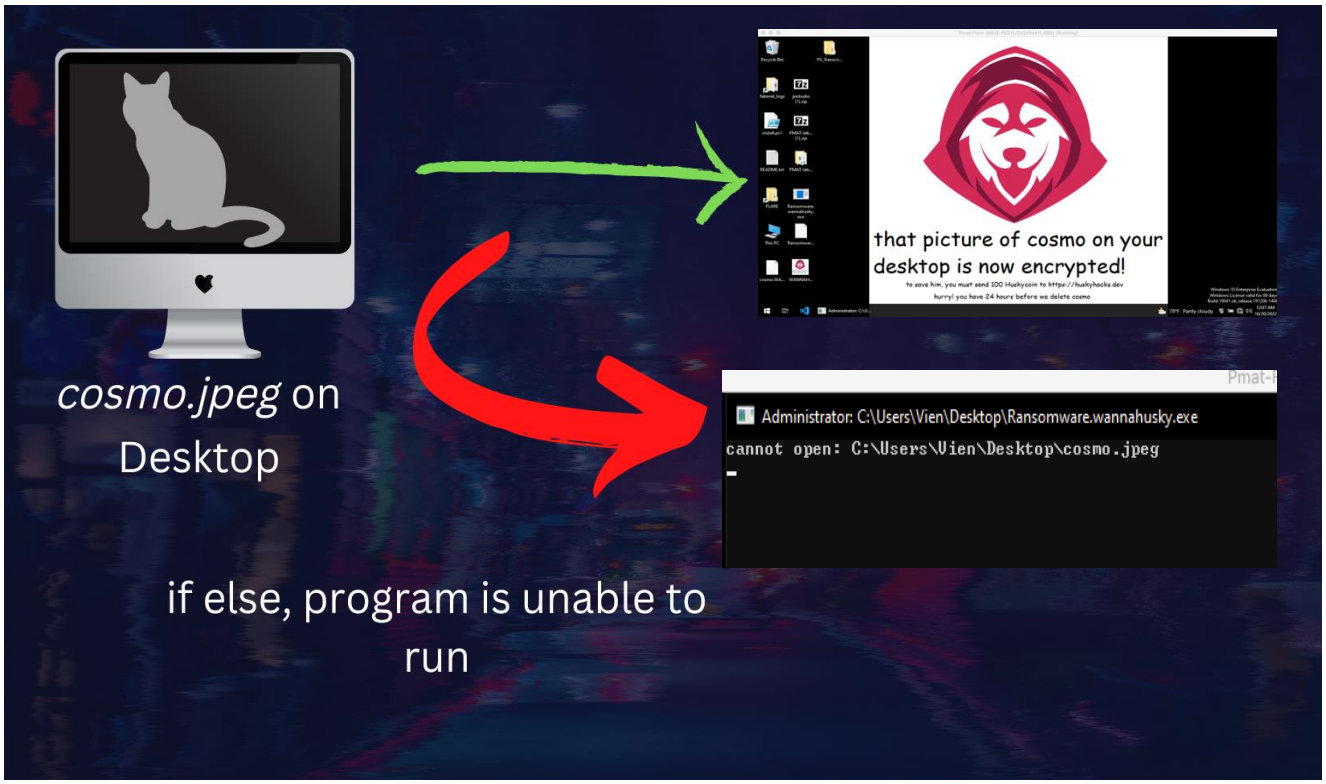
Executive Summary

Filename	Ransomware[.]wannahusky[.]exe
SHA256 hash	3d35cebcf40705c23124fdc4656a7f400a316b8e96f1f9e0c187e82a9d17dca3

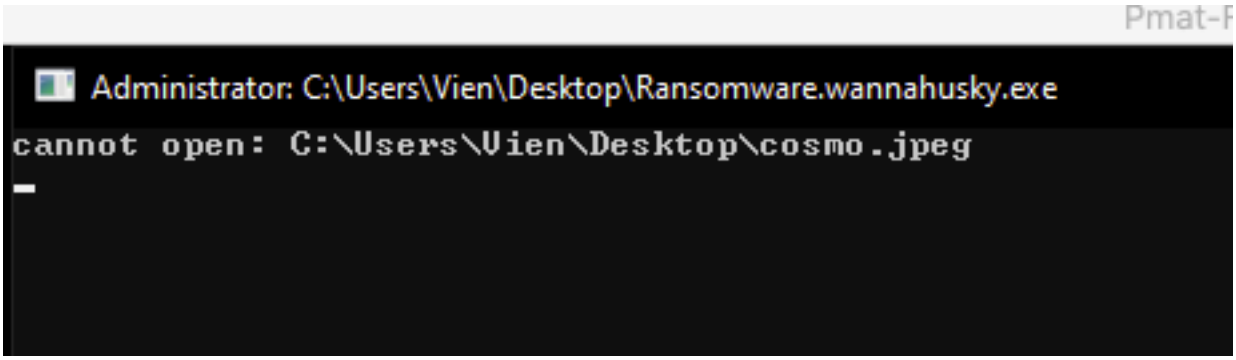
WannaHusky is a ransomware that requires encrypts files, specifically *cosmo.jpeg,* and demands Huskycoin as payment for files to be decrypted. It is a binary compiled using Nim and defaces the users' desktop. For this program to be executed, *cosmos.jpeg,* must be enabled on the users' desktop; if not, a cmd prompt will execute and program will not run intended. When executed correctly, user experiences a large HuskyHacks Logo in the background under the filename *WANNAHUSKY.png*. File *cosmo.jpeg* now becomes *cosmo.WANNAHUSKY* (encrypted). Finally, executable file *ps1.ps1* is located on the desktop which runs a command prompt executing a process tree.

High-Level Technical Summary

For WannaHusky to be executed, a filename *cosmos.jpeg* must be on the user's desktop. If cosmos is not located on desktop, we get an error and ransomware does not run as intended.



(Figure 1.1 – Enlarged Successful Ransomware Execution)



(Figure 1.2 – Enlarged Unsuccessful Ransomware Execution)

Malware Composition

WannaHusky consists of the following components:

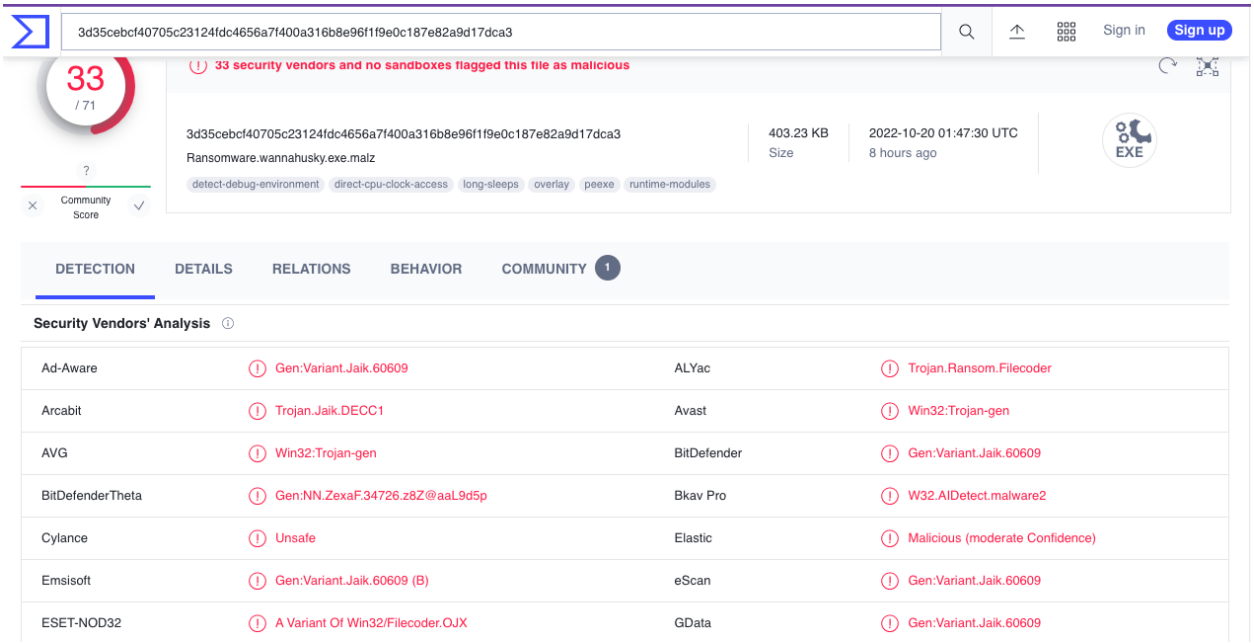
Hash Values:

File Name	SHA256 Hash
Ransomware[.]Wannahusky[.]exe	3d35cebcf40705c23124fdc4656a7f400a316b8e96f1f9e0c187e82a9d17dca3
Ps1.ps1	D6317374F879CD4E67FBE9DDC0D283926489F4C0D6CF07D912A247E5CFDE99

Additional resources

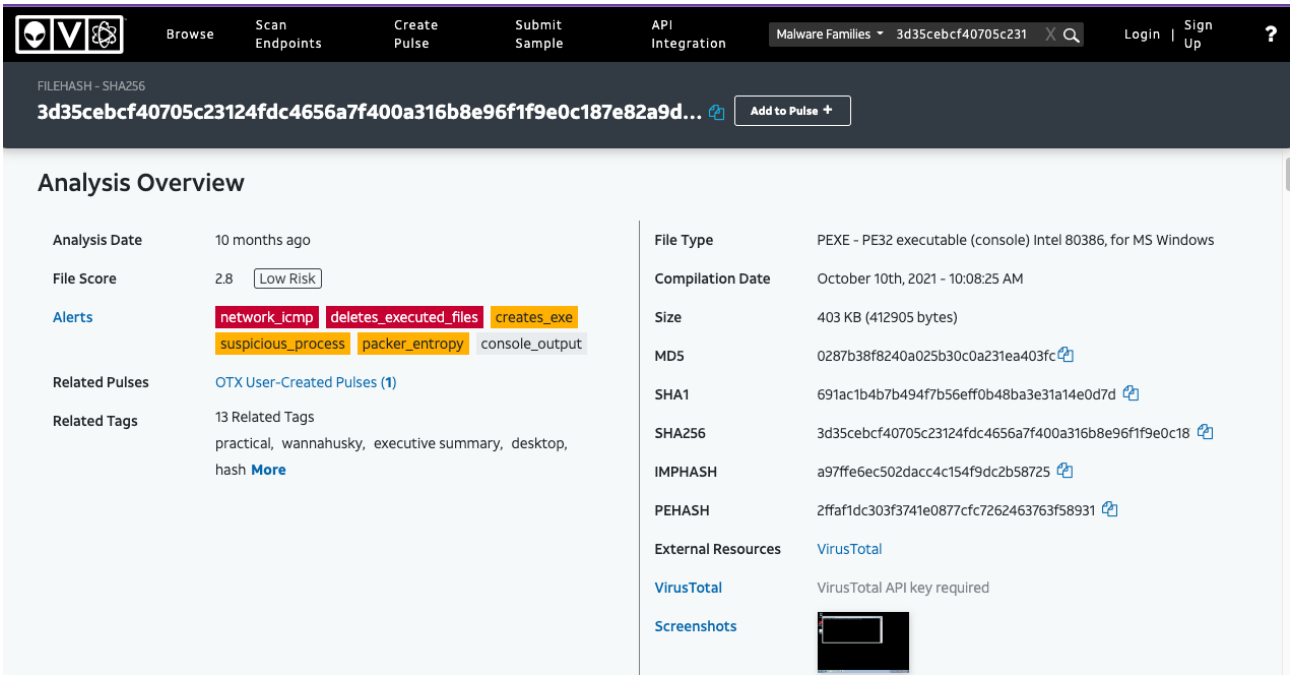
OSINT Tools	Description
VirusTotal	**33 / 71 security vendors and no sandboxes flagged this file as malicious**
AlienVault	**File Score 2.8 Low Risk**

VirusTotal

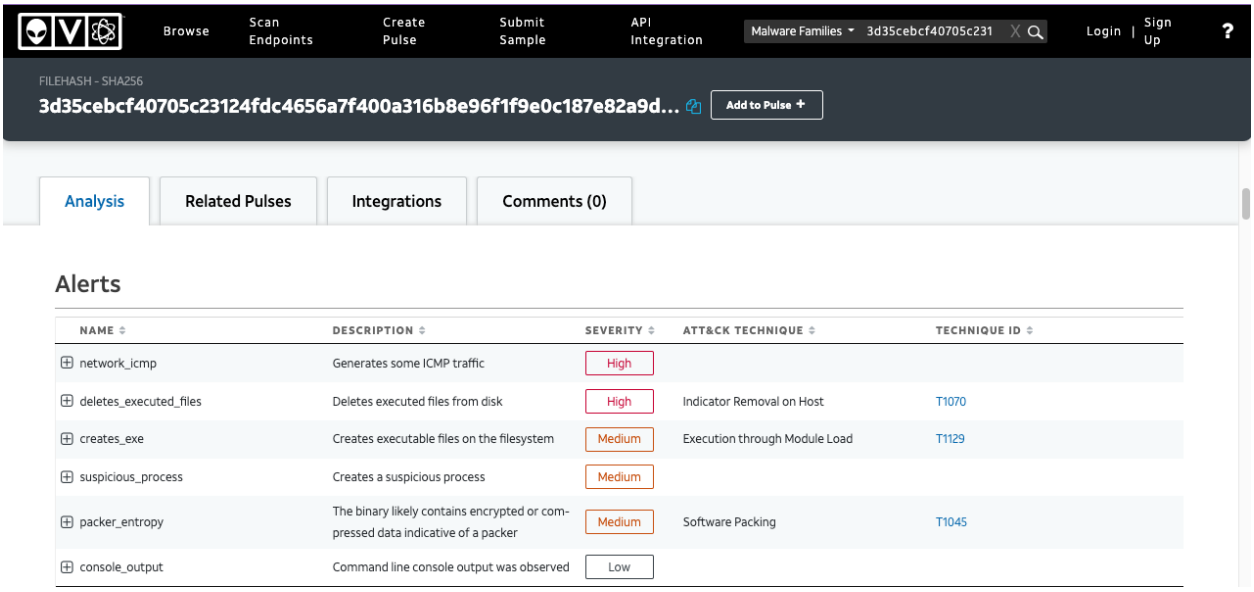


(Figure 1.3 – VirusTotal Analysis Report)

AlienVault



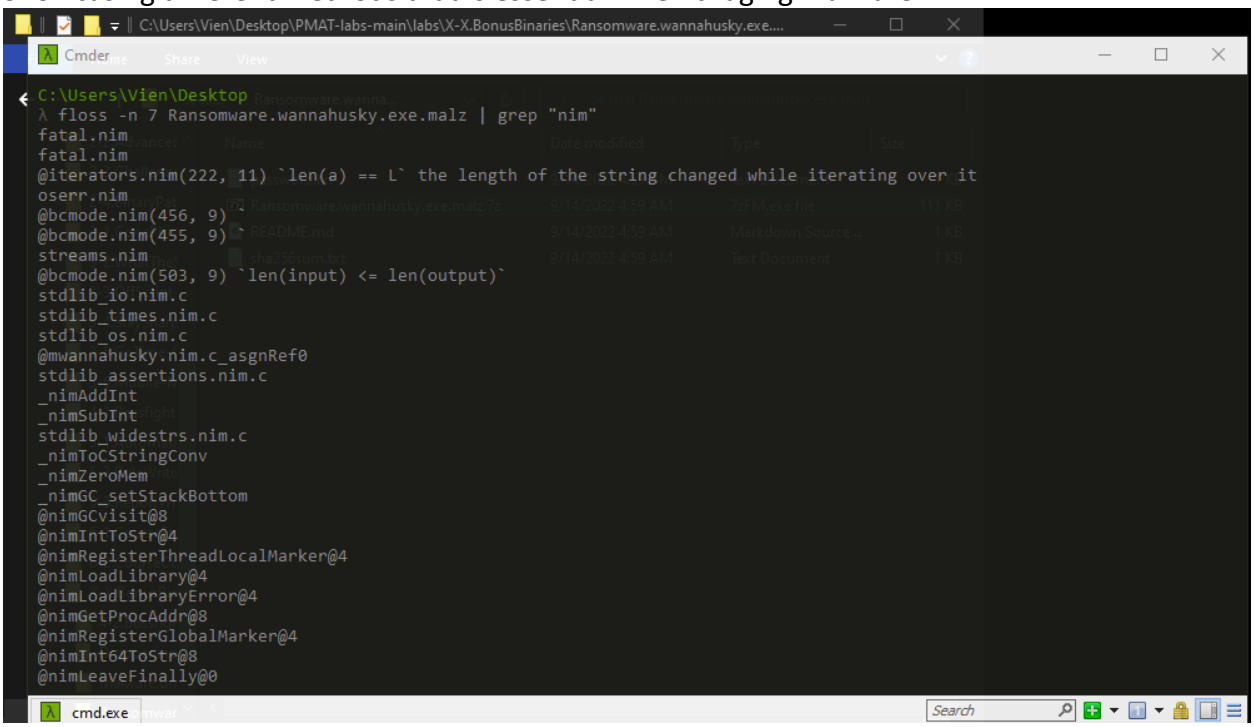
(Figure 1.4 – AlienVault Analysis)



(Figure 1.5 – AlienVault Alert Analysis. Additional indicator that IoC removes itself from host if not executed correctly.)

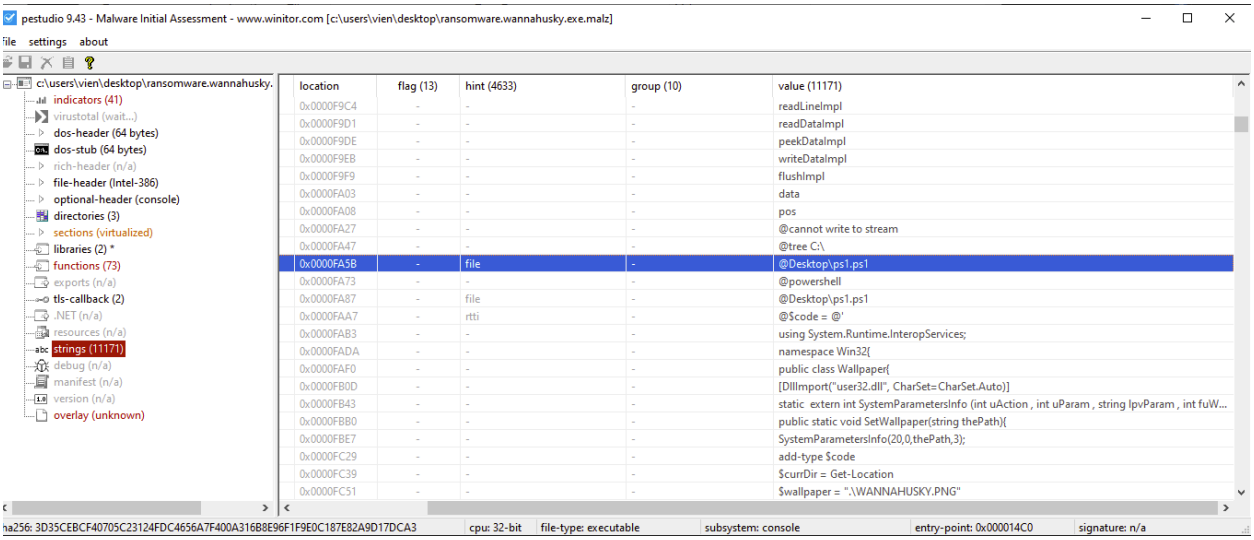
Basic Static Analysis

Showcasing different methods that is essential when triaging malware



(Figure 1.6 – Using Floss to initially see the strings of the binary. Here we see few examples showing the ransomware sample being built by the Nim binary.)

(Figure 1.7 – Another juicy string found in the CLI. Here we see class *Desktop\WANNAHUSKY.png* source-code being built onto the desktop.)



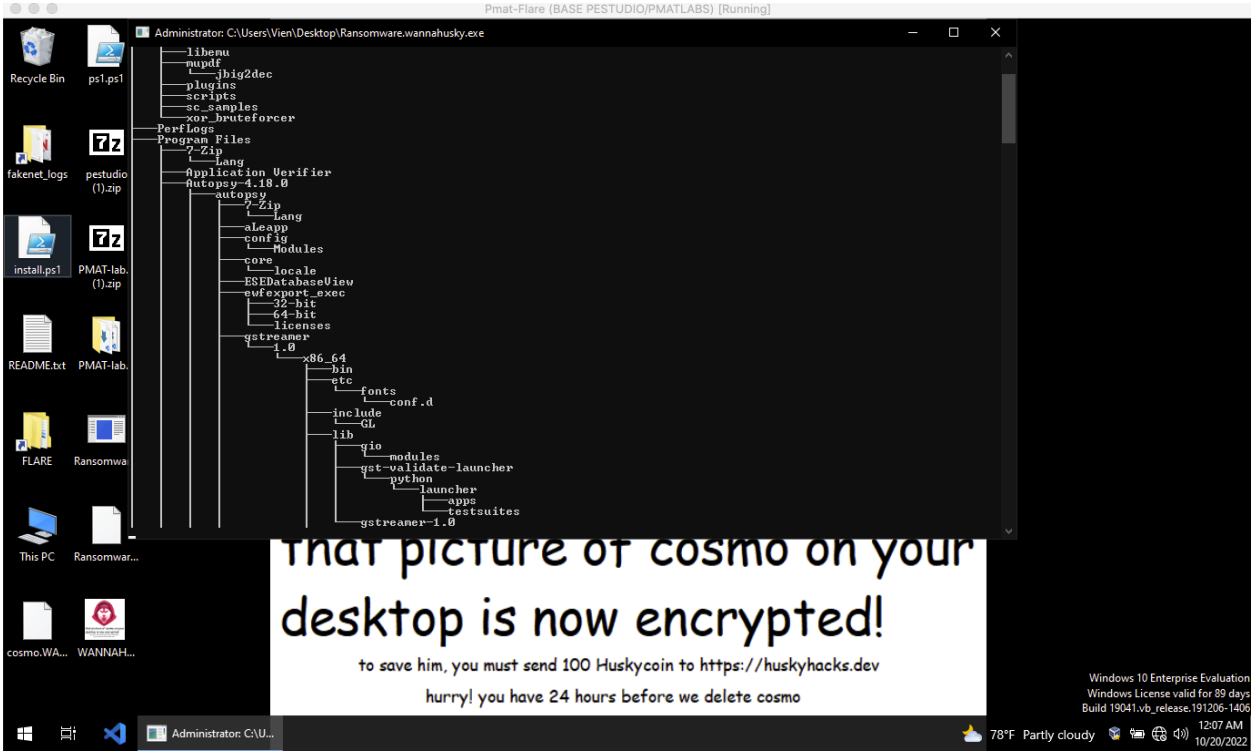
(Figure 1.8 – Additional confirmation of strings within ransomeware[.]wannahusky[.]exe using pestudio)

Basic Dynamic Analysis

Initial Detonation

When executable file has been successfully "Run as Administrator", host computer has now been altered and four new changes has been committed to the users' desktop. Changes include wallpaper change, encryption of *cosmos.jpeg -> cosmos.WANNAHUSKY,* and two new file additions – *WANNAHUSKY.png and ps1.ps1* – have been added to the desktop. (See figure 1.1)

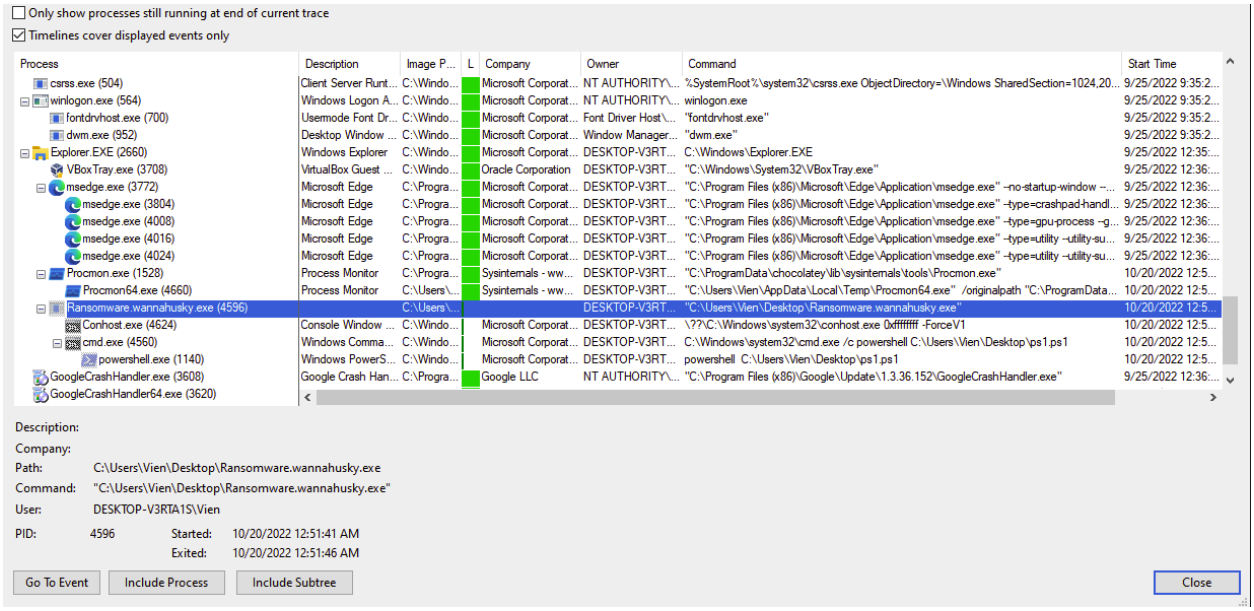
Additionally, a command prompt executes onto our computer and we see a stream of processes happening in the command line.



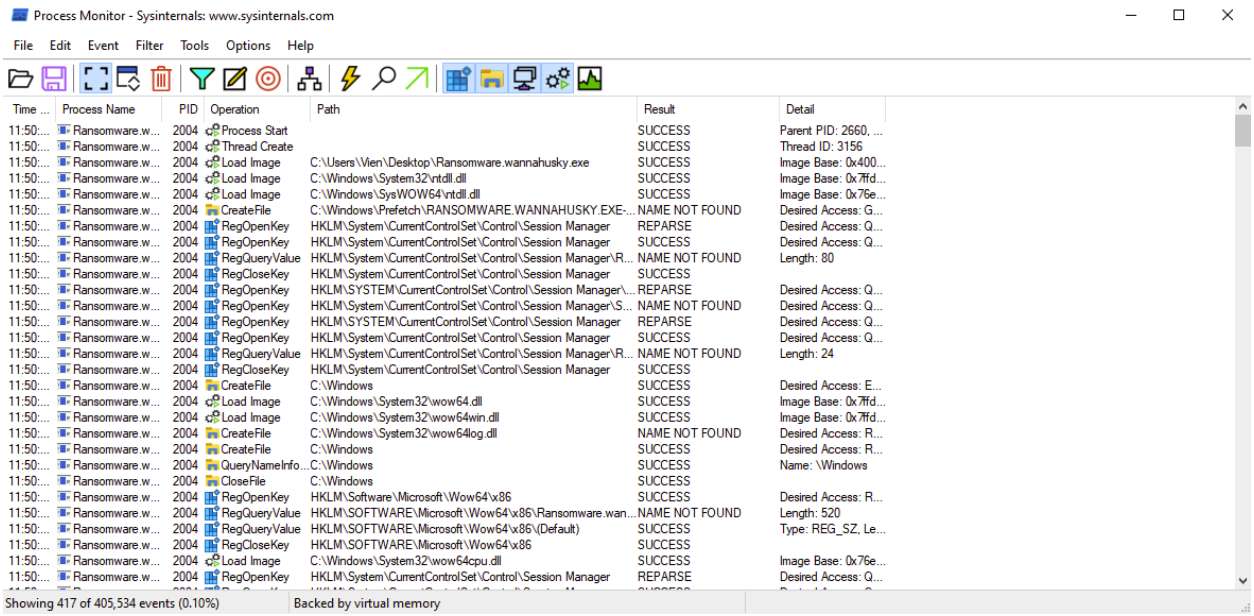
(Figure 1.9 – Commands being executed when binary runs successfully.)

Procmon

Upon detonation we utilize Procmon to see the process of WannaHusky and the Files created within the computer.



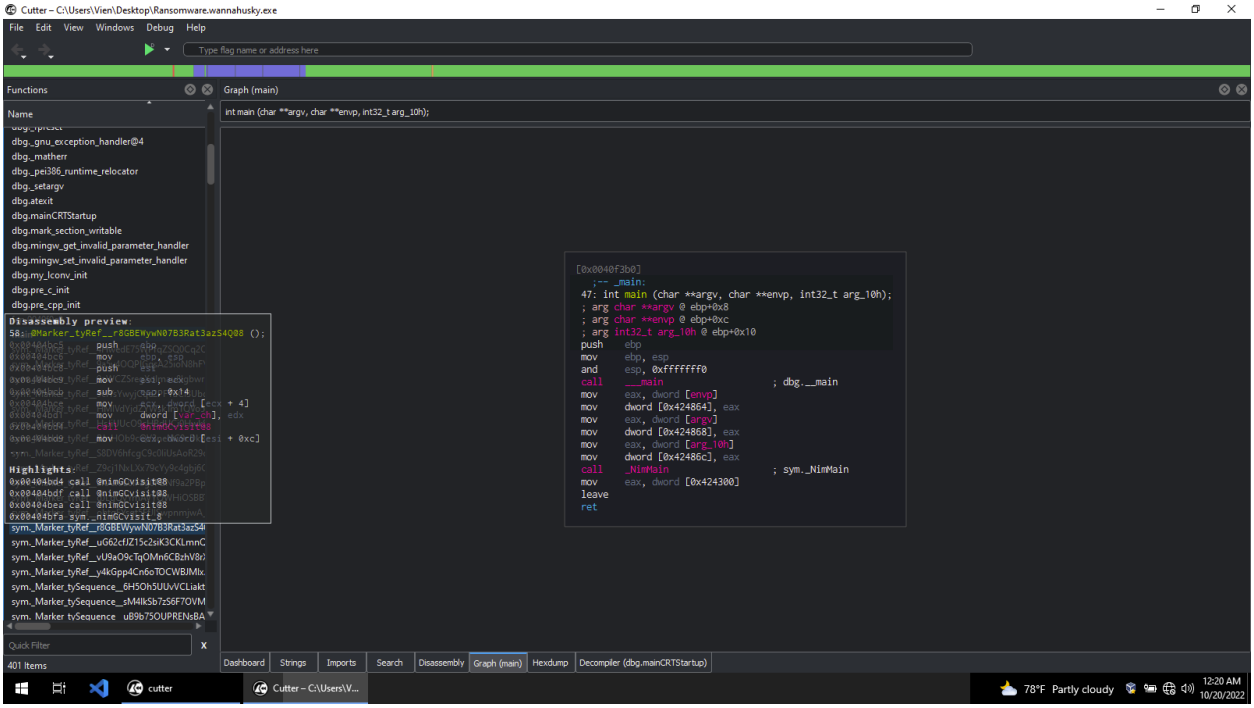
(Figure 2.0 – Utilizing Procmon to see the parent tree of Ransomware WannaHusky.)



(Figure 2.1 – Deeper Procmon Investigation. Here we filter out Parent Name understand the process of how WannaHusky is executed within the hosts' binary.)

Advanced Static Analysis

We dig deeper utilizing Cutter to understand the innards of this malware to fully grasp what is going on.



(Figure 2.2 – Finding the main call function in the malware sample. Note: "main" is the primary function where the executable file begins its execution.)

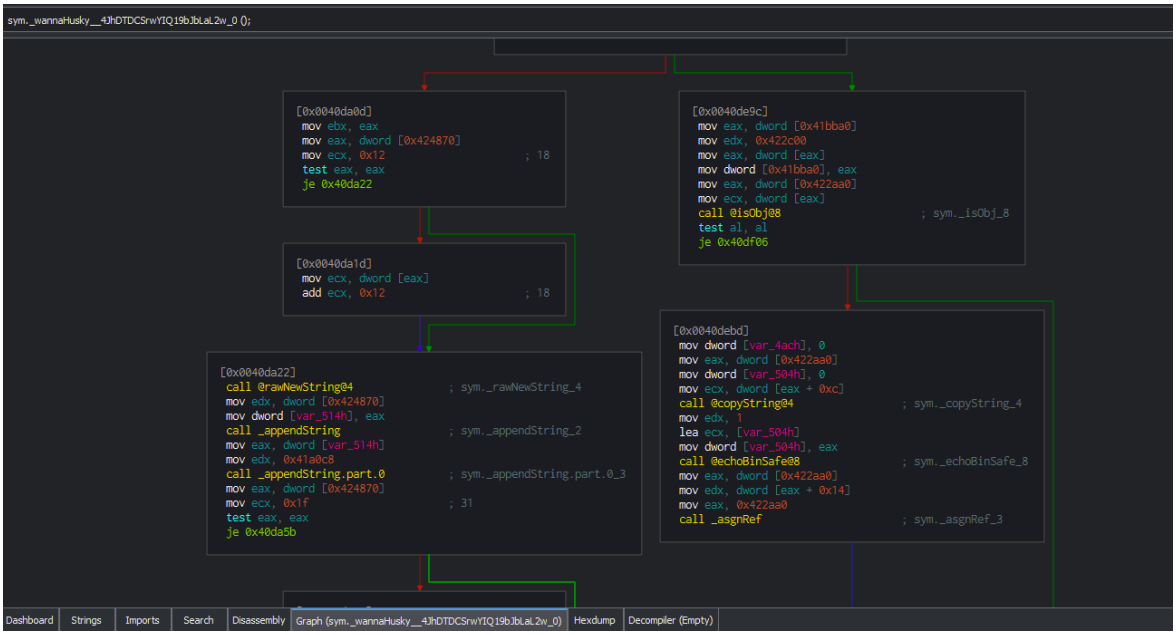
There will be three functions/strings that we will be utilizing in within Cutter.

- Sym._wannaHusky_4JhTDCSrWYIQ19bJbLaL2w_0
  - Function responsible for encryption and deletion of *cosmos.jpeg* and rewriting it as *cosmo.WANNNAHUSKY.png.*
- Sym._changeBackground_4JhDTDCSrrwYIQ19bJbLaL2w_2_0
  - Responsible for changing the Desktop Background.
- Sym._nosexecShellCmd_4
  - Responsible for spawning *ps1.ps1* on the Desktop.

Sym._wannaHusky_4JhTDCSrWYIQ19bJbLaL2w_0

```
[0x0040d9c7]
1365: @wannaHusky__4JhDTDCSrwYIQ19bJbLaL2w@0 ();
; var int32_t var_534h @ ebp-0x534
; var int32_t var_530h @ ebp-0x530
; var int32_t var_52ch @ ebp-0x52c
; var int32_t var_528h @ ebp-0x528
; var int32_t var_524h @ ebp-0x524
; var int32_t var_520h @ ebp-0x520
; var uint32_t var_51ch @ ebp-0x51c
; var uint32_t var_518h @ ebp-0x518
; var int32_t var_514h @ ebp-0x514
; var uint32_t var_510h @ ebp-0x510
; var uint32_t var_50ch @ ebp-0x50c
; var int32_t var_504h @ ebp-0x504
; var int32_t var_500h @ ebp-0x500
; var int32_t var_4f0h @ ebp-0x4f0
; var int32_t var_4d0h @ ebp-0x4d0
; var int32_t var_4b0h @ ebp-0x4b0
; var uint32_t var_4ach @ ebp-0x4ac
; var int32_t var_4a8h @ ebp-0x4a8
; var int32_t var_468h @ ebp-0x468
; var int32_t var_240h @ ebp-0x240
; var int32_t var_ch @ ebp-0xc
; var int32_t var_4h_4 @ esp+0x10
; var int32_t var_4h_3 @ esp+0x1c
; var int32_t var_8h @ esp+0x20
; var int32_t var_4h_2 @ esp+0x28
; var int32_t var_4h @ esp+0x2c
; var int32_t var_8h_2 @ esp+0x30
; var int32_t var_sp_ch @ esp+0x34
; var int32_t var_10h @ esp+0x38
; var int32_t var_14h @ esp+0x3c
push ebp
mov ebp, esp
push edi
push esi
push ebx
sub esp, 0x54c
mov eax, dword [0x41bba0]
mov dword [var_4h], 0
mov dword [var_4b0h], eax
```

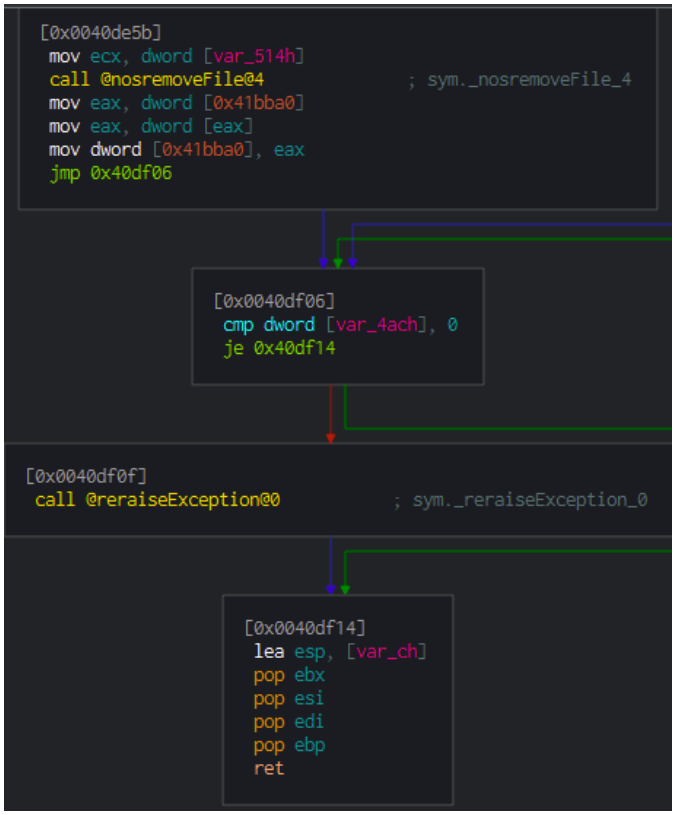
(Figure 2.3 – Main call for wannaHusky beginning to allocate different strings and functions within Cutter)



(Figure 2.4 – Main Calls breaking out into different functions)

```
[0x0040dde7]
 call @rawNewString@4                ; sym._rawNewString_4
 mov edx, dword [0x424870]
 call _appendString                  ; sym._appendString_2
 mov edx, 0x41a000
 call _appendString.part.0           ; sym._appendString.part.0_3
 mov edx, ebx
 mov ecx, eax
 call @writeFile__D6Pj9c29aCLEJP9beOWa08HYA@8 ; sym._writeFile__D6Pj9c29aCLEJP9beOWa08HYA_8
 mov ecx, 0x412100
 call @newStringStream__9aLRtgEYeRMrZKrObtoOslQ@4 ; sym._newStringStream__9aLRtgEYeRMrZKrObtoOslQ_4
 mov eax, dword [0x424870]
 mov ecx, 0x16                       ; 22
 test eax, eax
 je 0x40de27
```

(Figure 2.5 – Function where cosmo.jpeg begins to be rewritten.)



(Figure 2.6 – Final removal of cosmo.jpeg and spawn of encrypted comso.WANNACRY)

Advanced Dynamic Analysis

Further Analysis within Cutter debugging WannaHusky.

Sym._changeBackground_4JhDTDCSrrwYIQ19bJbLaL2w_2_0

*(Figure 2.7 – Decompiler of changeBackground__*** showing files being encrypted, encoded, and removed. Note: For this specific language Ghidra language must be enabled.)*

Sym. nosexecShellCmd_4

```
Decompiler (sym._wannaHusky__4JhDTDCSrwYIQ19bJbLaL2w_0)
        }
        @init__QeKCvRTxwnkv4EgDHKgXYA@20(0x20, (int32_t)&var_500h, 0x10);
        uVar6 = arg_ch;
        if (placeholder_19 != (uint32_t *)0x0) {
            uVar6 = *placeholder_19;
        }
        _encrypt__dcoBdmUaaCC9cnR23eFxSLAbcmode((int32_t)(placeholder_19 + 2), uVar6);
        @burnMem__4FZHyz34TGxTmMy6XY9cOSg@8();
        @init__QeKCvRTxwnkv4EgDHKgXYA@20(0x20, (int32_t)&var_500h, 0x10);
        if (placeholder_15 != (uint32_t *)0x0) {
            arg_ch = *placeholder_15;
        }
        _encrypt__dcoBdmUaaCC9cnR23eFxSLAbcmode((int32_t)(placeholder_15 + 2), arg_ch);
        @burnMem__4FZHyz34TGxTmMy6XY9cOSg@8();
        @encode__npLRSgmGJDNX8bfurW5iRw@12(0);
        @rawNewString@4(extraout_ECX);
        _appendString();
        _appendString.part.0();
        @writeFile__D6Pj9c29aCLEJP9beOWa08HYA@8();
        @newStringStream__9aLRtgEYeRMrZKrObtoOslQ@4();
        @rawNewString@4();
        _appendString();
        _appendString.part.0();
        iVar5 = @newFileStream__cwYJiP3D7DOTCJxCdBqBZQ@12(-1);
        if (iVar5 != 0) {
            uVar2 = @writeLine__2KoDZXJB4LmoH7PHLGmZ9cg@12(1);
            @close__y1KA3B0U09bKtU09am9a9avRYQ_4@4(uVar2);
        }
        @nosremoveFile@4();
        *(int32_t *)0x41bba0 = (int32_t *)**(int32_t **)0x41bba0;
    } else {
        *(int32_t *)0x41bba0 = (int32_t *)**(int32_t **)0x41bba0;
        cVar1 = @isObj@8();
        if (cVar1 != '\0') {
            var_4ach = 0;
            @copyString@4();
            @echoBinSafe@8();
            _asgnRef();
```

(Figure 2.8 – Decompiler of nosexecShellCmd also displaying encryption, encoding, and writeFiles.)

Indicators of Compromise

Network Indicators

Detonating WannaHusky does not beacon out to any external URL file while having REMNux, inetsim, and wireshark enabled. There has not been any attempt within ransomware[.]wannacry[.]exe to reach out to any other hosts or domain, therefore no network indicators currently.

Host-Based Indicators

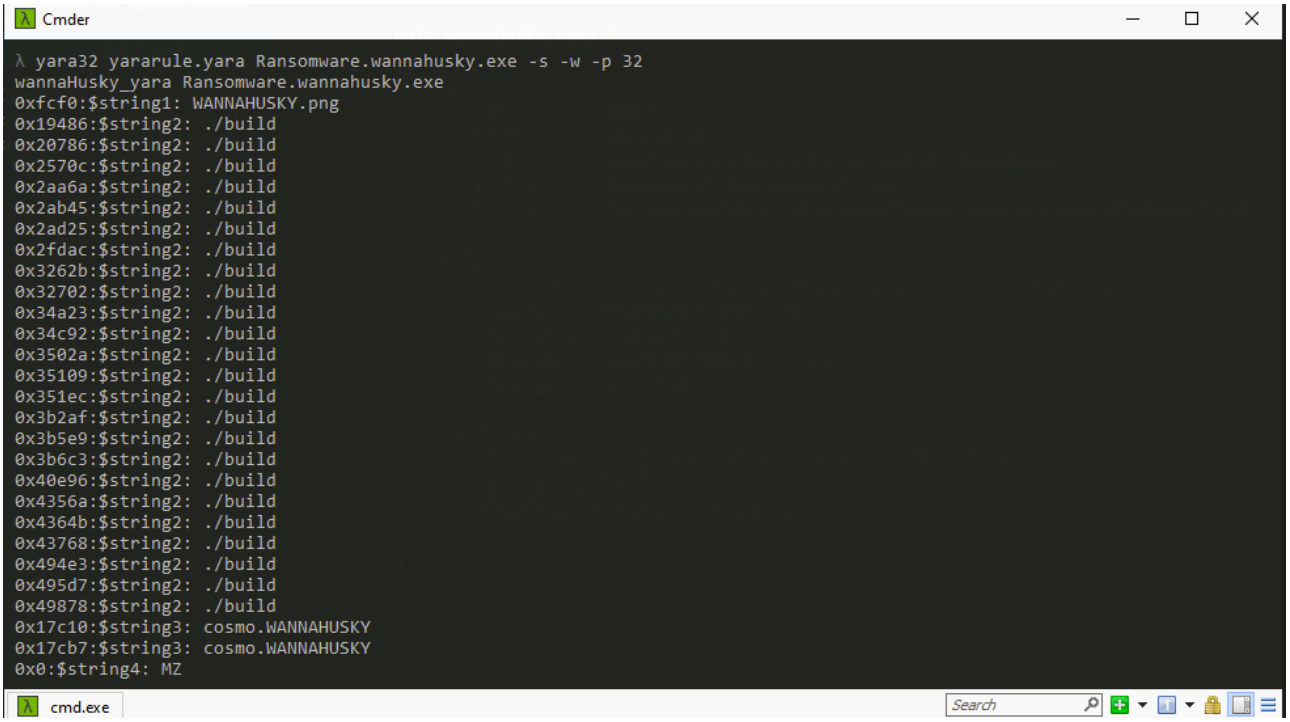
- Ransomware note saved on the Desktop both as background and *WANNAHUSKY.png*
- Spawn of ps1.ps1
- *Cmd.exe* window and a Processing tree is being executed

Yara Rules & Signatures

Yara rules based on investigation and triaging. These rules will help mitigate and detect if ransomware[.]wannahusky[.]exe will be executed on the host computer.

```
// V. Yara Rule Writing

rule wannaHusky_yara {

   meta:
      author =    "Vien"
      date =      "2022-10-20"
      desc =      "Creating Yara Rules for WannaHusky Ransomware"
      file =      "Ransomware[.]wannahusky[.]exe"
      hash =      "3d35cebcf40705c23124fdc4656a7f400a316b8e96f1f9e0c187e82a9d17dca3"

   strings:
      // Generating a set of strings to set the criteria for our rule
      $string1 =  "WANNAHUSKY.png" ascii
      $string2 =  "./build" ascii
      $string3 =  "cosmo.WANNAHUSKY" ascii
      $string4 =  "MZ"  ascii

   condition:
      // Fill out conditions that must be met to identify binary
      $string4 at 0 and
      ($string2 or $string3 or $string1)

}
```

(*Figure 2.9 – Yara rules based on criteria found within the ransomware binary)*



(Figure 3.0 – Successful Yara Rule execution against Ransomware[.]wannahusky[.]exe

Conclusion

WannaHusky[.]exe is a ransomware developed by Threat Actor HuskyHacks, where group demands crypto payment for the decryption of file *cosmos.jpeg.* However, for program to fully run, *cosmos.jpeg* must be located on the Desktop. At the time of writing, there has been no remediation or mitigation techniques to remove ransomware[.]wannahusky[.]exe at this time.