



Practical Malware Analysis & Triage

Malware Analysis Report

WannaHusky.exe Ransomware

Jan 2022 | Crypt0ace | v1.0



Table of Contents

Table of Contents	2
Executive Summary	3
High-Level Technical Summary	4
Malware Composition	5
Basic Static Analysis	5
Basic Dynamic Analysis.....	6
Advanced Static Analysis.....	7
Advanced Dynamic Analysis	11
Indicators of Compromise	12
Rules & Signatures	13
Appendices	14
A. Yara Rules:.....	14

Executive Summary

SHA256 hash	3d35cebcf40705c23124fdc4656a7f400a316b8e96f1f9e0c187e82a9d17dca3
MD5 hash	0287b38f8240a025b30c0a231ea403fc

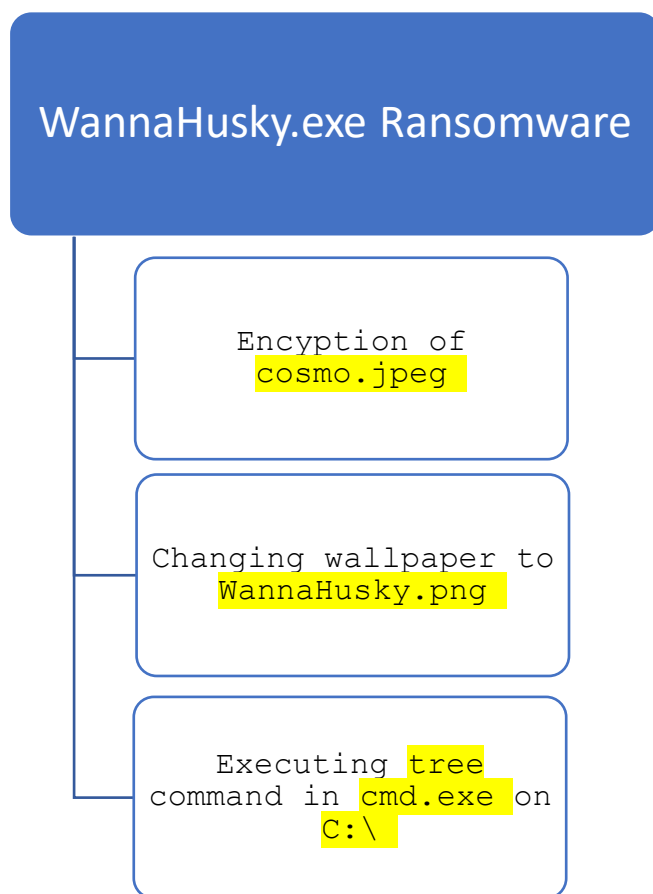
WannaHusky.exe Ransomware is a ransomware sample that was provided for analysis and triage. It has 3 stages:

1. Looks for cosmo.jpeg in the user's Desktop folder. Encrypts it and deletes the original file.
2. Creates a ps1.ps1 powershell script that is used to change the background to WannaHusky.png.
3. Runs the tree command on a command prompt.

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination. Results of which can be seen on Page 5.

High-Level Technical Summary

As stated above the WannaHusky.exe Ransomware runs in 3 stages, encryption of cosmo.jpeg, changing wallpaper and running the tree command.



Malware Composition

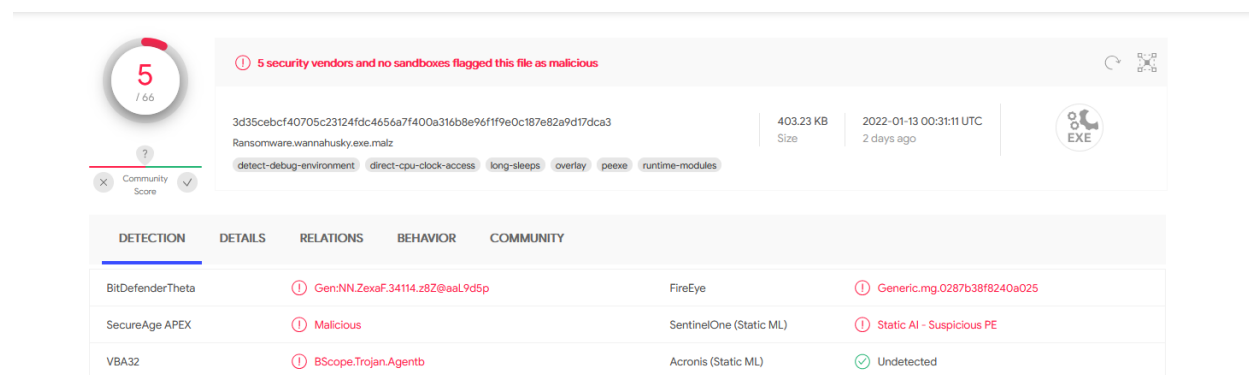
WannaHusky.exe Ransomware consisted of the only one component:

File Name	SHA256 Hash
WannaHusky.exe	3d35cebcf40705c23124fdc4656a7f400a316b8e96f1f9e0c187e82a9d17dca3

Basic Static Analysis

VirusTotal:

Submitting the binary to VirusTotal shows 5 detections out of 66



Strings:

Some interesting strings we found were classified as:

- Several mentions of nim indicate that this binary was written in nim language.
- The binary uses NimCrypto library to encrypt the cosmo.jpeg file, saves it as cosmo.WANNAHUSKY.
- Saves a WANNAHUSKY.png file on Desktop (Later seen as wallpaper as per the ps1.ps1 powershell script)
- Retrieved a powershell script that comes into play later on. (See page 11 to read the script).
- The command tree is executed on C:\

Basic Dynamic Analysis

Initial Detonation:

The initial detonation of the binary shows that upon execution, it encrypts the cosmo.jpeg file from Desktop, changes its extension to cosmo.WANNAHUSKY and deletes the original file. It then proceeds to change the wallpaper to WANNAHUSKY.png, which mentions that this is a ransomware and demands 100 husky coins in 24 hours at “hxxps[:]hussyhacks[.]dev” to decrypt the cosmo.jpeg file. It also executes a command prompt with tree command on C:\.



Conditions required for it to run successfully:

In order for the binary to run successfully, the cosmo.jpeg file needs to be on the user's desktop.



Advanced Static Analysis

Opening the binary in a disassembler, we can see 3 main functions in the NimMainModule:

- wannaHusky
- changeBackground
- nosexecShellCmd

The first function wannaHusky is responsible of encrypting the cosmo.jpeg, also can be classified as the primary function of the binary.

The second function is responsible for changing the background to WANNAHUSKY.png.

The third function is responsible of running the tree command on C:\.

```
[0x0040e052]
81: @NimMainModule@0 ();
push    ebp
mov     ecx, @TM__njFKfyRiYvmtvTKocFwDw_2@0 ; 0x40d8a1
mov     ebp, esp
sub     esp, 8
call    @nimRegisterGlobalMarker@4 ; sym.__nimRegisterGlobalMarker_4
mov     ecx, @TM__njFKfyRiYvmtvTKocFwDw_3@0 ; 0x40d894
call    @nimRegisterGlobalMarker@4 ; sym.__nimRegisterGlobalMarker_4
call    @nosGetCurrentDir@0 ; sym.__nosGetCurrentDir_0
mov     edx, eax
mov     eax, 0x424860 ; 'HB'
call    _asgnRef ; sym.__asgnRef_3
call    @nosgetHomeDir@0 ; sym.__nosgetHomeDir_0
mov     edx, eax
mov     eax, 0x424870 ; 'pHB'
call    _asgnRef ; sym.__asgnRef_3
call    @wannaHusky__4JhDTDCSrwYIQ19bJbLaL2w@0 ; sym.__wannaHusky__4JhDTDCSrwYIQ19bJbLaL2w_0
call    @changeBackground__4JhDTDCSrwYIQ19bJbLaL2w_2@0 ; sym.__changeBackground__4JhDTDCSrwYIQ19bJ...
mov     ecx, 0x411e40
leave
jmp     @nosexecShellCmd@4 ; sym.__nosexecShellCmd_4
```

i: Important Functions

```
[0x0040da5b]
call    @rawNewString@4          ; sym._rawNewString_4
mov     edx, dword [0x424870]
call    _appendString            ; sym._appendString_2
mov     edx, 0x41a0a0
call    _appendString.part.0     ; sym._appendString.part.0_3
mov     ecx, dword [var_514h]
call    @readFile__4PGnM9bWmsh0Nu7dnr3XzgA@4 ; sym._readFile__4PGnM9bWmsh0Nu7dnr3XzgA_4
mov     ecx, 0x41a078
mov     esi, eax
```

Time	Process	Operation	Path	Result	Details
2:05:...	Ransomware.w...	negoperkey	C:\Windows\System32\CurrentControlSet\Drivers\wm...	NAME NOT FOUND	Desired Access: C...
2:05:...	Ransomware.w...	CreateFile	C:\Users\crypt0ace\Desktop\cosmo.jpeg	SUCCESS	Desired Access: G...
2:05:...	Ransomware.w...	QueryStandard...	C:\Users\crypt0ace\Desktop\cosmo.jpeg	SUCCESS	AllocationSize: 1,7...
2:05:...	Ransomware.w...	ReadFile	C:\Users\crypt0ace\Desktop\cosmo.jpeg	SUCCESS	Offset: 0, Length: 1...
2:05:...	Ransomware.w...	ReadFile	C:\Users\crypt0ace\Desktop\cosmo.jpeg	SUCCESS	Offset: 1,753,088, ...
2:05:...	Ransomware.w...	ReadFile	C:\Users\crypt0ace\Desktop\cosmo.jpeg	END OF FILE	Offset: 1,754,626, ...
2:05:...	Ransomware.w...	CloseFile	C:\Users\crypt0ace\Desktop\cosmo.jpeg	SUCCESS	

Showing 61 of 284,797 events (0.021%) Backed by virtual memory

Then it writes to cosmo.WANNAHUSKY file on desktop. It also creates 2 more files, WANNAHUSKY.png and ps1.ps1 on desktop. We can see these processes in Procmon.

2.05.5	Ransomware w...	676	CreateFile	C:\Windows\SysWow64\lsipics.dll	SUCCESS	Desired Access: M...
2.05.5	Ransomware w...	676	CreateFile	C:\Windows\SysWow64\ipct4.dll	SUCCESS	Desired Access: G...
2.05.5	Ransomware w...	676	CreateFile	C:\Windows\SysWow64\ws2_32.dll	SUCCESS	Desired Access: R...
2.05.5	Ransomware w...	676	CreateFile	C:\Users\crypt0ace\Desktop\cosmo.jpeg	SUCCESS	Desired Access: G...
2.16.1	Ransomware w...	676	CreateFile	C:\Users\crypt0ace\Desktop\cosmo.WANNAHUSKY	SUCCESS	Desired Access: G...
2.16.1	Ransomware w...	676	CreateFile	C:\Users\crypt0ace\Desktop\WANNAHUSKY.png	SUCCESS	Desired Access: G...
2.16.1	Ransomware w...	676	CreateFile	C:\Users\crypt0ace\Desktop\cosmo.jpeg	SUCCESS	Desired Access: R...
2.16.1	Ransomware w...	676	CreateFile	C:\Users\crypt0ace\Desktop\ps1.ps1	SUCCESS	Desired Access: G...
2.16.1	Ransomware w...	676	CreateFile	C:\Windows\SysWow64\cmd.exe	SUCCESS	Desired Access: R...
2.16.1	Ransomware w...	676	CreateFile	C:\Windows\SysWow64\cmd.exe	SUCCESS	Desired Access: R...
2.16.1	Ransomware w...	676	CreateFile	C:\Windows\apparch\evsmain.sdb	SUCCESS	Desired Access: G...
2.16.1	Ransomware w...	676	CreateFile	C:\Windows\apparch\evsmain.sdb	SUCCESS	Desired Access: G...

```
[0x0040dd8c]
mov     dword [var_4h_4], edx
mov     edx, esi
mov     dword [esp], eax
lea     eax, [var_240h]
call    _encrypt__dcoBdmUaaCC9cnR23eFxSLAbcmode ; sym._encrypt__dcoBdmUaaCC9cnR23eFxSLAbcmode
mov     edx, 0x228 ; 552
lea     ecx, [var_240h]
call    @burnMem__4FZHyZ34TGxTmMy6XY9c0Sg@8 ; sym._burnMem__4FZHyZ34TGxTmMy6XY9c0Sg_8
cmp     dword [var_50ch], 0
je      0x40ddc1
```

After the creation of WANNAHUSKY.png file, it gets used by the ps1.ps1 powershell script.



```
[0x0040de27]
call @rawNewString@4 ; sym._rawNewString_4
mov     edx, dword [0x424870]
call    _appendString ; sym._appendString_2
mov     edx, 0x4120e0
call    _appendString.part.0 ; sym._appendString.part.0_3
mov     dword [esp], 0xffffffff ; -1
mov     edx, 1
mov     ecx, eax
call    @newFileStream__cwYJiP3D7DOTCJxCdBqBZQ@12 ; sym._newFileStream__cwYJiP3D7DOTCJxCdBqBZQ_12
mov     ecx, ecx
push    edx
test    eax, eax
jne     0x40de77
```

Figure iv: Creation of WANNAHUSKY.png

It then deletes the original cosmo.png file.

```
[0x0040de5b]
mov     ecx, dword [var 514h]
call    @nosremoveFile@4 ; sym._nosremoveFile_4
mov     eax, dword [0x41bba0]
mov     eax, dword [eax]
mov     dword [0x41bba0], eax
jmp     0x40df06
```

Figure v: Deletion of cosmo.jpeg file



We can see the execution of ps1.ps1 from here



Figure vi: Running powershell on ps1.ps1 and deleting it afterwards

And lastly, it executes the tree command on C:\.



Advanced Dynamic Analysis

The only thing required to analyze through advanced dynamic analysis was the ps1.ps1 powershell script that gets placed in the user's desktop folder. This file is responsible for changing of wallpaper. We can control the flow of execution using x32dbg and pause the execution right before the file gets deleted.

```
$code = @'
using System.Runtime.InteropServices;

namespace Win32{

    public class Wallpaper{

        [DllImport("user32.dll", CharSet=CharSet.Auto)]
        static extern int SystemParametersInfo (int uAction , int uParam , string
lpvParam , int fuWinIni) ;

        public static void SetWallpaper(string thePath){
            SystemParametersInfo(20,0,thePath,3);
        }
    }
}
'@
add-type $code

$currDir = Get-Location
$wallpaper = ".\WANNAHUSKY.PNG"
$fullpath = Join-Path -path $currDir -ChildPath $wallpaper

[Win32.Wallpaper]::SetWallpaper($fullpath)
```

We can understand that user32.dll is imported and SetWallpaper is used to change the wallpaper to WANNAHUSKY.png. This was also recovered from strings on the binary.

Indicators of Compromise

Host Based Indicators:

A file ps1.ps1 appears on the desktop for a short interval before deleting itself. The cosmo.jpeg file is encrypted with the .WANNAHUSKY file extension. The wallpaper is changed to WANNAHUSKY.png and a command prompt is spawned with tree command running on "C:\".

Network Based Indicators:

No network-based indicators were found.

Rules & Signatures

A full set of YARA rules is included in Appendix A.

Strings:

The strings added for YARA Rules:

```
@tree C:\
@Desktop\ps1.ps1
@powershell
@Desktop\ps1.ps1
@$code = '@'
@Desktop\WANNAHUSKY.png
@Desktop\cosmo.WANNAHUSKY
@COSMO
@Desktop\target\cosmo.WANNAHUSKY
@Desktop\cosmo.jpeg
```

Magic Bytes:

The magic bytes found was “MZ” that indicates it as a Portable Executable (PE).



Appendices

A. Yara Rules:

```
rule WANNAHUSKY {  
  
    meta:  
        last_updated = "2022-01-15"  
        author = "Crypt0ace"  
        description = "YARA Rules for WannaHusky Ransomware"  
  
    strings:  
        $string1 = "tree" ascii  
        $string2 = "ps1.ps1" ascii  
        $string3 = "powershell" ascii  
        $string4 = "WANNAHUSKY.png" ascii  
        $string5 = "cosmo.WANNAHUSKY" ascii  
        $string6 = "cosmo.jpeg" ascii  
        $PE_magic_byte = "MZ"  
  
    condition:  
        $PE_magic_byte at 0 and  
        ($string1 and $string2 and $string3 and $string4 and $string5 and  
$string6)  
}
```