Practical Malware Analysis & Triage

Malware Analysis Report

Ransomware Malware

Jan 2022 | Richard G | v1.0

# Table of Contents

[**Table of Contents 2**](#_heading=h.gjdgxs)

[**Executive Summary 3**](#_heading=h.30j0zll)

[**High-Level Technical Summary 4**](#_heading=h.1fob9te)

[**Malware Composition 5**](#_heading=h.3znysh7)

wannahusky[.exe 5](#_heading=h.2et92p0)

ps1.ps1[: 5](#_heading=h.tyjcwt)

[**Basic Static Analysis 6**](#_heading=h.3dy6vkm)

[**Basic Dynamic Analysis 7**](#_heading=h.1t3h5sf)

[**Advanced Static Analysis 8**](#_heading=h.4d34og8)

[**Advanced Dynamic Analysis 9**](#_heading=h.2s8eyo1)

[**Indicators of Compromise 10**](#_heading=h.17dp8vu)

[Network Indicators](#_heading=h.3rdcrjn) 10

[Host-based Indicators](#_heading=h.26in1rg) 10

[**Rules & Signatures**](#_heading=h.lnxbz9) **11**

[**Appendices 1**](#_heading=h.35nkun2)**2**

[A.](#_heading=h.1ksv4uv) Yara Rules 12

# Executive Summary

| SHA256 hash | 3D35CEBCF40705C23124FDC4656A7F400A316B8E96F1F9E0C187E82A9D17DCA3 |
| --- | --- |

WannaHusky is a ransomware malware binary provided as a '*Practical Malware Analysis & Triage*' course sample. It is a ransomware binary that is written in Nim and executes on 32-bit Windows operating systems. When detonated, this binary changes the desktop wallpaper with the image displayed on the title page of this report, looks for and encrypts a file named '*cosmo.jpeg*' with a '*.WANNAHUSKY*' file extension and spawns a command prompt that runs the 'tree' command.

A YARA signature rule is attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination (see below link).

<https://www.virustotal.com/gui/file/3d35cebcf40705c23124fdc4656a7f400a316b8e96f1f9e0c187e82a9d17dca3>

# High-Level Technical Summary

WannaHusky consists of three parts: the encryption of cosmo.jpeg, the changing of wallpaper via powershell 'ps1.ps1' script, and the spawning of a cmd.exe shell that runs the 'tree' command.



# Malware Composition

WannaHusky consists of the following components:

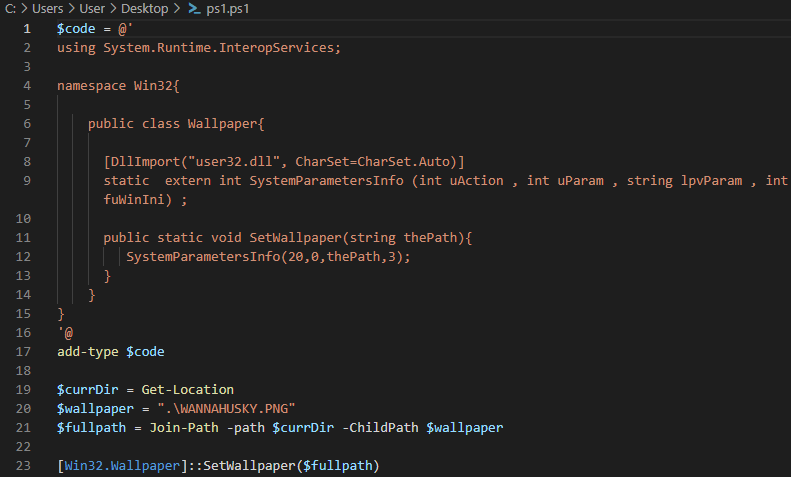
| File Name | SHA256 Hash |
| --- | --- |
| wannahusky.exe | 3D35CEBCF40705C23124FDC4656A7F400A316B8E96F1F9E0C187E82A9D17DCA3 |
| ps1.ps1 | D6317F374F879CD4E67FB4E9DDC0D283926489F4C0D6CF07D912A247E5CFDE99 |

## wannahusky.exe

The initial executable that runs and executes the WannaHusky ransomware. This executable contains all the functions that will execute various stages of the malware (this will be discussed in greater detail below).

ps1.ps1:

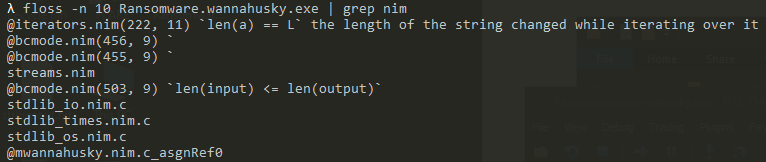
A PowerShell file that runs and sets the desktop wallpaper to '*WANNAHUSKY.png*' as shown on the title page of this report. It contains the following content:



*Fig 1: PowerShell script dropped from WannaHusky.exe file. This file will later be removed from disk during process execution.*

# Basic Static Analysis

**Interesting Strings**  
There were many interesting strings obtained from basic static analysis of this binary. Firstly, there are many references to '*nim*' and '*nim*' libraries, giving us an insight into the binary's creation language. Subsequent analysis confirms this theory.



There are also references to 'nim' crypto libraries, which were early indicators that this was the 'ransomware' class of malware:  
  
  
The most interesting output from strings, however, is what would later turn out to be identified as the entire output of the '*ps1.ps1*' file shown above in the 'Malware Composition' section of this report.

# Basic Dynamic Analysis

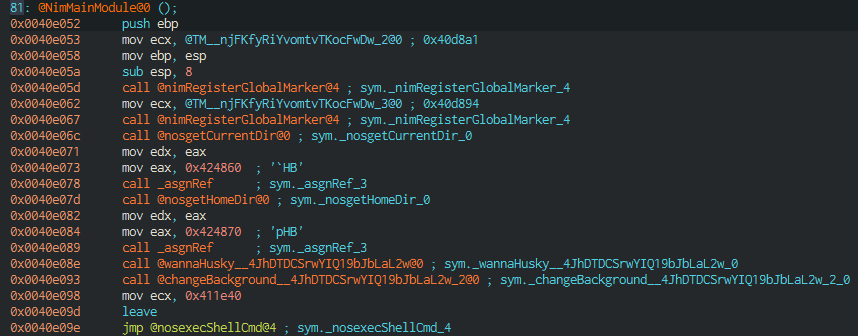
**Signs of the Initial Infection**  
Upon execution of this binary, we see quite a number of things happen. Firstly, we notice that the desktop wallpaper is changed to an image of a husky graphic with the following text: "*that picture of cosmo on your desktop is now encrypted! To save him, you must send 100 Huskycoin to hXXps://huskyhacks[.]dev hurry! You have 24 hours before we delete cosmo*".  
  
  


The file that was once named '*cosmo.jpeg*' on the Desktop has had its thumbnail changed and it's file extension changed to '*cosmo.WANNAHUSKY*'. We can also see a command prompt window displaying the output of the '*tree*' command.

# Advanced Static Analysis

**Disassembling via Cutter**

After opening '*wannahusky.exe*' in Cutter, we can delve a little deeper into the functionality of this binary. There are numerous functions with 'main' in their names, however we quickly learn that '*NimMainModule\_0*' is the function worth investigating.

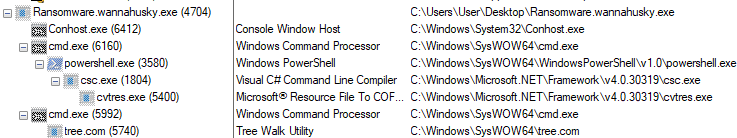


*wannaHusky\_\_4JhDTDCSrwYIQ19bJbLaL2w@0*All actions to do with the '*cosmo.jpeg*' file, including the encryption and file extension changing happen within this function.   
  
*changeBackground\_\_4JhDTDCSrwYIQ19bJbLaL2w\_2@0*  
This function is responsible for assembling the wallpaper image '*WANNAHUSKY.png*', loading the '*ps1.ps1*' script and executing it to change the desktop wallpaper.  
  
*nosexecShellCmd@4*  
This function is responsible for all actions pertaining to the cmd.exe window that spawns, running the 'tree' command.

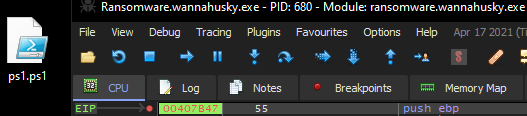
# Advanced Dynamic Analysis

**Further investigation with ProcMon and x32dbg**

The following image shows the process tree at time of binary execution. The powershell.exe process is responsible for assembling the wallpaper and running the '*ps1.ps1*' script that actually sets the wallpaper. The tree command in the command prompt window appears to be a decoy and serves no further functionality.

  
  
  
  
*Fig 2: Process tree showing execution processes. Below that is the ps1 file that is run at time of execution, and finally the exact 'tree' command that is run.*

As this powershell file deletes itself from disk after a short period of time, it was obtained by setting a breakpoint at address 0x00407b47 (this is the address of the '*@nosremoveFile@4'* function - the jmp call to which is located within the '*@changeBackground*' function), this prevents the file deletion from occurring. The contents of this PowerShell script are seen above in the 'Malware Composition' section of the report in Fig 1.

  
*Fig 3: PowerShell script obtained by setting a breakpoint and controlling process execution flow in x32dbg.*

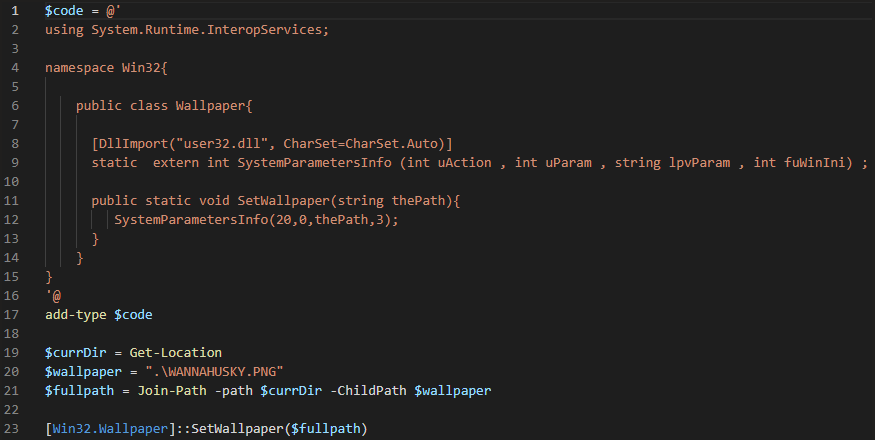
# Indicators of Compromise

The full list of IOCs can be found in the Appendices.

## Network Indicators

No network indicators were found for this binary.

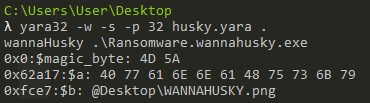
## Host-based Indicators

'*wannahusky.exe*' - 3D35CEBCF40705C23124FDC4656A7F400A316B8E96F1F9E0C187E82A9D17DCA3  
  
'*ps1.ps1*' - D6317F374F879CD4E67FB4E9DDC0D283926489F4C0D6CF07D912A247E5CFDE99  
  
  
  
'*cosmo.WANNAHUSKY*':



# Rules & Signatures

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

$magic\_byte = { 4D 5A }  
This is the hex equivalent of the ascii characters 'MZ' which denotes that a file is a Portable Executable and can be used to narrow our search for binaries only.  
  
$a = {40 77 61 6e 6e 61 48 75 73 6b 79}   
This string is looking for the ascii characters of "@wannaHusky" - this is a function name within the binary itself and is a good string to match on as it is a fairly unique string.  
  
$b = "@Desktop\\WANNAHUSKY.png" ascii  
This looks for the ascii string of the above, which is a file that is compiled and set as the desktop wallpaper during binary execution.   
  
All three of these conditions are used in conjunction to locate this ransomware binary.   


# Appendices

## Yara Rules

Full Yara repository located at: https://github.com/ventdrop/PMATlabs

rule wannaHusky {

meta:

last\_updated = "2022-01-26"

author = "@ventdrop"

description = "A rule for locating wannaHusky ransomware"

strings:

$magic\_byte = { 4D 5A } // MZ byte

$a = {40 77 61 6e 6e 61 48 75 73 6b 79} // @wannaHusky function in hex

$b = "@Desktop\\WANNAHUSKY.png" ascii

condition:

($magic\_byte at 0x00) and ($a and $b)

}

## 