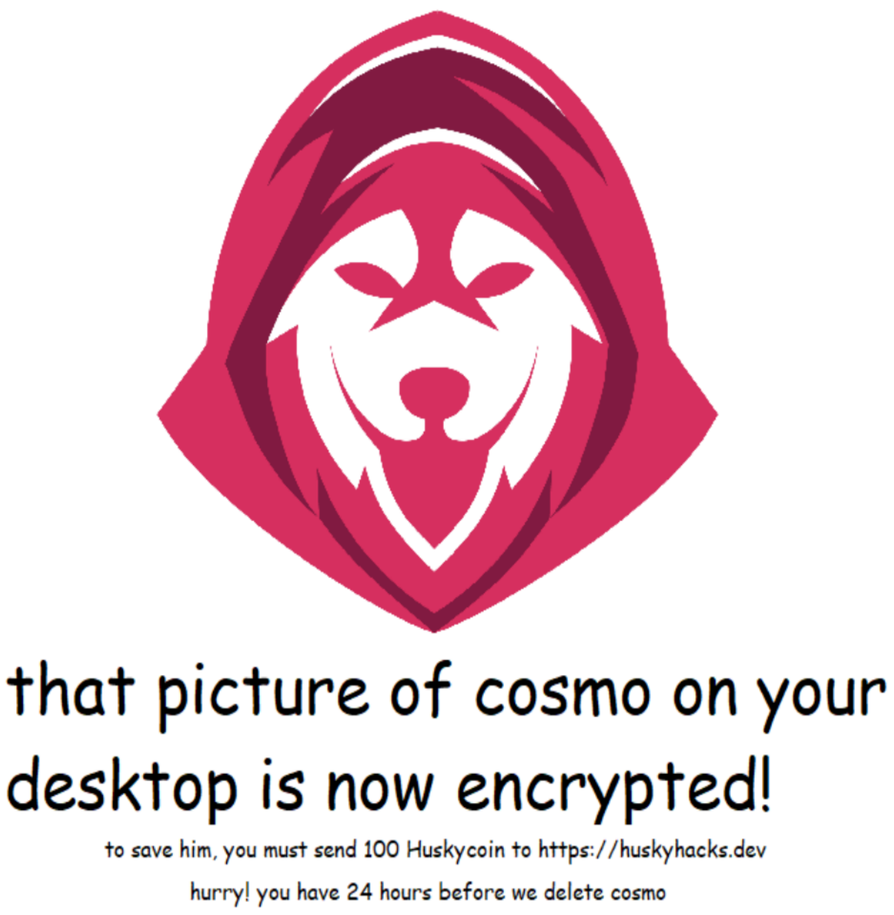
Practical Malware Analysis & Triage



that picture of cosmo on your desktop is now encrypted!

to save him, you must send 100 Huskycoin to https://huskyhacks.dev

hurry! you have 24 hours before we delete cosmo

Malware Analysis Report

Ransomware Malware

Jan 2022 | Richard G | v1.0

Table of Contents

Executive Summary

SHA256 hash	3D35CEBCF40705C23124FDC4656A7F400A316B8E96F1F9E0C187E82A9D17DCA3

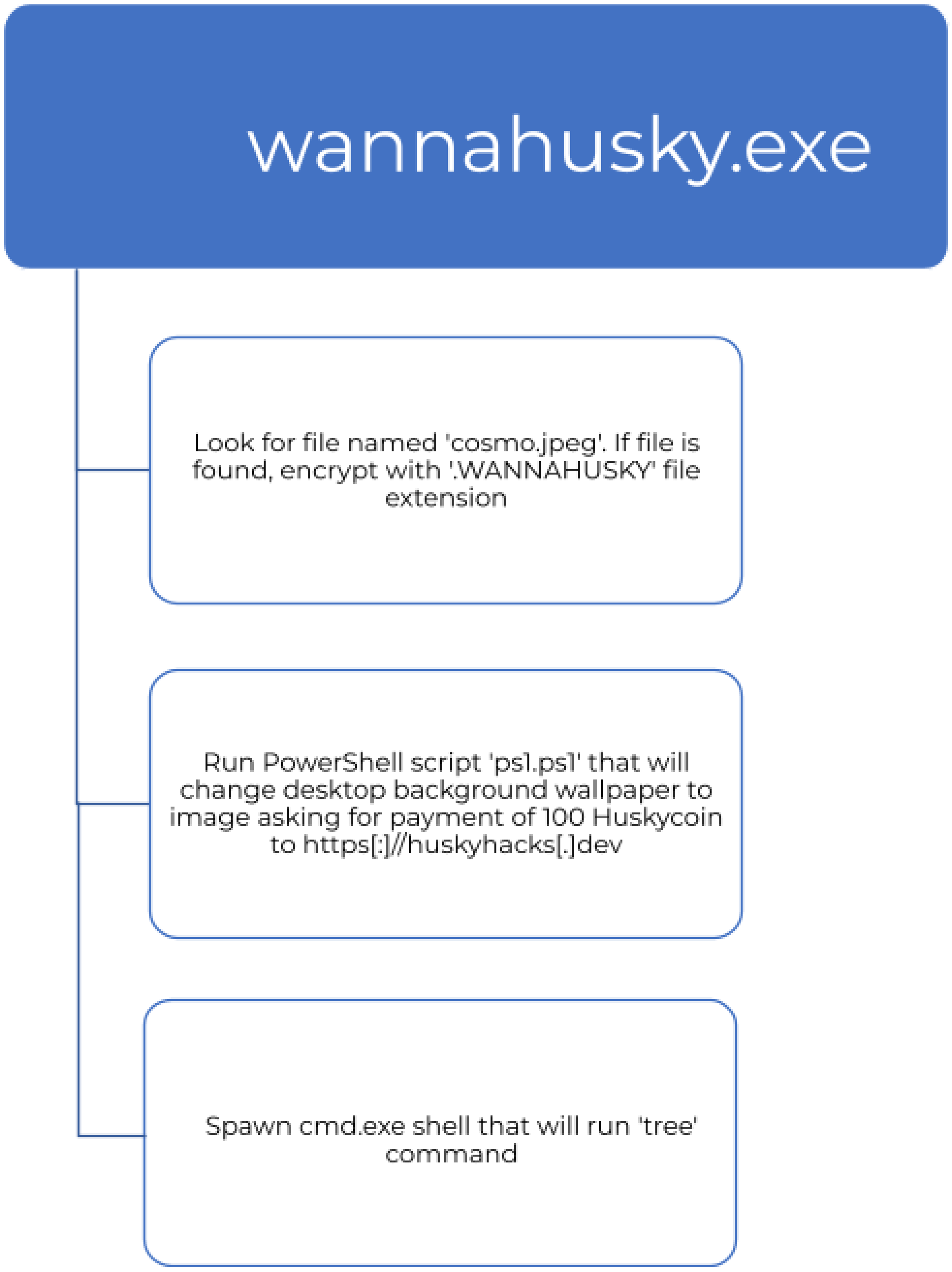
WannaHusky is a ransomware malware binary provided as a '*Practical Malware Analysis & Triage*' course sample. It is a ransomware binary that is written in Nim and executes on 32-bit Windows operating systems. When detonated, this binary changes the desktop wallpaper with the image displayed on the title 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.


wannahusky.exe
Look for file named 'cosmo.jpeg'. If file is found, encrypt with '.WANNAHUSKY' file extension
Run PowerShell script 'ps1.ps1' that will change desktop background wallpaper to image asking for payment of 100 Huskycoin to https[:]//huskyhacks[.]dev
Spawn cmd.exe shell that will run 'tree' command

Malware Composition

WannaHusky consists of the following components:

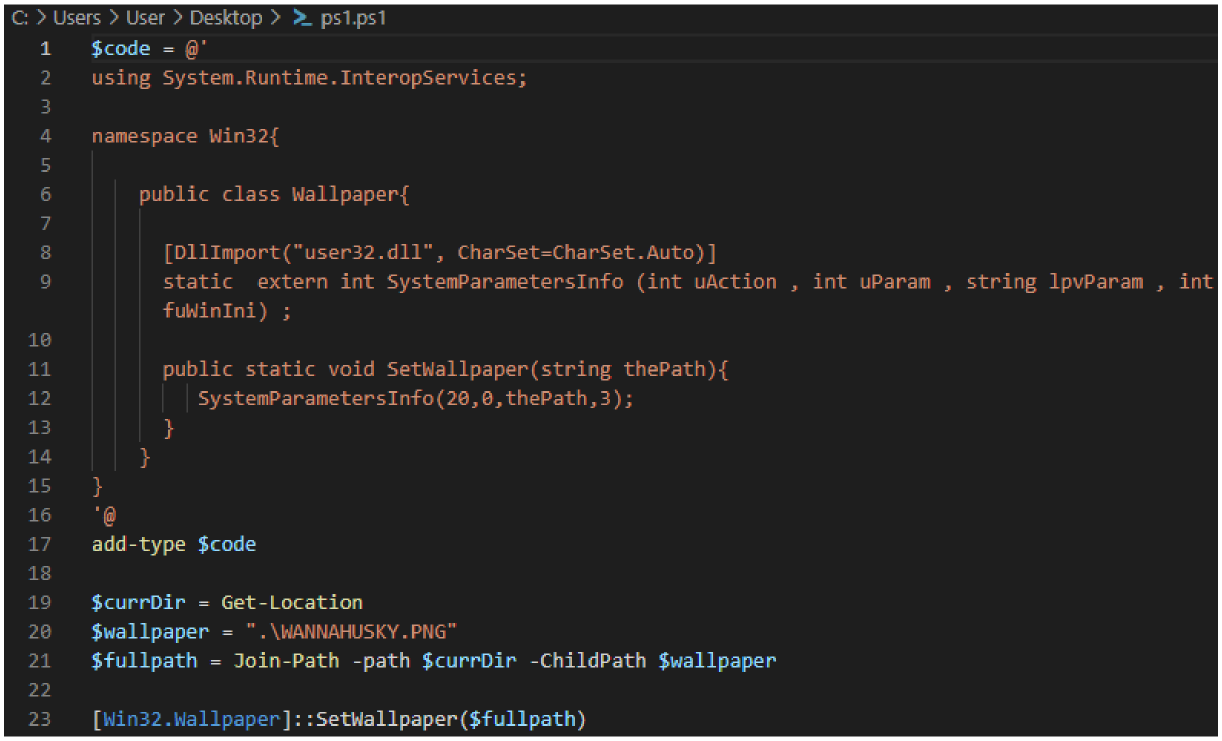
File Name	SHA256 Hash
wannahusky.exe	3D35CEBCF40705C23124FDC4656A7F400A316B8E96F1F9E0C187E82A9D17DCA3
ps1.ps1	D6317F374F879CD4E67FB4E9DDC0D283926489F4C0D6CF07D912A247E5CFDE99

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:



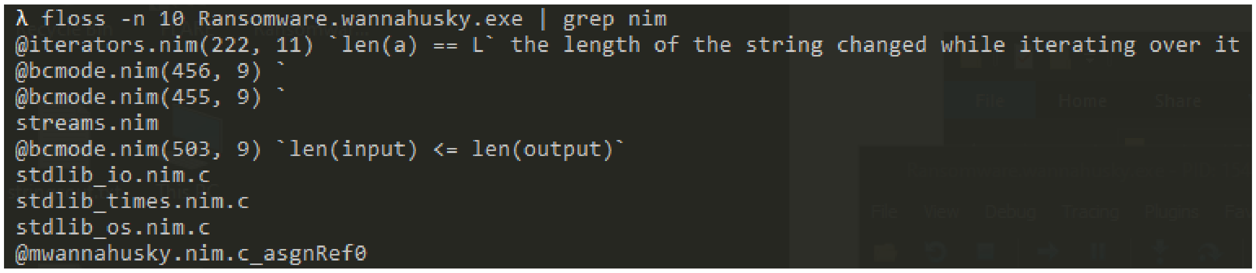
```
C: > Users > User > Desktop > ps1.ps1
$code = @'
using System.Runtime.InteropServices;

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)
```

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

Basic Static Analysis

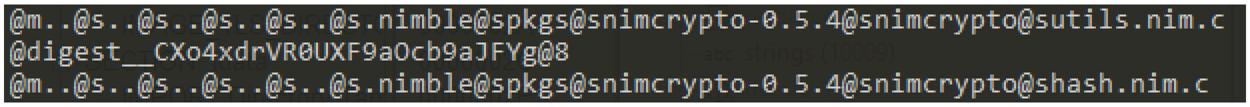
Interesting Strings

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.



```
λ floss -n 10 Ransomware.wannahusky.exe | grep nim
@iterators.nim(222, 11) `len(a) == L` the length of the string changed while iterating over it
@bcmode.nim(456, 9) `
@bcmode.nim(455, 9) `
streams.nim
@bcmode.nim(503, 9) `len(input) <= len(output)`
stdlib_io.nim.c
stdlib_times.nim.c
stdlib_os.nim.c
@mwannahusky.nim.c_asgnRef0
```

There are also references to 'nim' crypto libraries, which were early indicators that this was the 'ransomware' class of malware:



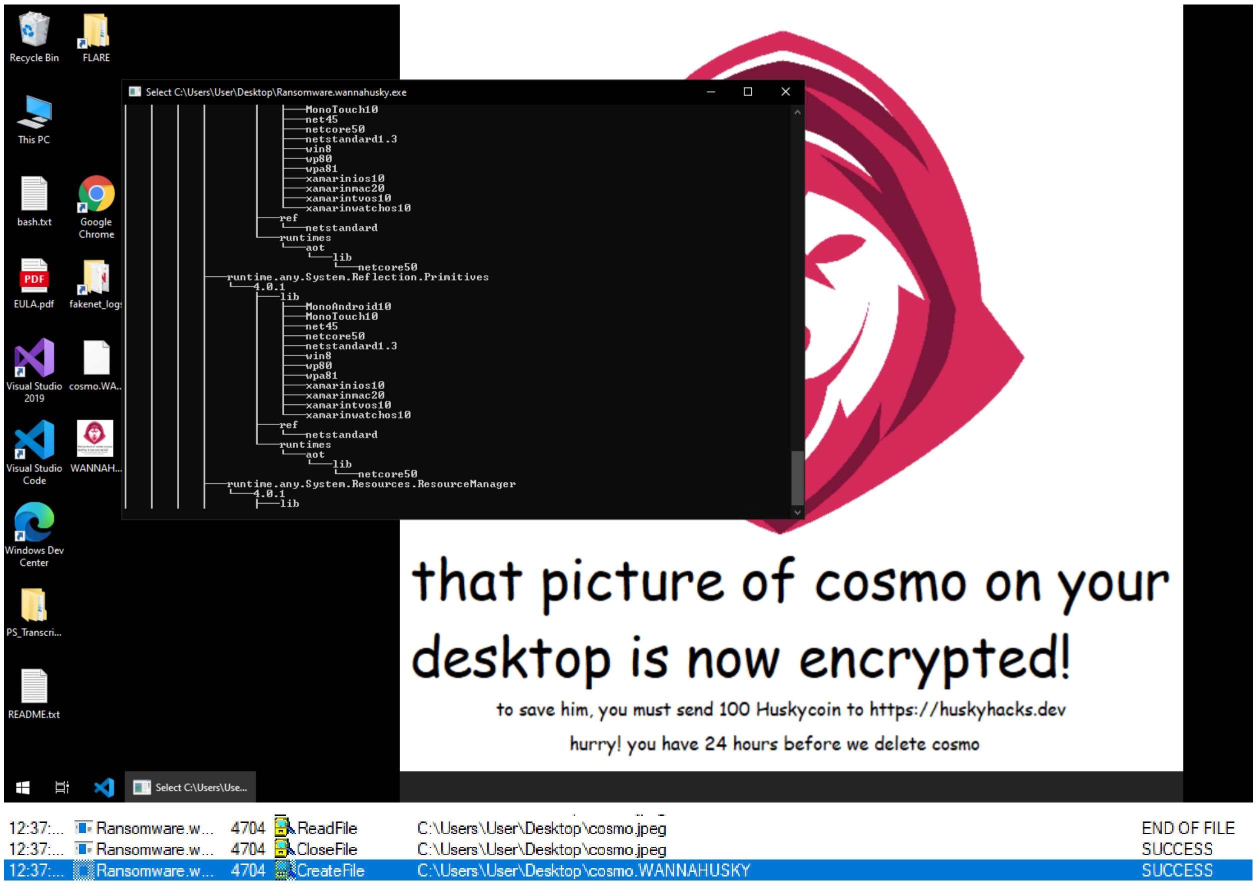
```
@m..@s..@s..@s..@s..@s.nimble@spkgs@snimcrypto-0.5.4@snimcrypto@sutils.nim.c
@digest__CXo4xdrVR0UXF9aOcb9aJFYg@8
@m..@s..@s..@s..@s..@s.nimble@spkgs@snimcrypto-0.5.4@snimcrypto@shash.nim.c
```

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*".



Select C:\Users\User\Desktop\Ransomware.wannahusky.exe

```
│   │   │   │       │   ├───MonoTouch10
│   │   │   │       │   ├───net45
│   │   │   │       │   ├───netcore50
│   │   │   │       │   ├───netstandard1.3
│   │   │   │       │   ├───win8
│   │   │   │       │   ├───wp80
│   │   │   │       │   ├───wpa81
│   │   │   │       │   ├───xamarinios10
│   │   │   │       │   ├───xamarinmac20
│   │   │   │       │   ├───xamarintvos10
│   │   │   │       │   └───xamarinwatchos10
│   │   │   │       ├───ref
│   │   │   │       │   └───netstandard
│   │   │   │       └───runtimes
│   │   │   │           └───aot
│   │   │   │               └───lib
│   │   │   │                   └───netcore50
│   │   │   ├───runtime.any.System.Reflection.Primitives
│   │   │   │   └───4.0.1
│   │   │   │       ├───lib
│   │   │   │       │   ├───MonoAndroid10
│   │   │   │       │   ├───MonoTouch10
│   │   │   │       │   ├───net45
│   │   │   │       │   ├───netcore50
│   │   │   │       │   ├───netstandard1.3
│   │   │   │       │   ├───win8
│   │   │   │       │   ├───wp80
│   │   │   │       │   ├───wpa81
│   │   │   │       │   ├───xamarinios10
│   │   │   │       │   ├───xamarinmac20
│   │   │   │       │   ├───xamarintvos10
│   │   │   │       │   └───xamarinwatchos10
│   │   │   │       ├───ref
│   │   │   │       │   └───netstandard
│   │   │   │       └───runtimes
│   │   │   │           └───aot
│   │   │   │               └───lib
│   │   │   │                   └───netcore50
│   │   │   ├───runtime.any.System.Resources.ResourceManager
│   │   │   │   └───4.0.1
│   │   │   │       ├───lib
```

that picture of cosmo on your desktop is now encrypted!

to save him, you must send 100 Huskycoin to https://huskyhacks.dev

hurry! you have 24 hours before we delete cosmo

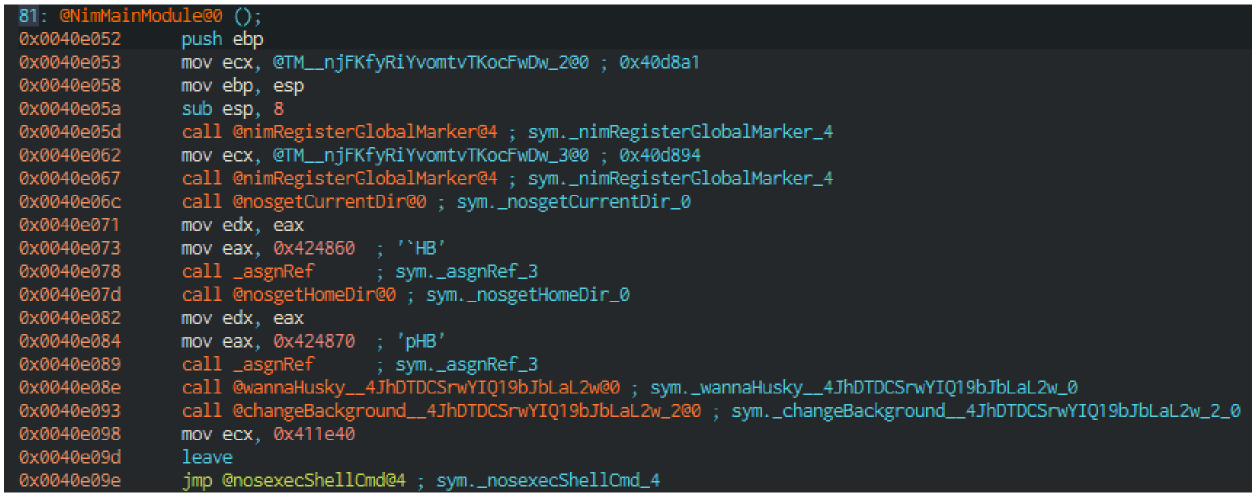
```
12:37:...  Ransomware.w...  4704  ReadFile    C:\Users\User\Desktop\cosmo.jpeg           END OF FILE
12:37:...  Ransomware.w...  4704  CloseFile   C:\Users\User\Desktop\cosmo.jpeg           SUCCESS
12:37:...  Ransomware.w...  4704  CreateFile  C:\Users\User\Desktop\cosmo.WANNAHUSKY     SUCCESS
```

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.



```
81: @NimMainModule@0 ();
0x0040e052      push ebp
0x0040e053      mov ecx, @TM__njFKfyRiYvomtvTKocFwDw_2@0 ; 0x40d8a1
0x0040e058      mov ebp, esp
0x0040e05a      sub esp, 8
0x0040e05d      call @nimRegisterGlobalMarker@4 ; sym._nimRegisterGlobalMarker_4
0x0040e062      mov ecx, @TM__njFKfyRiYvomtvTKocFwDw_3@0 ; 0x40d894
0x0040e067      call @nimRegisterGlobalMarker@4 ; sym._nimRegisterGlobalMarker_4
0x0040e06c      call @nosgetCurrentDir@0 ; sym._nosgetCurrentDir_0
0x0040e071      mov edx, eax
0x0040e073      mov eax, 0x424860  ; '`HB'
0x0040e078      call _asgnRef      ; sym._asgnRef_3
0x0040e07d      call @nosgetHomeDir@0 ; sym._nosgetHomeDir_0
0x0040e082      mov edx, eax
0x0040e084      mov eax, 0x424870  ; 'pHB'
0x0040e089      call _asgnRef      ; sym._asgnRef_3
0x0040e08e      call @wannaHusky__4JhDTDCSrwYIQ19bJbLaL2w@0 ; sym._wannaHusky__4JhDTDCSrwYIQ19bJbLaL2w_0
0x0040e093      call @changeBackground__4JhDTDCSrwYIQ19bJbLaL2w_2@0 ; sym._changeBackground__4JhDTDCSrwYIQ19bJbLaL2w_2_0
0x0040e098      mov ecx, 0x411e40
0x0040e09d      leave
0x0040e09e      jmp @nosexecShellCmd@4 ; sym._nosexecShellCmd_4
```

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.

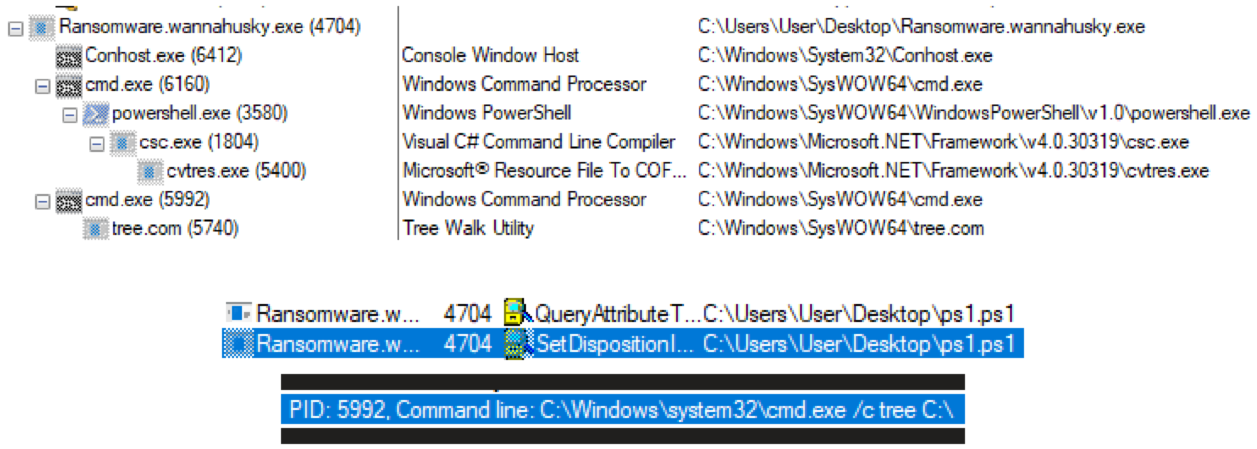
nosexecShellCmd@4

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

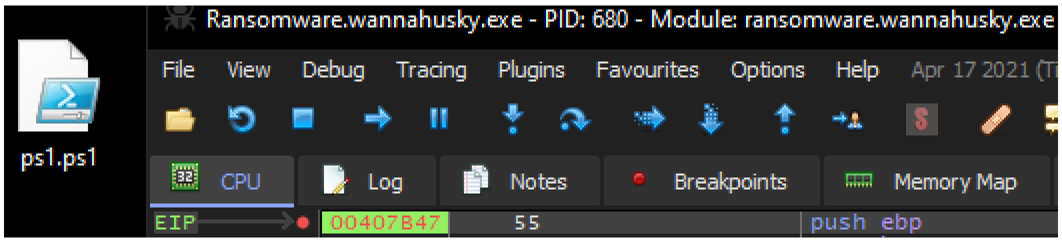
The following image shows the process tree at time of binary execution. The powershell.exe process is responsible for assembling the wallpaper and running the '*ps1.ps1*' script that actually sets the wallpaper. The tree command in the command prompt window appears to be a decoy and serves no further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



```
Ransomware.wannahusky.exe (4704)                             C:\Users\User\Desktop\Ransomware.wannahusky.exe
  Conhost.exe (6412)        Console Window Host              C:\Windows\System32\Conhost.exe
  cmd.exe (6160)            Windows Command Processor        C:\Windows\SysWOW64\cmd.exe
    powershell.exe (3580)   Windows PowerShell               C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
      csc.exe (1804)        Visual C# Command Line Compiler  C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe
        cvtres.exe (5400)   Microsoft® Resource File To COF... C:\Windows\Microsoft.NET\Framework\v4.0.30319\cvtres.exe
  cmd.exe (5992)            Windows Command Processor        C:\Windows\SysWOW64\cmd.exe
    tree.com (5740)         Tree Walk Utility                C:\Windows\SysWOW64\tree.com

Ransomware.w...  4704  QueryAttributeT... C:\Users\User\Desktop\ps1.ps1
Ransomware.w...  4704  SetDispositionI... C:\Users\User\Desktop\ps1.ps1

PID: 5992, Command line: C:\Windows\system32\cmd.exe /c tree C:\
```

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.



```
Ransomware.wannahusky.exe - PID: 680 - Module: ransomware.wannahusky.exe
EIP  00407B47  55  push ebp
```

ps1.ps1
File View Debug Tracing Plugins Favourites Options Help Apr 17 2021
CPU Log Notes Breakpoints Memory Map

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.

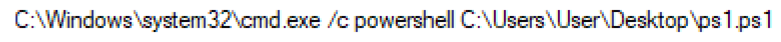
Network Indicators

No network indicators were found for this binary.

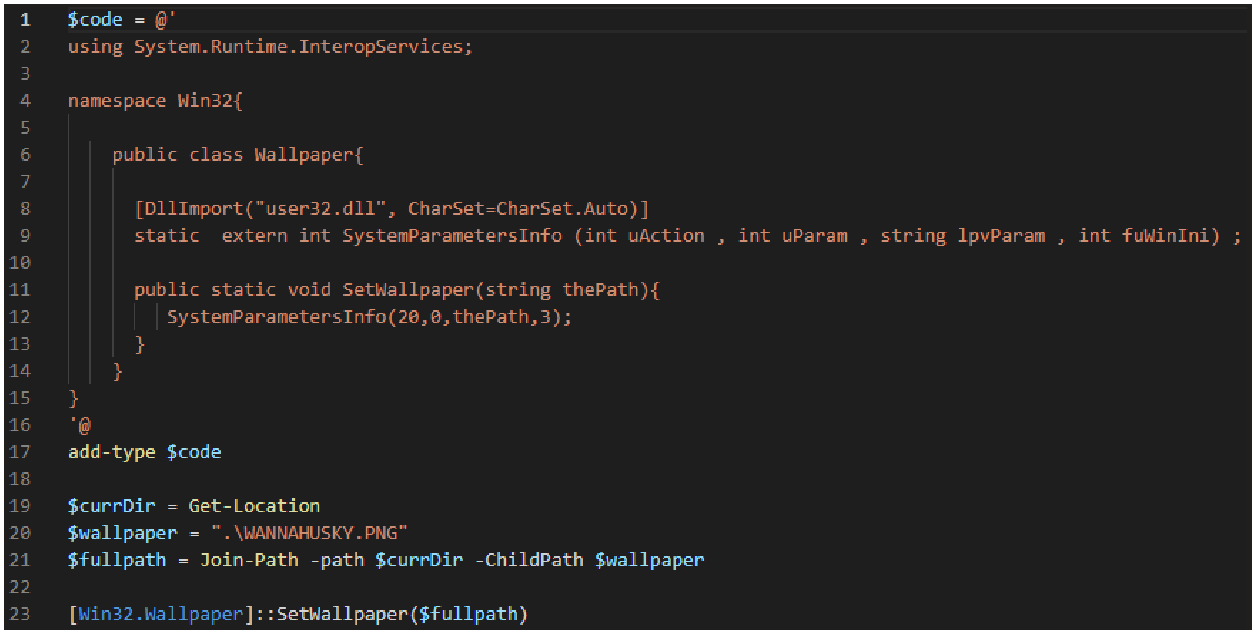
Host-based Indicators

'*wannahusky.exe*' -
3D35CEBCF40705C23124FDC4656A7F400A316B8E96F1F9E0C187E82A9D17DCA3

'*ps1.ps1*' -
D6317F374F879CD4E67FB4E9DDC0D283926489F4C0D6CF07D912A247E5CFDE99

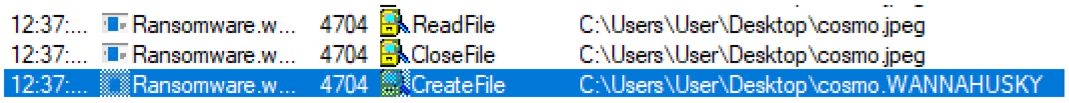


```
C:\Windows\system32\cmd.exe /c powershell C:\Users\User\Desktop\ps1.ps1
```



```
$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)
```

'*cosmo.WANNAHUSKY*':



```
12:37:...  Ransomware.w...  4704  ReadFile    C:\Users\User\Desktop\cosmo.jpeg
12:37:...  Ransomware.w...  4704  CloseFile   C:\Users\User\Desktop\cosmo.jpeg
12:37:...  Ransomware.w...  4704  CreateFile  C:\Users\User\Desktop\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.

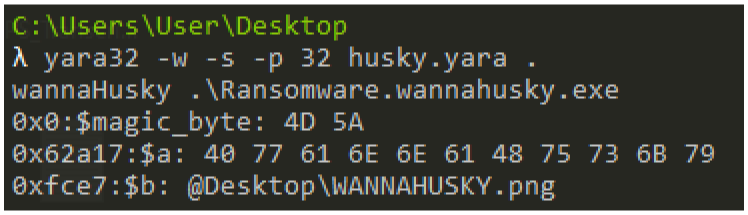
$a = {40 77 61 6e 6e 61 48 75 73 6b 79}

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.



```
C:\Users\User\Desktop
λ yara32 -w -s -p 32 husky.yara .
wannaHusky .\Ransomware.wannahusky.exe
0x0:$magic_byte: 4D 5A
0x62a17:$a: 40 77 61 6E 6E 61 48 75 73 6B 79
0xfce7:$b: @Desktop\WANNAHUSKY.png
```

Appendices

A. 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)
}
```