Icon

Description automatically generated

Practical Malware Analysis & Triage

Malware Analysis Report

Ransomware.wannahusky.exe

Feb 2024 | Amna Jasser | v1.0

Ransomware.wannahusky.exe:

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

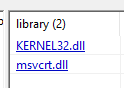
Looking at the strings output:

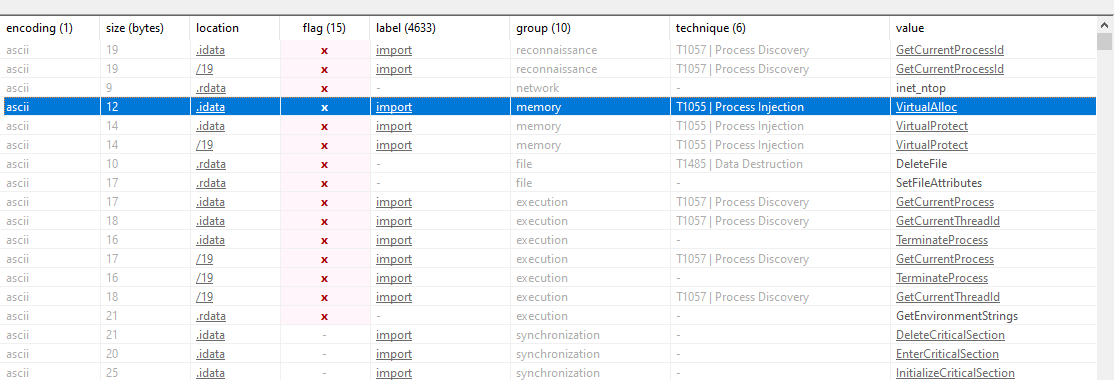
We see NIM in use:

**PESTUDIO:**

A screenshot of a computer

Description automatically generated





0000EC10 fatal.nim

0000EC49 io.nim

0000EE94 fatal.nim

0000F267 @iterators.nim(222, 11) `len(a) == L` the length of the string changed while iterating over it

0000F72C oserr.nim

0000F867 @bcmode.nim(456, 9) `

0000F8A7 @bcmode.nim(455, 9) `

0000F968 streams.nim

It uses NIM as a programming language.

0000FAA7 @$code = @'

0000FAB3 using System.Runtime.InteropServices;

0000FADA namespace Win32{

0000FAEB

0000FAF0 public class Wallpaper{

0000FB0D [DllImport("user32.dll", CharSet=CharSet.Auto)]

0000FB43 static extern int SystemParametersInfo (int uAction , int uParam , string lpvParam , int fuWinIni) ;

0000FBB0 public static void SetWallpaper(string thePath){

0000FBE7 SystemParametersInfo(20,0,thePath,3);

0000FC16 }

0000FC1E }

0000FC29 add-type $code

0000FC39 $currDir = Get-Location

0000FC51 $wallpaper = ".\WANNAHUSKY.PNG"

0000FC71 $fullpath = Join-Path -path $currDir -ChildPath $wallpaper

0000FCAD [Win32.Wallpaper]::SetWallpaper($fullpath)

0000FCE7 @Desktop\WANNAHUSKY.png

This is a clear code snippet, that changes the wallpaper to an image WANNAHUSKY.png

0000FD5D ~rIDATx^

0000FDA5 }1JsPt.:

0000FF10 F</@

0000FF96 ?!My

0000FFB4 ?!dM0

0001002A Xn#i

00010047 ?!Y@&

00010059 UBH}

000100DC wsR[

00010172 ?CJ^J

00010209 5D/G

00010235 r\ s

00010268 r8 U$+

000102A2 !M!7

00010307 Qir\_

00010325 |Idm|

0001040F 'We`4\*

0001042E t-r}

0001043F O6Of

000104AD c%VN

000104CB Ob%V

000104FA +=|C

00010530 p Vb

00010653 Xyc5

000107BB dy`%

00010837 NXj%

000108A3 w/==

00010941 %d}0

0001096C `3P`'\

000109D6 I#`M

00010B09 CP7P|

00010B1B `,Tt+

Junk strings potentially means it is either an image or extra binary data being reference by the sample. It could potentially be an encoded/encrypted/compressed payload.

00017CCF @Desktop\cosmo.jpeg

0000FC51 $wallpaper = ".\WANNAHUSKY.PNG"

0000FCE7 @Desktop\WANNAHUSKY.png

@Desktop\ps1.ps1

00017C07 @Desktop\cosmo.WANNAHUSKY

Here we see what could be the wallpaper image, and we see two version of cosmo one with regular PNG type image and the other is WANAHUSKY.

0002ACAA GNU C99 9.2-win32 20191008 -m32 -mtune=generic -march=pentiumpro -g -O2 -std=gnu99 -fno-PIE

0002AD07 ./mingw-w64-crt/crt/dllargv.c

0002AD25 ./build/i686-w64-mingw32-i686-w64-mingw32-crt

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-\_ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/

static extern int SystemParametersInfo (int uAction , int uParam , string lpvParam , int fuWinIni) ;

The provided information appears to be a snippet of a build or compilation log using the MinGW-w64 toolchain for a 32-bit Windows target, and it includes details about the compiler version, flags, source file paths, and output directory paths.

Static Analysis using Cutter:



In this main function we have 7 function calls, the significant ones are:

1. wannaHusky:
2. changeBackround: this function as shown below the code snippet, it changes the background wallpaper to WANAHUSKY.png
3. nosexecShellCmd: it Excexute a shell code, which it just end all processes and exit.

Looking at WannaHusky function: it looks like it encrypts a file, I would suggest at this point it would be cosmo.jpeg image file and write down its new encrypted version to Destop with WANNAHUSKY file type.

A screenshot of a computer program

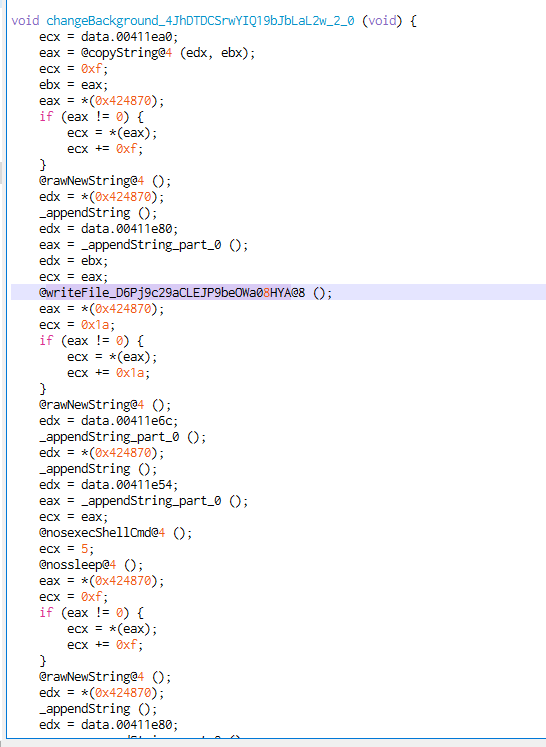
Description automatically generated



A screenshot of a computer code

Description automatically generated

Decompiled we see writefile function, that write “WANNAHUSKY.png” image we say earlier, and then it execute the poweshell ecript to set the background image.



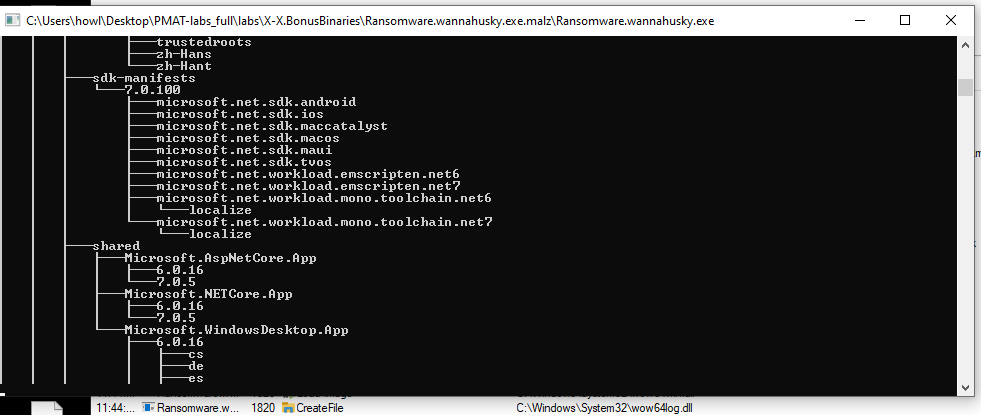


Dynamic Analysis:

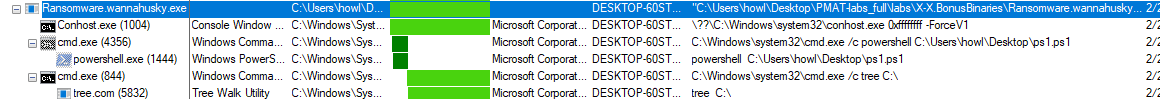
**Before executing the sample:** Make sure Set-ExecutionPolicy Unrestricted -Scope CurrentUser.

When excuting the sample, the first thing we see is a command prompt woth tree c:/ command:

At first if conso.jpeg does not exit then it will show cannot open “cosmo.jpeg” and continue to run tree c: then exit, if the image exists then it will continue to execute the code.



From procmon we see that it runs this command along with poweshell script PS1.PS1



C:\Windows\system32\cmd.exe /c tree C:\



The code snippet for 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

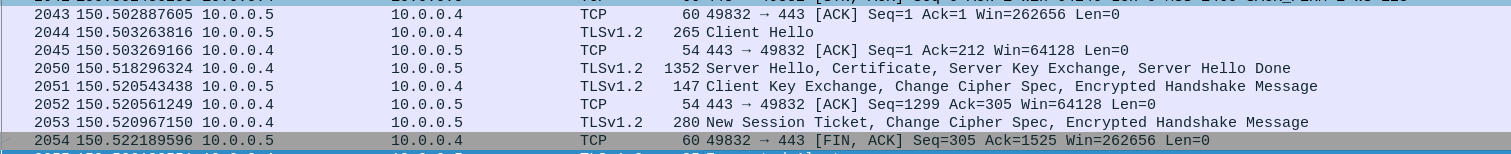
$currDir = Get-Location

$wallpaper = ".\WANNAHUSKY.PNG"

$fullpath = Join-Path -path $currDir -ChildPath $wallpaper

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

this script sets the desktop wallpaper to the specified image (in this case, "WANNAHUSKY.PNG") by leveraging a C# class through the Add-Type cmdlet in PowerShell.



There also was a TLS traffic using INETSIM to the gateway, this indicated that it gets the key to encrypt files?



The downloaded image from the sample malware.

A close-up of a logo

Description automatically generated

The encrypted image after executing the sample.



The ransom note:



## Yara Rules

rule YARA\_example {

    meta:

        description = "Ransomware.wannahusky.exe"

        sha256 = "3D35CEBCF40705C23124FDC4656A7F400A316B8E96F1F9E0C187E82A9D17DCA3"

    strings:

        $string1="$wallpaper = '.\\WANNAHUSKY.PNG'" ascii

        $string2 = "@Desktop\\WANNAHUSKY.png" ascii

        $string3="@Desktop\\cosmo.jpeg" ascii

        $string4="@Desktop\\cosmo.WANNAHUSKY" ascii

        $string5="@Desktop\\ps1.ps1" ascii

        $IS\_PE\_FILE="MZ"ascii

    condition:

        $IS\_PE\_FILE at 0  and

        ($string1 and $string2 and $string3 and $string4 and $string5 )

}