Name - Manish Arora   Mis - 111803036   BTech Comp Div1

Malware Reverse Engineering

Malware :

Malware, sometimes known as "malicious software," is a catch-all word for any malicious programme or code that is destructive to computers.
Malware is hostile, intrusive, and purposefully malicious software that aims to infiltrate, damage, or disable computers, computer systems, networks, tablets, and mobile devices by gaining partial control over their activities. It interferes with regular functioning in the same way that the human flu does.

Malware Reverse Engineering :

Disassembling (and, in certain cases, decompiling) a computer programme is part of the reverse engineering process. In this procedure, binary instructions are converted to code mnemonics (or higher-level structures), allowing engineers to study what the programme does and how it affects other systems. Engineers may only build strategies to limit the program's malicious impacts if they understand its complexities. A reverse engineer (sometimes known as a "reverser") employs a number of ways to figure out how a programme works its way through a system and what it's supposed to do. As a result, the reverser would be aware of the vulnerabilities that the software intended to exploit.

Malware Used : CryptoWall RansomWare

Cryptowall is a ransomware virus that encrypts files on a hacked computer using a Trojan horse and demands a ransom in exchange for a decryption key. Cryptowall is usually spread through spam emails, harmful online ads, hacked websites, or other malware. Cryptowall encrypts any files on the drive with particular extensions and leaves files with instructions for paying the ransom and obtaining the decryption key when it is run.

Working of CryptoWall :

CryptoWall spreads through emails with ZIP attachments containing the virus disguised as PDF files. Bills, purchase orders, invoices, and other documents are frequently disguised as PDF files. When victims open the infected PDF files, the CryptoWall virus infects their computer and instals malware files in the AppData or Temp directories. CryptoWall 4.0 is now also available through the Nuclear Exploit

Kit (NEK). Once the computer has been infected, the installer will begin searching the computer's hard drives for data files to encrypt.

Features of CryptoWall RansomWare

1. It does not infect machines in Russia or the former Soviet Union—a strong indicator that the author or authors are based in the region.
2. It follows an affiliate marketing business model, aka Ransomware-as-a-Service (RaaS), in which low-level cybercriminals do the heavy lifting of finding new victims while the threat authors are free to tinker with and improve their creation.
3. It targets consumers and businesses with PCs running Microsoft Windows.

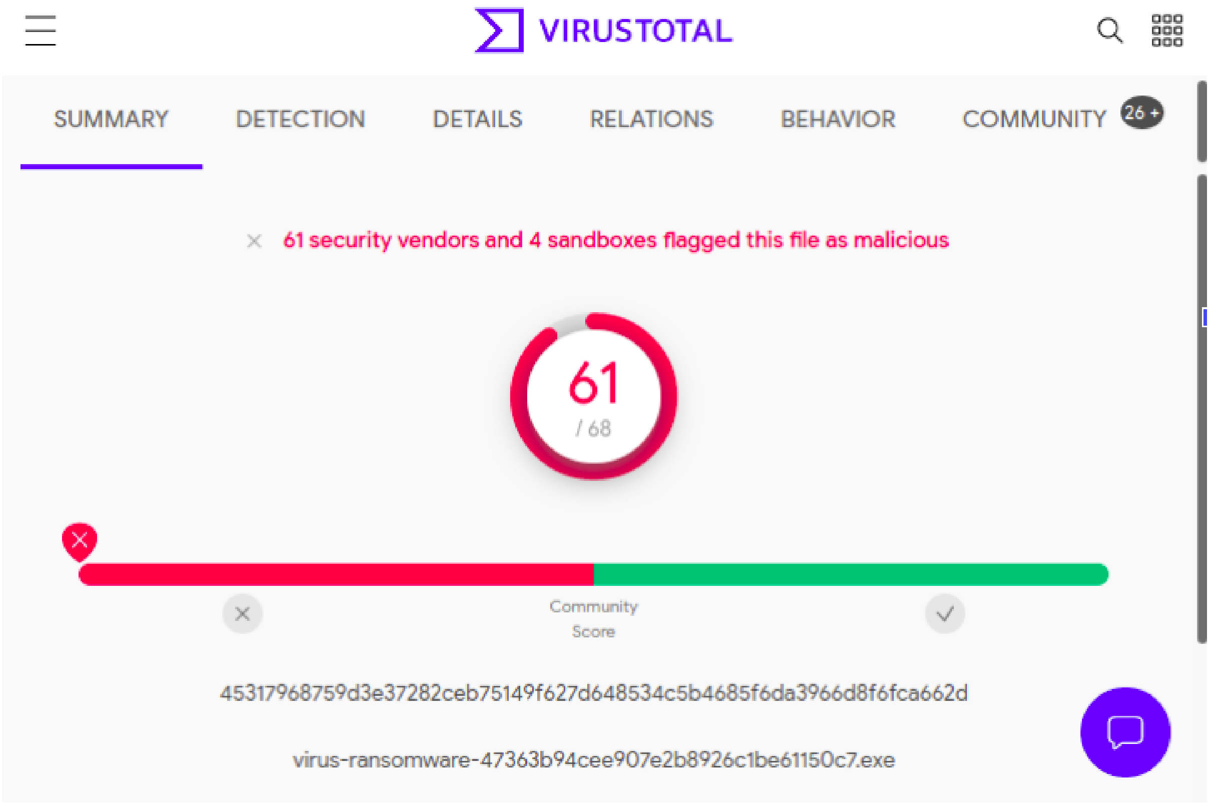
How to Protect our System against Cryptowall Ransomware?

To guard against Cryptowall, you'll need anti-ransomware software that can automatically prevent users from opening attachments that contain harmful code or clicking on links that download malware. Because it's unlikely that you'll be able to completely prevent ransomware and other advanced threats, you'll also need measures to mitigate the damage Cryptowall and other complex attacks might inflict.

Tool Used for Analysis - Radare2

Radare2 (also known as r2) is a complete framework for reverse-engineering and analysing binaries. It is made up of a collection of minor programmes that may be used together or separately from the command line. It is based on a disassembler for computer software that creates assembly language source code from machine-executable code and supports a wide range of executable formats for various processor architectures and operating systems.

File Used is



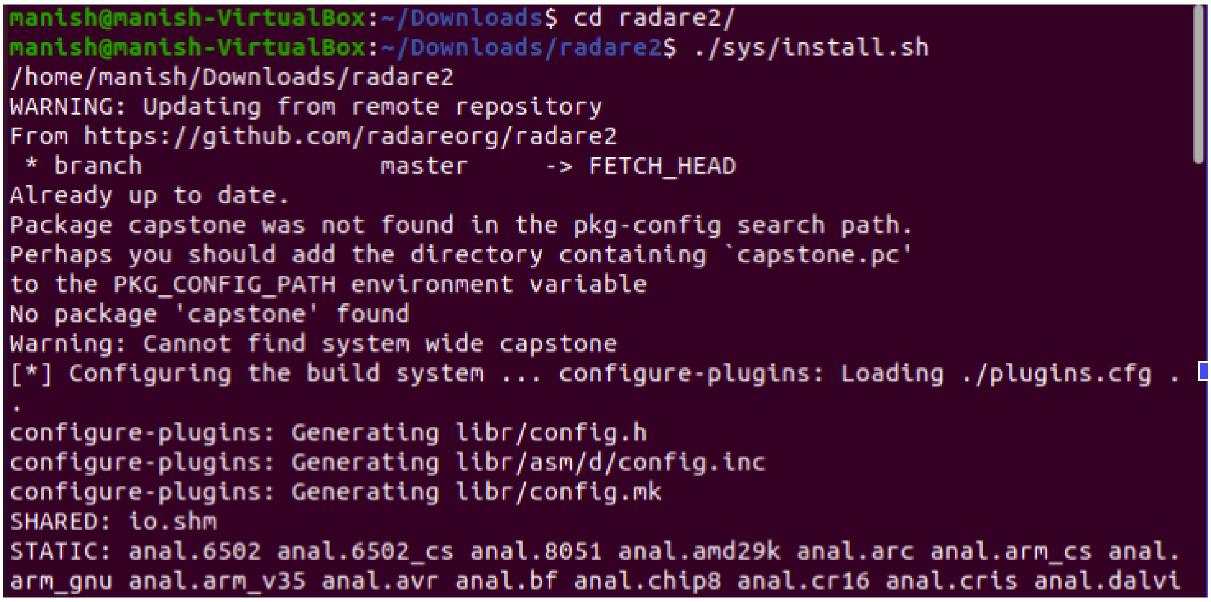
61 security vendors and 4 sandboxes flagged this file as malicious

45317968759d3e37282ceb75149f627d648534c5b4685f6da3966d8f6fca662d

virus-ransomware-47363b94cee907e2b8926c1be61150c7.exe

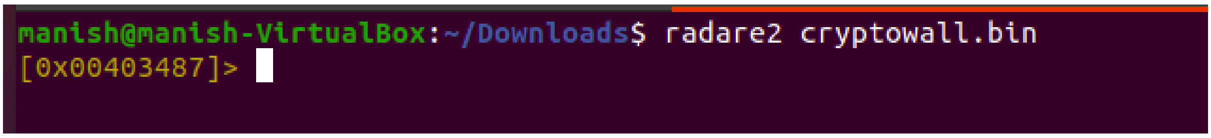
Steps for performing Malware Analysis

1. **Installing the radare2**



```
manish@manish-VirtualBox:~/Downloads$ cd radare2/
manish@manish-VirtualBox:~/Downloads/radare2$ ./sys/install.sh
/home/manish/Downloads/radare2
WARNING: Updating from remote repository
From https://github.com/radareorg/radare2
 * branch               master     -> FETCH_HEAD
Already up to date.
Package capstone was not found in the pkg-config search path.
Perhaps you should add the directory containing `capstone.pc'
to the PKG_CONFIG_PATH environment variable
No package 'capstone' found
Warning: Cannot find system wide capstone
[*] Configuring the build system ... configure-plugins: Loading ./plugins.cfg .
.
configure-plugins: Generating libr/config.h
configure-plugins: Generating libr/asm/d/config.inc
configure-plugins: Generating libr/config.mk
SHARED: io.shm
STATIC: anal.6502 anal.6502_cs anal.8051 anal.amd29k anal.arc anal.arm_cs anal.
arm_gnu anal.arm_v35 anal.avr anal.bf anal.chip8 anal.cr16 anal.cris anal.dalvi
```

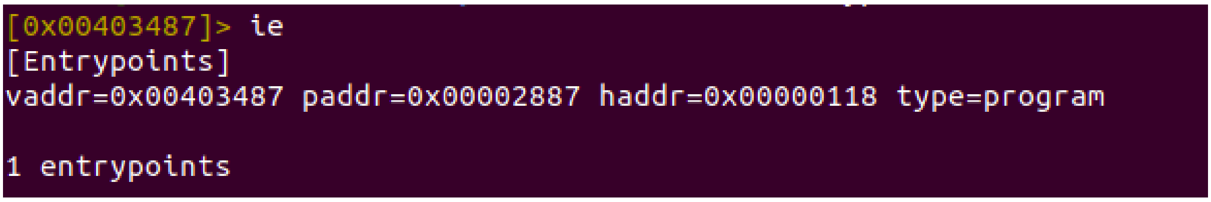
2. Opening the malware file



```
manish@manish-VirtualBox:~/Downloads$ radare2 cryptowall.bin
[0x00403487]> 
```

address 0x004044bb is the entry point for the executable of our ransomware sample. To navigate an executable within radare2, we have to use text-based commands to initiate processing and query information.

Example command → typing **ie** will provide information about the executable’s entry point.



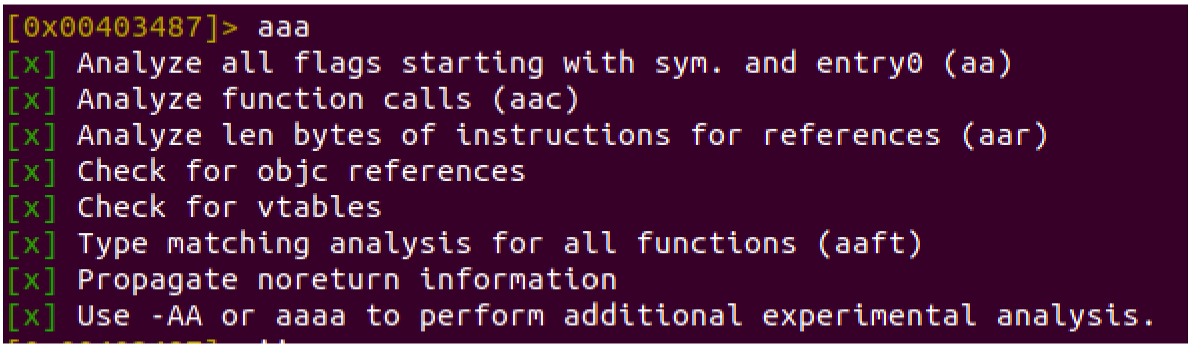
```
[0x00403487]> ie
[Entrypoints]
vaddr=0x00403487 paddr=0x00002887 haddr=0x00000118 type=program

1 entrypoints
```

3. Initiating code analysis

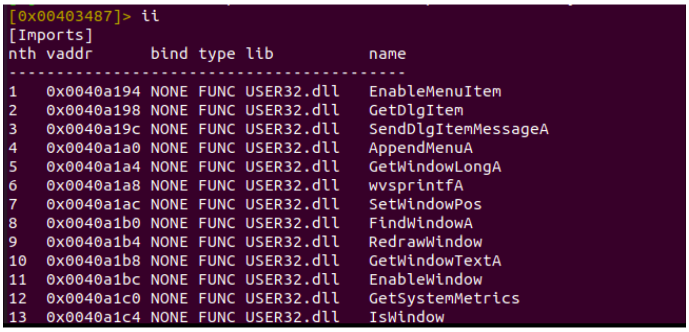
radare2, by default, does not perform any analysis at startup.

Aaa command → shows a variety of output messages as radare2 analyzes the binary



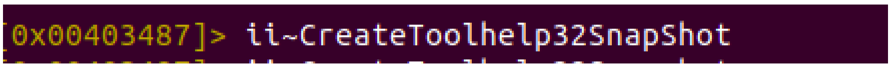
```
[0x00403487]> aaa
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze function calls (aac)
[x] Analyze len bytes of instructions for references (aar)
[x] Check for objc references
[x] Check for vtables
[x] Type matching analysis for all functions (aaft)
[x] Propagate noreturn information
[x] Use -AA or aaaa to perform additional experimental analysis.
```

4. Viewing Imports



```
[0x00403487]> ii
[Imports]
nth vaddr      bind type lib          name
------------------------------------------
1   0x0040a194 NONE FUNC USER32.dll   EnableMenuItem
2   0x0040a198 NONE FUNC USER32.dll   GetDlgItem
3   0x0040a19c NONE FUNC USER32.dll   SendDlgItemMessageA
4   0x0040a1a0 NONE FUNC USER32.dll   AppendMenuA
5   0x0040a1a4 NONE FUNC USER32.dll   GetWindowLongA
6   0x0040a1a8 NONE FUNC USER32.dll   wvsprintfA
7   0x0040a1ac NONE FUNC USER32.dll   SetWindowPos
8   0x0040a1b0 NONE FUNC USER32.dll   FindWindowA
9   0x0040a1b4 NONE FUNC USER32.dll   RedrawWindow
10  0x0040a1b8 NONE FUNC USER32.dll   GetWindowTextA
11  0x0040a1bc NONE FUNC USER32.dll   EnableWindow
12  0x0040a1c0 NONE FUNC USER32.dll   GetSystemMetrics
13  0x0040a1c4 NONE FUNC USER32.dll   IsWindow
```

CreateToolHelp32Snapshot - This API is used to capture a snapshot of running processes on a system

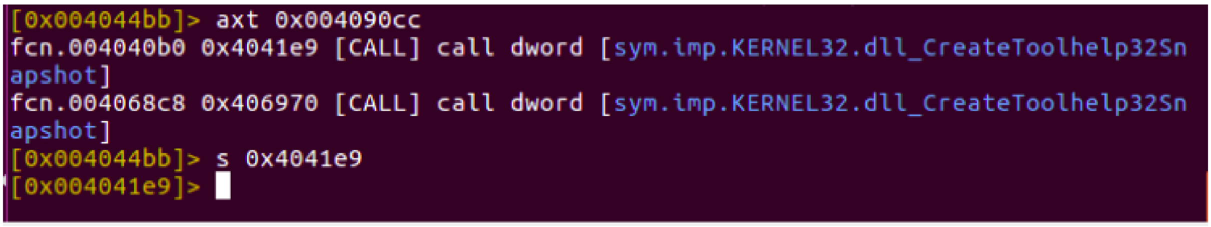


```
[0x00403487]> ii~CreateToolhelp32SnapShot
```

5. **Finding an API reference**

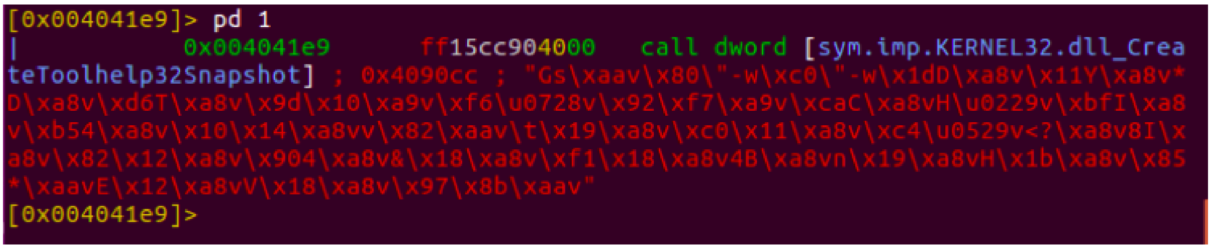
   to locate references to this API, command is axt followed by the address of the imported function

   Where, a - analysis, x - cross references, t - find references to the specified address.



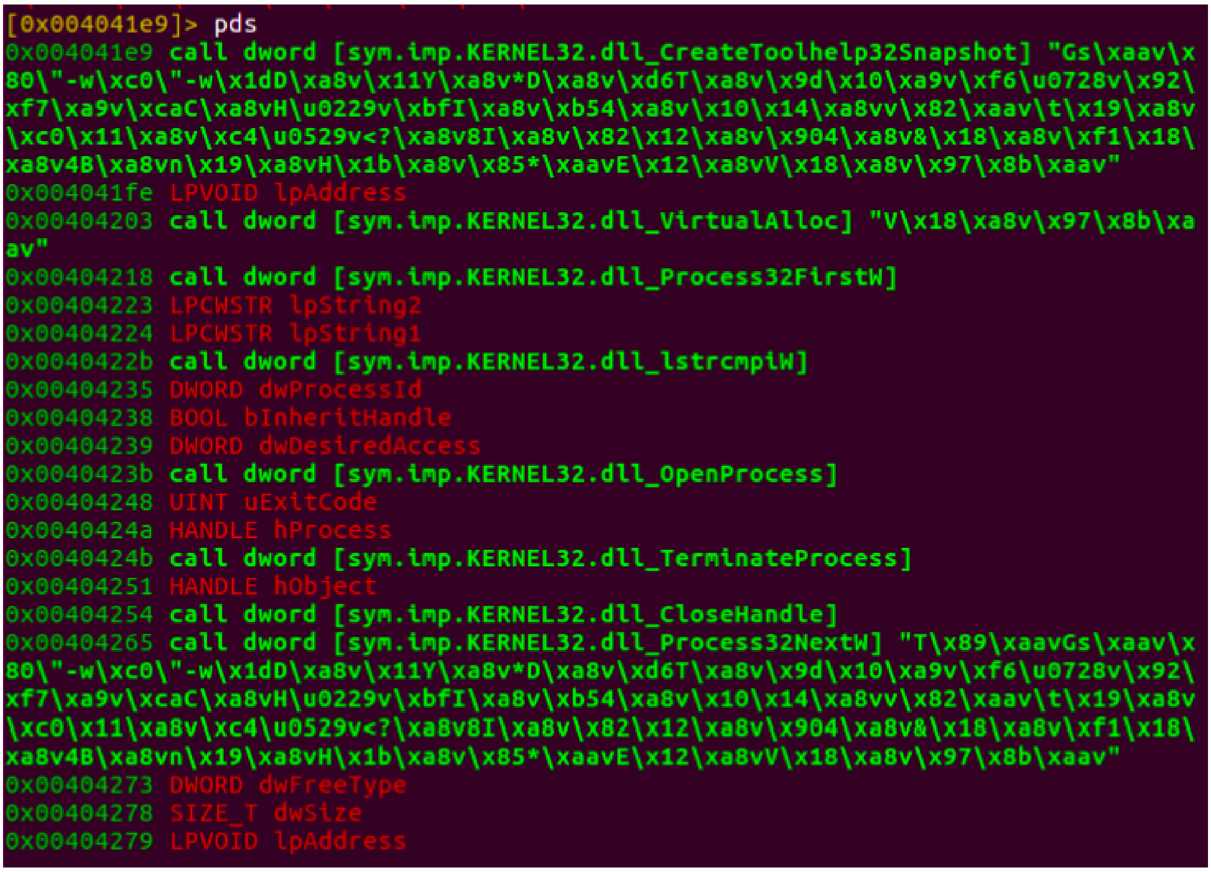
```
[0x004044bb]> axt 0x004090cc
fcn.004040b0 0x4041e9 [CALL] call dword [sym.imp.KERNEL32.dll_CreateToolhelp32Sn
apshot]
fcn.004068c8 0x406970 [CALL] call dword [sym.imp.KERNEL32.dll_CreateToolhelp32Sn
apshot]
[0x004044bb]> s 0x4041e9
[0x004041e9]> █
```

   Here are two CALL instructions, which represent instructions that call CreateToolhelp32Snapshot. To jump to the first reference address, use the s (seek) command. Here, the address in the prompt changed to our destination address, a signal that we have arrived at the desired location. To confirm this, pd(print disassembly) command is used and pd1 will print one disassembly line.



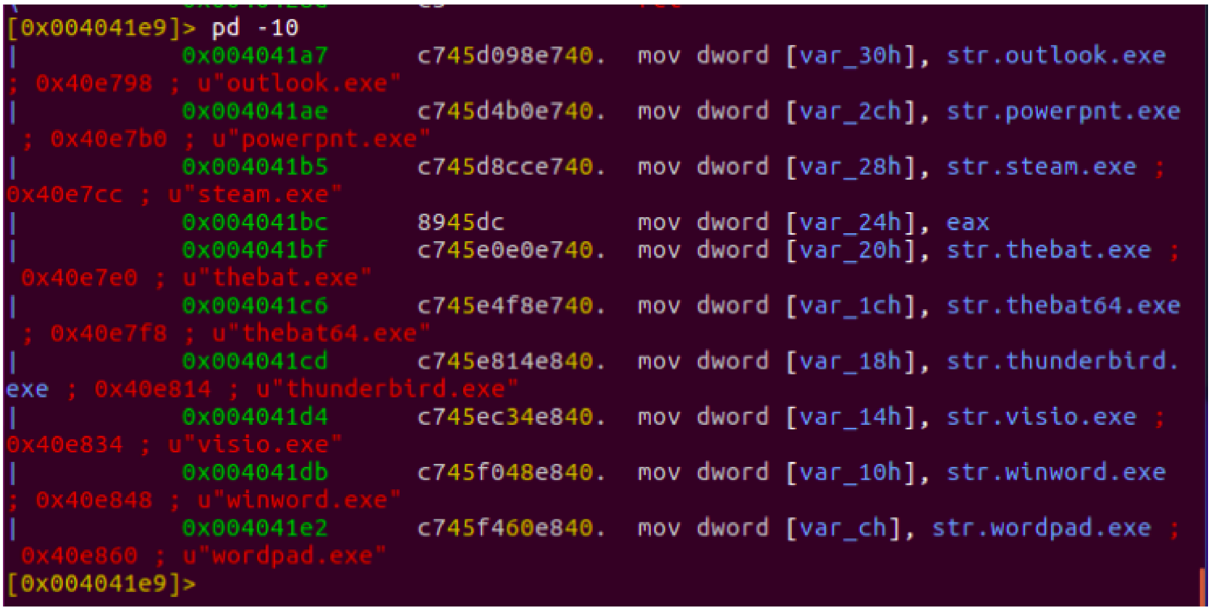
```
[0x004041e9]> pd 1
|           0x004041e9      ff15cc904000   call dword [sym.imp.KERNEL32.dll_Crea
teToolhelp32Snapshot] ; 0x4090cc ; "Gs\xaav\x80\"-w\xc0\"-w\x1dD\xa8v\x11Y\xa8v*
D\xa8v\xd6T\xa8v\x9d\x10\xa9v\xf6\u0728v\x92\xf7\xa9v\xcaC\xa8vH\u0229v\xbfI\xa8
v\xb54\xa8v\x10\x14\xa8vv\x82\xaav\t\x19\xa8v\xc0\x11\xa8v\xc4\u0529v<?\xa8v8I\x
a8v\x82\x12\xa8v\x904\xa8v&\x18\xa8v\xf1\x18\xa8v4B\xa8vn\x19\xa8vH\x1b\xa8v\x85
*\xaavE\x12\xa8vV\x18\xa8v\x97\x8b\xaav"
[0x004041e9]>
```

6. View the summary of the function by pds(print disassembly summary) command. pds focuses on strings, calls, jumps, and references to provide an overview of the function



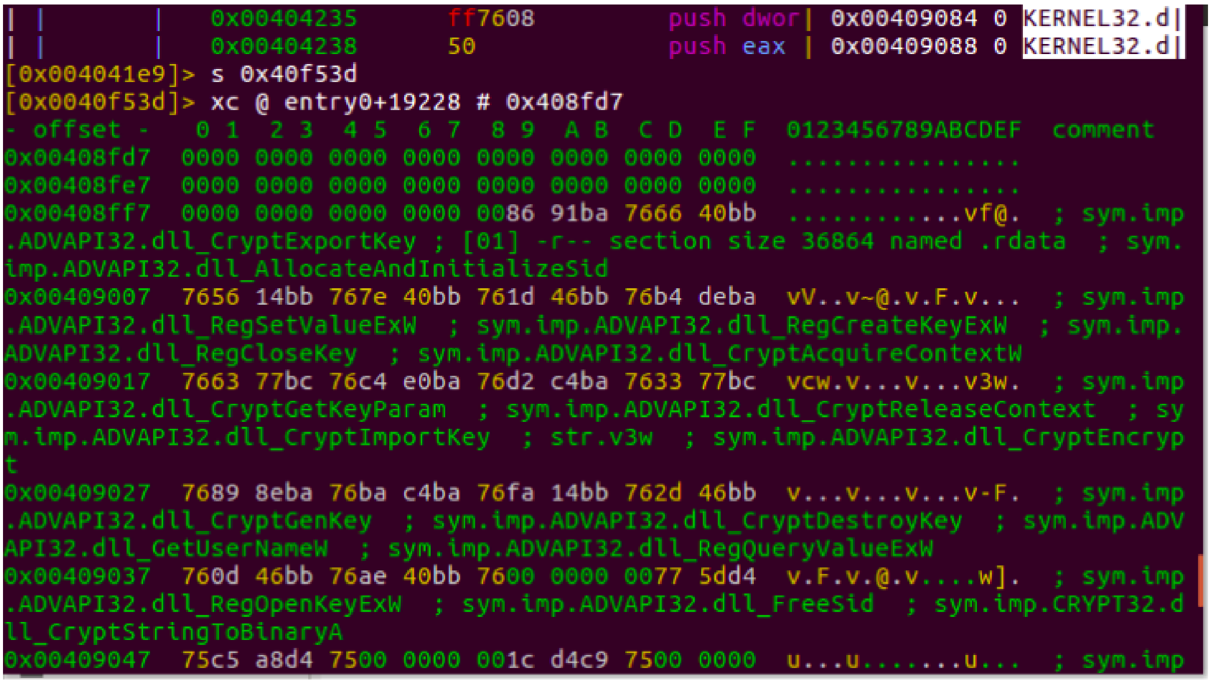
```
[0x004041e9]> pds
0x004041e9 call dword [sym.imp.KERNEL32.dll_CreateToolhelp32Snapshot] "Gs\xaav\x
80\"-w\xc0\"-w\x1dD\xa8v\x11Y\xa8v*D\xa8v\xd6T\xa8v\x9d\x10\xa9v\xf6\u0728v\x92\
xf7\xa9v\xcaC\xa8vH\u0229v\xbfI\xa8v\xb54\xa8v\x10\x14\xa8vv\x82\xaav\t\x19\xa8v
\xc0\x11\xa8v\xc4\u0529v<?\xa8v8I\xa8v\x82\x12\xa8v\x904\xa8v&\x18\xa8v\xf1\x18\
xa8v4B\xa8vn\x19\xa8vH\x1b\xa8v\x85*\xaavE\x12\xa8vV\x18\xa8v\x97\x8b\xaav"
0x004041fe LPVOID lpAddress
0x00404203 call dword [sym.imp.KERNEL32.dll_VirtualAlloc] "V\x18\xa8v\x97\x8b\xa
av"
0x00404218 call dword [sym.imp.KERNEL32.dll_Process32FirstW]
0x00404223 LPCWSTR lpString2
0x00404224 LPCWSTR lpString1
0x0040422b call dword [sym.imp.KERNEL32.dll_lstrcmpiW]
0x00404235 DWORD dwProcessId
0x00404238 BOOL bInheritHandle
0x00404239 DWORD dwDesiredAccess
0x0040423b call dword [sym.imp.KERNEL32.dll_OpenProcess]
0x00404248 UINT uExitCode
0x0040424a HANDLE hProcess
0x0040424b call dword [sym.imp.KERNEL32.dll_TerminateProcess]
0x00404251 HANDLE hObject
0x00404254 call dword [sym.imp.KERNEL32.dll_CloseHandle]
0x00404265 call dword [sym.imp.KERNEL32.dll_Process32NextW] "T\x89\xaavGs\xaav\x
80\"-w\xc0\"-w\x1dD\xa8v\x11Y\xa8v*D\xa8v\xd6T\xa8v\x9d\x10\xa9v\xf6\u0728v\x92\
xf7\xa9v\xcaC\xa8vH\u0229v\xbfI\xa8v\xb54\xa8v\x10\x14\xa8vv\x82\xaav\t\x19\xa8v
\xc0\x11\xa8v\xc4\u0529v<?\xa8v8I\xa8v\x82\x12\xa8v\x904\xa8v&\x18\xa8v\xf1\x18\
xa8v4B\xa8vn\x19\xa8vH\x1b\xa8v\x85*\xaavE\x12\xa8vV\x18\xa8v\x97\x8b\xaav"
0x00404273 DWORD dwFreeType
0x00404278 SIZE_T dwSize
0x00404279 LPVOID lpAddress
```

7. **Analyze the code before calls - pd -10**



```
[0x004041e9]> pd -10
|           0x004041a7      c745d098e740.  mov dword [var_30h], str.outlook.exe
; 0x40e798 ; u"outlook.exe"
|           0x004041ae      c745d4b0e740.  mov dword [var_2ch], str.powerpnt.exe
 ; 0x40e7b0 ; u"powerpnt.exe"
|           0x004041b5      c745d8cce740.  mov dword [var_28h], str.steam.exe ;
0x40e7cc ; u"steam.exe"
|           0x004041bc      8945dc         mov dword [var_24h], eax
|           0x004041bf      c745e0e0e740.  mov dword [var_20h], str.thebat.exe ;
 0x40e7e0 ; u"thebat.exe"
|           0x004041c6      c745e4f8e740.  mov dword [var_1ch], str.thebat64.exe
 ; 0x40e7f8 ; u"thebat64.exe"
|           0x004041cd      c745e814e840.  mov dword [var_18h], str.thunderbird.
exe ; 0x40e814 ; u"thunderbird.exe"
|           0x004041d4      c745ec34e840.  mov dword [var_14h], str.visio.exe ;
0x40e834 ; u"visio.exe"
|           0x004041db      c745f048e840.  mov dword [var_10h], str.winword.exe
; 0x40e848 ; u"winword.exe"
|           0x004041e2      c745f460e840.  mov dword [var_ch], str.wordpad.exe ;
 0x40e860 ; u"wordpad.exe"
[0x004041e9]>
```

8. **Final string matching**



```
| |         |    0x00404235        ff7608          push dwor| 0x00409084 0 KERNEL32.d|
| |         |    0x00404238        50              push eax | 0x00409088 0 KERNEL32.d|
[0x004041e9]> s 0x40f53d
[0x0040f53d]> xc @ entry0+19228 # 0x408fd7
- offset -   0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0123456789ABCDEF  comment
0x00408fd7  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x00408fe7  0000 0000 0000 0000 0000 0000 0000 0000  ................
0x00408ff7  0000 0000 0000 0000 0086 91ba 7666 40bb  ............vf@.  ; sym.imp
.ADVAPI32.dll_CryptExportKey ; [01] -r-- section size 36864 named .rdata  ; sym.
imp.ADVAPI32.dll_AllocateAndInitializeSid
0x00409007  7656 14bb 767e 40bb 761d 46bb 76b4 deba  vV..v~@.v.F.v...  ; sym.imp
.ADVAPI32.dll_RegSetValueExW  ; sym.imp.ADVAPI32.dll_RegCreateKeyExW  ; sym.imp.
ADVAPI32.dll_RegCloseKey  ; sym.imp.ADVAPI32.dll_CryptAcquireContextW
0x00409017  7663 77bc 76c4 e0ba 76d2 c4ba 7633 77bc  vcw.v...v...v3w.  ; sym.imp
.ADVAPI32.dll_CryptGetKeyParam  ; sym.imp.ADVAPI32.dll_CryptReleaseContext  ; sy
m.imp.ADVAPI32.dll_CryptImportKey  ; str.v3w  ; sym.imp.ADVAPI32.dll_CryptEncryp
t
0x00409027  7689 8eba 76ba c4ba 76fa 14bb 762d 46bb  v...v...v...v-F.  ; sym.imp
.ADVAPI32.dll_CryptGenKey  ; sym.imp.ADVAPI32.dll_CryptDestroyKey  ; sym.imp.ADV
API32.dll_GetUserNameW  ; sym.imp.ADVAPI32.dll_RegQueryValueExW
0x00409037  760d 46bb 76ae 40bb 7600 0000 0077 5dd4  v.F.v.@.v....w].  ; sym.imp
.ADVAPI32.dll_RegOpenKeyExW  ; sym.imp.ADVAPI32.dll_FreeSid  ; sym.imp.CRYPT32.d
ll_CryptStringToBinaryA
0x00409047  75c5 a8d4 7500 0000 001c d4c9 7500 0000  u...u.......u...  ; sym.imp
```

How to Detect CryptoWall RansomWare?

1. Watch out for known file extensions
2. Watch out for an increase in file renames
3. Create a sacrificial network share
4. Update your IDS systems with exploit kit detection rules
5. Use client based anti-ransomware agents

Steps to remove CryptoWall

1. Disconnect from the internet
2. Conduct an investigation with your internet security software
3. Use a ransomware decryption tool.
4. Restore your backup