

Forum:[Hack Forums](#)

Thread: [Malware Removal \(Botkiller\)](#)

Malware? - October 31, 2014, 10:16 am

Here is the source to a pretty simple botkiller.

I did NOT code this, I found it on the internet.

Code Class Removal

#Region "Declarations"

Private Declare Function MoveFileEx Lib "kernel32" Alias "MoveFileExA" (ByVal OldFilename As String, ByVal NewFileName As String, ByVal nWord As Long) As Long

Public AppLocal As String =

Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData)

Public Temp As String = Environment.GetEnvironmentVariable("temp")

Public DirStartup As String = Environment.GetFolderPath(Environment.SpecialFolder.Startup)

Public DirStartup2 As String = Environment.GetEnvironmentVariable("SystemDrive") &

"\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"

Public AppData As String =

Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData)

Public Users As String = Environment.GetEnvironmentVariable("userprofile")

#End Region

Private Strings1() As String = {"SetWindowsHookEx", "GetForegroundWindow",
"GetWindowText"}

Private Strings2() As String = {"GetAsyncKeyState", "GetForegroundWindow", "GetWindowText"}

Private Strings3() As String = {"SetThreadContext", "ZwUnmapViewOfSection", "VirtualAllocEx"}

Private Strings4() As String = {"PRIVMSG", "JOIN", "USER", "NICK"}

Private Strings5() As String = {"WSAStartup", "gethostbyname", "gethostbyaddr", "gethostname",
"bs_fusion_bot", "MAP_GETCOUNTRY", "VS_VERSION_INFO", "LookupAccountNameA",
"CryptUnprotectData", "Blackshades NET"}

Private Strings6() As String = {"MD5CryptoServiceProvider", "RijndaelManaged"}

Private Strings7() As String = {"GetExecutingAssembly", "System.Reflection"}

Private Const RemovalStatusHeader = 125

Private ListOfObjects As New List(Of PossibleThreat)

Public Structure PossibleThreat

Dim FullPath As String

Dim Running As Boolean

Dim ID As Integer

Dim Judged As JudgedAs

Dim RegKey As String

Dim ExeName As String

End Structure

Public Enum RemovalStatus

```
AttemptingRemoval = 1
ProcessTerminated = 2
RemovedFile = 4
RegistryKeyRemoved = 5
FailedToRemove = 6
RemovingOnReboot = 7
ThreatRemoved = 8
End Enum
```

```
Public Enum JudgedAs
    Unknown = 17
    Keylogger = 18
    GenericBot = 19
    Injector = 20
    IRC_bot = 21
    NET_Injector = 22
    NET_ExecuteBytes = 23
End Enum
```

```
#Region "Scanning"
```

```
Public Sub ScanThread()
    On Error Resume Next
    Dim ExternalScan As New Thread(AddressOf ScanBots)
    ExternalScan.SetApartmentState(ApartmentState.STA)
    ExternalScan.Start()
    GC.Collect()
End Sub
```

```
Public Sub ScanBots()
    Try
        ListOfObjects.Clear()
        Dim SuspicionList As New List(Of String)
        For Each S As String In
ReturnHKCUStrings("Software\Microsoft\Windows\CurrentVersion\Run")
            SuspicionList.Add(S)
        Next
        For Each S As String In
ReturnHKLMStrings("Software\Microsoft\Windows\CurrentVersion\Run")
            SuspicionList.Add(S)
        Next
        For Each S As String In
ReturnHKCUStrings("Software\Microsoft\Windows\CurrentVersion\RunOnce")
            SuspicionList.Add(S)
        Next
        For Each S As String In
ReturnHKLMStrings("Software\Microsoft\Windows\CurrentVersion\RunOnce")
            SuspicionList.Add(S)
        Next
        For Each S As String In
ReturnDirStrings(Environment.GetFolderPath(Environment.SpecialFolder.Startup))
            SuspicionList.Add(S)
        Next
        For Each S As String In
```

```

ReturnDirStrs(Environment.GetEnvironmentVariable("SystemDrive") &
"\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup")
    SuspicionList.Add(S)
Next
For Each FilePath As String In SuspicionList
    Try
        If IsFilePathAlright(FilePath.Split(MainModule.SplitMark1)(0)) Then
            ListOfObjects.Add(ScanFile(FilePath))
        Catch : End Try
    Next
    Dim SendingString As String = Nothing
    For x = 0 To ListOfObjects.Count - 1
        If Not ListOfObjects(x).FullPath = Nothing And Not ListOfObjects(x).Judged = Nothing
Then
            SendingString &= ListOfObjects(x).ID & MainModule.SplitMark1
            SendingString &= ListOfObjects(x).FullPath & MainModule.SplitMark1
            SendingString &= ListOfObjects(x).RegKey & MainModule.SplitMark1

            If ListOfObjects(x).Running Then
                SendingString &= Chr(2) & MainModule.SplitMark1
            Else
                SendingString &= Chr(1) & MainModule.SplitMark1
            End If
            SendingString &= Chr(x + 8) & MainModule.SplitMark1
            SendingString &= Chr(ListOfObjects(x).Judged) & MainModule.splitMark2
        End If
    Next
    If Not SendingString = Nothing Then
        MainModule.Send(Chr(MainModule.CommandType.MalwareRemoval) & SendingString)
    End If
Catch ex As Exception
End Try
End Sub

```

```
Public Function IsFilePathAlright(ByVal path As String) As Boolean
    If path.Contains(Users) Then Return True
    Return False
End Function
```

```
Public Function ScanFile(ByVal path As String) As PossibleThreat
    Try
        If IO.File.Exists(path.Split(MainModule.SplitMark1)(0)) Then
            Dim Info As New PossibleThreat
            Info.FullPath = path.Split(MainModule.SplitMark1)(0)
            Info.RegKey = path.Split(MainModule.SplitMark1)(1)
            Info.Running = IsProcessRunning(path)
            Info.ExeName = System.IO.Path.GetFileName(Info.FullPath)
            Dim TempString As String =
                System.Text.Encoding.UTF8.GetString(IO.File.ReadAllBytes(Info.FullPath)).Trim(Chr(0))
            If Not TempString = Nothing Then
                If TempString.Contains(Strings5(0)) Or TempString.Contains(Strings5(1)) Or
                TempString.Contains(Strings5(2)) Or TempString.Contains(Strings5(3)) Then
                    Info.Judged = JudgedAs.GenericBot
                End If
            End If
        End If
    Catch
    End Try
End Function
```

```

        ElseIf TempString.Contains(Strings1(0)) Or TempString.Contains(Strings1(1)) Or
TempString.Contains(Strings1(2)) Then
            Info.Judged = JudgedAs.Keylogger
        ElseIf TempString.Contains(Strings2(0)) Or TempString.Contains(Strings2(1)) Or
TempString.Contains(Strings2(2)) Then
            Info.Judged = JudgedAs.Keylogger
        ElseIf TempString.Contains(Strings3(0)) Or TempString.Contains(Strings3(1)) Or
TempString.Contains(Strings3(2)) Then
            Info.Judged = JudgedAs.Injector
        ElseIf TempString.Contains(Strings4(0)) Or (TempString.Contains(Strings4(1)) And
TempString.Contains(Strings4(2)) And TempString.Contains(Strings4(3))) Then
            Info.Judged = JudgedAs.IRC_bot
        ElseIf TempString.Contains(Strings6(0)) Or TempString.Contains(Strings6(1)) Then
            Info.Judged = JudgedAs.NET_Injector
        ElseIf TempString.Contains(Strings7(0)) Or TempString.Contains(Strings7(1)) Then
            Info.Judged = JudgedAs.NET_ExecuteBytes
        Else
            Info.Judged = JudgedAs.Unknown
        End If
        Return Info
    Else
        Return Nothing
    End If
End If
Return Nothing
Catch ex As Exception
    Return Nothing
End Try

```

End Function

```

Public Function ReturnHKCUStrs(ByVal key As String) As List(Of String)
    On Error Resume Next
    Dim StringsToRet As New List(Of String)
    For Each R1 In My.Computer.Registry.CurrentUser.OpenSubKey(key, False).GetValueNames
        Dim rVal As String = My.Computer.Registry.CurrentUser.OpenSubKey(key,
False).GetValue(R1)
        If rVal.Contains(Chr(34)) Then
            rVal = rVal.Split(Chr(34))(1)
        End If
        If rVal.Contains(" -") Then
            rVal = Split(rVal, " -")(0)
        End If
        If rVal.Contains(" /") Then
            rVal = Split(rVal, " /")(0)
        End If
        If rVal.Contains(".exe") Or rVal.Contains(".dll") Or rVal.Contains(".bat") Or
rVal.Contains(".vbs") Or rVal.Contains(".scr") Then StringsToRet.Add(rVal & MainModule.SplitMark1
& "HKEY_CURRENT_USER\" & key & "\" & R1)
    Next
    Return StringsToRet
End Function

```

```

Public Function ReturnHKLMStrs(ByVal key As String) As List(Of String)
    On Error Resume Next
    Dim StringsToRet As New List(Of String)
    For Each R1 In My.Computer.Registry.LocalMachine.OpenSubKey(key,
False).GetValueNames
        Dim rVal As String = My.Computer.Registry.LocalMachine.OpenSubKey(key,
False).GetValue(R1)
        If rVal.Contains(Chr(34)) Then
            rVal = rVal.Split(Chr(34))(1)
        End If
        If rVal.Contains(" -") Then
            rVal = Split(rVal, " -")(0)
        End If
        If rVal.Contains(" /") Then
            rVal = Split(rVal, " /")(0)
        End If
        If rVal.Contains(".exe") Or rVal.Contains(".dll") Or rVal.Contains(".bat") Or
rVal.Contains(".vbs") Or rVal.Contains(".scr") Then StringsToRet.Add(rVal & MainModule.SplitMark1
& "HKEY_LOCAL_MACHINE\ " & key & "\ " & R1)
    Next
    Return StringsToRet
End Function

```

```

Public Function ReturnDirStrs(ByVal path As String) As List(Of String)
    On Error Resume Next
    Dim StringsToRet As New List(Of String)
    For Each f As IO.FileInfo In New IO.DirectoryInfo(path).GetFiles
        If f.FullName.Contains(".exe") Or f.FullName.Contains(".dll") Or f.FullName.Contains(".bat")
Or f.FullName.Contains(".vbs") Or f.FullName.Contains(".scr") Then StringsToRet.Add(f.FullName &
MainModule.SplitMark1 & f.FullName)
    Next
    Return StringsToRet
End Function
#End Region

```

#Region "Processes"

```

Public Function IsProcessRunning(ByVal fullpath As String) As Boolean

    For Each p As Process In Process.GetProcesses
        Try
            If p.MainModule.FileName = fullpath Then
                Return True
            End If
        Catch
        End Try
    Next
    Return False
End Function
#End Region

```

#Region "Actions"

```

Public Sub StartRemoval(ByVal RemovalData As Object)
    Try

```

```

Dim RemoveReboot As Boolean = False
If RemovalData.Split(SplitMark1)(0) = Chr(2) Then
    RemoveReboot = True
Else
    RemoveReboot = False
End If
For x = 1 To RemovalData.Split(SplitMark1).Length - 1
    RemoveThreat(Asc(RemovalData.Split(SplitMark1)(x)) - 8, RemoveReboot)
Next

```

```

Catch ex As Exception

```

```

End Try
End Sub

```

```

Public Function RemoveThreat(ByVal ListId As Integer, ByVal DelOnReboot As Boolean) As
Boolean
    Try

```

```

        For Each p As Process In Process.GetProcesses
            Try
                If p.MainModule.FileName = ListOfObjects(ListId).FullPath Then
                    p.Kill()
                    Send(Chr(RemovalStatusHeader) & Chr(ListId + 8) &
Chr(RemovalStatus.ProcessTerminated))
                    Thread.Sleep(1000)
                End If
            Catch : End Try
        Next
        Try
            File.Delete(ListOfObjects(ListId).FullPath)
            Send(Chr(RemovalStatusHeader) & Chr(ListId + 8) & Chr(RemovalStatus.RemovedFile))
        Catch : End Try
        Thread.Sleep(1000)
        If Not ListOfObjects(ListId).RegKey = Nothing Then
            Try
                Select Case ListOfObjects(ListId).RegKey.Split("\")(0)
                    Case "HKEY_CURRENT_USER"

                        My.Computer.Registry.CurrentUser.DeleteValue(ListOfObjects(ListId).RegKey.Re
move(0, ListOfObjects(ListId).RegKey.IndexOf("\", StringComparison.Ordinal) + 1), False)
                    Case "HKEY_LOCAL_MACHINE"

                        My.Computer.Registry.CurrentUser.DeleteValue(ListOfObjects(ListId).RegKey.Re
move(0, ListOfObjects(ListId).RegKey.IndexOf("\", StringComparison.Ordinal) + 1), False)
                End Select
                Send(Chr(RemovalStatusHeader) & Chr(ListId + 8) &
Chr(RemovalStatus.RegistryKeyRemoved))
            Catch

        End Try
        End If
        Thread.Sleep(1000)

```

```

Try
    If File.Exists(ListOfObjects(ListId).FullPath) Then
        If SystemInformation.Privileges = SystemInformation.UserPrivileges.Admin And
DelOnReboot Then
            Dim RemoveRebootSuccess As Integer =
DeleteFileOnReboot(ListOfObjects(ListId).FullPath)
            If RemoveRebootSuccess Then
                Send(Chr(RemovalStatusHeader) & Chr(ListId + 8) &
Chr(RemovalStatus.RemovingOnReboot))
                Return True
            Else
                Send(Chr(RemovalStatusHeader) & Chr(ListId + 8) &
Chr(RemovalStatus.FailedToRemove))
                Return False
            End If
        End If
        Return False
    End If
    Send(Chr(RemovalStatusHeader) & Chr(ListId + 8) &
Chr(RemovalStatus.ThreatRemoved))
    Return True
Catch
    If File.Exists(ListOfObjects(ListId).FullPath) Then
        If SystemInformation.Privileges = SystemInformation.UserPrivileges.Admin And
DelOnReboot Then
            Dim RemoveRebootSuccess As Integer =
DeleteFileOnReboot(ListOfObjects(ListId).FullPath)
            If RemoveRebootSuccess Then
                Send(Chr(RemovalStatusHeader) & Chr(ListId + 8) &
Chr(RemovalStatus.RemovingOnReboot))
                Return True
            Else
                Send(Chr(RemovalStatusHeader) & Chr(ListId + 8) &
Chr(RemovalStatus.RemovingOnReboot))
                Return True
            End If
        End If
        Return False
    End If
End Try
Catch ex As Exception
    Return False
End Try
End Function

```

```

Public Function DeleteFileOnReboot(ByVal FilePath As String) As Boolean
Try
    Return MoveFileEx(FilePath, 0, 4)
Catch
    Return False
End Try
End Function
#End Region

```

End Class

AngularJS - October 31, 2014, 10:42 am

May I know where you got this from?

Malware? - October 31, 2014, 11:32 am

(Oct 31, 2014 10:42 AM)XAnacondaX Wrote: [\[CITE\]](#) May I know where you got this from?

I do not remember, it was in my folder of downloaded sources. Was going through it and decided to release it because I didn't think it was on HF.

iMeetYourMum - October 31, 2014, 3:39 pm

From IM 2.0, lel. (MainModule.SplitMark1)

Malware? - October 31, 2014, 4:05 pm

(Oct 31, 2014 03:39 PM)iMeetYourMum Wrote: [\[CITE\]](#) From IM 2.0, lel. (MainModule.SplitMark1)

Looks like this is the source? Where did you find this

Aristocles - October 31, 2014, 4:31 pm

So both iMeetYourMum and Venomous Net has decompiled IM2 ?

Malware? - October 31, 2014, 4:40 pm

(Oct 31, 2014 04:31 PM)N?etzsche Wrote: [\[CITE\]](#) So both iMeetYourMum and Venomous Net has decompiled IM2 ?

I have the full source I guess lol. V2 though. Pretty well written functions.

SIM-1676753 - October 31, 2014, 5:08 pm

(Oct 31, 2014 04:31 PM)N?etzsche Wrote: [\[CITE\]](#) So both iMeetYourMum and Venomous Net has decompiled IM2 ?

Decompiled stub is out there. Someone re'd it and got it.

Malware? - October 31, 2014, 6:26 pm

(Oct 31, 2014 05:08 PM)Mr. Trvp Wrote: [\[CITE\]](#) Decompiled stub is out there. Someone re'd it and got it.

I thought Shockwave compiled it on a remote server. The variables and everything are named good, not random. It makes me think that it was compiled locally or someone spent a good amount of time on this.

dispirited - October 31, 2014, 9:38 pm

Thanks man for this awesome source

So do you offer any tutorials for Reverse Engineering

IYS-Security - November 1, 2014, 7:47 am

is anyone tested this anti bot ? and how we can ON it ? put it in timer or ?

DragonHunter - November 1, 2014, 8:35 am

Lol people hyping about some old version of IM2 while I'm having nearly the newest version of IM3 source, take that Reverse Engineers kapouw :D

?eumonic - November 1, 2014, 11:26 am

(Nov 1, 2014 08:35 AM)DragonHunter Wrote: [\[CITE\]](#) Lol people hyping about some old version of IM2 while I'm having nearly the newest version of IM3 source, take that Reverse Engineers kapouw :D

Might I ask how this is relevant to the actual "release" of this source? Or are you just trying to show off?

SIM-1456245 - November 1, 2014, 4:41 pm

(Nov 1, 2014 08:35 AM)DragonHunter Wrote: [\[CITE\]](#) Lol people hyping about some old version of IM2 while I'm having nearly the newest version of IM3 source, take that Reverse Engineers kapouw :D

I'm pretty sure that you are just trying to piss of shock...

Aristocles - November 1, 2014, 5:31 pm

(Nov 1, 2014 04:51 PM)Shockwave™ Wrote: [\[CITE\]](#) Well that's basically impossibru, at least a clean source :P

Appfuscator can be deob with 1 de4dot regex and some CodeCracker's tool.

And Cleaning a deobed assembly by exporting it in a new brand project with Reflector and then clean it by hand is just a matter of time.

Your reply sounds selfish btw ?

Malware? - November 1, 2014, 5:36 pm

(Nov 1, 2014 05:31 PM)N?etzsche Wrote: [\[CITE\]](#) Appfuscator can be deob with 1 de4dot regex and some CodeCracker's tool.

And Cleaning a deobed assembly by exporting it in a new brand project with Reflector and then clean it by hand is just a matter of time.

Your reply sounds selfish btw ?

Cleaning an assembly with like 5000+ lines would not be fun

Aristocles - November 1, 2014, 5:37 pm

(Nov 1, 2014 05:36 PM)Venomous Net Wrote: [\[CITE\]](#) Cleaning an assembly with like 5000+ lines would not be fun

Yep, not that cool nor easy, bot can be done. Not «impossibru» at all as Shockwave said

Juggernauten - November 1, 2014, 5:45 pm

isnt this VERY overkill?

"WSAStartup", "gethostbyname", "gethostbyaddr", "gethostname"

So killing all applications that use Winsock?

Aristocles - November 1, 2014, 6:04 pm

(Nov 1, 2014 05:42 PM)Shockwave™ Wrote: [\[CITE\]](#) I am not using appfuscator

Yes you are dude, yes you are ?

SIM-1676753 - November 1, 2014, 6:11 pm

(Nov 1, 2014 06:04 PM)N?etzsche Wrote: [\[CITE\]](#) Yes you are dude, yes you are ?

Perhaps a paid version with the stubs being obfuscated and all. Who the fuck cares anyway?

Aristocles - November 1, 2014, 6:24 pm

(Nov 1, 2014 06:11 PM)Mr. Trvp Wrote: [\[CITE\]](#) Perhaps a paid version with the stubs being obfuscated and all. Who the fuck cares anyway?

Stubs are "online" he isnt dumb ^^ but yeah, who cares

Sho?kwave - November 1, 2014, 7:33 pm

(Nov 1, 2014 06:04 PM)N?etzsche Wrote: [\[CITE\]](#) Yes you are dude, yes you are ?

The main application uses the following: Appfuscator for the decompression shell -> decompression shell -> main application obfuscated with something else.

The Client uses the following: Custom obfuscator for the decompression shell -> decompression shell -> client application obfuscated with something else

So no, at the core where you would actually find the original source it isn't using appfuscator, don't tell me about my own application, that's extremely idiotic.

On-Topic, I really don't care if OP keeps this up, it's ancient code from a very old version of IM, so I don't really mind.

SIM-1456245 - November 2, 2014, 12:11 pm

(Nov 1, 2014 07:33 PM)Shockwave™ Wrote: [\[CITE\]](#) The main application uses the following: Appfuscator for the decompression shell -> decompression shell -> main application obfuscated with something else.

The Client uses the following: Custom obfuscator for the decompression shell -> decompression shell -> client application obfuscated with something else

So no, at the core where you would actually find the original source it isn't using appfuscator, don't tell me about my own application, that's extremely idiotic.

On-Topic, I really don't care if OP keeps this up, it's ancient code from a very old version of IM, so I don't really mind.

Yep, that version was dammn bad, I mean BAD, compared to the newer ones.

TheExclusive - November 6, 2014, 5:01 am

Nice to see how it scan's on strings inside the file.

nuk - November 6, 2014, 5:17 am

This is actually pretty cool... thanks for sharing this. Nice seeing something new on HF :)