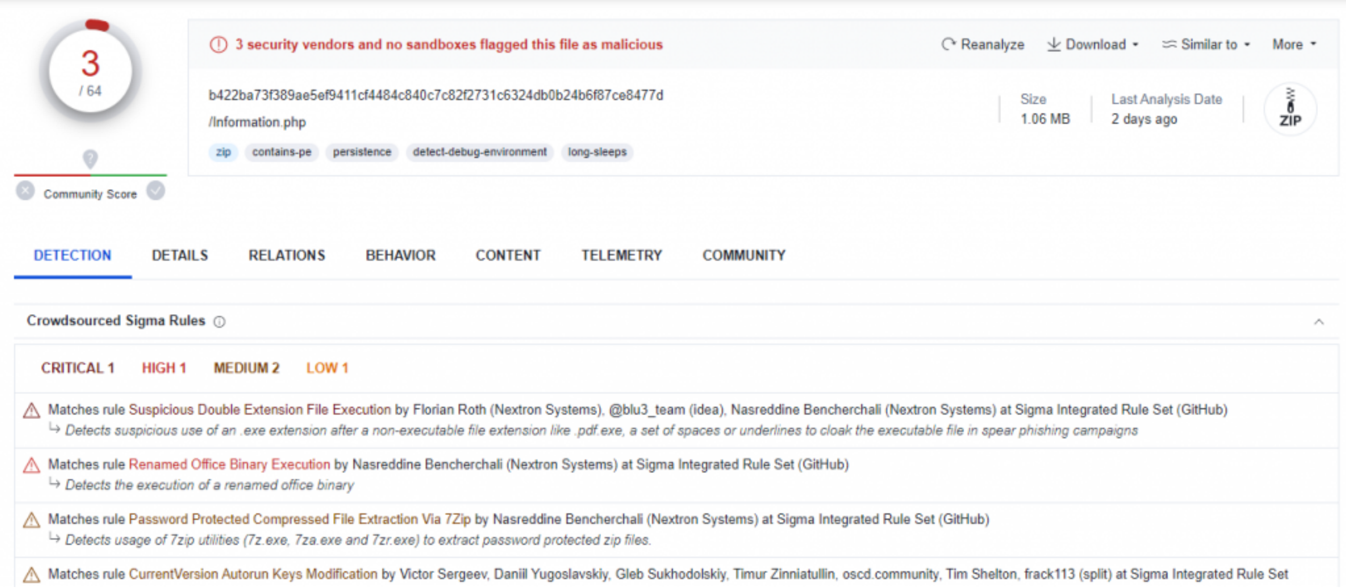
New tricks of APT29 – update on the CERT.PL report

lab52.io/blog/2162-2/

A new sample has been uploaded to VirusTotal, and its characteristics strongly resemble QUARTERRIG, a malware recently analyzed by CERT.PL and linked to APT29. In said analysis, the campaign was named “**Note**“. Based on the aforementioned report, the purpose of this post is to show the new features of this new campaign that we named “**Information**“.

The hash of the sample made public in VirusTotal is b422ba73f389ae5ef9411cf4484c840c7c82f2731c6324db0b24b6f87ce8477d, and only 3 antivirus engines target the sample as malicious during the writing of this post.



3 security vendors and no sandboxes flagged this file as malicious

b422ba73f389ae5ef9411cf4484c840c7c82f2731c6324db0b24b6f87ce8477d
/Information.php

zip contains-pe persistence detect-debug-environment long-sleeps

Size 1.06 MB
Last Analysis Date 2 days ago

Crowdsourced Sigma Rules

CRITICAL 1 HIGH 1 MEDIUM 2 LOW 1

⚠ Matches rule Suspicious Double Extension File Execution by Florian Roth (Nextron Systems), @blu3_team (idea), Nasreddine Bencherchali (Nextron Systems) at Sigma Integrated Rule Set (GitHub)
↳ *Detects suspicious use of an .exe extension after a non-executable file extension like .pdf.exe, a set of spaces or underlines to cloak the executable file in spear phishing campaigns*

⚠ Matches rule Renamed Office Binary Execution by Nasreddine Bencherchali (Nextron Systems) at Sigma Integrated Rule Set (GitHub)
↳ *Detects the execution of a renamed office binary*

⚠ Matches rule Password Protected Compressed File Extraction Via 7Zip by Nasreddine Bencherchali (Nextron Systems) at Sigma Integrated Rule Set (GitHub)
↳ *Detects usage of 7zip utilities (7z.exe, 7za.exe and 7zr.exe) to extract password protected zip files.*

⚠ Matches rule CurrentVersion Autorun Keys Modification by Victor Sergeev, Daniil Yugoslavskiy, Gleb Sukhodolskiy, Timur Zinniatullin, oscd.community, Tim Shelton, frack113 (split) at Sigma Integrated Rule Set

Sample in VirusTotal analysed in this report

APT29 is a hacker group allegedly affiliated with **one or more Russian intelligence agencies**. It is a sophisticated group that has been carrying out attacks against European governments and diplomatic agencies since 2008.

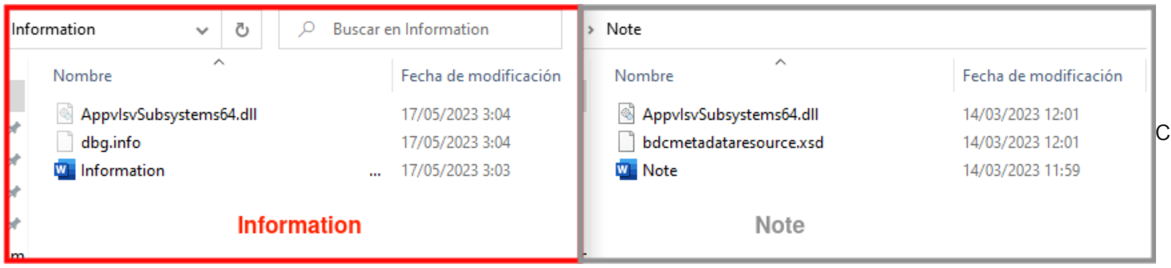
The main entry vector for APT29 is email. Using this input vector, attackers attach a PDF with a link that will download an ISO.

New campaign: “Information”

This new campaign, which will be referred to as **Information**, contains a structure very similar to the **Note** campaigns shown in the CERT.PL report. The samples analyzed in that report are from March. However, from Lab52, **we have observed a change in the operation of this type of malware since April, and in the latest analyzed samples, the injection method has varied**. In this post, we use one of the latest samples to highlight the new changes in the mechanisms employed.

This time **the file containing the shellcode is located in a file called “dbg.info”** unlike what we have been observing in previous campaings. The **Information.iso** contains:

- AppvlsSubsystems64.dll – DLL used to load a legitime system DLL and inject the shellcode into it.
- dbg.info – shellcode.
- Information .exe – Legitime binary **signed by Microsoft**. This will be use to load AppvlsSubsystems64.dll (by DLL Side-Load).

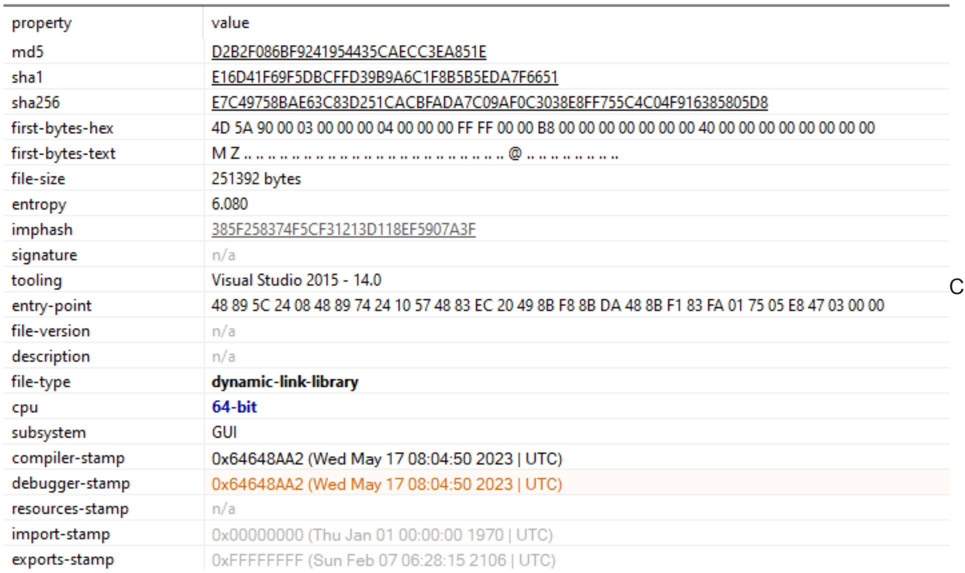


Nombre	Fecha de modificación
AppvlsvSubsystems64.dll	17/05/2023 3:04
dbg.info	17/05/2023 3:04
Information	17/05/2023 3:03

Nombre	Fecha de modificación
AppvlsvSubsystems64.dll	14/03/2023 12:01
bdcmetadataresource.xsd	14/03/2023 12:01
Note	14/03/2023 11:59

Contents of “Information.iso” versus “Note.iso” (past campaign)

Thecompilation date for **AppvlsvSubsystems64.dll** in this new campaign is more recent than the previous one. This could suggest that changes were made to improve the sample.



property	value
md5	D2B2F086BF9241954435CAECC3EA851E
sha1	E16D41F69F5DBCFFD39B9A6C1F8B5B5EDA7F6651
sha256	E7C49758BAE63C83D251CACBFADA7C09AF0C3038E8FF755C4C04F916385805D8
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	251392 bytes
entropy	6.080
imphash	385F258374F5CF31213D118EF5907A3F
signature	n/a
tooling	Visual Studio 2015 - 14.0
entry-point	48 89 5C 24 08 48 89 74 24 10 57 48 83 EC 20 49 8B F8 8B DA 48 8B F1 83 FA 01 75 05 E8 47 03 00 00
file-version	n/a
description	n/a
file-type	**dynamic-link-library**
cpu	**64-bit**
subsystem	GUI
compiler-stamp	0x64648AA2 (Wed May 17 08:04:50 2023 | UTC)
debugger-stamp	0x64648AA2 (Wed May 17 08:04:50 2023 | UTC)
resources-stamp	n/a
import-stamp	0x00000000 (Thu Jan 01 00:00:00 1970 | UTC)
exports-stamp	0xFFFFFFFF (Sun Feb 07 06:28:15 2106 | UTC)

Compilation date

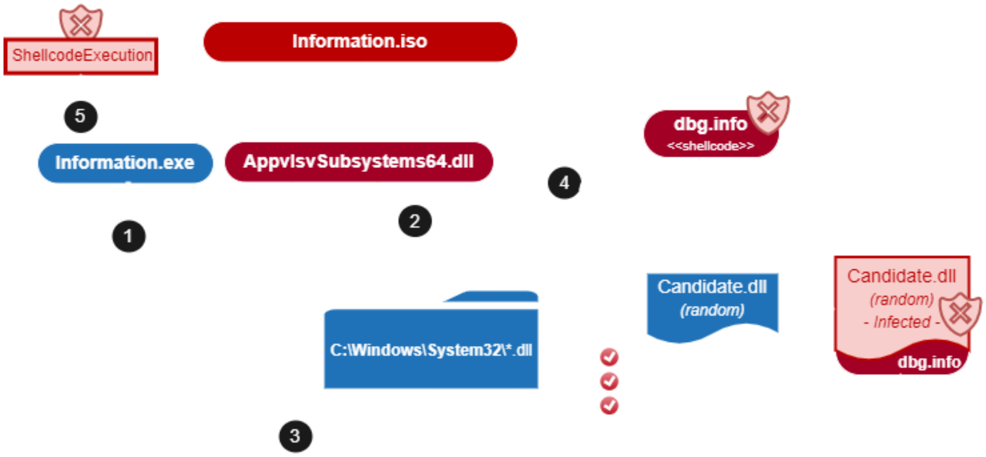
AppvIsvSubsystems64.dll – Information

This post focuses on the main diferences between the previous campaing and the new one, in order to contribute to the community. The most noticeable change is the injection technique. Nevertheless some addional notes are added for the curious.

The objective is the same: the executable (Information.exe in this case) will be used to execute two DLLs. The first DLL is AppvIsvSubsystems64.dll, that will be loaded by the process as part of its execution. The second DLL will be loaded by AppvIsvSubsystem64.dll. **In this case, however, the second DLL will be carefully modified with different techniques in order to minimize the detection methods.**

Therefore, Information.exe, that is a legitime binary, will be the container for the malware to be executed. In order to do that, it will load AppvIsvSubsystems64.dll who needs to identify a suitable system DLL to be modified – with the shellcode – before the load in Information.exe. **The main changes are located in AppvIsvSubsystems64.dll**. Also, the command and control (C2) varies.

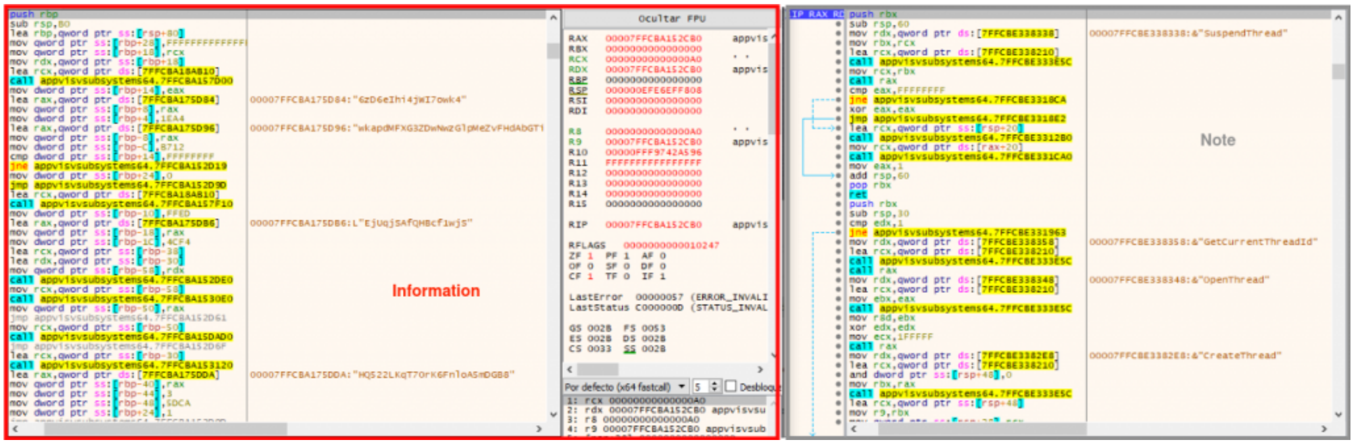
The following diagram describes the infection chain analysed in this post.



Infection chain – Information campaign

Description of the APIs

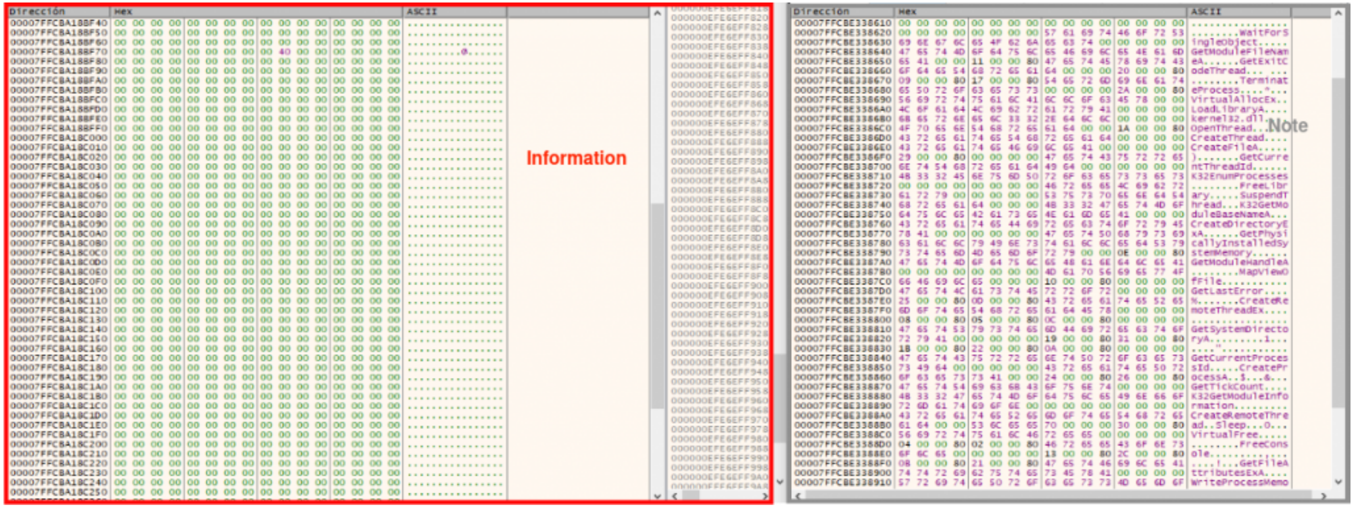
When running the sample, the first difference lies in how the **AppvIsvSubsystems64.dll** loads the functions needed for the execution. In the “Note” campaing, the funcions are loaded at an early stage than the Information campaign. Moreover, the decription in this last case occurs at a different point, later in during the execution.



```
push rbp
sub rsp,B0
lea rbp,qword ptr ss:[rsp+80]
mov qword ptr ss:[rbp+28],FFFFFFFFFFFFF
mov qword ptr ss:[rbp+18],rcx
mov rdx,qword ptr ss:[rbp+18]
lea rcx,qword ptr ds:[7FFCBA18AB10]
call appvisvsubsystems64.7FFCBA157D00
mov dword ptr ss:[rbp+14],eax
lea rax,qword ptr ds:[7FFCBA175D84]      00007FFCBA175D84:"6zD6eIhi4jWI7owk4"
mov qword ptr ss:[rbp+8],rax
mov dword ptr ss:[rbp+4],1EA4
lea rax,qword ptr ds:[7FFCBA175D96]      00007FFCBA175D96:"wkapdMFXG3ZDwNwzGlpMeZvFHdAbGT1
mov qword ptr ss:[rbp-8],rax
mov dword ptr ss:[rbp-C],B712
cmp dword ptr ss:[rbp+14],FFFFFFFF
jne appvisvsubsystems64.7FFCBA152D19
mov dword ptr ss:[rbp+24],0
jmp appvisvsubsystems64.7FFCBA152D9D
lea rcx,qword ptr ds:[7FFCBA18AB10]
call appvisvsubsystems64.7FFCBA157F10
mov dword ptr ss:[rbp-10],FFED
lea rax,qword ptr ds:[7FFCBA175DB6]      00007FFCBA175DB6:L"EjUqjSAfQHBcf1wjS"
mov qword ptr ss:[rbp-18],rax
mov dword ptr ss:[rbp-1C],4CF4
lea rcx,qword ptr ss:[rbp-38]
lea rdx,qword ptr ss:[rbp-30]
mov qword ptr ss:[rbp-58],rdx
call appvisvsubsystems64.7FFCBA152DE0
mov rcx,qword ptr ss:[rbp-58]
call appvisvsubsystems64.7FFCBA1530E0
mov qword ptr ss:[rbp-50],rax
jmp appvisvsubsystems64.7FFCBA152D61
mov rcx,qword ptr ss:[rbp-50]
call appvisvsubsystems64.7FFCBA15DAD0
jmp appvisvsubsystems64.7FFCBA152D6F
lea rcx,qword ptr ss:[rbp-30]
call appvisvsubsystems64.7FFCBA153120
lea rax,qword ptr ds:[7FFCBA175DDA]      00007FFCBA175DDA:"HQS22LKqT7OrK6FnloASmDGB8"
mov qword ptr ss:[rbp-40],rax
mov dword ptr ss:[rbp-44],3
mov dword ptr ss:[rbp-48],5DCA
mov dword ptr ss:[rbp+24],1
```

```
push rbx
sub rsp,60
mov rdx,qword ptr ds:[7FFCBE338338]      00007FFCBE338338:&"SuspendThread"
mov rbx,rcx
lea rcx,qword ptr ds:[7FFCBE338210]
call appvisvsubsystems64.7FFCBE333E5C
mov rcx,rbx
call rax
cmp eax,FFFFFFFF
jne appvisvsubsystems64.7FFCBE3318CA
xor eax,eax
jmp appvisvsubsystems64.7FFCBE3318E2
lea rcx,qword ptr ss:[rsp+20]
call appvisvsubsystems64.7FFCBE3312B0
mov rcx,qword ptr ds:[rax+20]
call appvisvsubsystems64.7FFCBE331CA0
mov eax,1
add rsp,60
pop rbx
ret
push rbx
sub rsp,30
cmp edx,1
jne appvisvsubsystems64.7FFCBE331963
mov rdx,qword ptr ds:[7FFCBE338358]      00007FFCBE338358:&"GetCurrentThreadId"
lea rcx,qword ptr ds:[7FFCBE338210]
call appvisvsubsystems64.7FFCBE333E5C
call rax
mov rdx,qword ptr ds:[7FFCBE338348]      00007FFCBE338348:&"OpenThread"
lea rcx,qword ptr ds:[7FFCBE338210]
mov ebx,eax
call appvisvsubsystems64.7FFCBE333E5C
mov r8d,ebx
xor edx,edx
mov ecx,1FFFFF
call rax
mov rdx,qword ptr ds:[7FFCBE3382E8]      00007FFCBE3382E8:&"CreateThread"
lea rcx,qword ptr ds:[7FFCBE338210]
and dword ptr ss:[rsp+48],0
mov rbx,rax
call appvisvsubsystems64.7FFCBE333E5C
lea rcx,qword ptr ss:[rsp+48]
mov r9,rbx
```

Entry point of AppvIsvSubsystems64.dll – Information vs Note



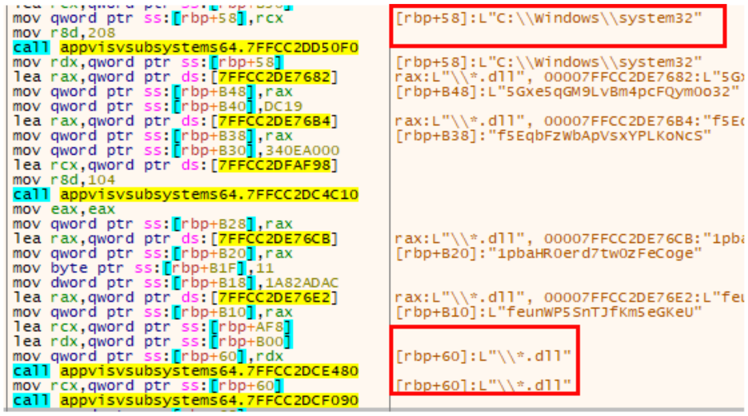
Differences in .data – Informtion vs Note

Shellcode injection

The way in which the sample injects shellcode into memory also varies from campaign to campaign. As can be seen in the report, “Note” reserves a memory space in its process and writes the shellcode into it.

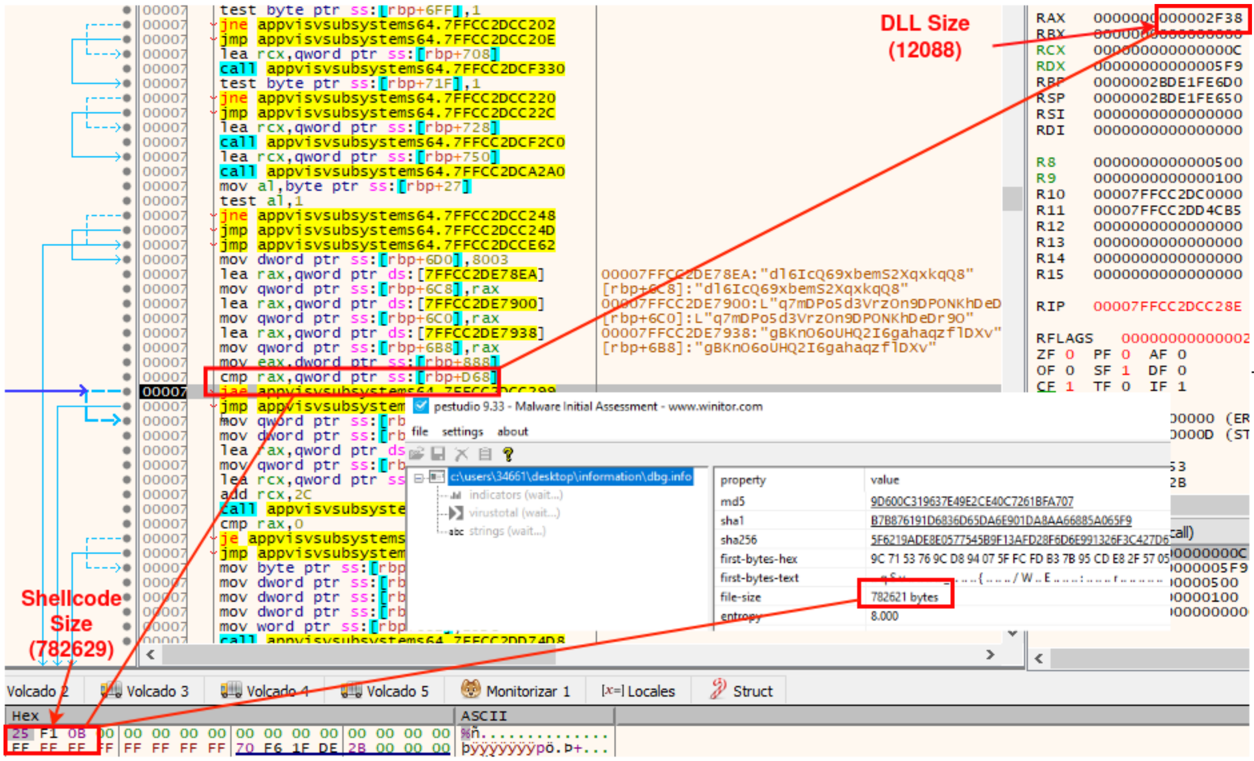
In this case the injection process is more sophisticated, **the injection will be triggered by modifying the .text section of legitimate libraries**. The following actions are performed by **AppvIsvSubsystems64.dll**.

In order to do that, first it goes through each of the DLLs stored in **System32**.



```
mov qword ptr ss:[rbp+58],rcx                 [rbp+58]:L"C:\\Windows\\system32"
mov r8d,208
call appvisvsubsystems64.7FFCC2DD50F0
mov rdx,qword ptr ss:[rbp+58]                 [rbp+58]:L"C:\\Windows\\system32"
lea rax,qword ptr ds:[7FFCC2DE7682]           rax:L"\\*.dll", 00007FFCC2DE7682:L"5G
mov qword ptr ss:[rbp+B48],rax                [rbp+B48]:L"5Gxe5qGM9LvBm4pcFQymOo32"
mov qword ptr ss:[rbp+B40],DC19
lea rax,qword ptr ds:[7FFCC2DE76B4]           rax:L"\\*.dll", 00007FFCC2DE76B4:"f5E
mov qword ptr ss:[rbp+B38],rax                [rbp+B38]:"f5EqbFzWbApVsxYPLKoNcS"
mov qword ptr ss:[rbp+B30],340EA000
lea rcx,qword ptr ds:[7FFCC2DFAF98]
mov r8d,104
call appvisvsubsystems64.7FFCC2DC4C10
mov eax,eax
mov qword ptr ss:[rbp+B28],rax
lea rax,qword ptr ds:[7FFCC2DE76CB]           rax:L"\\*.dll", 00007FFCC2DE76CB:"1pba
mov qword ptr ss:[rbp+B20],rax                [rbp+B20]:"1pbaHR0erd7twOzFeCoge"
mov byte ptr ss:[rbp+B1F],11
mov dword ptr ss:[rbp+B18],1A82ADAC
lea rax,qword ptr ds:[7FFCC2DE76E2]           rax:L"\\*.dll", 00007FFCC2DE76E2:L"feu
mov qword ptr ss:[rbp+B10],rax                [rbp+B10]:L"feunWP5SnTJfKm5eGKeU"
lea rcx,qword ptr ss:[rbp+AF8]
lea rdx,qword ptr ss:[rbp+B00]
mov qword ptr ss:[rbp+60],rdx                 [rbp+60]:L"\\*.dll"
call appvisvsubsystems64.7FFCC2DCE480
mov rcx,qword ptr ss:[rbp+60]                 [rbp+60]:L"\\*.dll"
call appvisvsubsystems64.7FFCC2DCF090
```

Next check the size of the DLL, it will only be a candidate if it has a size greater than or equal to **782629 bytes** (Shellcode size).



```
00007  test byte ptr ss:[rbp+6FF],1
00007  jne appvisvsubsystems64.7FFCC2DCC202
00007  jmp appvisvsubsystems64.7FFCC2DCC20E
00007  lea rcx,qword ptr ss:[rbp+708]
00007  call appvisvsubsystems64.7FFCC2DCF330
00007  test byte ptr ss:[rbp+71F],1
00007  jne appvisvsubsystems64.7FFCC2DCC220
00007  jmp appvisvsubsystems64.7FFCC2DCC22C
00007  lea rcx,qword ptr ss:[rbp+728]
00007  call appvisvsubsystems64.7FFCC2DCF2C0
00007  lea rcx,qword ptr ss:[rbp+750]
00007  call appvisvsubsystems64.7FFCC2DCA2A0
00007  mov al,byte ptr ss:[rbp+27]
00007  test al,1
00007  jne appvisvsubsystems64.7FFCC2DCC248
00007  jmp appvisvsubsystems64.7FFCC2DCC24D
00007  jmp appvisvsubsystems64.7FFCC2DCCE62
00007  mov dword ptr ss:[rbp+6D0],8003
00007  lea rax,qword ptr ds:[7FFCC2DE78EA]     00007FFCC2DE78EA:"dl6IcQ69xbemS2XqxkqQ8"
00007  mov qword ptr ss:[rbp+6C8],rax          [rbp+6C8]:"dl6IcQ69xbemS2XqxkqQ8"
00007  lea rax,qword ptr ds:[7FFCC2DE7900]     00007FFCC2DE7900:L"q7mDPo5d3VrzOn9DPONKhDeD
00007  mov qword ptr ss:[rbp+6C0],rax          [rbp+6C0]:L"q7mDPo5d3VrzOn9DPONKhDeDr9O"
00007  lea rax,qword ptr ds:[7FFCC2DE7938]     00007FFCC2DE7938:"gBKnO6oUHQ2I6gahaqzflDXv"
00007  mov qword ptr ss:[rbp+6B8],rax          [rbp+6B8]:"gBKnO6oUHQ2I6gahaqzflDXv"
00007  mov eax,dword ptr ss:[rbp+888]
00007  cmp rax,qword ptr ss:[rbp+D68]
00007  jae appvisvsubsystems64.7FFCC2DCC298
00007  jmp appvisvsubsystem
00007  mov qword ptr ss:[rb
00007  mov dword ptr ss:[rb
00007  lea rax,qword ptr ds
00007  mov qword ptr ss:[rb
00007  lea rcx,qword ptr ss
00007  add rcx,2C
00007  call appvisvsubsyste
00007  cmp rax,0
00007  je appvisvsubsystems
00007  jmp appvisvsubsystem
00007  mov byte ptr ss:[rbp
00007  mov dword ptr ss:[rb
00007  mov dword ptr ss:[rb
00007  mov dword ptr ss:[rb
00007  mov word ptr ss:[rbp
00007  call appvisvsubsystems64.7FFCC2DD74D8

RAX     000000000002F38
RBX     0000000000000000
RCX     000000000000000C
RDX     00000000000005F9
RBP     0000002BDE1FE6D0
RSP     0000002BDE1FE650
RSI     0000000000000000
RDI     0000000000000000

R8      0000000000000500
R9      0000000000000100
R10     00007FFCC2DC0000
R11     00007FFCC2DD4CB5
R12     0000000000000000
R13     0000000000000000
R14     0000000000000000
R15     0000000000000000

RIP     00007FFCC2DCC28E

RFLAGS  00000000000002
ZF 0  PF 0  AF 0
OF 0  SF 1  DF 0
CF 1  TF 0  IF 1
```

pestudio 9.33 - Malware Initial Assessment - www.winitor.com

c:\users\34661\desktop\information\dbg.info

property	value
md5	9D600C319637E49E2CE40C7261BFA707
sha1	B7B876191D6836D65DA6E901DA8AA66885A065F9
sha256	5F6219ADE8E0577545B9F13AFD28F6D6E991326F3C427D6
first-bytes-hex	9C 71 53 76 9C D8 94 07 5F FC FD B3 7B 95 CD E8 2F 57 05
first-bytes-text	..{..../W..E.....:.....r......
file-size	782621 bytes
entropy	8.000

```
Hex                                                 ASCII
25 F1 0B 00 00 00 00 00 00 00 00 00 00 00 00 00   %ñ..............
FE FF FF FF FF FF FF FF 70 F6 1F DE 2B 00 00 00   þÿÿÿÿÿÿÿpö.Þ+...
```

Test DLL size

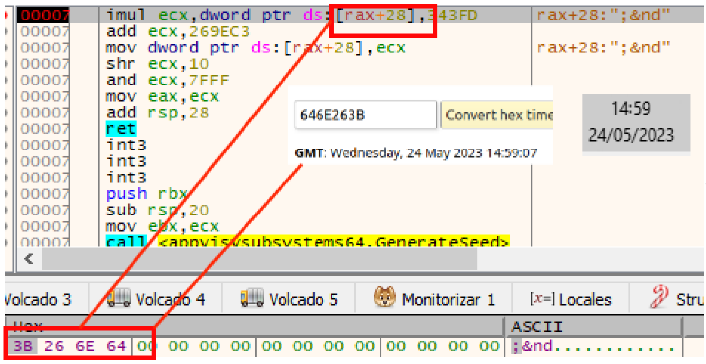
The sample also checks that the candidate DLL is not already loaded in the executable (Information.exe). That is, the following list are the DLLs already loaded in Information.exe, and, therefore, discarded by the malware:

msvcp140.dll	combase.dll
msi.dll	sechost.dll
appvisvsubsystem64.dll	msvcrt.dll
vcruntime140.dll	kernel32.dll
vcruntime140_1.dll	imm32.dll
ucrtbase.dll	gdi32.dll
bcrypt.dll	rpcrt4.dll
win32u.dll	shell32.dll
msvcp_win.dll	advapi32.dll
kernelbase.dll	user32.dll
gdi32full.dll	ntdll.dll

After said checks, the malware has a set of candidate DLLs. The DLL selected for injection will vary in each execution, thanks to a randomization function implemented in the sample. **A list of candidate DLLs by default – those that satisfy the requirements needed by the malware – in Windows 10 64b is provided at the end of this post**. This may vary depending on the operating system. However, following the steps in this post is not difficult to prepare your own script. A similar technique is described in the Netero1010 post.

The malware selects the DLL used for the injection as follows.

First, it uses the system time as a seed to apply a series of arithmetic operations on it. The result will be the seed of the next DLL to be checked.

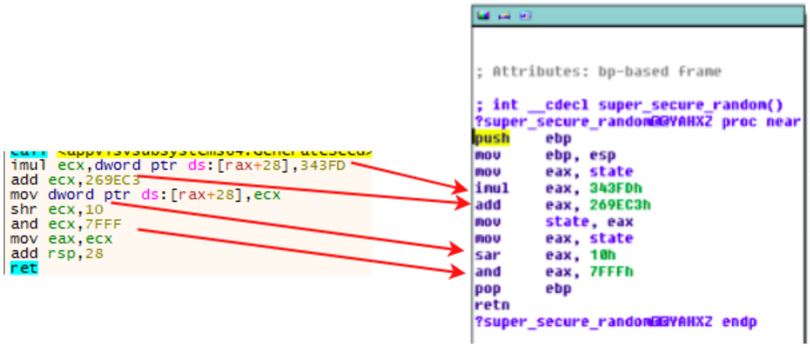


```
imul ecx,dword ptr ds:[rax+28],343FD        rax+28:";&nd"
add ecx,269EC3
mov dword ptr ds:[rax+28],ecx               rax+28:";&nd"
shr ecx,10
and ecx,7FFF
mov eax,ecx
add rsp,28
ret
int3
int3
int3
push rbx
sub rsp,20
mov ebx,ecx
call <appvisvsubsystems64.GenerateSeed>
```

```
Hex                                               ASCII
3B 26 6E 64 00 00 00 00 00 00 00 00 00 00 00 00   ;&nd............
```

Seed generation using system time

Interestingly, the values it uses for multiplication and addition are identical to those provided in a “Holiday Hack Challenge 2019” challenge organized by SANS. In particular, in the challenge there is a function called “**super_secure_random**” that performs the same operations with the same operands to a seed (“state”).

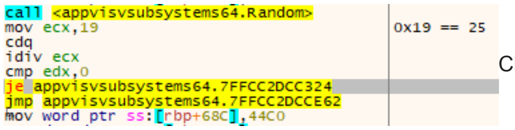


```
imul ecx,dword ptr ds:[rax+28],343FD
add ecx,269EC3
mov dword ptr ds:[rax+28],ecx
shr ecx,10
and ecx,7FFF
mov eax,ecx
add rsp,28
ret
```

```
; Attributes: bp-based frame

; int __cdecl super_secure_random()
?super_secure_random@@YAHXZ proc near
push    ebp
mov     ebp, esp
mov     eax, state
imul    eax, 343FDh
add     eax, 269EC3h
mov     state, eax
mov     eax, state
sar     eax, 10h
and     eax, 7FFFh
pop     ebp
retn
?super_secure_random@@YAHXZ endp
```

Comparison of the CTF algorithm (left) with the sample (right)

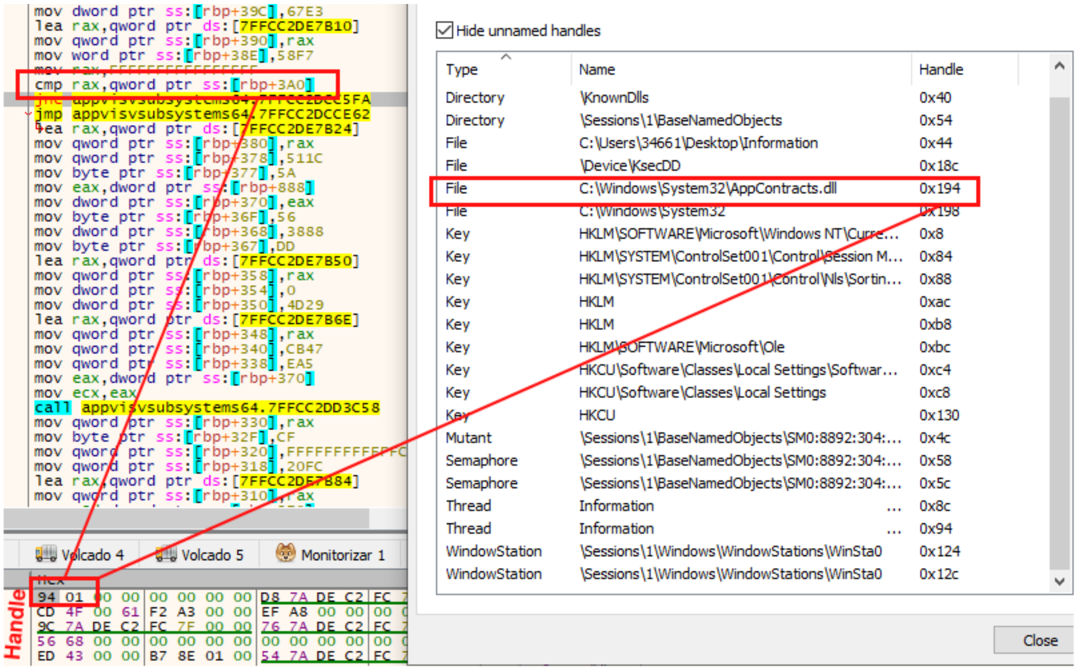
Finally, it checks if this resulting number is a multiple of 25 and if it is, it continues with the injection.



```
call <appvisvsubsystems64.Random>
mov ecx,19                                  0x19 == 25
cdq
idiv ecx
cmp edx,0
je appvisvsubsystems64.7FFCC2DCC324
jmp appvisvsubsystems64.7FFCC2DCCE62
mov word ptr ss:[rbp+68C],44C0
```

Choice of random DLL

Once the DLL has been chosen, the sample accesses the library with CreateFileW (in the execution used to take the pictures for this example the DLL selected was “AppContracts.dll”).



```
mov dword ptr ss:[rbp+39C],67E3
lea rax,qword ptr ds:[7FFCC2DE7B10]
mov qword ptr ss:[rbp+390],rax
mov word ptr ss:[rbp+38E],58F7
mov rax,FFFFFFFFFFFFFFFF
cmp rax,qword ptr ss:[rbp+3A0]
jne appvisvsubsystems64.7FFCC2DCC5FA
jmp appvisvsubsystems64.7FFCC2DCCE62
lea rax,qword ptr ds:[7FFCC2DE7B24]
mov qword ptr ss:[rbp+380],rax
mov qword ptr ss:[rbp+378],511C
mov byte ptr ss:[rbp+377],5A
mov eax,dword ptr ss:[rbp+888]
mov dword ptr ss:[rbp+370],eax
mov byte ptr ss:[rbp+36F],56
mov dword ptr ss:[rbp+368],3888
mov byte ptr ss:[rbp+367],DD
lea rax,qword ptr ds:[7FFCC2DE7B50]
mov qword ptr ss:[rbp+358],rax
mov dword ptr ss:[rbp+354],0
mov dword ptr ss:[rbp+350],4D29
lea rax,qword ptr ds:[7FFCC2DE7B6E]
mov qword ptr ss:[rbp+348],rax
mov qword ptr ss:[rbp+340],CB47
mov qword ptr ss:[rbp+338],EA5
mov eax,dword ptr ss:[rbp+370]
mov ecx,eax
call appvisvsubsystems64.7FFCC2DD3C58
mov qword ptr ss:[rbp+330],rax
mov byte ptr ss:[rbp+32F],CF
mov qword ptr ss:[rbp+320],FFFFFFFFFFFFFFFC
mov qword ptr ss:[rbp+318],20FC
lea rax,qword ptr ds:[7FFCC2DE7B84]
mov qword ptr ss:[rbp+310],rax
```

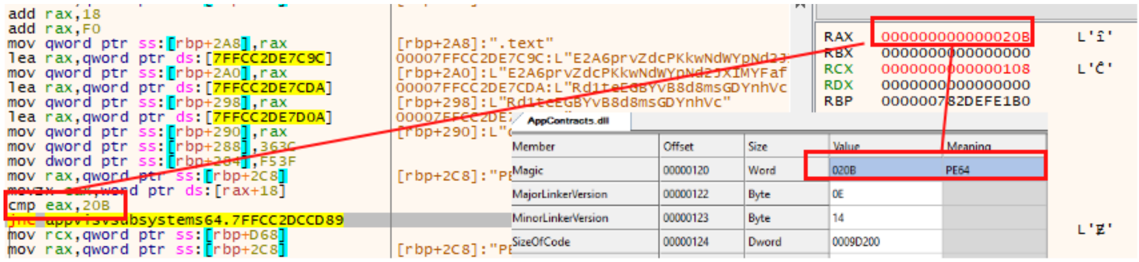
```
Hex
94 01 00 00 00 00 00 00 D8 7A DE C2 FC
CD 4F 00 61 F2 A3 00 00 EF A8 00 00 00
9C 7A DE C2 FC 7F 00 00 76 7A DE C2 FC
56 68 00 00 00 00 00 00 00 00 00 00 00
ED 43 00 00 B7 8E 01 00 54 7A DE C2 FC
```

☑ Hide unnamed handles

Type	Name	Handle
Directory	\KnownDlls	0x40
Directory	\Sessions\1\BaseNamedObjects	0x54
File	C:\Users\34661\Desktop\Information	0x44
File	\Device\KsecDD	0x18c
File	C:\Windows\System32\AppContracts.dll	0x194
File	C:\Windows\System32	0x198
Key	HKLM\SOFTWARE\Microsoft\Windows NT\Curre...	0x8
Key	HKLM\SYSTEM\ControlSet001\Control\Session M...	0x84
Key	HKLM\SYSTEM\ControlSet001\Control\Nls\Sortin...	0x88
Key	HKLM	0xac
Key	HKLM	0xb8
Key	HKLM\SOFTWARE\Microsoft\Ole	0xbc
Key	HKCU\Software\Classes\Local Settings\Softwar...	0xc4
Key	HKCU\Software\Classes\Local Settings	0xc8
Key	HKCU	0x130
Mutant	\Sessions\1\BaseNamedObjects\SM0:8892:304:...	0x4c
Semaphore	\Sessions\1\BaseNamedObjects\SM0:8892:304:...	0x58
Semaphore	\Sessions\1\BaseNamedObjects\SM0:8892:304:...	0x5c
Thread	Information ...	0x8c
Thread	Information ...	0x94
WindowStation	\Sessions\1\Windows\WindowStations\WinSta0	0x124
WindowStation	\Sessions\1\Windows\WindowStations\WinSta0	0x12c

DLL opened

At this point, the sample reads the headers of the libraries to pass a round of checks again. First, look at the “Magic” value to verify that it is a 64-bit system DLL (PE64).



```
add rax,18
add rax,F0
mov qword ptr ss:[rbp+2A8],rax
lea rax,qword ptr ds:[7FFCC2DE7C9C]
mov qword ptr ss:[rbp+2A0],rax
lea rax,qword ptr ds:[7FFCC2DE7CDA]
mov qword ptr ss:[rbp+298],rax
lea rax,qword ptr ds:[7FFCC2DE7D0A]
mov qword ptr ss:[rbp+290],rax
mov qword ptr ss:[rbp+288],363C
mov dword ptr ss:[rbp+284],F53F
mov rax,qword ptr ss:[rbp+2C8]
movzx eax,word ptr ds:[rax+18]
cmp eax,20B
jne appvisvsubsystems64.7FFCC2DCCD89
mov rcx,qword ptr ss:[rbp+D68]
mov rax,qword ptr ss:[rbp+2C8]
```

```
[rbp+2A8]:".text"
00007FFCC2DE7C9C:L"E2A6prvZdcPKkwNdWYpNd2J
[rbp+2A0]:L"E2A6prvZdcPKkwNdWYpNd2JXIMYFaf
00007FFCC2DE7CDA:L"Rd1teEGBYvB8d8msGDYnhVc
[rbp+298]:L"Rd1teEGBYvB8d8msGDYnhVc"
00007FFCC2DE
[rbp+290]:L"
[rbp+2C8]:"PE
[rbp+2C8]:"PE
```

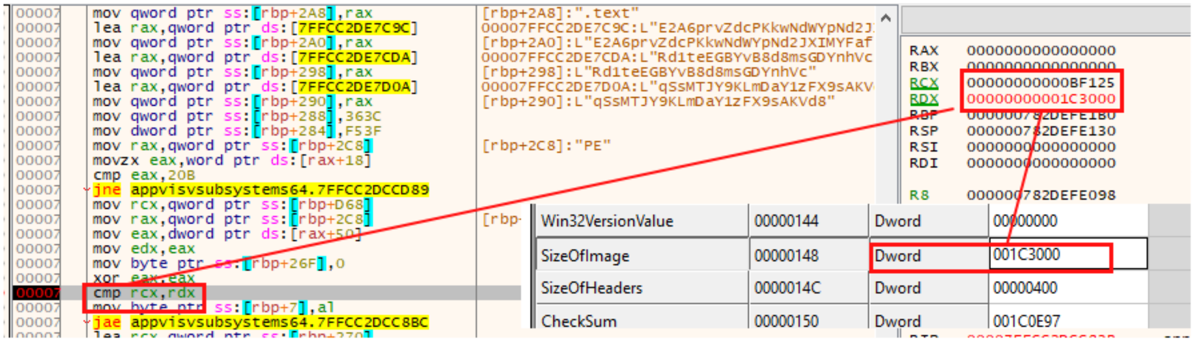
```
RAX     000000000000020B    L'ȋ'
RBX     0000000000000000
RCX     0000000000000108    L'Ĉ'
RDX     0000000000000000
RBP     000000782DEFE1B0
                            L'Ȩ'
```

AppContracts.dll

Member	Offset	Size	Value	Meaning
Magic	00000120	Word	020B	PE64
MajorLinkerVersion	00000122	Byte	0E	
MinorLinkerVersion	00000123	Byte	14	
SizeOfCode	00000124	Dword	0009D200	

Magic value of the DLL candidate

Another comprobation is to **check the "SizeOfImage"** field, which refers to the **size to reserve in memory to load the executable**. If it is less than **782629 bytes** (shellcode size), it is discarded.



```
00007  mov qword ptr ss:[rbp+2A8],rax
00007  lea rax,qword ptr ds:[7FFCC2DE7C9C]
00007  mov qword ptr ss:[rbp+2A0],rax
00007  lea rax,qword ptr ds:[7FFCC2DE7CDA]
00007  mov qword ptr ss:[rbp+298],rax
00007  lea rax,qword ptr ds:[7FFCC2DE7D0A]
00007  mov qword ptr ss:[rbp+290],rax
00007  mov qword ptr ss:[rbp+288],363C
00007  mov dword ptr ss:[rbp+284],F53F
00007  mov rax,qword ptr ss:[rbp+2C8]
00007  movzx eax,word ptr ds:[rax+18]
00007  cmp eax,20B
00007  jne appvisvsubsystems64.7FFCC2DCCD89
00007  mov rcx,qword ptr ss:[rbp+D68]
00007  mov rax,qword ptr ss:[rbp+2C8]
00007  mov eax,dword ptr ds:[rax+50]
00007  mov edx,eax
00007  mov byte ptr ss:[rbp+26F],0
00007  xor eax,eax
00007  cmp rcx,rdx
00007  mov byte ptr ss:[rbp+7],al
00007  jae appvisvsubsystems64.7FFCC2DCC8BC
```

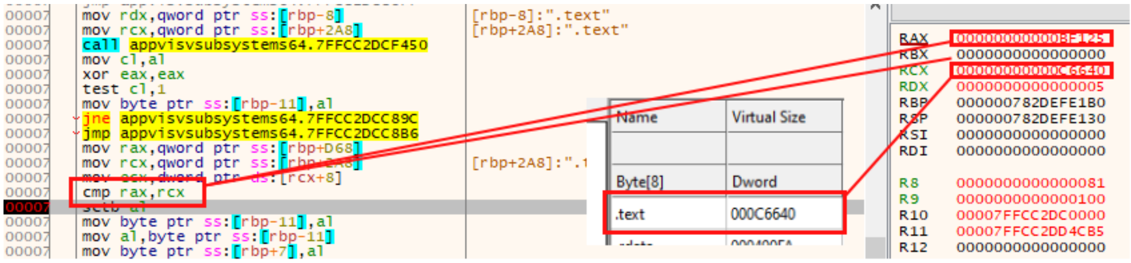
```
[rbp+2A8]:".text"
00007FFCC2DE7C9C:L"E2A6prvZdcPKkwNdWYpNd2J
[rbp+2A0]:L"E2A6prvZdcPKkwNdWYpNd2JXIMYFaf
00007FFCC2DE7CDA:L"Rd1teEGBYvB8d8msGDYnhVc
[rbp+298]:L"Rd1teEGBYvB8d8msGDYnhVc"
00007FFCC2DE7D0A:L"qSsMTJY9KLmDaY1zFX9sAKV
[rbp+290]:L"qSsMTJY9KLmDaY1zFX9sAKVd8"

[rbp+2C8]:"PE"

[rbp
```

```
RAX     0000000000000000
RBX     0000000000000000
RCX     00000000000BF125
RDX     00000000001C3000
RBP     000000782DEFE1B0
RSP     000000782DEFE130
RSI     0000000000000000
RDI     0000000000000000

R8      000000782DEFE098
```

Win32VersionValue	00000144	Dword	00000000
SizeOfImage	00000148	Dword	001C3000
SizeOfHeaders	0000014C	Dword	00000400
CheckSum	00000150	Dword	001C0E97

SizeOfCode of the candidate DLL

Finally, the sample checks the size of the ".text" section (where the shellcode will be injected) and verifies that it is larger than **782629 bytes**.



```
00007  mov rdx,qword ptr ss:[rbp-8]
00007  mov rcx,qword ptr ss:[rbp+2A8]
00007  call appvisvsubsystems64.7FFCC2DCF450
00007  mov cl,al
00007  xor eax,eax
00007  test cl,1
00007  mov byte ptr ss:[rbp-11],al
00007  jne appvisvsubsystems64.7FFCC2DCC89C
00007  jmp appvisvsubsystems64.7FFCC2DCC8B6
00007  mov rax,qword ptr ss:[rbp+D68]
00007  mov rcx,qword ptr ss:[rbp+2A8]
00007  mov ecx,dword ptr ds:[rcx+8]
00007  cmp rax,rcx
00007  setb al
00007  mov byte ptr ss:[rbp-11],al
00007  mov al,byte ptr ss:[rbp-11]
00007  mov byte ptr ss:[rbp+7],al
```

```
[rbp-8]:".text"
[rbp+2A8]:".text"

[rbp+2A8]:".t
```

```
RAX     00000000000BF125
RBX     0000000000000000
RCX     00000000000C6640
RDX     0000000000000005
RBP     000000782DEFE1B0
RSP     000000782DEFE130
RSI     0000000000000000
RDI     0000000000000000

R8      0000000000000081
R9      0000000000000100
R10     00007FFCC2DC0000
R11     00007FFCC2DD4CB5
R12     0000000000000000
```

Name	Virtual Size
Byte[8]	Dword
.text	000C6640

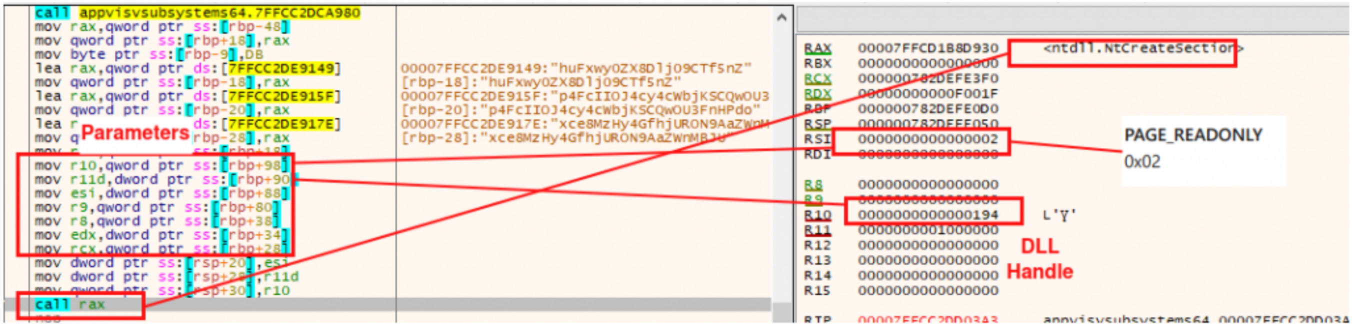
Size of ".text" section of the candidate DLL

Unfortunately, at this point AppContracts.dll (the DLL selected in this execution) does not meet the injection requirements, and the search for the next optimal DLL is **AppxPackaging.dll**, in this example.

After this screening, **the malware will have about 283 candidates for injection** in the system used. At the end of the post you will find a table with the possible DLLs in which the malware could be injected, **consiering the size required for the injection by this sample (782629) and the system where it be executed**.

At this point, the chosen DLL (in this example, at this moment, AppxPackaging.dll) is ready to receive the shellcode, using the following injection method.

Step1. Subsequently, it makes a call to **NtCreateSection** to create a memory section in the process.



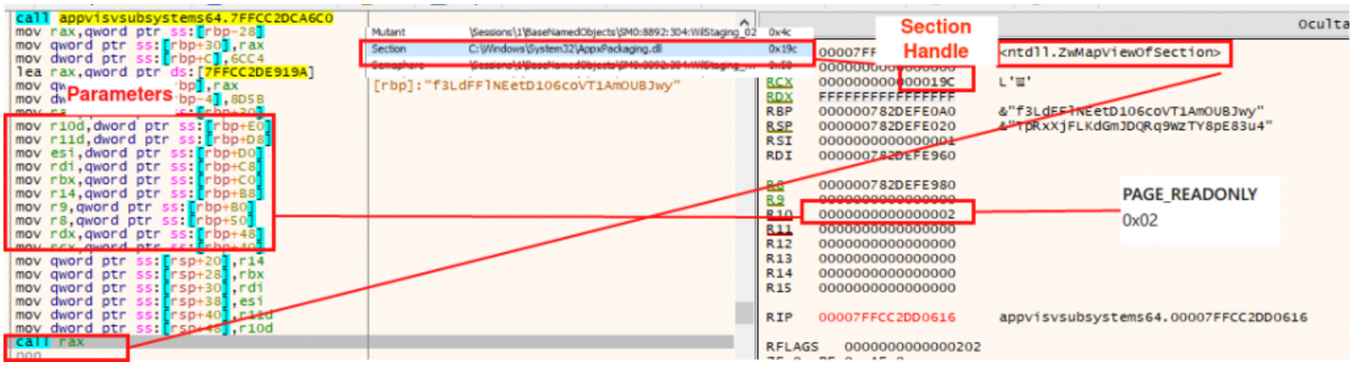
```
call appvisvsubsystems64.7FFCC2DCA980
mov rax,qword ptr ss:[rbp-48]
mov qword ptr ss:[rbp+18],rax
mov byte ptr ss:[rbp-9],DB
lea rax,qword ptr ds:[7FFCC2DE9149]
mov qword ptr ss:[rbp-18],rax
lea rax,qword ptr ds:[7FFCC2DE915F]
mov qword ptr ss:[rbp-20],rax
lea r                 ds:[7FFCC2DE917E]
mov q                 rbp-28],rax
mov r10,qword ptr ss:[rbp+98]
mov r11d,dword ptr ss:[rbp+90]
mov esi,dword ptr ss:[rbp+88]
mov r9,qword ptr ss:[rbp+80]
mov r8,qword ptr ss:[rbp+38]
mov edx,dword ptr ss:[rbp+34]
mov rcx,qword ptr ss:[rbp+28]
mov dword ptr ss:[rsp+20],esi
mov dword ptr ss:[rsp+28],r11d
mov qword ptr ss:[rsp+30],r10
call rax
```

```
00007FFCC2DE9149:"huFxwy0ZX8Dlj09CTf5nZ"
[rbp-18]:"huFxwy0ZX8Dlj09CTf5nZ"
00007FFCC2DE915F:"p4FcIIOJ4cy4cWbjKSCQwOU3
[rbp-20]:"p4FcIIOJ4cy4cWbjKSCQwOU3FnHPdo"
00007FFCC2DE917E:"xce8MzHy4GfhjURON9AaZWnM
[rbp-28]:"xce8MzHy4GfhjURON9AaZWnMBJU"
```

```
RAX     00007FFCD1B8D930    <ntdll.NtCreateSection>
RBX     0000000000000000
RCX     000000782DEFE3F0
RDX     00000000000F001F
RBP     000000782DEFE0D0
RSP     000000782DEFE050
RSI     0000000000000002
RDI     0000000000000000

R8      0000000000000000
R9      0000000000000000
R10     0000000000000194    L'Ɣ'
R11     0000000001000000
R12     0000000000000000
R13     0000000000000000
R14     0000000000000000
R15     0000000000000000

RIP     00007FFCC2DD03A3    appvisvsubsystems64.00007FFCC2DD03A
```

Create section to map the DLL

Step2. And map the library in that section with **ZwMapViewOfSection.**



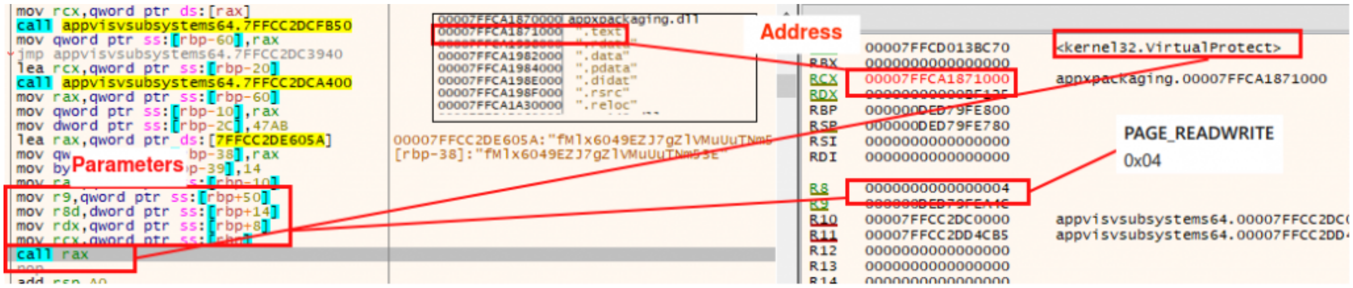
```
call appvisvsubsystems64.7FFCC2DCA6C0
mov rax,qword ptr ss:[rbp-28]
mov qword ptr ss:[rbp+30],rax
mov dword ptr ss:[rbp+C],6CC4
lea rax,qword ptr ds:[7FFCC2DE919A]
mov qw           bp],rax
mov dw           bp-4],8D5B
mov r10d,dword ptr ss:[rbp+E0]
mov r11d,dword ptr ss:[rbp+D8]
mov esi,dword ptr ss:[rbp+D0]
mov rdi,qword ptr ss:[rbp+C8]
mov rbx,qword ptr ss:[rbp+C0]
mov r14,qword ptr ss:[rbp+B8]
mov r9,qword ptr ss:[rbp+B0]
mov r8,qword ptr ss:[rbp+50]
mov rdx,qword ptr ss:[rbp+48]
mov qword ptr ss:[rsp+20],r14
mov qword ptr ss:[rsp+28],rbx
mov qword ptr ss:[rsp+30],rdi
mov dword ptr ss:[rsp+38],esi
mov dword ptr ss:[rsp+40],r11d
mov dword ptr ss:[rsp+48],r10d
call rax
```

```
Mutant     \Sessions\1\BaseNamedObjects\SM0:8892:304:WilStaging_02   0x4c
Section    C:\Windows\System32\AppxPackaging.dll                     0x19c

[rbp]:"f3LdFF1NEetD106coVT1AmOUBJwy"
```

```
        00007FF                  <ntdll.ZwMapViewOfSection>
RCX     000000000000019C         L'Ⅲ'
RDX     FFFFFFFFFFFFFFFF
RBP     000000782DEFE0A0         &"f3LdFF1NEetD106coVT1AmOUBJwy"
RSP     000000782DEFE020         &"TpRxXjFLKdGmJDQRq9WzTY8pE83u4"
RSI     0000000000000001
RDI     000000782DEFE960

R8      000000782DEFE980
R9      0000000000000000
R10     0000000000000002
R11     0000000000000000
R12     0000000000000000
R13     0000000000000000
R14     0000000000000000
R15     0000000000000000

RIP     00007FFCC2DD0616         appvisvsubsystems64.00007FFCC2DD0616

RFLAGS  0000000000000202
```

Map the DLL

Step3. Once the library is mapped in memory, the write permissions are modified to be able to write the shellcode in it. **The address pointed to by VirtualProtect corresponds to the ".text" section of the DLL**.



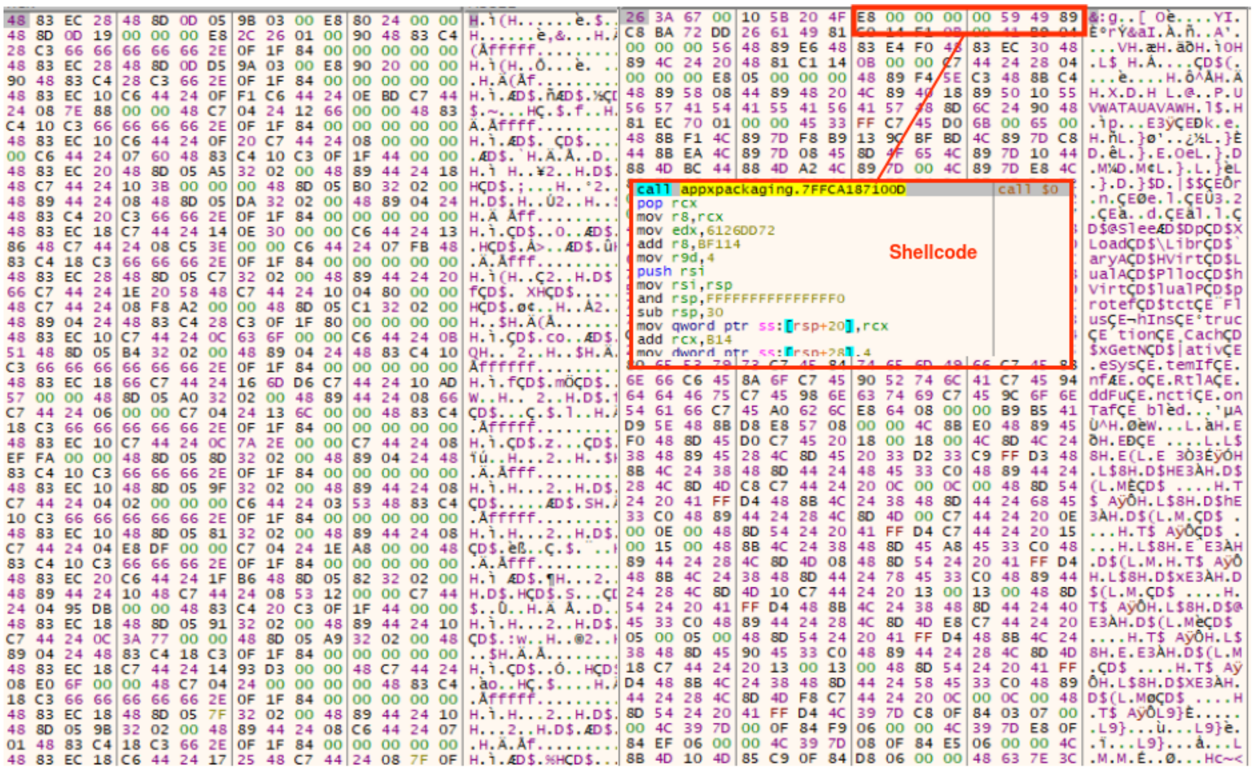
```
mov rcx,qword ptr ds:[rax]
call appvisvsubsystems64.7FFCC2DCFB50
mov qword ptr ss:[rbp-60],rax
jmp appvisvsubsystems64.7FFCC2DC3940
lea rcx,qword ptr ss:[rbp-20]
call appvisvsubsystems64.7FFCC2DCA400
mov rax,qword ptr ss:[rbp-60]
mov qword ptr ss:[rbp-10],rax
mov dword ptr ss:[rbp-2C],47AB
lea rax,qword ptr ds:[7FFCC2DE605A]
mov qw           bp-38],rax
mov by           p-39],14
mov r9,qword ptr ss:[rbp+50]
mov r8d,dword ptr ss:[rbp+14]
mov rdx,qword ptr ss:[rbp+8]
call rax
```

```
00007FFCA1870000  appxpackaging.dll
00007FFCA1871000  ".text"
00007FFCA1982000  ".data"
00007FFCA1984000  ".pdata"
00007FFCA198E000  ".didat"
00007FFCA198F000  ".rsrc"
00007FFCA1A30000  ".reloc"

00007FFCC2DE605A:"fM1x6049EZJ7gZ1VMuUuTNm5
[rbp-38]:"fM1x6049EZJ7gZ1VMuUuTNm53E"
```

```
        00007FFCD013BC70         <kernel32.VirtualProtect>
RBX     0000000000000000
RCX     00007FFCA1871000         appxpackaging.00007FFCA1871000
RBP     000000DED79FE800
RSP     000000DED79FE780
RSI     0000000000000000
RDI     0000000000000000

R8      0000000000000004
R10     00007FFCC2DC0000         appvisvsubsystems64.00007FFCC2DC
R11     00007FFCC2DD4CB5         appvisvsubsystems64.00007FFCC2DD
R12     0000000000000000
R13     0000000000000000
```

Change protection to RW (.text section)

Step4. After that, write the shellcode in the ".text" section.



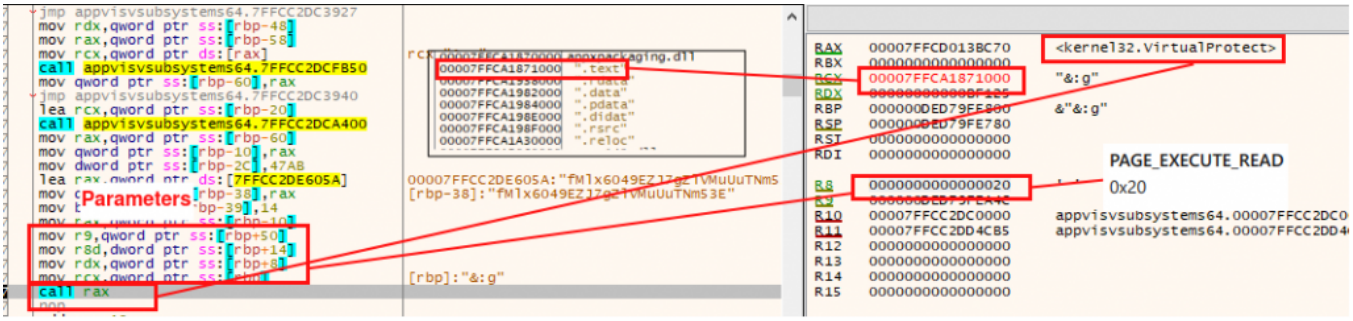
```
48 83 EC 28 48 8D 0D 05 9B 03 00 E8 80 24 00 00  H.ì(H.......è.$..
48 8D 0D 19 00 00 00 E8 2C 26 01 00 90 48 83 C4  H.......è,&...H.Ä
28 C3 66 66 66 66 66 2E 0F 1F 84 00 00 00 00 00  (Ãfffff.........
48 83 EC 28 48 8D 0D D5 9A 03 00 E8 90 20 00 00  H.ì(H..Õ...è. ..
90 48 83 C4 28 C3 66 2E 0F 1F 84 00 00 00 00 00  .H.Ä(Ãf.........
48 83 EC 10 C6 44 24 0F F1 C6 44 24 0E BD C7 44  H.ì.ÆD$.ñÆD$.½ÇD
24 08 7E 88 00 00 48 C7 04 24 12 66 00 00 48 83  $.~...HÇ.$.f..H.
C4 10 C3 66 66 66 66 2E 0F 1F 84 00 00 00 00 00  Ä.Ãffff.........
48 83 EC 10 C6 44 24 0F 20 C7 44 24 08 00 00 00  H.ì.ÆD$. ÇD$....
00 C6 44 24 07 60 48 83 C4 10 C3 0F 1F 44 00 00  .ÆD$.`H.Ä.Ã..D..
48 83 EC 20 48 8D 05 A5 32 02 00 48 89 44 24 18  H.ì H..¥2..H.D$.
48 C7 44 24 10 3B 00 00 00 48 8D 05 B0 32 02 00  HÇD$.;...H..°2..
48 89 44 24 08 48 8D 05 DA 32 02 00 48 89 04 24  H.D$.H..Ú2..H..$
48 83 C4 20 C3 66 66 2E 0F 1F 84 00 00 00 00 00  H.Ä Ãff.........
48 83 EC 18 C7 44 24 14 0E 30 00 00 C6 44 24 13  H.ì.ÇD$..0..ÆD$.
86 48 C7 44 24 08 C5 3E 00 00 C6 44 24 07 FB 48  .HÇD$.Å>..ÆD$.ûH
83 C4 18 C3 66 66 66 2E 0F 1F 84 00 00 00 00 00  .Ä.Ãfff.........
48 83 EC 28 48 8D 05 C7 32 02 00 48 89 44 24 20  H.ì(H..Ç2..H.D$ 
66 C7 44 24 1E 20 58 48 C7 44 24 10 04 80 00 00  fÇD$. XHÇD$.....
48 C7 44 24 08 F8 A2 00 00 48 8D 05 C1 32 02 00  HÇD$.ø¢..H..Á2..
48 89 04 24 48 83 C4 28 C3 0F 1F 80 00 00 00 00  H..$H.Ä(Ã.......
48 83 EC 10 C7 44 24 0C 63 6F 00 00 C6 44 24 0B  H.ì.ÇD$.co..ÆD$.
51 48 8D 05 B4 32 02 00 48 89 04 24 48 83 C4 10  QH..´2..H..$H.Ä.
C3 66 66 66 66 66 66 2E 0F 1F 84 00 00 00 00 00  Ãffffff.........
48 83 EC 18 66 C7 44 24 16 6D D6 C7 44 24 10 AD  H.ì.fÇD$.mÖÇD$..
57 00 00 48 8D 05 A0 32 02 00 48 89 44 24 08 66  W..H.. 2..H.D$.f
C7 44 24 06 00 00 C7 04 24 13 6C 00 00 48 83 C4  ÇD$...Ç.$.l..H.Ä
18 C3 66 66 66 66 66 2E 0F 1F 84 00 00 00 00 00  .Ãfffff.........
48 83 EC 10 C7 44 24 0C 7A 2E 00 00 C7 44 24 08  H.ì.ÇD$.z...ÇD$.
EF FA 00 00 48 8D 05 8D 32 02 00 48 89 04 24 48  ïú..H...2..H..$H
83 C4 10 C3 66 66 66 2E 0F 1F 84 00 00 00 00 00  .Ä.Ãfff.........
48 83 EC 10 48 8D 05 9F 32 02 00 48 89 44 24 08  H.ì.H...2..H.D$.
C7 44 24 04 02 00 00 00 C6 44 24 03 53 48 83 C4  ÇD$.....ÆD$.SH.Ä
10 C3 66 66 66 66 66 2E 0F 1F 84 00 00 00 00 00  .Ãfffff.........
48 83 EC 10 48 8D 05 81 32 02 00 48 89 44 24 08  H.ì.H...2..H.D$.
C7 44 24 04 E8 DF 00 00 C7 04 24 1E A8 00 00 48  ÇD$.èß..Ç.$.¨..H
83 C4 10 C3 66 66 66 2E 0F 1F 84 00 00 00 00 00  .Ä.Ãfff.........
48 83 EC 20 C6 44 24 1F B6 48 8D 05 82 32 02 00  H.ì ÆD$.¶H...2..
48 89 44 24 10 48 C7 44 24 08 53 12 00 00 C7 44  H.D$.HÇD$.S...ÇD
24 04 95 DB 00 00 48 83 C4 20 C3 0F 1F 44 00 00  $..Û..H.Ä Ã..D..
48 83 EC 18 48 8D 05 91 32 02 00 48 89 44 24 10  H.ì.H...2..H.D$.
C7 44 24 0C 3A 77 00 00 48 8D 05 A9 32 02 00 48  ÇD$.:w..H..©2..H
89 04 24 48 83 C4 18 C3 0F 1F 84 00 00 00 00 00  ..$H.Ä.Ã........
48 83 EC 18 C7 44 24 14 93 D3 00 00 48 C7 44 24  H.ì.ÇD$..Ó..HÇD$
08 E0 6F 00 00 48 C7 04 24 00 00 00 00 48 83 C4  .ào..HÇ.$....H.Ä
18 C3 66 66 66 66 66 2E 0F 1F 84 00 00 00 00 00  .Ãfffff.........
48 83 EC 18 48 8D 05 7F 32 02 00 48 89 44 24 10  H.ì.H...2..H.D$.
48 8D 05 9B 32 02 00 48 89 44 24 08 C6 44 24 07  H...2..H.D$.ÆD$.
01 48 83 C4 18 C3 66 2E 0F 1F 84 00 00 00 00 00  .H.Ä.Ãf.........
48 83 EC 18 C6 44 24 17 25 48 C7 44 24 08 7F 0F  H.ì.ÆD$.%HÇD$...
```

```
26 3A 67 00 10 5B 20 4F E8 00 00 00 00 59 49 89  &:g..[ Oè....YI.
C8 BA 72 DD 26 61 49 81 C0 14 F1 0B 00 41 B0 04  È°rÝ&aI.À.ñ..A'.
00 00 00 56 48 89 E6 48 83 E4 F0 48 83 EC 30 48  ...VH.æH.äðH.ì0H
89 4C 24 20 48 81 C1 14 0B 00 00 C7 44 24 28 04  .L$ H.Á....ÇD$(.
00 00 00 E8 05 00 00 00 48 89 F4 5E C3 48 8B C4  ...è....H.ô^ÃH.Ä
48 89 58 08 44 89 48 20 4C 89 40 18 89 50 10 55  H.X.D.H L.@..P.U
56 57 41 54 41 55 41 56 41 57 48 8D 6C 24 90 48  VWATAUAVAWH.l$.H
81 EC 70 01 00 00 45 33 FF C7 45 D0 6B 00 65 00  .ìp...E3ÿÇEÐk.e.
48 8B F1 4C 89 7D F8 B9 13 9C BF BD 4C 89 7D C8  H.ñL.}ø¹..¿½L.}È
44 8B EA 4C 89 7D 08 45 8D 4F 65 4C 89 7D 10 44  D.êL.}.E.OeL.}.D
88 4D BC 44 88 4D A2 4C 89 7D 00 4C 89 7D E8 4C  .M¼D.M¢L.}.L.}èL
                                                 .}.D.}$D.|$$ÇEÔr
                                                 .n.ÇEØe.l.ÇEÜ3.2
                                                 .ÇEà..d.ÇEäl.l.Ç
                                                 D$@SleeÆD$DpÇD$X
                                                 LoadÇD$\LibrÇD$`
                                                 aryAÇD$HVirtÇD$L
                                                 ualAÇD$PllocÇD$h
                                                 VirtÇD$lualPÇD$p
                                                 rotefÇD$tctÇE¨Fl
                                                 usÇE¬hInsÇE°truc
                                                 ÇE´tionÇE¸CachÇD
                                                 $xGetNÇD$|ativÇE
80 65 53 79 73 C7 45 84 74 65 6D 49 66 C7 45 88  .eSysÇE.temIfÇE.
6E 66 C6 45 8A 6F C7 45 90 52 74 6C 41 C7 45 94  nfÆE.oÇE.RtlAÇE.
64 64 46 75 C7 45 98 6E 63 74 69 C7 45 9C 6F 6E  ddFuÇE.nctiÇE.on
54 61 66 C7 45 A0 62 6C E8 64 08 00 00 B9 B5 41  TafÇE blèd...¹µA
D9 5E 48 8B D8 E8 57 08 00 00 4C 8B E0 48 89 45  Ù^H.ØèW...L.àH.E
F0 48 8D 45 D0 C7 45 20 18 00 18 00 4C 8D 4C 24  ðH.EÐÇE ....L.L$
38 48 89 45 28 4C 8D 45 20 33 D2 33 C9 FF D3 48  8H.E(L.E 3Ò3ÉÿÓH
8B 4C 24 38 48 8D 44 24 48 45 33 C0 48 89 44 24  .L$8H.D$HE3ÀH.D$
28 4C 8D 4D C8 C7 44 24 20 0C 00 0C 00 48 8D 54  (L.MÈÇD$ ....H.T
24 20 41 FF D4 48 8B 4C 24 38 48 8D 44 24 68 45  $ AÿÔH.L$8H.D$hE
33 C0 48 89 44 24 28 4C 8D 4D 00 C7 44 24 20 0E  3ÀH.D$(L.M.ÇD$ .
00 0E 00 48 8D 54 24 20 41 FF D4 C7 44 24 20 15  ...H.T$ AÿÔÇD$ .
00 15 00 48 8B 4C 24 38 48 8D 45 A8 45 33 C0 48  ...H.L$8H.E¨E3ÀH
89 44 24 28 4C 8D 4D 08 48 8D 54 24 20 41 FF D4  .D$(L.M.H.T$ AÿÔ
48 8B 4C 24 38 48 8D 44 24 78 45 33 C0 48 89 44  H.L$8H.D$xE3ÀH.D
24 28 4C 8D 4D 10 C7 44 24 20 13 00 13 00 48 8D  $(L.M.ÇD$ ....H.
54 24 20 41 FF D4 48 8B 4C 24 38 48 8D 44 24 40  T$ AÿÔH.L$8H.D$@
45 33 C0 48 89 44 24 28 4C 8D 4D E8 C7 44 24 20  E3ÀH.D$(L.MèÇD$ 
05 00 05 00 48 8D 54 24 20 41 FF D4 48 8B 4C 24  ....H.T$ AÿÔH.L$
38 48 8D 45 90 45 33 C0 48 89 44 24 28 4C 8D 4D  8H.E.E3ÀH.D$(L.M
18 C7 44 24 20 13 00 13 00 48 8D 54 24 20 41 FF  .ÇD$ ....H.T$ Aÿ
D4 48 8B 4C 24 38 48 8D 44 24 58 45 33 C0 48 89  ÔH.L$8H.D$XE3ÀH.
44 24 28 4C 8D 4D F8 C7 44 24 20 0C 00 0C 00 48  D$(L.MøÇD$ ....H
8D 54 24 20 41 FF D4 4C 39 7D C8 0F 84 03 07 00  .T$ AÿÔL9}È.....
00 4C 39 7D 00 0F 84 F9 06 00 00 4C 39 7D E8 0F  .L9}...ù...L9}è.
84 EF 06 00 00 4C 39 7D 08 0F 84 E5 06 00 00 4C  .ï...L9}...å...L
8B 4D 10 4D 85 C9 0F 84 D8 06 00 00 48 63 7E 3C  .M.M.É..Ø...Hc~<
```

```
call appxpackaging.7FFCA187100D        call $0
pop rcx
mov r8,rcx
mov edx,6126DD72
add r8,BF114
mov r9d,4
push rsi
mov rsi,rsp
and rsp,FFFFFFFFFFFFFFF0
sub rsp,30
mov qword ptr ss:[rsp+20],rcx
add rcx,B14
mov dword ptr ss:[rsp+28],4
```

Write the shellcode in the .text section. Left: not infected. Right: infected

Step5. Finally the malware changes the permissions of the ".text" section back to "EXECUTABLE_READ" again.



```
jmp appvisvsubsystems64.7FFCC2DC3927
mov rdx,qword ptr ss:[rbp-48]
mov rax,qword ptr ss:[rbp-58]
mov rcx,qword ptr ds:[rax]
call appvisvsubsystems64.7FFCC2DCFB50
mov qword ptr ss:[rbp-60],rax
jmp appvisvsubsystems64.7FFCC2DC3940
lea rcx,qword ptr ss:[rbp-20]
call appvisvsubsystems64.7FFCC2DCA400
mov rax,qword ptr ss:[rbp-60]
mov qword ptr ss:[rbp-10],rax
mov dword ptr ss:[rbp-2C],47AB
lea rax,qword ptr ds:[7FFCC2DE605A]
mov c                  [rbp-38],rax
mov b                  bp-39],14
mov rax,qword ptr ss:[rbp-10]
mov r9,qword ptr ss:[rbp+50]
mov r8d,dword ptr ss:[rbp+14]
mov rdx,qword ptr ss:[rbp+8]
mov rcx,qword ptr ss:[rbp]
call rax
nop
```

```
rcx 00007FFCA1870000 appxpackaging.dll
    00007FFCA1871000  ".text"
    00007FFCA1958000  ".rdata"
    00007FFCA1982000  ".data"
    00007FFCA1984000  ".pdata"
    00007FFCA198E000  ".didat"
    00007FFCA198F000  ".rsrc"
    00007FFCA1A30000  ".reloc"

00007FFCC2DE605A:"fMlx6049EZJ7gZTVMuUuTNm5
[rbp-38]:"fMlx6049EZJ7gZTVMuUuTNm53E"

[rbp]:"&:g"
```

```
RAX   00007FFCD013BC70   <kernel32.VirtualProtect>
RBX   0000000000000000
RCX   00007FFCA1871000   "&:g"
RDX   00000000000BF125
RBP   000000DED79FE800   &"&:g"
RSP   000000DED79FE780
RSI   0000000000000000
RDI   0000000000000000

R8    0000000000000020
R9    000000DED79FEA4C
R10   00007FFCC2DC0000   appvisvsubsystems64.00007FFCC2DC0
R11   00007FFCC2DD4CB5   appvisvsubsystems64.00007FFCC2DD4
R12   0000000000000000
R13   0000000000000000
R14   0000000000000000
R15   0000000000000000
```

Change the .text section back to RX

Once the shellcode is loaded, the execution is identical to what can be seen in the QUARTERRIG report of CERT.PL, but with a different C2. In this case, the actors use the URL **hxxps:]]//pizzais.com/order.php**.

Timeline

As mentioned before, **Lab52 has observed the evolution in techniques since April**. Specifically, here we provide an additional hash of a different file published on VirusTotal that uses this injection technique, found during the writing of this post.

Hash	Description	Date
C71EC48A59631BFA3F33383C1F25719E95E5A80936D913AB3BFE2FEB172C1C5E	Notes.iso injecting the shellcode in the .txt section of the DLL	28/04/2023

Therefore, at the least in a previous registered case, the .iso file still bears the name “Note.iso,” which could indicate that **the new technique was already in use** in some samples of the previous campaign. However, the most recent observations show a change in their name to “Information”.

Short timeline – Changes in the injection technique

Conclusions

Just as QUARTERRIG was the evolution of HALFRIG, in this new campaign APT29 has modified the logic of its dll loader “Applvsubsystem64.dll” to make it more sophisticated.

The fact of using legitimate random DLLs for injection instead of the process memory itself, adds another layer of complexity to the way the shellcode is loaded.

A list of system candidates to be injected is provided in this post. However, it must be considered carefully because these are extracted of a specific system, following the previous steps. The analysts can follow the steps mentioned here in order to build their own script to get the list of DLL candidates to be used by the authors this new campaing.

Indicators of Compromise (IOC)

File	Hash
Information.iso	B422BA73F389AE5EF9411CF4484C840C7C82F2731C6324DB0B24B6F87CE8477D
Information .exe	6C55195F025FB895F9D0EC3EDBF58BC0AA46C43EEB246CFB88EEF1AE051171B3
AppvIsvSubsystems64.dll	E7C49758BAE63C83D251CACBFADA7C09AF0C3038E8FF755C4C04F916385805D8
dbg.info	5F6219ADE8E0577545B9F13AFD28F6D6E991326F3C427D671D1C1765164B0D57

C2
hxxps:]]//pizzais.com/order.php

Filesystem	Description
C:\Users\user\AppData\Local\MSOfficeUpdate\	AppvIsvSubsystems64.dll, Information .exe, dbg.info

Persistence	Value
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	C:\Users\user\AppData\Local\MSOfficeUpdate\Information .exe

Candidate system DLLs for injection

The following DLLs will pass the checks for this sample (size and other requirements). This list may vary on the target system depending on the version and the system software installed. Please check.

aadtb.dll	ActiveSyncProvider.dll	adtschema.dll
APMon.dll	appraiser.dll	AppXDeploymentExtensions.desktop.dll
AppXDeploymentExtensions.onecore.dll	AppXDeploymentServer.dll	AppxPackaging.dll
AudioEng.dll	AudioSes.dll	audiosrv.dll
AuthFWSnapin.dll	AuthFWSnapIn.Resources.dll	AzureSettingSyncProvider.dll
bcastdvruserservice.dll	BingMaps.dll	cavo2gui.dll
cdp.dll	cdprt.dll	CertEnroll.dll
Chakra.dll	cimwin32.dll	ClipSVC.dll
CloudExperienceHostCommon.dll	cmiv2.dll	comsvcs.dll
ConstraintIndex.Search.dll	ContentDeliveryManager.Utilities.dll	CoreShell.dll
CoreUIComponents.dll	crypt32.dll	d2d1.dll
d3d10.dll	d3d10warp.dll	d3d11.dll
D3D12Core.dll	d3d9.dll	D3DCompiler_47.dll
dbgeng.dll	dbghelp.dll	dcomp.dll
DeviceFlows.DataModel.dll	dfshim.dll	diagperf.dll
diagtrack.dll	directml.dll	DMWmiBridgeProv.dll
dosvc.dll	drvstore.dll	dui70.dll
dwmcore.dll	dwmscene.dll	DWrite.dll
dxilconv.dll	edgeangle.dll	EdgeContent.dll
edgehtml.dll	efscore.dll	EmailApis.dll
enterprisecsps.dll	esent.dll	ExplorerFrame.dll
FaceProcessor.dll	FaceRecognitionEngineAdapter.dll	fhuxpresentation.dll
FluencyDS.dll	FntCache.dll	FrameServer.dll
GdiPlus.dll	gpsvc.dll	HologramWorld.dll
Hydrogen.dll	icu.dll	ieframe.dll
inetcomm.dll	InputHost.dll	InputService.dll
InstallService.dll	IntelWifiIhv08.dll	ISM.dll
jscript9.dll	kerberos.dll	KernelBase.dll
libcrypto.dll	localspl.dll	LocationFramework.dll
lpasvc.dll	lsasrv.dll	MapGeocoder.dll
MapRouter.dll	MCRecvSrc.dll	MdmDiagnostics.dll
MemoryAnalyzer.dll	MessagingDataModel2.dll	mfasfsrcsnk.dll

mfc140.dll	mfc140u.dll	mfc42.dll
mfc42u.dll	mfcore.dll	MFMediaEngine.dll
mfmkvsrcsnk.dll	mfmp4srcsnk.dll	mfmpeg2srcsnk.dll
mfnetcore.dll	mfnetsrc.dll	mfperfhelper.dll
mfplat.dll	mfreadwrite.dll	mfsrcsnk.dll
mfsvr.dll	Microsoft.Bluetooth.Service.dll	Microsoft.Graphics.Display.DisplayEnhancementServ
migcore.dll	MiracastReceiver.dll	mispace.dll
mmcndmgr.dll	mmgaclient.dll	MSAJApi.dll
msctf.dll	msdtctm.dll	msftedit.dll
mshtml.dll	msmpeg2vdec.dll	MSPhotography.dll
mssrch.dll	mstscax.dll	MSTTSEngine_OneCore.dll
MSVidCtl.dll	msvproc.dll	msxml3.dll
msxml6.dll	MSxpsPCL6.dll	MSxpsPS.dll
mxdwdrv.dll	NetworkMobileSettings.dll	NotificationController.dll
ole32.dll	OpcServices.dll	opengl32.dll
PCPKsp.dll	perf_nt.dll	pidgenx.dll
pla.dll	PresentationNative_v0300.dll	PrintConfig.dll
PrintConfig.dll	PrintConfig.dll	PrintConfig.dll
qmgr.dll	quartz.dll	rasapi32.dll
rasmans.dll	rdpbase.dll	rdpcore.dll
rdpcorets.dll	rdpnano.dll	rdpserverbase.dll
rdpsharercom.dll	reseteng.dll	ResetEngine.dll
RP2DSN32.dll	rpcss.dll	rtmcodecs.dll
rtmpal.dll	rtmpltfm.dll	sapi.dll
sapi_onecore.dll	sbe.dll	sdengin2.dll
SettingsHandlers_nt.dll	setupapi.dll	SpeechPal.dll
sppobjs.dll	spsreng.dll	spsreng_onecore.dll
spwizimg.dll	SRH.dll	StartTileData.dll
storagewmi.dll	sysmain.dll	SystemSettings.Handlers.dll
SystemSettingsThresholdAdminFlowUI.dll	TaskFlowDataEngine.dll	termsrv.dll
TextInputMethodFormatter.dll	TokenBroker.dll	TpmCoreProvisioning.dll
tquery.dll	tsf3gip.dll	twinapi.appcore.dll
twinui.dll	twinui.pcshell.dll	uDWM.dll
UIAutomationCore.dll	UIRibbon.dll	UIRibbonRes.dll
Unistore.dll	UpdateAgent.dll	urlmon.dll
usbmon.dll	UserDataService.dll	usermgr.dll
VBoxDispD3D.dll	VBoxDispD3D.dll	VBoxGL.dll
VBoxGL.dll	VBoxGL-x86.dll	VBoxMRXNP.dll
VBoxNine.dll	VBoxNine.dll	VBoxSVGA.dll
VBoxSVGA.dll	VBoxSVGA-x86.dll	vo28gui.dll

vssapi.dll	wbemcore.dll	webplatstorageserver.dll
WebRuntimeManager.dll	webservices.dll	wevtsvc.dll
win32spl.dll	WindowManagement.dll	Windows.AI.MachineLearning.dll
Windows.ApplicationModel.Store.dll	Windows.CloudStore.dll	WindowsCodecs.dll
WindowsCodecsRaw.dll	Windows.Data.Pdf.dll	Windows.Devices.Bluetooth.dll
Windows.Devices.Perception.dll	Windows.Devices.PointOfService.dll	Windows.Globalization.dll
Windows.Graphics.Printing.3D.dll	Windows.Graphics.Printing.Workflow.dll	Windows.Internal.Signals.dll
WindowsInternal.Xaml.Controls.Tabs.dll	Windows.Media.dll	Windows.Media.Editing.dll
Windows.Media.Protection.PlayReady.dll	Windows.Media.Speech.dll	Windows.Media.Streaming.dll
Windows.Mirage.dll	Windows.Networking.BackgroundTransfer.dll	Windows.Security.Authentication.Web.Core.dll
Windows.StateRepository.dll	windows.storage.dll	windowsudk.shellcommon.dll
Windows.UI.Cred.dll	Windows.UI.Immersive.dll	Windows.UI.Input.Inking.Analysis.dll
Windows.UI.Input.Inking.dll	Windows.UI.Logon.dll	Windows.UI.Shell.Internal.AdaptiveCards.dll
Windows.UI.Xaml.Controls.dll	Windows.UI.Xaml.dll	Windows.UI.Xaml.Maps.dll
Windows.UI.Xaml.Phone.dll	Windows.Web.Http.dll	winhttp.dll
wininet.dll	winmde.dll	winmsipc.dll
winsetup.dll	wlansvc.dll	wlidsvc.dll
WMNetMgr.dll	wmp.dll	workfolderssvc.dll
WpcDesktopMonSvc.dll	Wpc.dll	wpncore.dll
WsmSvc.dll	wsp_fs.dll	wsp_health.dll
wuaueng.dll	wwansvc.dll	XblAuthManager.dll
XblGameSave.dll	XboxNetApiSvc.dll	XpsPrint.dll
xpsservices.dll		