Zloader Strikes Back

labs.k7computing.com/index.php/zloader-strikes-back/

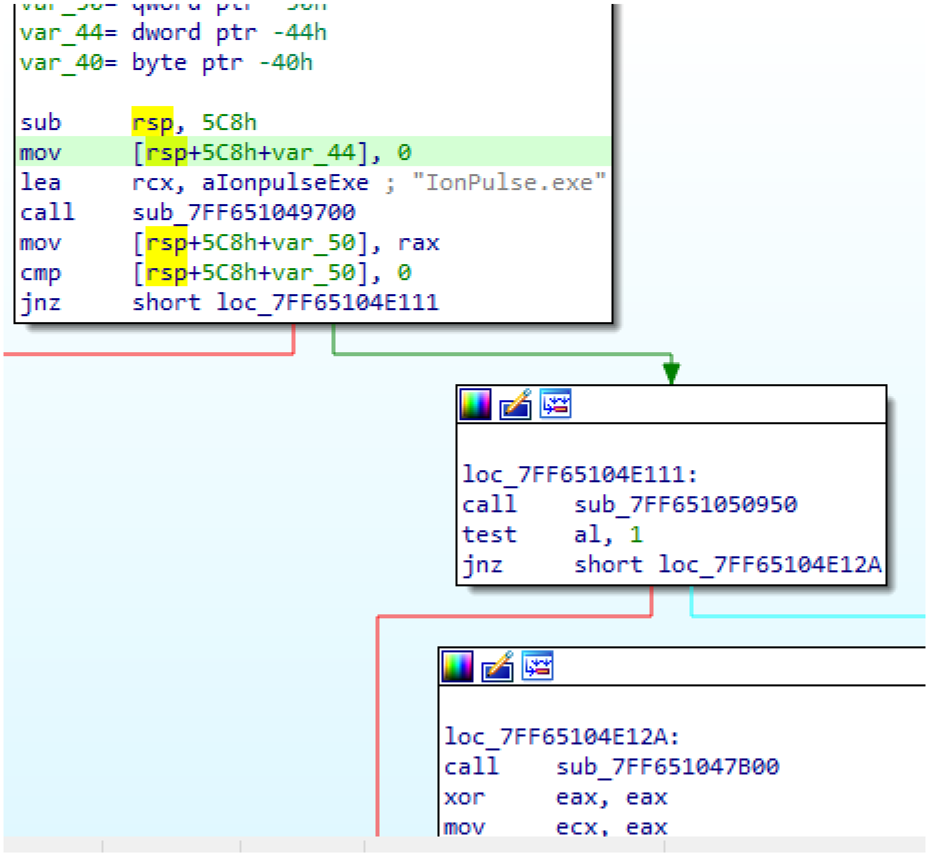
By Sudeep

February 14, 2024

Recently, we came across an update from PolySwarm regarding a new Variant of Zloader. Zloader is a malware based on Zeus, which has been targeting financial institutions and its customers. This blog gets into the nuances of the new techniques used by Zloader.

Technical Analysis

It was observed that Zloader had very few Import functions and it was obfuscated and threat actors were making sure that Zloader only runs with the filename “IonPulse.exe”.



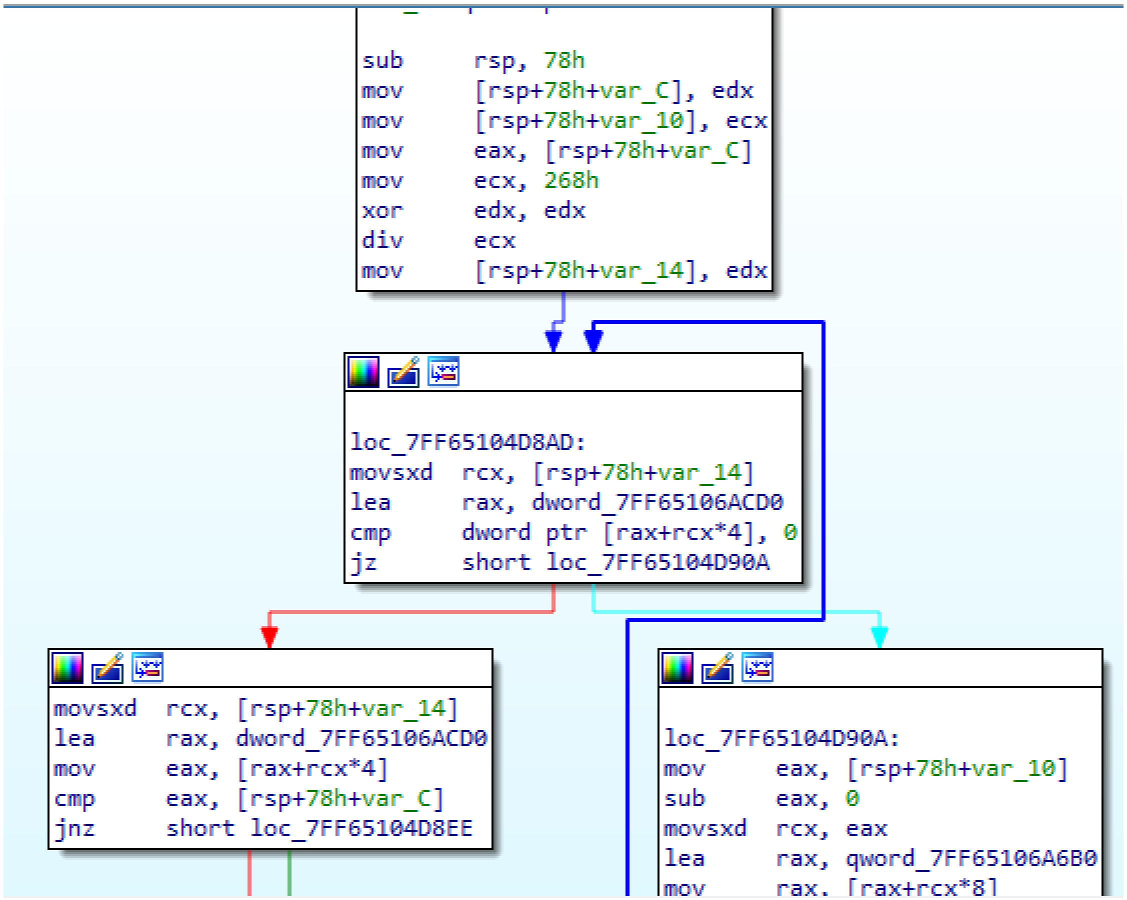
```
var_50= qword ptr -50h
var_44= dword ptr -44h
var_40= byte ptr -40h

sub     rsp, 5C8h
mov     [rsp+5C8h+var_44], 0
lea     rcx, aIonpulseExe ; "IonPulse.exe"
call    sub_7FF651049700
mov     [rsp+5C8h+var_50], rax
cmp     [rsp+5C8h+var_50], 0
jnz     short loc_7FF65104E111
```

```
loc_7FF65104E111:
call    sub_7FF651050950
test    al, 1
jnz     short loc_7FF65104E12A
```

```
loc_7FF65104E12A:
call    sub_7FF651047B00
xor     eax, eax
mov     ecx, eax
```

Figure 1: Precheck before running

Once it checks that the name is IonPulse.exe, it gets the handle of Ntdll.dll using CreateFileA.



```
sub     rsp, 78h
mov     [rsp+78h+var_C], edx
mov     [rsp+78h+var_10], ecx
mov     eax, [rsp+78h+var_C]
mov     ecx, 268h
xor     edx, edx
div     ecx
mov     [rsp+78h+var_14], edx
```

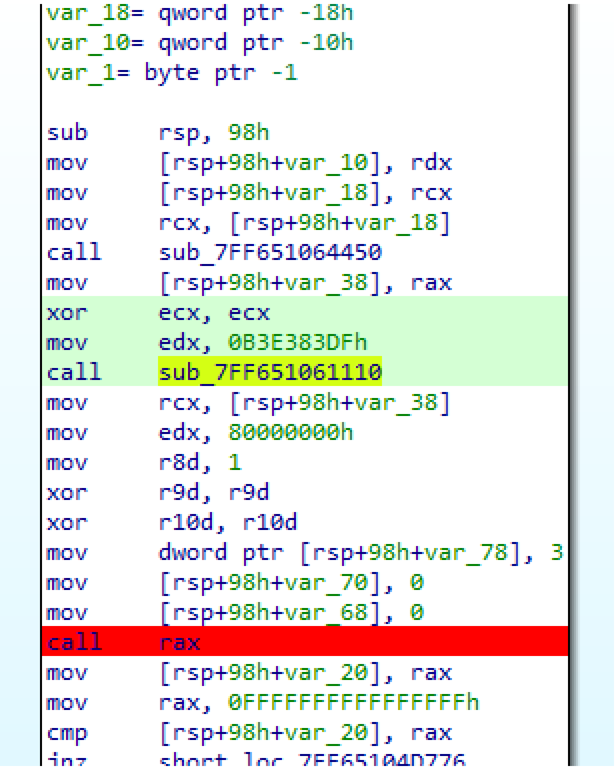
```
loc_7FF65104D8AD:
movsxd  rcx, [rsp+78h+var_14]
lea     rax, dword_7FF65106ACD0
cmp     dword ptr [rax+rcx*4], 0
jz      short loc_7FF65104D90A
```

```
movsxd  rcx, [rsp+78h+var_14]
lea     rax, dword_7FF65106ACD0
mov     eax, [rax+rcx*4]
cmp     eax, [rsp+78h+var_C]
jnz     short loc_7FF65104D8EE
```

```
loc_7FF65104D90A:
mov     eax, [rsp+78h+var_10]
sub     eax, 0
movsxd  rcx, eax
lea     rax, qword_7FF65106A6B0
mov     rax, [rax+rcx*8]
```

Figure 2: Mapping API with hashes

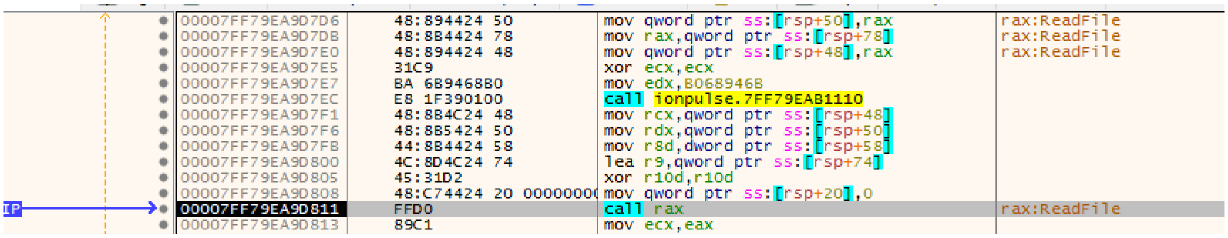
It is making use of the above mentioned Function in Figure 2 to resolve the API.



```
var_18= qword ptr -18h
var_10= qword ptr -10h
var_1= byte ptr -1

sub     rsp, 98h
mov     [rsp+98h+var_10], rdx
mov     [rsp+98h+var_18], rcx
mov     rcx, [rsp+98h+var_18]
call    sub_7FF651064450
mov     [rsp+98h+var_38], rax
xor     ecx, ecx
mov     edx, 0B3E383DFh
call    sub_7FF651061110
mov     rcx, [rsp+98h+var_38]
mov     edx, 80000000h
mov     r8d, 1
xor     r9d, r9d
xor     r10d, r10d
mov     dword ptr [rsp+98h+var_78], 3
mov     [rsp+98h+var_70], 0
mov     [rsp+98h+var_68], 0
call    rax
mov     [rsp+98h+var_20], rax
mov     rax, 0FFFFFFFFFFFFFFFFh
cmp     [rsp+98h+var_20], rax
jnz     short loc_7FF65104D776
```

Figure 3: CreateFileA

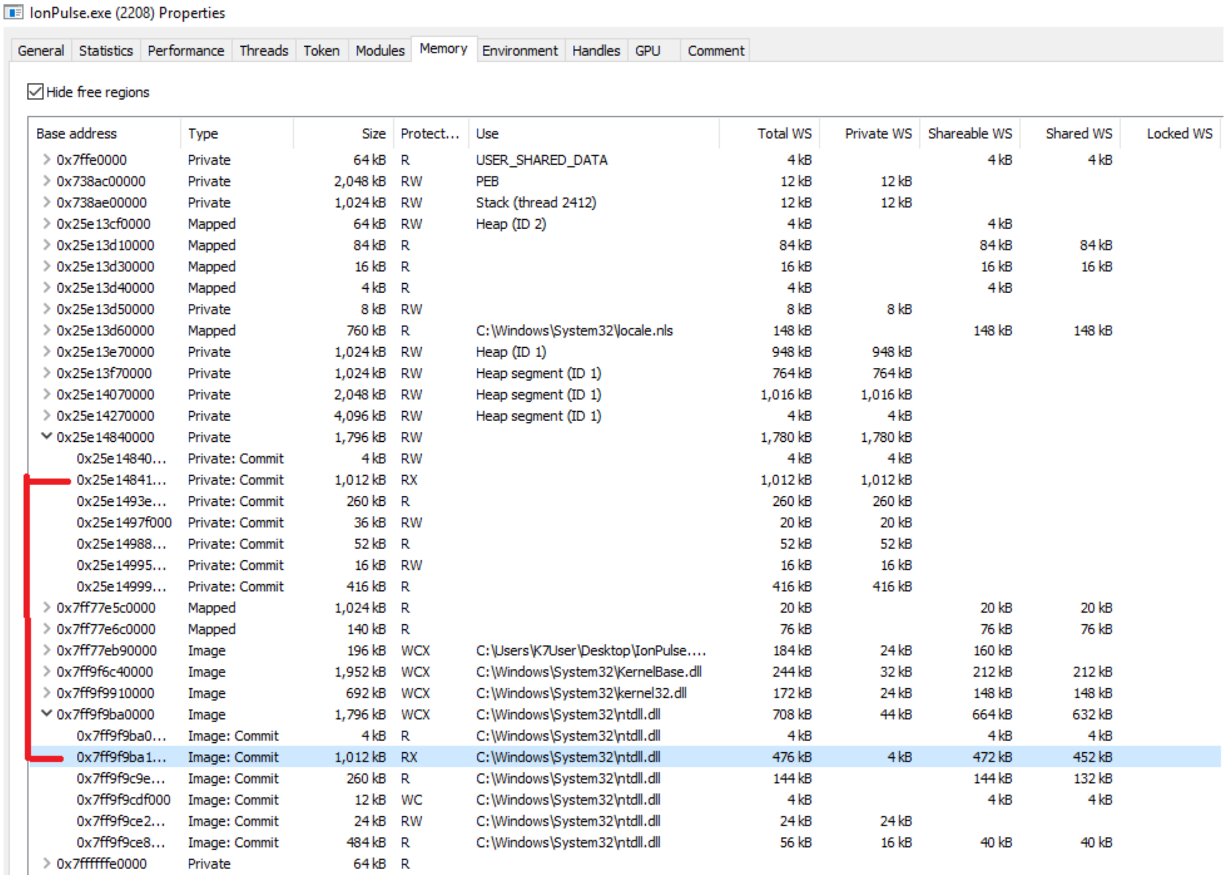
It gets the handle of Ntdll.dll using CreateFileA.



```
00007FF79EA9D7D6   48:894424 50            mov qword ptr ss:[rsp+50],rax          rax:ReadFile
00007FF79EA9D7DB   48:8B4424 78            mov rax,qword ptr ss:[rsp+78]          rax:ReadFile
00007FF79EA9D7E0   48:894424 48            mov qword ptr ss:[rsp+48],rax          rax:ReadFile
00007FF79EA9D7E5   31C9                    xor ecx,ecx
00007FF79EA9D7E7   BA 6B9468B0             mov edx,B068946B
00007FF79EA9D7EC   E8 1F390100             call ionpulse.7FF79EAB1110
00007FF79EA9D7F1   48:8B4C24 48            mov rcx,qword ptr ss:[rsp+48]
00007FF79EA9D7F6   48:8B5424 50            mov rdx,qword ptr ss:[rsp+50]
00007FF79EA9D7FB   44:8B4424 58            mov r8d,dword ptr ss:[rsp+58]
00007FF79EA9D800   4C:8D4C24 74            lea r9,qword ptr ss:[rsp+74]
00007FF79EA9D805   45:31D2                 xor r10d,r10d
00007FF79EA9D808   48:C74424 20 0000000(   mov qword ptr ss:[rsp+20],0
IP → 00007FF79EA9D811   FFD0               call rax                               rax:ReadFile
00007FF79EA9D813   89C1                    mov ecx,eax
```

Figure 4: Reading ntdll

Then uses ReadFile to copy the contents of Ntdll.dll. Before doing that it allocates memory using VirtualAlloc.



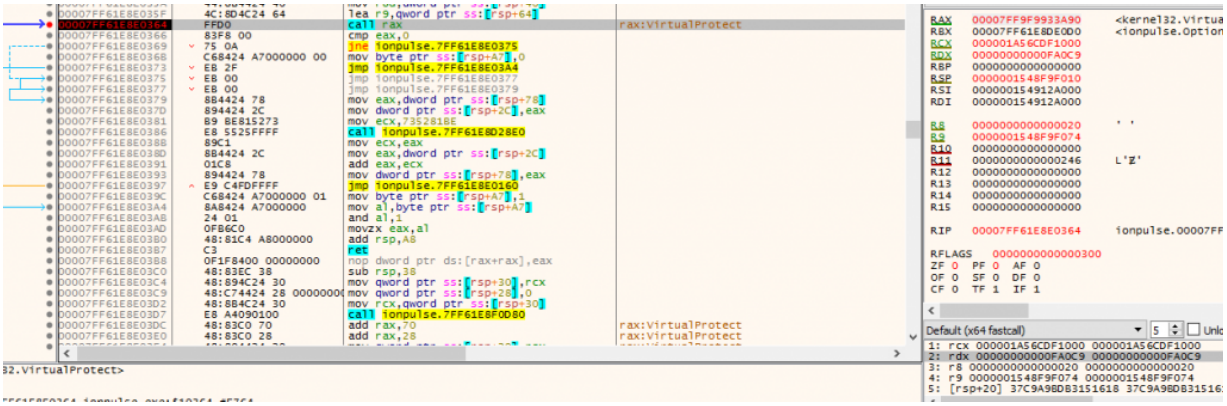
IonPulse.exe (2208) Properties

☑ Hide free regions

Base address	Type	Size	Protect...	Use	Total WS	Private WS	Shareable WS	Shared WS	Locked WS
> 0x7ffe0000	Private	64 kB	R	USER_SHARED_DATA	4 kB		4 kB	4 kB	
> 0x738ac00000	Private	2,048 kB	RW	PEB	12 kB	12 kB			
> 0x738ae00000	Private	1,024 kB	RW	Stack (thread 2412)	12 kB	12 kB			
> 0x25e13cf0000	Mapped	64 kB	RW	Heap (ID 2)	4 kB		4 kB		
> 0x25e13d10000	Mapped	84 kB	R		84 kB		84 kB	84 kB	
> 0x25e13d30000	Mapped	16 kB	R		16 kB		16 kB	16 kB	
> 0x25e13d40000	Mapped	4 kB	R		4 kB		4 kB		
> 0x25e13d50000	Private	8 kB	RW		8 kB	8 kB			
> 0x25e13d60000	Mapped	760 kB	R	C:\Windows\System32\locale.nls	148 kB		148 kB	148 kB	
> 0x25e13e70000	Private	1,024 kB	RW	Heap (ID 1)	948 kB	948 kB			
> 0x25e13f70000	Private	1,024 kB	RW	Heap segment (ID 1)	764 kB	764 kB			
> 0x25e14070000	Private	2,048 kB	RW	Heap segment (ID 1)	1,016 kB	1,016 kB			
> 0x25e14270000	Private	4,096 kB	RW	Heap segment (ID 1)	4 kB	4 kB			
⌄ 0x25e14840000	Private	1,796 kB	RW		1,780 kB	1,780 kB			
0x25e14840...	Private: Commit	4 kB	RW		4 kB	4 kB			
0x25e14841...	Private: Commit	1,012 kB	RX		1,012 kB	1,012 kB			
0x25e1493e...	Private: Commit	260 kB	R		260 kB	260 kB			
0x25e1497f000	Private: Commit	36 kB	RW		20 kB	20 kB			
0x25e14988...	Private: Commit	52 kB	R		52 kB	52 kB			
0x25e14995...	Private: Commit	16 kB	RW		16 kB	16 kB			
0x25e14999...	Private: Commit	416 kB	R		416 kB	416 kB			
> 0x7ff77e5c0000	Mapped	1,024 kB	R		20 kB		20 kB	20 kB	
> 0x7ff77e6c0000	Mapped	140 kB	R		76 kB		76 kB	76 kB	
> 0x7ff77eb90000	Image	196 kB	WCX	C:\Users\K7User\Desktop\IonPulse....	184 kB	24 kB	160 kB		
> 0x7ff9f6c40000	Image	1,952 kB	WCX	C:\Windows\System32\KernelBase.dll	244 kB	32 kB	212 kB	212 kB	
> 0x7ff9f9910000	Image	692 kB	WCX	C:\Windows\System32\kernel32.dll	172 kB	24 kB	148 kB	148 kB	
⌄ 0x7ff9f9ba0000	Image	1,796 kB	WCX	C:\Windows\System32\ntdll.dll	708 kB	44 kB	664 kB	632 kB	
0x7ff9f9ba0...	Image: Commit	4 kB	R	C:\Windows\System32\ntdll.dll	4 kB		4 kB	4 kB	
0x7ff9f9ba1...	Image: Commit	1,012 kB	RX	C:\Windows\System32\ntdll.dll	476 kB	4 kB	472 kB	452 kB	
0x7ff9f9c9e...	Image: Commit	260 kB	R	C:\Windows\System32\ntdll.dll	144 kB		144 kB	132 kB	
0x7ff9f9cdf000	Image: Commit	12 kB	WC	C:\Windows\System32\ntdll.dll	4 kB		4 kB	4 kB	
0x7ff9f9ce2...	Image: Commit	24 kB	RW	C:\Windows\System32\ntdll.dll	24 kB	24 kB			
0x7ff9f9ce8...	Image: Commit	484 kB	R	C:\Windows\System32\ntdll.dll	56 kB	16 kB	40 kB	40 kB	
> 0x7ffffffe0000	Private	64 kB	R						

Figure 5: Ntdll.dll copied

Above figure shows the copied content of Ntdll.dll.

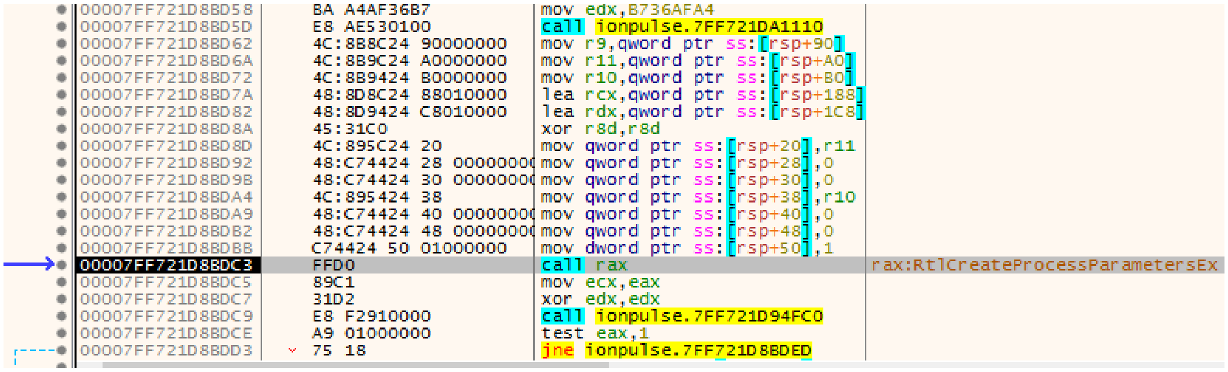


```
00007FF61E8E035F   4C:8D4C24 64              lea r9,qword ptr ss:[rsp+64]
00007FF61E8E0364   FFD0                      call rax                              rax:VirtualProtect
00007FF61E8E0366   83F8 00                   cmp eax,0
00007FF61E8E0369 ˅ 75 0A                     jne ionpulse.7FF61E8E0375
00007FF61E8E036B   C68424 A7000000 00        mov byte ptr ss:[rsp+A7],0
00007FF61E8E0373 ˅ EB 2F                     jmp ionpulse.7FF61E8E03A4
00007FF61E8E0375 ˅ EB 00                     jmp ionpulse.7FF61E8E0377
00007FF61E8E0377 ˅ EB 00                     jmp ionpulse.7FF61E8E0379
00007FF61E8E0379   8B4424 78                 mov eax,dword ptr ss:[rsp+78]
00007FF61E8E037D   894424 2C                 mov dword ptr ss:[rsp+2C],eax
00007FF61E8E0381   B9 BE815273               mov ecx,735281BE
00007FF61E8E0386   E8 5525FFFF               call ionpulse.7FF61E8D28E0
00007FF61E8E038B   89C1                      mov ecx,eax
00007FF61E8E038D   8B4424 2C                 mov eax,dword ptr ss:[rsp+2C]
00007FF61E8E0391   01C8                      add eax,ecx
00007FF61E8E0393   894424 78                 mov dword ptr ss:[rsp+78],eax
00007FF61E8E0397 ˄ E9 C4FDFFFF               jmp ionpulse.7FF61E8E0160
00007FF61E8E039C   C68424 A7000000 01        mov byte ptr ss:[rsp+A7],1
00007FF61E8E03A4   8A8424 A7000000           mov al,byte ptr ss:[rsp+A7]
00007FF61E8E03AB   24 01                     and al,1
00007FF61E8E03AD   0FB6C0                    movzx eax,al
00007FF61E8E03B0   48:81C4 A8000000          add rsp,A8
00007FF61E8E03B7   C3                        ret
00007FF61E8E03B8   0F1F8400 00000000         nop dword ptr ds:[rax+rax],eax
00007FF61E8E03C0   48:83EC 38                sub rsp,38
00007FF61E8E03C4   48:894C24 30              mov qword ptr ss:[rsp+30],rcx
00007FF61E8E03C9   48:C74424 28 0000000(     mov qword ptr ss:[rsp+28],0
00007FF61E8E03D2   48:8B4C24 30              mov rcx,qword ptr ss:[rsp+30]
00007FF61E8E03D7   E8 A4090100               call ionpulse.7FF61E8F0D80
00007FF61E8E03DC   48:83C0 70                add rax,70                            rax:VirtualProtect
00007FF61E8E03E0   48:83C0 28                add rax,28                            rax:VirtualProtect

32.VirtualProtect>
```

```
RAX     00007FF9F9933A90      <kernel32.Virtua
RBX     00007FF61E8DE0D0      <ionpulse.Option
RCX     000001A56CDF1000
RDX     00000000000FA0C9
RBP     0000000000000000
RSP     000001548F9F010
RSI     00000154912A000
RDI     00000154912A000

R8      0000000000000020      ' '
R9      000001548F9F074
R10     0000000000000000
R11     0000000000000246      L'Ɇ'
R12     0000000000000000
R13     0000000000000000
R14     0000000000000000
R15     0000000000000000

RIP     00007FF61E8E0364      ionpulse.00007FF

RFLAGS     0000000000000300
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
CF 0   TF 1   IF 1

Default (x64 fastcall)   5   Unlo
1: rcx 000001A56CDF1000 000001A56CDF1000
2: rdx 00000000000FA0C9 00000000000FA0C9
3: r8 0000000000000020 0000000000000020
4: r9 000001548F9F074 000001548F9F074
5: [rsp+20] 37C9A9BDB3151618 37C9A9BDB31516
```

Figure 6: VirtualProtect

After copying Ntdll.dll it is using VirtualProtect to change the memory protection accordingly.



```
00007FF721D8BD58   BA A4AF36B7               mov edx,B736AFA4
00007FF721D8BD5D   E8 AE530100               call ionpulse.7FF721DA1110
00007FF721D8BD62   4C:8B8C24 90000000        mov r9,qword ptr ss:[rsp+90]
00007FF721D8BD6A   4C:8B9C24 A0000000        mov r11,qword ptr ss:[rsp+A0]
00007FF721D8BD72   4C:8B9424 B0000000        mov r10,qword ptr ss:[rsp+B0]
00007FF721D8BD7A   48:8D8C24 88010000        lea rcx,qword ptr ss:[rsp+188]
00007FF721D8BD82   48:8D9424 C8010000        lea rdx,qword ptr ss:[rsp+1C8]
00007FF721D8BD8A   45:31C0                   xor r8d,r8d
00007FF721D8BD8D   4C:895C24 20              mov qword ptr ss:[rsp+20],r11
00007FF721D8BD92   48:C74424 28 00000000     mov qword ptr ss:[rsp+28],0
00007FF721D8BD9B   48:C74424 30 00000000     mov qword ptr ss:[rsp+30],0
00007FF721D8BDA4   4C:895424 38              mov qword ptr ss:[rsp+38],r10
00007FF721D8BDA9   48:C74424 40 00000000     mov qword ptr ss:[rsp+40],0
00007FF721D8BDB2   48:C74424 48 00000000     mov qword ptr ss:[rsp+48],0
00007FF721D8BDBB   C74424 50 01000000        mov dword ptr ss:[rsp+50],1
00007FF721D8BDC3   FFD0                      call rax                    rax:RtlCreateProcessParametersEx
00007FF721D8BDC5   89C1                      mov ecx,eax
00007FF721D8BDC7   31D2                      xor edx,edx
00007FF721D8BDC9   E8 F2910000               call ionpulse.7FF721D94FC0
00007FF721D8BDCE   A9 01000000               test eax,1
00007FF721D8BDD3 ˅ 75 18                     jne ionpulse.7FF721D8BDED
```

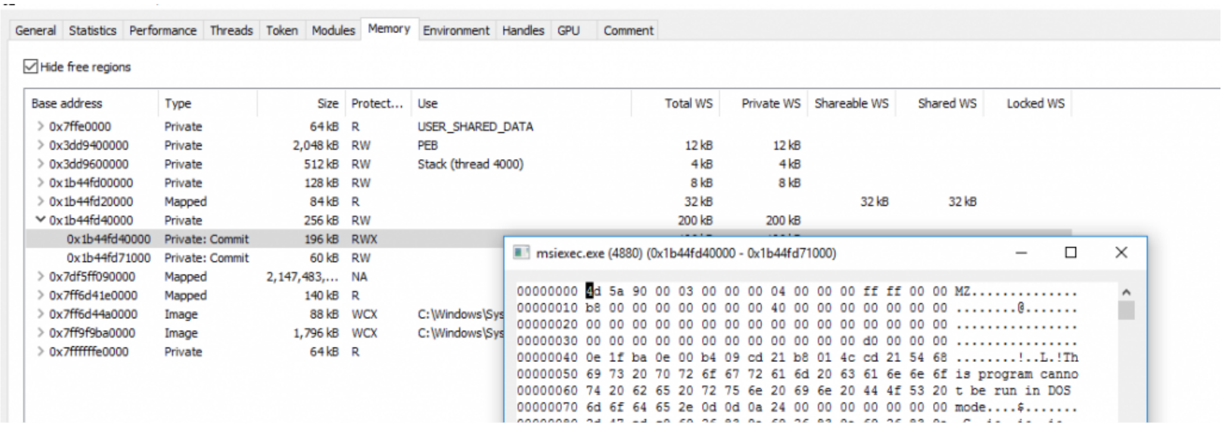
Figure 7: Creating msiexec.exe

It is making use of RtlInitUnicodeString, RtlCreateProcessParametersEx to create a structure which can be used by NtCreateUserProcess later. Then it make use of Associated syscall to NtCreateUserProcess to run msiexec.exe.

e.exe - Thread: Main Thread 464 - x64dbg
ourites
Options
Help
Jun 15 2023 (TitanEngine)

Figure 8: Syscall

It was making use of Syscall to Write into msiexec.exe and had allocated memory before doing that. This syscall is related to NtWriteVirtualMemory which is Similar to WriteProcessMemory in WinAPI.



☑ Hide free regions

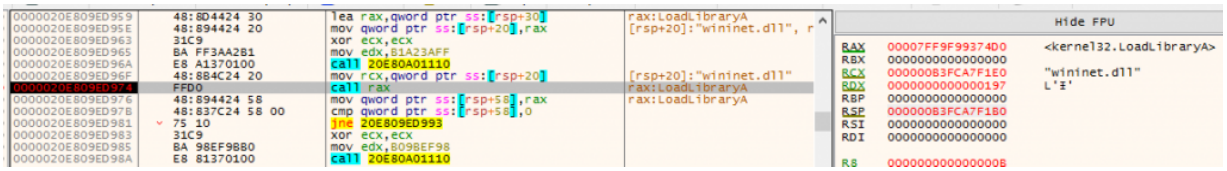
Base address	Type	Size	Protect...	Use	Total WS	Private WS	Shareable WS	Shared WS	Locked WS
0x7ffe0000	Private	64 kB	R	USER_SHARED_DATA					
0x3dd9400000	Private	2,048 kB	RW	PEB	12 kB	12 kB			
0x3dd9600000	Private	512 kB	RW	Stack (thread 4000)	4 kB	4 kB			
0x1b44fd00000	Private	128 kB	RW		8 kB	8 kB			
0x1b44fd20000	Mapped	84 kB	R		32 kB		32 kB	32 kB	
0x1b44fd40000	Private	256 kB	RW		200 kB	200 kB			
0x1b44fd40000	Private: Commit	196 kB	RWX						
0x1b44fd71000	Private: Commit	60 kB	RW						
0x7df5ff090000	Mapped	2,147,483,...	NA						
0x7ff6d41e0000	Mapped	140 kB	R						
0x7ff6d44a0000	Image	88 kB	WCX	C:\Windows\Sys					
0x7ff9f9ba0000	Image	1,796 kB	WCX	C:\Windows\Sys					
0x7ffffffe0000	Private	64 kB	R						

msiexec.exe (4880) (0x1b44fd40000 - 0x1b44fd71000)

```
00000000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ..............
00000010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ........@.......
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000030 00 00 00 00 00 00 00 00 00 00 00 00 d0 00 00 00 ................
00000040 0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68 ........!..L.!Th
00000050 69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f is program canno
00000060 74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20 t be run in DOS 
00000070 6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00 mode....$.......
```

Figure 9: Zloader injected in msiexec.exe

Then makes use of another syscall to the adjacent function of NtProtectVirtualMemory, to change its memory protection to ‘Execute’. Along with that it will use Syscall associated with NtGetContextThread, NtSetContextThread and NtResumeThread. Doing this it is hijacking the Thread.

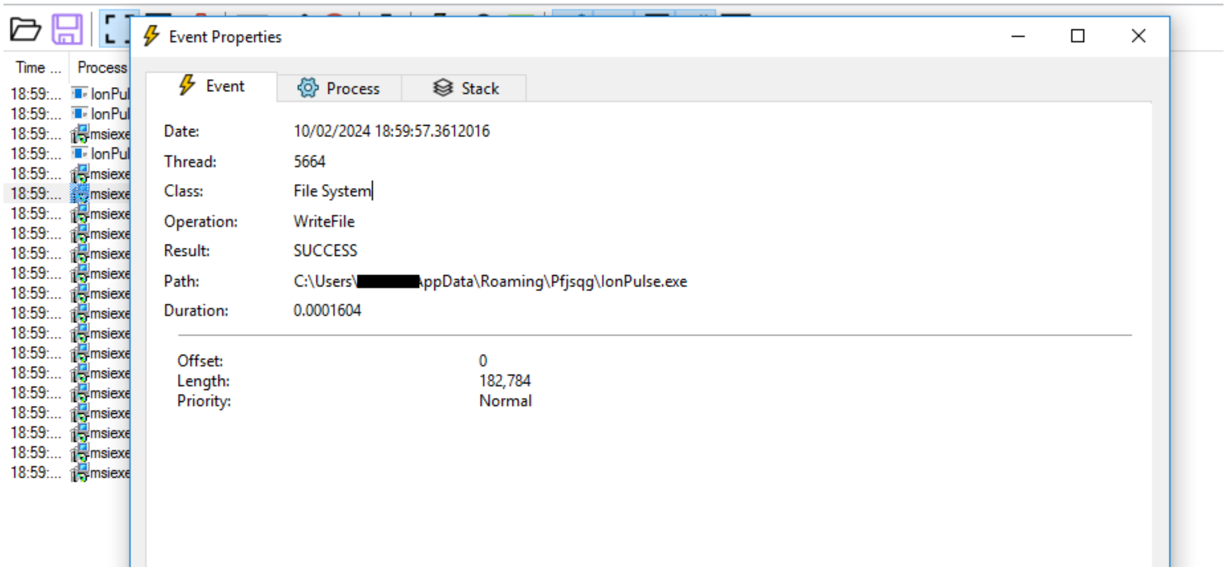


```
0000020E809ED959   48:8D4424 30       lea rax,qword ptr ss:[rsp+30]      rax:LoadLibraryA
0000020E809ED95E   48:894424 20       mov qword ptr ss:[rsp+20],rax      [rsp+20]:"wininet.dll", r
0000020E809ED963   31C9               xor ecx,ecx
0000020E809ED965   BA FF3AA2B1        mov edx,B1A23AFF
0000020E809ED96A   E8 A1370100        call 20E80A01110
0000020E809ED96F   48:8B4C24 20       mov rcx,qword ptr ss:[rsp+20]      [rsp+20]:"wininet.dll"
0000020E809ED974   FFD0               call rax                           rax:LoadLibraryA
0000020E809ED976   48:894424 58       mov qword ptr ss:[rsp+58],rax      rax:LoadLibraryA
0000020E809ED97B   48:837C24 58 00    cmp qword ptr ss:[rsp+58],0
0000020E809ED981   75 10              jne 20E809ED993
0000020E809ED983   31C9               xor ecx,ecx
0000020E809ED985   BA 98EF9BB0        mov edx,B09BEF98
0000020E809ED98A   E8 81370100        call 20E80A01110
```

```
RAX   00007FF9F99374D0   <kernel32.LoadLibraryA>
RBX   0000000000000000
RCX   000000B3FCA7F1E0   "wininet.dll"
RDX   0000000000000197   L'Ɨ'
RBP   0000000000000000
RSP   000000B3FCA7F1B0
RSI   0000000000000000
RDI   0000000000000000

R8    000000000000000B
```

Figure 10: Loading wininet.dll

It will then load wininet.dll and ws2_32.dll using LoadLibraryA to connect to C2.

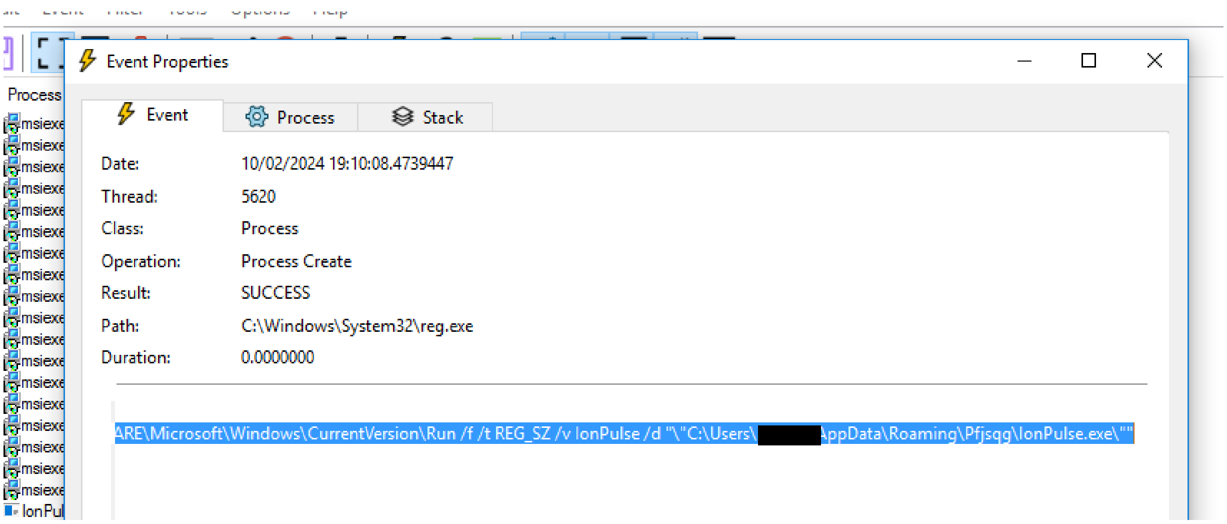


Date: 10/02/2024 18:59:57.3612016
Thread: 5664
Class: File System
Operation: WriteFile
Result: SUCCESS
Path: C:\Users\[redacted]ppData\Roaming\Pfjsqg\IonPulse.exe
Duration: 0.0001604

Offset: 0
Length: 182,784
Priority: Normal

Figure 11: Self Copy

It will then make a self Copy in AppData\Roaming



Date: 10/02/2024 19:10:08.4739447
Thread: 5620
Class: Process
Operation: Process Create
Result: SUCCESS
Path: C:\Windows\System32\reg.exe
Duration: 0.0000000

```
ARE\Microsoft\Windows\CurrentVersion\Run /f /t REG_SZ /v IonPulse /d "\"C:\Users\[redacted]ppData\Roaming\Pfjsqg\IonPulse.exe\""
```

Figure 12: Run Entry

Persistence is ensured through the Run registry and msiexec.exe starts connecting to C2 and then IonPulse.exe exits.

By this we can see that Zloader has started using Syscall for evasion, along with loading new Ntdll.dll.

We at K7 Labs provide detection for Zloader and all the latest threats. Users are advised to use a reliable security product such as “K7 Total Security” and keep it up-to-date to safeguard their devices.

Indicators of Compromise (IOCs)

FileName	Hash	Detection Name
IonPulse.exe	71C72AD0DA3AF2FCA53A729EF977F344	Trojan (005afb2c1)

References

https://www.zscaler.com/blogs/security-research/zloader-no-longer-silent-night

https://captmeelo.com/redteam/maldev/2022/05/10/ntcreateuserprocess.html