# Zloader Strikes Back

Published: 2024-02-14 · Archived: 2026-04-05 13:01:37 UTC

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”.

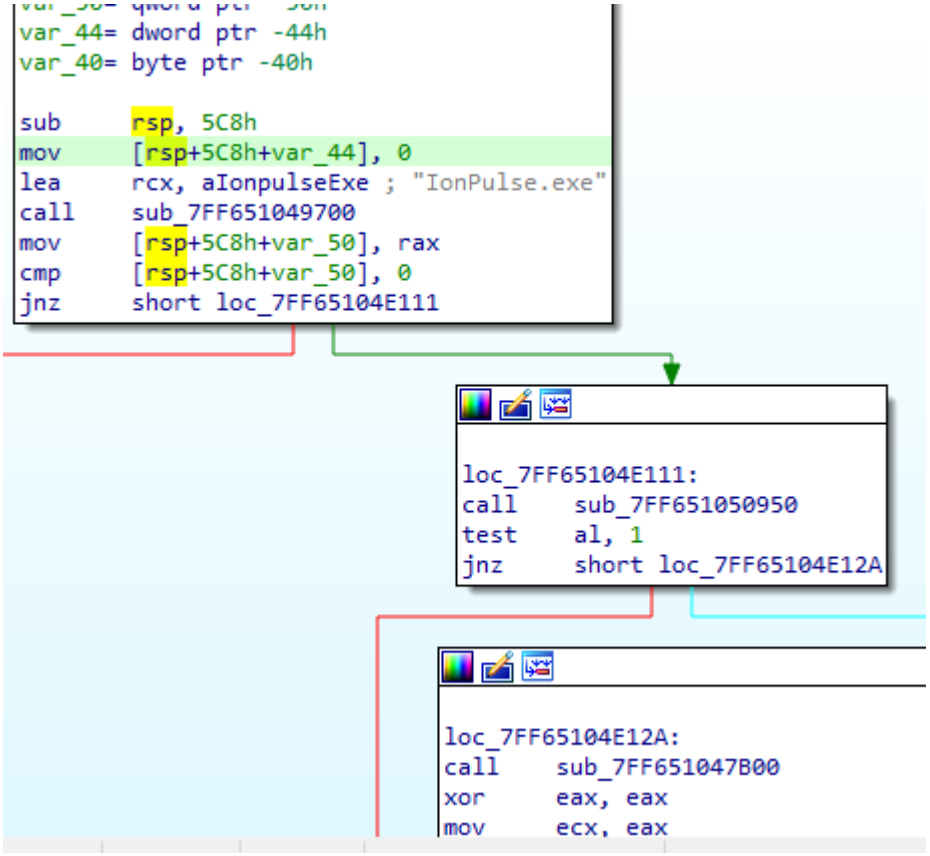


Figure 1: Precheck before running

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

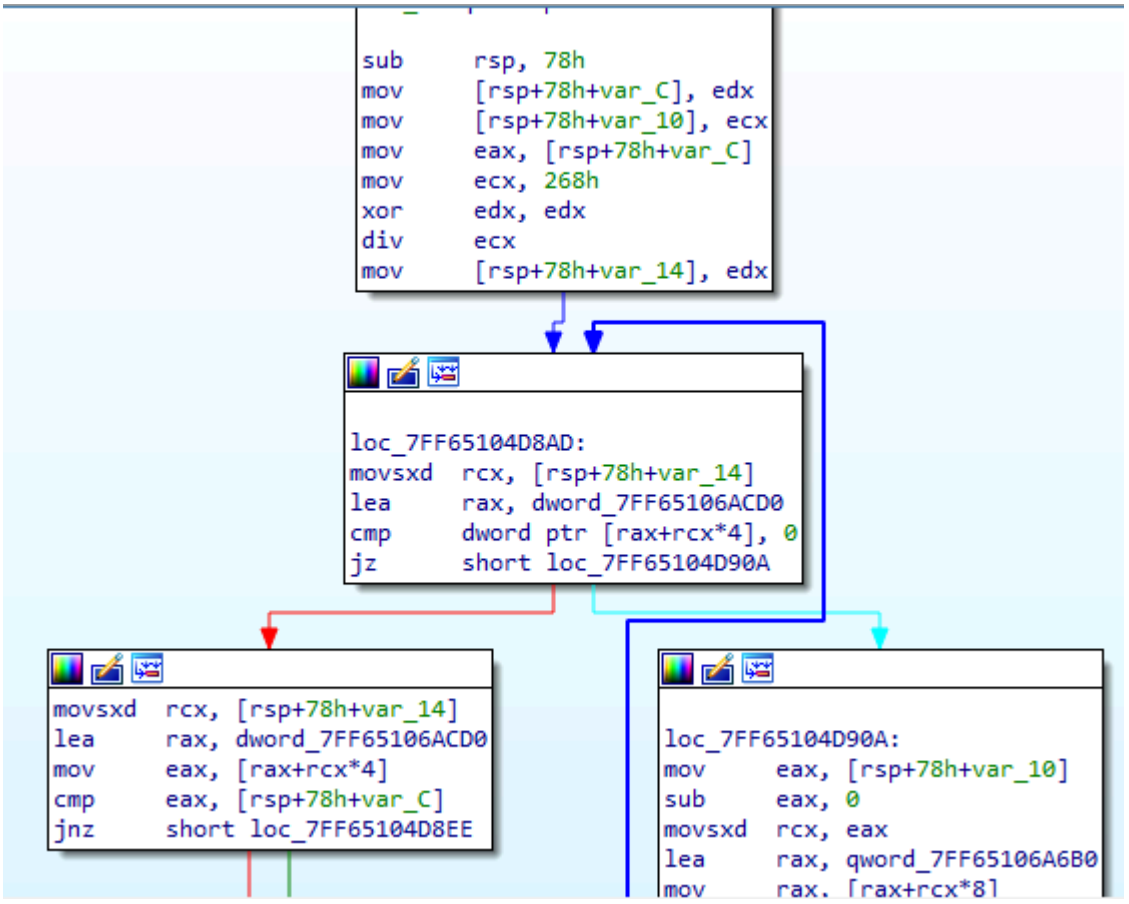


Figure 2: Mapping API with hashes

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

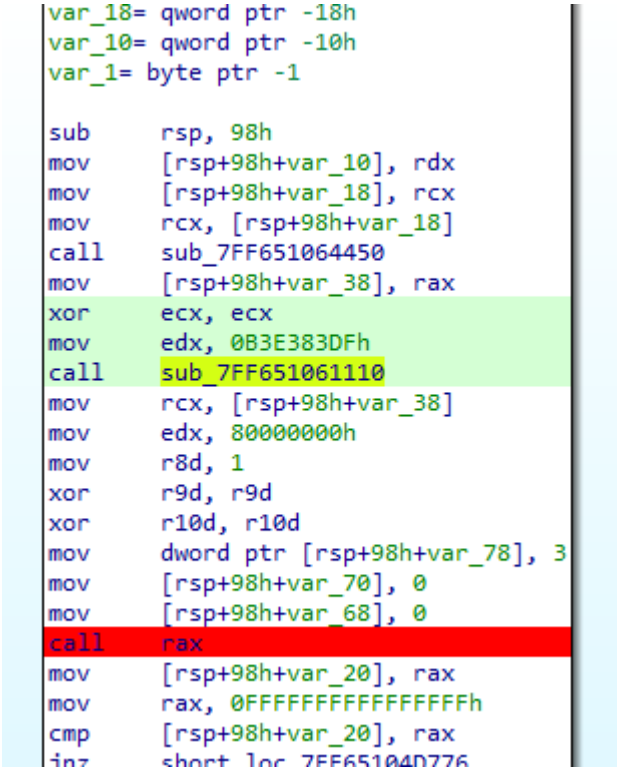


Figure 3: CreateFileA

It gets the handle of Ntdll.dll using CreateFileA.

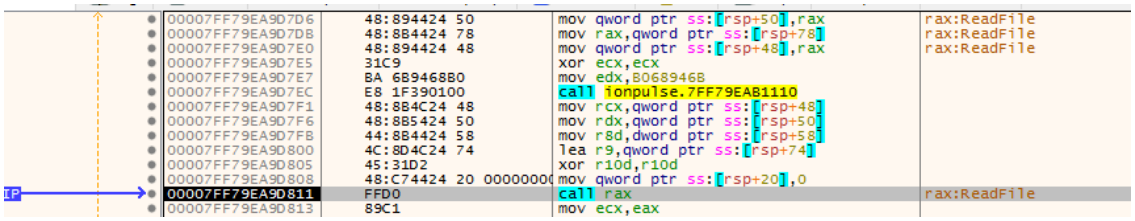


Figure 4: Reading ntdll

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

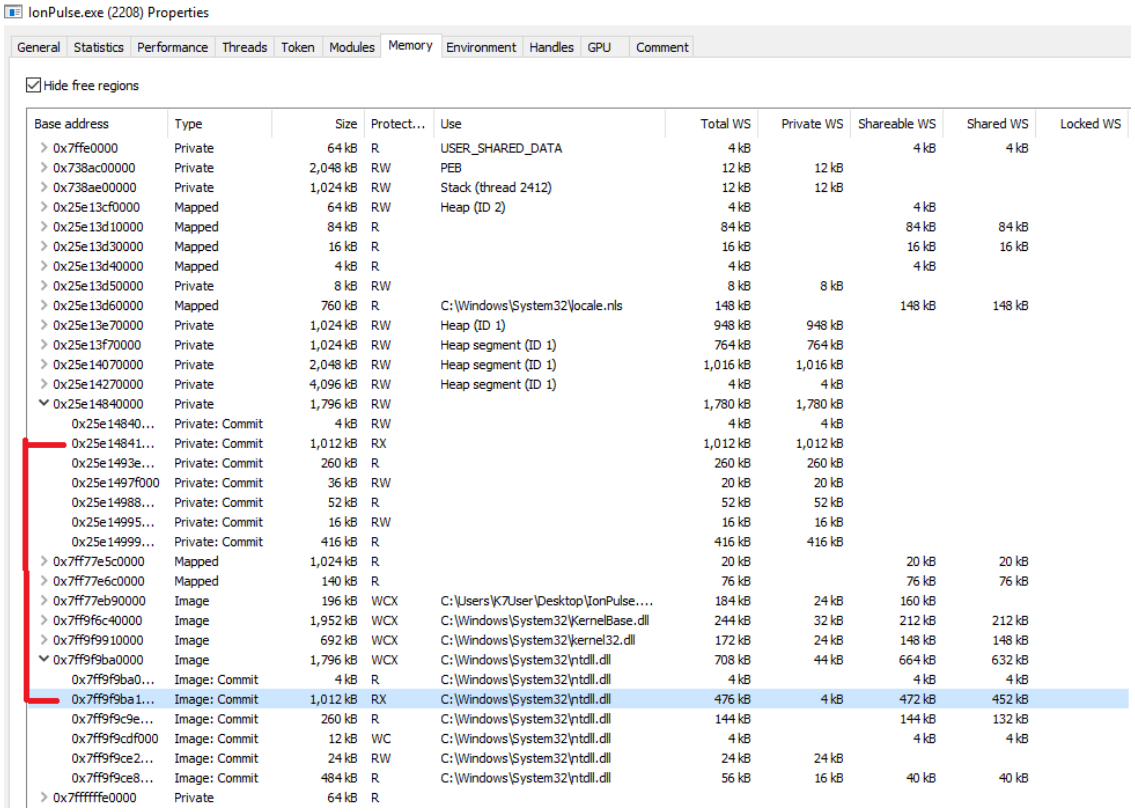


Figure 5: Ntdll.dll copied

Above figure shows the copied content of Ntdll.dll.

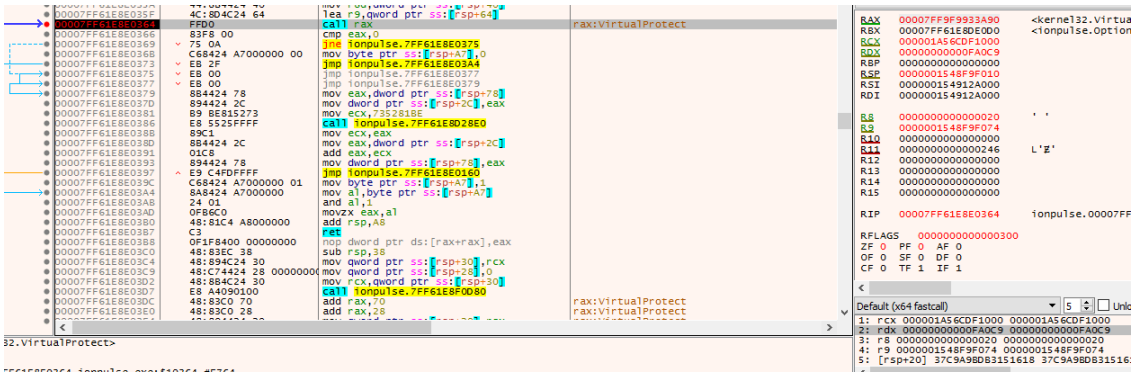


Figure 6: VirtualProtect

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

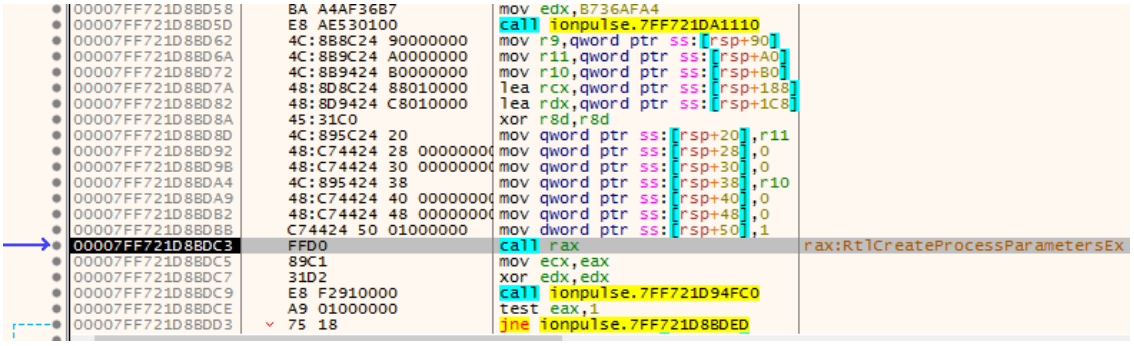


Figure 7: Creating msiexec.exe

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.

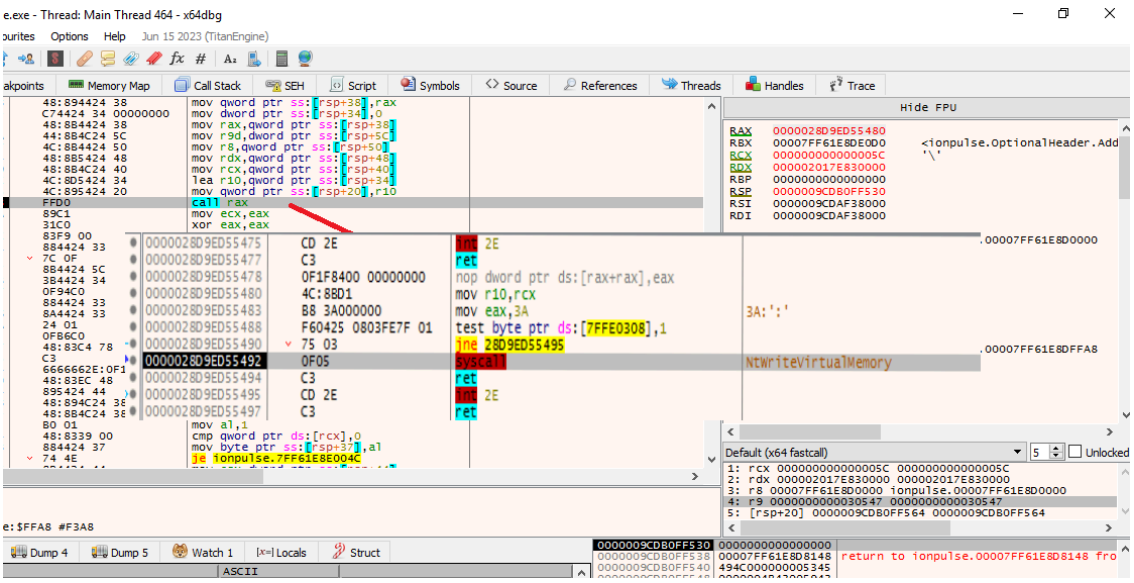


Figure 8: Syscall

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.

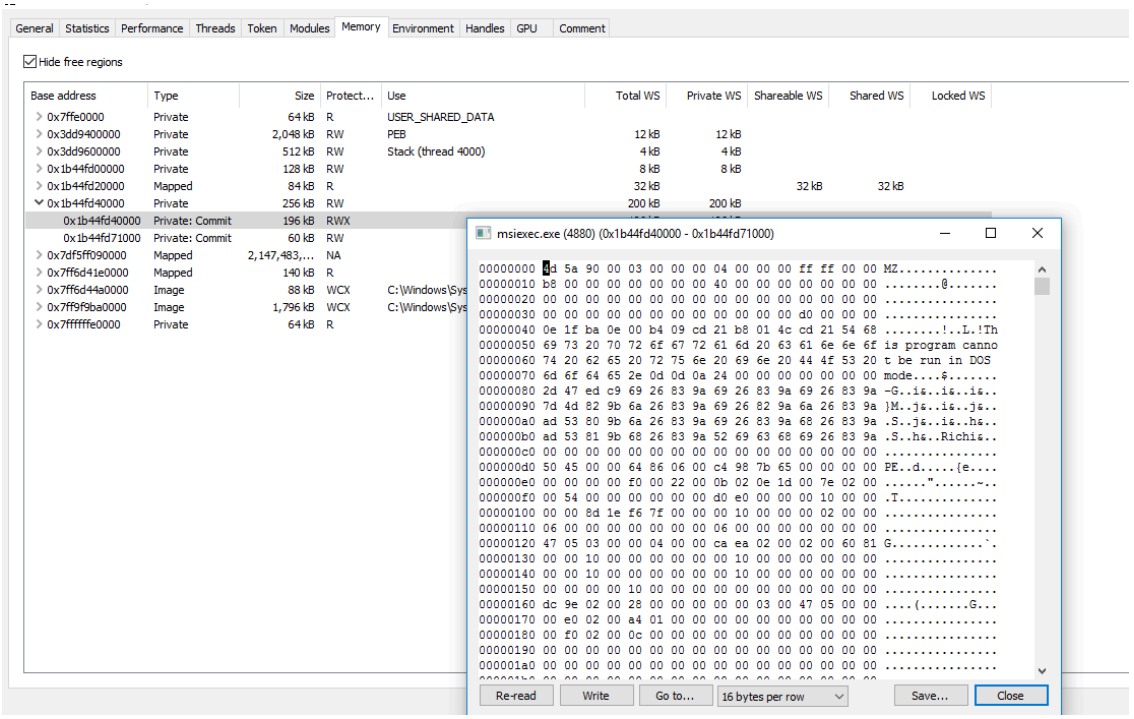


Figure 9: Zloader injected in msiexec.exe

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.

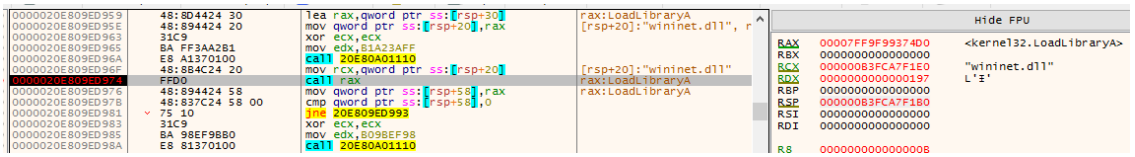


Figure 10: Loading wininet.dll

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

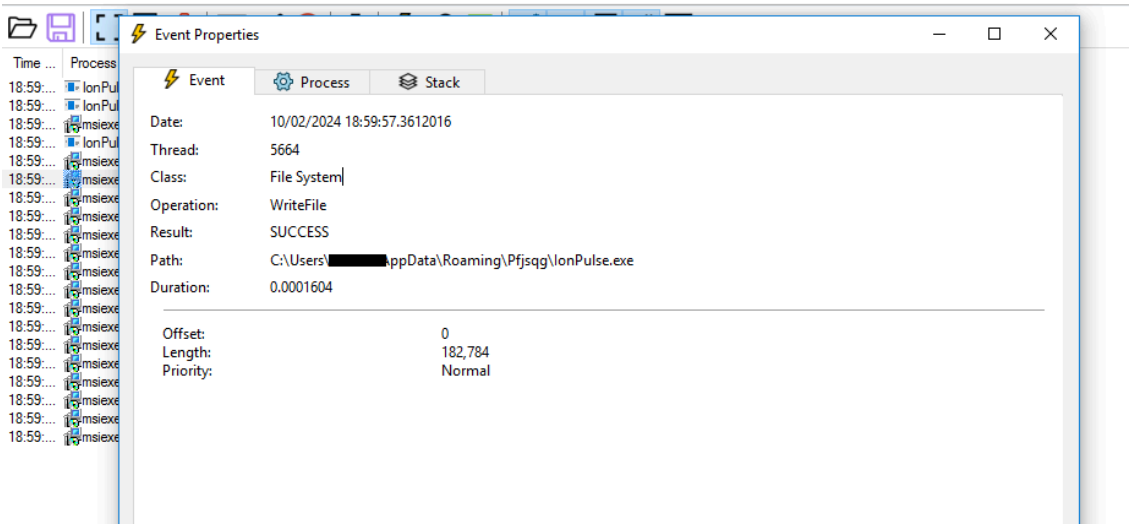


Figure 11: Self Copy

It will then make a self Copy in AppData\Roaming

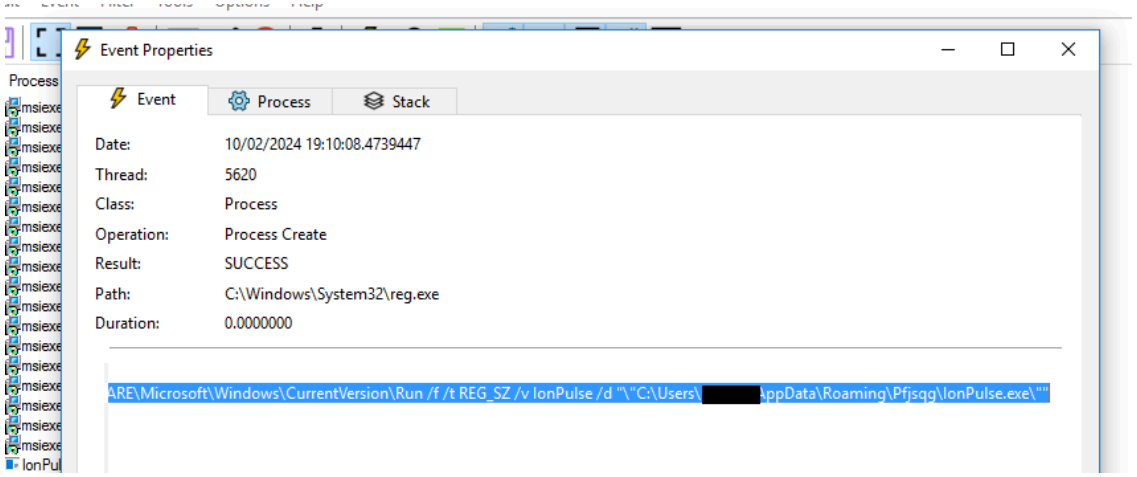


Figure 12: Run Entry

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

---