Halo's Gate Evolves -> Tartarus' Gate

trickster0.github.io/posts/Halo's-Gate-Evolves-to-Tartarus-Gate

November 27, 2021

trickster0 on Nov 27

Updated Nov 27 3 min read

A while ago in my twitter, I have mentioned what a huge fan I am of Hell's Gate and Halo's Gate. Hell's Gate originally is a very creative way to fetch the syscall numbers by parsing the InMemoryOrderModuleLIst from PEB structure. By finding the ntdll.dll address, which is usually the first entry in InMemoryOrderModuleLIst, it is possible to obtain the syscall numbers by parsing its exports for the necessary functions we need.

Even though this is an excellent technique to bypass most of the Antiviruses, unfortunately due to the evolution of EDRs and unhooking, this technique cannot succeed.

Below we can see a normal syscall where Hell's Gate would absolutely work.

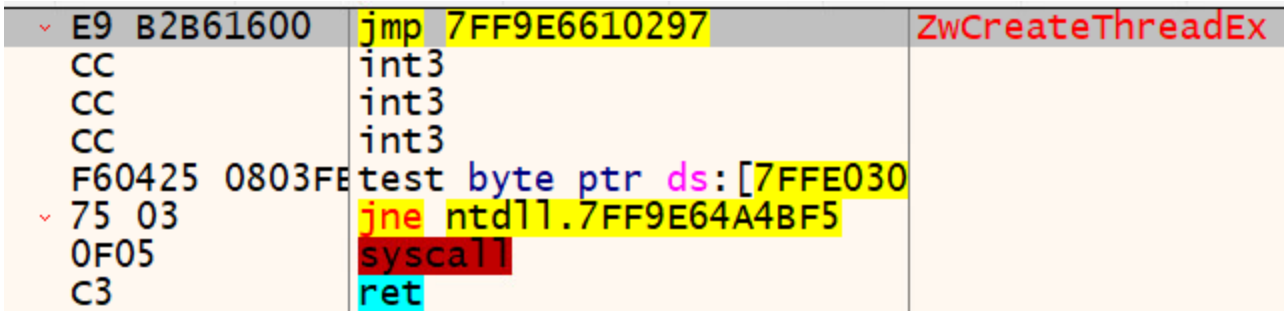
```
0:008> u ntdll!NtAllocateVirtualMemory
ntdll!NtAllocateVirtualMemory:
00007ffc`3a563650 4c8bd1          mov     r10,rcx
00007ffc`3a563653 b818000000      mov     eax,18h
00007ffc`3a563658 f604250803fe7f01 test    byte ptr [SharedUserData+0x308 (00000000`7ffe0308)],1
00007ffc`3a563660 7503            jne     ntdll!NtAllocateVirtualMemory+0x15 (00007ffc`3a563665)
00007ffc`3a563662 0f05            syscall
00007ffc`3a563664 c3              ret
```

As we have mentioned EDRs evolved and a new technique came to light by Reenz0h, called Halo's Gate. Halo's Gate is basically a modified version of Hell's Gate to unhook the WINAPI calls.

For anyone not aware, unhooking is the process where you evade the hooked WINAPI functions by the AVs/EDRs in order for them to check the parameters and the flow of a program.

Halo's Gate basically check the first bytes of the called WINAPI and if they are as they should "4c8bd1b8", then the WINAPI is not hooked and everything proceeds normally, but when the first byte is "e9", then a jmp assembly instructions redirects the execution of the program to the AV/EDR checking engine, hence it is hooked.

In the screenshot you can see what a hooked call looks like by certain EDRs.



Halo's Gate tackles this problem if the byte is "e9" by going up or down and check the syscall of the next or previous syscall, if it is not hooked then we grab the syscall and add +1 byte since they are all in order.

Since I am very fond of this technique and It was not working in different EDRs, I was curious why and I had to dig more since it was not the detection/prevention of the security product but it was just failing. Soon I realized that not all EDRs hook the same way, so I had to bypass and extend it Halo's Gate further into Tartarus' Gate.

Regarding the EDR, that I was against, I am sure it is easy to find out which one it is but apparently it starts with the bytes "4c8bd1e9" as you can see below when the WINAPI call is hooked.

```
0:008> u ntdll!NtAllocateVirtualMemory
ntdll!NtAllocateVirtualMemory:
00007ffe`9d7635c0 4c8bd1          mov     r10,rcx
00007ffe`9d7635c3 e9c0590800      jmp     ntdll!QueryRegistryValue+0x188 (00007ffe`9d7e8f88)
00007ffe`9d7635c8 f604250803fe7f01 test    byte ptr [SharedUserData+0x308 (00000000`7ffe0308)],1
00007ffe`9d7635d0 7503            jne     ntdll!NtAllocateVirtualMemory+0x15 (00007ffe`9d7635d5)
00007ffe`9d7635d2 0f05            syscall
00007ffe`9d7635d4 c3              ret
```

Basically what I did was to modify the Halo's Gate code by adding one more check, to check for the 4th byte if it is "e9", if it is, it will do the same as the explanation on Halo's Gate to unhook it, so I ended up calling this Tartarus' Gate.

I am certain there are more EDRs that have their own hooking method so I can see how this could evolve even further depending on the situation.

Source Code can be found here You will notice that the custom way to copy the shellcode to the allocated space is removed, for some reason it was not working very well against this EDR so I would avoid depending on the case. Also, this code might fail a few times depending on the EDR, so if it will not work on the first try, try a few times. If you use a different method that works better than CreateRemoteThread, it will work in a very stable manner.

Resources: https://sektor7.net/#!res/2021/halosgate.md
https://github.com/am0nsec/HellsGate

Credits to : Reenz0h from Sektor7 for Halo's Gate and the authors of Hell's Gate - Paul Laîné and smelly__vx