Bypassing Access Mask Auditing Strategies

Jonathan Johnson · Follow

was unable to talk about the potential strategies an attacker could take in hopes to fly under the radar of any access right detections. I would like to touch on that now and some of the research that went along with it.

Research

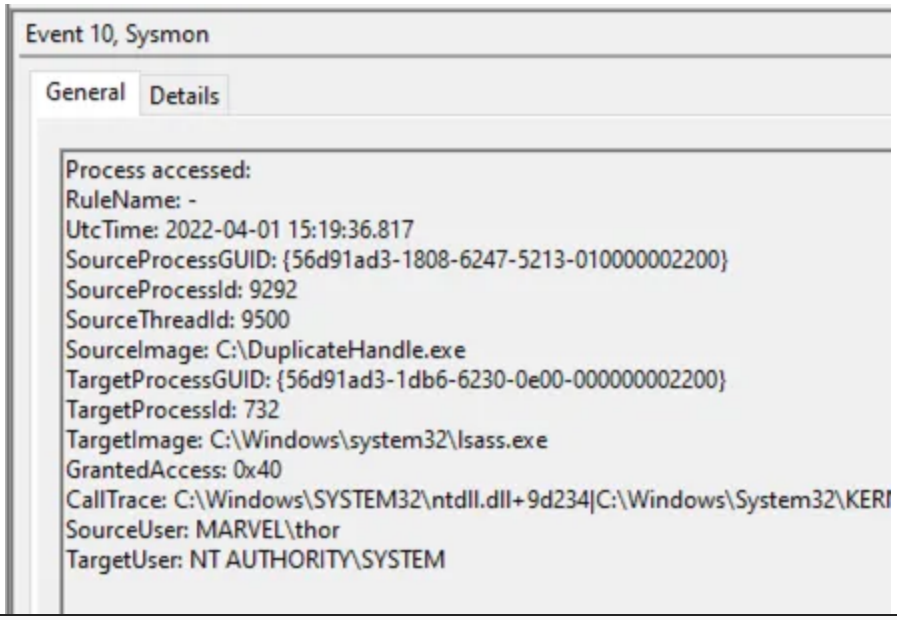
DuplicateHandle

Previously while doing access token research I ran across a Win32 API — DuplicateTokenEx that could be leveraged to duplicate a handle to a token and specify the updated rights wanted on the duplicated handle. This

attacker wanted to fly under the radar and make it look like they only got rights `0x40` to LSASS, but in reality they duplicated that handle and specified the correct rights within that duplicated handle allowing them to do some bad afterwards, say dump LSASS.

I ran this idea by James Forshaw and he was generous enough to send me a great blog — Bypassing SACL Auditing on LSASS that leverages this idea, but to bypass SACL auditing....best part it was from 2017.

This expanded my understanding on this strategy as it could now be used to:



Now there are a lot of handle requests from beacon (which Anton also mentions), but I will dive into that later. My theory was that beacon was leveraging James' duplicate handle strategy to fly under the radar for SACL auditing. Let's test this.

Beacon

The first thing I did before testing beacon was to track down any other functions that DuplicateHandle calls under the hood. I did this because a lot of code now a days seem to call either syscalls directly or some undocumented function (ntdll.dll) to avoid detection.

My process was as follows:

- Launch beacon with WinDbg attached in user-mode.
- Broke on **kernelbase!OpenProcess** where the process id was equal to my winlogon process (have to do this because beacon requests handles to System, csrss, smss, wininit, etc).
- Broke on **advapi32!DuplicateHandle**
- Broke on **advapi32!OpenProcessToken**

This was a pretty targeted approach. I approached it this way because I am comfortable with the function calls used for impersonation. There are other ways one could go about doing something similar if they were unaware of the API calls used, like using Time Travel Debugging or API Monitor.

I wanted to show this process to show that there are tools in the wild that are leveraging this bypass technique. Cobalt Strike isn't the only one, but being that it seems to be so popular I wanted to use it as a use-case.

- Cobalt Strike is leveraging this strategy

It is good to note, not only Cobalt Strike is leveraging this strategy but others might be as well. Attackers have to access objects when performing their techniques, my hope is to bring awareness of how defenders can look into protecting and auditing these objects better.

References

- Bypassing SACL Auditing on LSASS by James Forshaw

Written by Jonathan Johnson

870 Followers

Principal Security Engineer @Prelude | Windows Internals