Sign in

codewhitesec / SysmonEnte Public

Notifications Fork 6 Star 68

Code Issues 1 Pull requests Actions Projects Security Insights

main Go to file Code

- SACLProtect
- bin
- helpers
- screens
- src
- DISCLAIMER.md
- Makefile
- Readme.md

About

No description, website, or topics provided.

Readme
Activity
Custom properties
68 stars
2 watching
6 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 2

Languages

README

SysmonEnte

This is a POC attack on the integrity of Sysmon which emits a minimal amount of observable events even if a `SACL` is in place.
To our understanding, this attack is difficult to detect in

environments where no security sensors other than Sysmon or the Windows Event Log are in use.

For more technical information on the attack and possible mitigations, please see our blogpost.

Motivation

Multiple different attacks on Sysmon or the Event Log service exist. To the best of our knowledge, all of these are detectable using Sysmon or the Event Log itself:

- Driver (Un)Loading.
- Registry changes.
- Process Access events.
- Lack of events from certain hosts.
- ...

Motivated defenders can observe the corresponding events to identify attacks on Sysmon or the Event Log service.
Our goal was to create an attack on Sysmon which is difficult to fingerprint as no suspicious events are emitted during the attack.

Description

Similarly to SysmonQuiet or EvtMute, the idea is to inject code into Sysmon which redirects the execution flow in such a way that events can be manipulated before being forwarded to the SIEM.

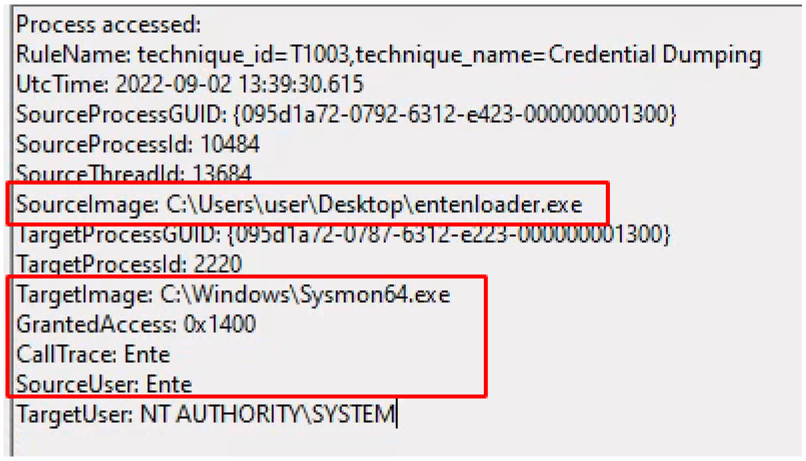
However, during this attack, **no suspicious ProcessAccess events on Sysmon are observable via Sysmon or the Event Log** making the detection (supposedly) non trivial.

The attack flow is as follows:

- Suspend all threads of Sysmon.

● C 92.8% ● Makefile 4.4%
● Assembly 1.6% ● Other 1.2%

- Create a limited handle to Sysmon and elevate it by duplication.
- Clone the pseudo handle of Sysmon to itself in order to bypass `SACL` as proposed by James Forshaw.
- Inject a hook manipulating all events (in particular ProcessAccess events on Sysmon).
- Resume all threads.



```
Process accessed:
RuleName: technique_id=T1003,technique_name=Credential Dumping
UtcTime: 2022-09-02 13:39:30.615
SourceProcessGUID: {095d1a72-0792-6312-e423-000000001300}
SourceProcessId: 10484
SourceThreadId: 13684
SourceImage: C:\Users\user\Desktop\entenloader.exe
TargetProcessGUID: {095d1a72-0787-6312-e223-000000001300}
TargetProcessId: 2220
TargetImage: C:\Windows\Sysmon64.exe
GrantedAccess: 0x1400
CallTrace: Ente
SourceUser: Ente
TargetUser: NT AUTHORITY\SYSTEM
```

Usage

SysmonEnte is implemented as fully position independent code (PIC) which can be called using the following prototype:

```
DWORD go(DWORD dwPidSysmon);
```

A sample loader is included and built during compilation when typing `make`.

```
.\EntenLoader.exe <PID Sysmon>
```

Additionally, SysmonEnte uses indirect syscalls to bypass possible userland hooks.

The open source variant tampers with process access events to Lsass and Sysmon and sets the access mask to a benign one.

Additionally, the source user and the callstack is set to **Ente**. You can change these to your needs.

Detections

To our understanding, the `SACL` bypass by James Forshaw can be identified by configuring a `SACL` with `PROCESS_DUP_HANDLE` on Sysmon. `Event 4656` should then be emitted upon the instantiation of a handle allowing to clone other handles from Sysmon.

However, configuring such a `SACL` using well known administration tools appears to be non trivial. A sample program to set such a `SACL` is included in the folder `SACLProtect`. Note that `Object Access Auditing` is not enabled by default.

References

- Implementation by our @testert01 and @thefLinkk
- SysmonQuiet by Scriptidiot
- EvtMute by Batsec
- James Forshaw's SACL Bypass

Trivia