

Product Solutions Resources Open Source Enterprise Pricing

Search

Sign inSign up

Yaxser / BackstabPublic

NotificationsFork237Star1.4k

CodeIssues5Pull requestsActionsProjectsSecurityInsights

masterGo to fileCode

Yaxser Alhazmi

bug fixes and code re-arrangement

ed1fd61 · 3 years ago

18 Commits

Backstab	bug fixes and code re-arrangement	3 years ago
resources	initial commit	3 years ago
.gitignore	reinstate change on line 92	3 years ago
Backstab.sln	initial commit	3 years ago
README.md	Update README.md	3 years ago

README

Backstab

Kill EDR Protected Processes

Have these local admin credentials but the EDR is standing in the way? Unhooking or direct syscalls are not working against the EDR? Well, why not just kill it? Backstab is a tool capable of killing antimalware protected processes by leveraging sysinternals' Process Explorer (ProcExp) driver, which is signed by Microsoft.

What can it do?

Usage: backstab.exe <-n name || -p PID> [options]

-n, Choose process by name, including the .exe suffix

-p, Choose process by PID

-l, List handles of protected process

-k, Kill the protected process by closing its handles

-x, Close a specific handle

-d, Specify path to where ProcExp will be extracted

-s, Specify service name registry key

-u, Unload ProcExp driver

-a, adds SeDebugPrivilege

-h, Print this menu

Examples:

backstab.exe -n cyserver.exe -k [kill process]

backstab.exe -n cyserver.exe -x E4C [Close handle]

backstab.exe -n cyserver.exe -l [list handles]

backstab.exe -p 4326 -k -d c:\\driver.sys [kill protected process]

About

A tool to kill antimalware protected processes

Readme

Activity

1.4k stars

26 watching

237 forks

Report repository


Releases

4 tags


Packages

No packages published

Contributors2



k4nfr3



Yaxser Yasser

Languages

C100.0%

OpSec

Page 1 of 2

Here is a quick rundown of what happens

1. Embedded driver is dropped to disk
2. Registry key under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services is created
3. The privilege SE_PRIVILEGE_ENABLED is acquired because it is necessary to load the driver
4. Driver is loaded using NtLoadDriver to avoid creating a service
5. The created Registry key is deleted (service not visible during execution)
6. Communication with the driver is via using DeviceIoControl
7. For handle enumeration, NtQuerySystemInformation is called

What you should also know

1. The behavior of the tool mimics that of ProcExp. ProcExp drops the driver to the disk, create registry key under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services, calls NtLoadDriver, and then delete the registry key
2. You can specify the location to which the driver is dropped and the service name
3. When done, the app will unload the driver. The driver is unloaded by first re-creating the registry keys and then calling NtUnloadDriver
4. The loaded driver is signed by MS
5. The process does not attempt to directly kill protected processes handles, it instructs ProcExp driver to kill them. You won't be accused of attempting to tamper with any processes

Further Research

While the tool purpose is listing and killing handles, the opportunities are vast. It is possible to duplicate the handles to your own process instead of killing them. This could allow for deeper tampering where you write to files, fire events, hold mutexes. To support further research, I tried to make the code readable and split it to many methods to facilitate reuse, I also left a description on all ProcExp related methods. Feel free to reach out to me on [Twitter](#) or by [Email](#)

Credits

- Author: Yasser Alhazmi (@Yas_o_h)
- Pavel Yosifovich: ([@Zodiacon](#)) mentioned to us during his awesome [Windows Internals Course](#) that kernel drivers like ProcExp might cause too much unintended damage
- Cornelis de Plaa [@Cn33liz](#) and Outflank Team [@OutflankNL](#): for [Ps-Tools](#) and their outstanding Github repos, always informative
- Mark Russinovich: for ProcExp, and all Sysinternals tools!