.. /Cdb.exe

Star 7,060

Execute

Debugging tool included with Windows Debugging Tools.

Paths:
C:\Program Files (x86)\Windows Kits\10\Debuggers\x64\cdb.exe
C:\Program Files (x86)\Windows Kits\10\Debuggers\x86\cdb.exe

Resources:

- http://www.exploit-monday.com/2016/08/windbg-cdb-shellcode-runner.html
- https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/cdb-command-line-options
- https://gist.github.com/mattifestation/94e2b0a9e3fe1ac0a433b5c3e6bd0bda
- https://mrd0x.com/the-power-of-cdb-debugging-tool/
- https://twitter.com/nas_bench/status/1534957360032120833

Acknowledgements:

- Matt Graeber (@mattifestation)
- mr.d0x (@mrd0x)
- Spooky Sec (@sec_spooky)
- Nasreddine Bencherchali (@nas_bench)

Detections:

- Sigma: proc_creation_win_lolbin_cdb.yml
- Elastic: defense_evasion_unusual_process_network_connection.toml
- Elastic: defense_evasion_network_connection_from_windows_binary.toml
- BlockRule: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/microsoft-recommended-block-rules

Execute

1. Launch 64-bit shellcode from the x64_calc.wds file using cdb.exe.

```
cdb.exe -cf x64_calc.wds -o notepad.exe
```

Use case:	Local execution of assembly shellcode.
Privileges required:	User
Operating systems:	Windows
ATT&CK® technique:	T1127: Trusted Developer Utilities Proxy Execution

2. Attaching to any process and executing shell commands.

```
cdb.exe -pd -pn <process_name>
.shell <cmd>
```

Use case:	Run a shell command under a trusted Microsoft signed binary
Privileges required:	User
Operating systems:	Windows
ATT&CK® technique:	T1127: Trusted Developer Utilities Proxy Execution

3. Execute arbitrary commands and binaries using a debugging script (see Resources section for a sample file).

```
cdb.exe -c C:\debug-script.txt calc
```

Use case:	Run commands under a trusted Microsoft signed binary
Privileges required:	User
Operating systems:	Windows
ATT&CK® technique:	T1127: Trusted Developer Utilities Proxy Execution