


 [scumjr](#) / [dirtycow-vdso](#)

PoC for Dirty COW (CVE-2016-5195)

 MIT License 360 stars  130 forks Star Watch ▼

< > Code

 Pull requests Actions Projects Security Insights master ▼

...



scumjr switch to MIT license ...

on Feb 27, 2017

 27[View code](#)

README.md

Oxdeadbeef

PoC for [Dirty COW](#) (CVE-2016-5195).

This PoC relies on ptrace (instead of `/proc/self/mem`) to patch vDSO. It has a few advantages over PoCs modifying filesystem binaries:

- no setuid binary required
- SELinux bypass
- container escape
- no kernel crash because of filesystem writeback

And a few cons:

- architecture dependent (since the payload is written in assembly)
- doesn't work on every Linux version
- subject to vDSO changes

Payload

The current payload is almost the same as in [The Sea Watcher](#) and is executed whenever a process makes a call to `clock_gettime()`. If the process has root privileges and `/tmp/.x` doesn't exist, it forks, creates `/tmp/.x` and finally creates a TCP reverse shell to the exploit. It isn't elegant but it could be used for container escape.

TODO

- payload improvement
- release of the tool for vDSO payloads testing

Detecting if vDSO is successfully patched isn't bulletproof. During the *restore* step, the vDSO is effectively restored but the exploit fails to report it correctly. Indeed, the vDSO changes don't seem to affect the exploit process.

Releases

No releases published

Packages

No packages published

Languages

● C 77.9% ● Assembly 20.5% ● Makefile 1.6%