



Obreros del Kernel por 20 minutos

Sebastián Alvarez - github.com/alvarezgarcia

\$ whoami

Sebastián Tomás Alvarez

- 36 años, pergaminense y bonarense por adopción, desde el 2018.
 - Linuxero desde el 99, en el rubro desde el 2008.
 - Actualmente **Tech Lead @ Fuse Finance**.
-
- NodeJS, React, AWS para comer.
 - Linux, Python, C, Go, Perl para divertirse.



Advertencia

La siguiente presentación contiene escenas de nerdismo frontal explícito, se aconseja la supervisión de un adulto.

Agenda

- Kernel
- eBPF
- Kernel + eBPF
- Ejemplos!





Kernel

Hardware

Kernel



User space



Hardware

Kernel



SYSCALLS

User space

Syscalls (algunas)

- open
- read
- write
- unlink
- chmod
- chown





eBPF

Berkeley Packet Filter

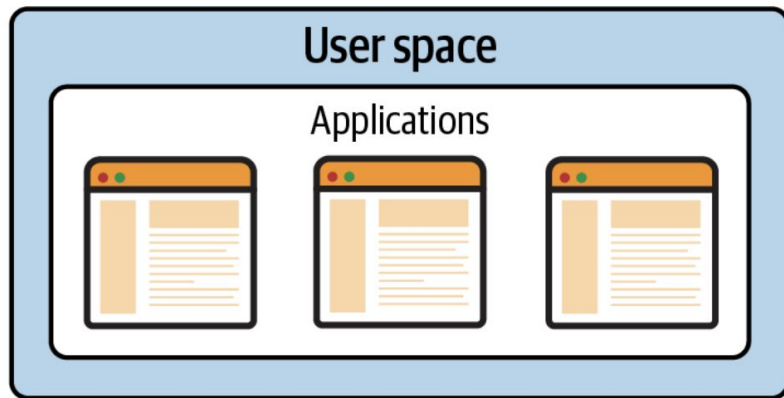
- **1993** - [The BSD Packet Filter: A new Architecture for User-level Packet Capture](#)
 - **1997** - Kernel 2.17.5 - backend de tcpdump
-
- **2012** - Kernel 3.5 - seccomp
 - **2014** - Kernel 3.18 - extended BPF



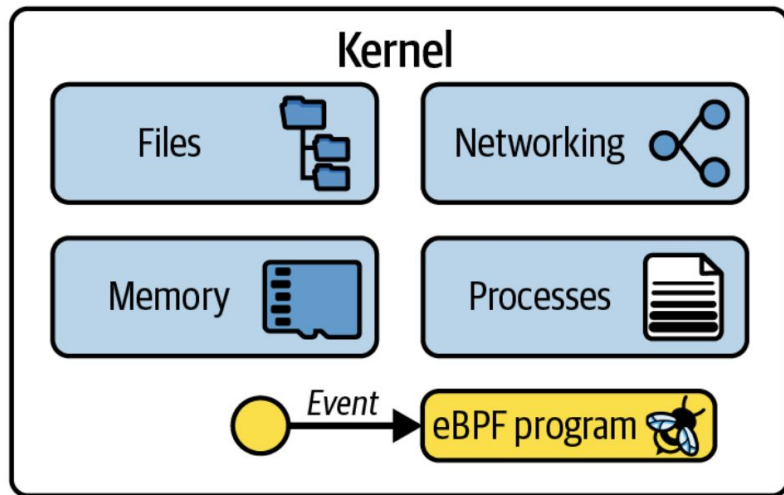
Kernel + eBPF

Kernel + eBPF =

**código attacheado a eventos del
kernel**



System calls

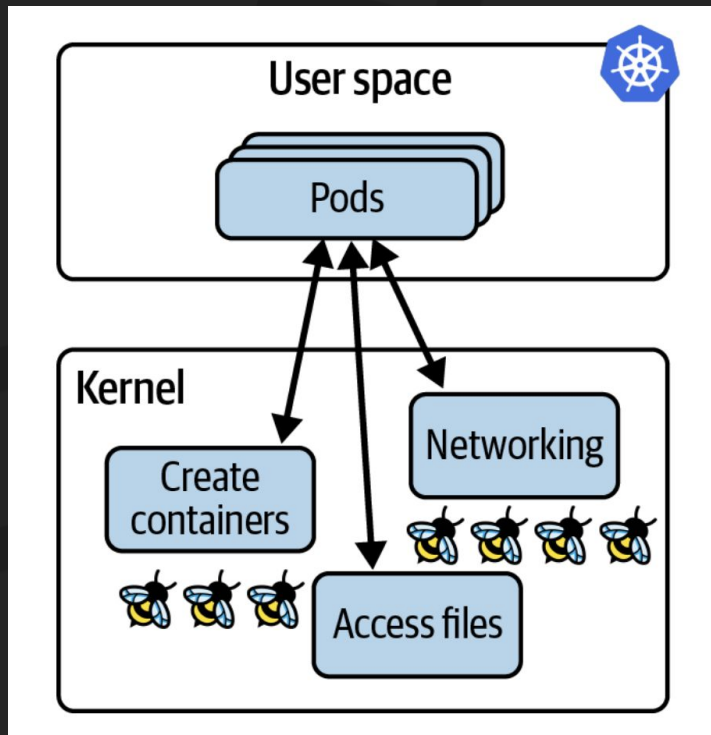


eBPF: Un lugar de privilegio

- Performance
- Visibilidad
- No downtime



eBPF: Un lugar de privilegio (también para k8s)



Basta de cháchara

Material

[Why I'm OBSESSED With eBPF](#)

[Kernelless Kernel Programming \(eBPF\) - Computerphile](#)

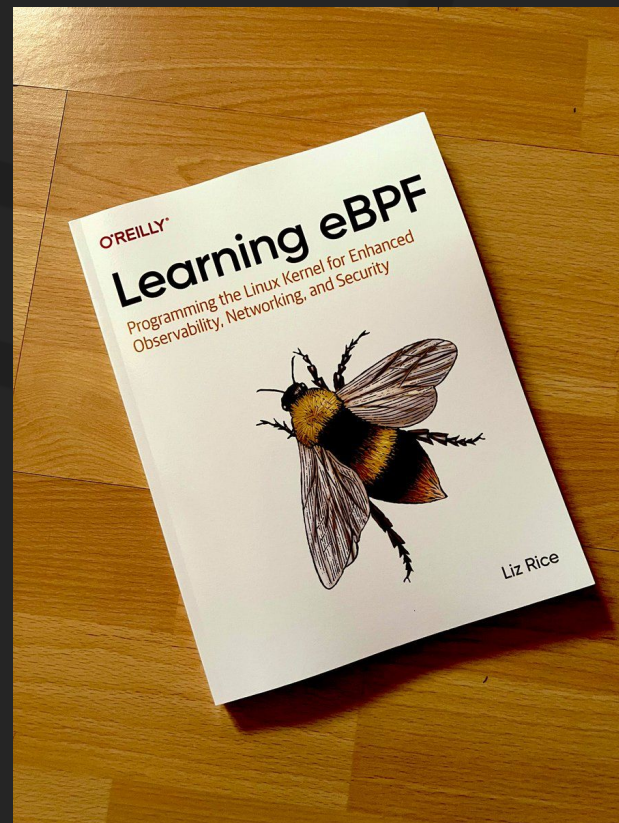
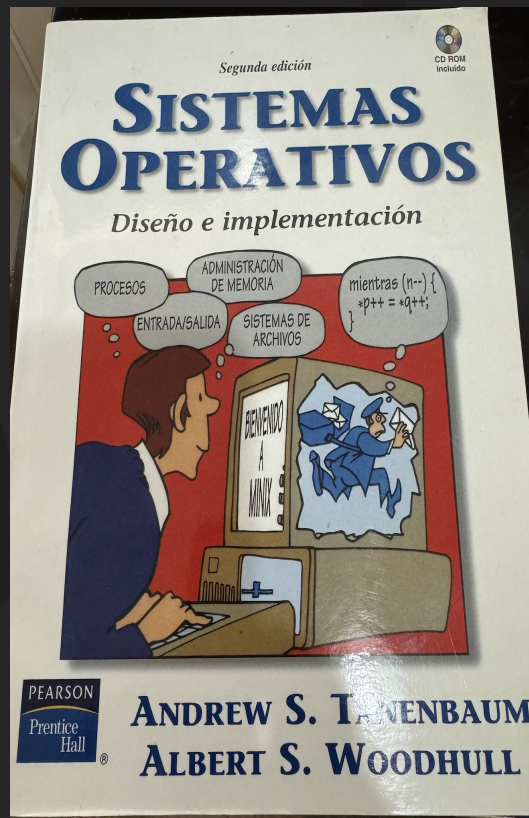
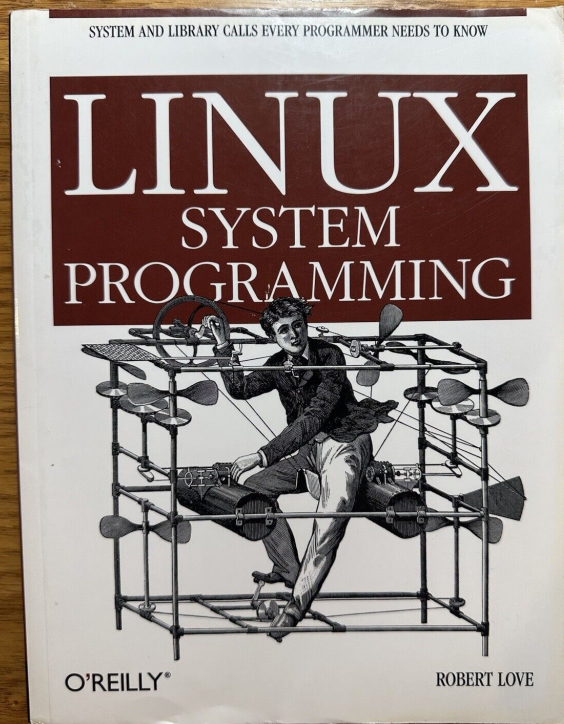
[Filip Nikolic – Demystifying eBPF - eBPF Firewall from scratch](#)

[Liz Rice – eBPF for Security](#)

[BPF With C](#)



Material



Llévate un recuerdo





Muchas gracias

