Monitoring and protecting SSH sessions with eBPF

SSTIC 2021

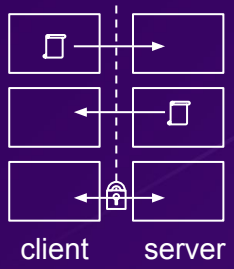
DATADOG

Guillaume Fournier

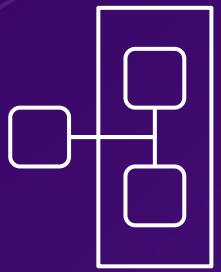
github.com/gui774ume

guillaume.fournier@datadoghq.com

- Secured access to an infrastructure over an untrusted network
- Least privilege principle should be applied
- 4 security recommendations:

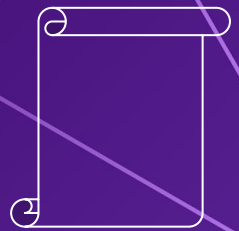


1. Certificate based authentication



2. Bastion host

3. MFA



4. Audit logs

- But ...
  - MFA isn't perfect
  - Access is rarely granular
  - SSH audit logs are limited

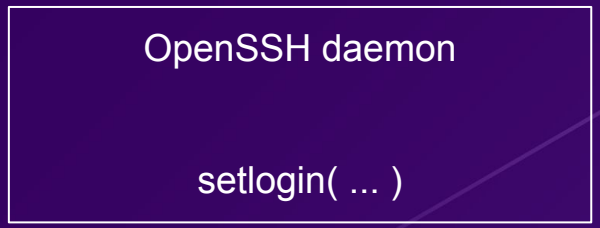
We're still missing a solution to mitigate the impact of stolen credentials.

October 2019

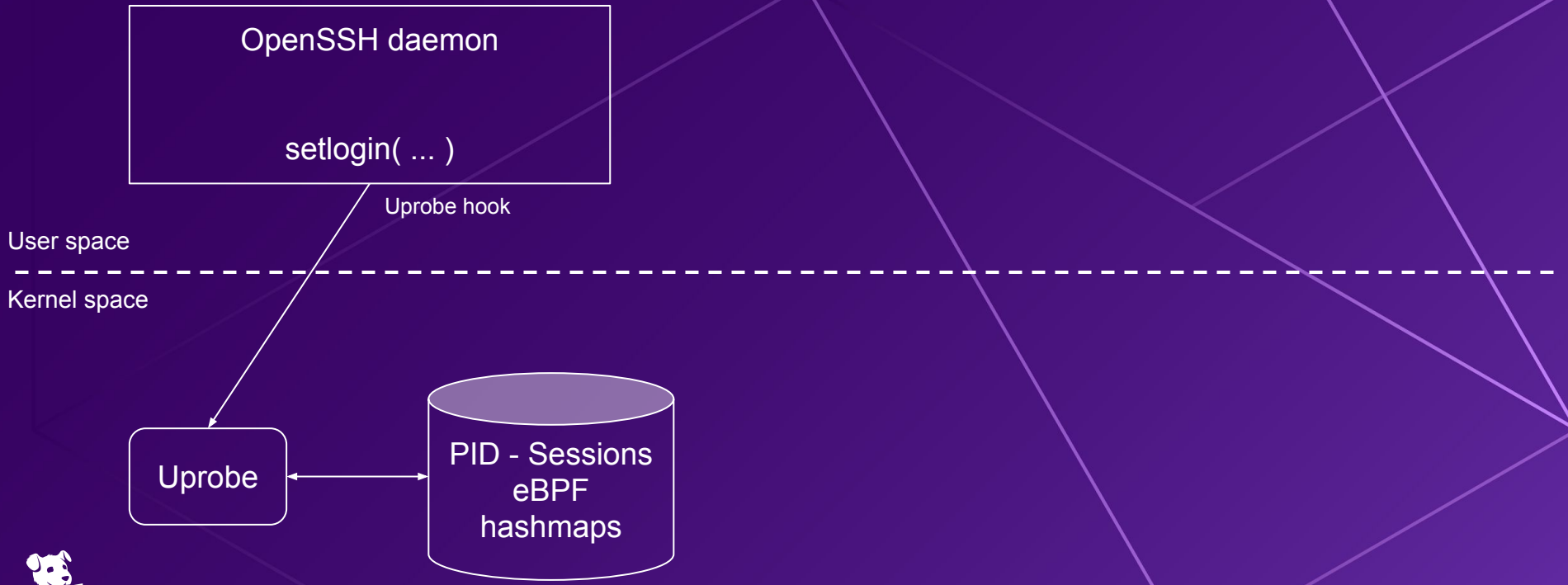
- Goals:
  - Provide real time visibility into active sessions
  - Time, session and scope based access control

- Security profiles per SSH user (**regardless** of UNIX user / sudoer):
  - Allow (audit)
  - MFA
  - Block
  - Kill session

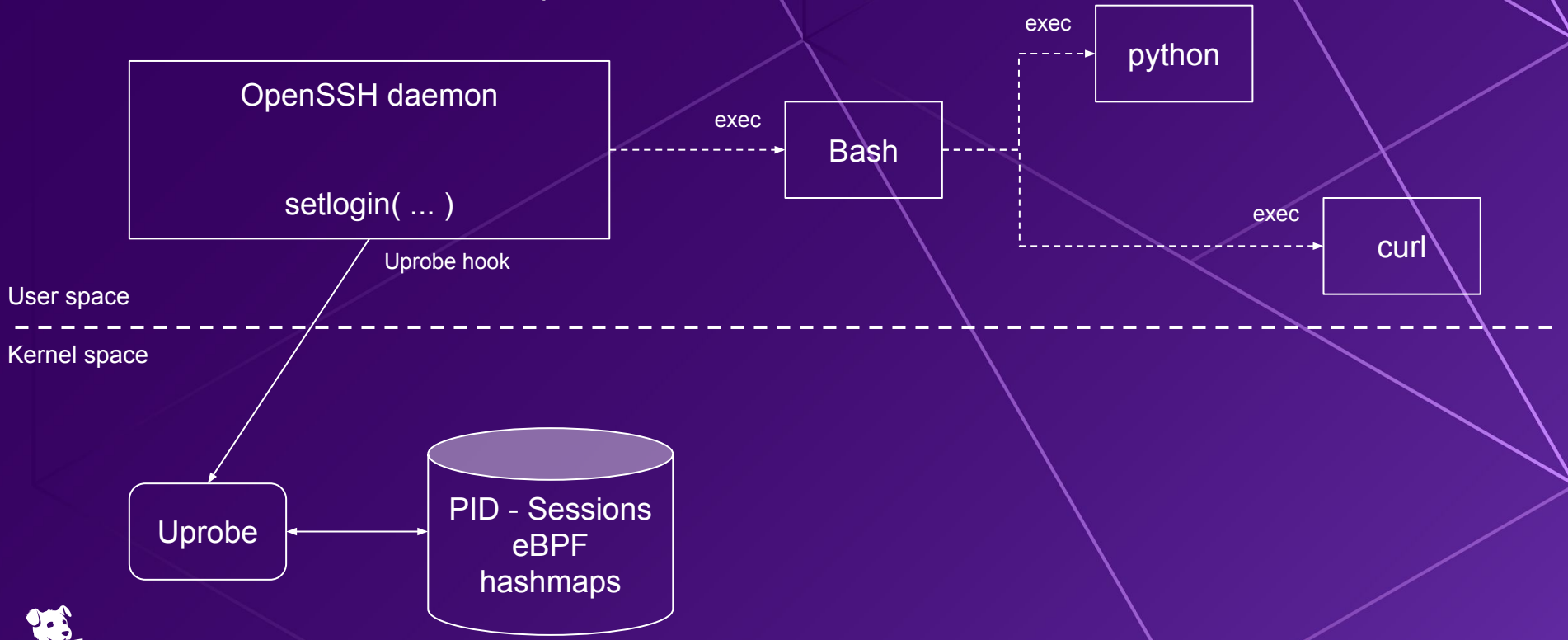
- Technical bits 1: eBPF hook points



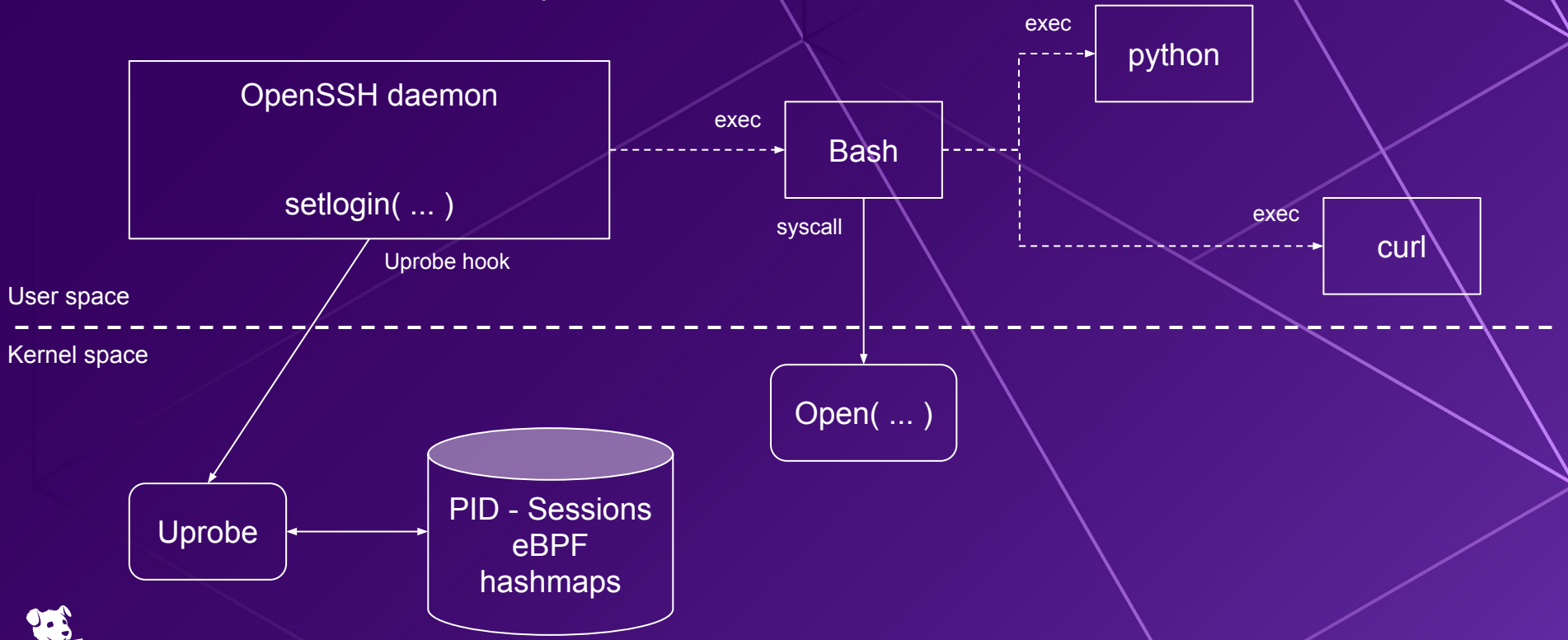
- Technical bits 1: eBPF hook points



- Technical bits 1: eBPF hook points

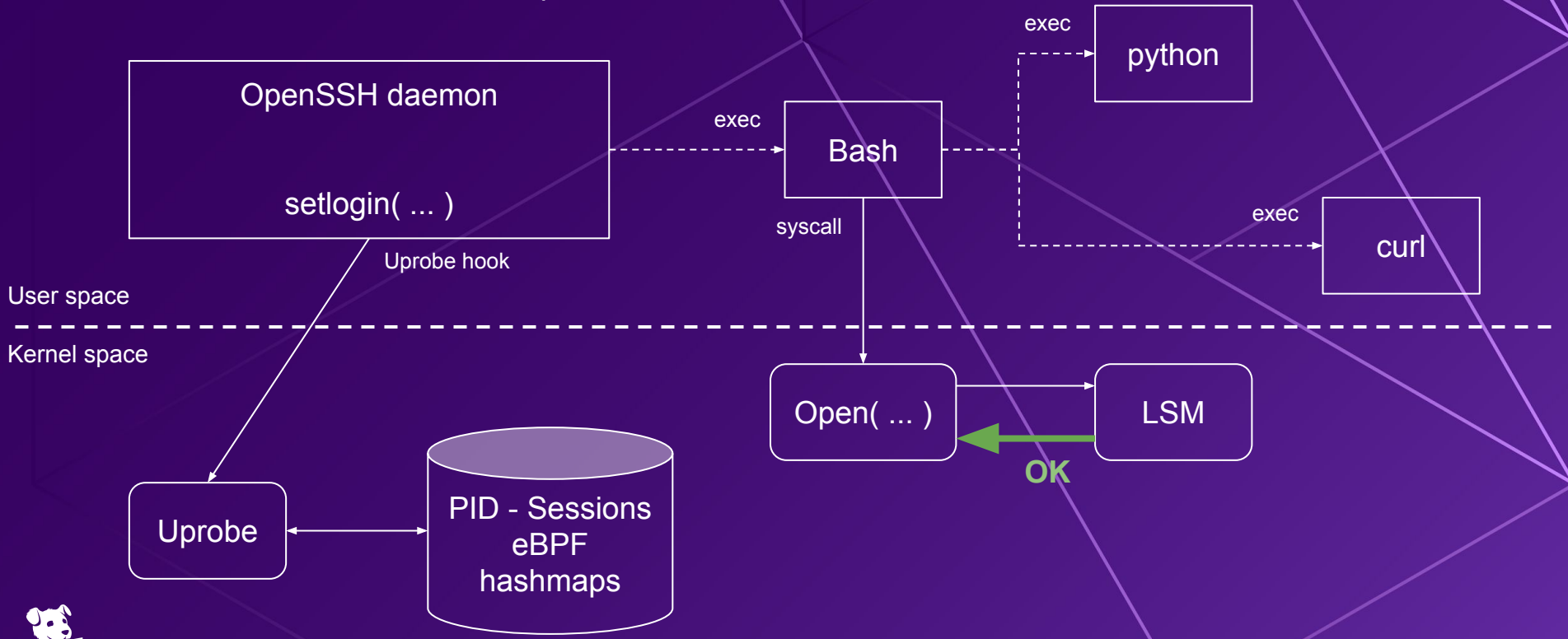


- Technical bits 1: eBPF hook points



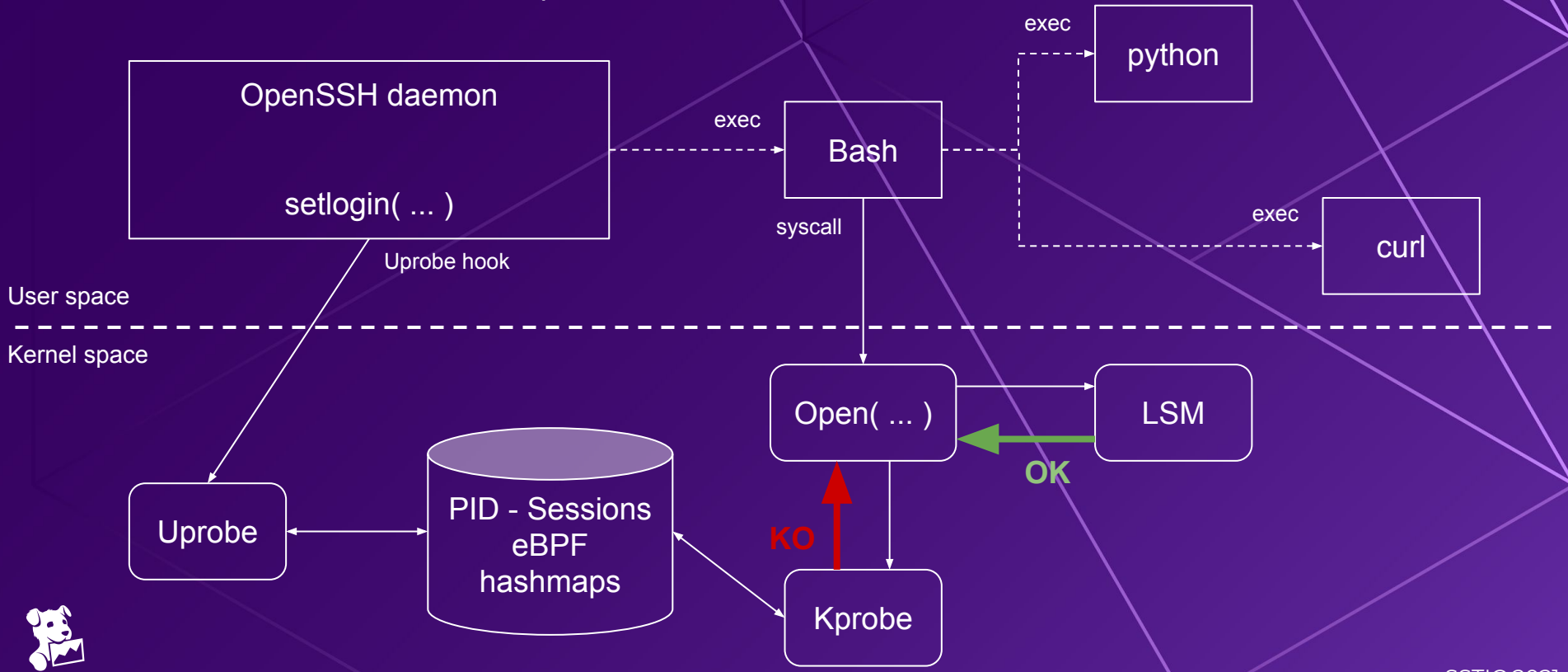
DATADOG

- Technical bits 1: eBPF hook points

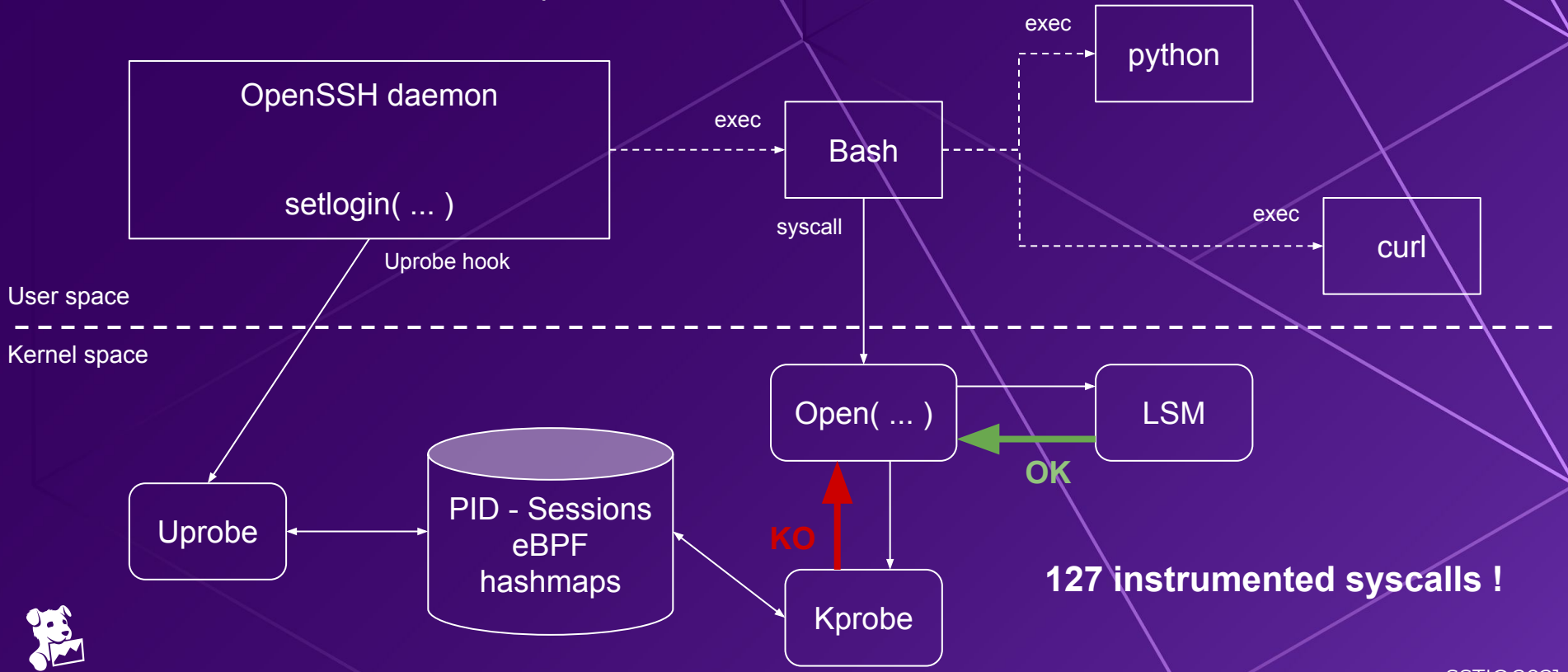


DATADOG

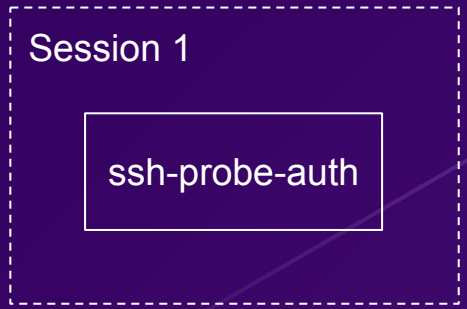
- Technical bits 1: eBPF hook points



- Technical bits 1: eBPF hook points



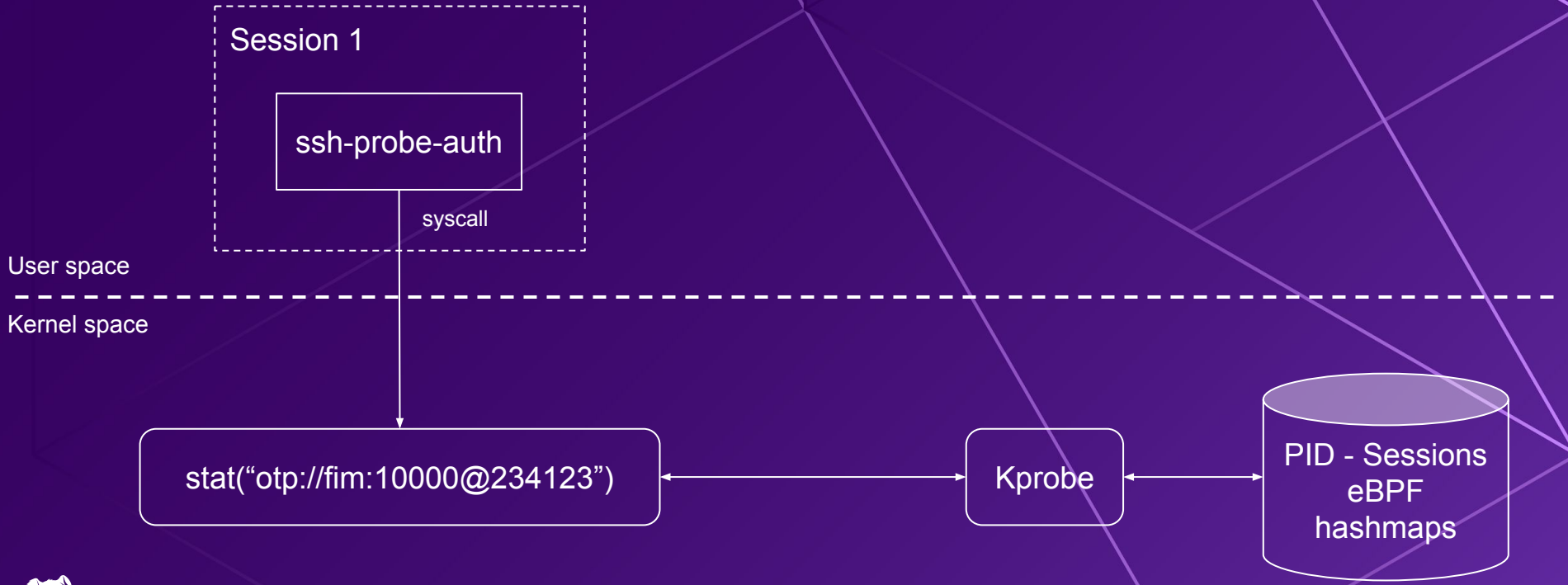
- Technical bits 2: MFA with eBPF



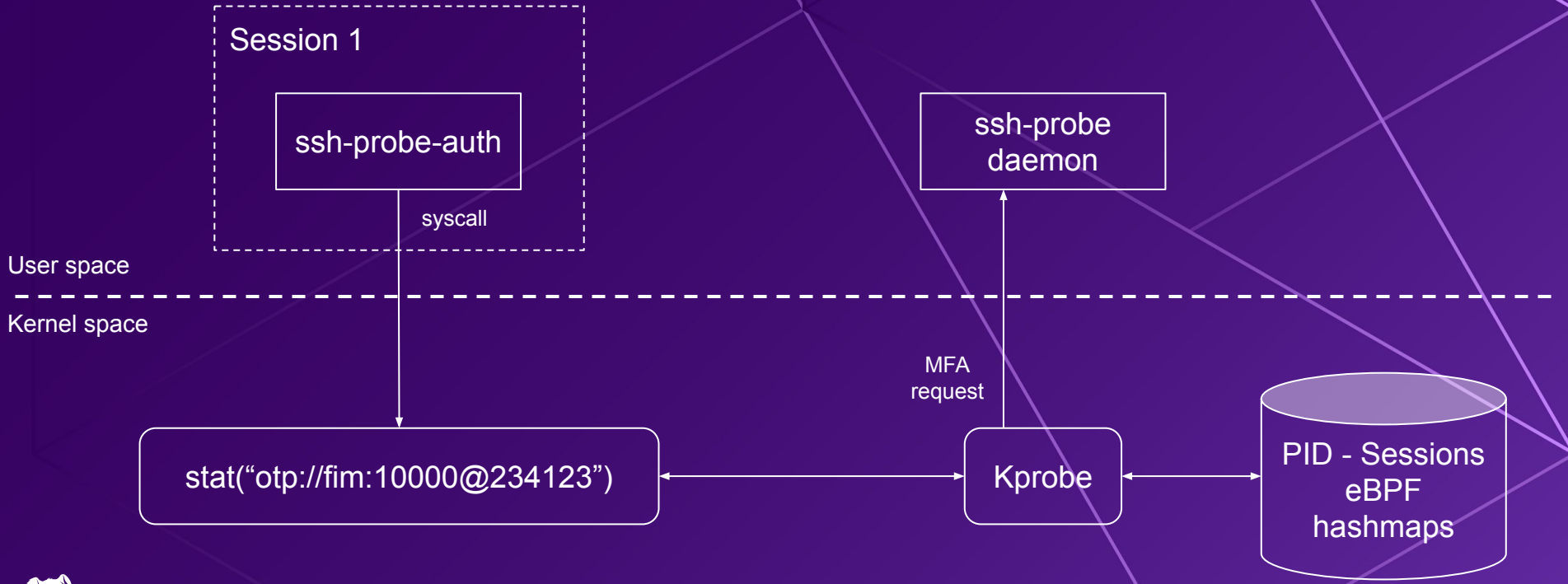
User space

Kernel space

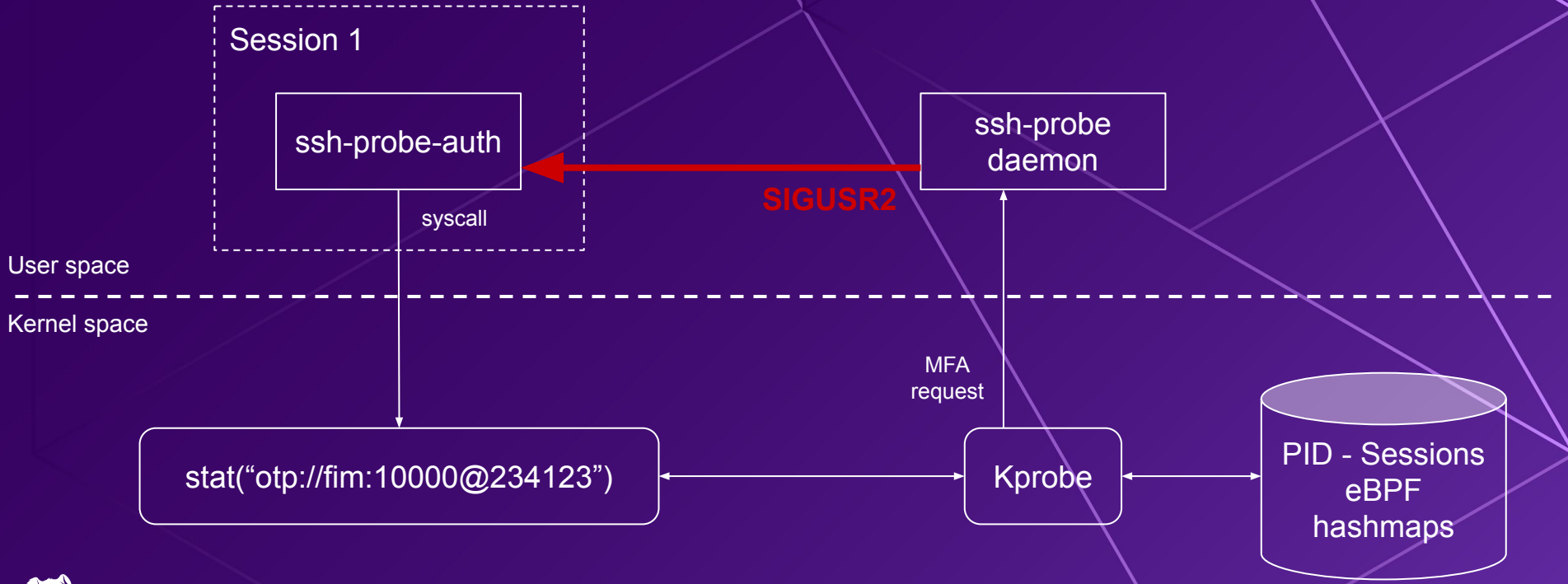
- Technical bits 2: MFA with eBPF



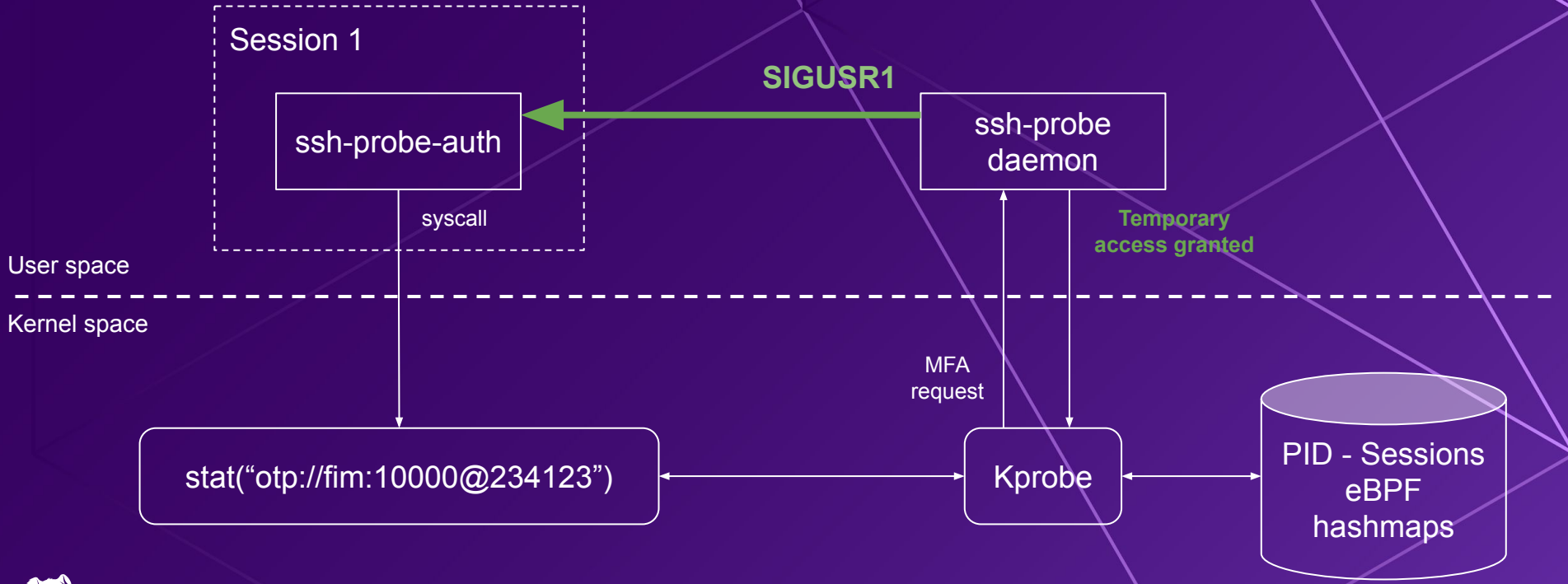
- Technical bits 2: MFA with eBPF



- Technical bits 2: MFA with eBPF



- Technical bits 2: MFA with eBPF



Demo

Thanks !

Source code:
github.com/Gui774ume/ssh-probe