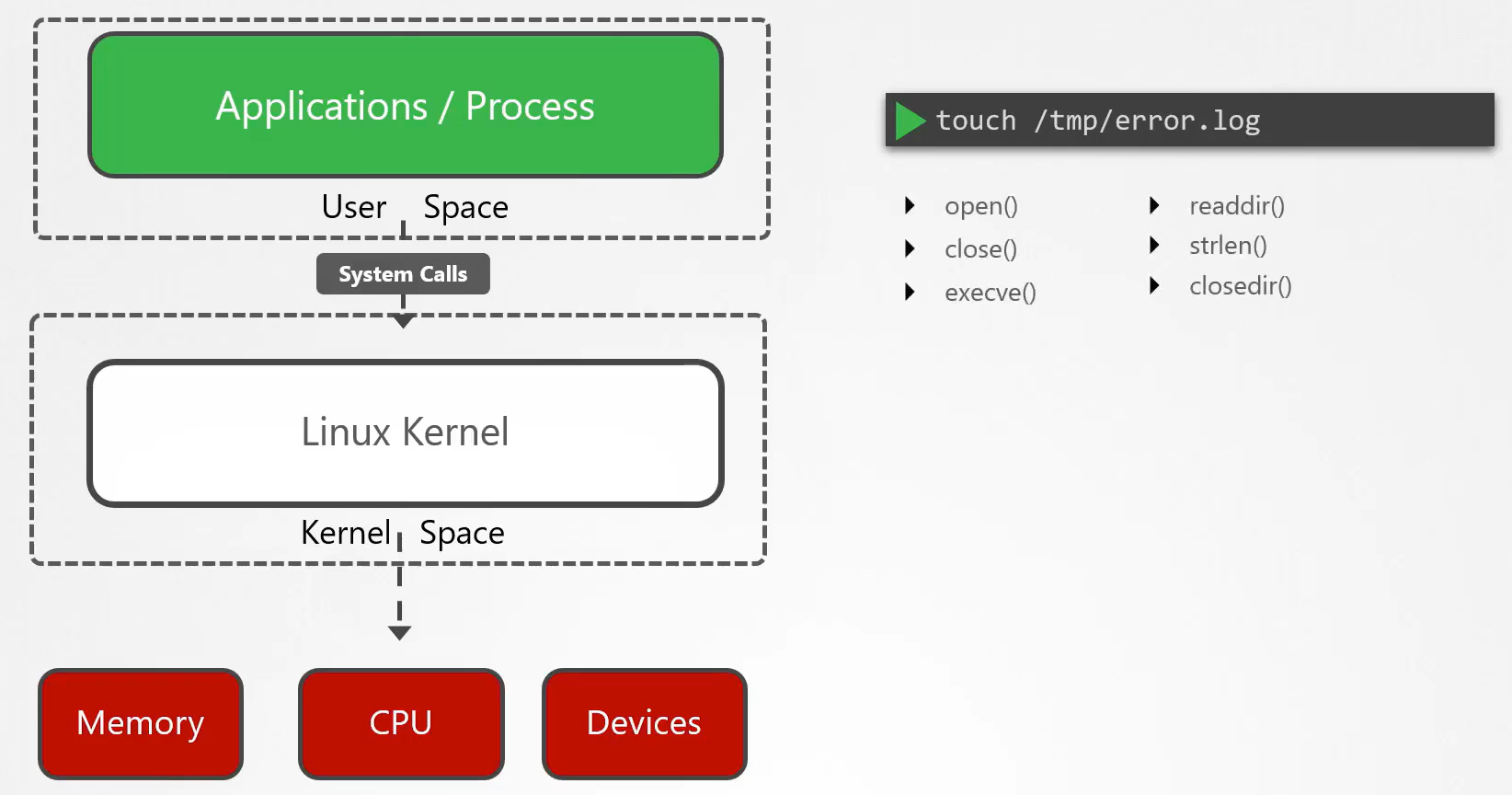
**Linux OS system call security**

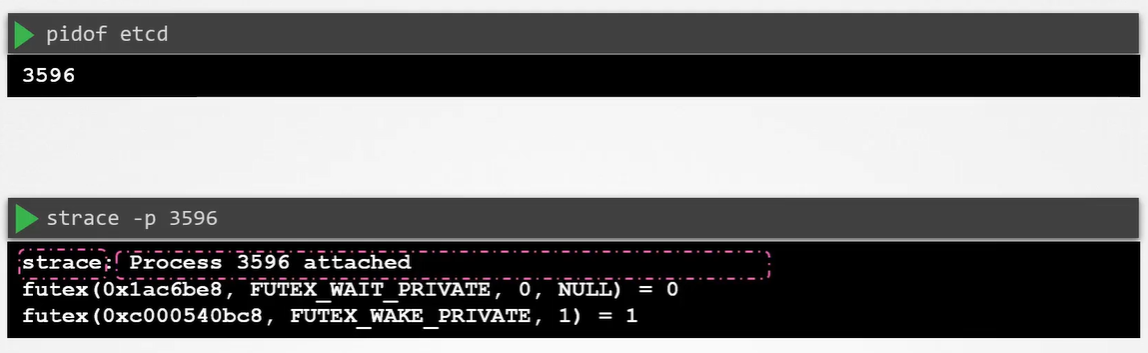


Tracing syscalls

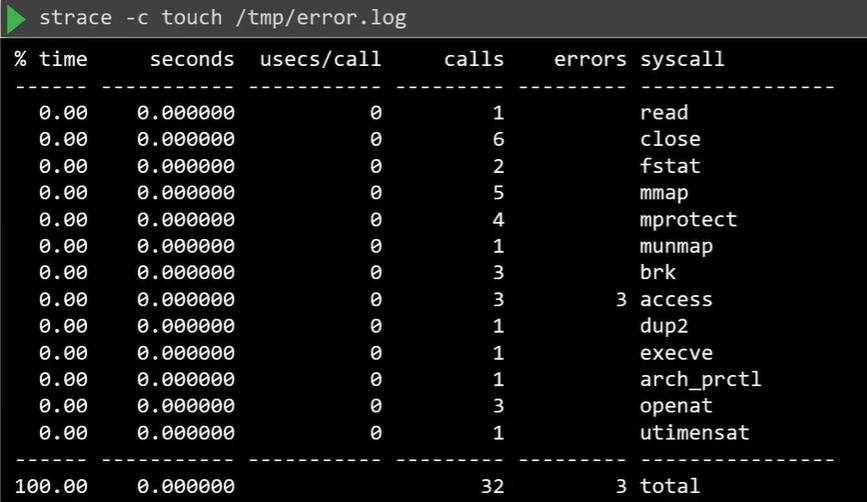


One of the syscalls used by the command

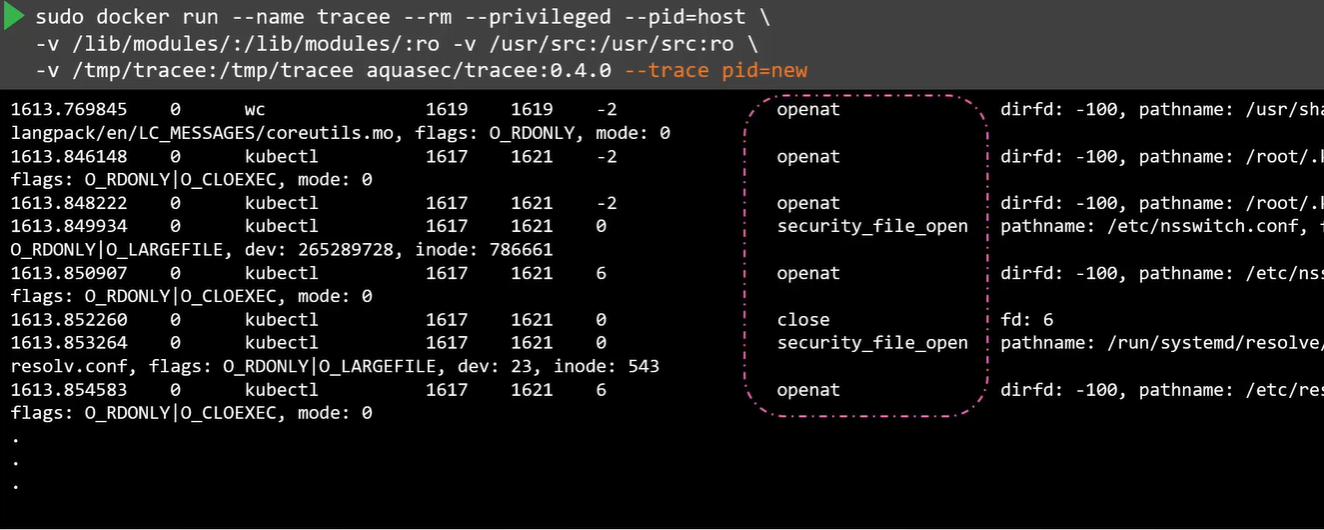




Summary of all syscalls used by the command



Another tool: AquaSec Tracee

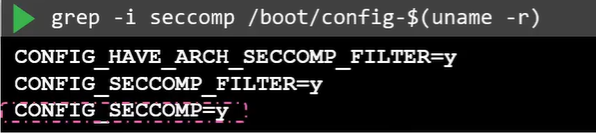


Restrict the app only make use of SYSCALLS that they absolutely need.

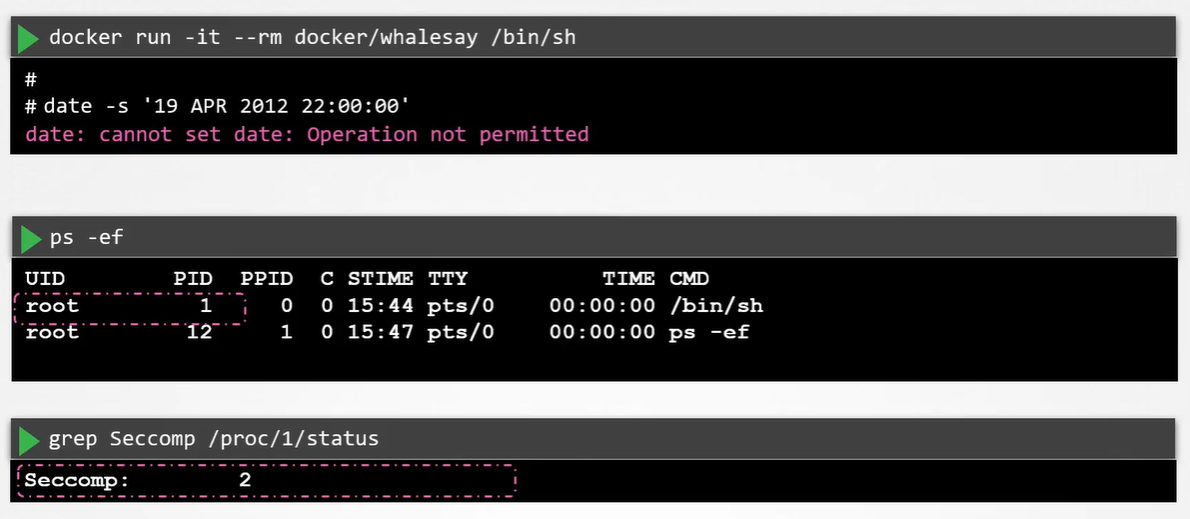
By default, the linux kernel will allow any SYSCALLS to be invoked by programs running inside the user-space which can increase the attack surface

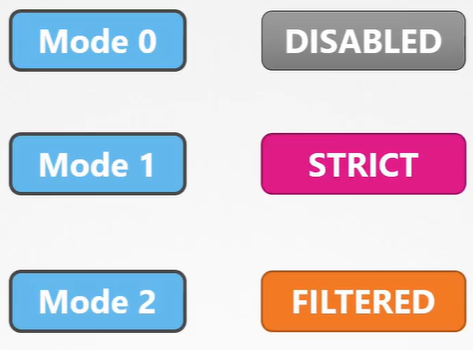
Use SECCOMP, which stands for secure computing and it’s a linux kernel level feature that can be used to sandbox applications to only use the SYSCALLS they need.

Docker has a built-in SECCOMP filter



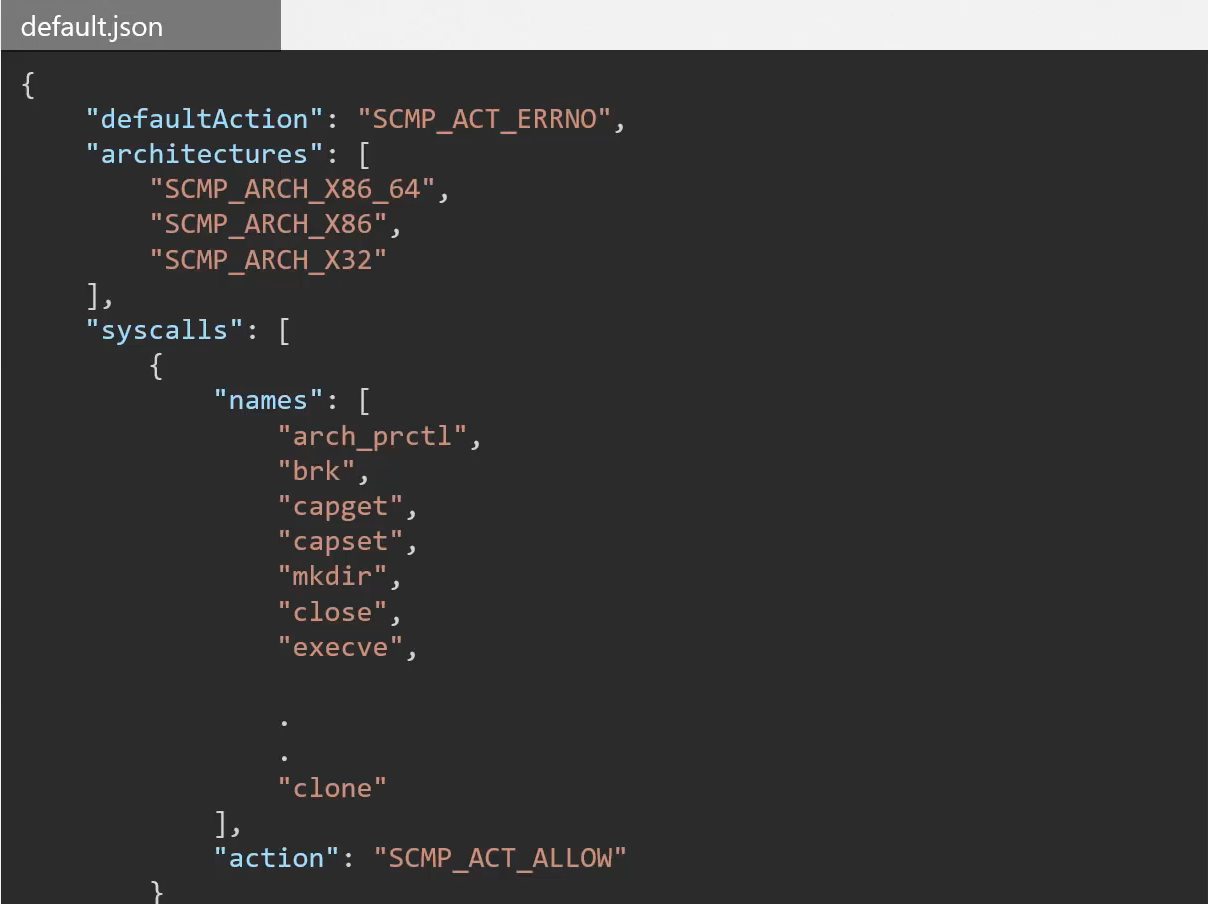
SECCOMP is active

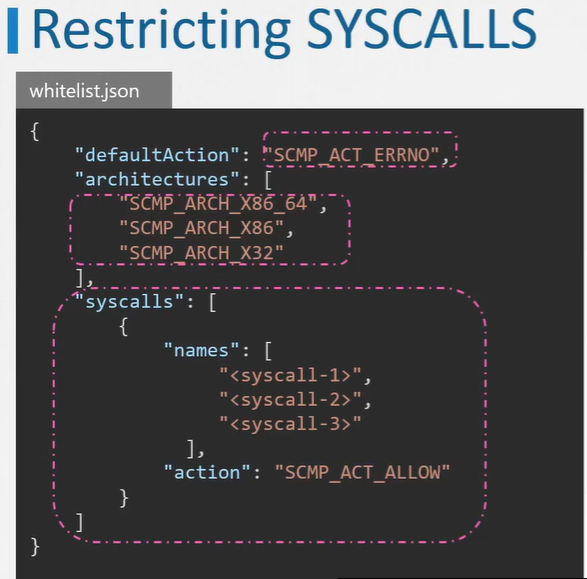




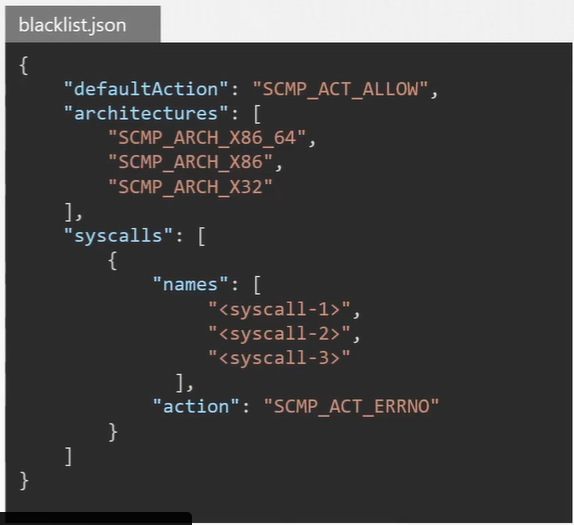
MODE1: strict mode that will block almost all SYSCALLS except read, write, exit and cigarette on SYSCALLS

MODE2 : Selectively filters SYSCALLS





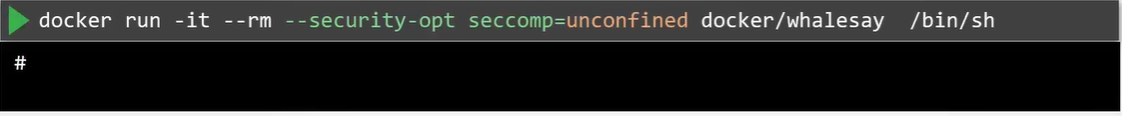
“SCMP\_ACT\_ERRNO”: reject all other SYSCALLS by default



Docker default blocks 60+ of the 300+ SYSCALLS on the X86 architecture



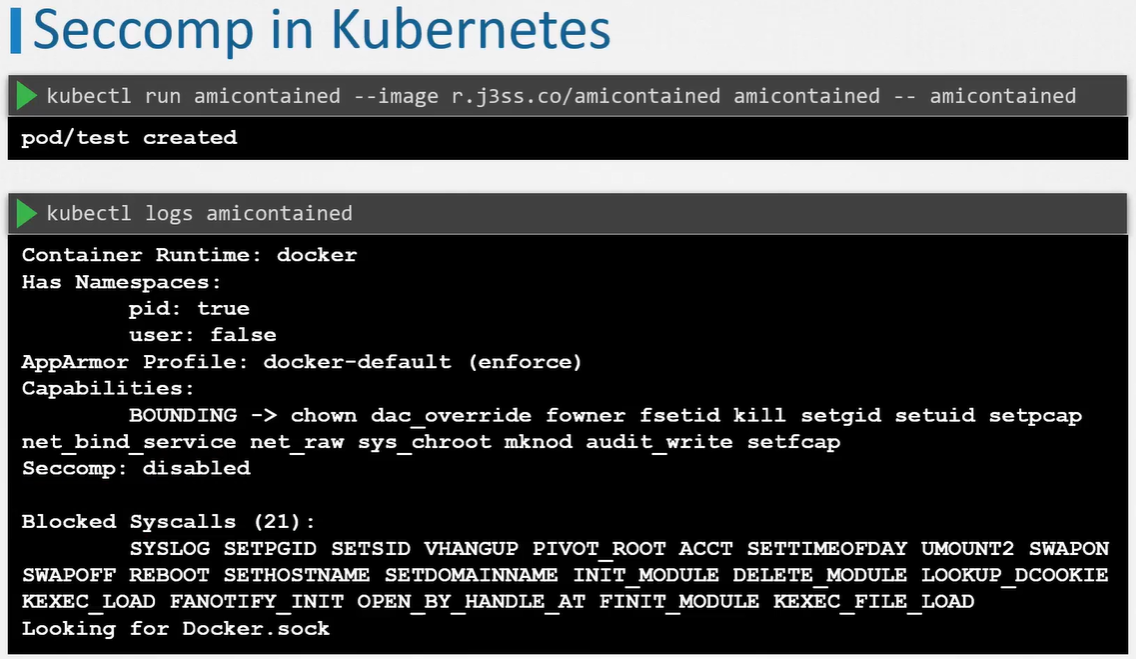
Disable SECCOMP(make use of all the SYSCALLS within the container)

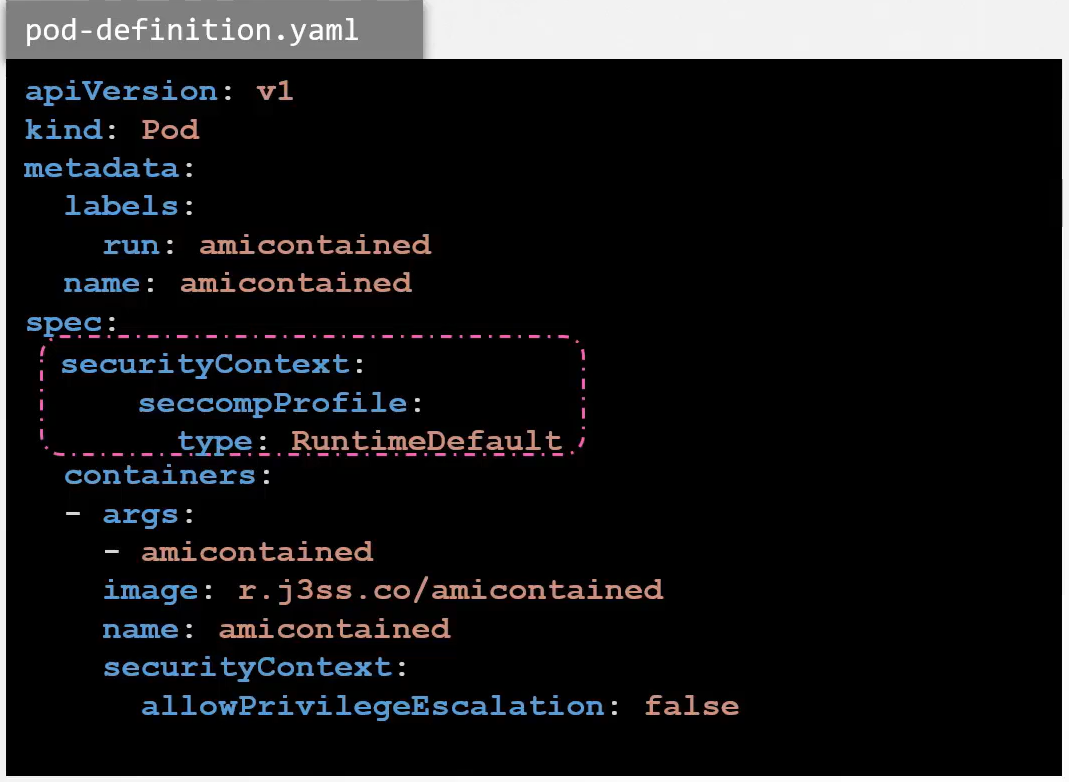


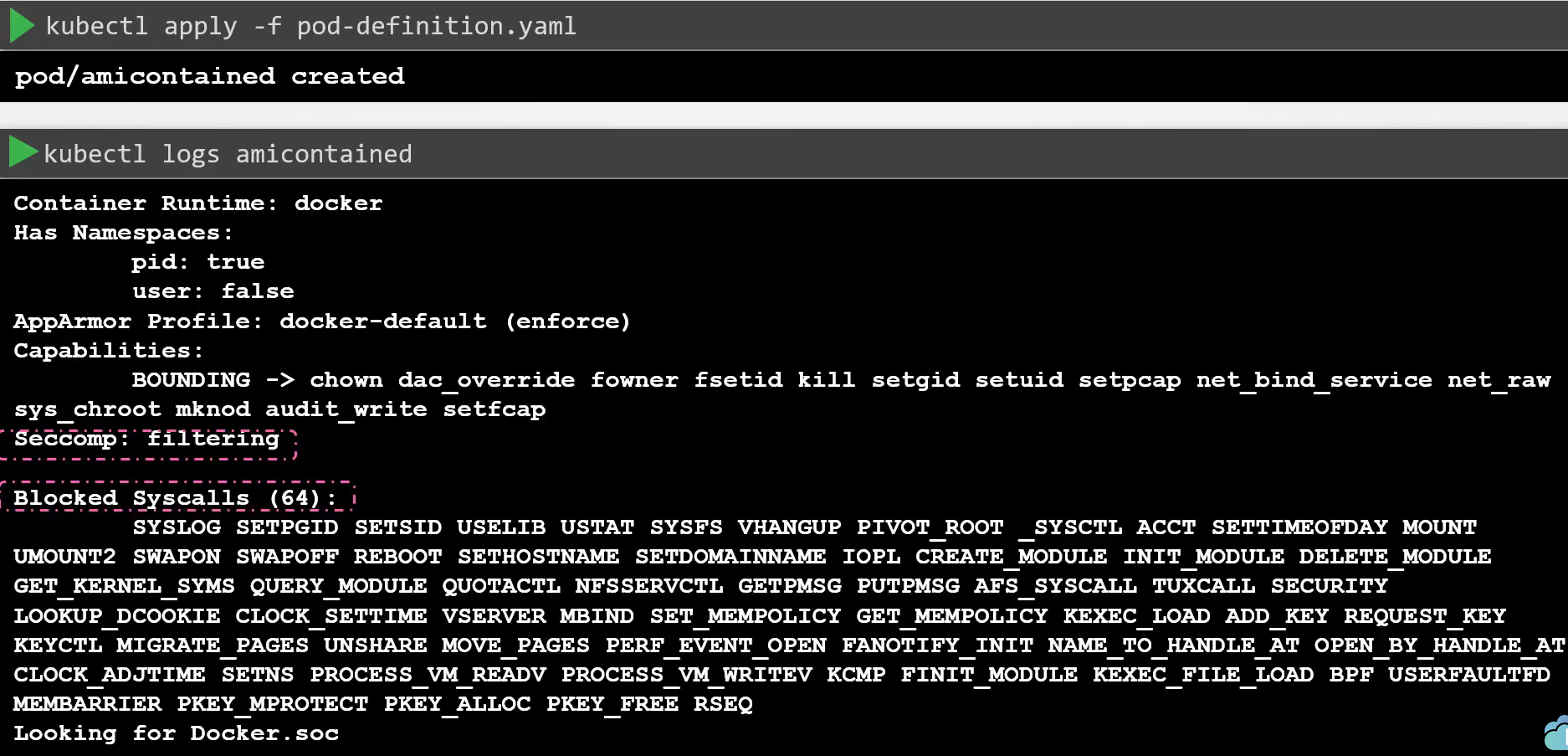
SECCOMP in Kubernetes

Run container to check which systemcalls are restricted.

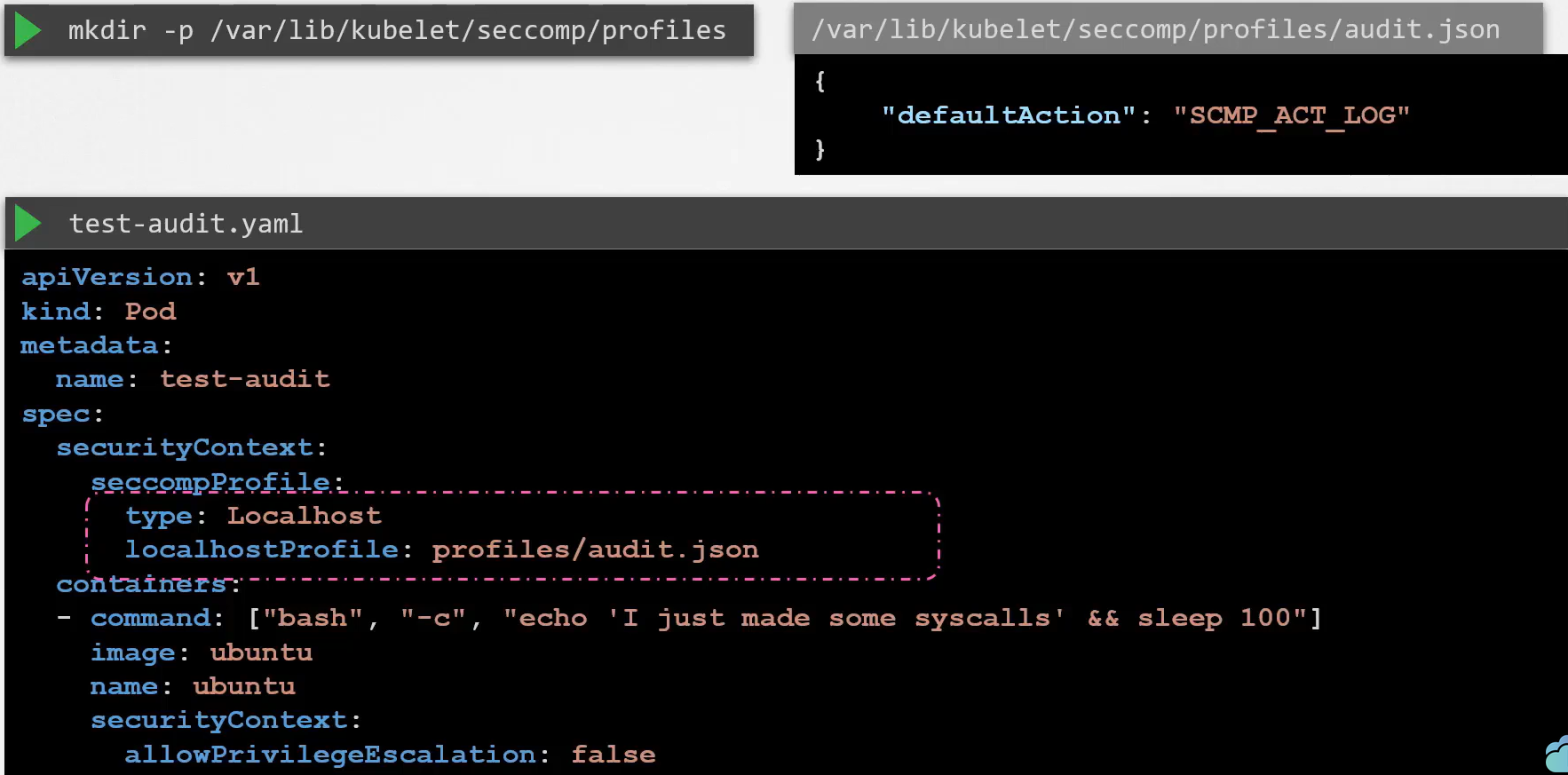
  
K8S default disable SECCOMP

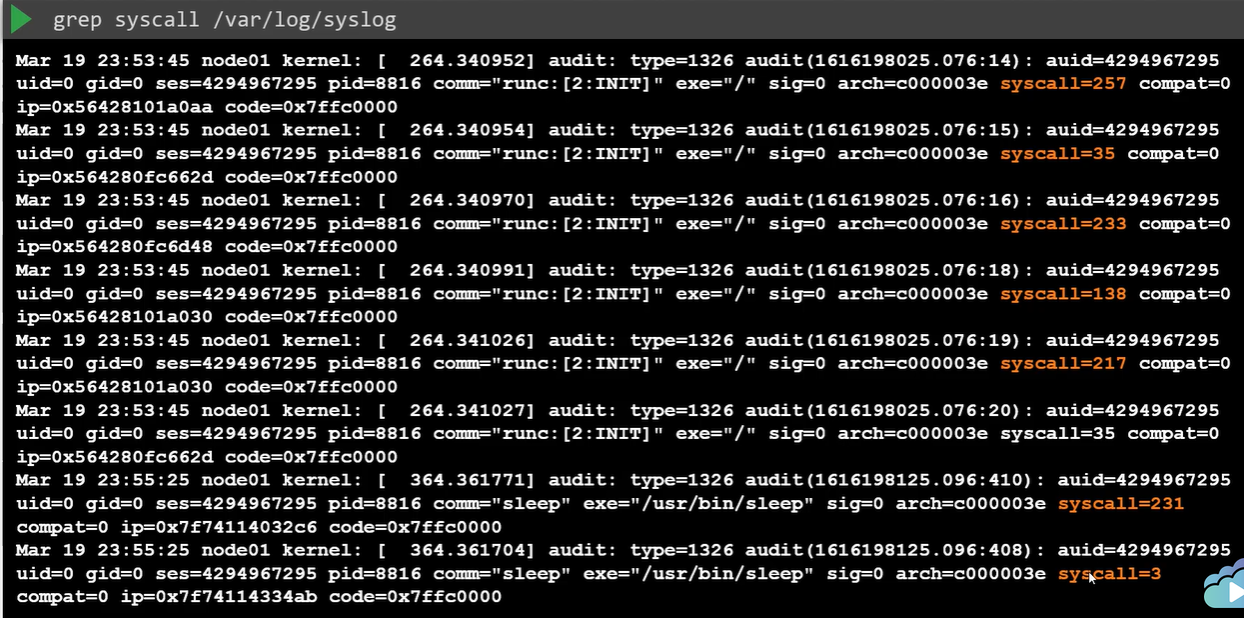




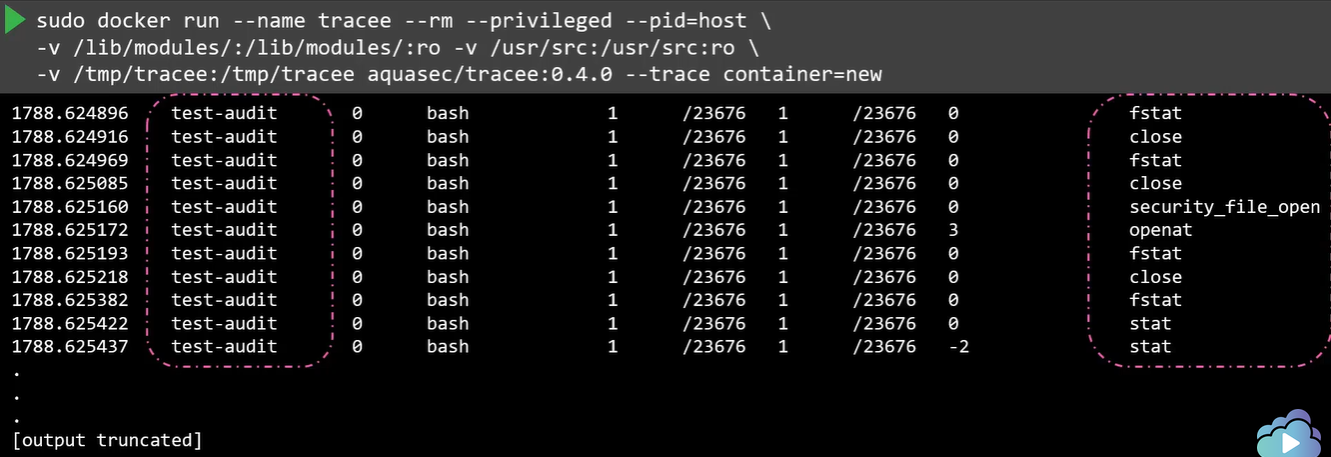


User defined SECCOMP









Reject any system call

