# Idea

How can gradebot prevent mal-homework from poaching testcase?

now it's using lxd, with no internet, but recall your fisrt hw.

# Philociphy

lxd: If you can't see the world, you can't destory it.

pros: Internet can be restricted to certain bandwidth.

cons: Internet is a yes or no question.( you either see it or not)

basic assumption: without syscall, you can do nothing dangerous.

ours: If you need my permission for every syscall you make, you can never destory the world because I won't allow it.

Same thing to the Internet, if you are only allowed to connect to certain ip, then everyting's fine.

# Implmentation

tracer & safe-box

Incapulsalization

struct tracee {

pid ( The most important thing)

blabla

}

tracee.deny();

file

we focus on open syscall rather write or read.

you got your testcase by file stream, but can't open a file to record it.

those outside /home are permitted. ( os will stop them anyway. )

those inside pwd are also permitted.

those outside pwd but inside /home will need conf file.

ip

we focus on connect, mostly client part.

client:

socket -> connect -> send <-> receive

server:

socket -> bind -> listen -> accept -> send <-> receive

only allowed ip will do.

# Demo

File: file.c

Internet:

lxd + wget

lxd + git clone

Future work

as to open file control, we hope we can check for file mode later.

we want to future trace dns server data to make sure that the programme is not access bad things in the same domain. (for example, the homework goes to github and download some third-party library.)

we hope we can focus on accept so that safe-box can be used on server application.

# Q&A

# File Syscall Intel x86-64

rax rdi

0 sys\_read unsigned int fd char \*buf size\_t count

1 sys\_write unsigned int fd const char \*buf size\_t count

2 sys\_open const char \*filename int flags int mode

3 sys\_close unsigned int fd

4 sys\_stat const char \*filename struct stat \*statbuf

5 sys\_fstat unsigned int fd struct stat \*statbuf

6 sys\_lstat fconst char \* filename struct stat \*statbuf

7 sys\_poll struct poll\_fd \*ufds unsigned int nfds long timeout\_msecs

8 sys\_lseek unsigned int fd off\_t offset unsigned int origin

17 sys\_pread64 unsigned long fd char \*buf size\_t count loff\_t pos

18 sys\_pwrite64 unsigned int fd const char \*buf size\_t count loff\_t pos

19 sys\_readv unsigned long fd const struct iovec \*vec unsigned long vlen

20 sys\_writev unsigned long fd const struct iovec \*vec unsigned long vlen

21 sys\_access const char \*filename int mode

22 sys\_pipe int \*filedes

# Network Syscall Intel x86-64

rax rdi rsi rdx r10 r8 r9

41 sys\_socket int family int type int protocol

42 sys\_connect int fd struct sockaddr \*uservaddr int addrlen

43 sys\_accept int fd struct sockaddr \*upeer\_sockaddr int \*upeer\_addrlen

44 sys\_sendto int fd void \*buff size\_t len unsigned flags struct sockaddr \*addr int addr\_len

45 sys\_recvfrom int fd void \*ubuf size\_t size unsigned flags struct sockaddr \*addr int \*addr\_len

46 sys\_sendmsg int fd struct msghdr \*msg unsigned flags

47 sys\_recvmsg int fd struct msghdr \*msg unsigned int flags

48 sys\_shutdown int fd int how

49 sys\_bind int fd struct sokaddr \*umyaddr int addrlen

50 sys\_listen int fd int backlog

51 sys\_getsockname int fd struct sockaddr \*usockaddr int \*usockaddr\_len

52 sys\_getpeername int fd struct sockaddr \*usockaddr int \*usockaddr\_len

53 sys\_socketpair int family int type int protocol int \*usockvec

54 sys\_setsockopt int fd int level int optname char \*optval int optlen

55 sys\_getsockopt int fd int level int optname char \*optval int \*optlen