

A large black MSC cargo ship is engulfed in thick, billowing black smoke and flames, appearing to be on fire. The ship is positioned in the lower half of the frame, with its name 'MSC' visible on the hull. The background is a dark, overcast sky.

containers / security / a fun time

pick two

Aleksa (cyphar)

CC BY-SA 4.0





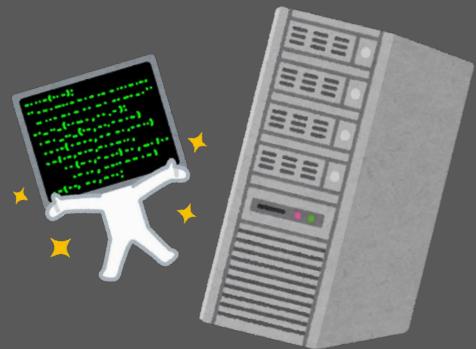
A large grid of shipping containers, approximately 10 columns by 10 rows, is shown. The containers are stacked in a staggered pattern. Many of the containers have "MSC" printed on them, indicating they belong to the Mediterranean Shipping Company. Other containers have "XINES" or "GOLD" printed on them. The colors of the containers vary, including yellow, blue, red, green, and brown. Some containers have additional markings like "NET", "CU.CAP.", "TARE", and "GLU".

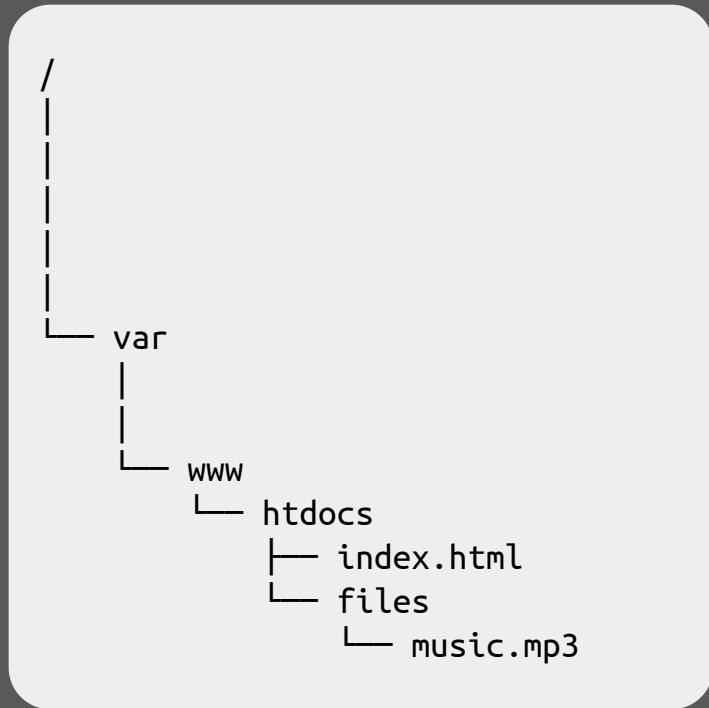
who's heard of containers?





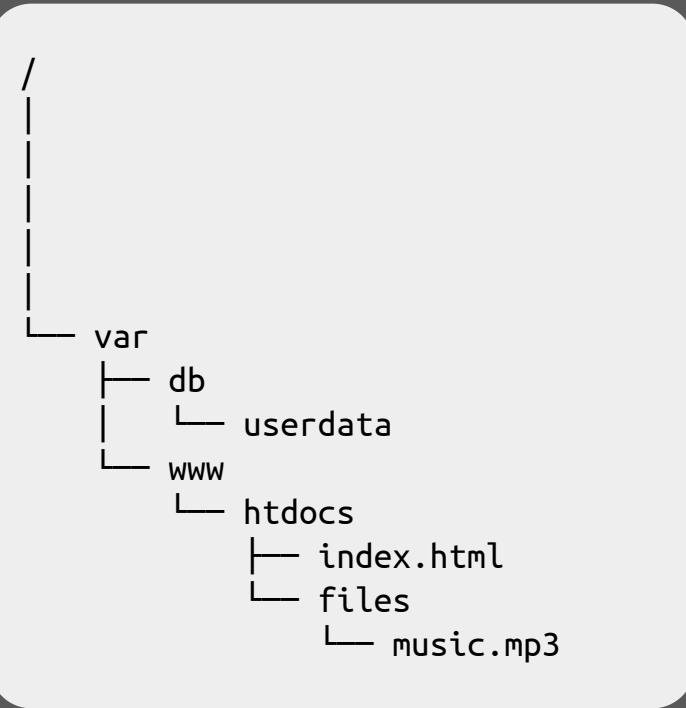
let's build a website!







```
/  
└ var  
    └ db  
        └ userdata  
└ www  
    └ htdocs  
        └ index.html  
    └ files  
        └ music.mp3
```





```
/  
└ etc  
    └ shadow  
  
└ var  
    └ db  
        └ userdata  
    └ www  
        └ htdocs  
            └ index.html  
        └ files  
            └ music.mp3
```



```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



Can I GET
files/music.mp3?



```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



Can I GET
files/music.mp3?

Hmm, where is
files/music.mp3?



```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



Can I GET
files/music.mp3?

Ah, here it is!



```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



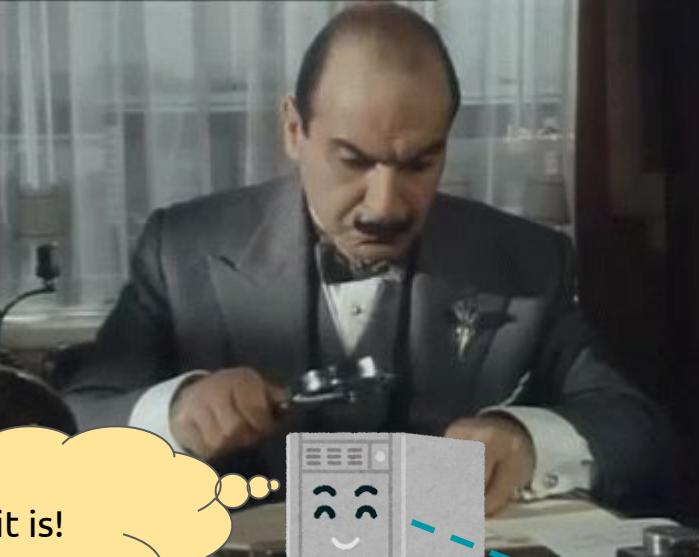
Can I GET
files/music.mp3?

Here you go!



```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```

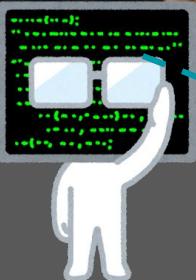
Ah, here it is!



```
hadow  
yphar  
— secrets.txt  
b  
— userdata  
WW  
— htdocs  
   — index.html  
   — files  
      — music.mp3
```



Ah, here it is!

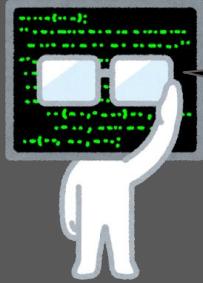


```
hadow  
yphar  
— secrets.txt  
b  
— userdata  
WW  
— htdocs  
   — index.html  
   — files  
      — music.mp3
```

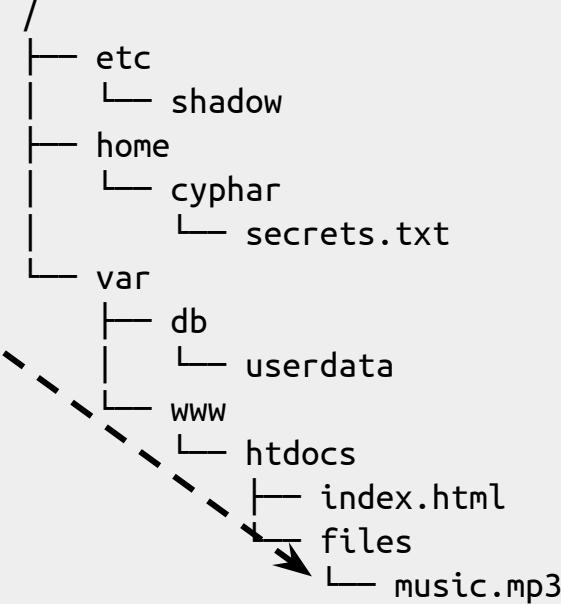


```
open("files/music.mp3")
```

```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```

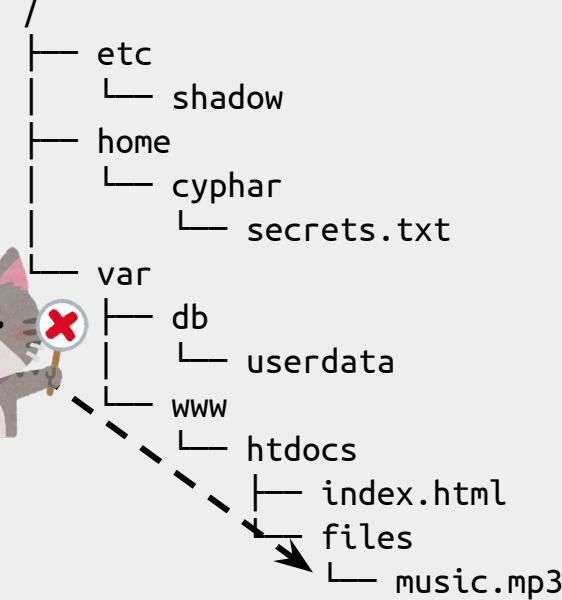


```
open("files/music.mp3")
```





`open("files/music.mp3")`

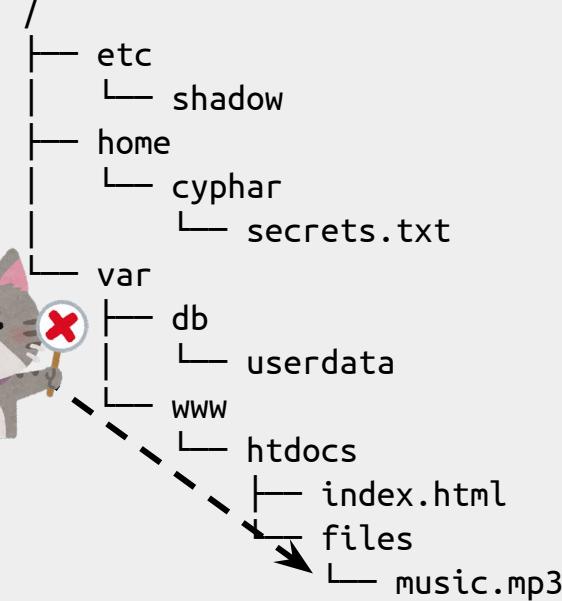


it's all about the kernel!



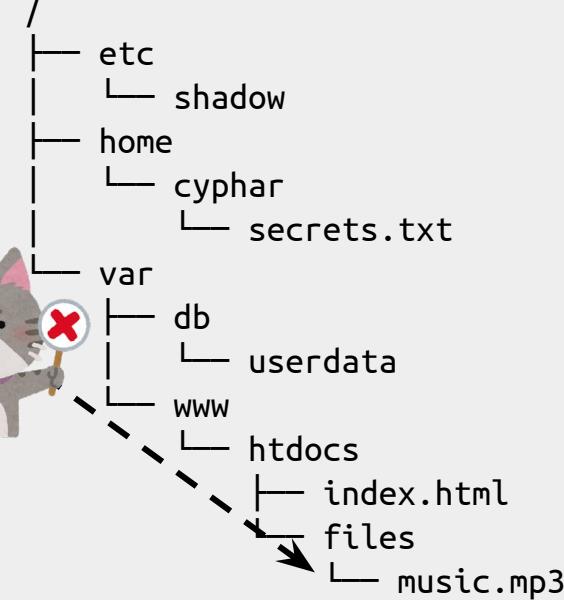


`open("files/music.mp3")`





`open("files/music.mp3")`



the kernel provides
abstractions



the kernel provides
abstractions
(among other things)





```
open("files/music.mp3")
```



```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



`open("files/music.mp3")`

`open`



```
/  
|   └── etc  
|       └── shadow  
|   └── home  
|       └── cyphar  
|           └── secrets.txt  
└── var  
    └── db  
        └── userdata  
            └── www  
                └── htdocs  
                    └── index.html  
                        └── files  
                            └── music.mp3
```



`open("files/music.mp3")`

`open`

Let's look up
`files/music.mp3`...



```
/  
  |- etc  
  |  └ shadow  
  |- home  
  |  └ cyphar  
  |    └ secrets.txt  
  |- var  
  |  |- db  
  |    └ userdata  
  |- www  
  |    |- htdocs  
  |      |- index.html  
  |      └ files  
        └ music.mp3
```

`./files/music.mp3`



`open("files/music.mp3")`

`open`

Let's look up
`files/music.mp3`...



```
/  
  |- etc  
  |  └ shadow  
  |- home  
  |  └ cyphar  
  |    └ secrets.txt  
  |- var  
  |  |- db  
  |    └ userdata  
  |- www  
  |    |- htdocs  
  |      |- index.html  
  |      └ files  
        └ music.mp3
```



`open("files/music.mp3")`

Let's look up
`files/music.mp3`...



`open`



`./files/music.mp3`

```
/  
  |- etc  
  |  └ shadow  
  |- home  
  |  └ cyphar  
  |    └ secrets.txt  
  |- var  
  |  |- db  
  |    └ userdata  
  |- www  
  |    |- htdocs  
  |      |- index.html  
  |      └ files  
        └ music.mp3
```



./files/music.mp3



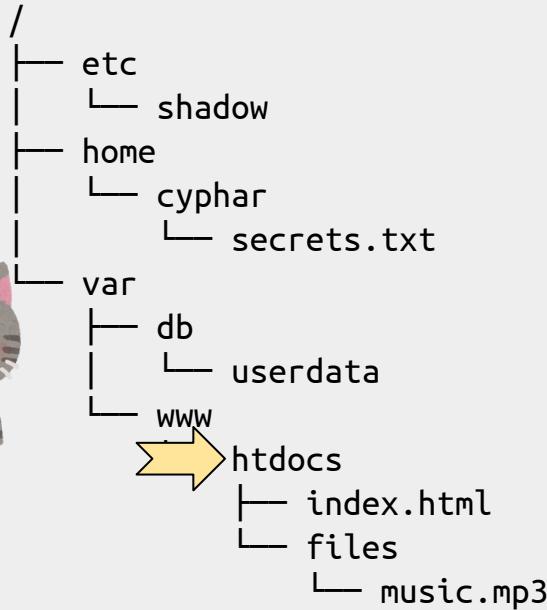
open("files/music.mp3")

open



Let's look up
files/music.mp3...

Program's working
directory is
/var/www/htdocs.



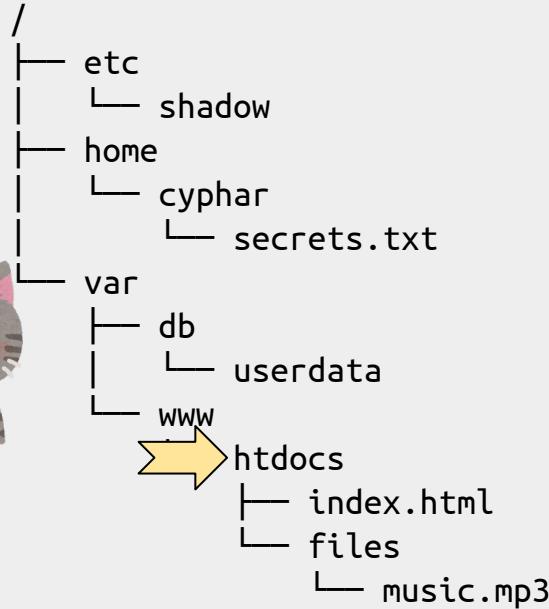
./files/music.mp3



open("files/music.mp3")

open

Let's look up
files/music.mp3...



./files/music.mp3

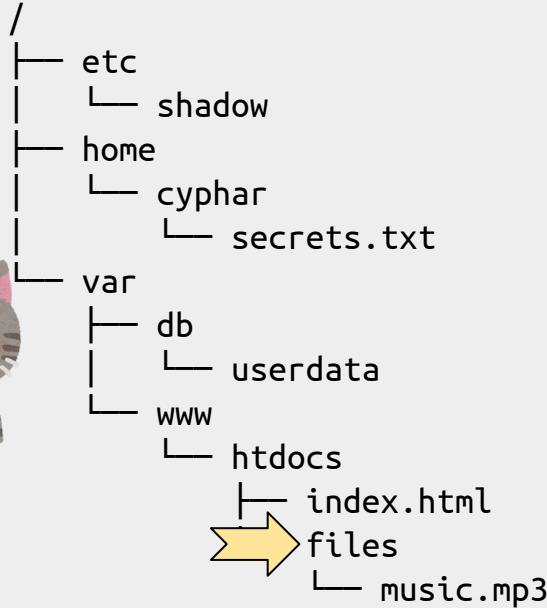


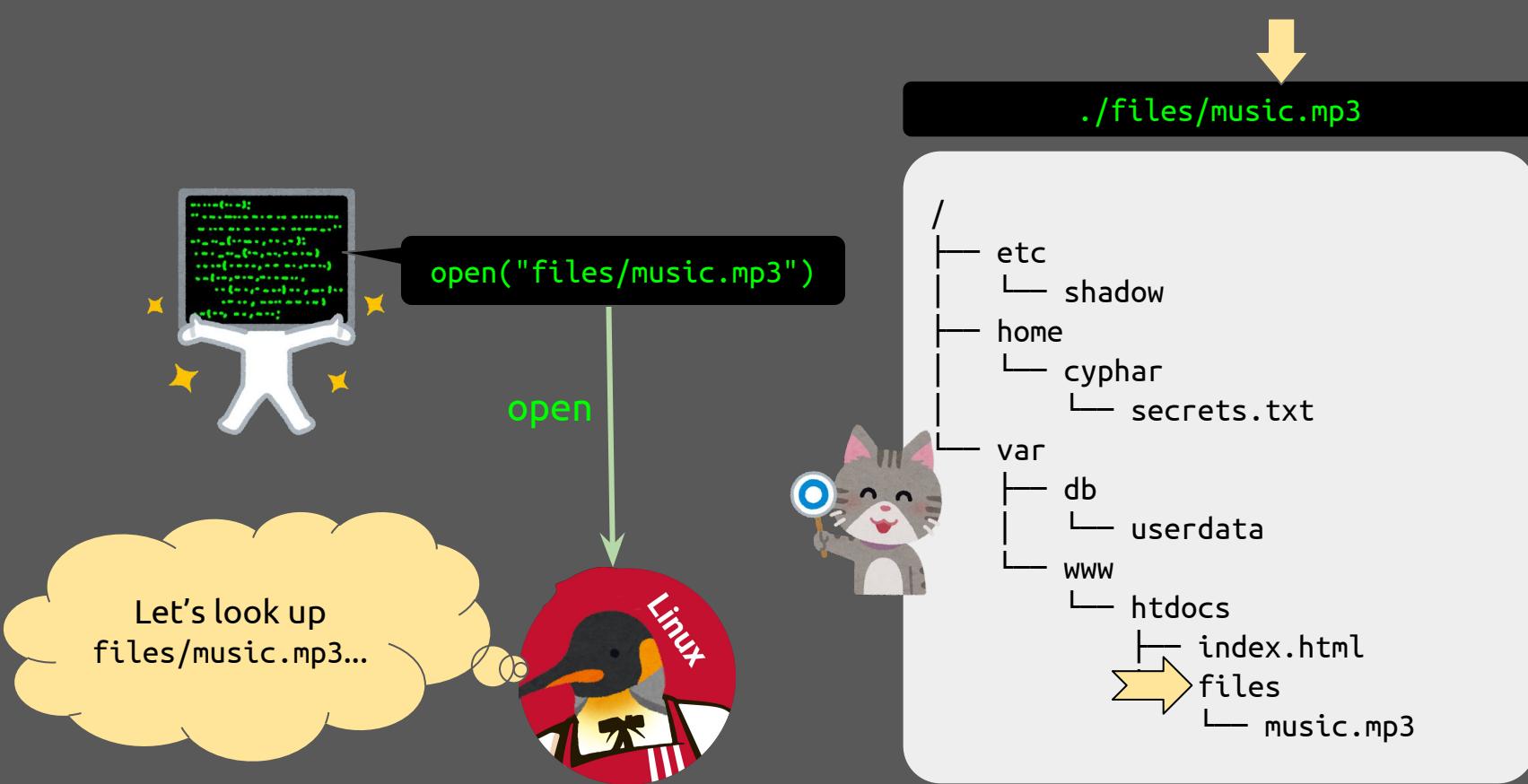
open("files/music.mp3")

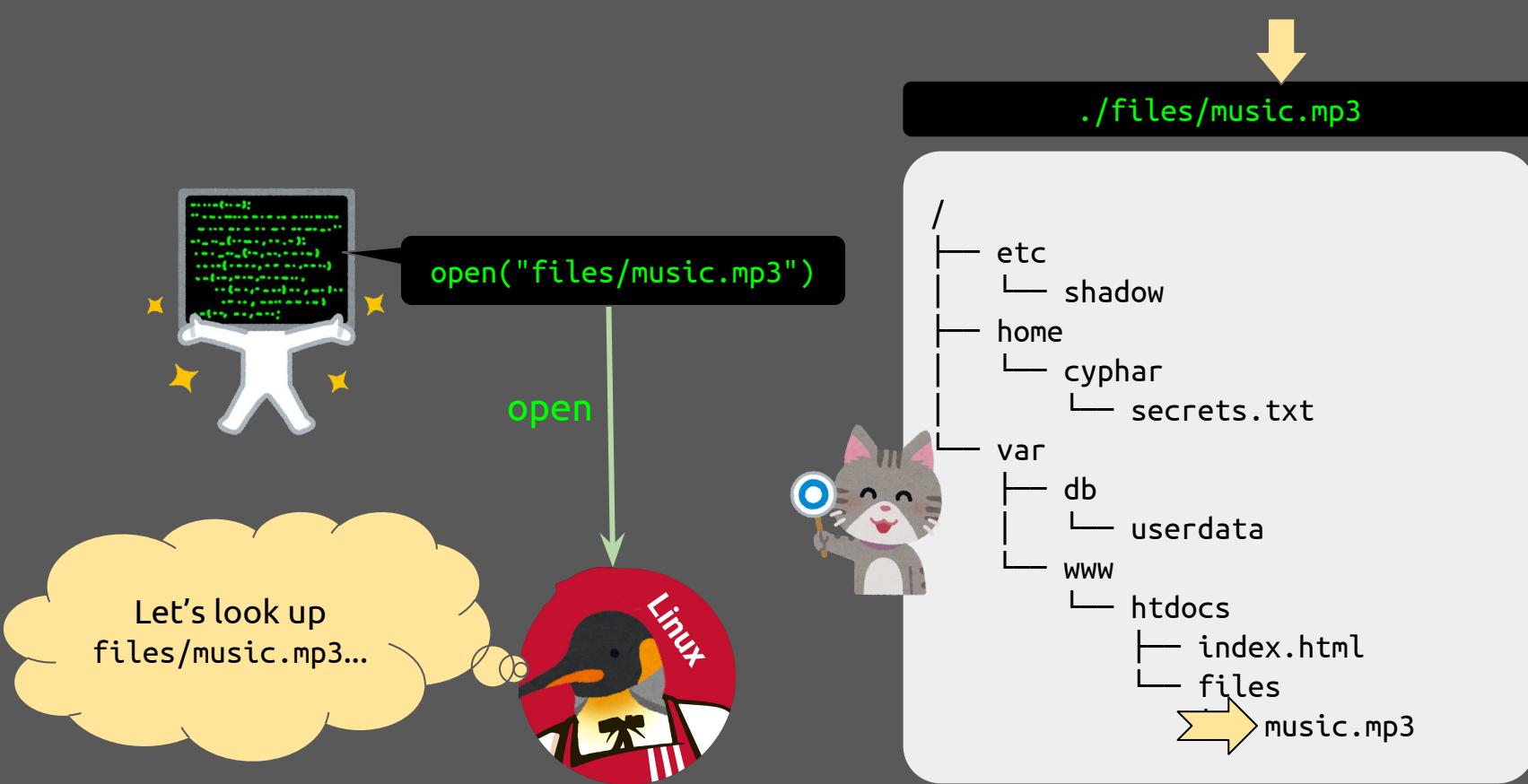
open



Let's look up
files/music.mp3...





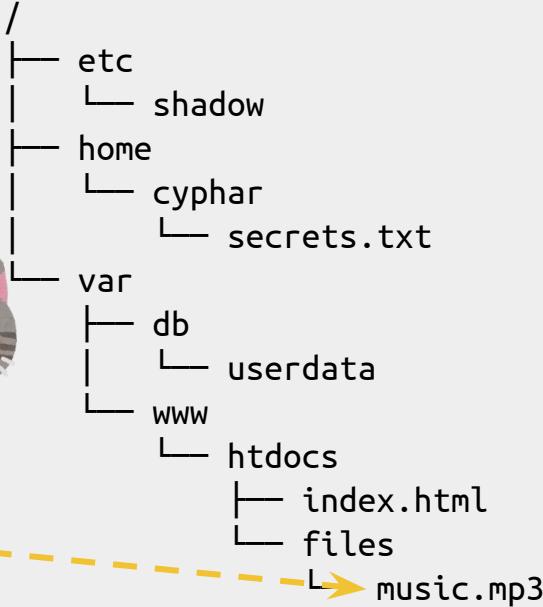


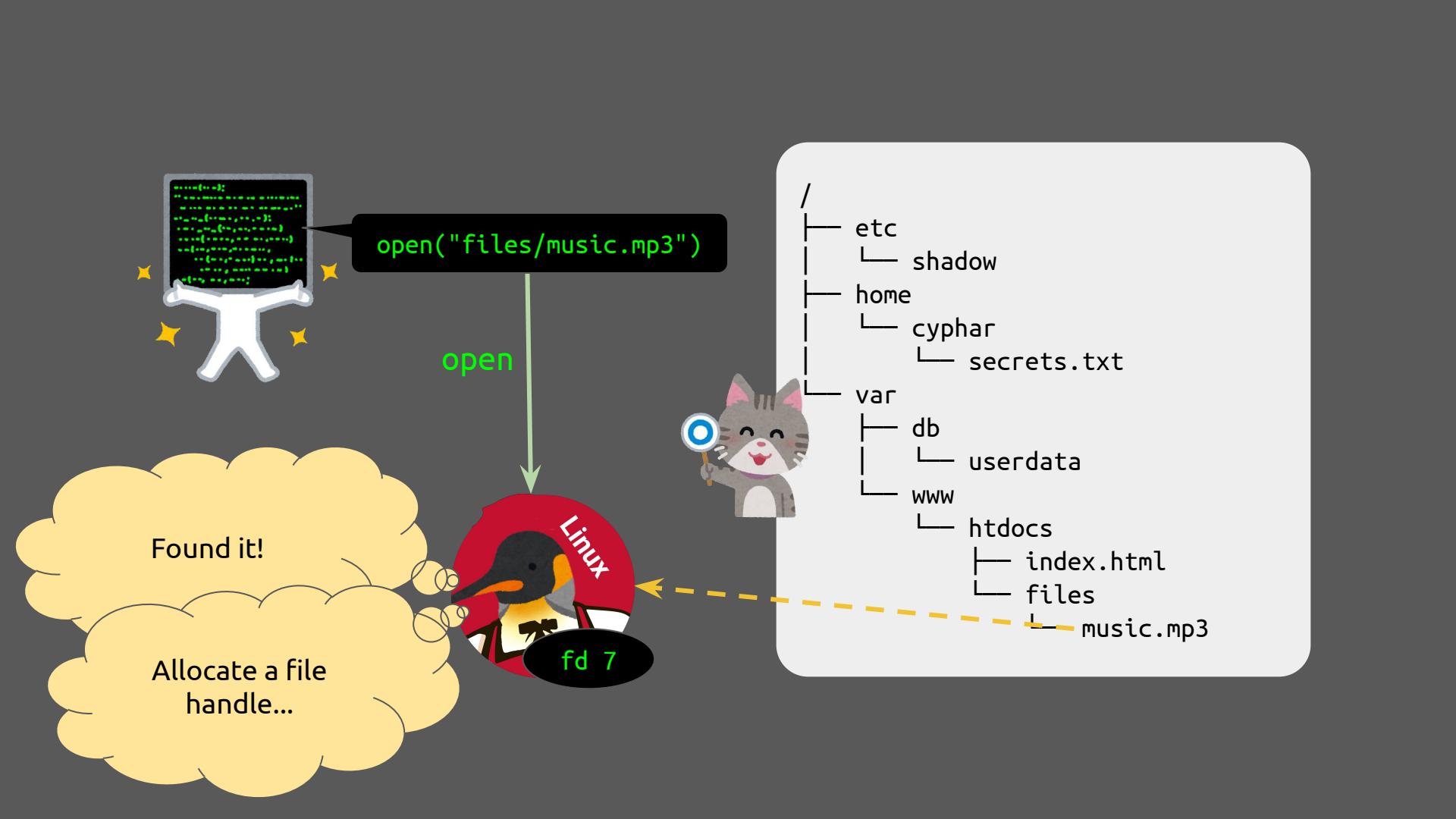


`open("files/music.mp3")`

`open`

Found it!







`open("files/music.mp3")`

`open`

`fd 7`



```
/  
|   └── etc  
|       └── shadow  
|  
|   └── home  
|       └── cyphar  
|           └── secrets.txt  
|  
└── var  
    └── db  
        └── userdata  
            └── www  
                └── htdocs  
                    └── index.html  
                        └── files  
                            └── music.mp3
```

Wow, that was easy!







I made this.

fd 7





I made this.



You made this?

fd 7





fd 7



I made this.

fd 7



I hope nothing
will go wro—

fd 7

A new foe has appeared



(or)



CHALLENGER APPROACHING





```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



Would you kindly GET
../../../../home/cyphar/secrets.txt?

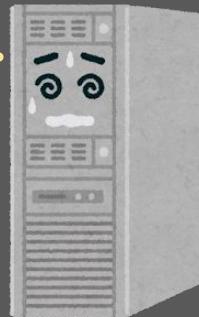


```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



Would you kindly GET
../../../../home/cyphar/secrets.txt?

Hmm, where is
that file...

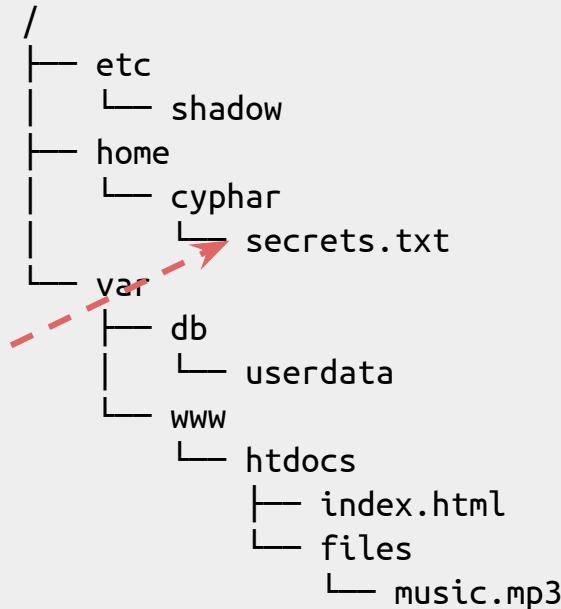


```
/  
└── etc  
    └── shadow  
└── home  
    └── cyphar  
        └── secrets.txt  
└── var  
    └── db  
        └── userdata  
└── www  
    └── htdocs  
        ├── index.html  
        └── files  
            └── music.mp3
```



Would you kindly GET
../../../../home/cyphar/secrets.txt?

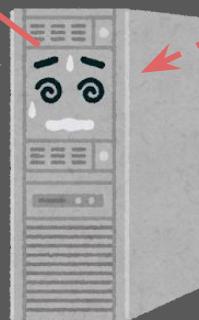
Ah, here it is...?





Would you kindly GET
../../../../home/cyphar/secrets.txt?

Here you go!



```
/  
└── etc  
    └── shadow  
└── home  
    └── cyphar  
        └── secrets.txt  
└── var  
    └── db  
        └── userdata  
└── www  
    └── htdocs  
        ├── index.html  
        └── files  
            └── music.mp3
```



```
open("../../../../../home/cyphar/secrets.txt")
```

open



```
/  
|   └── etc  
|       └── shadow  
|   └── home  
|       └── cyphar  
|           └── secrets.txt  
└── var  
    ├── db  
    |   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



```
../../../../../../../../home/cyphar/secrets.txt
```

```
/  
└── etc  
    └── shadow  
└── home  
    └── cyphar  
        └── secrets.txt  
└── var  
    └── db  
        └── userdata  
└── www  
    └── htdocs  
        ├── index.html  
        └── files  
            └── music.mp3
```



```
open("../../../../../home/cyphar/secrets.txt")
```

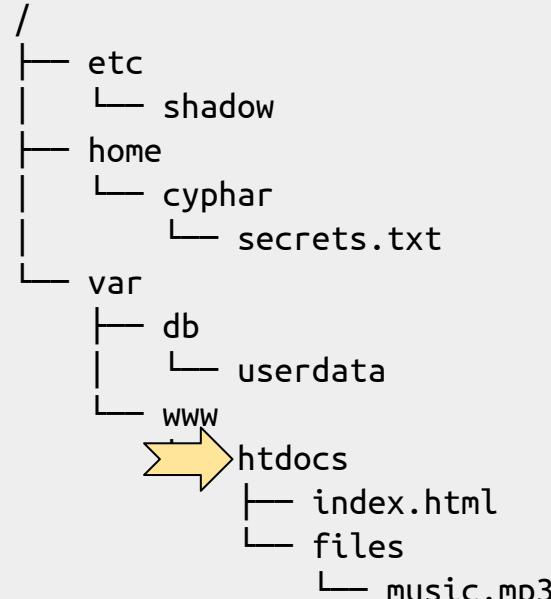
Let's look up
../../../../home/cyphar/
secrets.txt...

Program's working
directory is
/var/www/htdocs.

open



```
./../../../../home/cyphar/secrets.txt
```





```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



```
../../../../../../../../home/cyphar/secrets.txt
```

```
/  
└── etc  
    └── shadow  
└── home  
    └── cyphar  
        └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



```
open("../../../../../home/cyphar/secrets.txt")
```

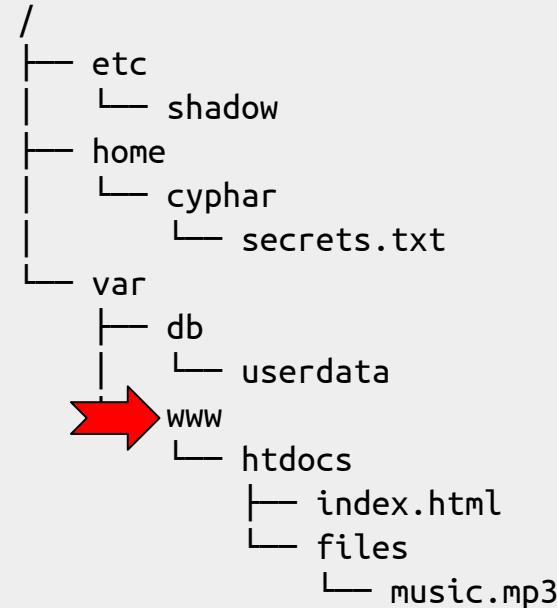
Let's look up
../../../../home/cyphar/
secrets.txt...



open



```
../../../../../../../../home/cyphar/secrets.txt
```





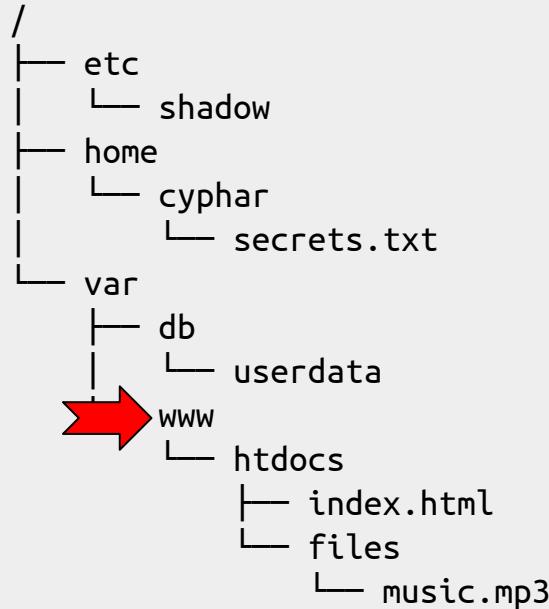
```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



open

./../../../../home/cyphar/secrets.txt



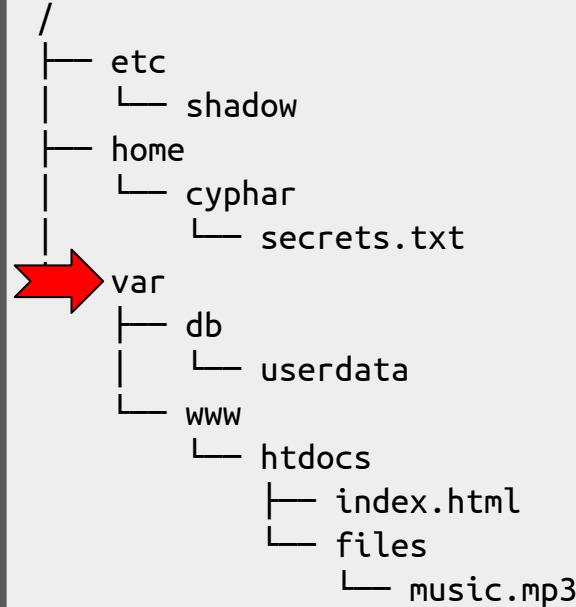


```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



```
./../../../../home/cyphar/secrets.txt
```





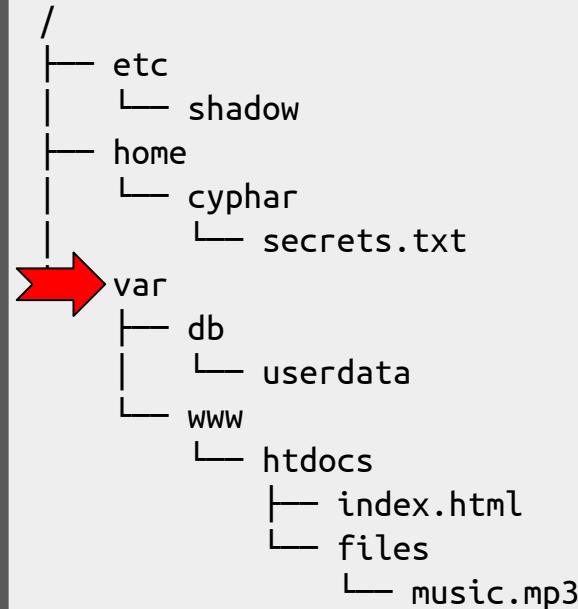
```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



open

./../../../../home/cyphar/secrets.txt





```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



```
./../../../../home/cyphar/secrets.txt
```

```
/  
└── etc  
    └── shadow  
└── home  
    └── cyphar  
        └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



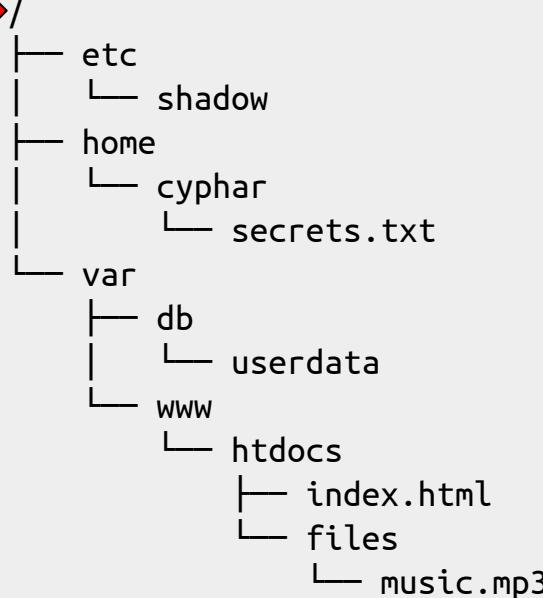
```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



```
open
```

```
./../../../../home/cyphar/secrets.txt
```



```
/  
└── etc  
    └── shadow  
└── home  
    └── cyphar  
        └── secrets.txt  
└── var  
    └── db  
        └── userdata  
└── www  
    └── htdocs  
        ├── index.html  
        └── files  
            └── music.mp3
```



```
open("../../../../../home/cyphar/secrets.txt")
```

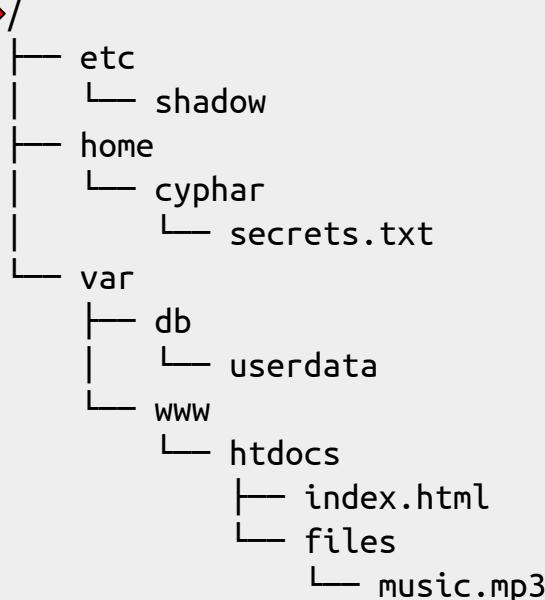
Let's look up
../../../../home/cyphar/
secrets.txt...

Okay, we've hit
the root and can't
go further.

open



```
../../../../../../../../home/cyphar/secrets.txt
```





```
open("../../../../../home/cyphar/secrets.txt")
```

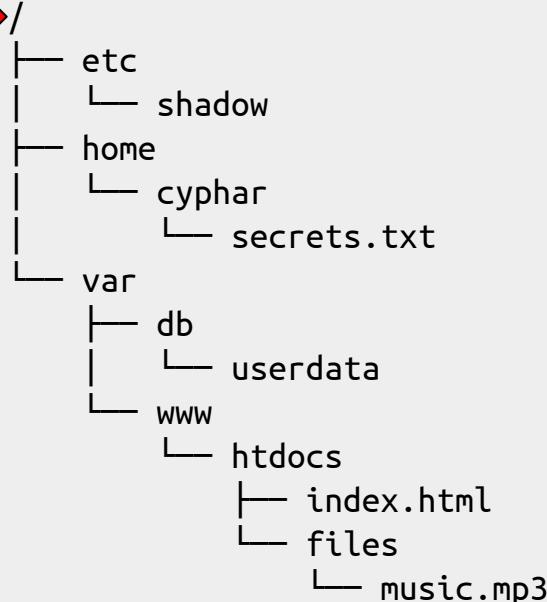
Let's look up
../../../../home/cyphar/
secrets.txt...

Okay, we've hit
the root and can't
go further.

open



```
../../../../../../../../home/cyphar/secrets.txt
```





```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



```
./../../../../home/cyphar/secrets.txt
```

```
/  
└── etc  
    └── shadow  
└── home  
    └── cyphar  
        └── secrets.txt  
└── var  
    └── db  
        └── userdata  
└── www  
    └── htdocs  
        ├── index.html  
        └── files  
            └── music.mp3
```



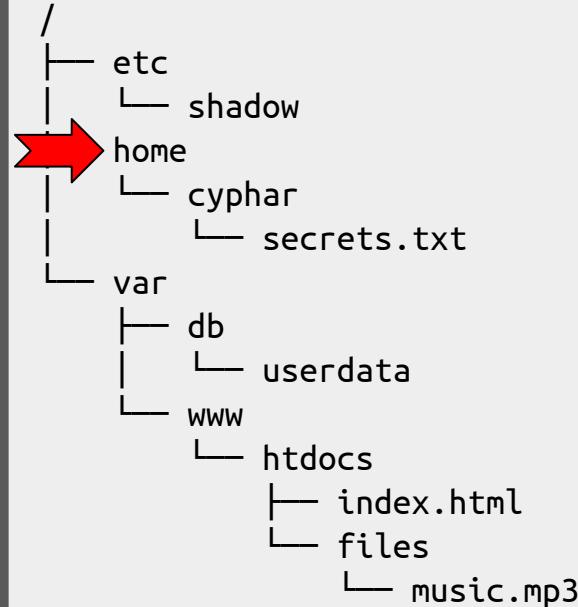
```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



open

```
./../../../../home/cyphar/secrets.txt
```





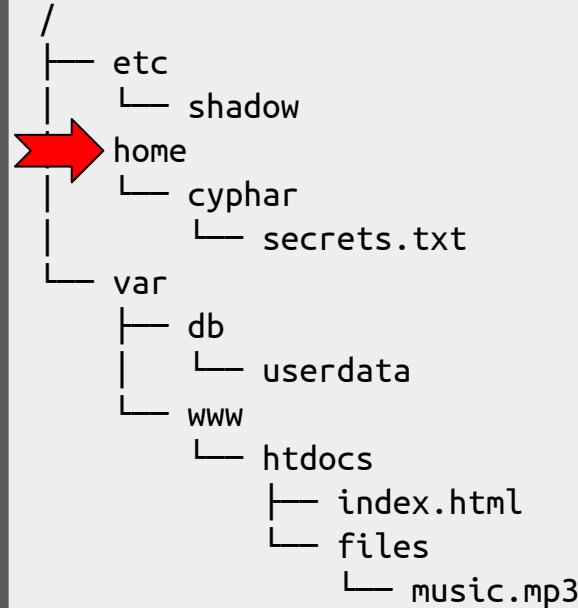
```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



open

```
../../../../../../../../home/cyphar/secrets.txt
```





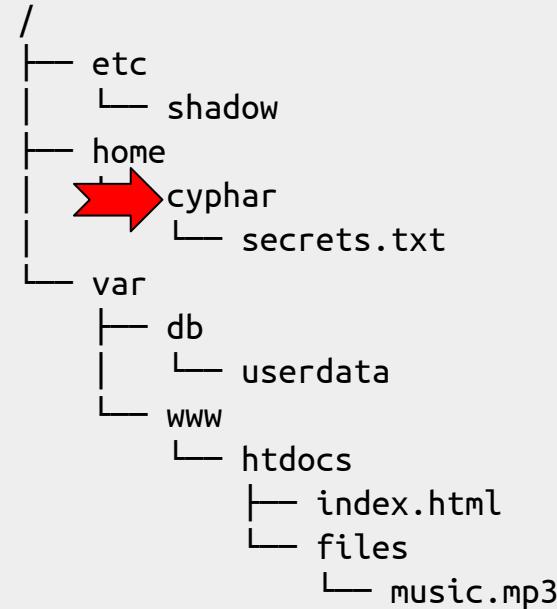
```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



open

```
../../../../../../../../home/cyphar/secrets.txt
```





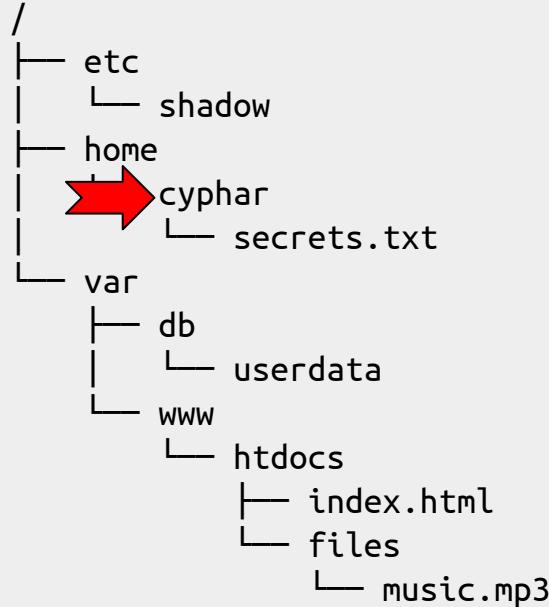
```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



open

```
../../../../../../../../home/cyphar/secrets.txt
```



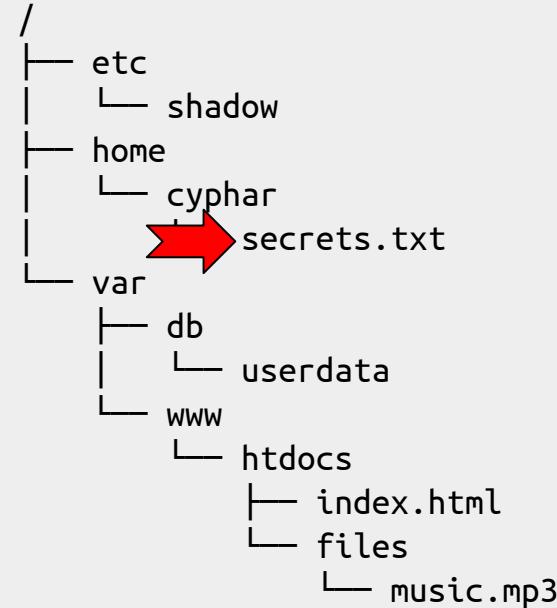


```
open("../../../../../home/cyphar/secrets.txt")
```

Let's look up
../../../../home/cyphar/
secrets.txt...



```
../../../../../../../../home/cyphar/secrets.txt
```



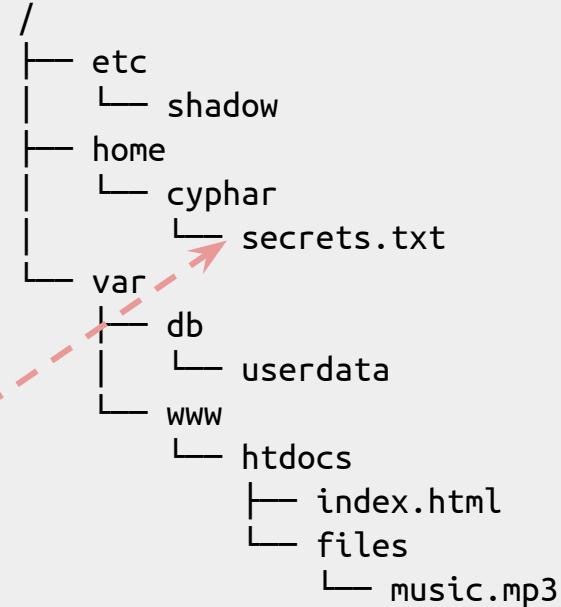


```
open("../../../../../home/cyphar/secrets.txt")
```

open



Found it!





```
open("../../../../../home/cyphar/secrets.txt")
```

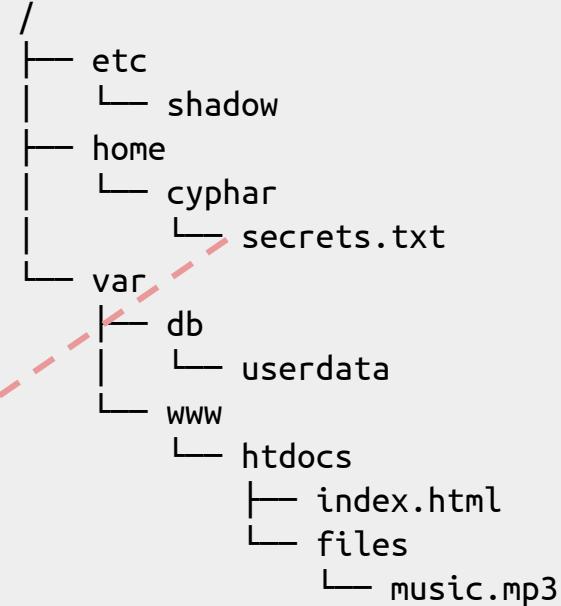
open

Linux

fd 8

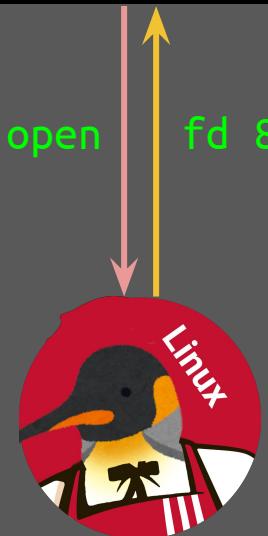
Found it!

Allocate a file
handle...



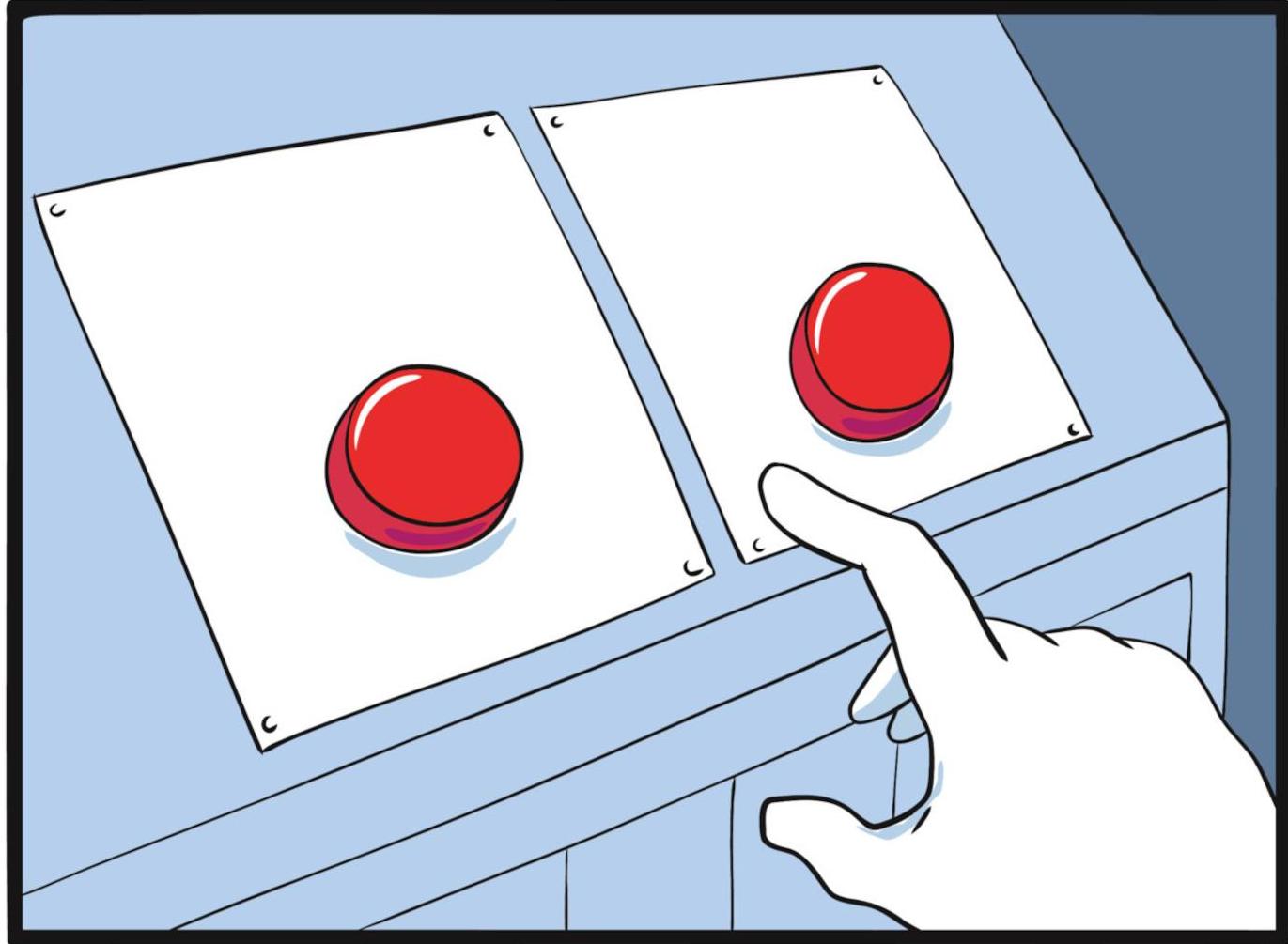


```
open("../../../../../home/cyphar/secrets.txt")
```



```
/  
|   └── etc  
|       └── shadow  
|   └── home  
|       └── cyphar  
|           └── secrets.txt  
└── var  
    └── db  
        └── userdata  
    └── www  
        └── htdocs  
            └── index.html  
            └── files  
                └── music.mp3
```







“just write bug-free software”



```
/  
├── etc  
│   └── shadow  
├── home  
│   └── cyphar  
│       └── secrets.txt  
└── var  
    ├── db  
    │   └── userdata  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



v2a



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            └── files  
                └── music.mp3
```



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            └── upload.php  
        └── files  
            └── music.mp3
```

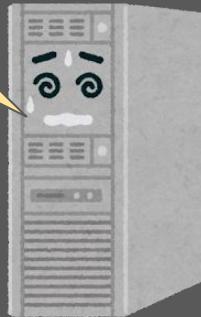


```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            └── upload.php  
        └── files  
            └── music.mp3
```



Would you kindly upload my
malicious script?

Sure!



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```



Would you kindly upload my
malicious script?

Sure!



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── shell.php  
        └── files  
            └── music.mp3
```



Would you kindly run
shell.php?

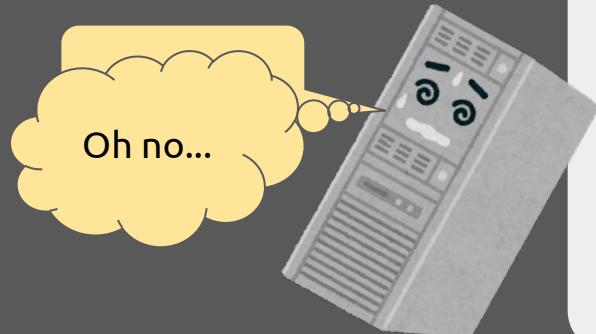


Sure!

```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── shell.php  
        └── files  
            └── music.mp3
```



Would you kindly run
shell.php?

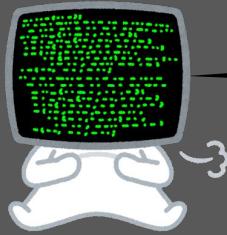


```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── shell.php  
        └── files  
            └── music.mp3
```



“just lie”





`open("/etc/shadow")`

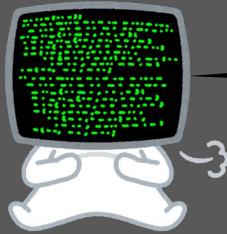
Let's look up
`/etc/shadow`.



`open`

`/etc/shadow`

```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```



`open("/etc/shadow")`

`open`

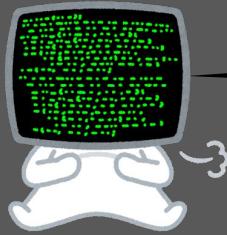
Let's look up
`/etc/shadow`.

Okay, the root is
here...



`/etc/shadow`

```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```



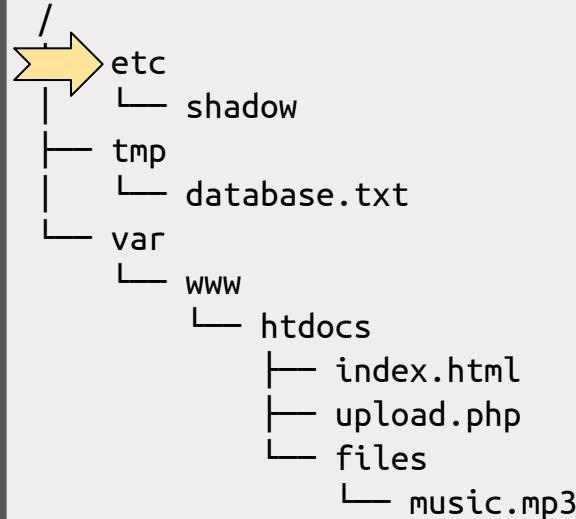
`open("/etc/shadow")`

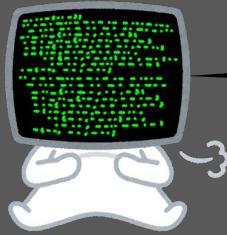
`open`

Let's look up
`/etc/shadow`.



↓
`/etc/shadow`





`open("/etc/shadow")`

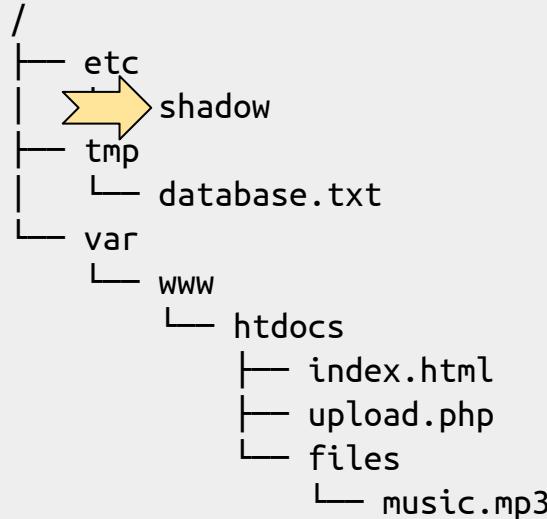
Let's look up
`/etc/shadow`.

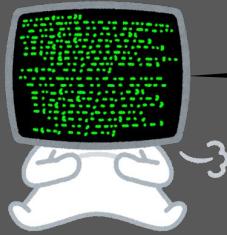


`open`



`/etc/shadow`





`open("/etc/shadow")`

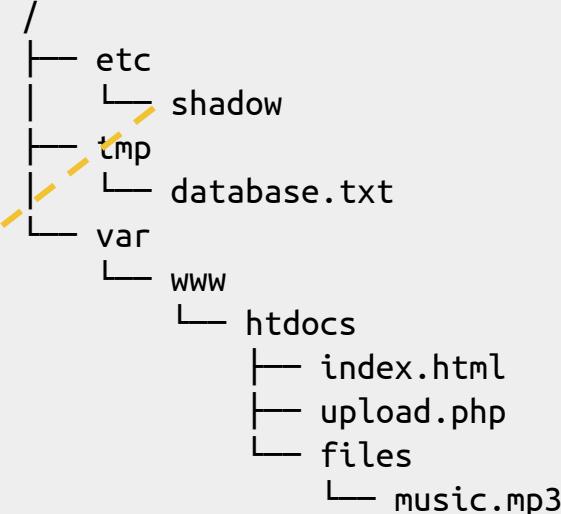
`open`



Let's look up
`/etc/shadow`.

Let's allocate a file
handle...

`fd 9`





`open("/etc/shadow")`

`open`

`fd 9`



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```

Okay, the root is
here...



chroot

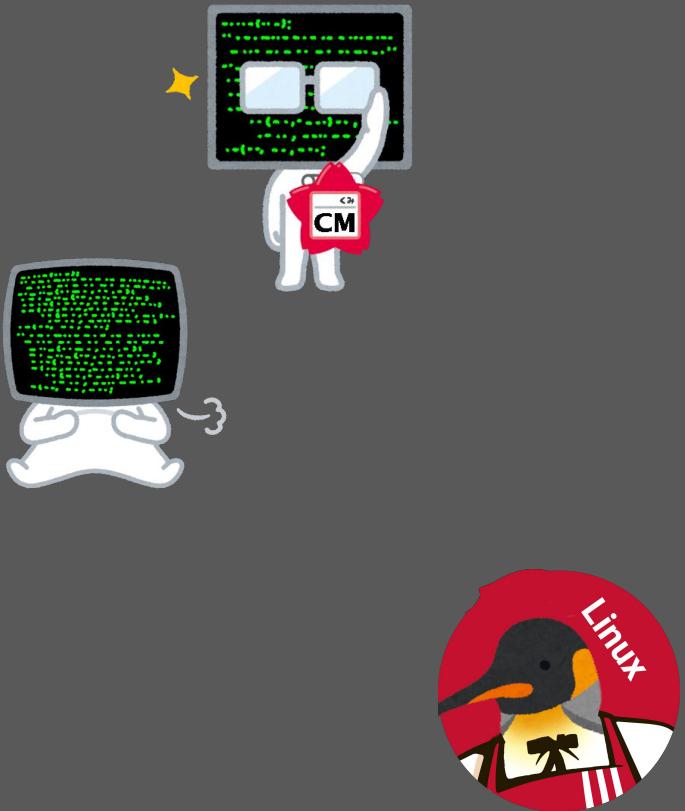
Okay, the root is
here...





Container
Manager





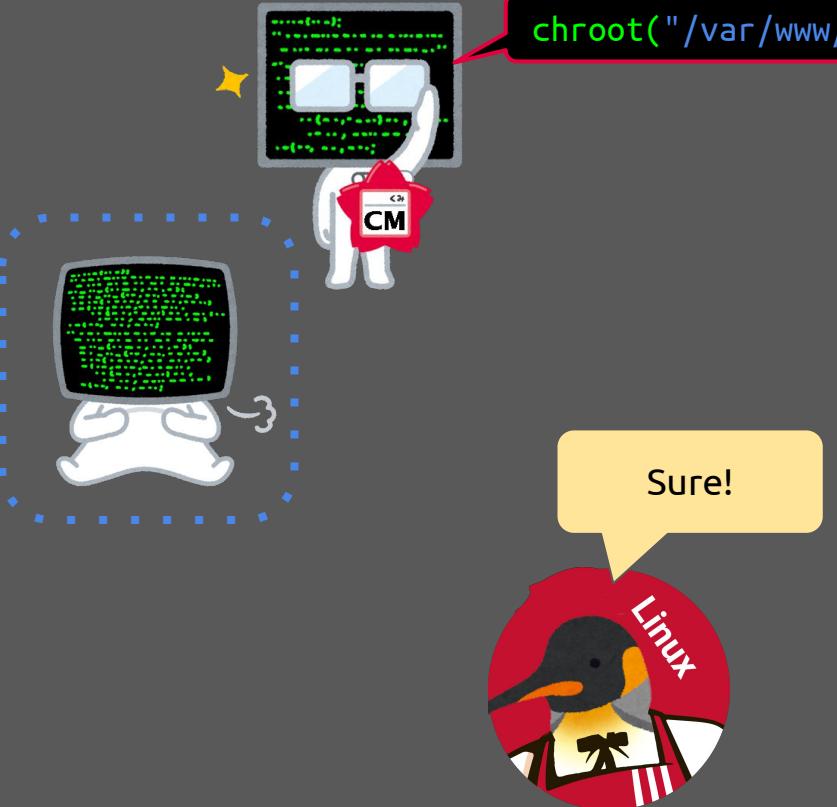
```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```



`chroot("/var/www/htdocs")`



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```



`chroot("/var/www/htdocs")`

...and run this process.



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```



`chroot("/var/www/htdocs")`

...and run this process.



Sure!



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── files  
                └── music.mp3
```



`open("/etc/shadow")`

`open`

Let's look up
`/etc/shadow`.



`/etc/shadow`

```
/  
|   └── etc  
|       └── shadow  
|   └── home  
|       └── cyphar  
|           └── secrets.txt  
└── var  
    └── db  
        └── userdata  
    └── www  
        └── htdocs  
            └── index.html  
                └── files  
                    └── music.mp3
```



`open("/etc/shadow")`

`open`

Let's look up
`/etc/shadow`.

Okay, the root is
here...



`/etc/shadow`

```
/  
└── etc  
    └── shadow  
└── home  
    └── cyphar  
        └── secrets.txt  
└── var  
    └── db  
        └── userdata  
└── www  
    └── htdocs  
        ├── index.html  
        └── files  
            └── music.mp3
```



`open("/etc/shadow")`

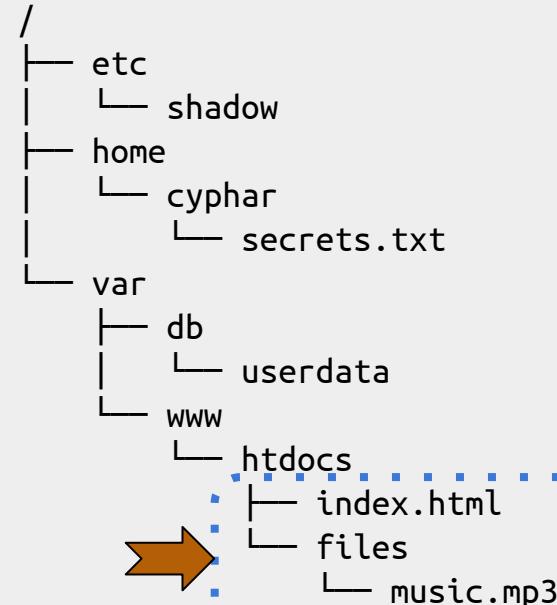
`open`

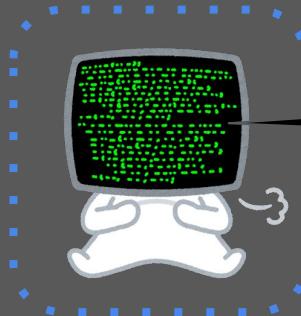
Let's look up
`/etc/shadow`.

Huh, there's no
etc here.



`/etc/shadow`





```
open("/etc/shadow")
```

```
open
```

```
err
```



```
/  
  └ etc  
      └ shadow  
  └ home  
      └ cyphar  
          └ secrets.txt  
  └ var  
      └ db  
          └ userdata  
  └ www  
      └ htdocs  
          └ index.html  
          └ files  
            └ music.mp3
```

Container
Manager





Wow! I'm Mr. Manager.



Manager

A new foe has app[★]

CHALLENGER APPROACHING





Would you kindly
connect to
10.42.123.45:22?

Sure!





Would you kindly
listen to 0.0.0.0:443?

Sure!



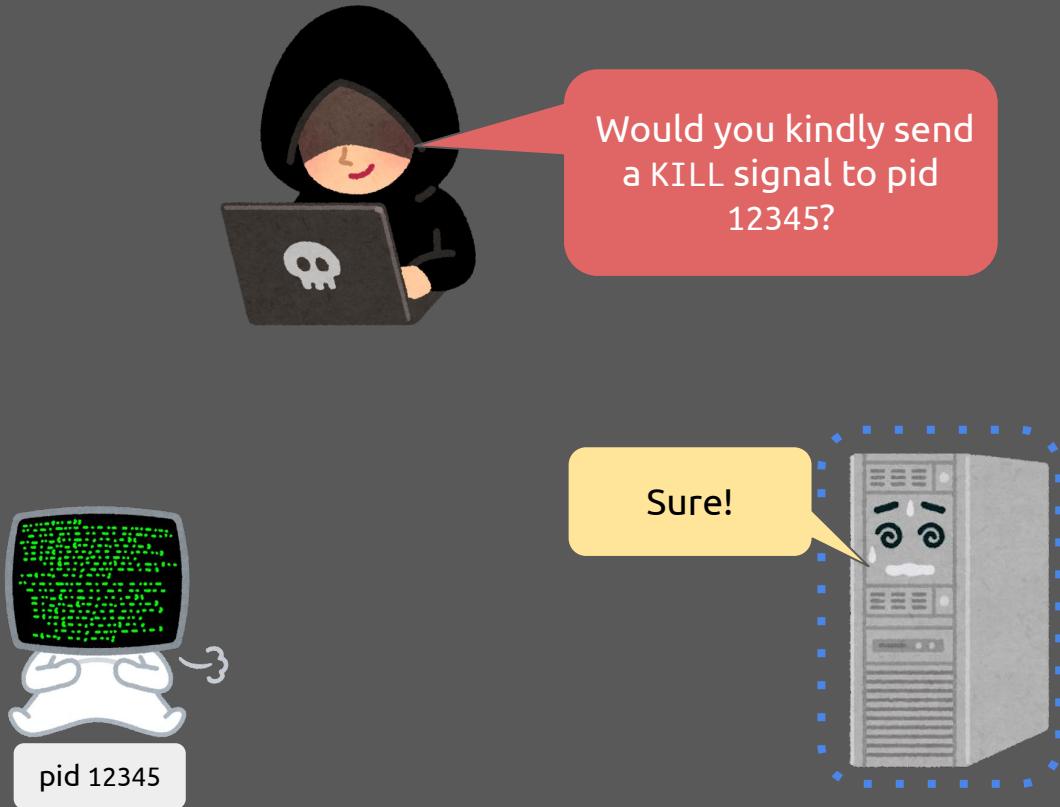


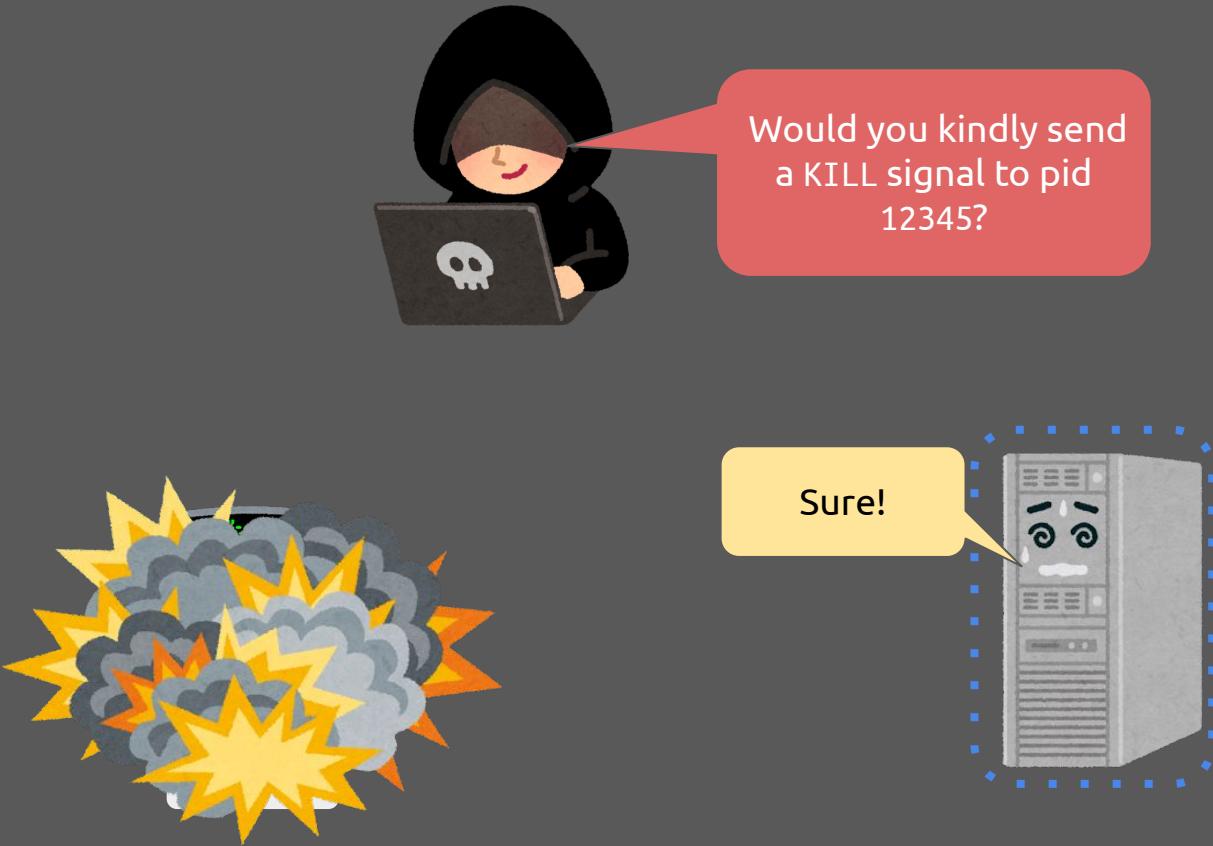
Would you kindly tell
me about pid 12345?



Sure!



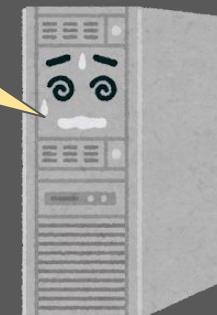






Would you kindly send
a KILL signal to pid
12345?

Wait, where did the
database go?!





while(1) { fork(); }
kthxbai





while(1) { fork(); }
kthxbai





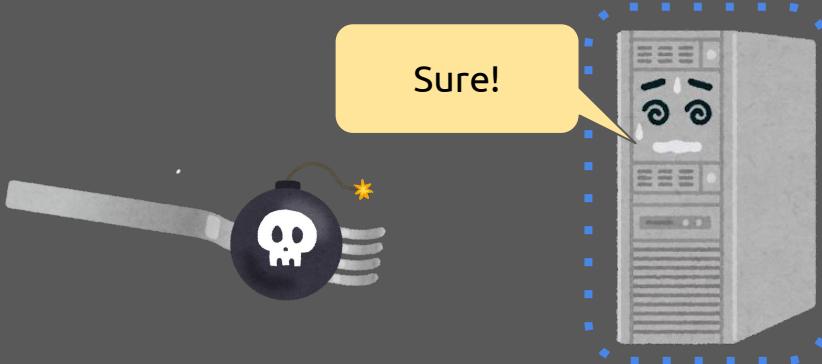
while(1) { fork(); }
kthxbai





while(1) { fork(); }
kthxbai

Sure!





while(1) { fork(); }
kthxbai

Sure!



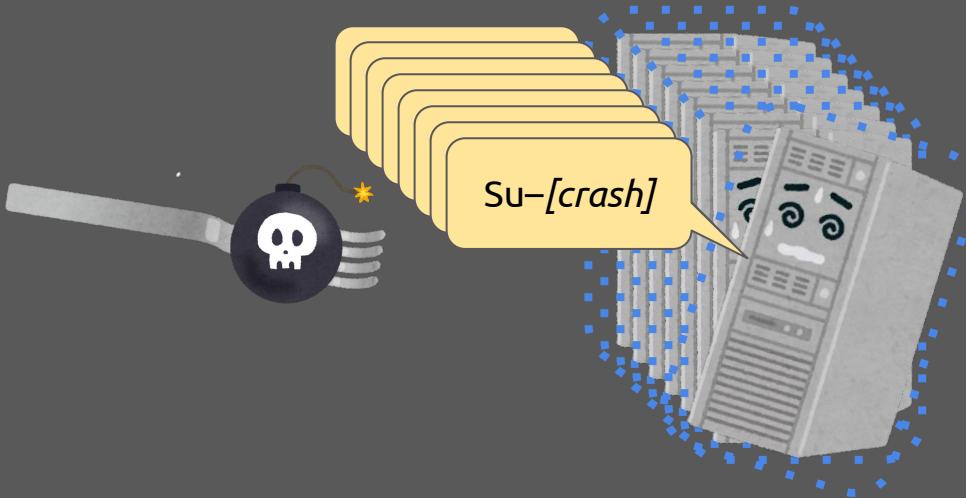


while(1) { fork(); }
kthxbai





while(1) { fork(); }
kthxbai







“just lie even more”





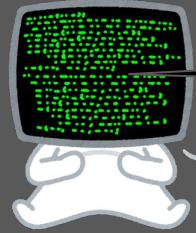
Connect to 10.42.10.123:22.

connect



eth0

wlan0



Connect to 10.42.10.123:22.

connect

Which interface
should I use...?



eth0

wlan0



Connect to 10.42.10.123:22.

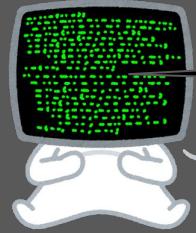
connect

Which interface
should I use...?



eth0

wlan0



Connect to 10.42.10.123:22.

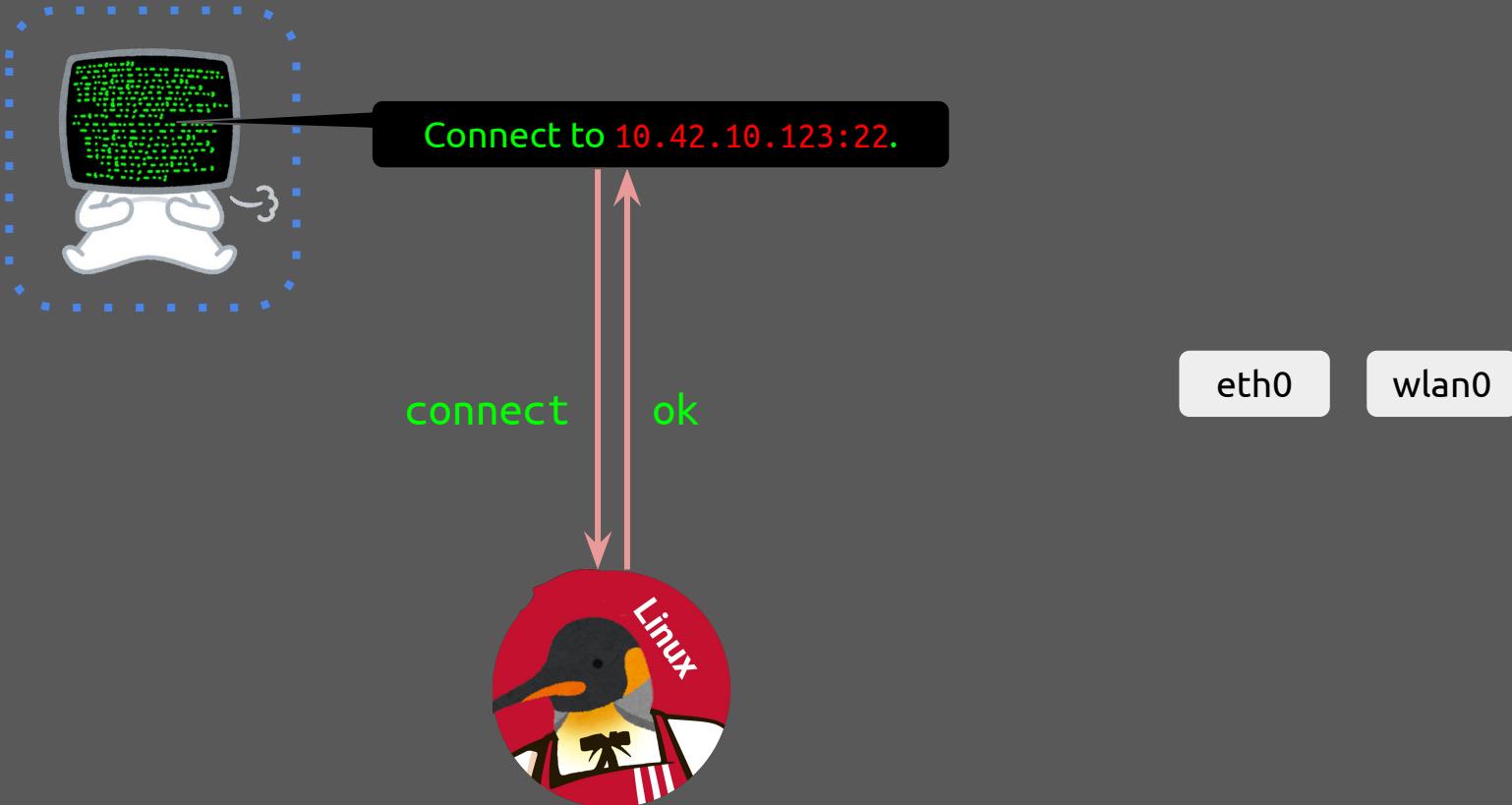
connect

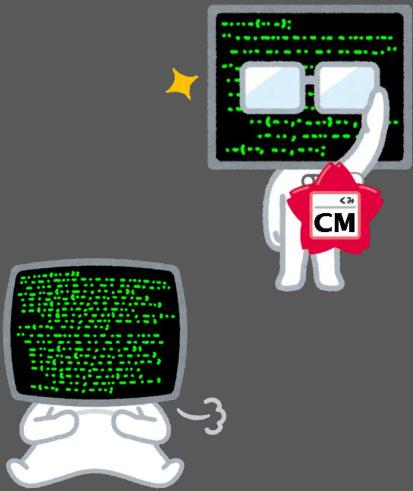
Ok, let's go with
this one.



eth0

wlan0

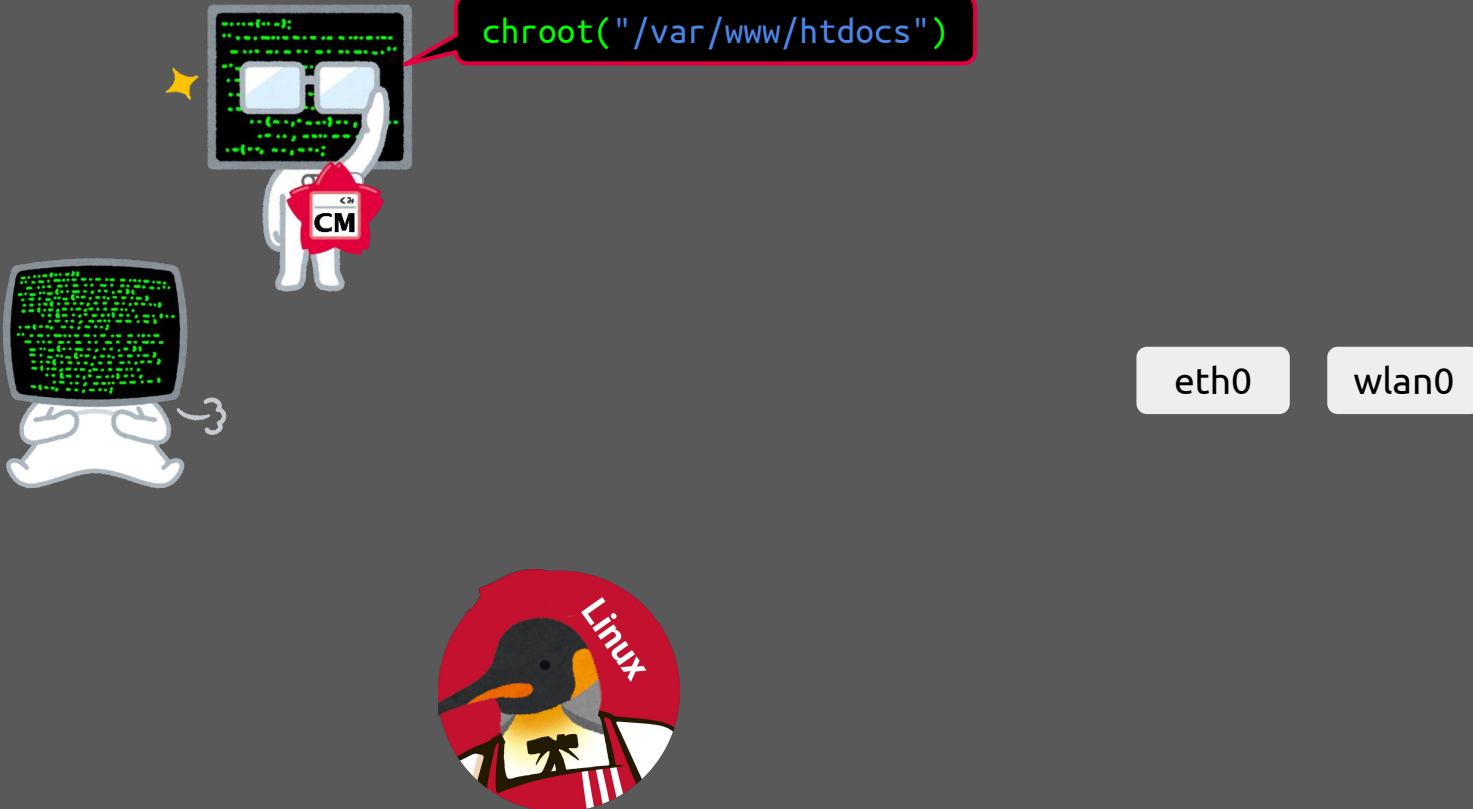


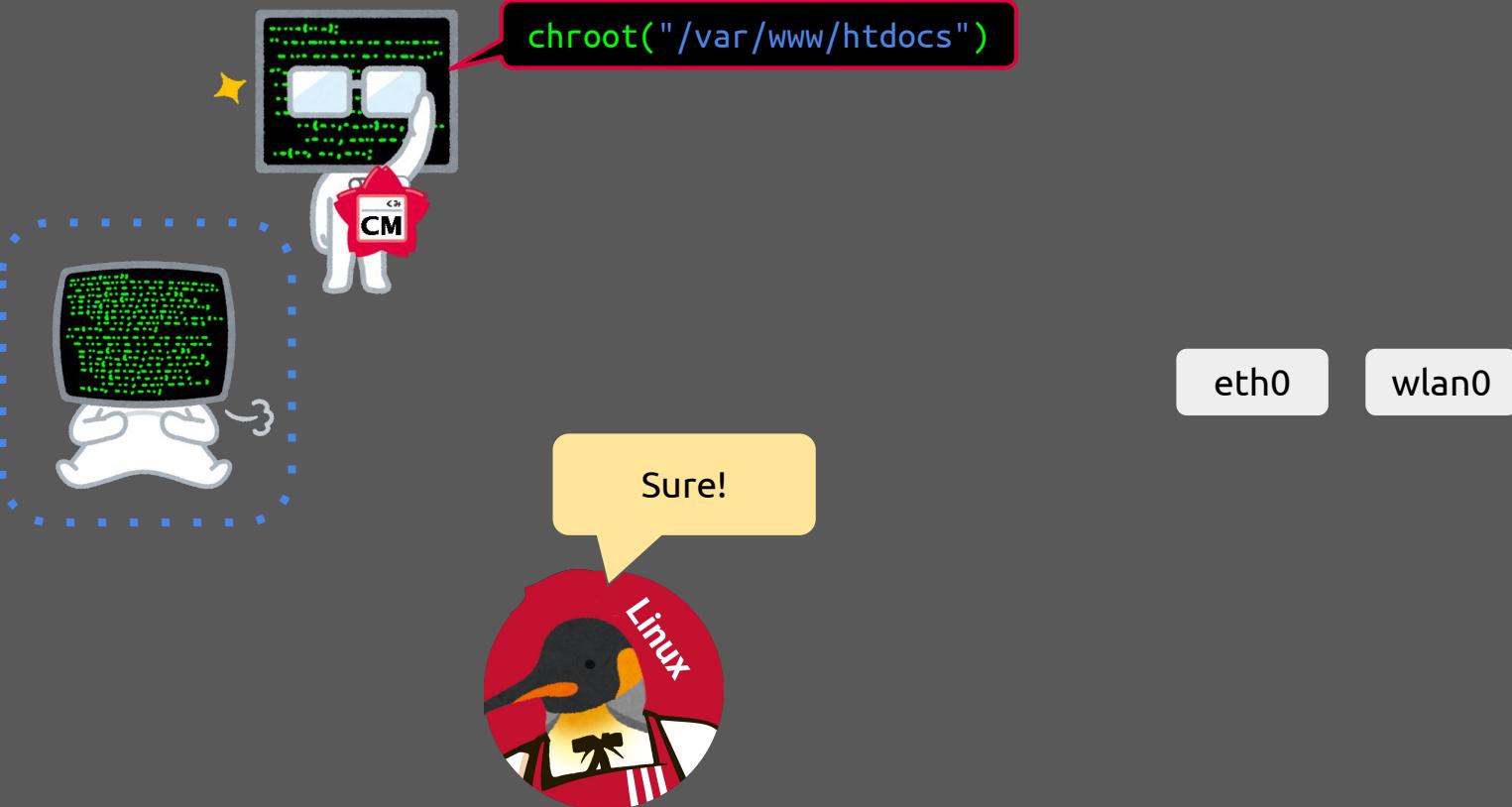


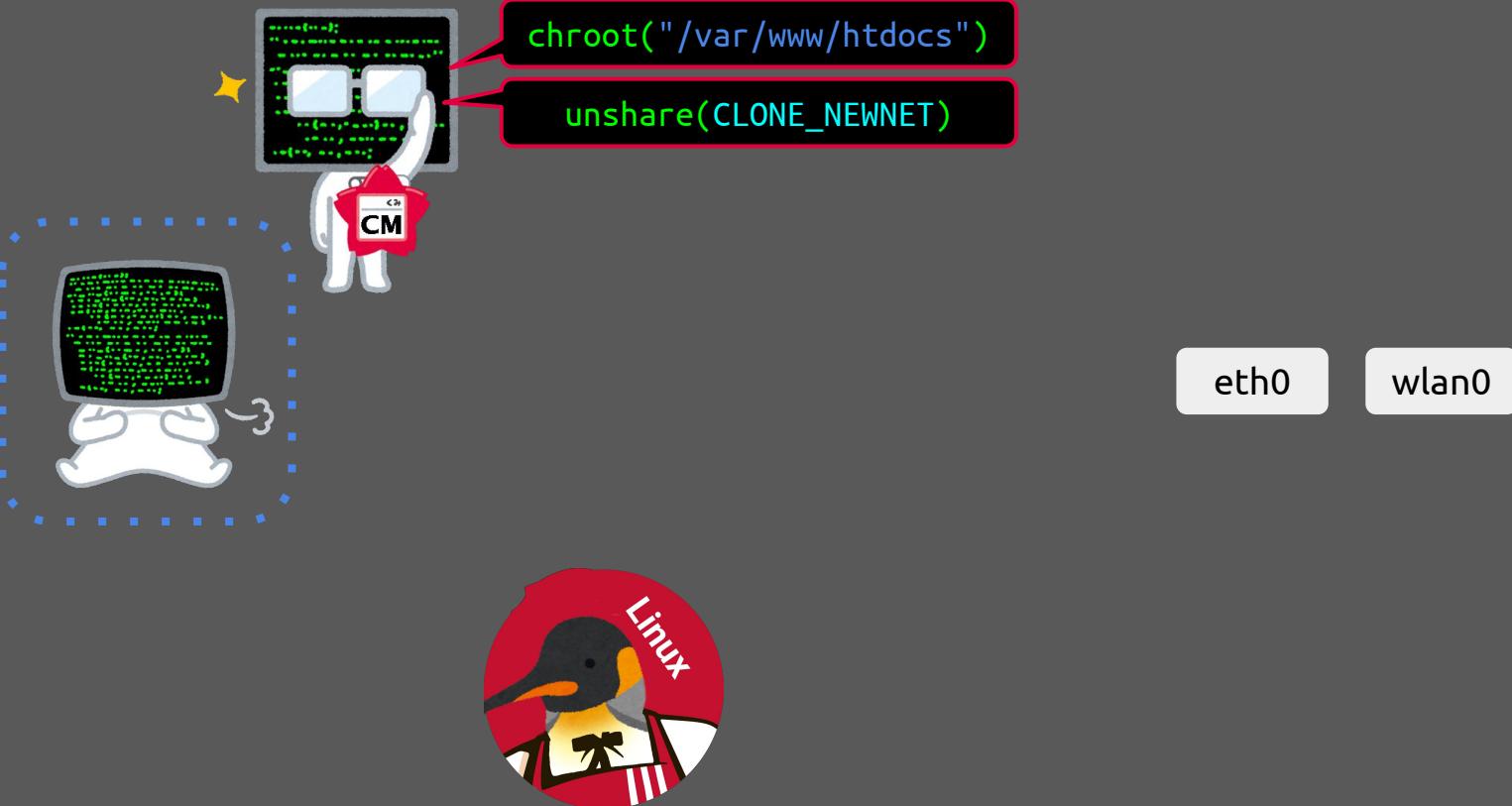
eth0

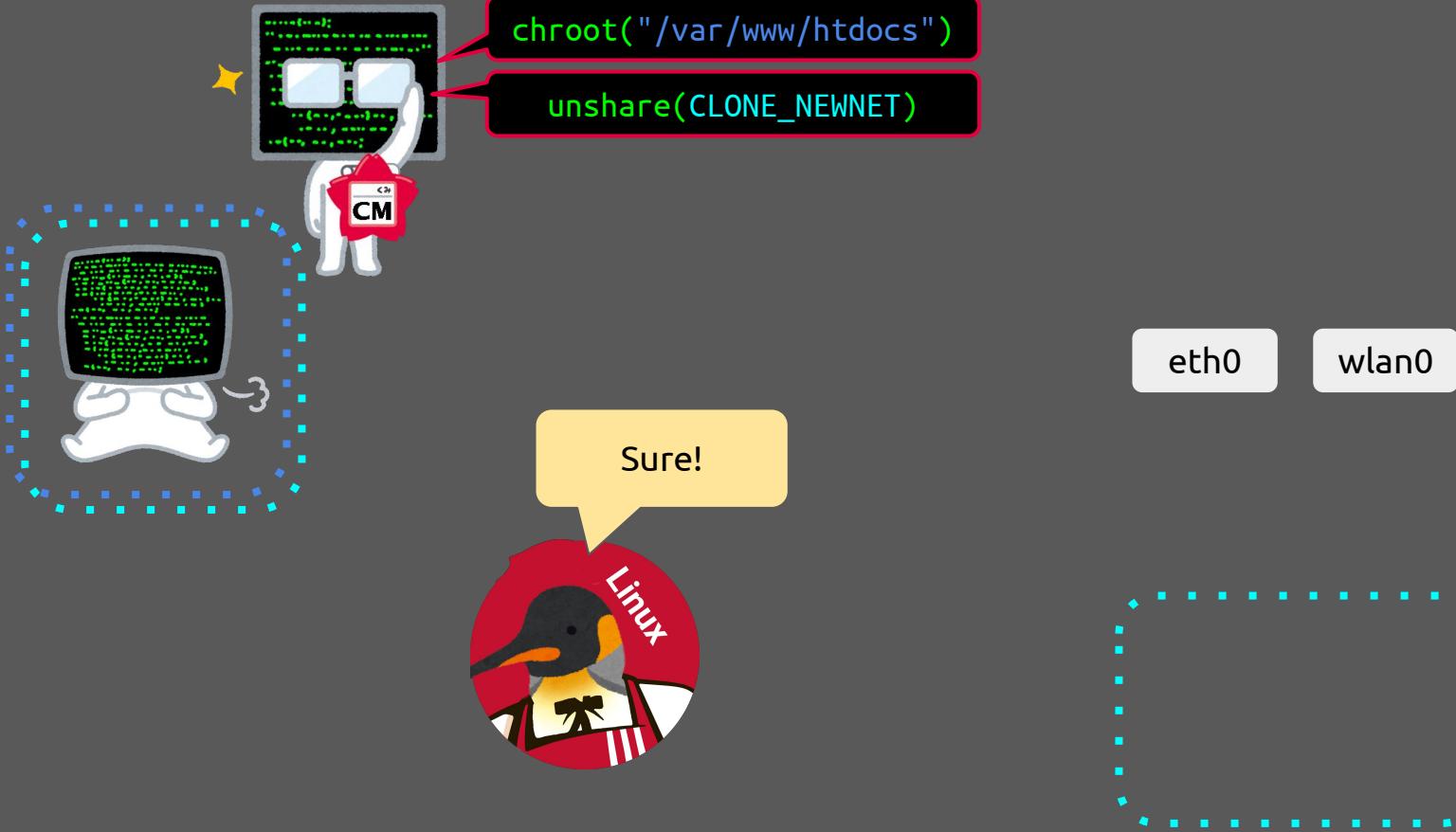
wlan0

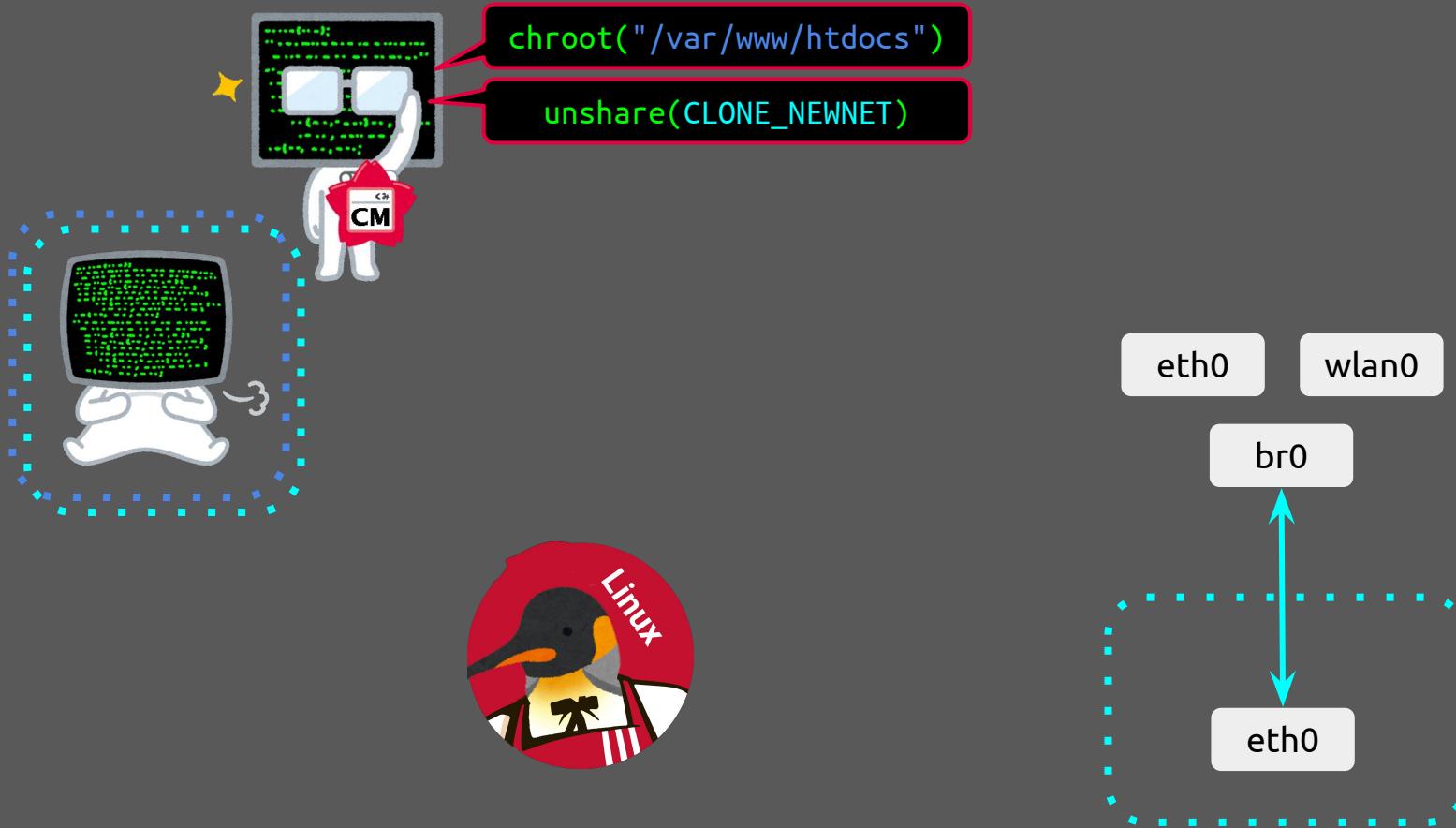


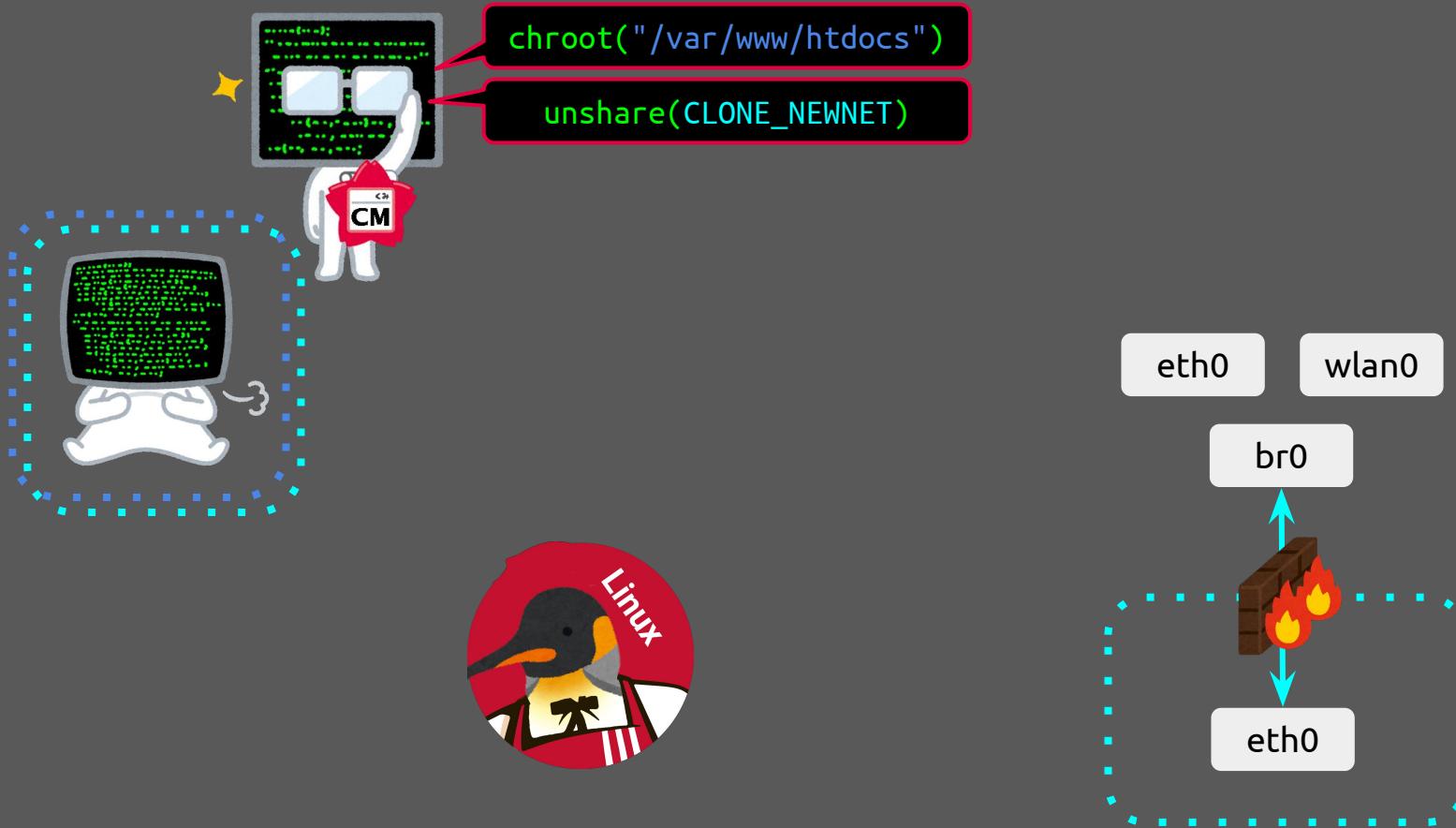


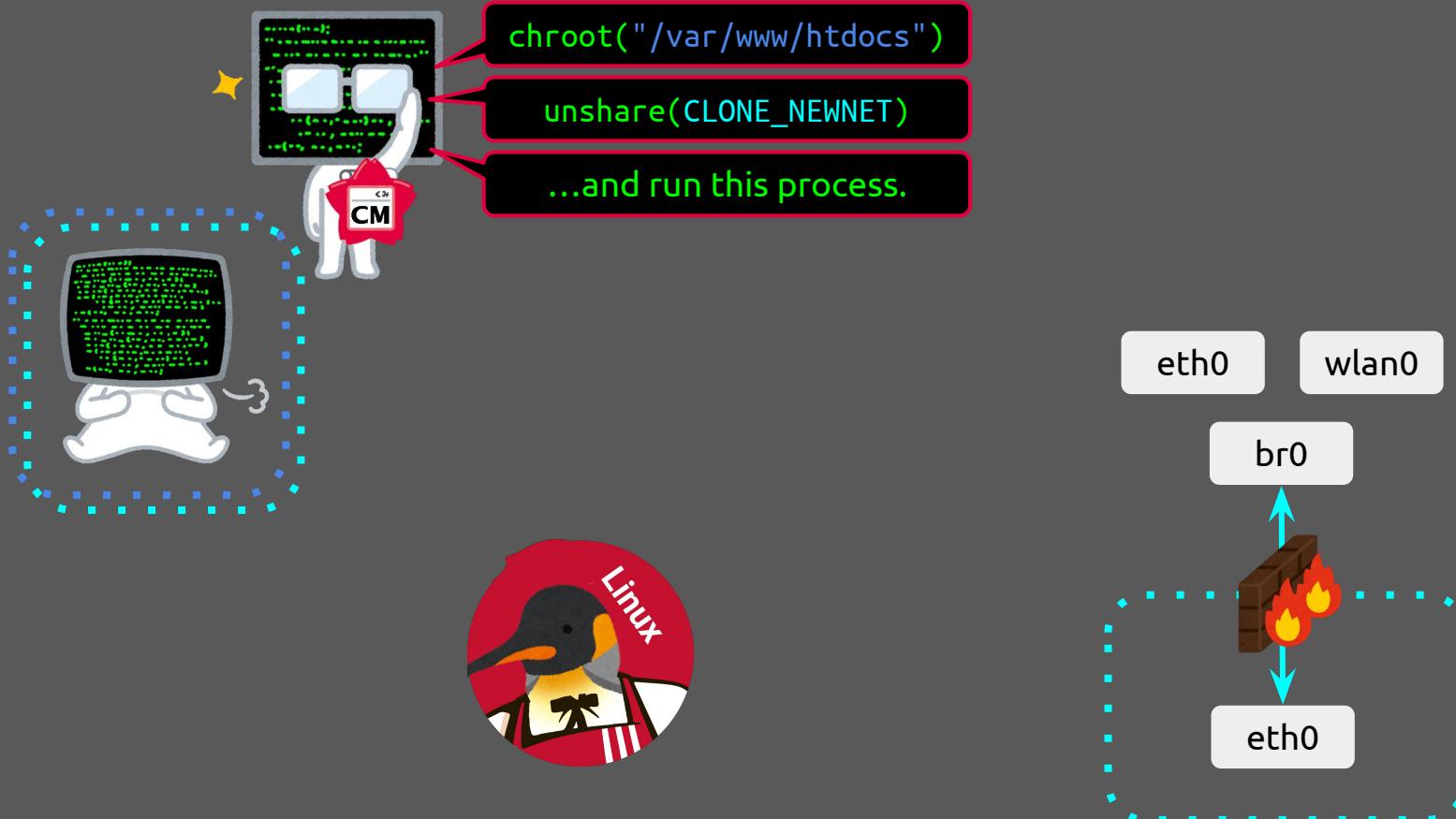


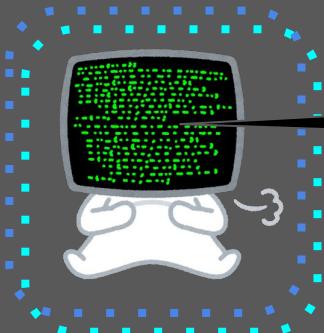






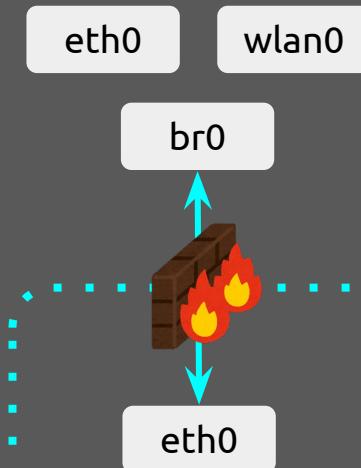


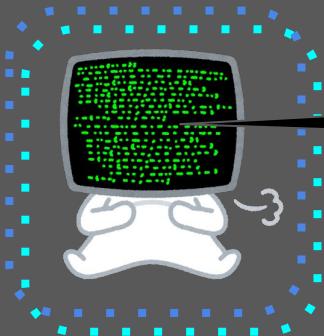




Connect to 10.42.10.123:22.

connect

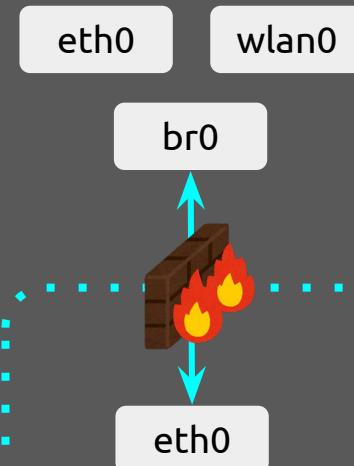


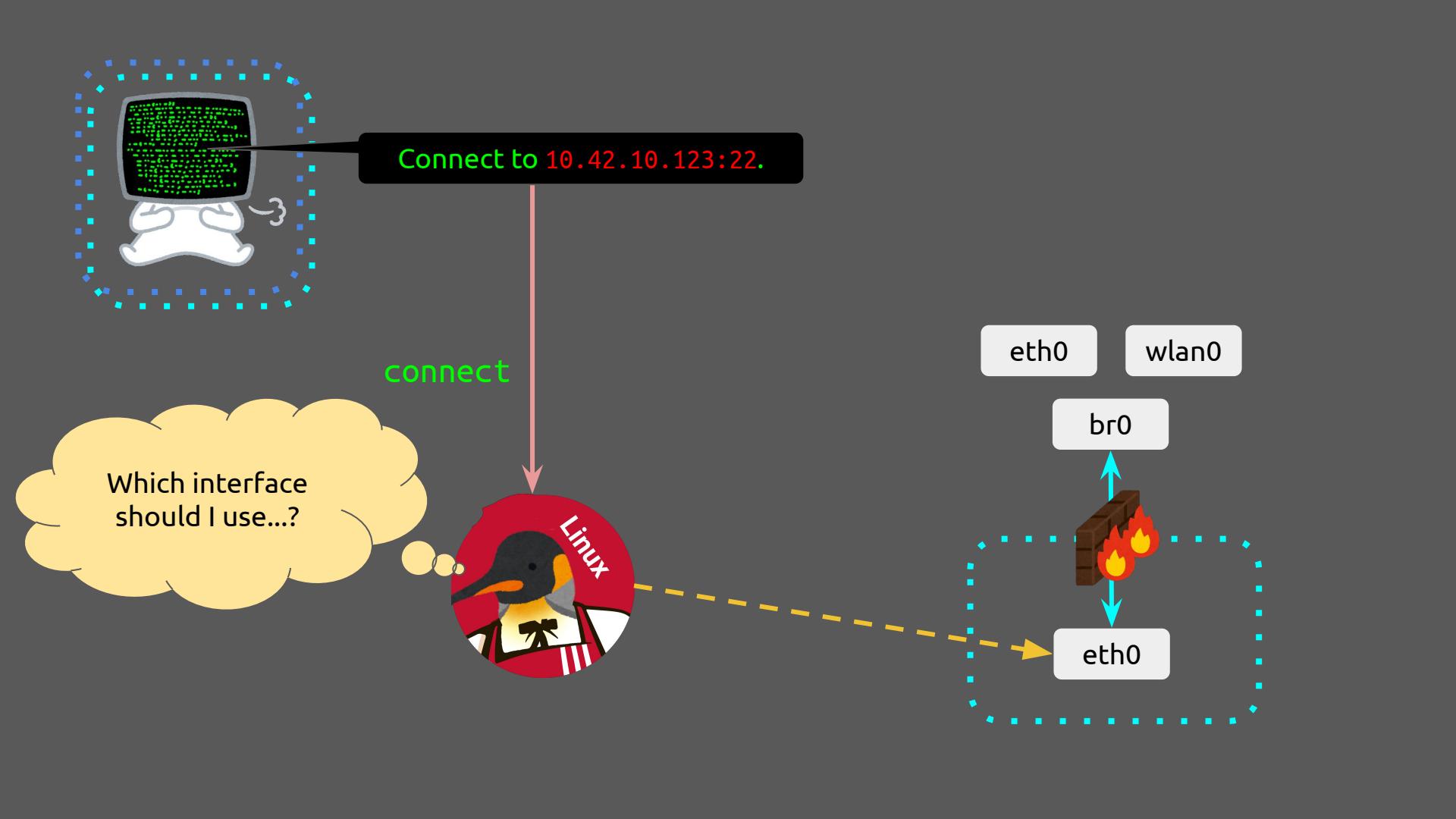


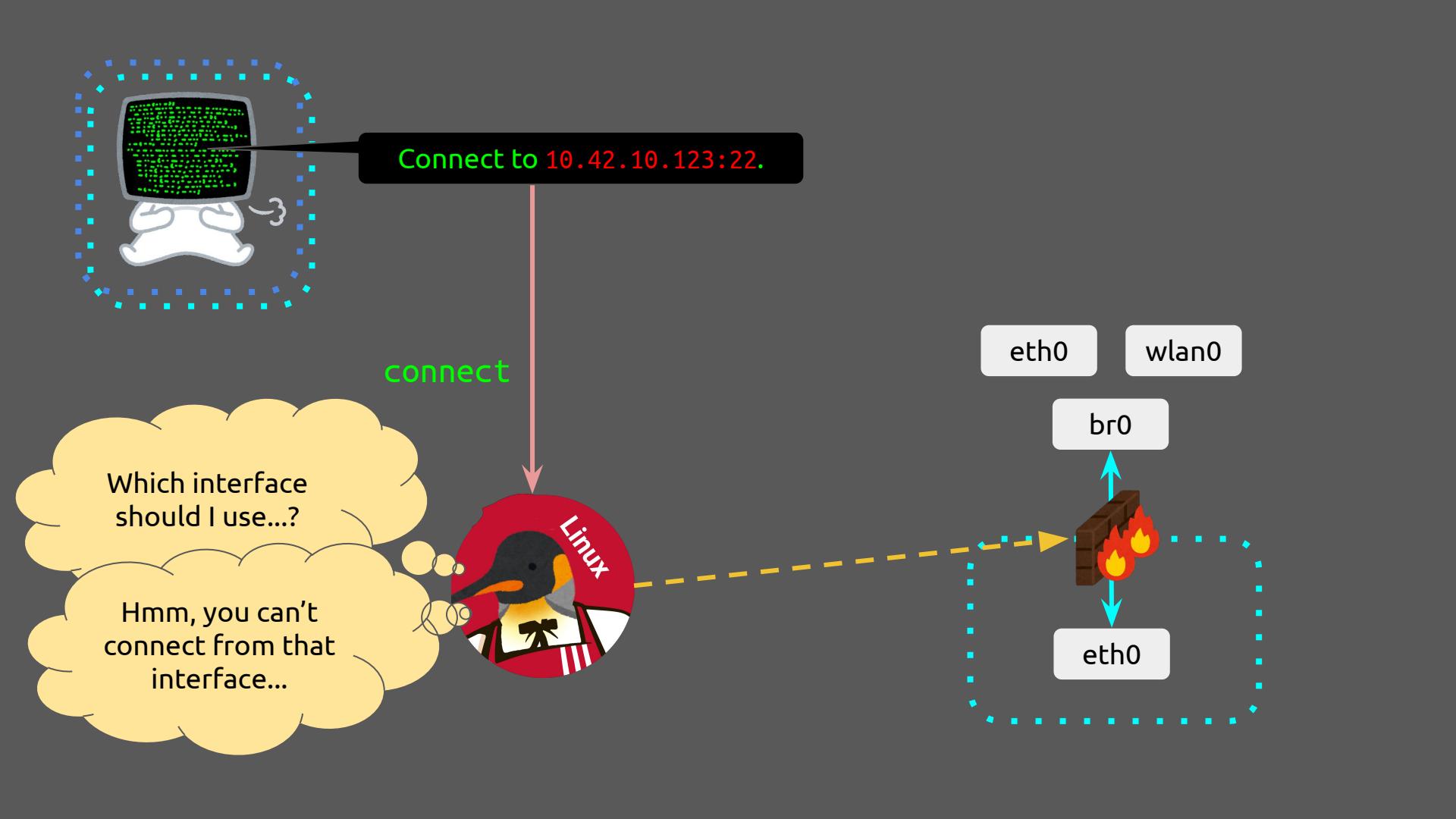
Connect to 10.42.10.123:22.

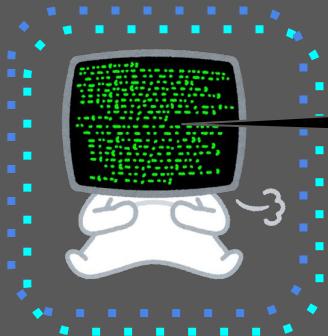
connect

Which interface
should I use...?









Connect to 10.42.10.123:22.

connect

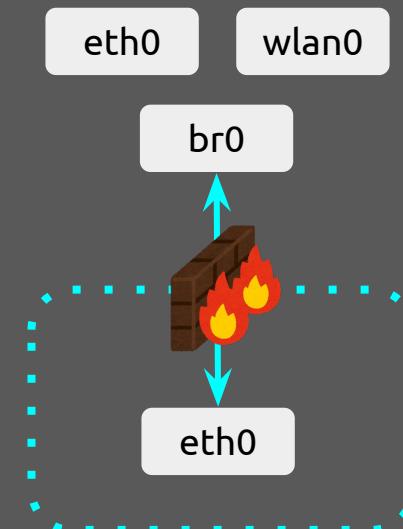
err

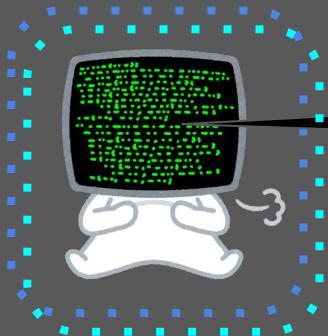


eth0

wlan0

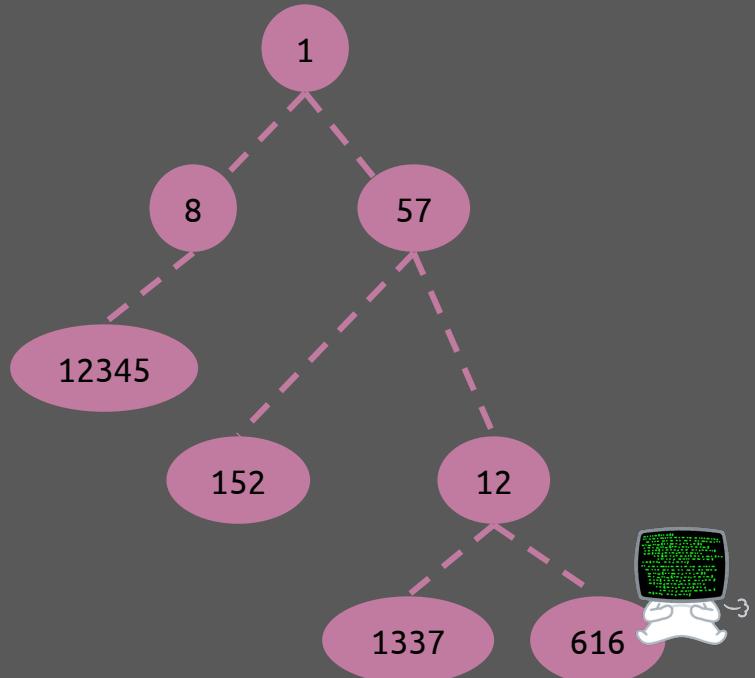
br0

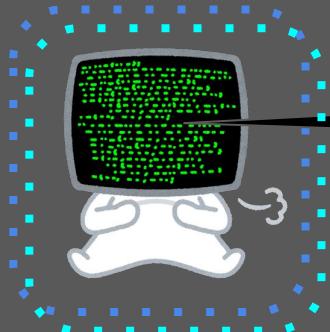




Kill PID 12345.

kill



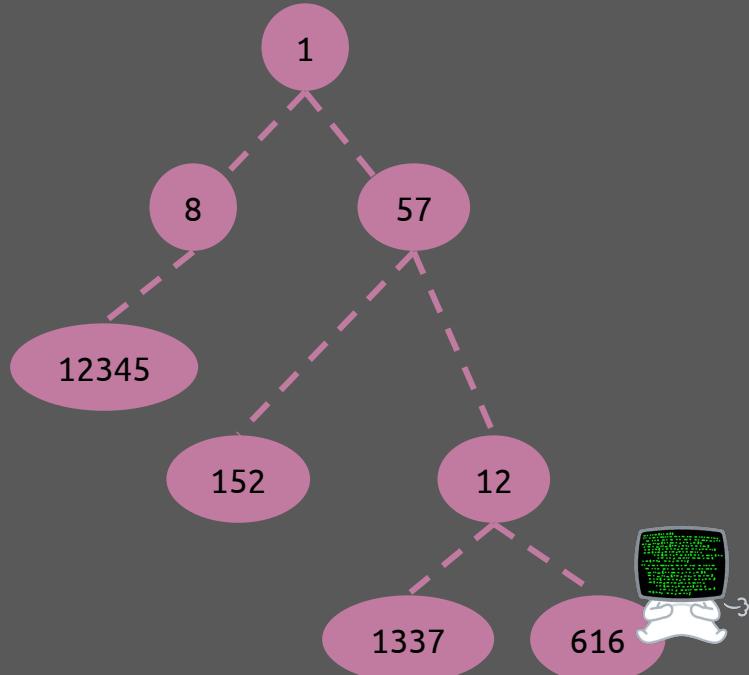


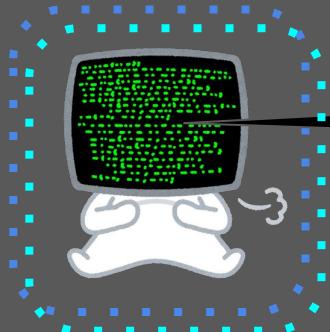
Kill PID 12345.

kill



Hmm, where is
PID 12345...

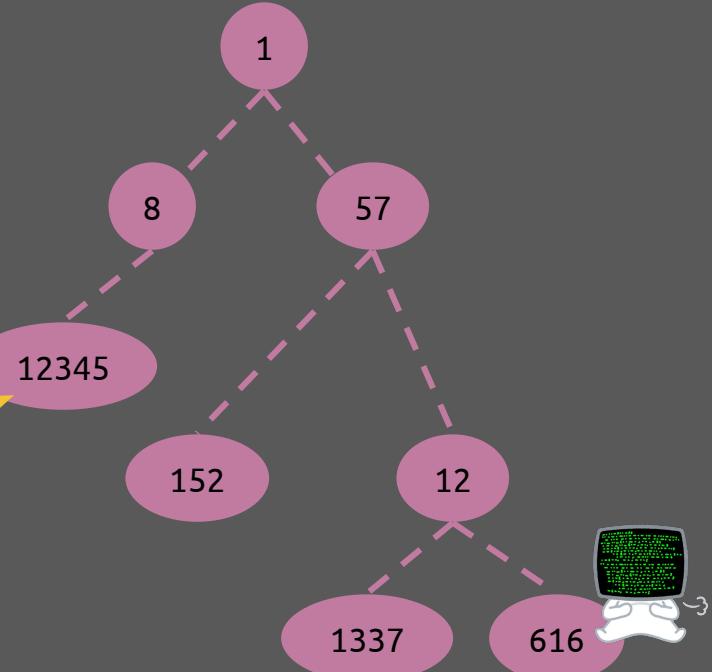


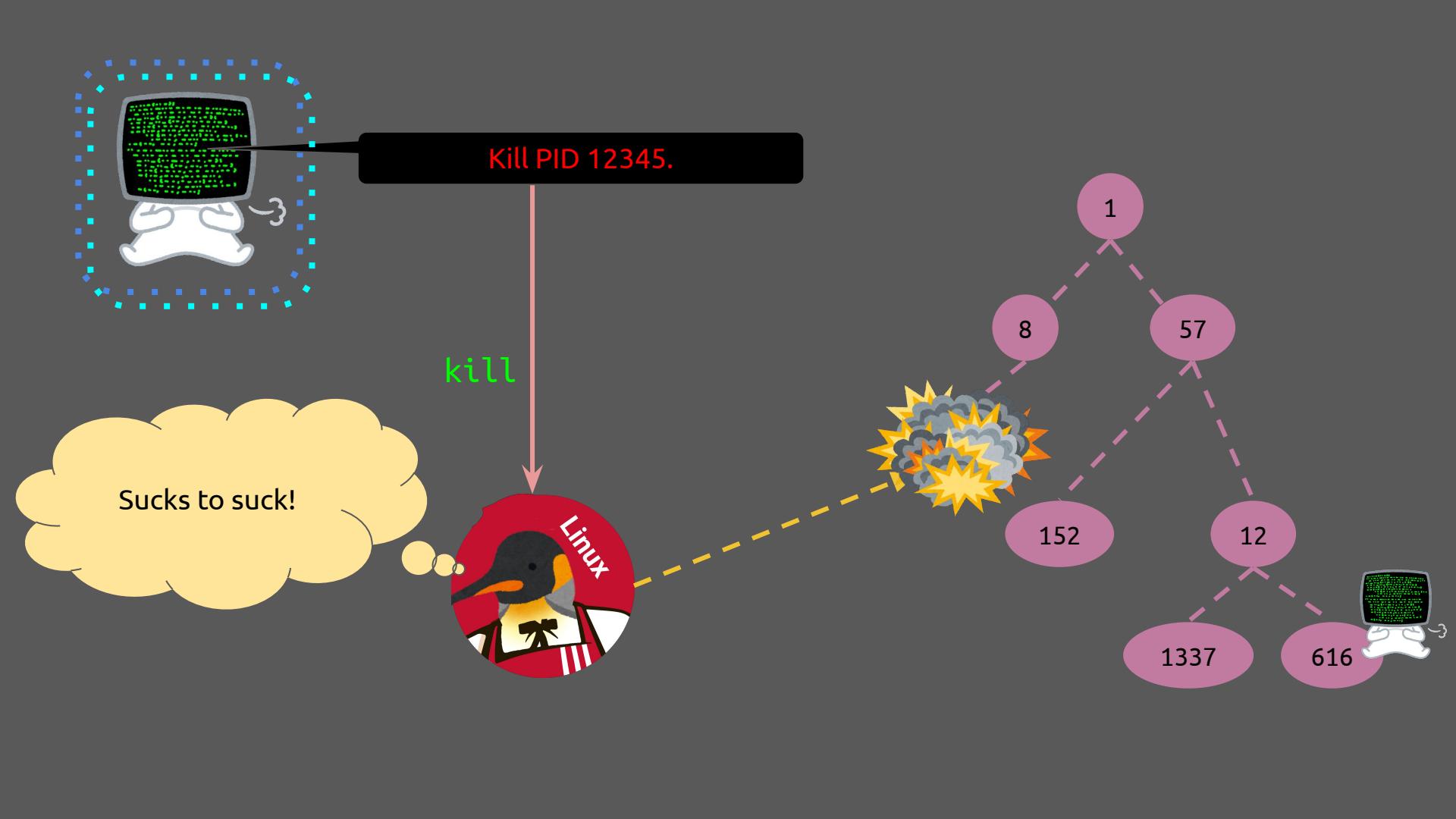


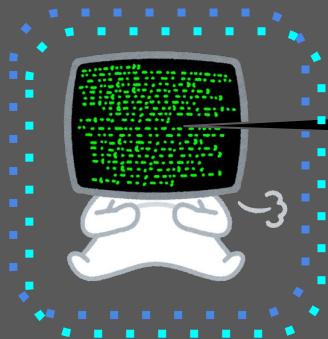
Kill PID 12345.

kill

Hmm, where is
PID 12345...



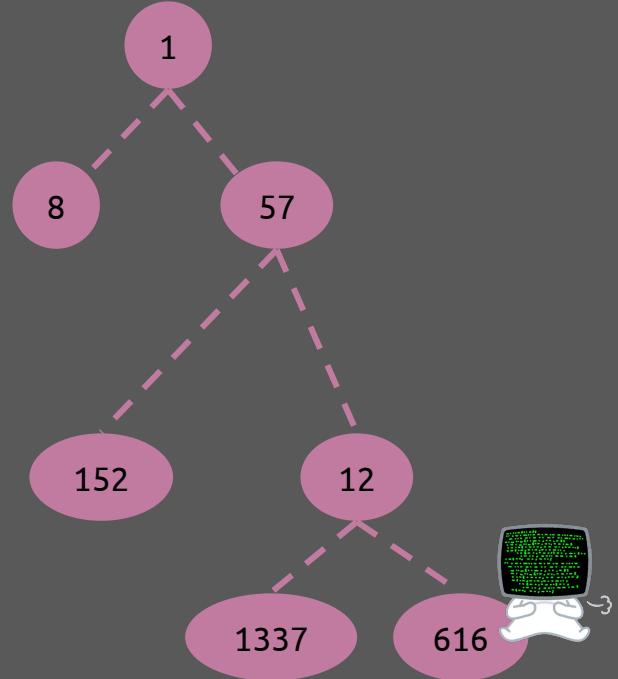


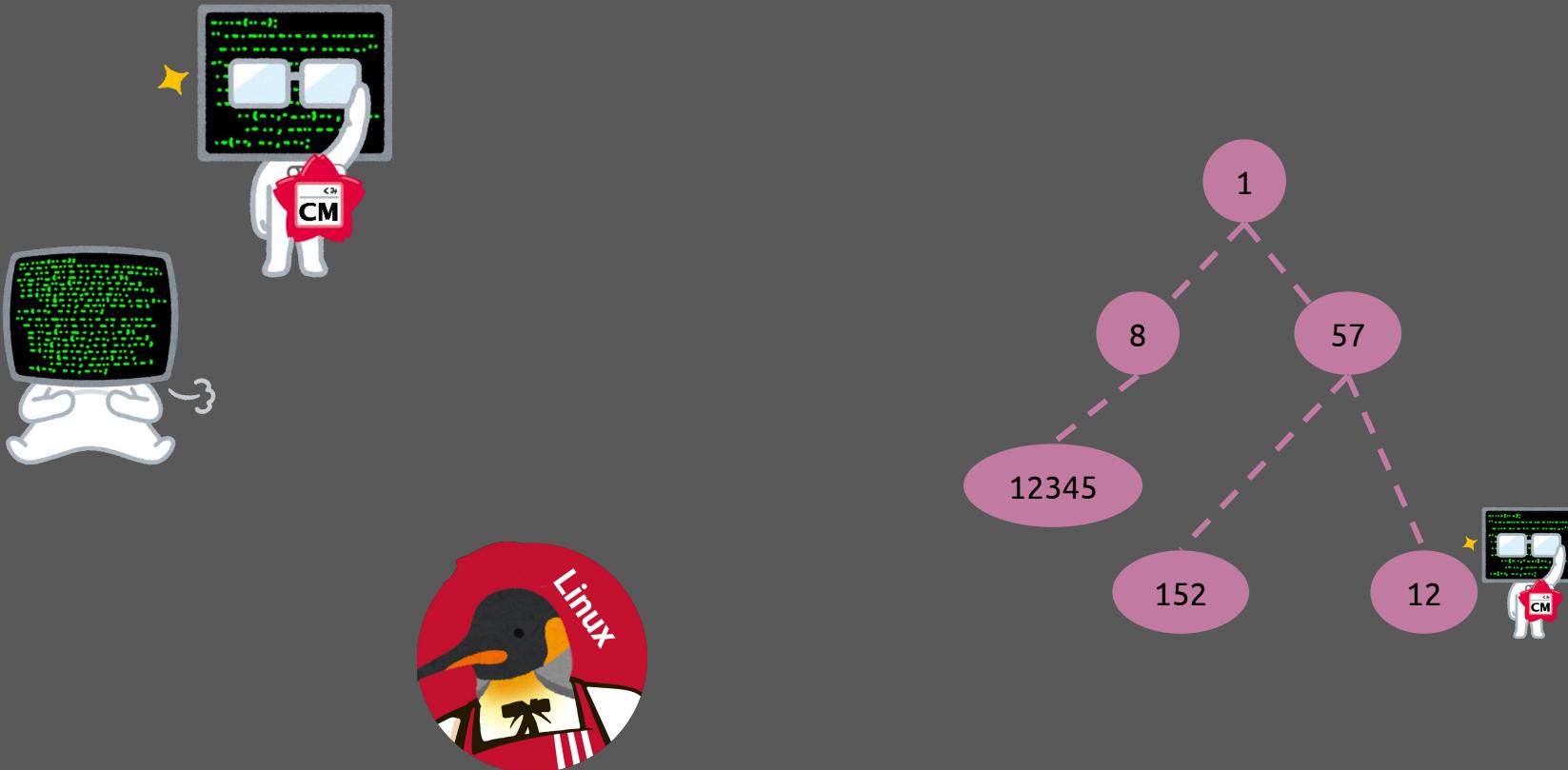


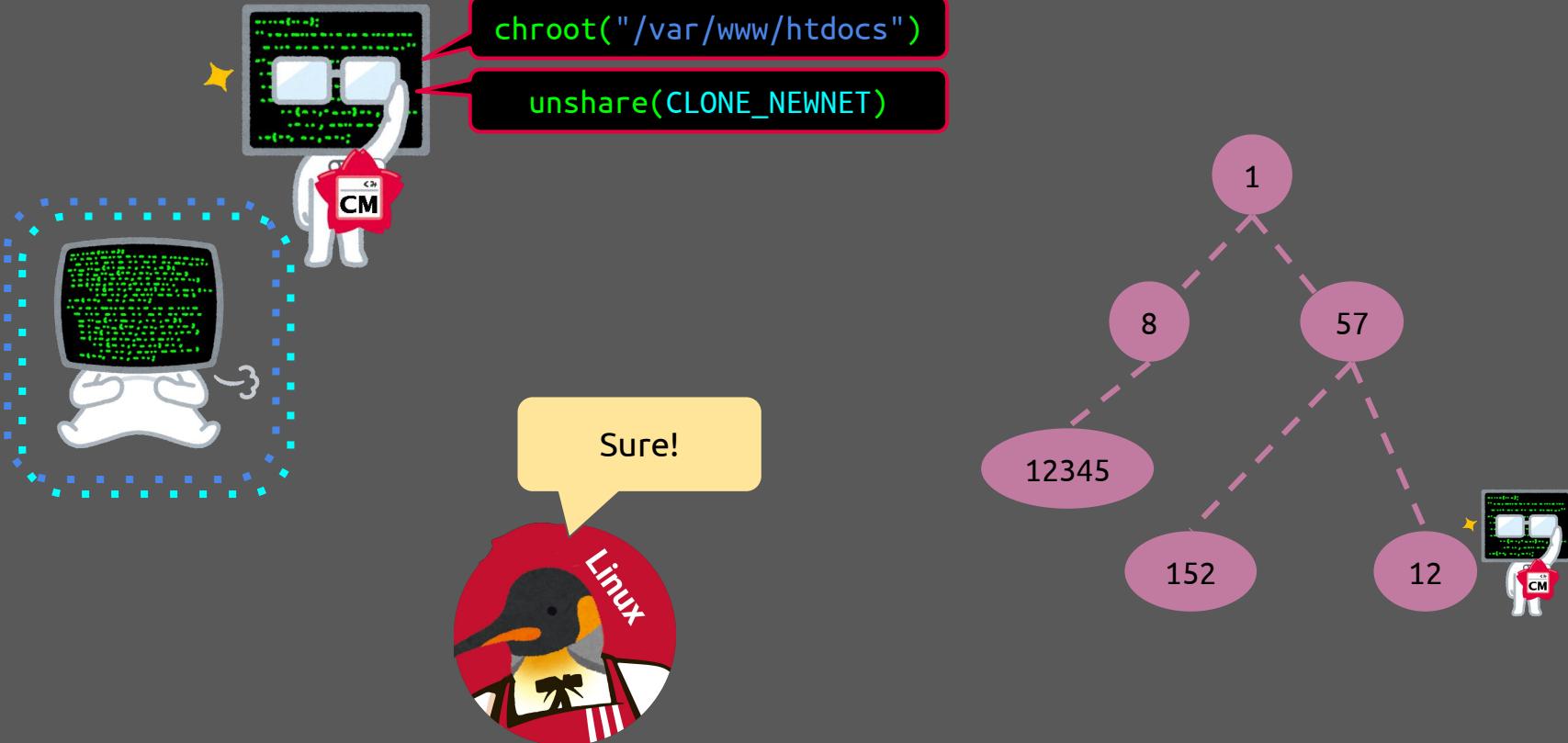
Kill PID 12345.

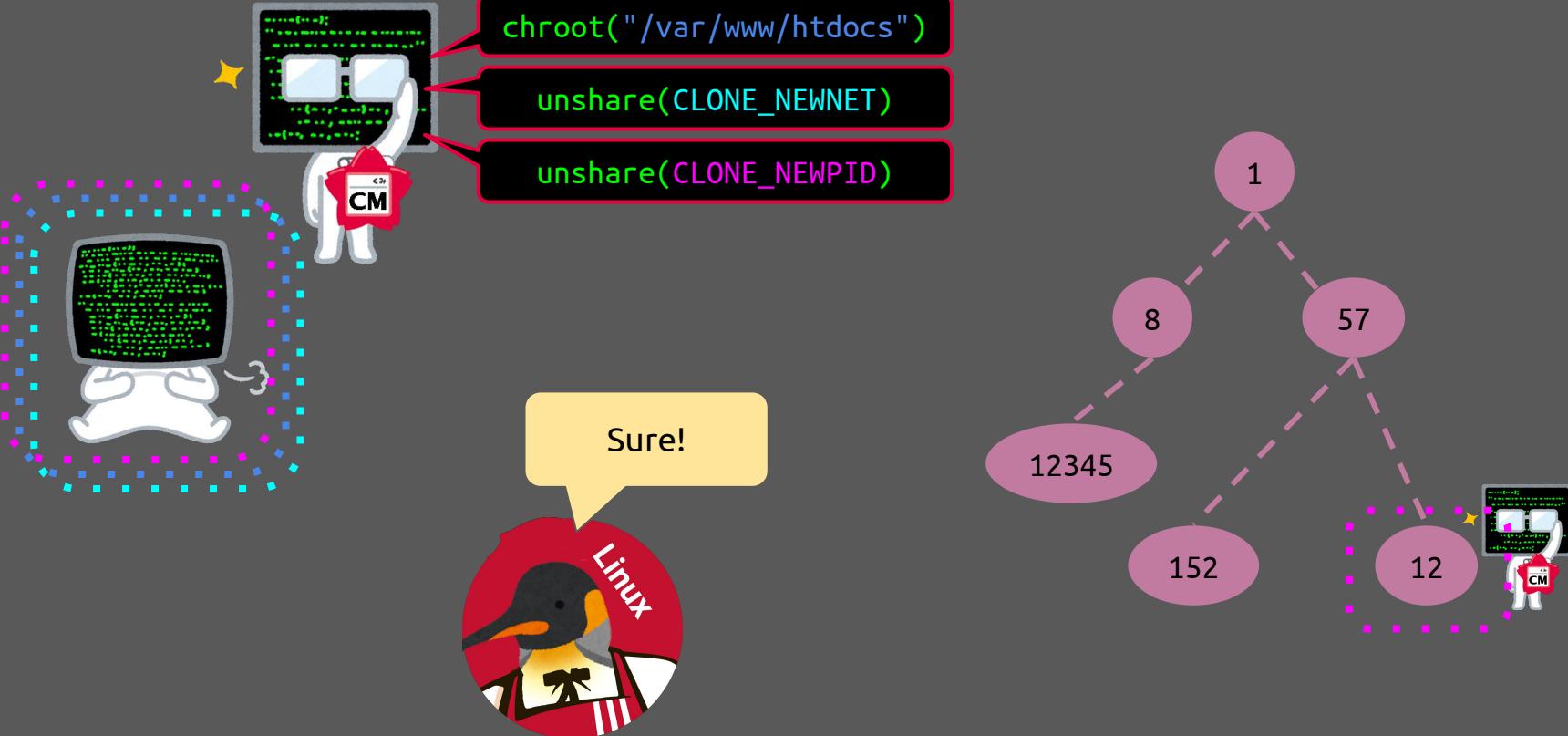
kill

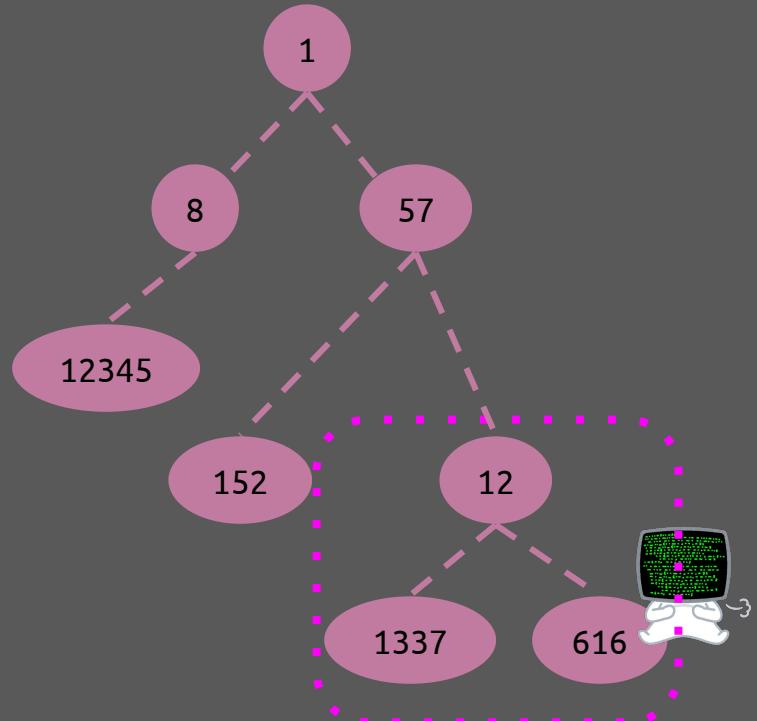
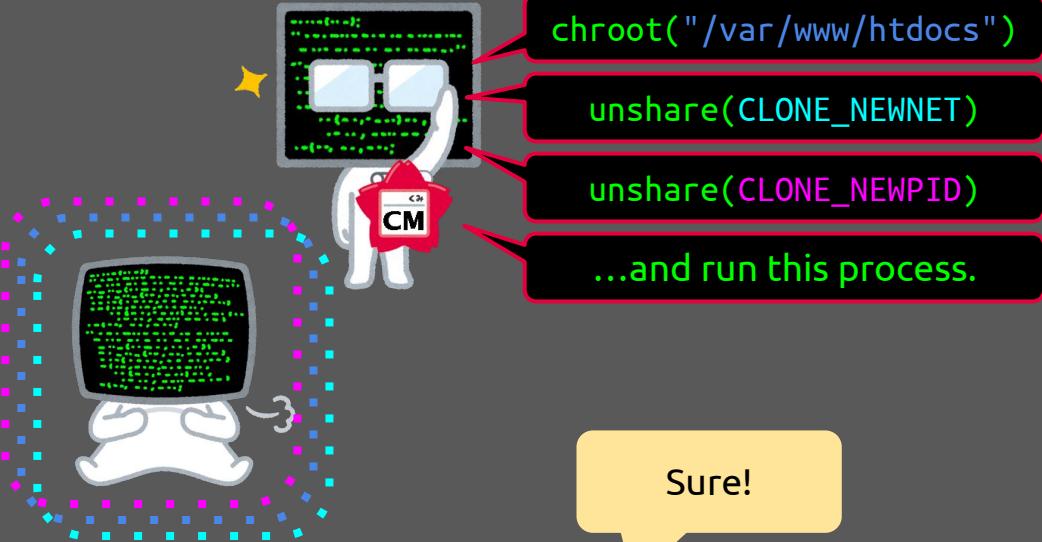
ok

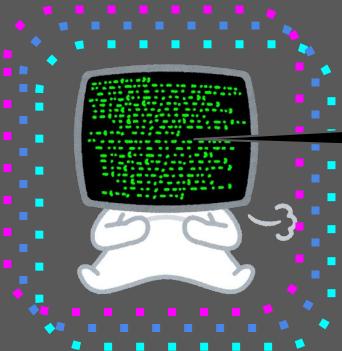






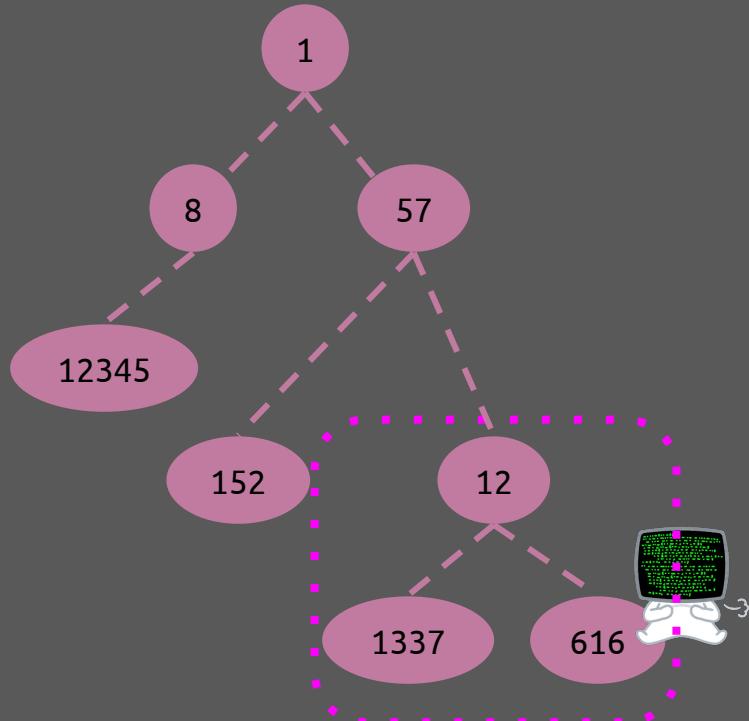


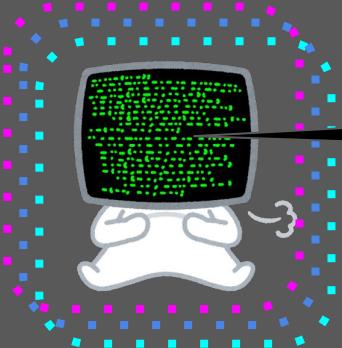




What is my PID?

getpid



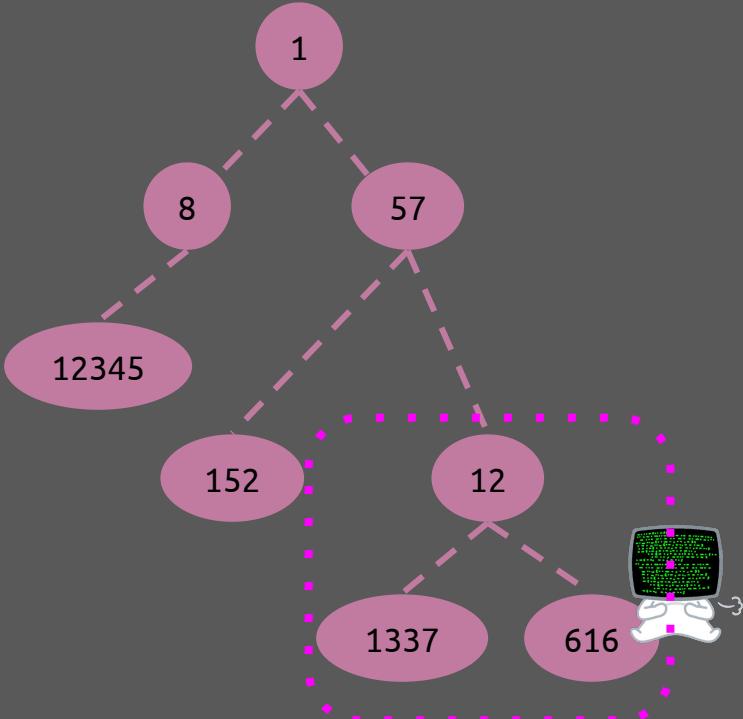


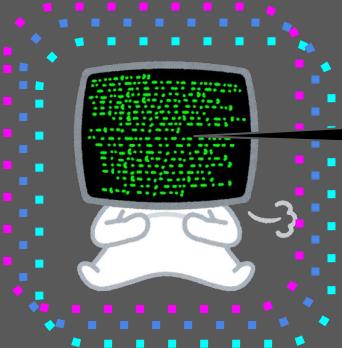
What is my PID?

getpid



What PID
namespace is this
process in?

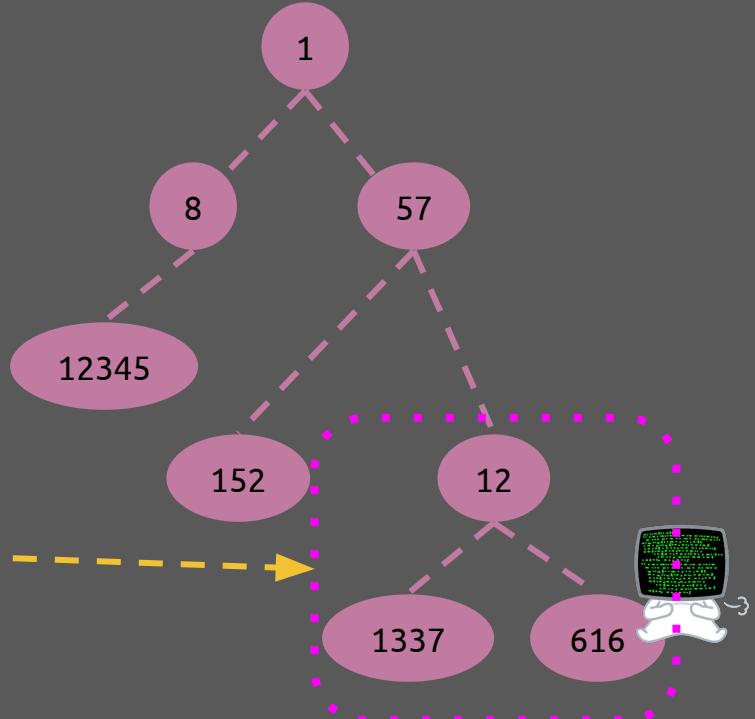


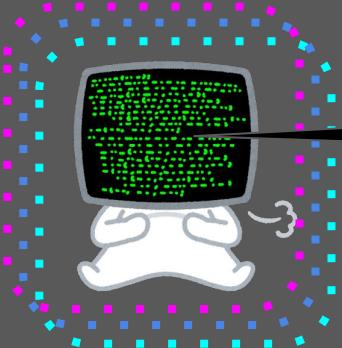


What is my PID?

getpid

What PID
namespace is this
process in?

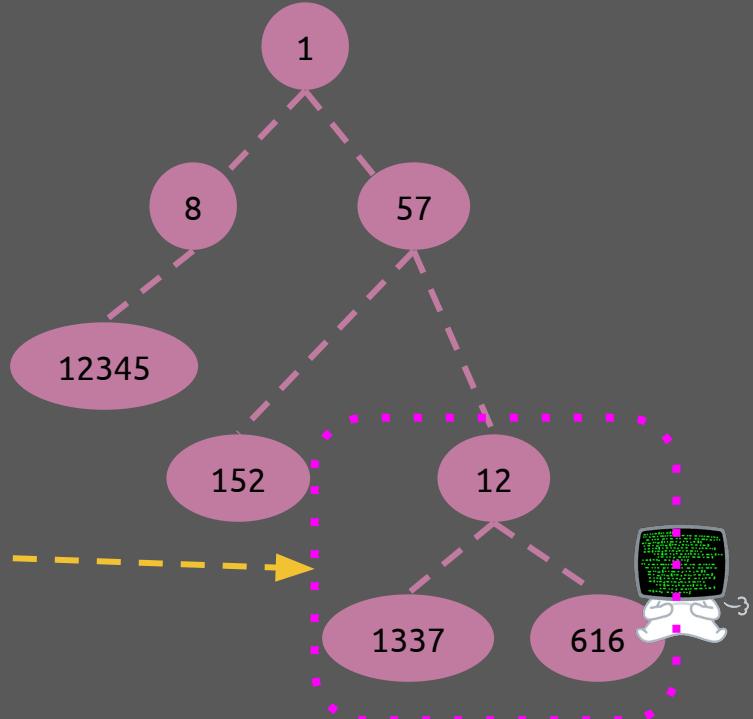


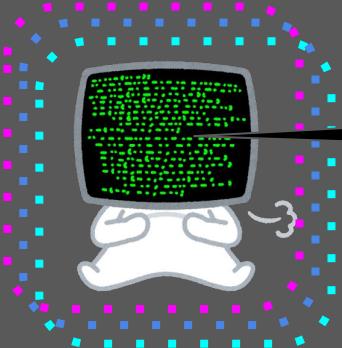


What is my PID?

getpid

Okay, what PID
should this
process see?

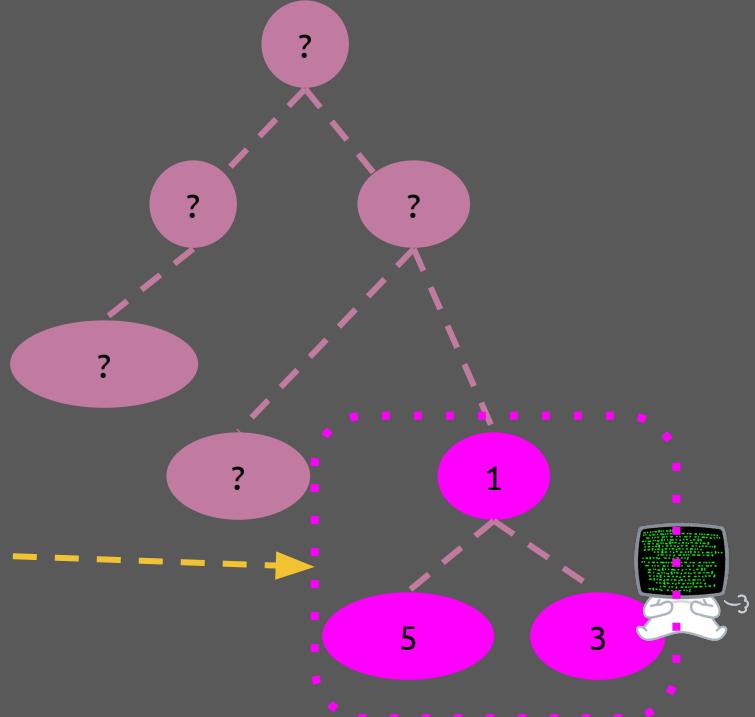


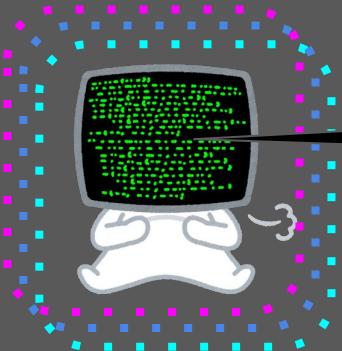


What is my PID?

getpid

Okay, what PID
should this
process see?

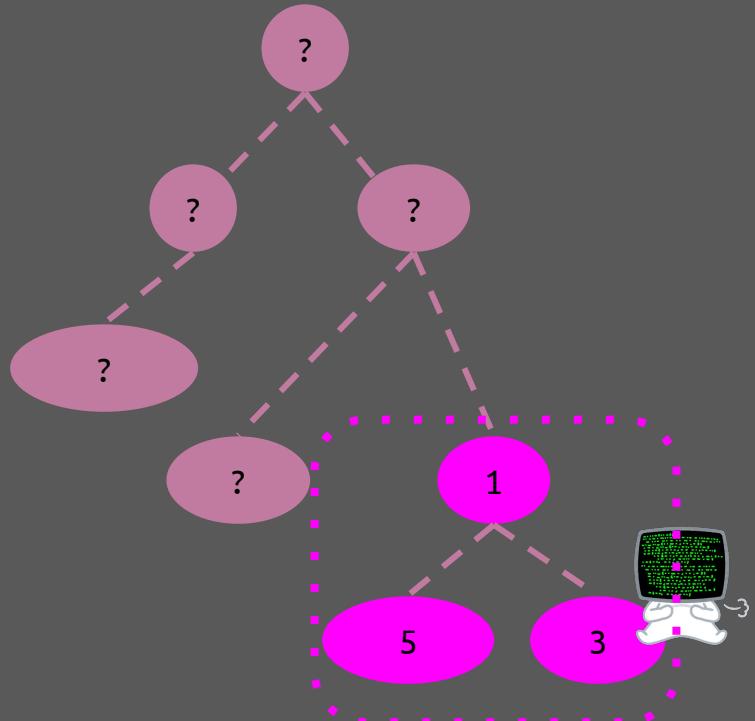


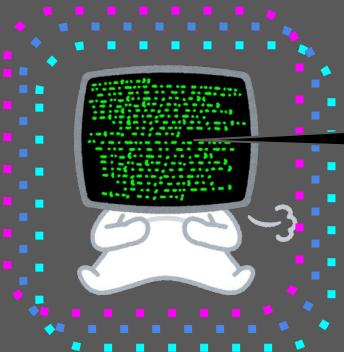


What is my PID?

getpid

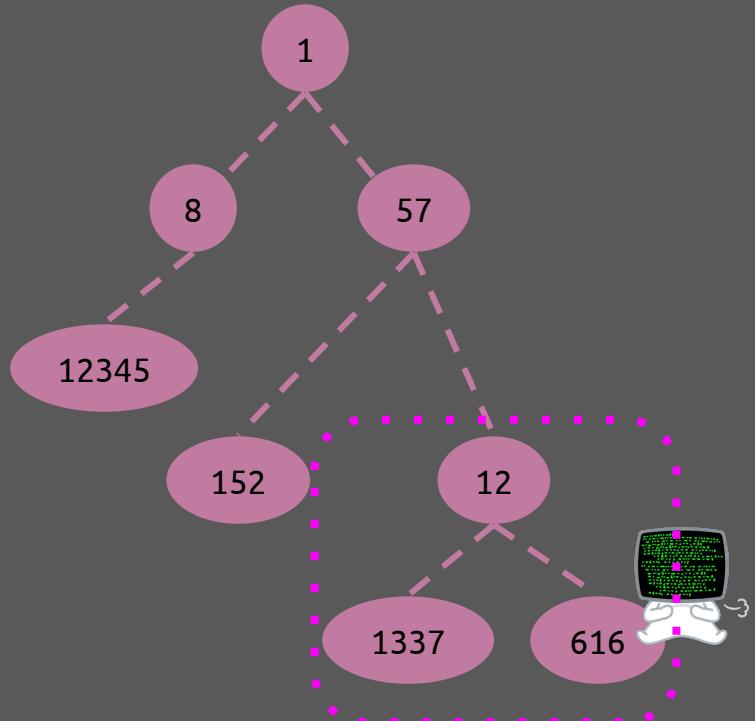
pid 3

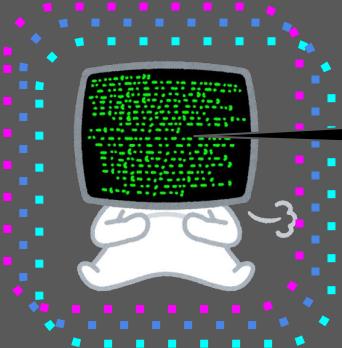




Kill PID 12345.

kill



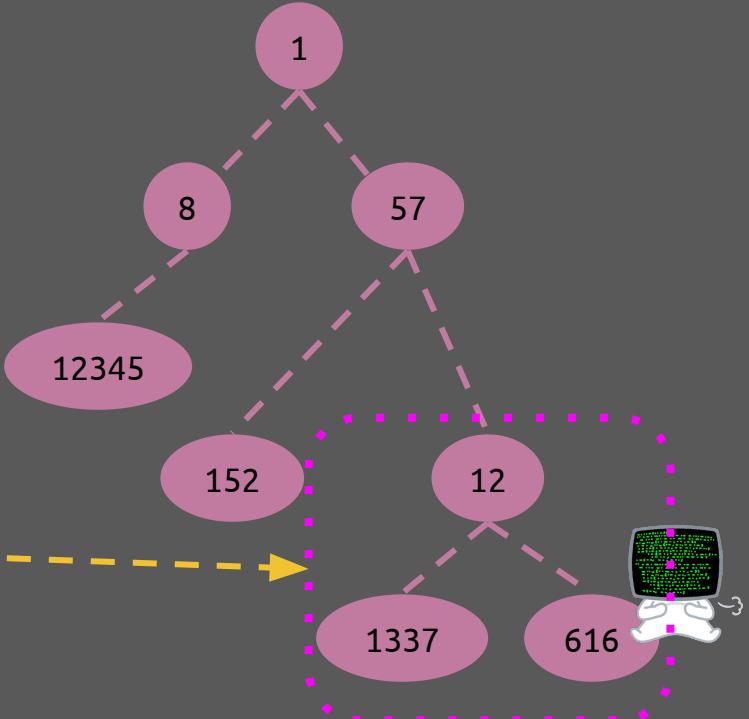


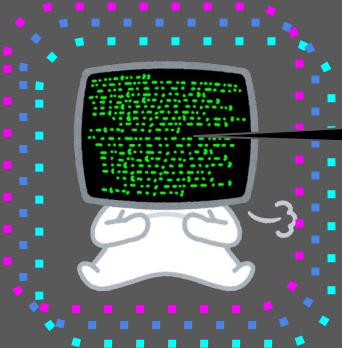
Kill PID 12345.

kill



What PID
namespace is this
process in?



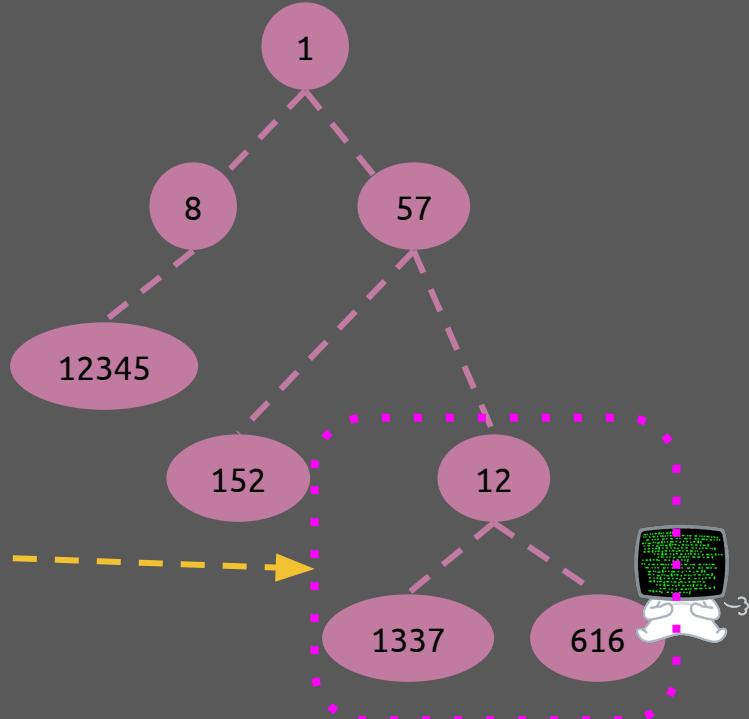


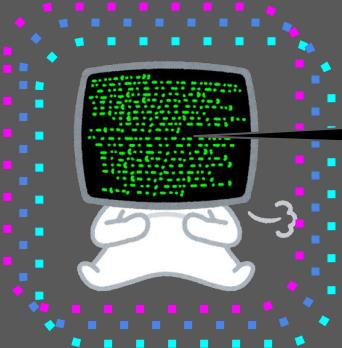
Kill PID 12345.

kill

What PID
namespace is this
process in?

What PIDs should
it see?



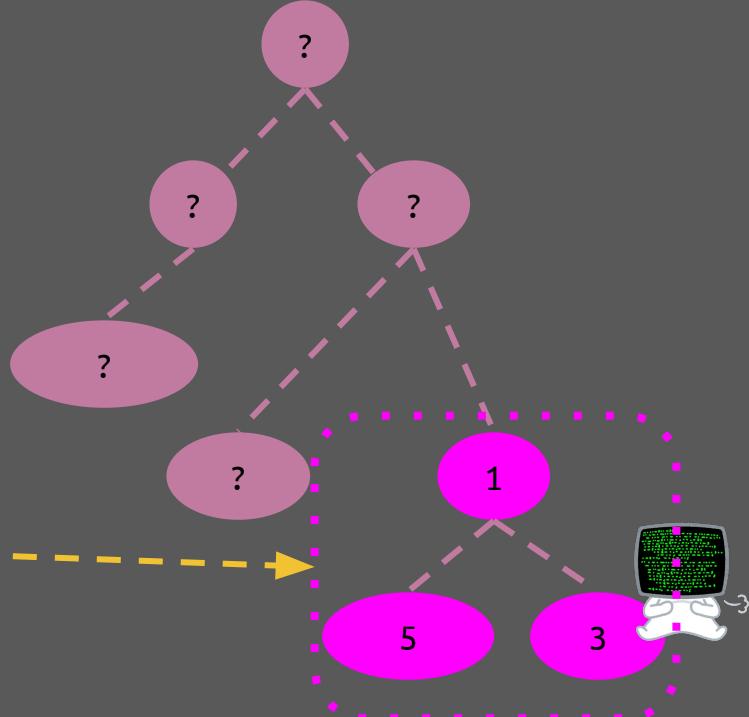


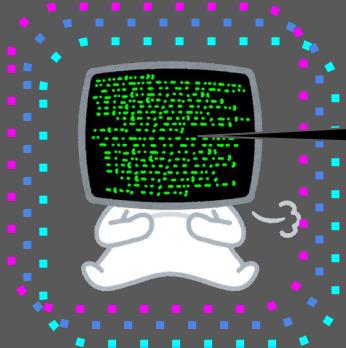
Kill PID 12345.

kill

What PID
namespace is this
process in?

What PIDs should
it see?



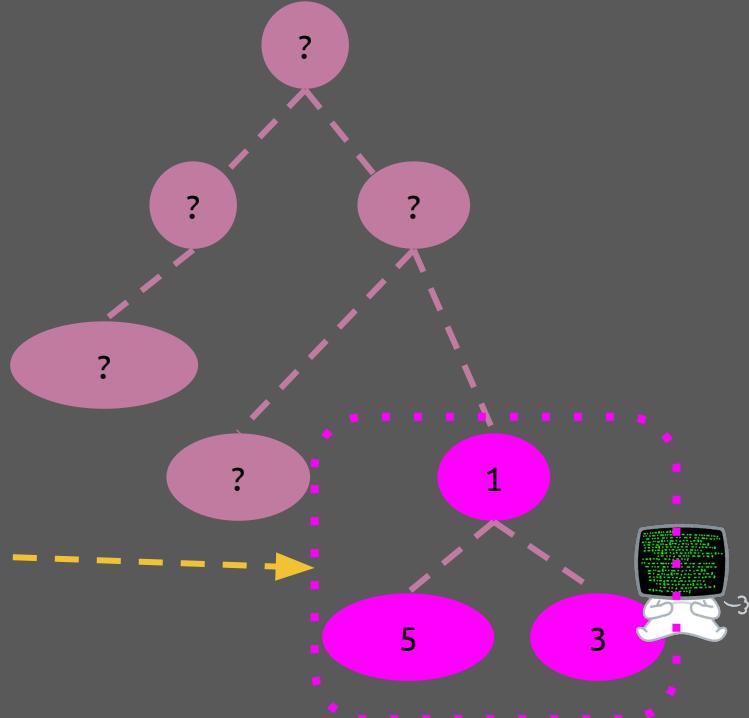


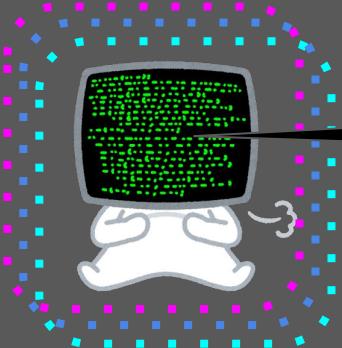
Kill PID 12345.

kill

What PID
namespace is this
process in?

Hmm, where is
PID 12345...



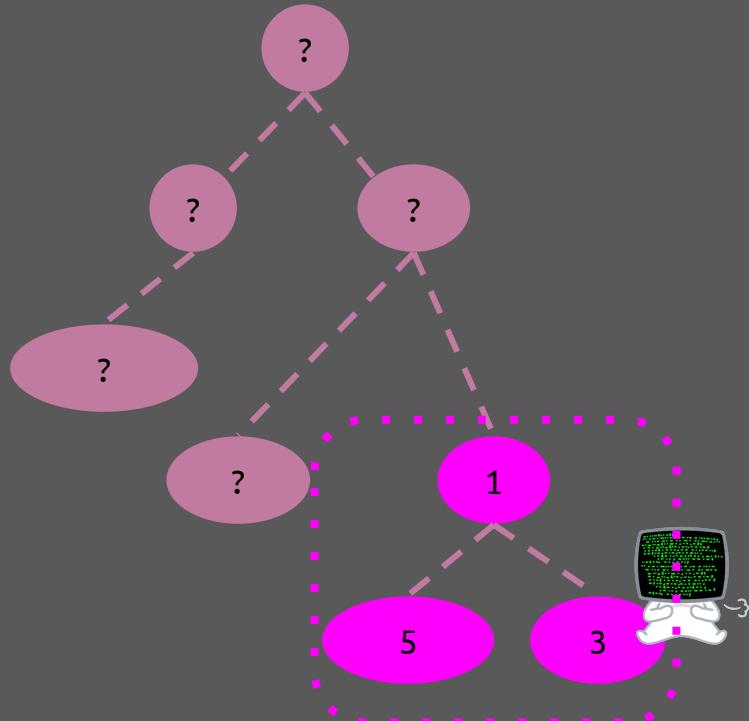


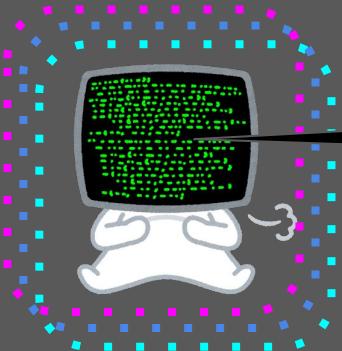
Kill PID 12345.

kill



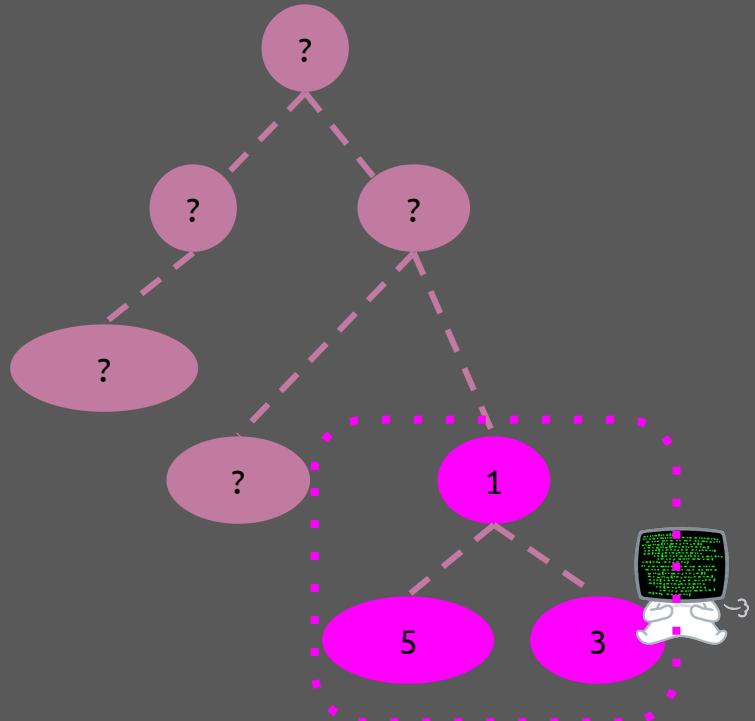
There's no
PID 12345.





Kill PID 12345.

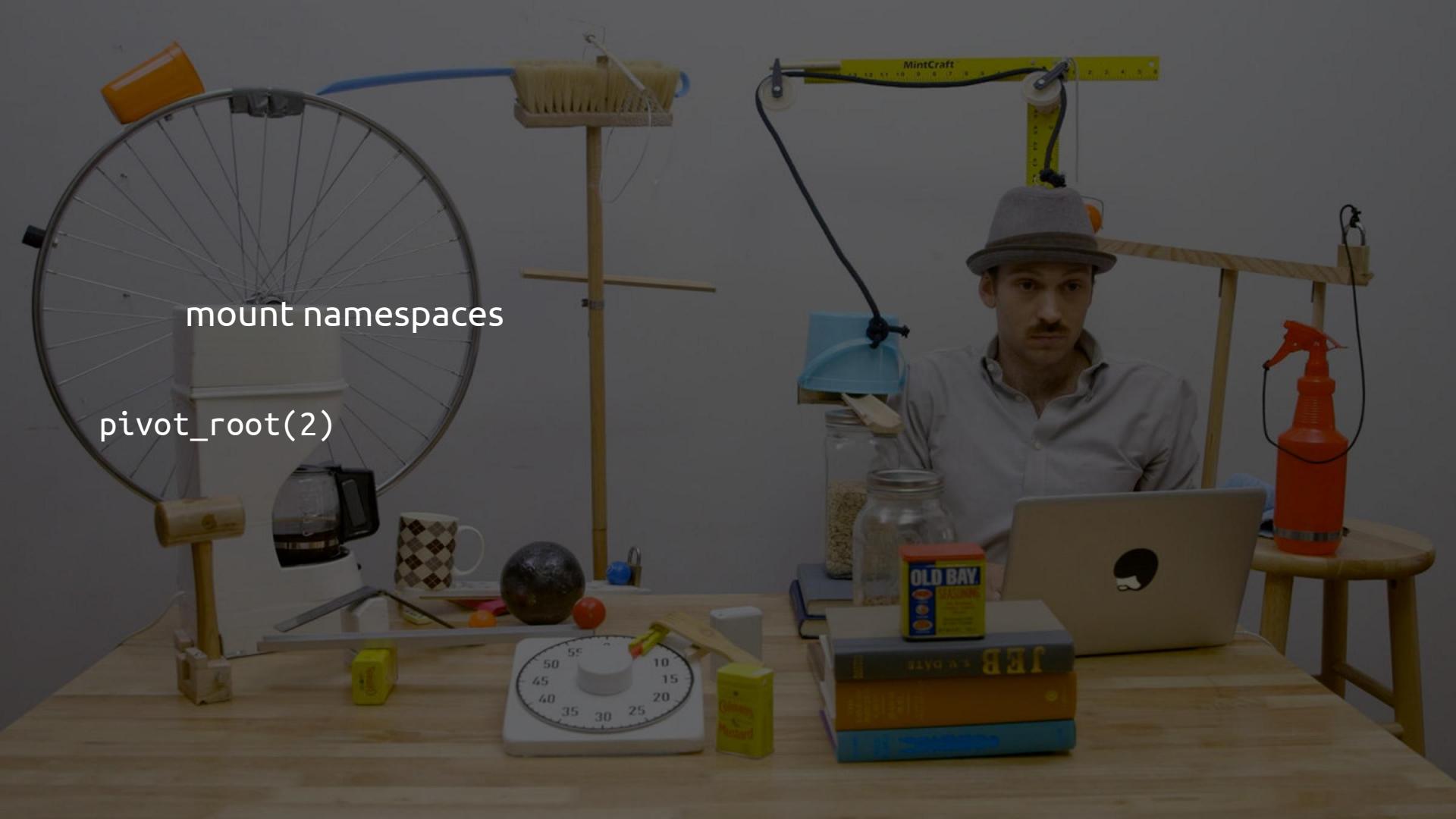
kill | err





chroot(2)





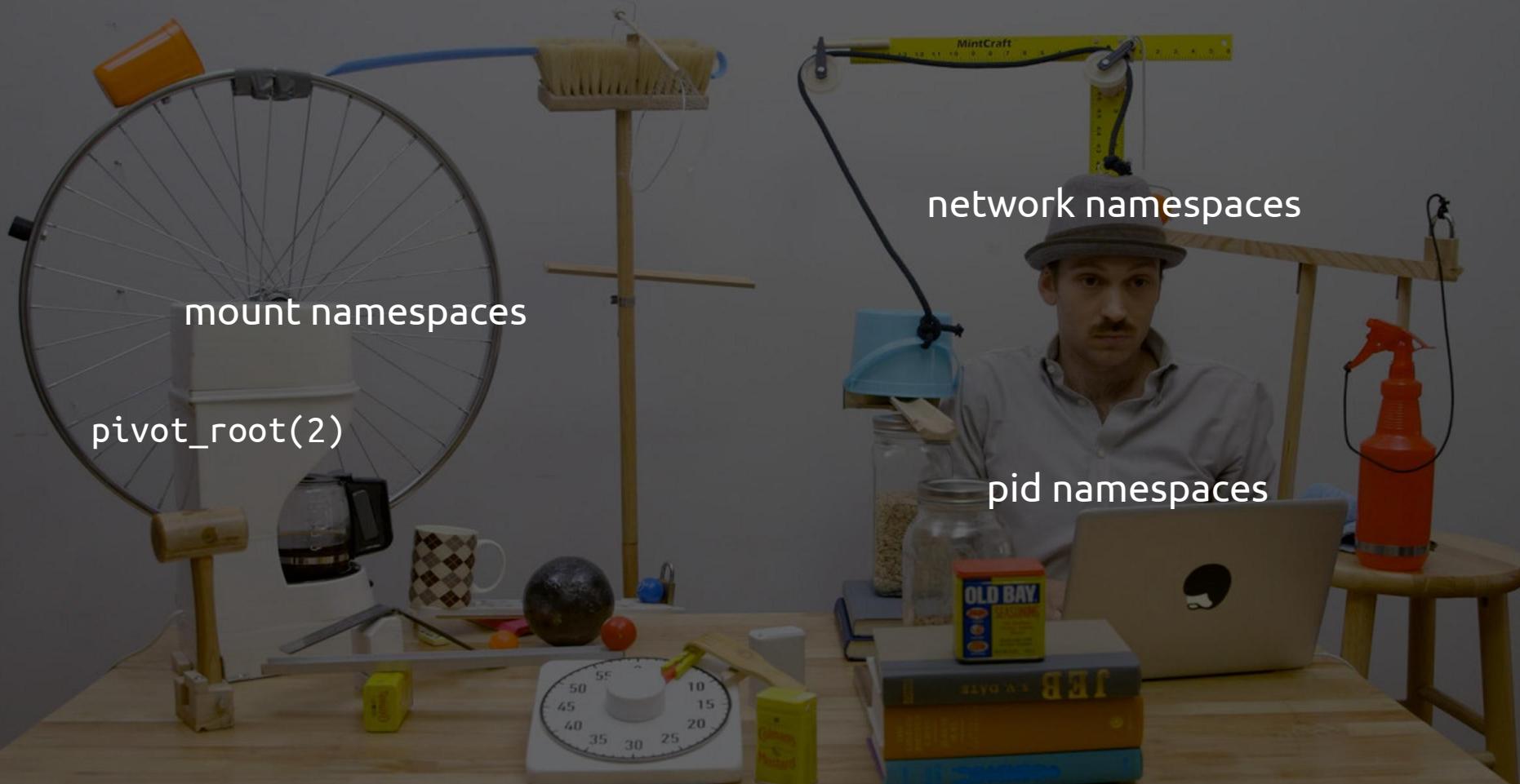
mount namespaces
pivot_root(2)

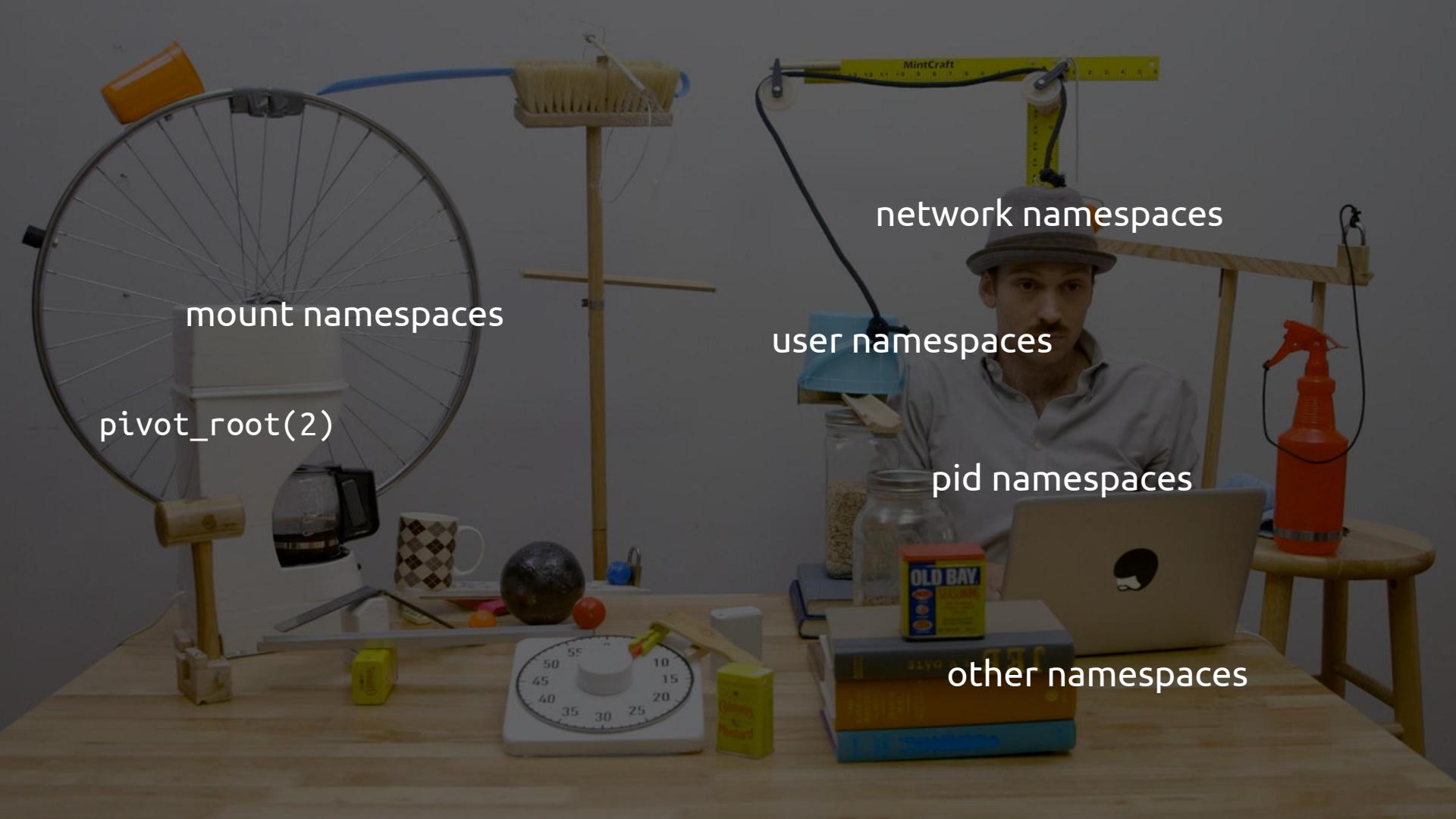
`pivot_root(2)`

`mount namespaces`

`network namespaces`

`pid namespaces`





pivot_root(2)

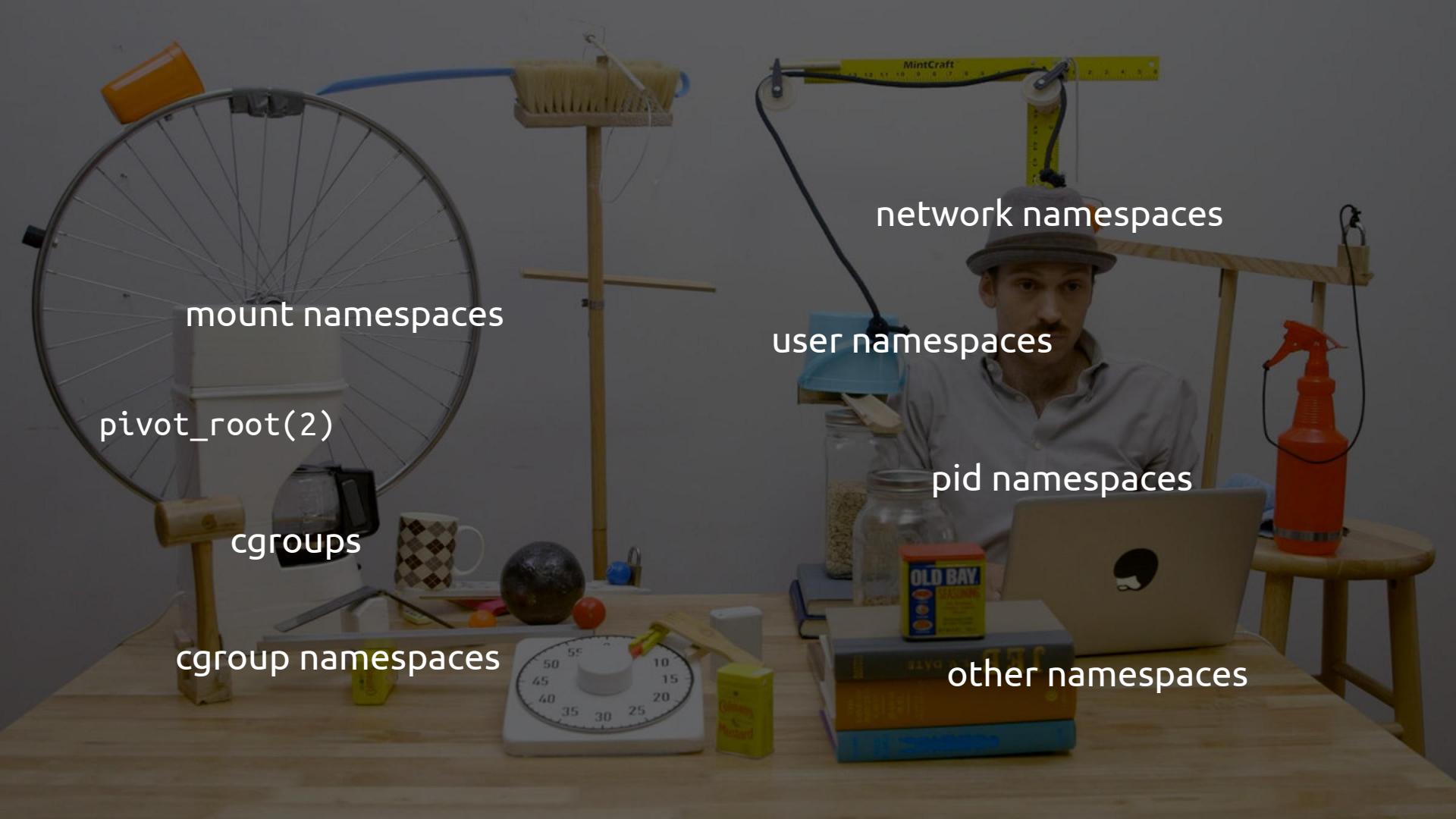
mount namespaces

network namespaces

user namespaces

pid namespaces

other namespaces



pivot_root(2)

cgroups

cgroup namespaces



mount namespaces

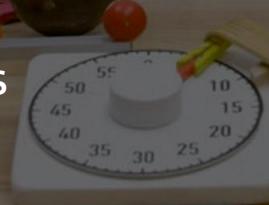


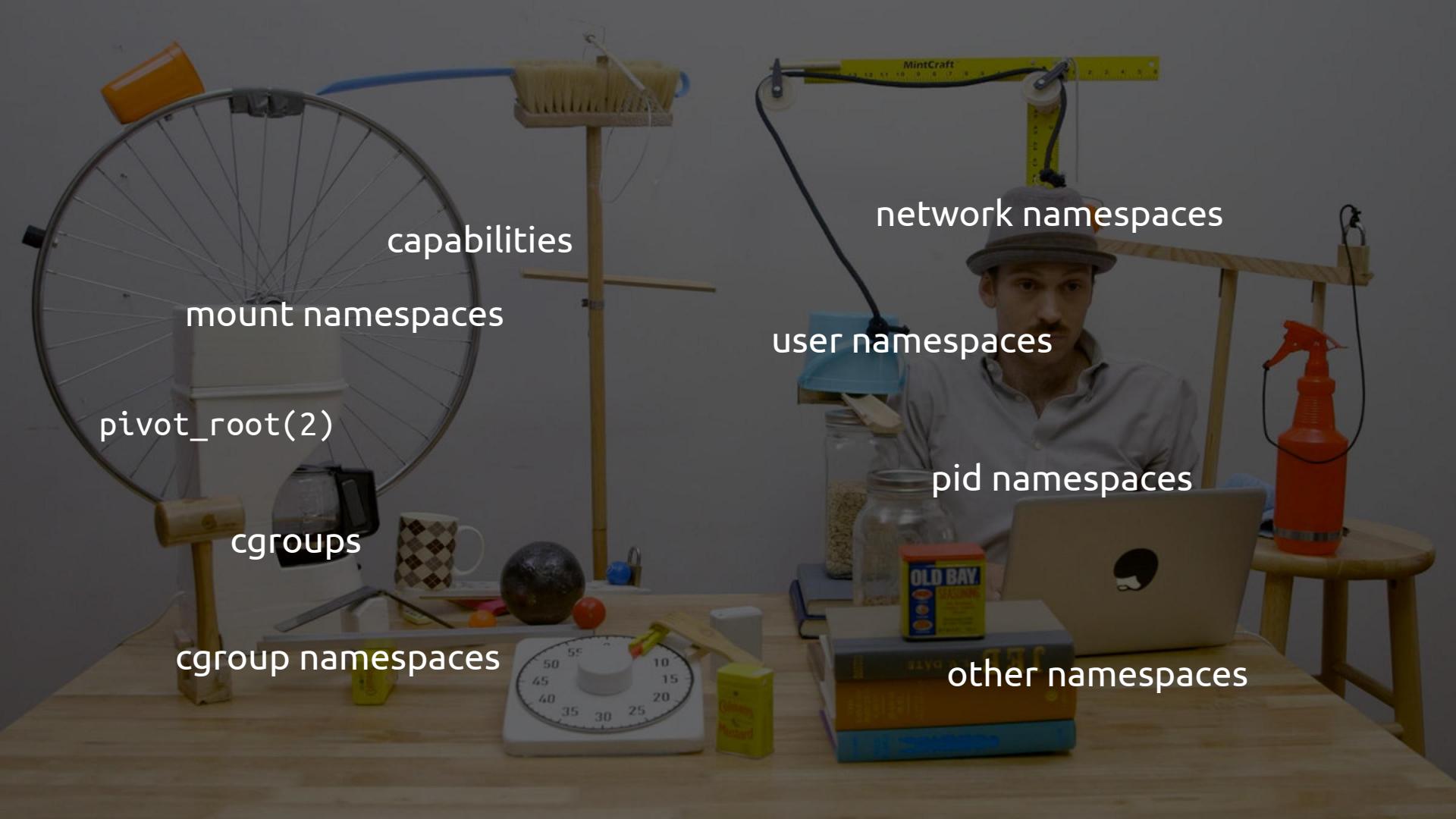
network namespaces

user namespaces

pid namespaces

other namespaces





capabilities

mount namespaces

pivot_root(2)

cgroups

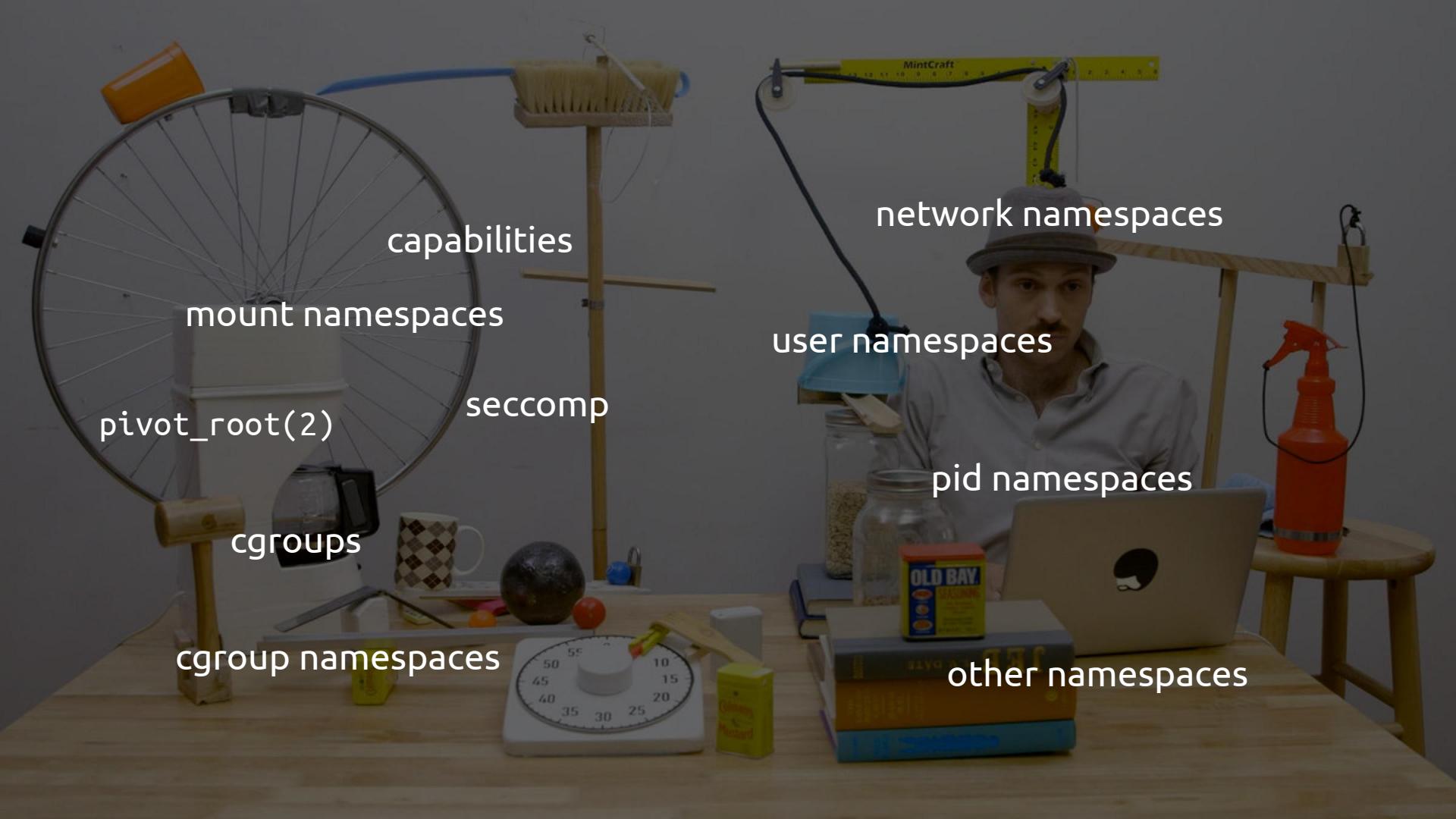
cgroup namespaces

network namespaces

user namespaces

pid namespaces

other namespaces



capabilities
mount namespaces
pivot_root(2)
seccomp

cgroups

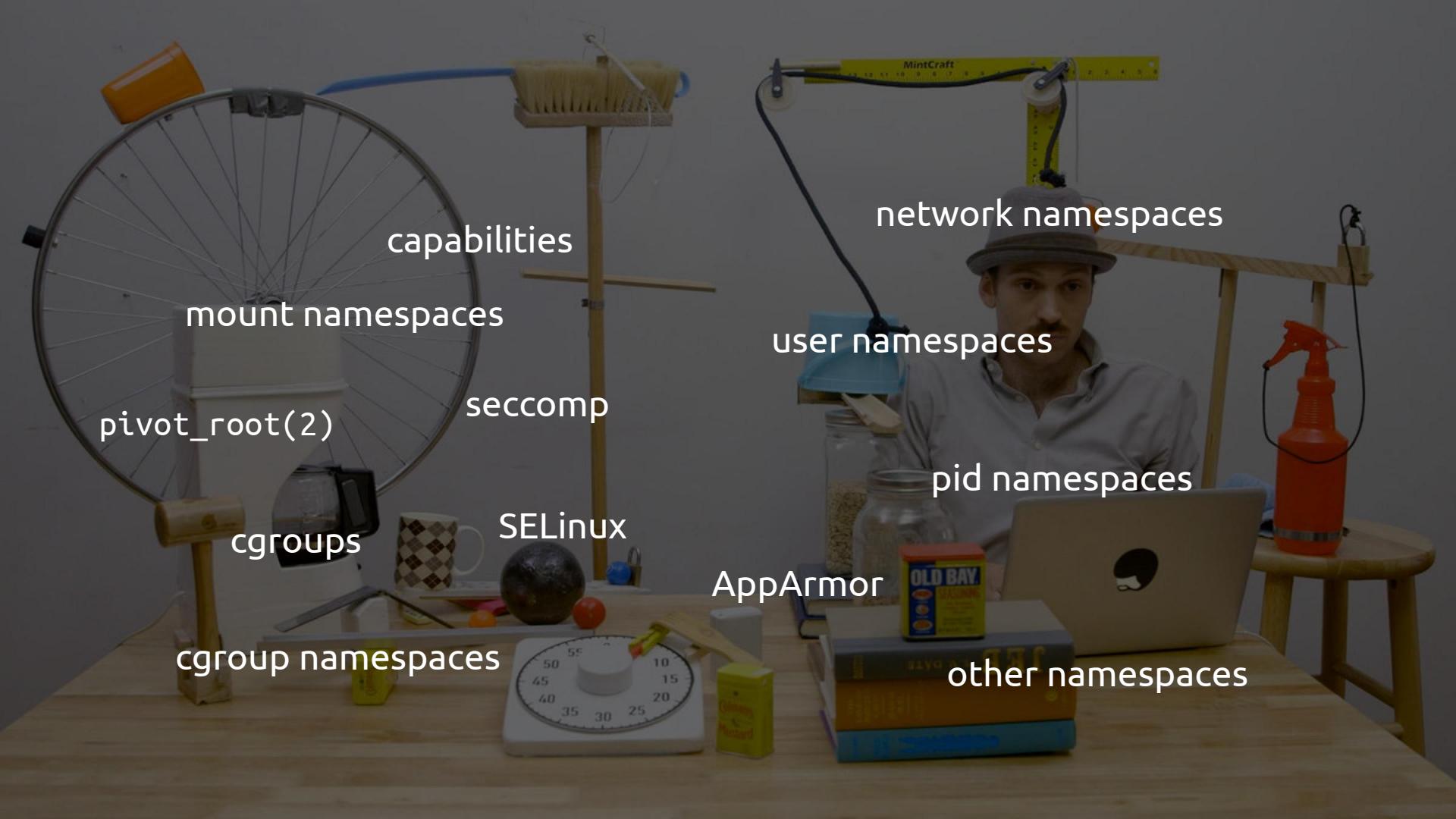
cgroup namespaces

network namespaces

user namespaces

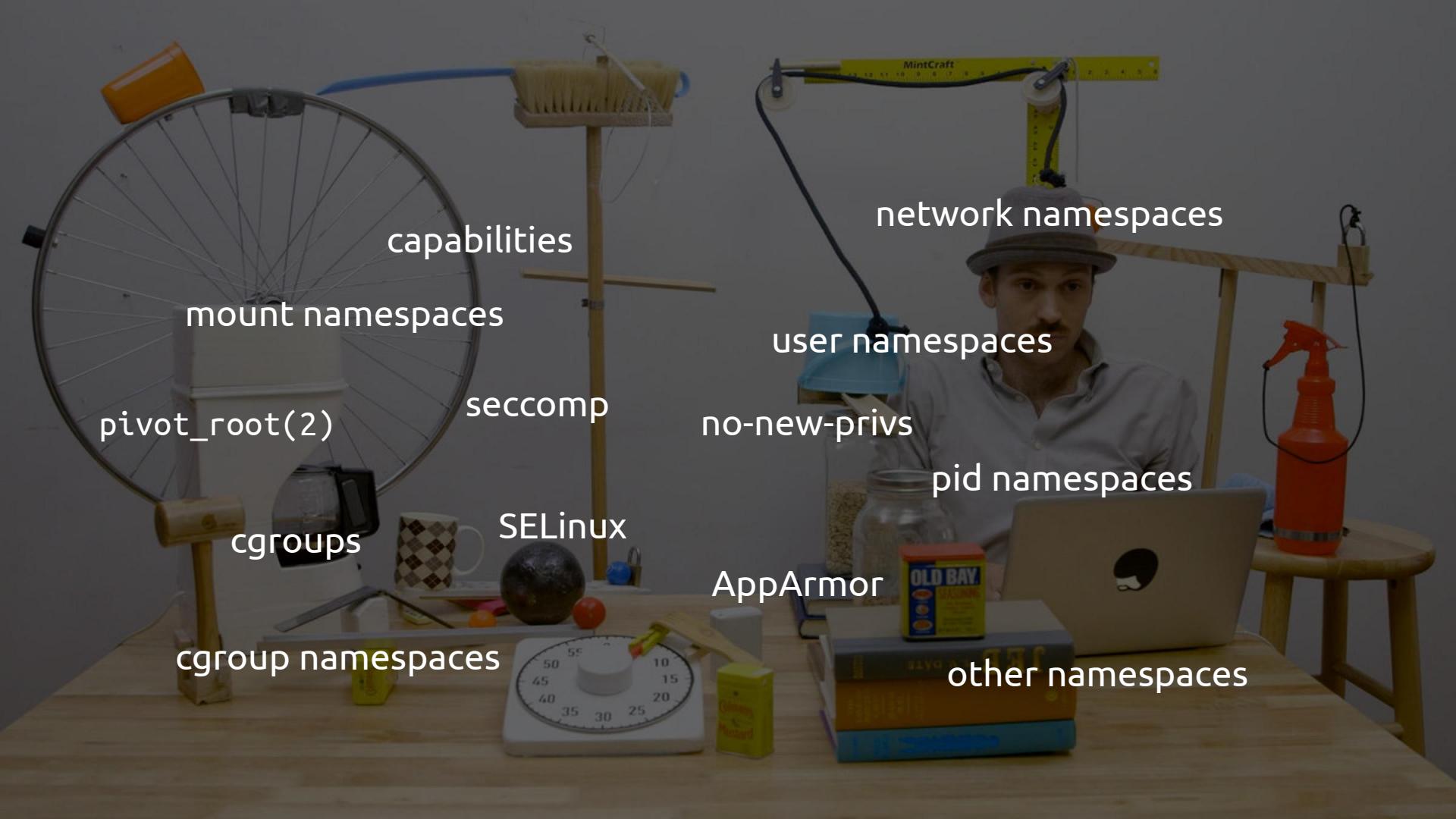
pid namespaces

other namespaces



capabilities
mount namespaces
pivot_root(2)
cgroups
cgroub namespaces
SELinux
seccomp

network namespaces
user namespaces
pid namespaces
AppArmor
other namespaces



capabilities
mount namespaces
pivot_root(2)

seccomp

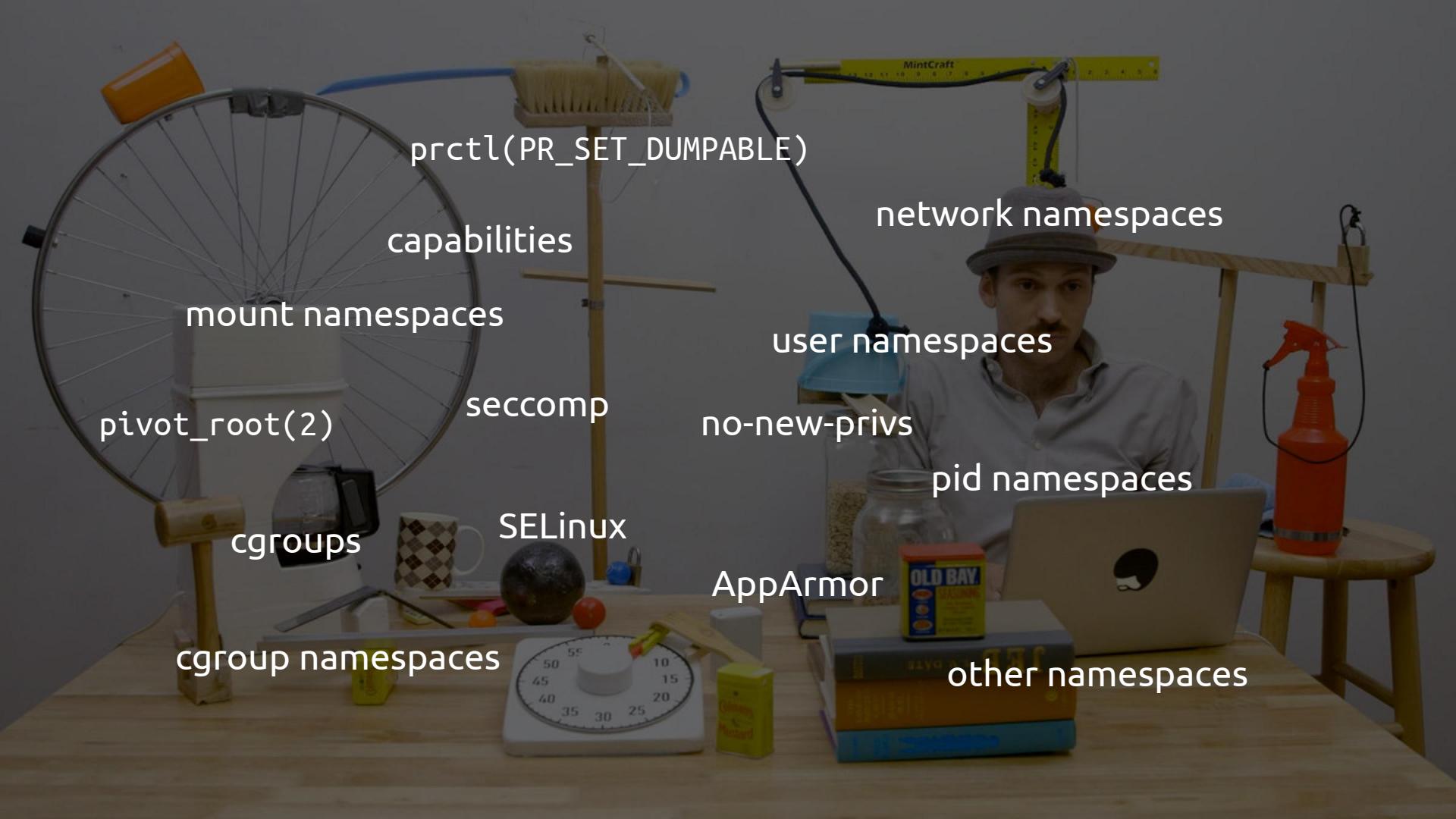
cgroups

cgroup namespaces

SELinux

network namespaces
user namespaces
no-new-privs
pid namespaces
AppArmor
other namespaces

OLD BAY
SEASONING



prctl(PR_SET_DUMPABLE)

capabilities

mount namespaces

pivot_root(2)

cgroups

seccomp

SELinux

cgroup namespaces

network namespaces

user namespaces

no-new-privs

pid namespaces

AppArmor

other namespaces

prctl(PR_SET_DUMPABLE)

capabilities

mount namespaces

pivot_root(2)

cgroups

cgroupl namespaces

seccomp

SELinux

AppArmor

other namespaces

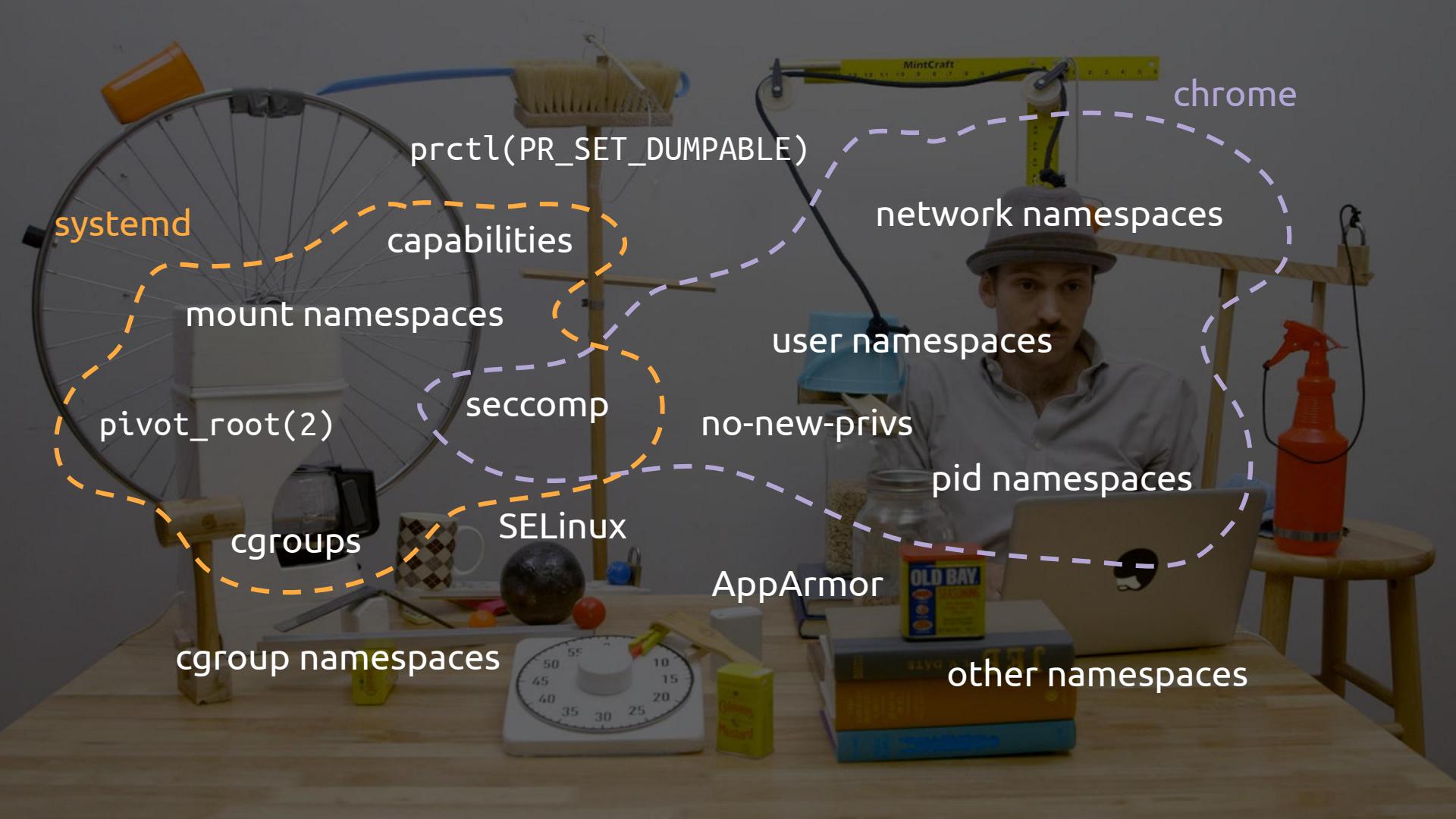
network namespaces

user namespaces

no-new-privs

pid namespaces

chrome



systemd

prctl(PR_SET_DUMPABLE)

capabilities

mount namespaces

pivot_root(2)

cgroups

cgroup namespaces

seccomp

SELinux

network namespaces

user namespaces

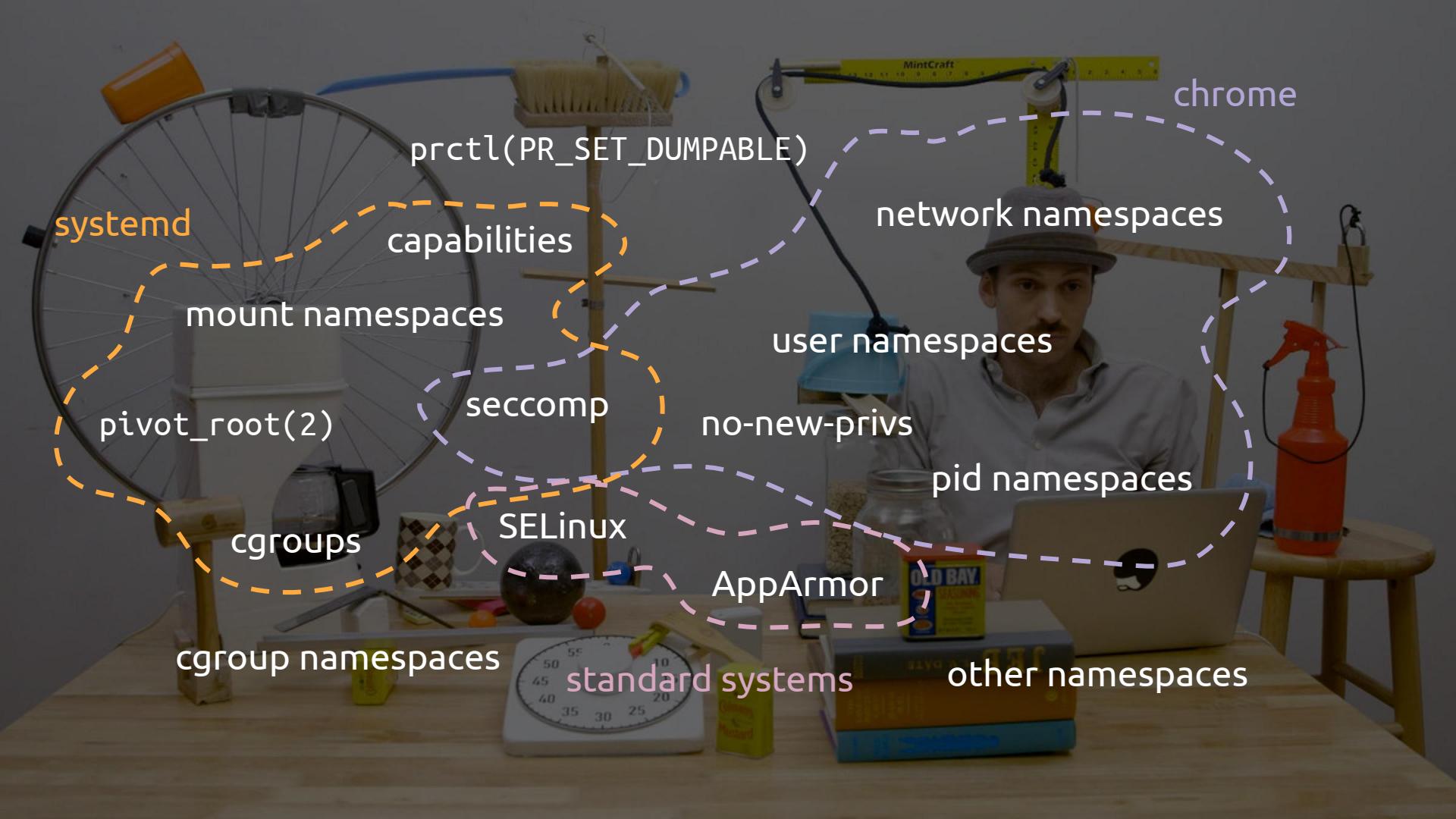
no-new-privs

pid namespaces

AppArmor

other namespaces

chrome



systemd

capabilities

mount namespaces

pivot_root(2)

cgroups

cgroup namespaces

standard systems

SELinux

seccomp

prctl(PR_SET_DUMPABLE)

no-new-privs

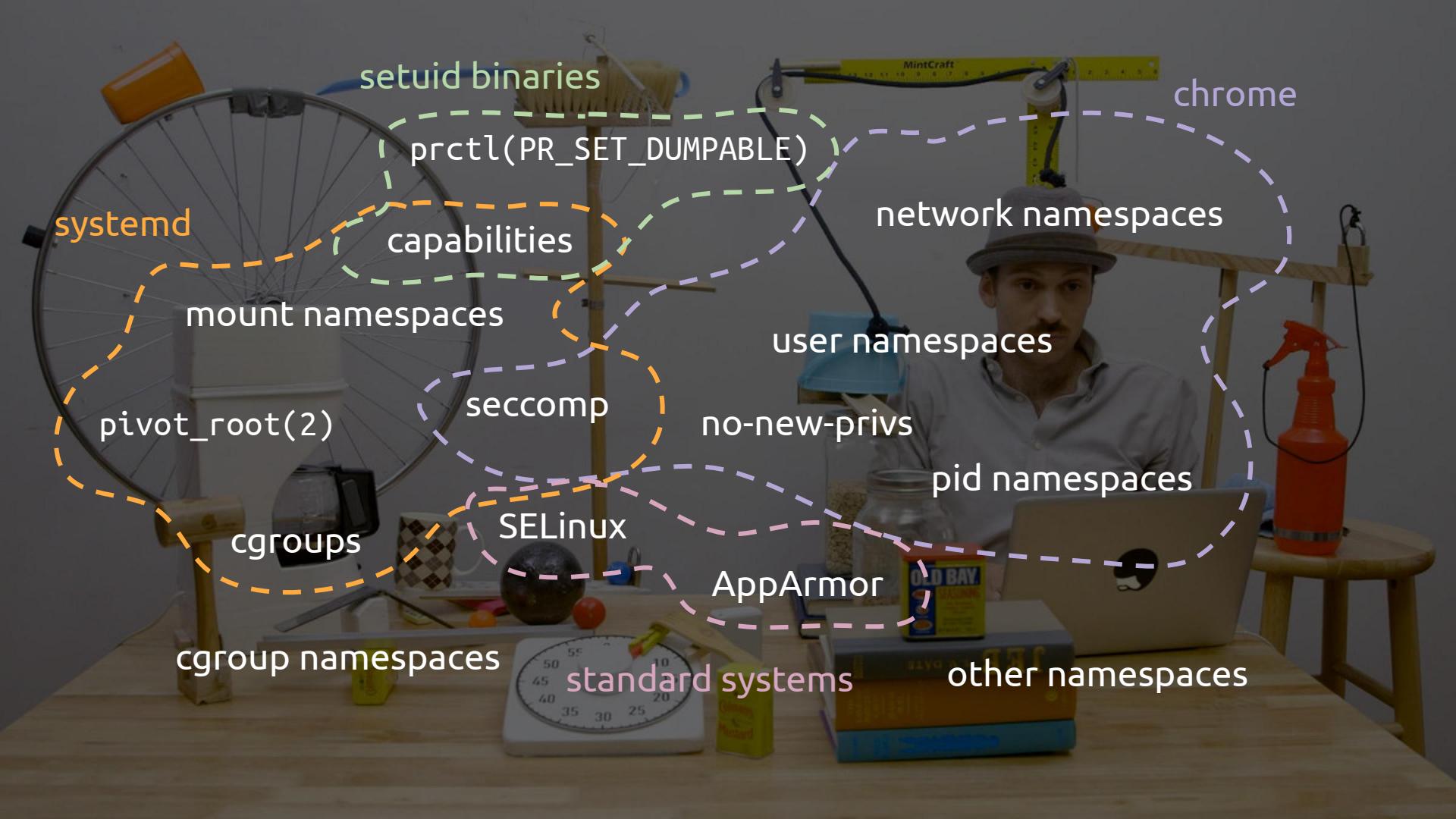
user namespaces

network namespaces

pid namespaces

other namespaces

chrome



systemd

setuid binaries

prctl(PR_SET_DUMPABLE)

capabilities

mount namespaces

pivot_root(2)

cgroups

cgroup namespaces

seccomp

SELinux

standard systems

network namespaces

user namespaces

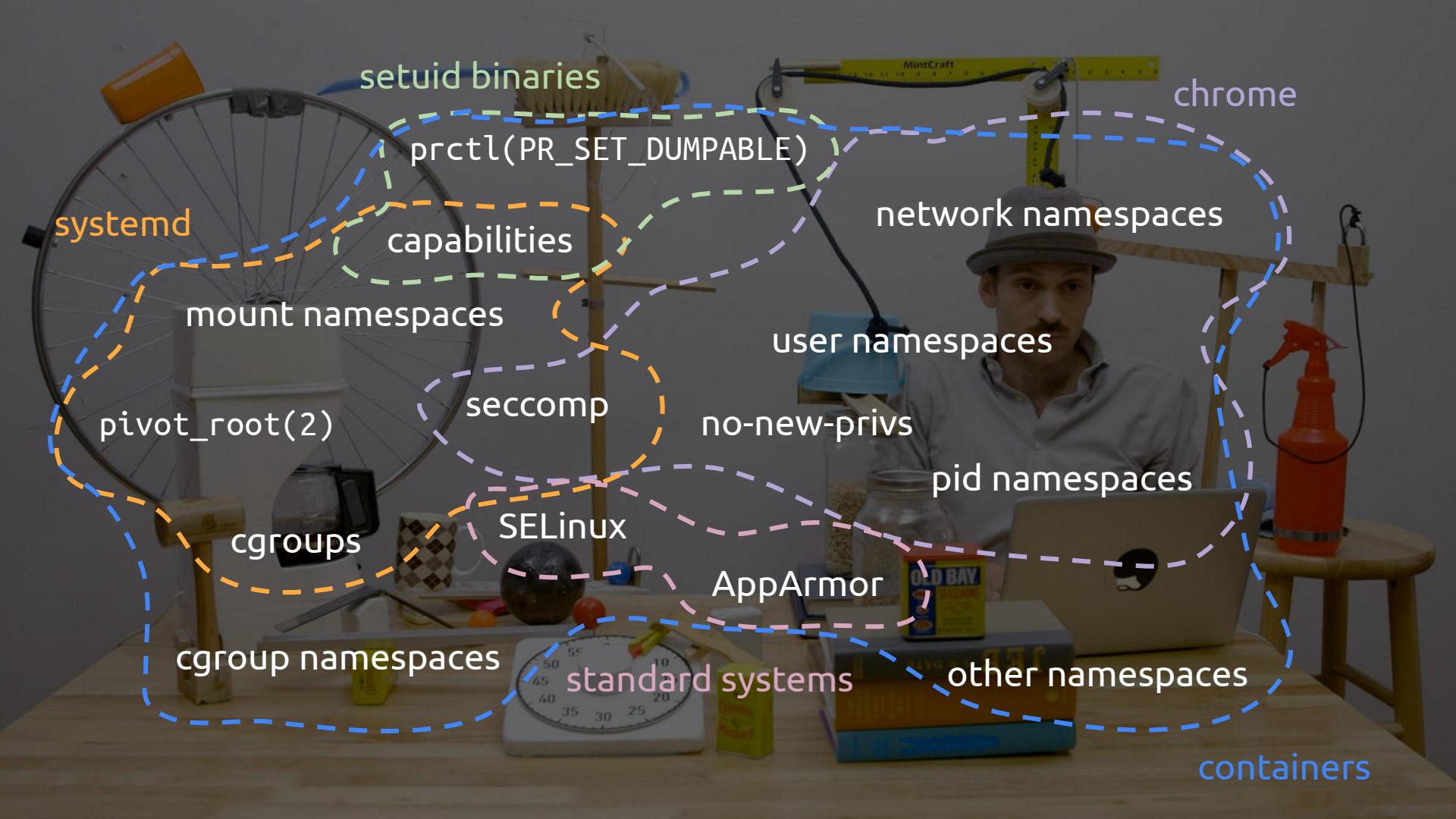
no-new-privs

pid namespaces

AppArmor

other namespaces

chrome





A new foe has appeared

CHALLENGER APPROACH

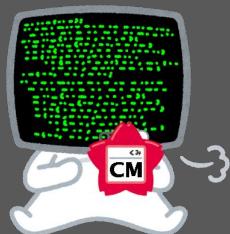
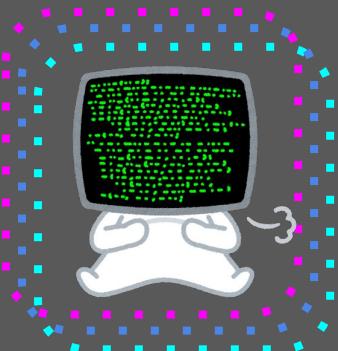








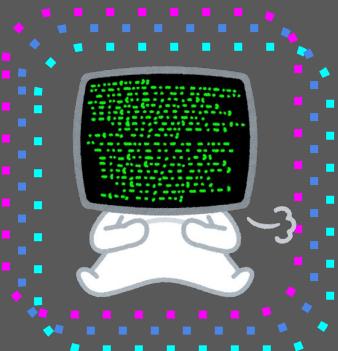
Admin



```
/  
└── etc  
    └── shadow  
---  
└── tmp  
    └── database.txt  
---  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt  
            └── files  
                └── music.mp3
```



I want to check
out
useful-data.txt.



```
/  
└── etc  
    └── shadow  
── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt  
            └── files  
                └── music.mp3
```



I want to check
out
useful-data.txt.
...but I can't trust
the container...



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt  
            └── files  
                └── music.mp3
```



Hey, container manager!



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt  
            └── files  
                └── music.mp3
```



Hey, container manager!

Can you get
useful-data.txt for me?



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── useful-data.txt  
                └── files  
                    └── music.mp3
```

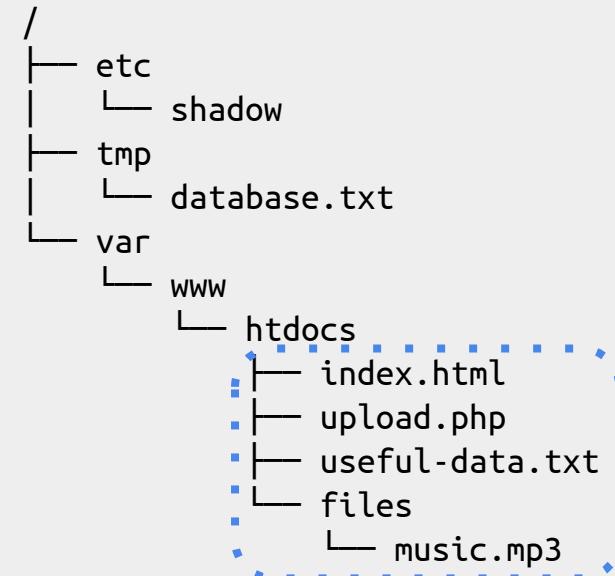


Hey, container manager!

Can you get
useful-data.txt for me?



Sure!



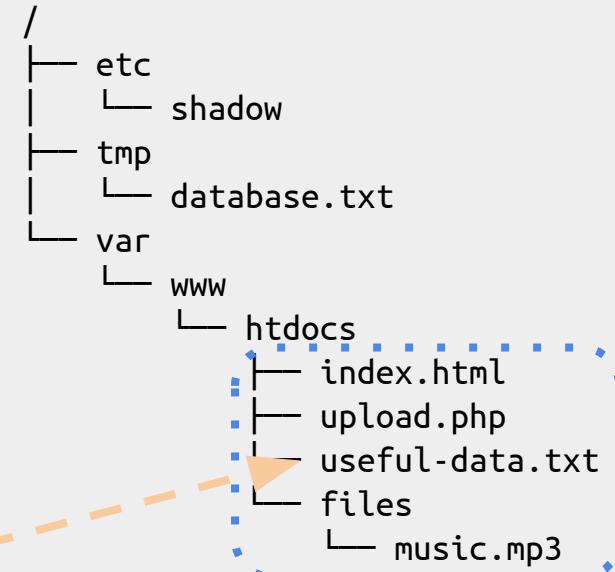


Hey, container manager!

Can you get
useful-data.txt for me?



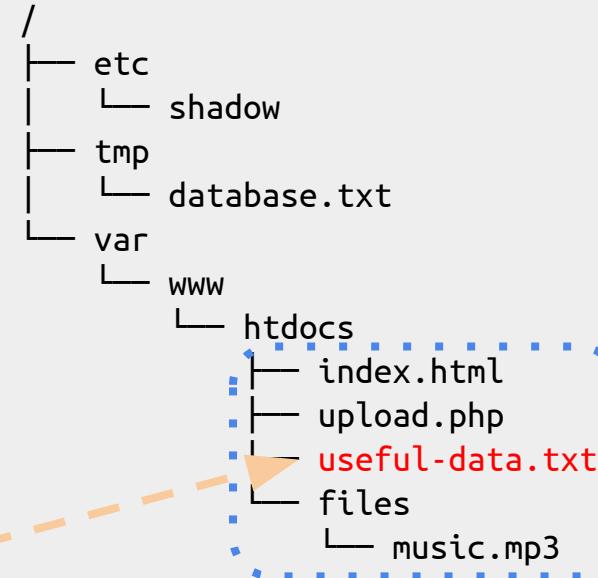
Sure!





Hey, **container manager!**

Can you get
useful-data.txt for me?





They always forget about symlinks!

```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            └── files  
                └── music.mp3
```



Nobody expects a symlink!

```
├── upload.php  
├── useful-data.txt -> /etc/shadow  
└── files  
    └── music.mp3
```



```
open("/var/www/htdocs/useful-data.txt")
```

Let's look that up...



open

```
/var/www/htdocs/useful-data.txt
```

```
/  
|   └── etc  
|       └── shadow  
|   └── tmp  
|       └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── useful-data.txt  
            └── files  
                └── music.mp3
```



```
open("/var/www/htdocs/useful-data.txt")
```

Let's look that up...



open

```
/var/www/htdocs/useful-data.txt
```

```
/  
└── etc  
    └── shadow  
── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── useful-data.txt  
                └── files  
                    └── music.mp3
```



```
open("/var/www/htdocs/useful-data.txt")
```

Let's look that up...

Ah, this points to /etc/shadow.

open



```
/var/www/htdocs/useful-data.txt
```

```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── useful-data.txt  
                └── files  
                    └── music.mp3
```



```
open("/var/www/htdocs/useful-data.txt")
```

open

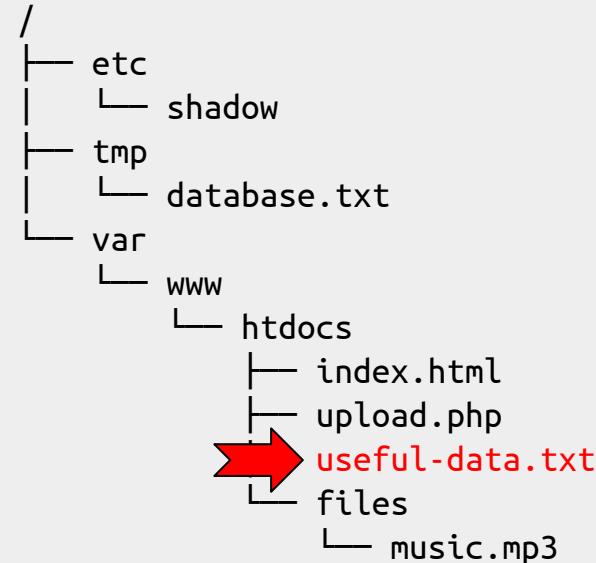


Let's look that up...

Ah, this points to /etc/shadow.



/etc/shadow





```
open("/var/www/htdocs/useful-data.txt")
```

open

Let's look that up...



/etc/shadow

```
/  
└── etc  
    └── shadow  
        └── database.txt  
└── tmp  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── useful-data.txt  
        └── files  
            └── music.mp3
```



```
open("/var/www/htdocs/useful-data.txt")
```

Let's look that up...

Ah, this points to
.../.../.../foo.

open



```
/var/www/htdocs/useful-data.txt
```

```
/  
└── etc  
    └── shadow  
── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── useful-data.txt  
                └── files  
                    └── music.mp3
```



```
open("/var/www/htdocs/useful-data.txt")
```

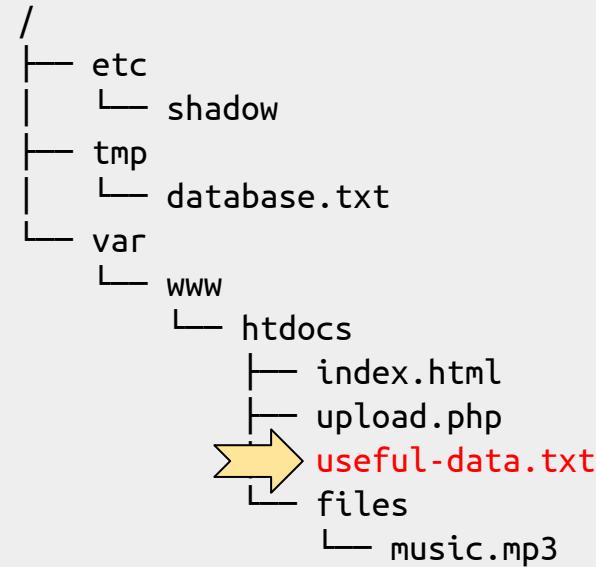
open

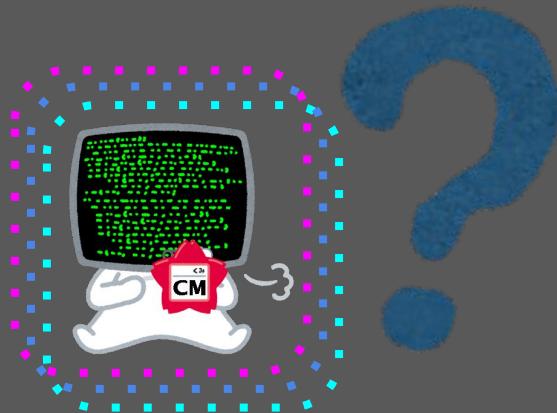
Let's look that up...

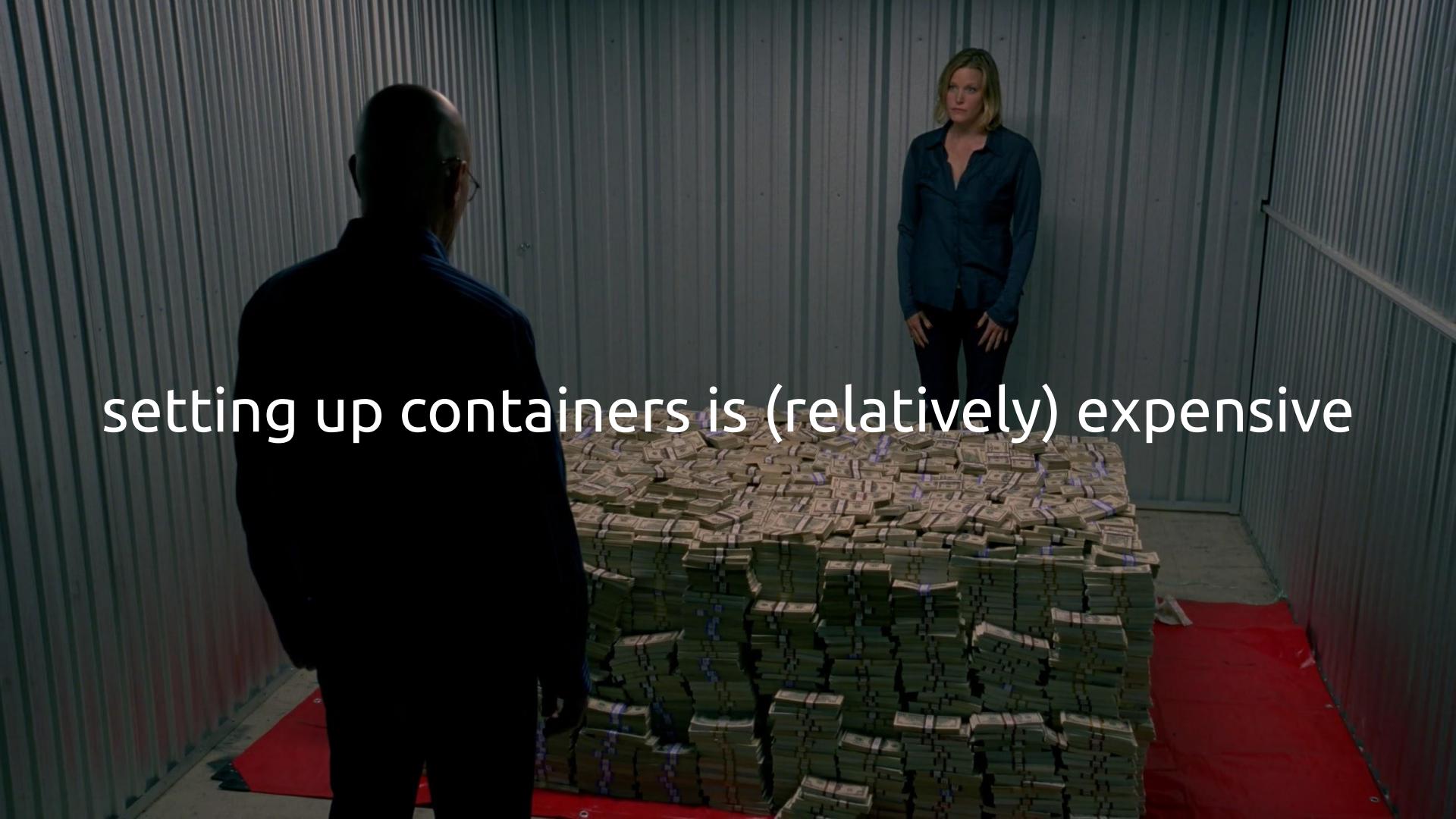
Ah, this points to
.../.../.../foo.



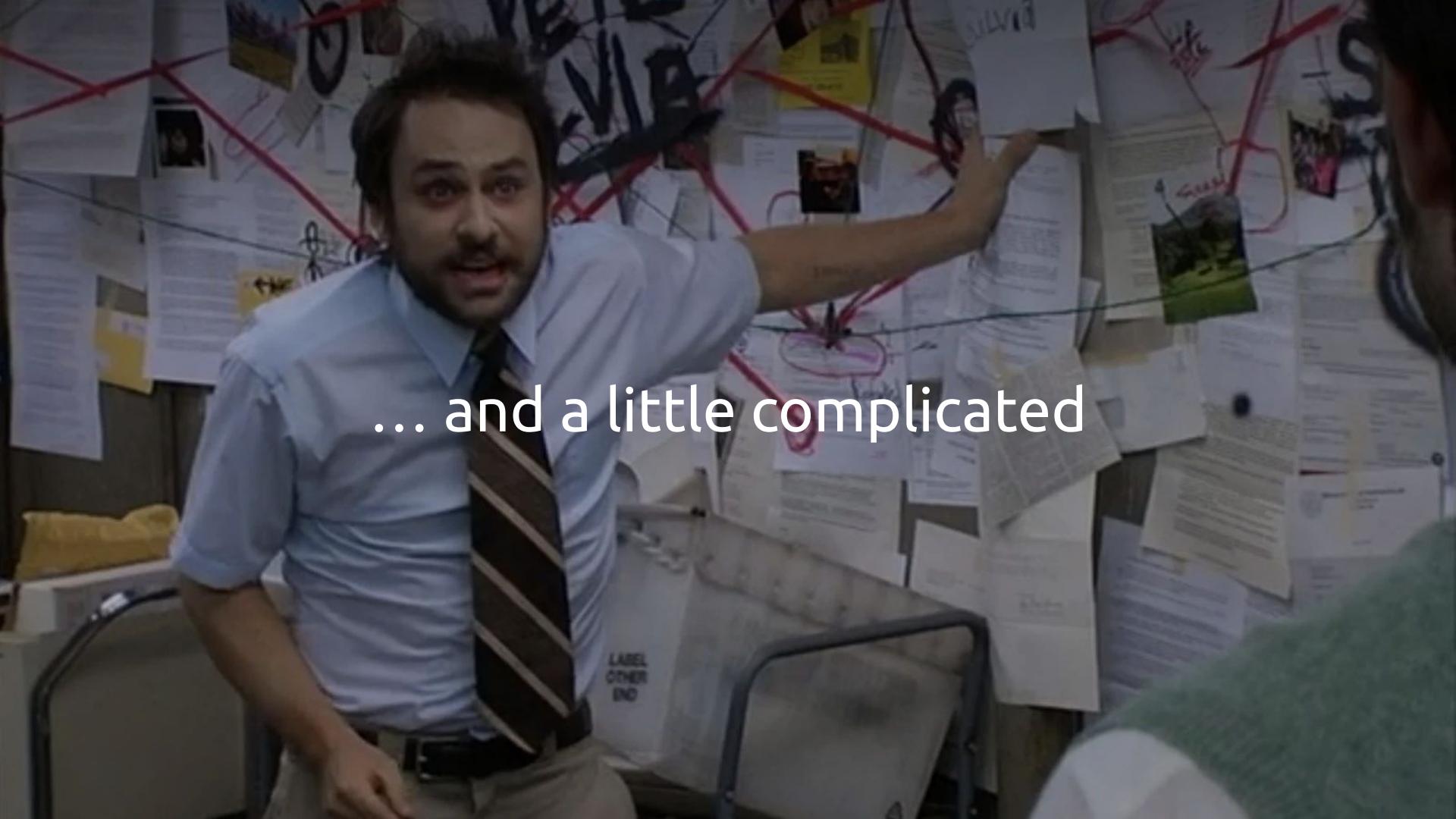
/var/www/htdocs/.../.../.../foo





A scene from a TV show or movie showing a man and a woman standing in a shipping container. The man is in the foreground, seen from behind, wearing a dark suit and glasses. The woman stands further back, looking towards the camera. Between them is a massive, towering pile of US dollar bills, stacked in thousands of individual bills. The container floor is covered with a red protective mat.

setting up containers is (relatively) expensive

A man with a beard and mustache, wearing a light blue short-sleeved shirt and a striped tie, stands in a cubicle. He is surrounded by numerous white papers pinned to the wall behind him with red 'X' marks. Some of the papers have handwritten names like 'SILVA' and 'GREG'. A small photograph of a dog is also visible. The man is looking directly at the camera with a slightly weary expression.

... and a little complicated

“just sanitise the path”





Before we actually do
open("/var/www/htdocs/useful-data.txt"),
let me sanitise the path first.



Before we actually do
`open("/var/www/htdocs/useful-data.txt"),`
let me sanitise the path first.

Ah, `useful-data.txt` was a symlink to `/etc/shadow`,
so I guess we should do
`open("/var/www/htdocs/etc/shadow")`.



Before we actually do
`open("/var/www/htdocs/useful-data.txt"),`
let me sanitise the path first.

Ah, `useful-data.txt` was a symlink to `/etc/shadow`,
so I guess we should do
`open("/var/www/htdocs/etc/shadow")`.

Nothing can possibly go wrong, right?



You can't beat me that easily!

```
/  
├── etc  
│   └── shadow  
├── tmp  
│   └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            └── files  
                └── music.mp3
```





You can't beat me that easily!

```
/  
├── etc  
│   └── shadow  
├── tmp  
│   └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            ├── dummy-file.txt  
            └── files  
                └── music.mp3
```





You can't beat me that easily!

```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            ├── dummy-file.txt  
            └── files  
                └── music.mp3
```





You can't beat me that easily!

```
/  
├── etc  
│   └── shadow  
├── tmp  
│   └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt  
            ├── dummy-file.txt -> /etc/shadow  
            └── files  
                └── music.mp3
```





You can't beat me that easily!

```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            ├── dummy-file.txt  
            └── files  
                └── music.mp3
```





You can't beat me that easily!

```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt  
            ├── dummy-file.txt -> /etc/shadow  
            └── files  
                └── music.mp3
```





You can't beat me that easily!

Let me sanitise
the path first.



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            ├── dummy-file.txt  
            └── files  
                └── music.mp3
```





You can't beat me that easily!

No symlinks found!



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt  
            └── dummy-file.txt -> /etc/shadow  
        └── files  
            └── music.mp3
```





You can't beat me that easily!

open("../useful-data.txt")



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            ├── dummy-file.txt  
            └── files  
                └── music.mp3
```





You can't beat me that easily!

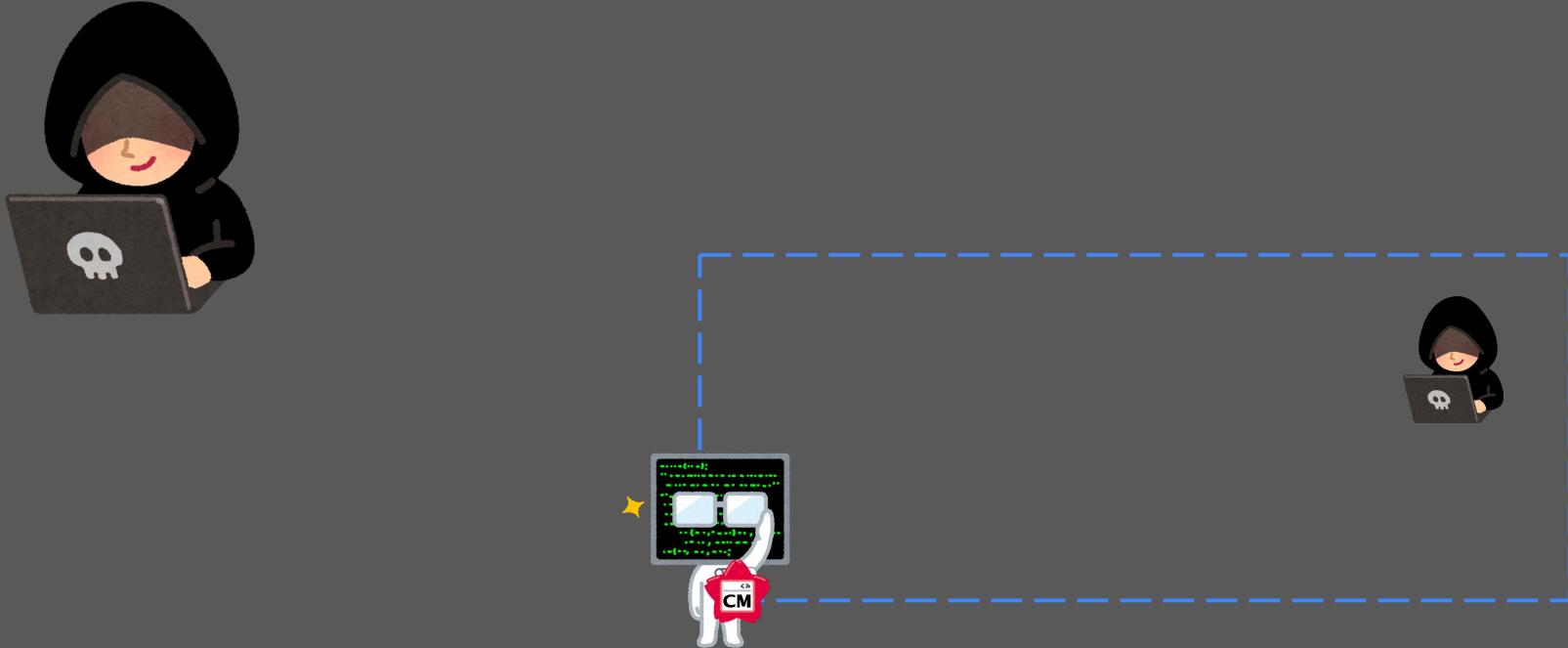
open(".".
Oh no!
a.txt")



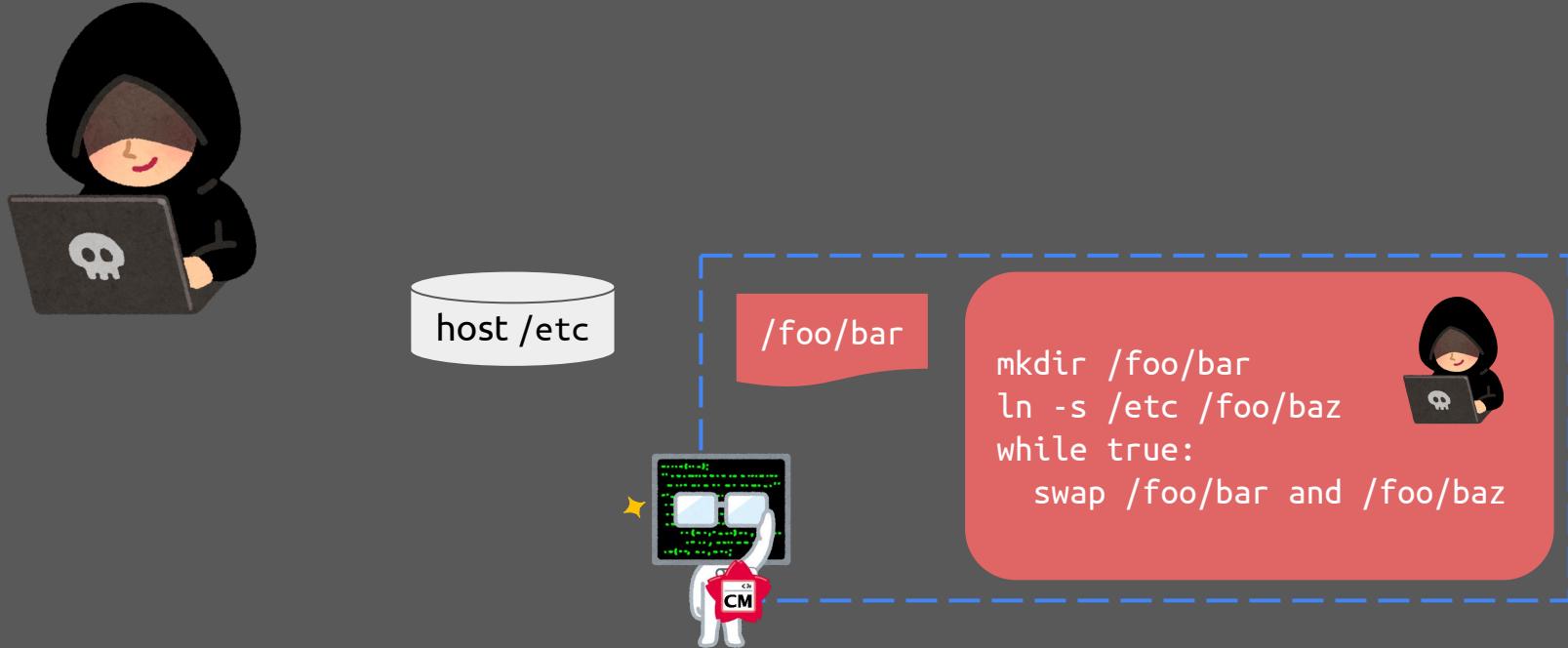
```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            └── useful-data.txt -> /etc/shadow  
            └── dummy-file.txt  
            └── files  
                └── music.mp3
```



CVE-2018-15664



CVE-2018-15664



CVE-2018-15664



```
docker cp ctr:/foo/bar/shadow .
docker cp myshadow ctr:/foo/bar/shadow
```

host /etc

/foo/bar



```
mkdir /foo/bar
ln -s /etc /foo/baz
while true:
    swap /foo/bar and /foo/baz
```



CVE-2018-15664



```
docker cp ctr:/foo/bar/shadow .
docker cp myshadow ctr:/foo/bar/shadow
```

host /etc

/foo/bar

Oh no!



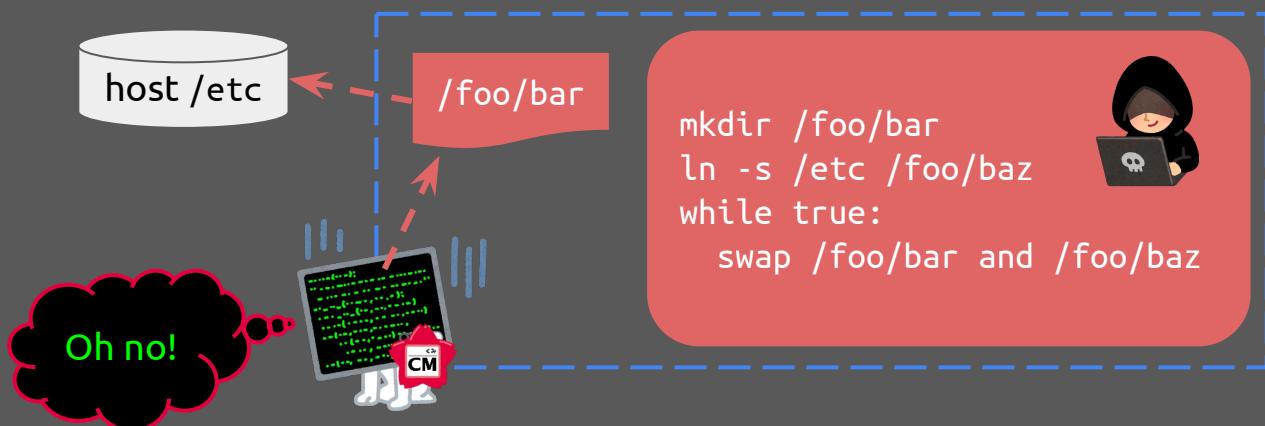
```
mkdir /foo/bar
ln -s /etc /foo/baz
while true:
    swap /foo/bar and /foo/baz
```



CVE-2018-15664



```
docker cp ctr:/foo/bar/shadow .
docker cp myshadow ctr:/foo/bar/shadow
```

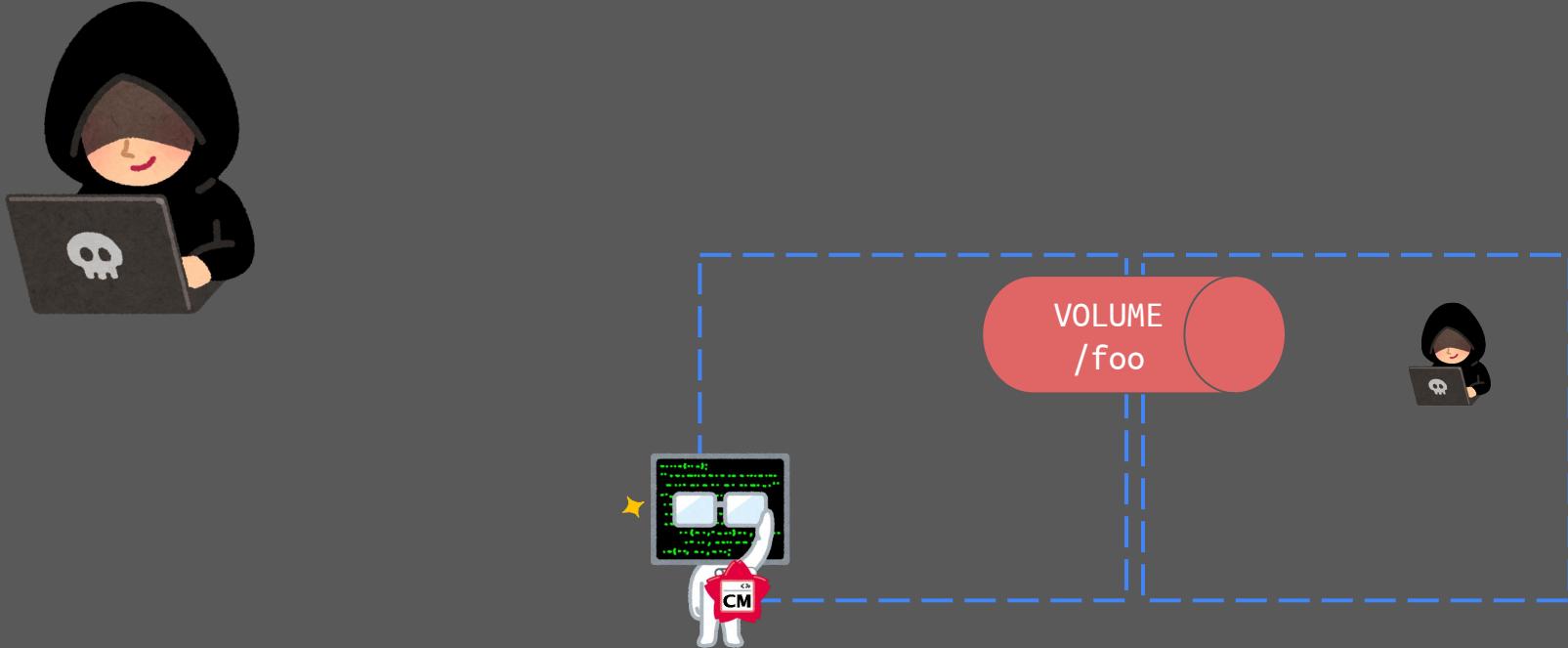


A large cargo ship, identified by the letters "MSC" on its hull, is engulfed in thick black smoke and flames. The ship is carrying numerous shipping containers stacked high. The background is a dark, hazy sky over the ocean.

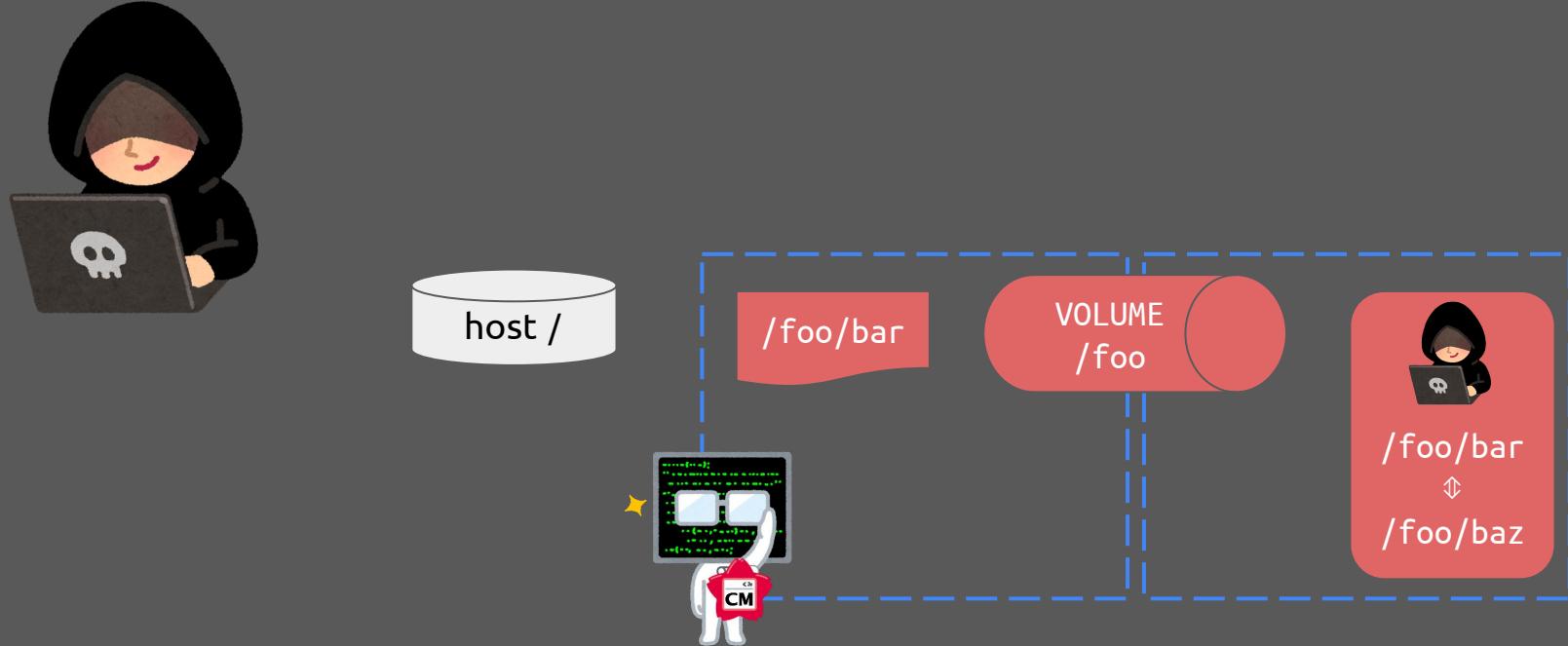
filesystems / / a bad time

pick two

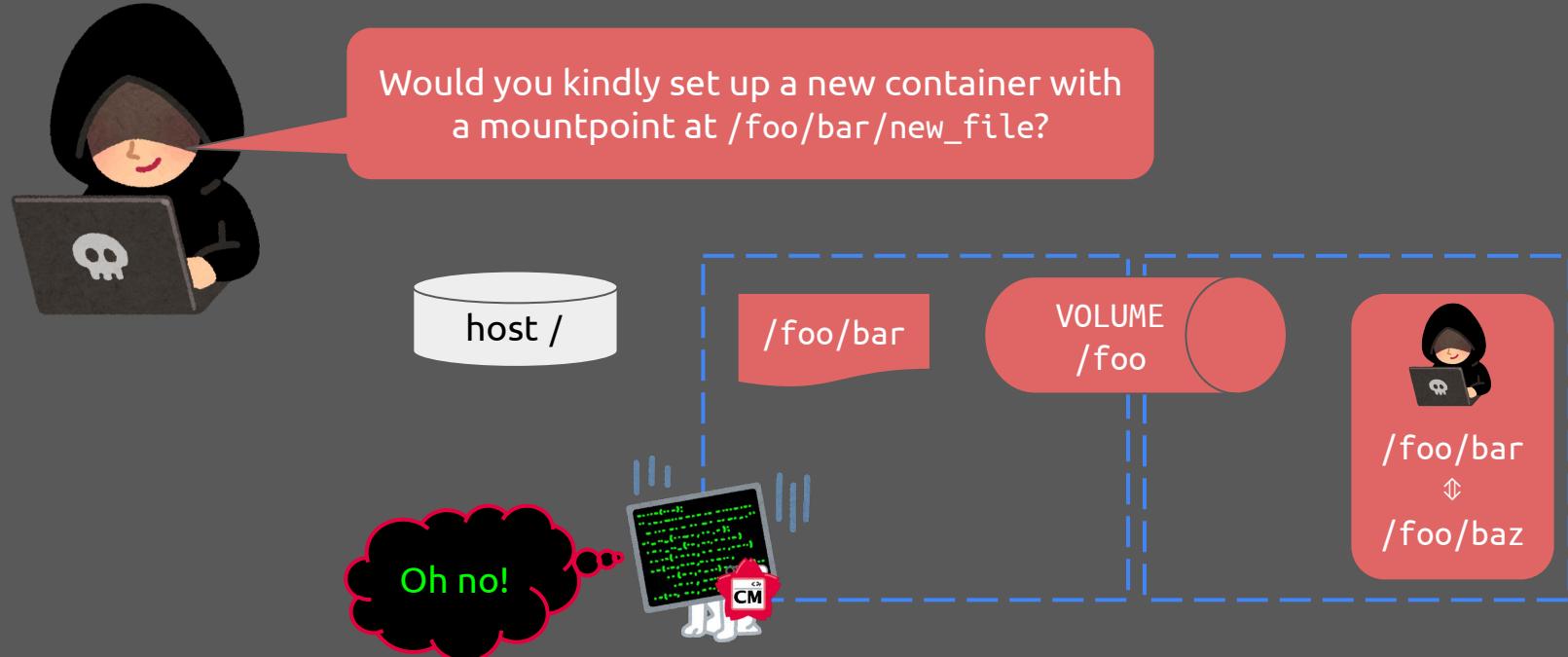
CVE-2024-45310



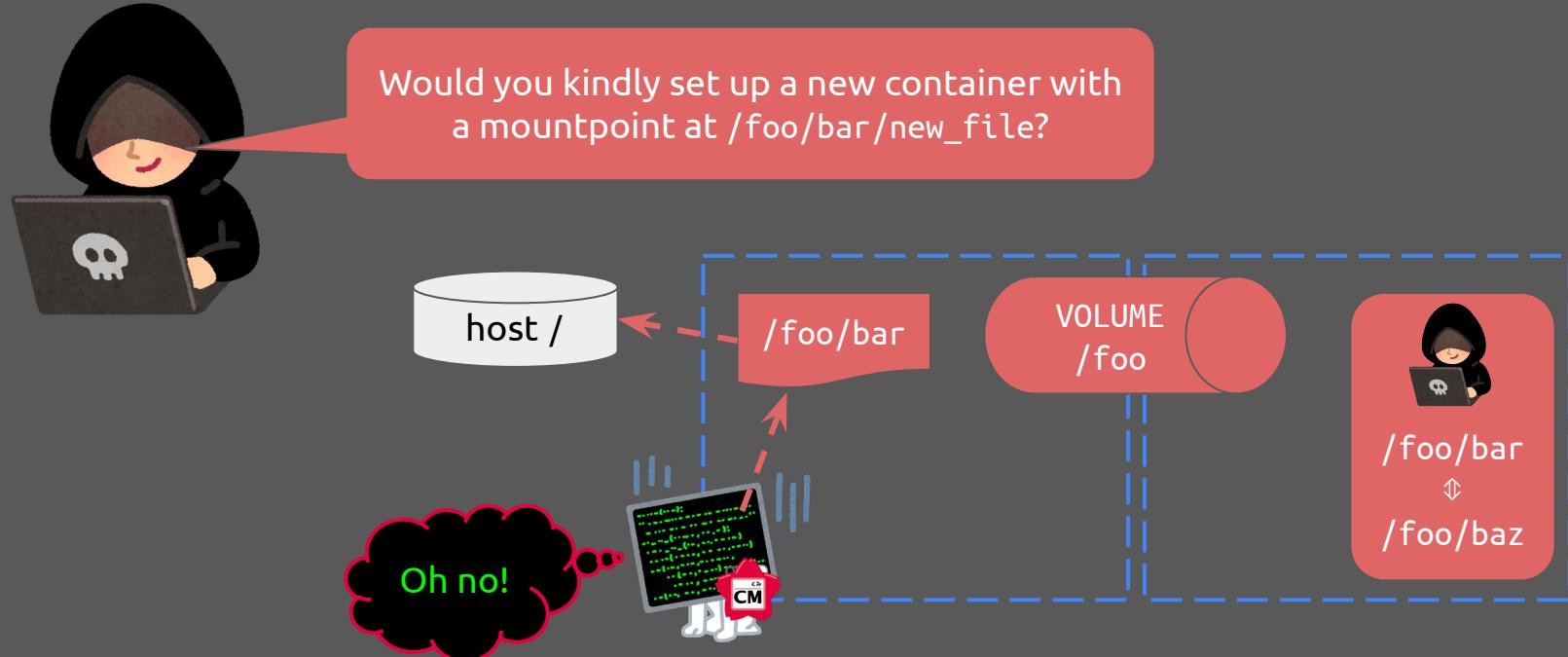
CVE-2024-45310



CVE-2024-45310



CVE-2024-45310

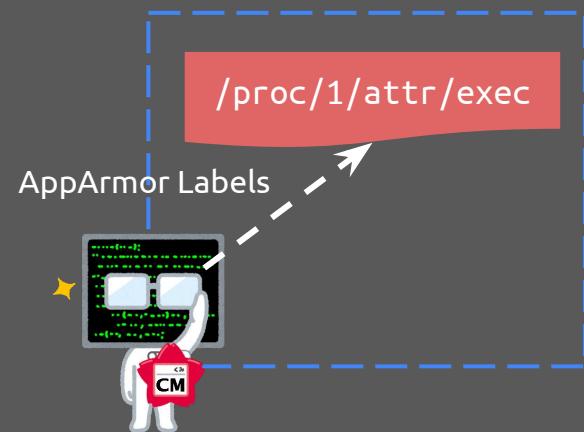


procfs

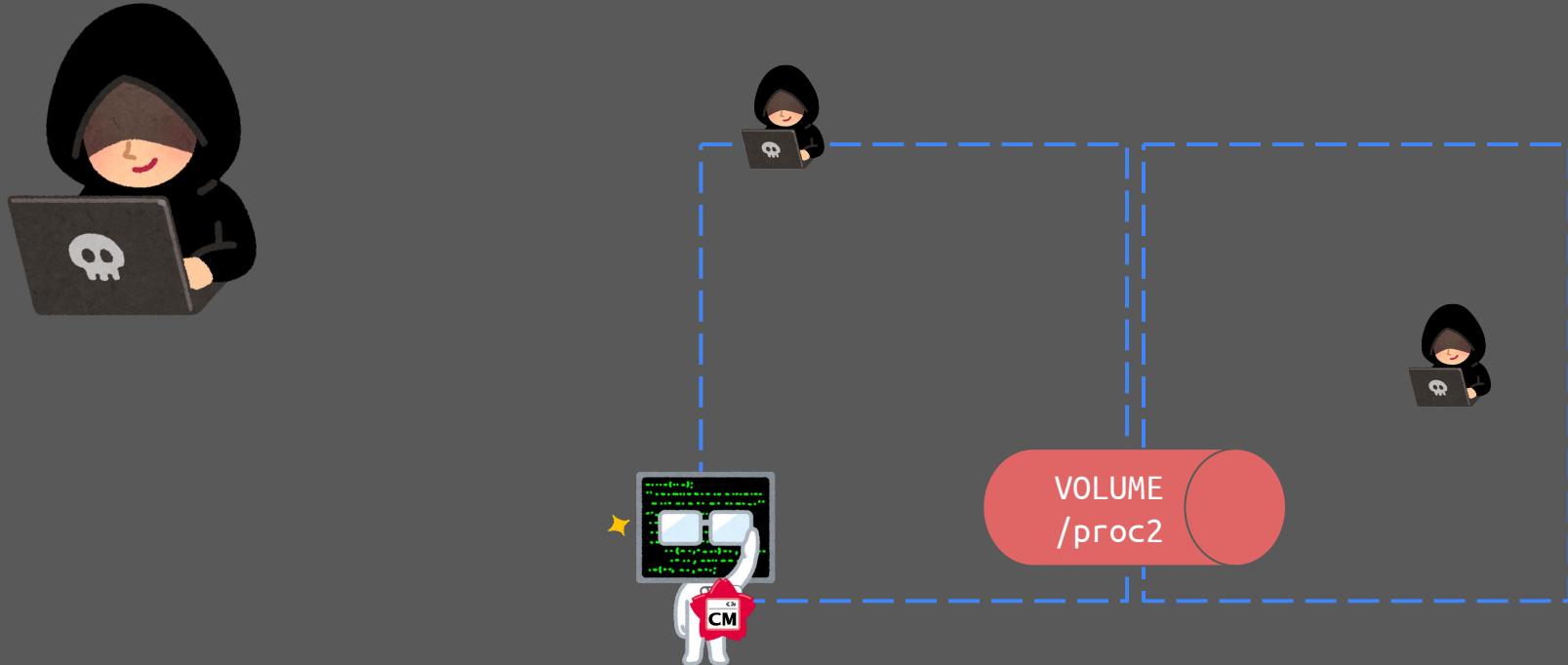
- Pseudofilesystem (more lies!)
- Only Linux API for some key features
- ✨ Magiclinks ✨

```
/proc
└── <pid>
    ├── attr
    │   ├── current
    │   └── exec
    ├── exe  -> ✨/bin/bash✨
    ├── fd
    │   ├── 0  -> ✨/dev/pts/1✨
    │   ├── 1  -> ✨/dev/pts/1✨
    │   ├── 2  -> ✨/dev/pts/1✨
    │   └── 3  -> ✨/secrets.txt✨
    └── mountinfo
└── cpuinfo
└── devices
```

CVE-2019-16884 / CVE-2019-19921



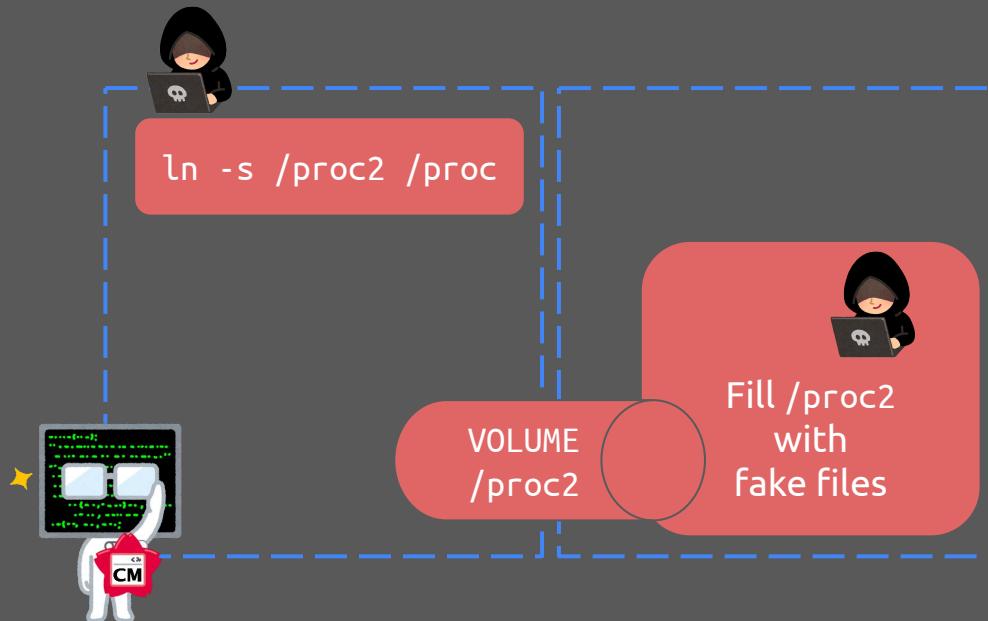
CVE-2019-16884 / CVE-2019-19921



CVE-2019-16884 / CVE-2019-19921



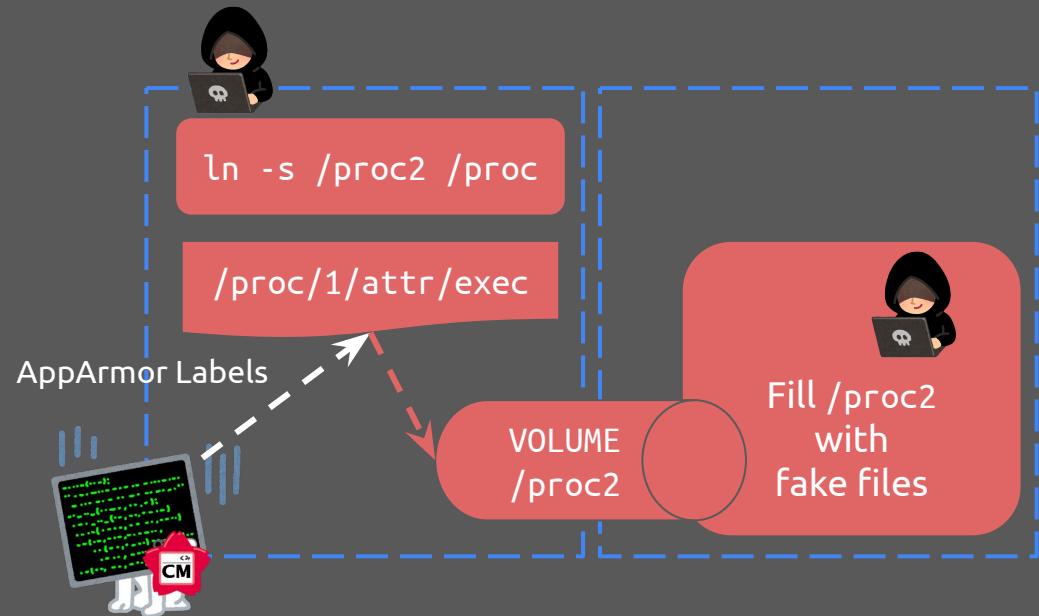
Would you kindly set up a new container with this weird /proc configuration?



CVE-2019-16884 / CVE-2019-19921 / CVE-2023-27561



Would you kindly set up a new container with this weird /proc configuration?





A new foe has appeared!

CHALLENGER APPROACHING



libpathrs



libpathrs



libpathrs



libpathrs





Let me sanitise
the path first.



```
/  
├── etc  
│   └── shadow  
├── tmp  
│   └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            ├── dummy-file.txt  
            └── files  
                └── music.mp3
```





Paths aren't safe! I'll help you!



```
/  
└── etc  
    └── shadow  
└── tmp  
    └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            ├── dummy-file.txt  
            └── files  
                └── music.mp3
```





Paths aren't safe! I'll help you!



```
/  
├── etc  
│   └── shadow  
├── tmp  
│   └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            ├── upload.php  
            ├── useful-data.txt -> /etc/shadow  
            ├── dummy-file.txt  
            └── files  
                └── music.mp3
```





```
/  
├── etc  
│   └── shadow  
├── tmp  
│   └── database.txt  
└── var  
    └── www  
        └── htdocs  
            ├── index.html  
            └── upload.php  
            └── useful-data.txt -> /etc/shadow  
            └── dummy-file.txt  
            └── files  
                └── music.mp3
```





libpathrs



openat2



libpathrs



O_PATH



KEEP
CALM
AND
USE
LIBPATHRS







XO

XXS

B

A

- LIFE -



IT'S DANGEROUS TO OPEN PATHS
ALONE! TAKE THIS.



That's all Folks!

CC BY-SA 4.0

