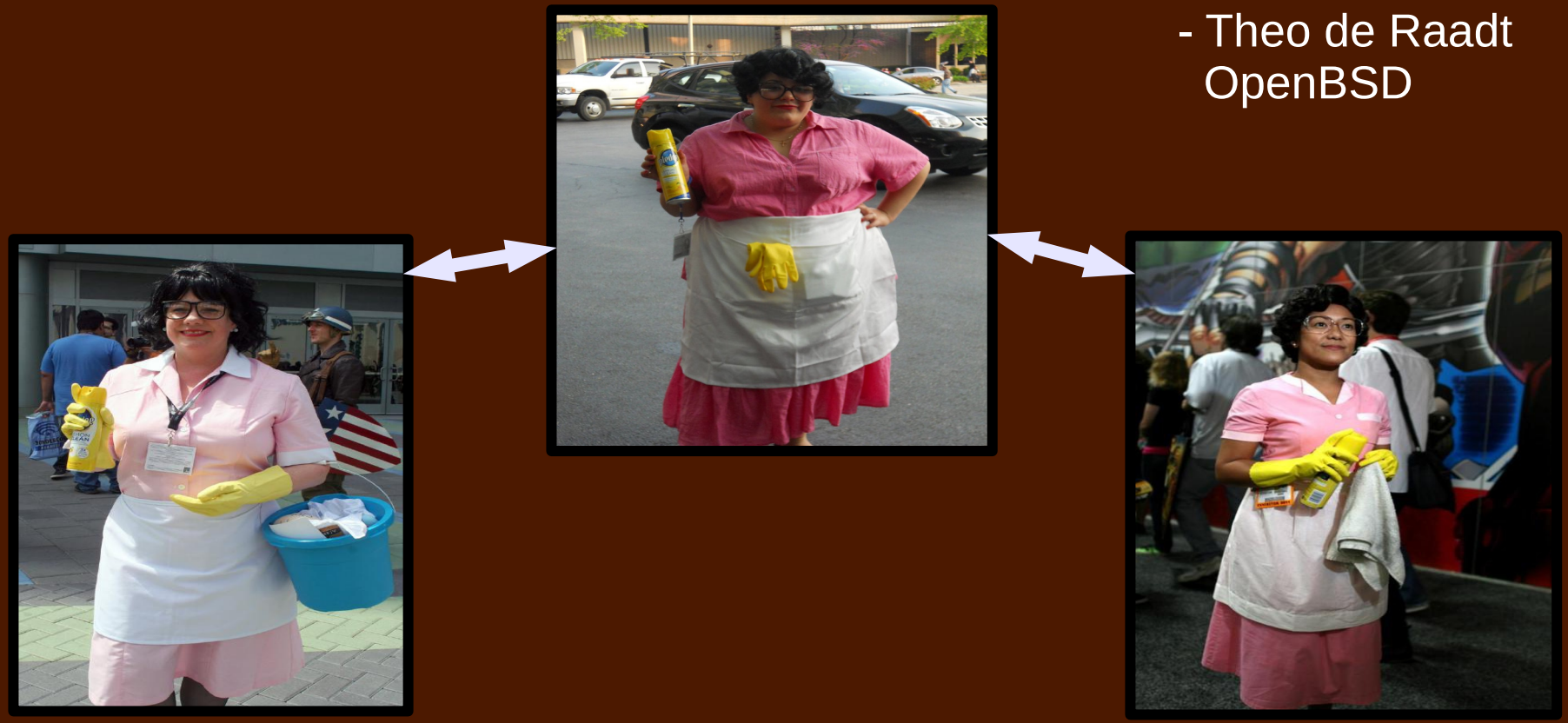
Privilege Separation and Pledge

- Theo de Raadt
  OpenBSD



OpenBSD Systems Interconnect

You and people on the Internet
(potential attackers)

Application design & architecture
(**Privilege Separation**, Privilege Drop, auditing, ...)

Address Space and resources
(Significant ASLR, W^X, various cookies)

libc routines
(POSIX, ANSI, defacto standards)

System call interface
(**pledge**)

Kernel
(Some ASLR, W^X, ...)

Hardware and BIOS
...

Focus on interaction between these two parts

Privilege Separation

A design pattern — splits a program into processes performing different sub-functions

Each process is designed to operate in a separate security domain

Processes cooperate over pipes using some protocol

Privsep – functional separation

(Our own ntpd as an example)

Master process

Runs as root, only does settimeofday()

DNS Servicer

Does DNS lookups

Internet Speaker

Speaks NTP to Internet

Defence in Depth

We designed & modified many programs to use the "privsep" design pattern

- Cooperative sub-processes connected by pipes, each operating in a different security domain — working together to get the job done

Experience gained with 60 programs

Let’s build a mechanism which enforces security domains!

Privsep – enforce with Pledge

(Our own ntpd as an example)

Master process

Pledge "settime"

DNS Servicer

Pledge "dns"

Internet Speaker

Pledge "inet"

Pledges are POSIX subsets

Pledge request permits only (carefully selected) subset of POSIX functionality

Subsets such as: stdio rpath wpath cpath fattr inet dns getpw proc exec …

Deep functional support in the kernel — much more than "seccomp" macros

Processes select own pledge

"I pledge this is the only subset of POSIX I will use"

Cannot undo your promise…

Process killed upon violation – good debugging experience

```
234    prog    CALL   socket(AF_LOCAL, 0x1<SOCK_STREAM,0)
234    prog    PLDG   socket, "inet", errno 1 Operation not permitted
234    prog    PSIG   SIGABRT SIG_DFL
234    prog    NAMI   "prog.core"
```

Privsep mistakes identified

Implementation errors found in 10% of privsep programs

Sub-processes did actions beyond design rule! tsk tsk.

Perfection is impossible to achieve without an enforcement mechanism keeping us honest...

Pledge helps us write better software.

Future work

OpenSSH privilege separation could be improved...

Continue refining semantics

Cooperate if another OS wants this