

# Privilege Separation and Pledge

- Theo de Raadt  
OpenBSD

Main maid



DNS maid



NTP protocol maid





# Many small changes to improve security

Application software (ports)  
(Educating upstream about better practices)

Own Applications: design & architecture  
(**Privilege Separation**, Privilege Drop, auditing, ...)

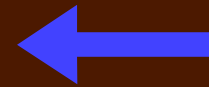
Address Space and other resources  
(ASLR, W<sup>X</sup>, cookies, ...)

Libraries (especially libc)  
(strcpy, arc4random, strict malloc, auditing, ...)

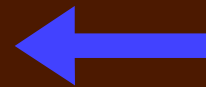
System call interface  
(**pledge**)

Kernel  
(Some ASLR, W<sup>X</sup>, ...)

Hardware and BIOS  
(cry into our beer...)



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

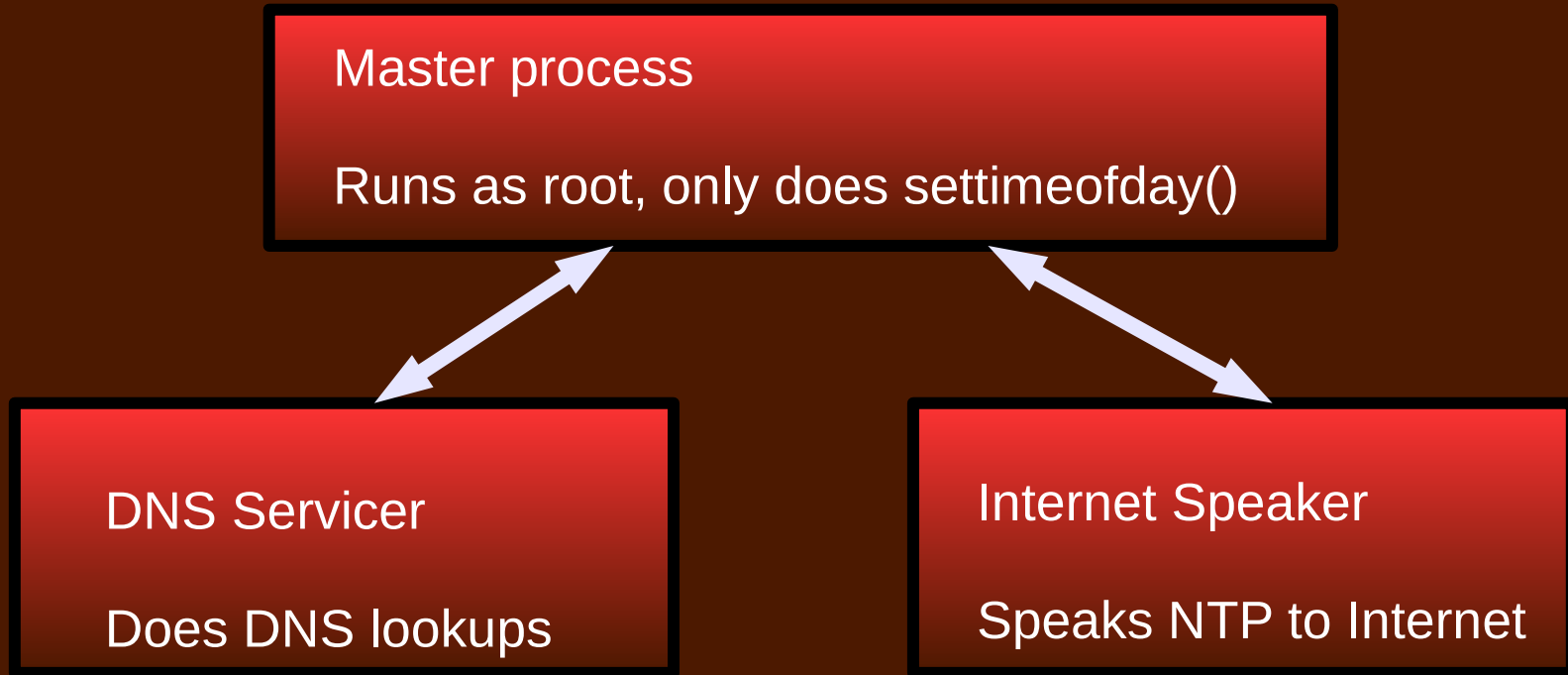
Processes cooperate over pipes using some protocol

Subset of “sandboxing” concept



# Separated at birth

(Our own ntpd as an example)





# Privilege Separation examples

The original 3:

Qmail

Postfix

OpenSSH

And.... Chrome



# Defence in Depth

We designed & modified many more programs to use this design pattern

Experience gained with 60 more programs!!!

Routing daemons, Mail daemons, dhcp tools, tcpdump...

Let's build a mechanism which enforces security domains!



# Major ones..

bgpd, dhclient, dhcpcd, dvmrpd, eigrpd, file, httpd, iked, ldapd, ldpd, mountd, npppd, ntpd, ospfd, ospf6d, pflogd, radiusd, relayd, ripd, script, smtpd, syslogd, tcpdump, tmux, xconsole, xdm, X server, ypldap, pkg\_add



# Pledges are POSIX subsets

Pledge syscall requests that only (a carefully selected) subset of POSIX functionality be permitted

Subsets such as: `stdio` `rpath` `wpath` `cpath` `fattr` `inet` `dns` `getpw` `proc` `exec` `sendfd` `recvfd` ...

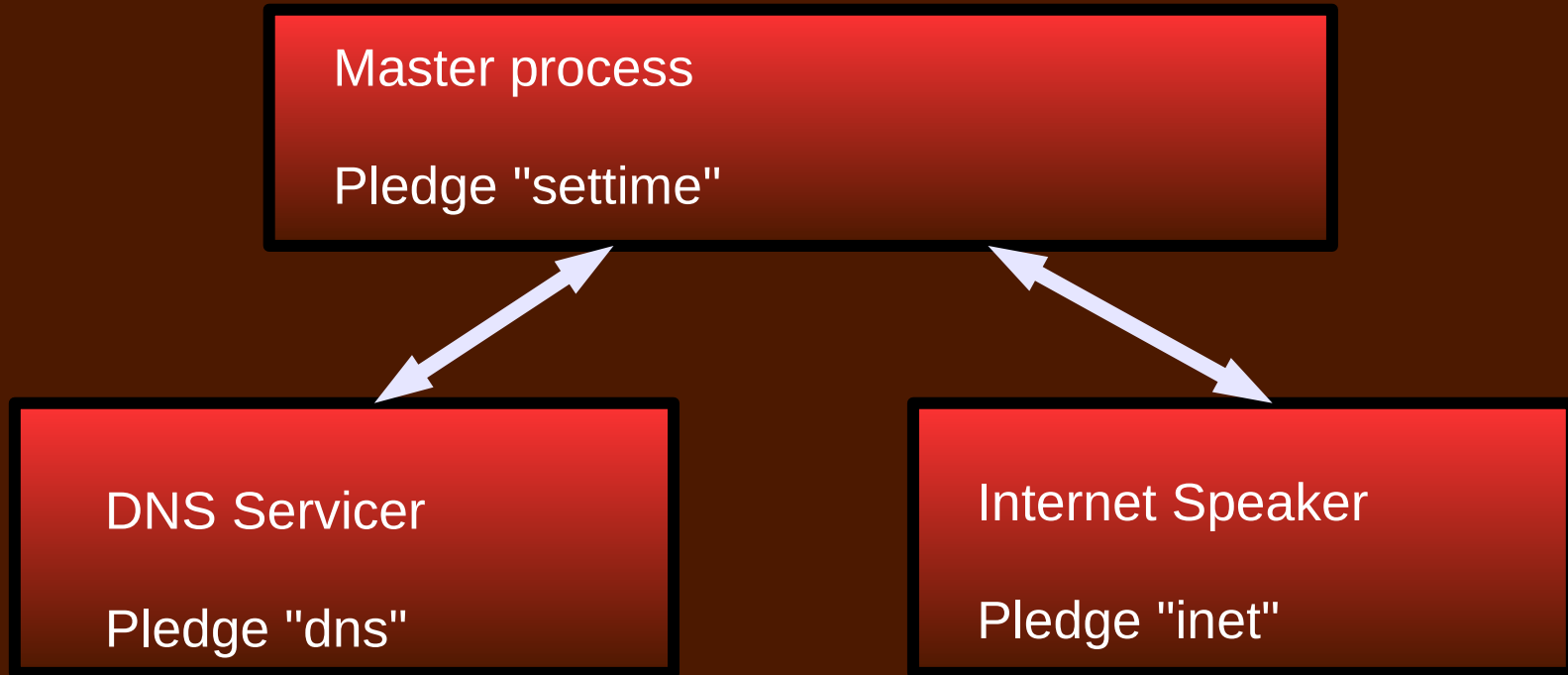
Deep functional support in the kernel — more sophisticated than "seccomp"





# Privsep – enforce with Pledge

(Our own ntpd as an example)





# Processes select own pledge – inline

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

Make the promise in the code when ready.

```
imsg_init(ibuf_dns, pipe_ntp[1]);  
  
if (pledge("stdio dns", NULL) == -1)  
    err(1, "pledge");  
  
while (quit_dns == 0) {
```

Cannot undo the promise...



# Good debugging experience

Most violations result in process being killed

```
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"
```

core is dumped — go ahead use gdb



# Privsep mistakes identified

Implementation errors found in 10% of privsep programs

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

ntpd, bgpd, tcpdump, ...

Validate program operation matches design rule



# Future work

OpenSSH privilege separation is dated, and could be improved...

Continue refining semantics

Cooperate if another OS wants pledge

Observe impact on upstream software, and assist



# General Observation

Perfection is impossible to achieve unless an enforcement mechanism keeps us honest