Recent work in OpenBSD relayd

AsiaBSDCon 2013

Reyk Flöter (reyk@openbsd.org)
ライクフローター

Agenda

- History & Background
- Recent work
  - SSL Interception
  - Socket Splicing
  - Filter rewrite

relayd

- buzzword bingo

Load Balancer	SSL Acceleration	Application Level Gateway (ALG)
Deep inspection	Link Balancer	IPv6 Gateway (NAT64/46)
Enterprise	Application Delivery Controller	SSL Interception


relayd
reyk@
Reyk Flöter
pyr@
Pierre-Yves
Ritschard

relayd

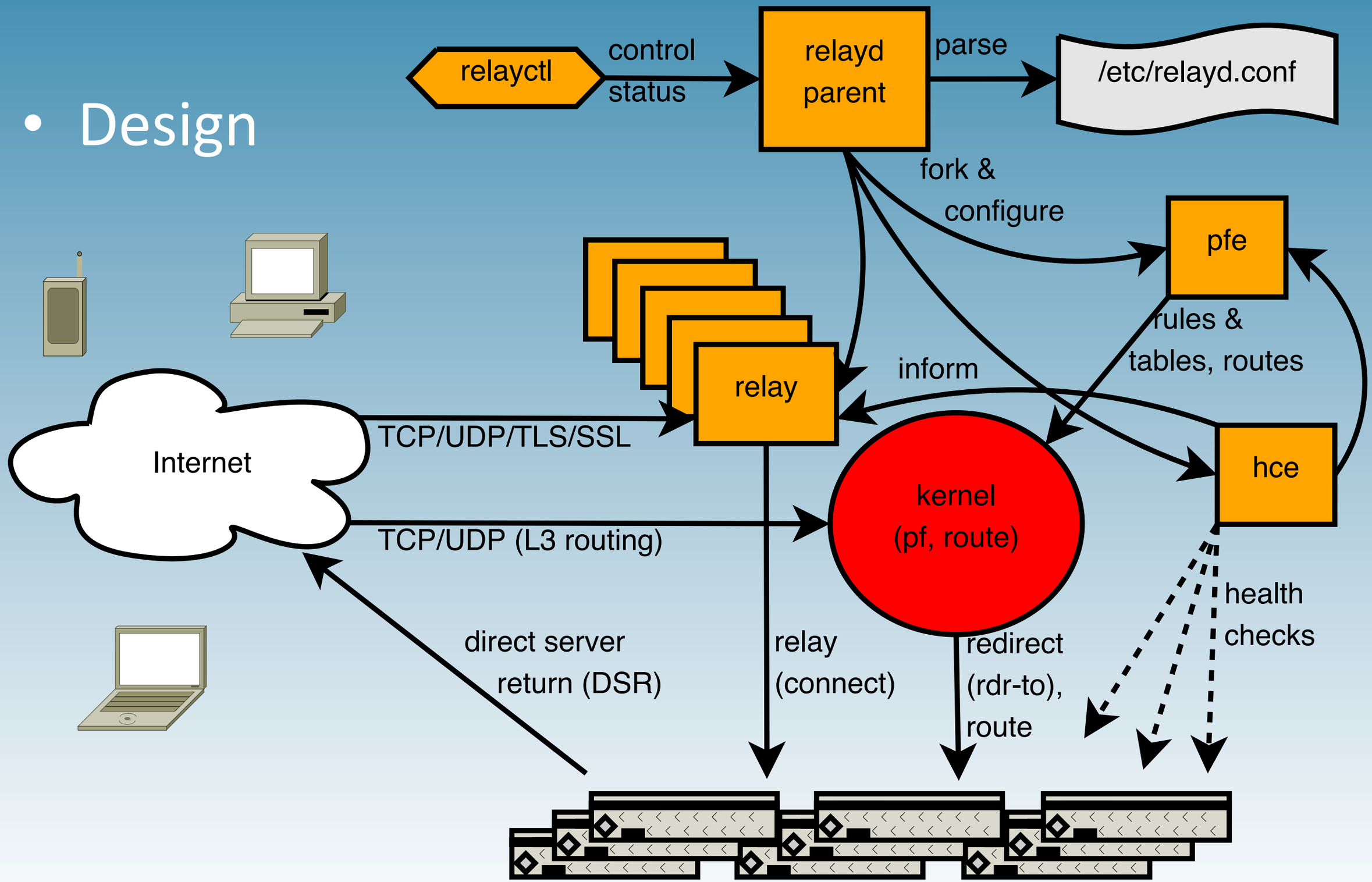
- History
  - 2006 two prototypes exist:
    slbd from pyr@, relayd from reyk@
  - 2006/12/16
    relayd first appeared as „hostated“ -
    a health-checking L3 server load balancer
  - 2007/01/09 renamed to „hoststated“
  - 2007/02/22 added L7 relay & SSL support
  - 2007/12/07 renamed to „relayd“

relayd

- Features
  - load balancer & application layer gateway
  - Protocols: TCP, SSL/TLS, HTTP, UDP (DNS)
  - Health checks: ICMP, TCP, SSL/TLS, HTTP, send/expect, external scripts
  - Configuration blocks or „subsystems“
    - redirect: L3 using PF rdr-to / route-to
    - relay: L7 from user space sockets
    - router: L3 routing table configuration

relayd

- Design



SSL Interception

SSL Interception I

- A „transparent proxy“ can attempt to intercept SSL connections:
  - Accept a redirected SSL connection …
  - … and connect to the original SSL server.

- Problem:
  - The client will reject the „broken“ certificate

SSL Interception II

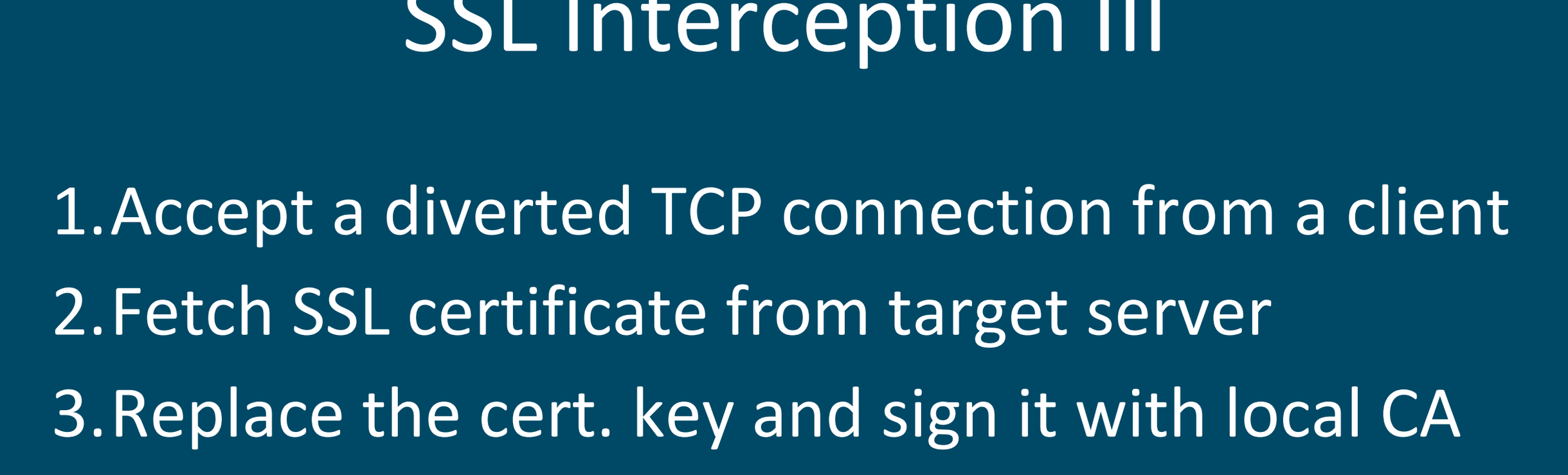
- SSL/TLS uses X.509 certificates to submit the public key and to validate a peers identity.
- A certificate is either self-signed or signed by a well-known „Certificate Authority“ (CA).
- HTTPS normally only checks and validates the server certificate (no mutual auth)

Idea:

- Generate a new server certificate „on the fly“ with a local trusted CA.

SSL Interception III

1. Accept a diverted TCP connection from a client
2. Fetch SSL certificate from target server
3. Replace the cert. key and sign it with local CA
4. Upgrade TCP connection to SSL server
5. SSL Relay & intercept



SSL Interception IV

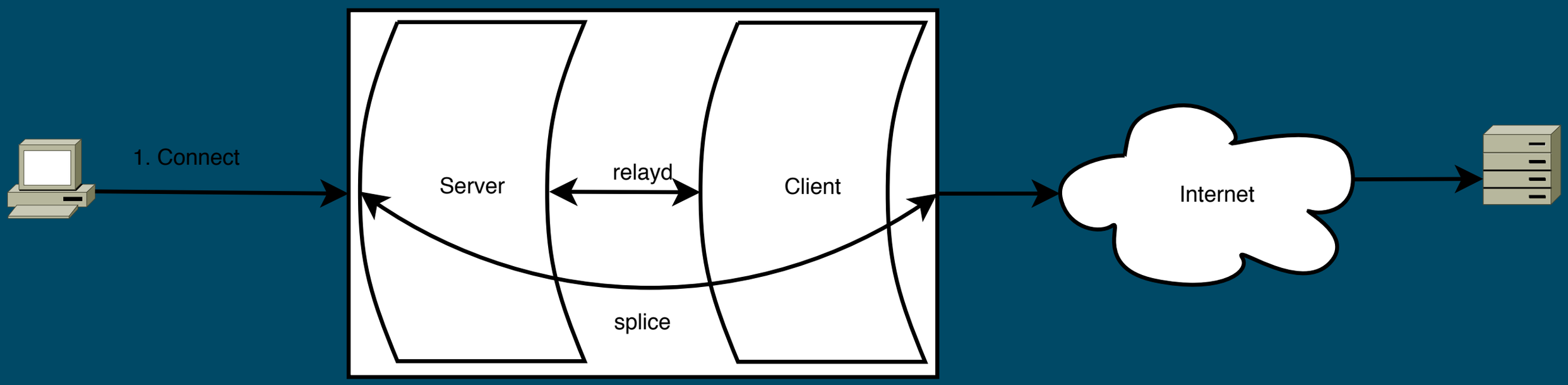
Configure SSL Interception:

```
http protocol httpfilter {
        return error
        label "Get back to work!"
        request url filter "facebook.com/"
        ssl ca key "/etc/ssl/private/ca.key" \
                password "humppa"
        ssl ca cert "/etc/ssl/ca.crt"
}
relay sslmitm {
        listen on 127.0.0.1 port 8443 ssl
        protocol httpfilter
        forward with ssl to destination
}
```

Socket Splicing

Socket Splicing I

- Objective:
  Increase the performance of TCP/HTTP relays.
- Connect two sockets in the kernel
- For example: handle HTTP body in user space, forward body in the kernel (splice sockets)



Socket Splicing II

- No configuration is required, relayd enables it by default
- You can turn it off with „no tcp splice“
- Can be used by other daemons, kernel Socket API:

```
bzero(&sp, sizeof(sp));
sp.sp_fd = fd2;
sp.sp_max = content_length;
sp.sp_idle = timeout;
if (setsockopt(fd1, SOL_SOCKET,
    SO_SPLICE, &sp, sizeof(sp)) == -1)
return (-1);
```

Filter Rewrite

- TODO

Filter Rewrite I

- Objective:
  Improve the flexibility of relayd‘s filtering
- PF is OpenBSD‘s in-kernel TCP/IP filter
  - Mostly L3-4 (IPv4, IPv6, TCP, UDP, ICMP, …)
  - No L7 inspection in the kernel (Hello, Linux)
- relayd extends PF as application layer gateway
  - Mostly L5-7
  - Privilege-separated L7 inspection

Filter Rewrite II

New rules in /etc/relayd.conf:

```
# Add X-Forwarded-For header (load balancer)
match request header append "X-Forwarded-For" value \
   "$REMOTE_ADDR"
# Simple URL Filter
block client in url "www.example.com/" tag "URL filtered!"
pass client in from 10.0.0.1 url "www.example.com/"
# Lists
match response tag "Instant messenger disallowed!"
block response header "Content-Type" value {
    "application/x-msn-messenger", "AIM/HTTP" }
# Alternate relay targets
match request path "/images" relay-to 10.1.1.1
match request path "/videos" relay-to <otherhosts>
```

Danke!

...thanks for supporting the OpenBSD project!