Introducing
OpenBSD's new httpd

AsiaBSDCon 2015
Reyk Flöter (reyk@openbsd.org) – ライクフローター
ESDENERA NETWORKS GmbH
Why do we need a web server in base?

- Serve the OpenBSD page.

OpenBSD 5.7

Released May 1, 2015
Copyright 1997-2015, Theo de Raadt.
5.7 Song: "XXX"

- Order a CDROM from our ordering system.
- See the information on the FTP page for a list of mirror machines.
- Go to the pub/OpenBSD/5.7/ directory on one of the mirror sites.
- Have a look at the 5.7 errata page for a list of bugs and workarounds.
- See a detailed log of changes between the 5.6 and 5.7 releases.

• signify(1) pubkeys for this release:
  
based: RWSvUXnw9gUb70PdeSNnpSmmodCyIPJEGN1wWr+6Time1eP7KiWJ5eAM
  tw: RWSuRBL44FKb20uvtw0UjmzS9UJtbKzd7GEYc0L8HxP04unj/Ct1LoZr
  pkg: RWTJ1iHLn/zcvJJSbxJIEU9ChlfAlU16XoLLxmxc1i:0FWfTLy0v0vQs

All applicable copyrights and credits can be found in the applicable file sources found in the files src.tar.gz, sys.tar.gz, xenocara.tar.gz, or in the files fetched via ports.tar.gz. The distribution files used to build packages from the ports.tar.gz file are not included on the CDROM because of lack of space.

What's New

This is a partial list of new features and systems included in OpenBSD 5.7. For a comprehensive list, see the changelog leading to 5.7.
Why do we need a web server in base?

• Serve our own kitten pages – securely.
Why do we need a web server in base?

- Many people use it for simple CGIs (e.g. bgplg).
OpenBSD's HISTORY OF WEBSERVERS
Webserver Timeline

March 1998
OpenBSD 2.3 includes Apache 1.3

September 2011
nginx imported for OpenBSD 5.1

March 2014
Apache removed from OpenBSD

August 2014, g2k14 Hackathon
OpenBSD’s new httpd shows up

November 2015
httpd in 5.6, nginx removed after 5.6.
We constantly improve our code base for better security & quality
    - Aiming for perfection.

#heartbleed, #shellshock, and many other issues happened in 2014
    - As one response to #heartbleed, OpenBSD forked LibreSSL

We also introduced new safer APIs like reallocarray()

I wrote a big diff for nginx to adopt reallocarray() other such techniques
    - And it got rejected.
    - Too big to maintain in OpenBSD, not suitable for upstream.
OpenBSD’s new HTTPD

”Today I woke up with sorrow and realized that I committed a web server last night” (reykfloeter@ on twitter)

- The situation of nginx in OpenBSD frustrated me.
  - nginx is not bad, it is some fine software, but it didn’t fit for us.
- At the g2k14 General OpenBSD Hackathon, I made an experiment:
  - I used relayd and turned it into a web server.
- At the same day, beck@ and deraadt@ tricked me into importing it.
- Two weeks later, we had httpd with TLS and FastCGI in 5.6.
httpd(8)

DESIGN & IMPLEMENTATION
Simplicity

• httpd is designed to be a simple and secure web server.
• Only the most important features will be supported:
  – Serve static files
  – Support FastCGI
  – Do (proper) TLS
  – Provide “core” features like directory listing, logging, basic auth.
• Current code is about 10,000 lines.
• Avoid “featuritis” in the future, track such feature requests:
  – https://github.com/reyk/httpd/issues?q=label%3Afeaturitis
Simplicity

• httpd is designed to be a simple and secure web server.
• Only the most important features will be supported:
  – Serve static files
  – Support FastCGI
  – Do (proper) TLS
  – Provide “core” features like directory listing, logging, basic auth.
• Current code is about 10,000 lines.
• Avoid “featuritis” in the future, track such feature requests:
  – https://github.com/reyk/httpd/issues?q=label%3Afeaturitis
# wc -l *

0  CVS
19 Makefile
589  config.c
334  control.c
253 http.h
102 httpd.8
1281 httpd.c
533 httpd.conf.5
688 httpd.h
242  log.c
312  logger.c
2062 parse.y
622  proc.c
1221 server.c
729 server_fcgi.c
469  server_file.c
1425 server_http.c
10881 total
Features

- **Static files**: Serves static files and directories via optional auto-indexing.
- **FastCGI**: Supports asynchronous and direct FastCGI.
- **Secure**: Non-optional security, chroot'ed and with privsep by default.
- **SSL/TLS**: Support secure connections via TLS powered by LibreSSL.
- **Virtual servers**: Flexible, name- and IP-based virtual servers.
- **Reconfiguration**: Reload the running configuration without interruption.
- **Logging**: Supports per-server logging via log files or via syslog.
- **Blocking**: Block, drop, and redirect connections.
Security

• Runs chroot’ed by default.
• Use privilege separation:
  – **parent**: Load the configuration, open servers sockets, load keys etc.
  – **server**: One or more processes to handle HTTP connections.
  – **logger**: Log to local files (or syslog), in our outside of the chroot.
• Don’t reinvent APIs, use libc whenever possible.
• Don’t pre-allocate large chunks of memory to use our safety belts.
• Don’t sacrifice security for performance.
TLS with LibreSSL

- “Safer TLS”
- Better API:
  - LibreSSL provides a new “libtls” API on top of libssl/libcrypto
  - Primarily written by Joel Sing (jsing@)
  - httpd was the reference implementation for the server API
- Use strong defaults:
  - In current, httpd only does TLS 1.2 by default.
  - Only strong ciphers and PFS.
FastCGI

- Florian Obser (fobser@) wrote slowcgi(8) to run CGIs with FastCGI
  - It was used to run bgplg(8) with nginx.
- He implemented the FastCGI server in httpd based on slowcgi.
  "I implemented slowcgi because you didn’t stop whining on icb that nginx can’t execute bgplg". And "fastcgi in httpd: (Bob) Beck has asked me if I can help you with it”.
- FastCGI is supported via UNIX or local TCP socket.
- Direct streaming, no buffering to a file.
httpd.conf(5)

CONFIGURATION
server "www.example.com" {
    listen on * port 80
}

Configuration

ext_ip="10.1.1.1"
server "www.example.com" {
    listen on $ext_ip port 80
}

types {
    include "/usr/share/mime.types"
}
server "www.example.com" {
    listen on * port 80
    listen on * tls port 443

    # Logging is enabled by default
    #no log

    location "/download/*" {
        directory auto index
        log style combined
    }
}

location "/pub/*" {
    block return 301 \
    "http://ftp.example.com/
    $REQUEST_URI"
}

location "*.php" {
    fastcgi socket \
    "/run/php-fpm.sock"
}

location "/cgi-bin/*" {
    fastcgi
    root "/"
}

root "/htdocs/www.example.com"
Conclusion

- httpd is almost finished
  - But it will take many more years to make it perfect
- We’re going to improve security
- And add a few more features,
  - eg. Server Name Indication (SNI)
  - Client certificates.
- More?
Thanks!

OpenBSD 5.7 will be released May 1st, 2015.

...and please keep supporting the OpenBSD project!
http://www.openbsdfoundation.org/campaign2015.html