Open/LibreSSL in FreeBSD

State of **LibreSSL** (and Open**SSL**)
In FreeBSD ports and base

Bernard (Barnerd) Spil
2016-06-11
BSDCan 2016

Bernard | Barnerd | Sp1l | brnrd

- FreeBSD user since 5.3 (ca. 2005)
- NB: Not a developer, not a cryptographer, …
- Active contributor on the #freebsd channel
- Maintainer of **LibreSSL** ports (and MariaDB)
- Author of collection of **LibreSSL** ports patches
- Day job: EAI Architect at **PHILIPS** Lighting
- Volunteer at **HSLnet** (local FttH cooperative) and for Bits of Freedom (Privacy Café & Toolbox)

How did we get here

- We all recall Heartbleed[1]?
- April 2014 OpenBSD forks OpenSSL[2]
- **LibreSSL** liveblogs the sourcecode flensing "OpenSSL Valhalla Rampage"[3]
- Support for old platforms is removed (Win16, OS/2, BeOS, VMS, etc.)
- Old, insecure features are removed (Export ciphers, compression, SSLv2, etc.)

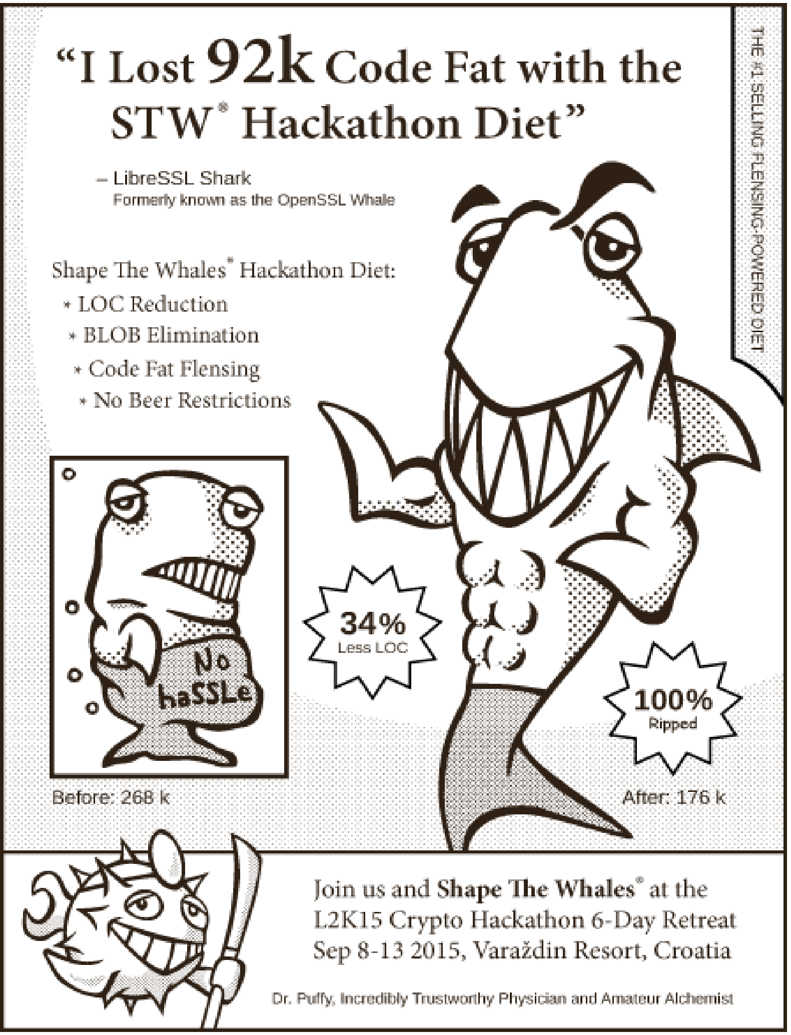
Recent SSL attacks

BEAST	Sep '11	CBC predictable IVs
CRIME	Sep '12	Compression before Encryption
RC4	Mar '13	Keystream biases
Lucky 13	May '13	MAC-Encode-Encrypt CBC
3Shake	Apr '14	Insecure resumption
POODLE	Dec '14	**SSLv3** MAC-Encode-Encrypt
SMACK	Jan '15	State machine attacks
FREAK	Mar '15	**Export**-grade 512-bit RSA
LOGJAM	May '15	**Export**-grade 512-bit DH
SLOTH	Jan '16	RSA-**MD5** signatures
DROWN	Mar '16	**SSLv2** RSA-PKCS#1 v1.5

Core Infrastructure Initiative[4]

- Formed by the Linux Foundation after Heartbleed was discovered (and after OpenBSD started LibreSSL)
- Commissions a security audit of OpenSSL by NCC Group
- Discovers numerous problems with the code
  - Fixed for the issues released by subsequent patch-releases of OpenSSL
  - Forcing frequent (emergency) patching for everyone

Where did LibreSSL end up?



“I Lost 92k Code Fat with the STW® Hackathon Diet”

– LibreSSL Shark
Formerly known as the OpenSSL Whale

Shape The Whales® Hackathon Diet:

- LOC Reduction
- BLOB Elimination
- Code Fat Flensing
- No Beer Restrictions

THE #1 SELLING FLENSING-POWERED DIET

Before: 268 k

After: 176 k

Join us and Shape The Whales® at the
L2K15 Crypto Hackathon 6-Day Retreat
Sep 8-13 2015, Varaždin Resort, Croatia

Dr. Puffy, Incredibly Trustworthy Physician and Amateur Alchemist

- New codebase ca 35% smaller (incl new libtls!)
- **LibreSSL**-portable first release 2.0.0 on 2015-07-11
- Further removal of features
- Addition of new libtls and netcat

PCBSD HardenedBSD OPNsense

VOID openelec embedded linux entertainment center

So what about FreeBSD ?

- Frequent updates to OpenSSL in base

FreeBSD-SA-14:03
FreeBSD-SA-14:06
FreeBSD-SA-14:09
FreeBSD-SA-14:10
FreeBSD-SA-14:14
FreeBSD-SA-14:18
FreeBSD-SA-14:23

FreeBSD-SA-15:01
FreeBSD-SA-15:06
FreeBSD-SA-15:12
FreeBSD-SA-15:26

FreeBSD-SA-16:11
FreeBSD-SA-16:17
FreeBSD-SA-16:??

- security/libressl ported within a day
- Currently 2.3.6 (and 2.4.1 for security/libressl-devel)

Vulnerabilities?

	LibreSSL	OpenSSL	LibreSSL	OpenSSL
	vs 1.0.1*		vs 1.0.2*	
Critical	0	0	0	0
High	0	4	0	2
Medium	14	25	12	17
Low	4	11	3	6
Total	**18**	**40**	**15**	**25**

NB: Yes, I know this is a stupid metric

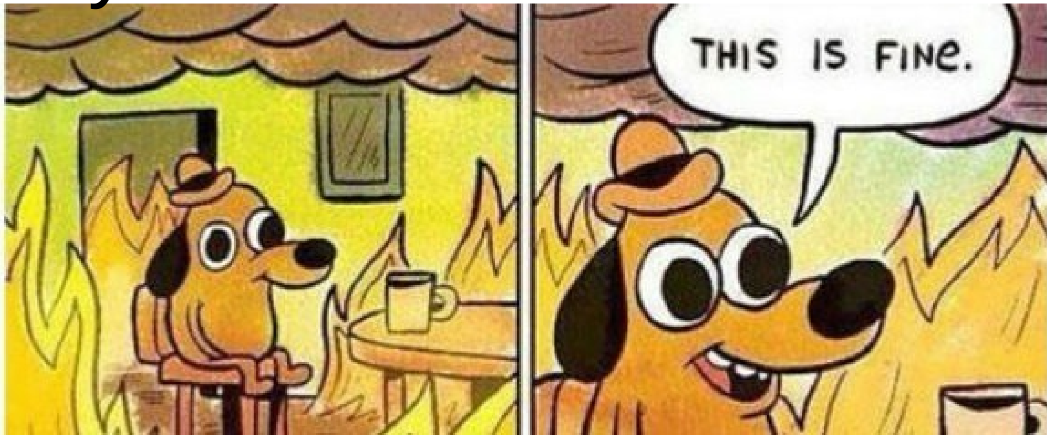
*Out-dated counts

That simple?!?

- With the first 2.0 release a lot of packages fail to build or run (ca 100 out of 25k)
  - Including major projects like Apache httpd, Python, OpenLDAP, cURL, …
- Then came 2.3 without SSLv3 and SHA-0
  - Again ca. 100 packages fail to build
  - Again including major projects like Apache httpd, Squid, haproxy, Python, Ruby, cURL

Bad examples

- Bad examples apparently proliferate
  I haven't tried to find the root of this but there are consistent troublesome ways to use the OpenSSL API
  - Makes patching easier…



- Please use the SSLv23 methods (or their TLS replacements) and SSL_OP_* flags
- Don’t check version-numbers for supported features... Features can and will be deprecated at some point!

Upstreaming

- The larger and more active projects are mostly very happy to include fixes.
- There are many abandoned, dormant, etc. projects out there! Patching all fall-out at times felt like trawling through a morgue...
- Still a large number of fixes to upstream
- Check the FreeBSD wiki[7,8]
- *Your help would be most welcome*

Additional OpenSSL issues

- Packages not honoring WITH_OPENSSL_PORT
  - Linking against base libssl/libcrypto instead
- Packages not specifying USE_OPENSSL
  - Yet linking against libssl/libcrypto
- Mix of base and ports OpenSSL causes issues (you ***must*** rebuild all ports when enabling WITH_OPENSSL_PORT)

Versions

FreeBSD version	OpenSSL version	Supported	Lifespan
9.x	0.9.8	EoL 2015-12-31	10.5 yrs
10.x	1.0.1	Security patches 2016-12-31	4.75 yrs
11	1.0.2	Full 2019-12-31	~ 5 yrs

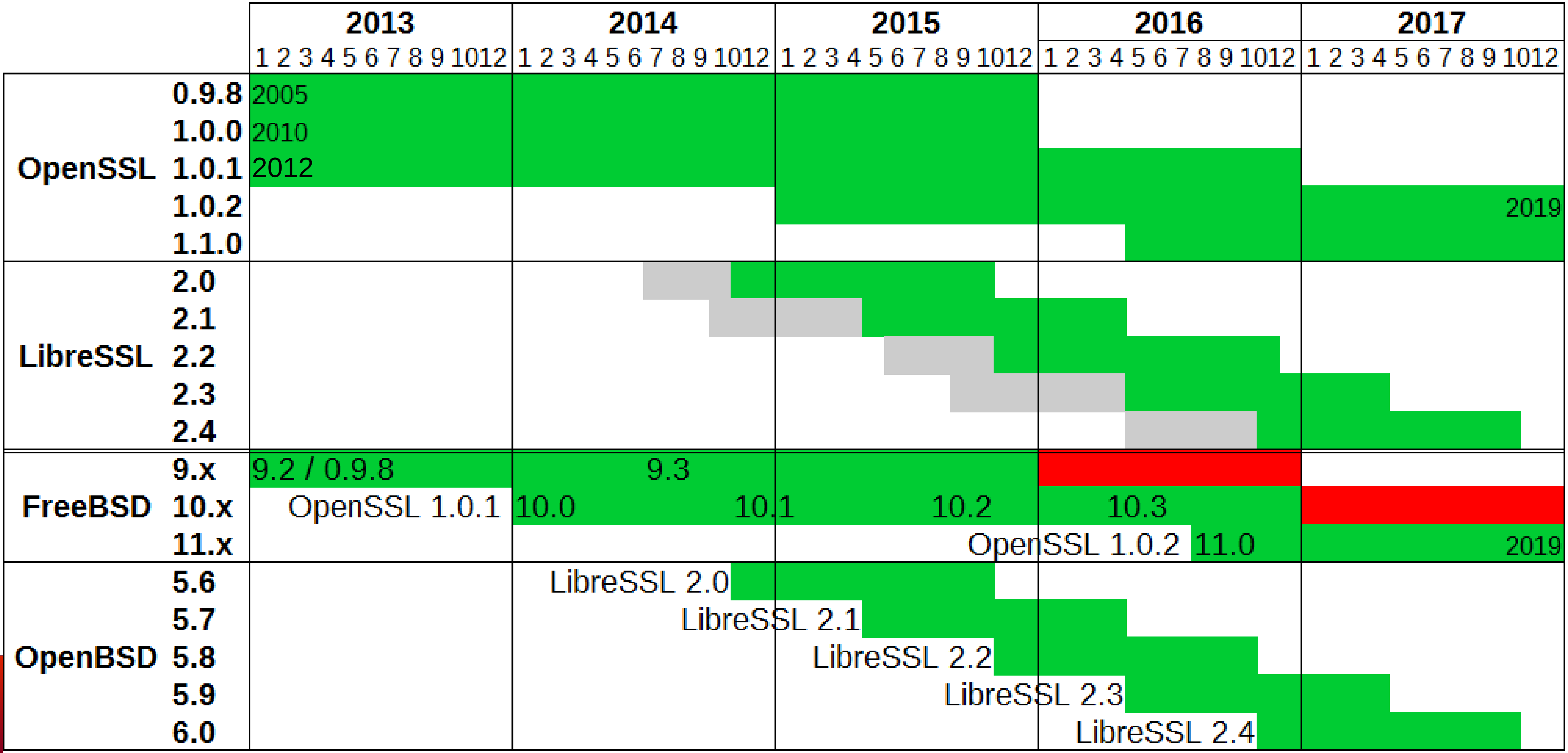
Most software is running with an outdated OpenSSL stack

OpenBSD version	LibreSSL version	Supported
5.7	2.1	2016-05-01
5.8	2.2	2016-11-01
5.9	2.3	2017-05-01
6.0	2.4	2017-11-01

Release every 6 months, supported 1 year

Lifecycle

- FreeBSD major versions have too long a lifespan to keep up with SSL versions



Building FreeBSD without OpenSSL libs

- Thanks to Adam McDougall
- WITHOUT_OPENSSL=yes in /etc/src.conf is not a complete solution
  - WITHOUT_LDNS, WITHOUT_LDNS_UTILS
  - WITHOUT_PKGBOOTSTRAP
  - WITHOUT_SVNLITE
  - Patch to disable ctld, iscsid, bsdinstall and ssl in libfetch (ouch!)
- Only really useful for a package building jail to force all packages to link to ports' OpenSSL

Making base SSL libs private

- FreeBSD base build framework can make libraries "private"
- 10.x: Moves these libraries to /usr/lib/private
- 11: Renames the library with libprivate prefix
- Ports can't “find” the private libs and will fail or link against the libraries provided by the port
- Why? Not all ports use the correct libraries (see https://bugs.freebsd.org/195796 for a list)

Result

- None of the files that originally linked against libssl or libcrypto still do
- E.g. /usr/bin/svnlite links to the private ssl and crypto.so
- All seems well
- Now that was simple…
- Not ***that*** simple, this leads to problems with `pkg libfetch`

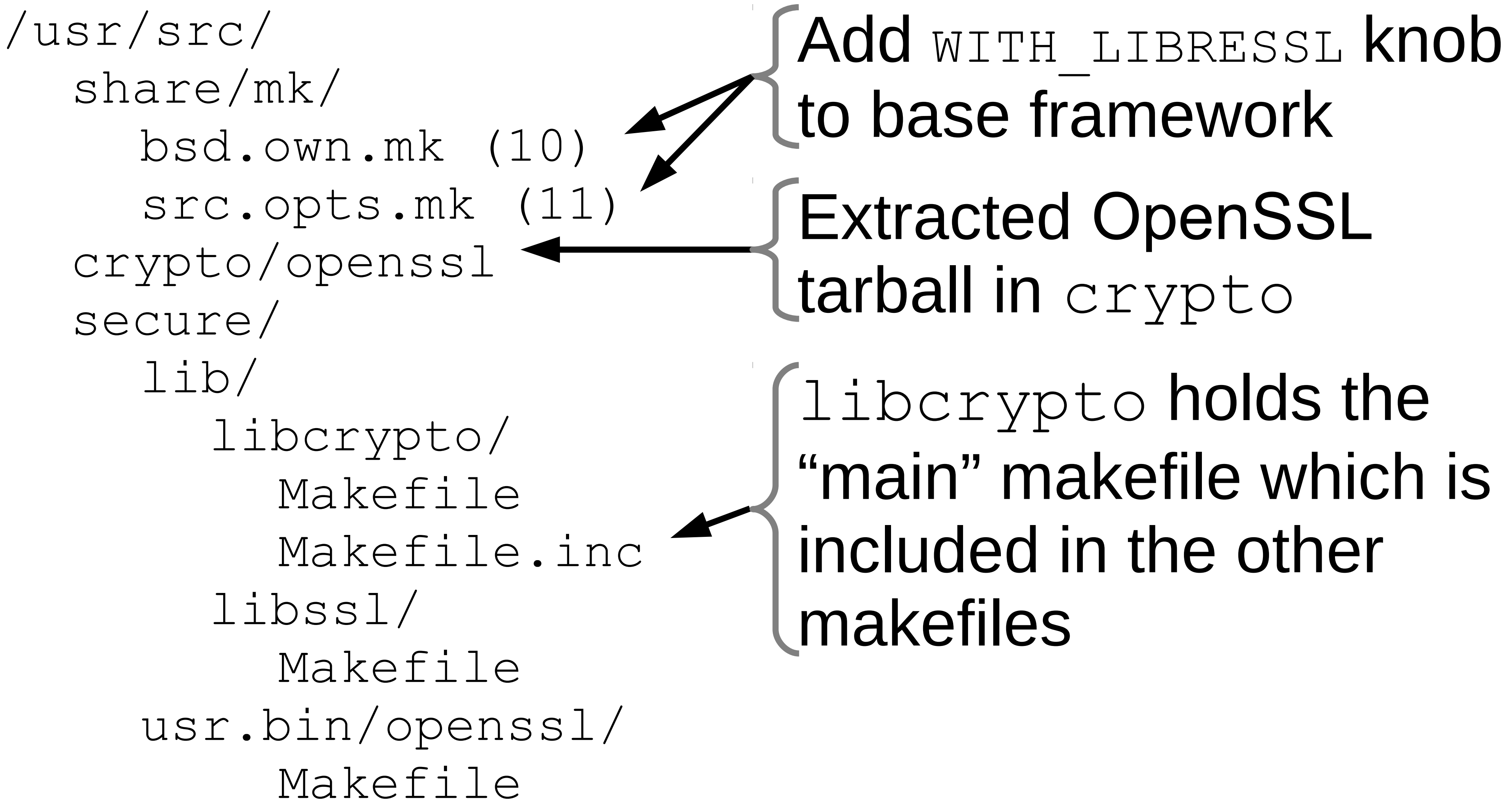
Replacing OpenSSL in base

- Tried this at the l2k15 (**LibreSSL** 2015) OpenBSD hackathon based on the existing makefiles, but failed…
- Back then Brent Cook advised me to just start with the OpenBSD makefiles but I wasn’t comfortable enough yet with make...

The challenge...

Integrate LibreSSL in HardenedBSD base
coexisting with OpenSSL
that allows switching between Open and Libre

/usr/src structure - OpenSSL



```
/usr/src/
  share/mk/
    bsd.own.mk (10)
    src.opts.mk (11)
  crypto/openssl
  secure/
    lib/
      libcrypto/
        Makefile
        Makefile.inc
      libssl/
        Makefile
    usr.bin/openssl/
        Makefile
```

Add `WITH_LIBRESSL` knob to base framework

Extracted OpenSSL tarball in `crypto`

`libcrypto` holds the “main” makefile which is included in the other makefiles

/usr/src structure - LibreSSL

```
crypto/libressl
secure/
   lib/
   libcrypto/
   Makefile.inc.libressl
   Makefile.libressl
   libssl/
   Makefile.libressl
   libtls/
   Makefile
   usr.bin/
   openssl/
   Makefile.libressl
   netcat/
   Makefile
```

Extract **LibreSSL** in `crypto`

“main” makefile

Add copies of base OpenBSD’s makefiles (not from -portable!)

Add `libtls` and `netcat`

WITH_LIBRESSL

```
__DEFAULT_NO_OPTIONS += LIBRESSL

/etc/src.conf → WITH_LIBRESSL=yes
```

FreeBSD 10.x

`bsd.own.mk`

`WITH_LIBRESSL`

transforms to

`MK_LIBRESSL`

HEAD / FreeBSD 11

`src.opts.mk`

`WITH_LIBRESSL`

transforms to

`MK_LIBRESSL`

add libtls to

`bsd.libnames.mk`

Modify existing Makefiles

```
# $FreeBSD$

.if ${MK_LIBRESSL} == "no"

Original Makefile

.else
.include "Makefile.libressl"
.endif
EOF
```

- HardenedBSD’s challenge: Allow easy switching between OpenSSL and LibreSSL
- Wrap the original Makefile in a conditional block
- Makes merging easy when OpenSSL is updated

Hi! Here’s 2001 again!

Fallout in base when building with LibreSSL:

- libtelnet and ppp use deprecated des_ methods
- Heimdal requires the Perl Entropy Gathering daemon
- And a bit of the future: wpa in HEAD uses checks OPENSSL_VERSION_NUMBER

base vs ports

- The LibreSSL ports patch OPENSSL_VERSION_NUMBER from 0x20000000L to 0x1000107fL (1.0.1f) to work around projects determining features by the version number.
  - LibreSSL added LIBRESSL_VERSION_NUMBER in version 2.3
  - Fallout in ports relatively low (work in progress)

e.g. contrib/wpa/src/crypto/tls_openssl.c

```
-#if OPENSSL_VERSION_NUMBER >= 0x10100000L
+#if OPENSSL_VERSION_NUMBER >= 0x10100000L && !defined(LIBRESSL_VERSION_NUMBER)
```

What's to come

- Finalizing and polishing LibreSSL in base
- Committing/upstreaming the LibreSSL patches for ports
- HardenedBSD and PCBSD with LibreSSL as default libcrypto provider
- `Mk/bsd.openssl.mk` to `Mk/Uses/openssl.mk` (mat@)
- Default to OpenSSL from ports?
- `WITH_LIBRESSL` in FreeBSD base???

Who benefits?

- **LibreSSL** paved the way for OpenSSL 1.1.0
  - SSLv3 methods removed in default build config
  - EGD removed from default build config
  - `des_old.h` has been removed after 15 years
- See `security/openssl-devel` port, disables all questionable features by default.

YOU!

Thanks

- OpenBSD (Bob, Joel, Theo, Brent, ...)
- Kris Moore from PCBSD for providing the build resources to repeatedly rebuild 10k ports
- The HardenedBSD team for their trust and patience
- 'frogs' from IRC for pushing me to get it done
- Allan Jude for the original work on Making SSL libs private in base.
- Vsevolod, Kubilay, Johannes and many more from the FreeBSD project for their invaluable help and guidance.

References/links

1)http://heartbleed.com/

2)http://www.tedunangst.com/flak/post/origins-of-libressl

3)http://opensslrampage.org/

4)https://www.coreinfrastructure.org/

5)https://wiki.freebsd.org/OpenSSL/Base

6)https://wiki.freebsd.org/LibreSSL

7)https://wiki.freebsd.org/OpenSSL/No-SSLv3

8)https://wiki.freebsd.org/LibreSSL/Ports

Exhibit 1: The Perl Entropy Gathering Daemon

- Back in the day, there was no /dev/random
- No current platform needs it (as of ca. 2004) yet projects are rife with RAND_egd

```
else if (pRandSeed->nSrc == SSL_RSSRC_EGD) {
   /*
    * seed in contents provided by the external
    * Entropy Gathering Daemon (EGD)
    */
   if ((n = RAND_egd(pRandSeed->cpPath)) == -1)
      continue;
      nDone += n;
}
```

(Apache 2.4.8)

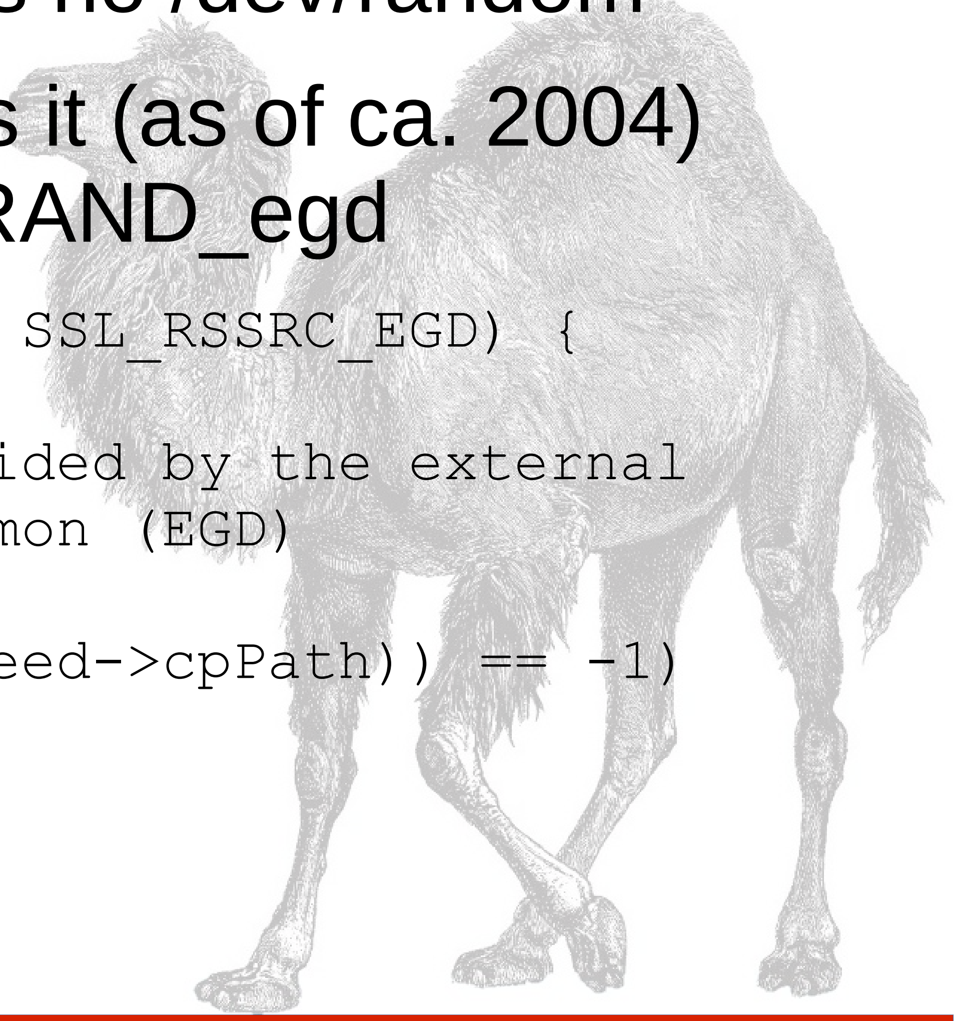


Exhibit 2: deprecation Promises, promises...

- **2001**-10-24: “the OpenSSL DES functions are renamed to begin with DES_ instead of des_. Compatibility routines are provided and declared by including openssl/des_old.h.
  The compatibility functions will be removed in some future release, **at the latest in version 1.0.”**

```
 static void
-des_ecb_encrypt( des_data_block *plain, des_data_block *encrypted,
-         des_context ctxt, int op)
+DES_ecb_encrypt( DES_data_block *plain, DES_data_block *encrypted,
+         DES_context ctxt, int op)
 {

- des_ecb_encrypt( &StdText, &PasswordHash2, schedule , DES_ENCRYPT );
+ DES_ecb_encrypt( &StdText, &PasswordHash2, &schedule , DES_ENCRYPT );
```

(OpenLDAP 2.4)

How broken is OpenSSL?

Last Thursday it was reported to the openssl-dev mailing list by Ben Kaduk that there was a defect in this optional code: it had a syntax error and didn't even compile. It had a typo of "!!" instead of "||":

```
     if (DES_set_key_checked(&deskey[0], &data(ctx)->ks1)
         !! DES_set_key_checked(&deskey[1], &data(ctx)->ks2))
```

...

This syntax error was present in the _original_ commit: the code in the #ifdefs had _never_ been compiled.

...
...

This code was commited in 2004.

...
...
(stop screaming and catch your breath)

Uhhh... pardon?

```
$ whois libressh.org
Domain Name: LIBRESSH.ORG
Domain ID: D172501991-LROR
Registrant ID: SM8731-GANDI
Registrant Name: Steve Marquess
Registrant Organization: OpenSSL Software Foundation, Inc.
Registrant Street: 1829 Mount Ephraim
Registrant City: Adamstown
```