RSAConference2016

San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

SESSION ID: PDAC-T10R

OpenSSL after HeartBleed

Tim Hudson

Cryptsoft ,
OpenSSL Team

Rich Salz

Akamai Technologies,
OpenSSL Team

#RSAC

The most important date

- April 3, 2014

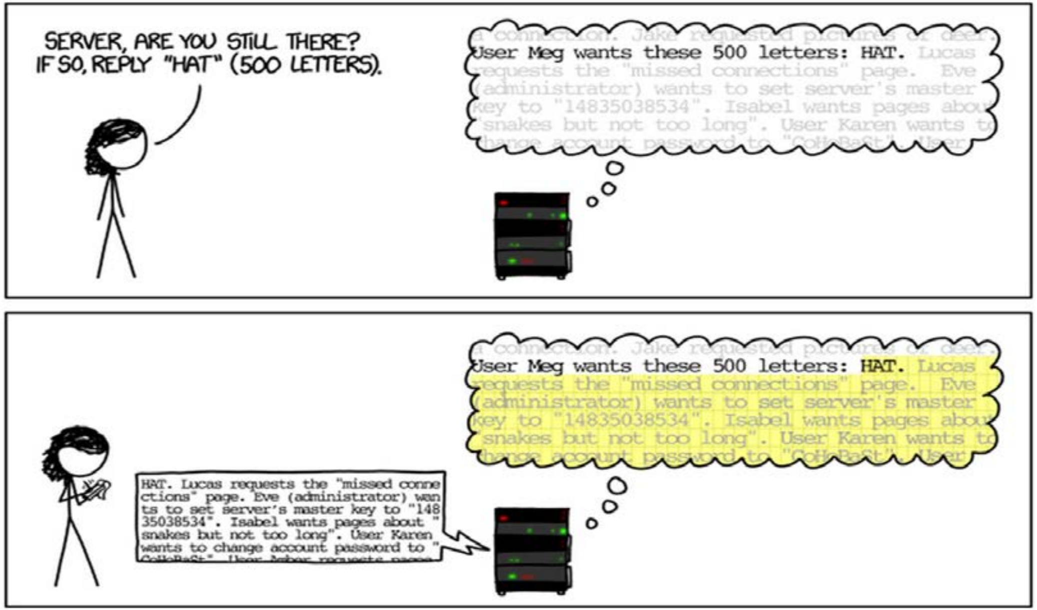
The most important date

- April 3, 2014

- HeartBleed
- Re-key the Internet

So what was HeartBleed?

- A very simple bug, the code didn't check a buffer length.


SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
a connection. Jake requested pictures of deer. User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoHoBaSt". User
HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoHoBaSt". User Amber requests pages

Source: http://xkcd.com/1354/ courtesy Randall Munroe

So what was HeartBleed?

- Massive mainstream press coverage

April 2014
The Heartbleed Bug attacking over 60% of websites Today.
The bug has the potential to affect the security of all your online accounts and in fact it has had 2 yrs to gather your personal information.
Solutions: Change all your passwords today.
https://www.
THE DEEPER GRAPHIC WEB DESIGN

DAILY Mirror

By George he looks like dad...

WORLD EXCLUSIVE

Why I sit in that court every day

BY REEVA'S MUM

CHANGE ALL YOUR PASSWORDS

Warning as millions hit by Heartbleed computer bug targeting online details

So what was HeartBleed?

To the best of our knowledge, Heartbleed is the first computer systems bug to have both its own website and its own logo, the cute bleeding heart. As such, Heartbleed sets a precedent that will have both positive and negative ramifications for future vulnerabilities and malware.

…

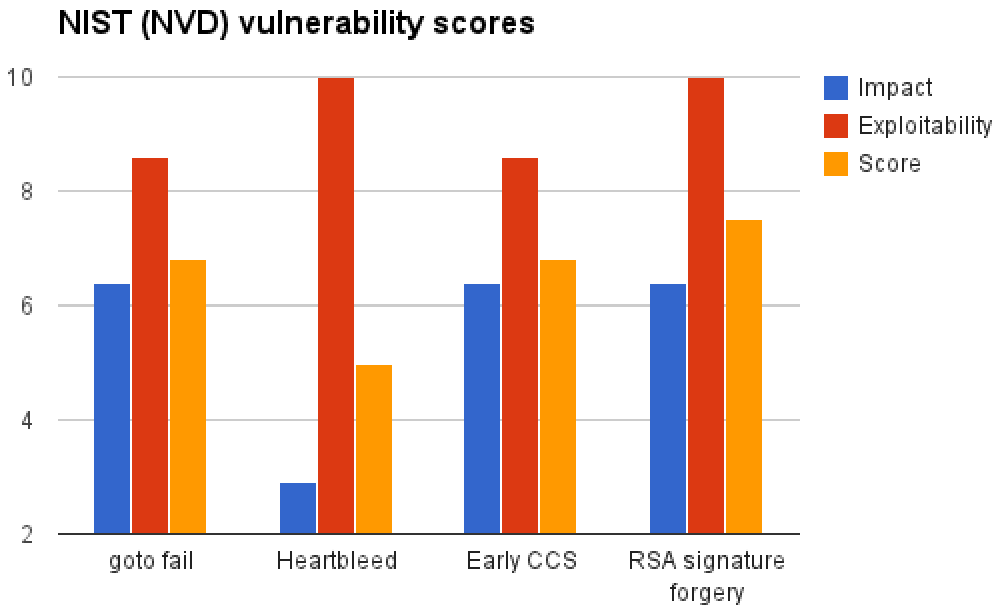
Even among the vast majority of the population who have no idea what OpenSSL is, people everywhere quickly found out that a major bug could compromise their Internet security.

Source: VDC Research - http://blog.vdcresearch.com/embedded_sw/2014/04/exploiting-the-exploit-the-marketing-of-heartbleed.html

The sky is falling …

- CVE-2011-0014 - infoleak, true impact unknown
- CVE-2012-2110 - possibly arbitrary code execution on reading certificates
- CVE-2012-2333 - buffer over-read, true impact unknown
- CVE-2014-1266 - "goto fail" server spoofing (Apple)
- CVE-2014-0160 – Heartbleed
- CVE-2014-0224 - "early CCS" disables encryption
- CVE-2014-1568 - RSA signature forgery (NSS)

Or is it …


NIST (NVD) vulnerability scores
10
8
6
4
2
goto fail
Heartbleed
Early CCS
RSA signature forgery
🟦 Impact
🟥 Exploitability
🟧 Score

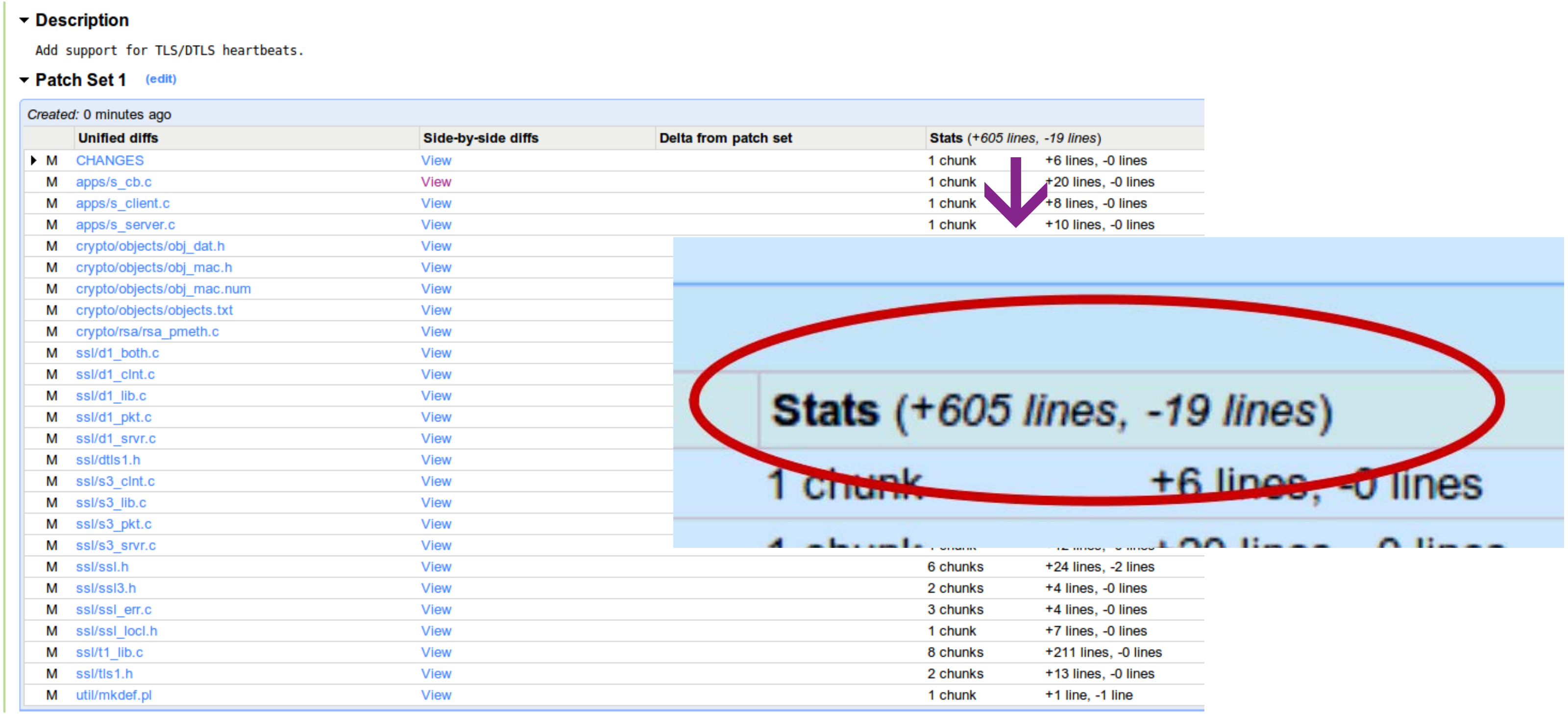
So what was HeartBleed?

- Basically missed validating a variable containing a length
- Contributed code had a bug – bug was in code base for **three years**!
- Project team member review missed the bug
- Other team members either didn’t review or also simply missed the bug
- Multiple external security reviewers and auditors missed the bug
- OpenSSL external developers and users missed the bug
- Security review teams in major OpenSSL using organisations missed the bug
- **All** existing code analysis tools missed the bug
- Bug allowed clients to attack servers **and** servers to attack clients

OpenSSL
Cryptography and SSL/TLS Toolkit

So what was HeartBleed?

Add support for TLS/DTLS heartbeats.



Description

```
Add support for TLS/DTLS heartbeats.
```

Patch Set 1 (edit)

Created: 0 minutes ago

	Unified diffs	Side-by-side diffs	Delta from patch set	Stats (+605 lines, -19 lines)	
M	CHANGES	View		1 chunk	+6 lines, -0 lines
M	apps/s_cb.c	View		1 chunk	+20 lines, -0 lines
M	apps/s_client.c	View		1 chunk	+8 lines, -0 lines
M	apps/s_server.c	View		1 chunk	+10 lines, -0 lines
M	crypto/objects/obj_dat.h	View			
M	crypto/objects/obj_mac.h	View			
M	crypto/objects/obj_mac.num	View			
M	crypto/objects/objects.txt	View			
M	crypto/rsa/rsa_pmeth.c	View			
M	ssl/d1_both.c	View			
M	ssl/d1_clnt.c	View			
M	ssl/d1_lib.c	View			
M	ssl/d1_pkt.c	View			
M	ssl/d1_srvr.c	View			
M	ssl/dtls1.h	View			
M	ssl/s3_clnt.c	View			
M	ssl/s3_lib.c	View			
M	ssl/s3_pkt.c	View			
M	ssl/s3_srvr.c	View			
M	ssl/ssl.h	View		6 chunks	+24 lines, -2 lines
M	ssl/ssl3.h	View		2 chunks	+4 lines, -0 lines
M	ssl/ssl_err.c	View		3 chunks	+4 lines, -0 lines
M	ssl/ssl_locl.h	View		1 chunk	+7 lines, -0 lines
M	ssl/t1_lib.c	View		8 chunks	+211 lines, -0 lines
M	ssl/tls1.h	View		2 chunks	+13 lines, -0 lines
M	util/mkdef.pl	View		1 chunk	+1 line, -1 line

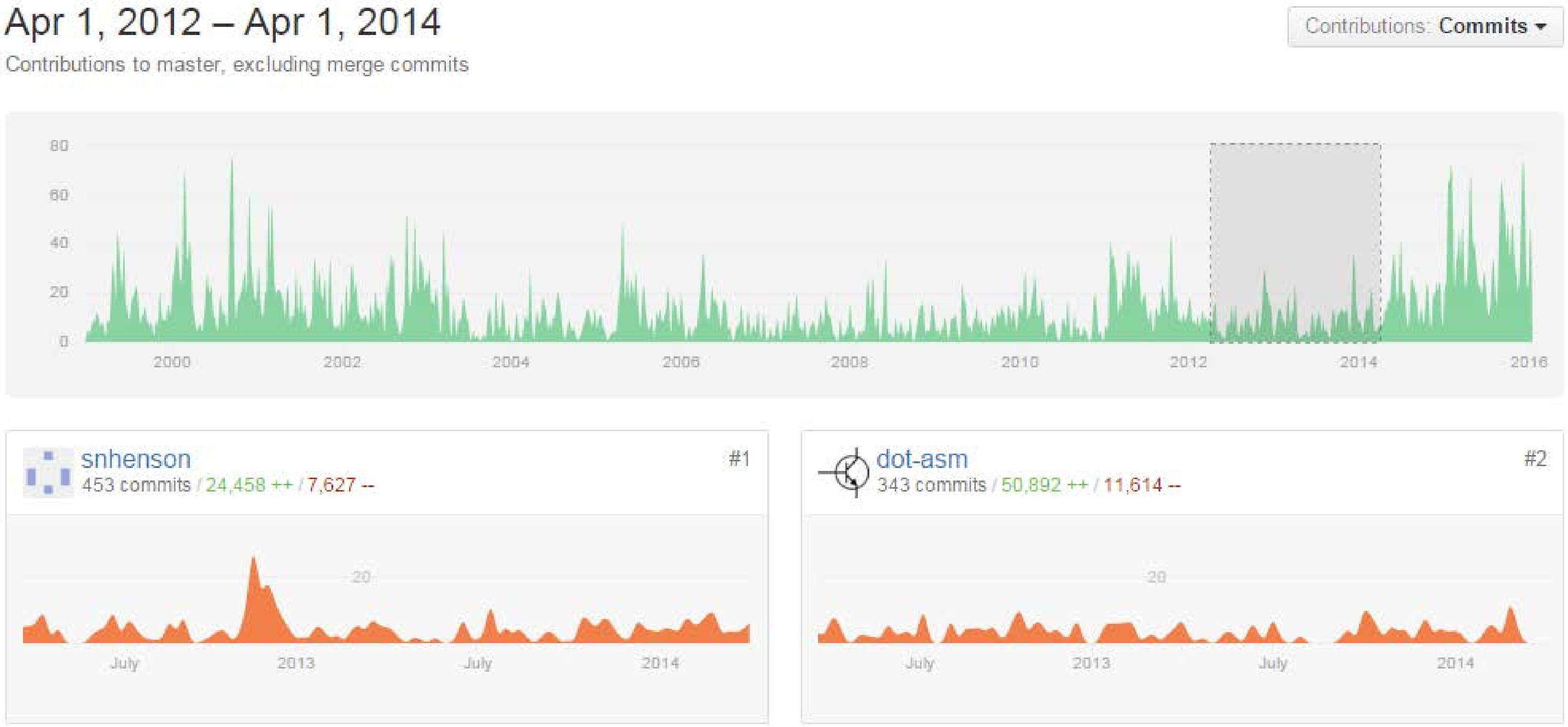
Stats (+605 lines, -19 lines)

1 chunk +6 lines, -0 lines

Life before HeartBleed

- Project had effectively become somewhat moribund
- Releases were not pre-announced, no documented policies
- Source code was complex and arcane
- Hard to maintain; harder to contribute
- Main developers were overworked and overcommitted
- Project donations minimal (sub USD$2000 per annum)

2012 - 2014



Apr 1, 2012 – Apr 1, 2014

Contributions to master, excluding merge commits

Contributions: **Commits** ▾

snhenson #1
453 commits / 24,458 ++ / 7,627 --

dot-asm #2
343 commits / 50,892 ++ / 11,614 --

OpenSSL
Cryptography and SSL/TLS Toolkit

How did we let this happen?

- Very little time spent on building community
- Long lead time to understand code
- Static project team membership
- Need to focus on consulting dollars (FIPS140) to keep project alive
- No ability to make, announce, and keep to plans
- … all added up to “ultra cautious” to any change attitude

The usual questions …

- How could the project let this happen?
- How could the project members be so stupid?
- What other nasty break-the-internet bugs are yet to be found?
- Why didn't the project fix this sooner?
- Why didn't all those companies making money off OpenSSL contribute?
- How could we possibly trust the team to not make the same mistake in future?
- Why shouldn't I simply switch over to one of the forks?

After-affects

- Wider recognition of dependency on critical under-funded projects
- Creation of the Core Infrastructure Initiative, a multi-million dollar effort to add effective resources to the open source projects that make the Internet work

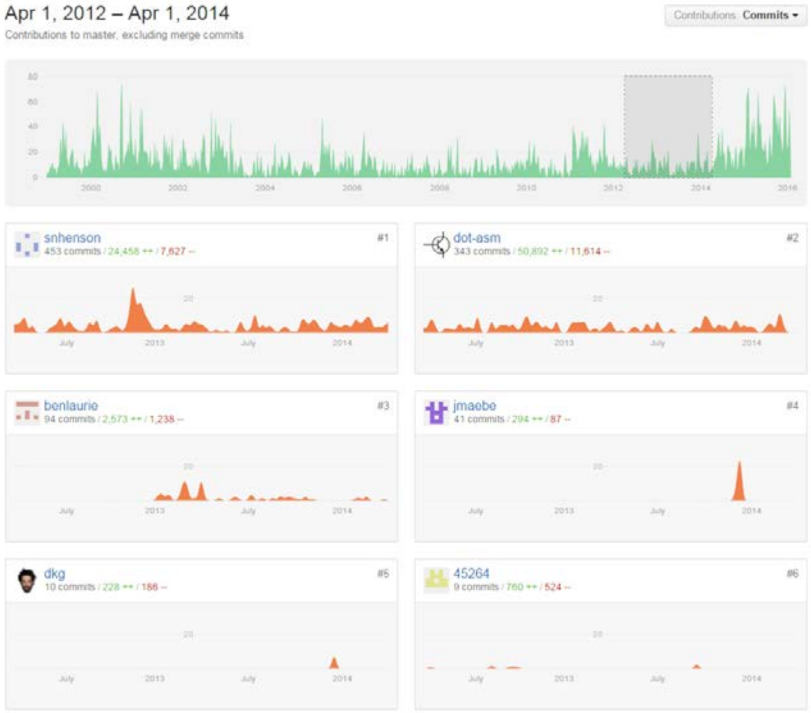
CORE
INFRASTRUCTURE
INITIATIVE

https://www.coreinfrastructure.org/

Growing the Team

- Prior to April 2014
  - Two main developers (one primary committer) entirely on volunteer basis; all other team members focused on other areas; main developer basically funded by paid OpenSSL consulting work
  - No formal decision making process
- As of December 2014
  - Fifteen project team members;
  - Two full time funded by CII; two full time funded by donations
  - Formal decision making process

2014-2016



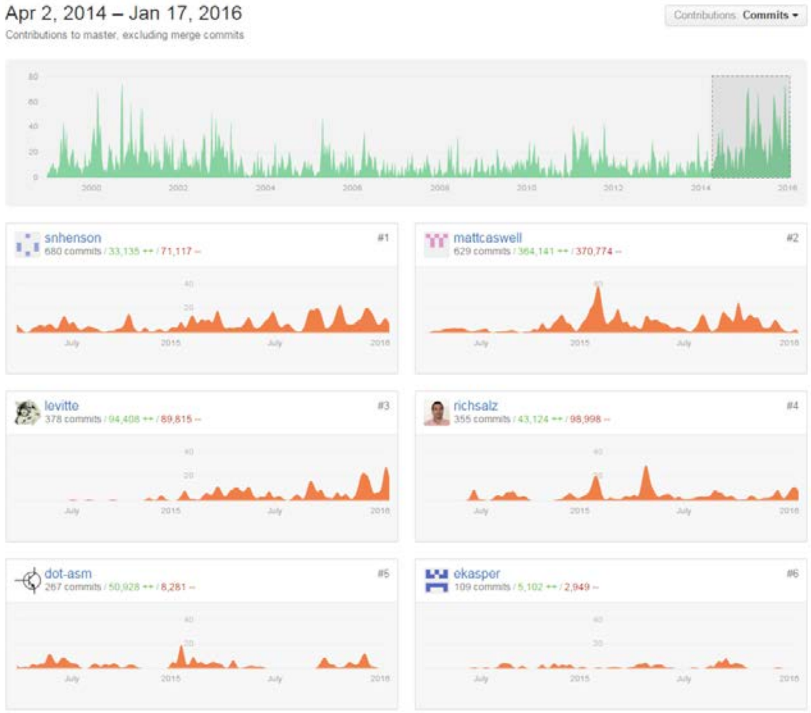
Apr 1, 2012 – Apr 1, 2014

Contributions to master, excluding merge commits

Contributions: Commits ▾

snhenson	453 commits / 24,458 ++ / 7,627 --	#1
dot-asm	343 commits / 50,892 ++ / 11,614 --	#2
benlaurie	94 commits / 2,573 ++ / 1,238 --	#3
jmaebe	41 commits / 294 ++ / 87 --	#4
dkg	10 commits / 228 ++ / 186 --	#5
45264	9 commits / 760 ++ / 524 --	#6

→



Apr 2, 2014 – Jan 17, 2016

Contributions to master, excluding merge commits

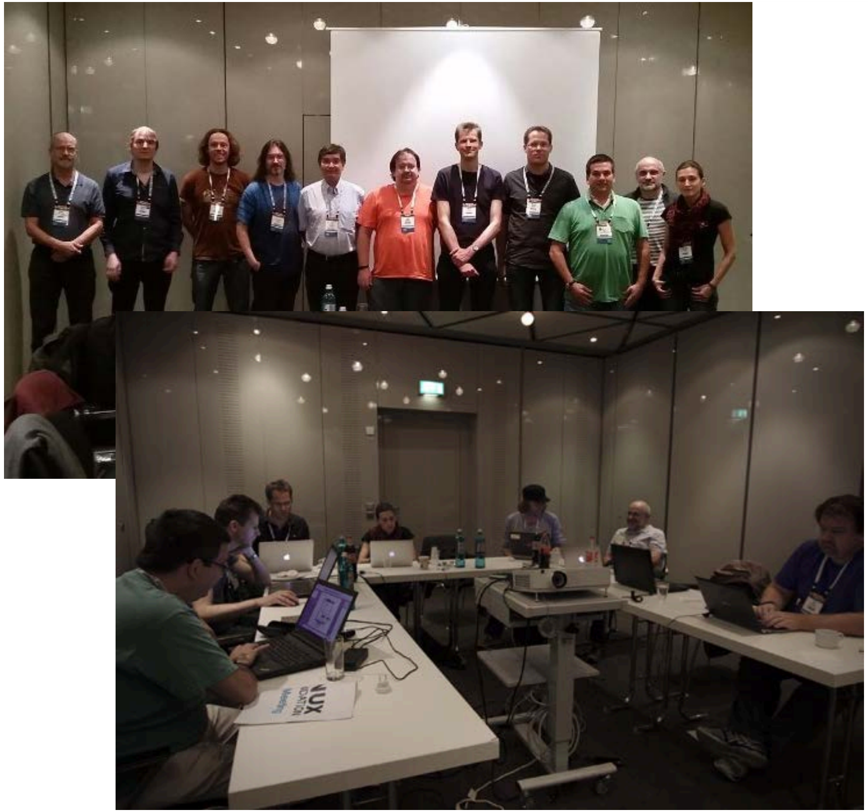
Contributions: Commits ▾

snhenson	680 commits / 33,135 ++ / 71,117 --	#1
mattcaswell	629 commits / 364,141 ++ / 370,774 --	#2
levitte	378 commits / 94,408 ++ / 89,815 --	#3
richsalz	355 commits / 43,124 ++ / 98,998 --	#4
dot-asm	267 commits / 50,928 ++ / 8,281 --	#5
ekasper	109 commits / 5,102 ++ / 2,949 --	#6

OpenSSL
Cryptography and SSL/TLS Toolkit

After-affects

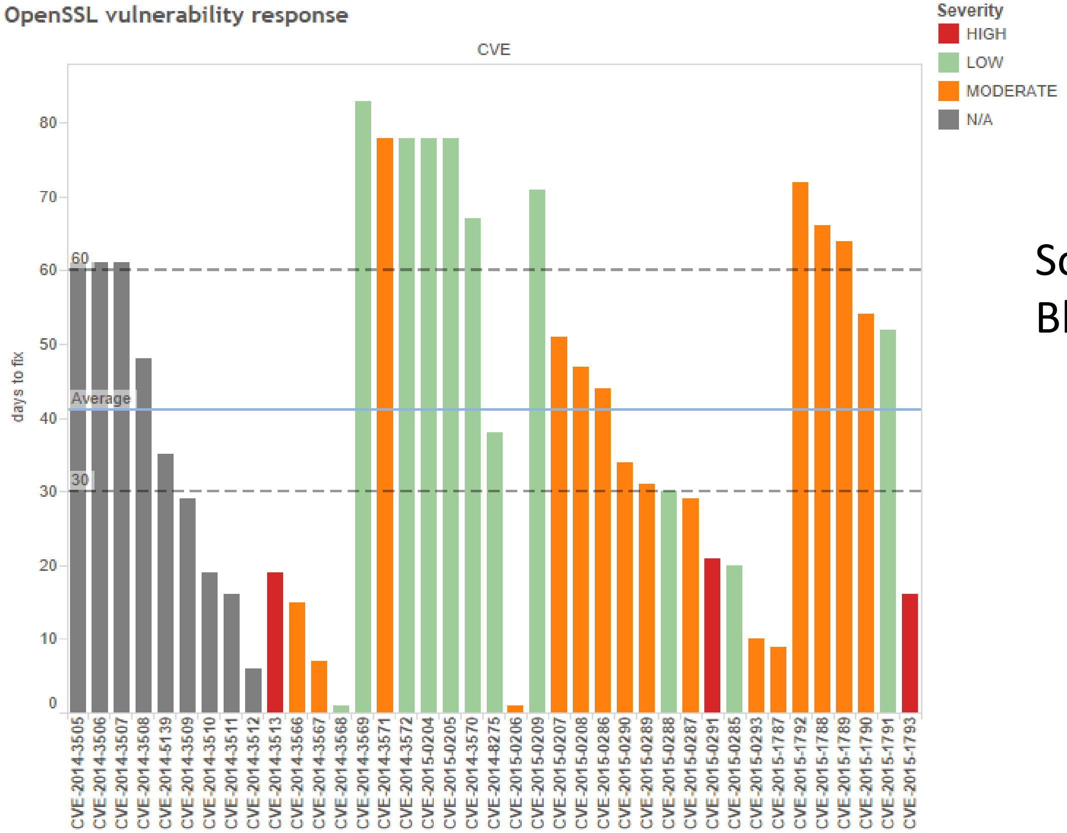
- We had the first-ever F2F
- Drafted major policies:
  - Release strategy
  - Security policy
  - Coding style
- Socialized with each other; POODLE helped



Transparency

- GitHub is used.
- We have public policies for security fixes, a release schedule and high-level content, code of conduct, and so on.
- Email traffic increased, and (seems) more useful

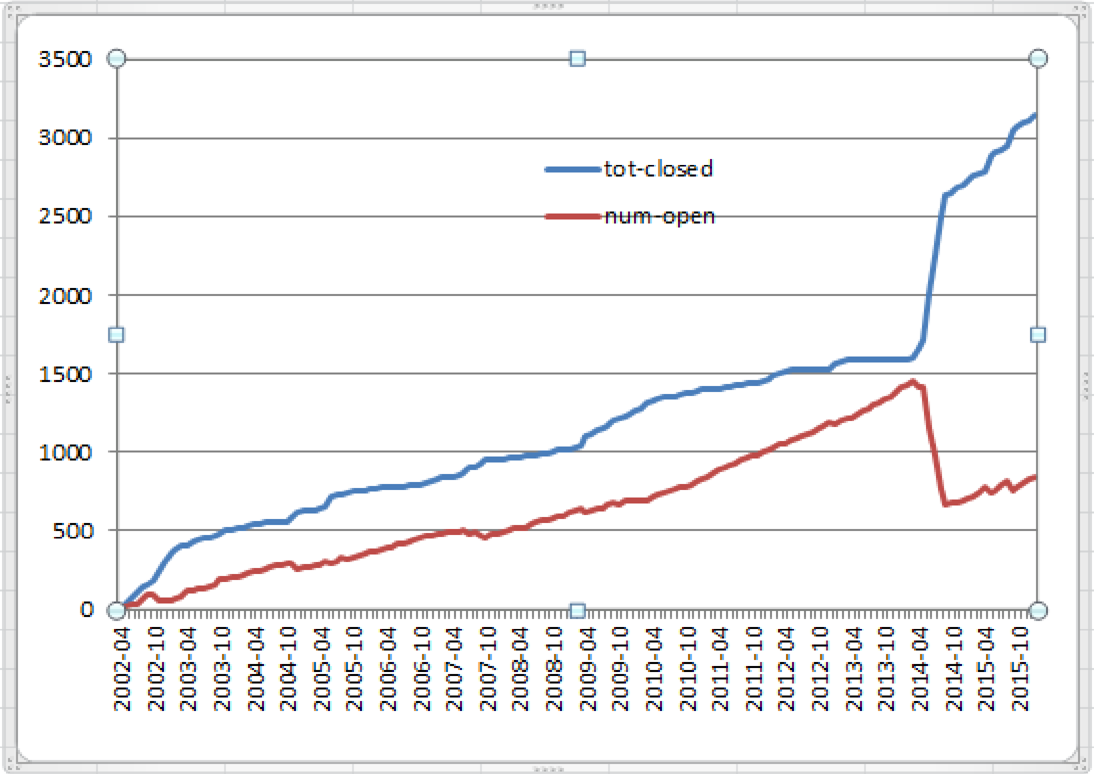
Transparency: security fixes



Source: OpenSSL Blog Entry

Bug tracking

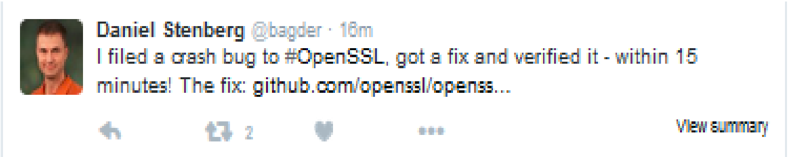
- Much better, but not yet good enough.
- Too many bugs
- Too many *old* bugs.



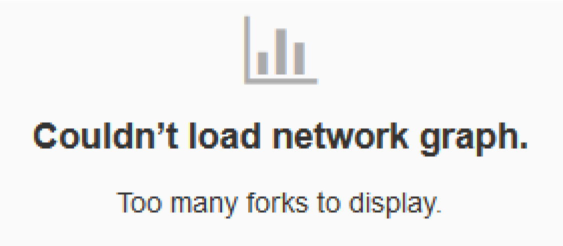
Renewed focus

- Security researchers more actively looking for issues
- More fuzz testing going on
- Increased focus on automated testing
- Static code analysis tools rapidly updated
- Reported issues more quickly analyzed
- **Mandatory team member code reviews**

Vitality is its own reward

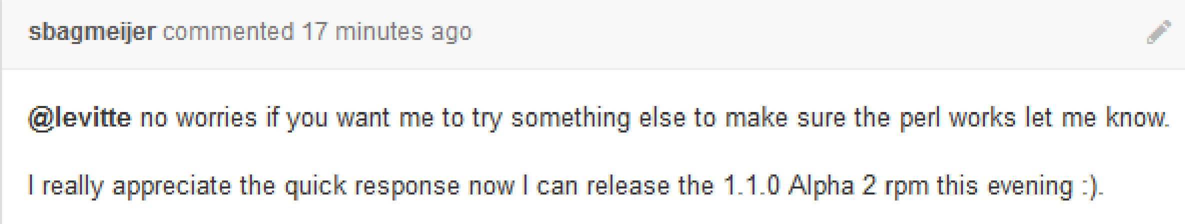


Daniel Stenberg @bagder · 16m
I filed a crash bug to #OpenSSL, got a fix and verified it - within 15 minutes! The fix: github.com/openssl/openss...



Couldn't load network graph.

Too many forks to display.



sbagmeijer commented 17 minutes ago

@levitte no worries if you want me to try something else to make sure the perl works let me know.

I really appreciate the quick response now I can release the 1.1.0 Alpha 2 rpm this evening :).

FIPS140

- FIPS140 related work effectively entirely funded the OpenSSL project for the last five years
- Selling into USA Government where FIPS140-2 support is mandatory is important to most large vendors
- The validation process is time consuming and subject to changed requirements
- Coordinating multiple sponsors on a multi-year journey with no guarantee of successful outcome is in itself challenging

FIPS140

- The OpenSSL FIPS 2.0 module works with OpenSSL-1.0.x
- The previous OpenSSL FIPS 1.0 module for OpenSSL-0.9.x is no longer usable
- A major update will be required for a new OpenSSL FIPS module to work with OpenSSL-1.1.x which currently remains unfunded and unplanned
- Objective is to make the FIPS140 related changes “less intrusive”

FIPS140-2 Operational Environments

- Android 2.2 (gcc Compiler Version 4.4.0); Android 2.2 running on Qualcomm QSD8250 (ARMv7) with NEON (gcc Compiler Version 4.4.0); Microsoft Windows 7 (32 bit) (Microsoft 32 bit C/C++ Optimizing Compiler Version 16.00); uCLinux 0.9.29 (gcc Compiler Version 4.2.1); Fedora 14 running on Intel Core i5 with AES-NI (gcc Compiler Version 4.5.1); HP-UX 11i (32 bit) (HP C/aC++ B3910B); HP-UX 11i (64 bit) (HP C/aC++ B3910B); Ubuntu 10.04 (32 bit) (gcc Compiler Version 4.1.3); Ubuntu 10.04 (64 bit) (gcc Compiler Version 4.1.3); Android 3.0 (gcc Compiler Version 4.4.0); Linux 2.6.27 (gcc Compiler Version 4.2.4); Microsoft Windows 7 (64 bit) (Microsoft C/C++ Optimizing Compiler Version 16.00); Ubuntu 10.04 running on Intel Core i5 with AES-NI (32 bit) (gcc Compiler Version 4.1.3); Linux 2.6.33 (gcc Compiler Version 4.1.0); Android 2.2 running on OMAP 3530 (ARMv7) with NEON (gcc Compiler Version 4.1.0); VxWorks 6.8 (gcc Compiler Version 4.1.2); Linux 2.6 (gcc Compiler Version 4.3.2); Linux 2.6.32 (gcc Compiler Version 4.3.2); Oracle Solaris 10 (32 bit) (gcc Compiler Version 3.4.3); Oracle Solaris 10 (64 bit) (gcc Compiler Version 3.4.3); Oracle Solaris 11(32 bit) (gcc Compiler Version 4.5.2); Oracle Solaris 11 (64 bit) (gcc Compiler Version 4.5.2); Oracle Solaris 11 running on Intel Xeon 5675 with AES-NI (32 bit) (gcc Compiler Version 4.5.2); Oracle Solaris 11 running on Intel Xeon 5675 with AES-NI (64 bit) (gcc Compiler Version 4.5.2); Oracle Linux 5 (64 bit) (gcc Compiler Version 4.1.2); CascadeOS 6.1 (32 bit) (gcc Compiler Version 4.4.5); CascadeOS 6.1 (64 bit) (gcc Compiler Version 4.4.5); Oracle Linux 5 running on Intel Xeon 5675 with AES-NI (gcc Compiler Version 4.1.2); Oracle Linux 6 (gcc Compiler Version 4.4.6); Oracle Linux 6 running on Intel Xeon 5675 with AES-NI (gcc Compiler Version 4.4.6); Oracle Solaris 11 (32 bit) (Sun C Version 5.12); Oracle Solaris 11 (64 bit) (Sun C Version 5.12); Android 4.0 (gcc Compiler Version 4.4.3); Apple iOS 5.1 (gcc Compiler Version 4.2.1); Microsoft Windows CE 6.0 (Microsoft C/C++ Optimizing Compiler Version 15.00 for ARM); Microsoft Windows CE 5.0 (Microsoft C/C++ Optimizing Compiler Version 13.10 for ARM); Linux 2.6 (gcc Compiler Version 4.1.0); DSP Media Framework 1.4 (TMS320C6x C/C++ Compiler v6.0.13); Android 4.0 running on TI OMAP 3 (ARMv7) with NEON (gcc Compiler Version 4.4.3); NetBSD 5.1 (gcc Compiler Version 4.1.3); Microsoft Windows 7 running on Intel Core i5-2430M (64bit) with AES-NI (Microsoft C/C++ Optimizing Compiler Version 16.00 for x64); Android 4.1 running on TI DM3730 (ARMv7) (gcc Compiler Version 4.6); Android 4.1 running on TI DM3730 (ARMv7) with NEON (gcc Complier Version 4.6); Android 4.2 running on Nvidia Tegra 3 (ARMv7) (gcc Compiler Version 4.6); Android 4.2 running on Nvidia Tegra 3 (ARMv7) with Neon (gcc Compiler Version 4.6); Windows Embedded Compact 7 running on Freescale i.MX53xA (ARMv7) with NEON (Microsoft C/C++ Optimizing Compiler Version 15.00.20720); Windows Embedded Compact 7 running on Freescale i.MX53xD (ARMv7) with NEON (Microsoft C/C++ Optimizing Compiler Version 15.00.20720); Android 4.0 running on Qualcomm Snapdragon APQ8060 (ARMv7) with NEON (gcc compiler Version 4.4.3); Apple OS X 10.7 running on Intel Core i7-3615QM (Apple LLVM version 4.2); Apple iOS 5.0 running on ARM Cortex A8 (ARMv7) with NEON (gcc Compiler Version 4.2.1); OpenWRT 2.6 running on MIPS 24Kc (gcc Compiler Version 4.6.3); QNX 6.4 running on Freescale i.MX25 (ARMv4) (gcc Compiler Version 4.3.3); Apple iOS 6.1 running on Apple A6X SoC (ARMv7s) (gcc Compiler Version 4.2.1); eCos 3 running on Freescale i.MX27 926ejs (ARMv5TEJ) (gcc Compiler Version 4.3.2); Ubuntu 13.04 running on AM335x Cortex-A8 (ARMv7) (gcc Compiler Version 4.7.3); Ubuntu 13.04 running on AM335x Cortex-A8 (ARMv7) with NEON (gcc Compiler Version 4.7.3); Linux 3.8 running on ARM926 (ARMv5TEJ) (gcc Compiler Version 4.7.3); Linux 3.4 64bit under Citrix XenServer running on Intel Xeon E5-2430L (x86) without AES-NI (gcc Compiler Version 4.8.0)

Lessons

- Relying on any single individual to perform superhuman feats ultimately results in disappointment
- Code reviews actually require the reviewers to review the code in detail
- Assuming that users will review code is clearly a flawed strategy
- Assuming that automated code analysis tools by themselves can completely replace experienced code reviews is incorrect

Project Roadmap

- Roadmap has been published and progress against roadmap updated - https://www.openssl.org/policies/roadmap.html
- Major items:
  - clear bug backlog
  - documentation
  - complexity
  - coding style
  - code reviews
  - release plan
  - platform strategy
  - security strategy

Apply What You Have Learned Today

- Download the pre-releases and build your applications
- *Help is a two-way street*, join the virtuous circle. Or at least join the openssl-dev and/or openssl-users mailing lists
- Report bugs through RT, submit patches on GitHub. *Help close bugs.*
- If you are doing more than TLS for HTTP, *please let us* know
- More ideas on the Community page of www.openssl.org

The OpenSSL Development Team

- Matt Caswell
- Mark J. Cox
- Viktor Dukhovni
- Steve Henson
- Tim Hudson
- Lutz Jänicke
- Emilia Käsper
- Ben Laurie
- Richard Levitte
- Steve Marquess
- Bodo Möller
- Andy Polyakov
- Kurt Roeckx
- Rich Salz
- Geoff Thorpe

Questions

- Rich Salz - rsalz@openssl.org

- Tim Hudson – tjh@openssl.org