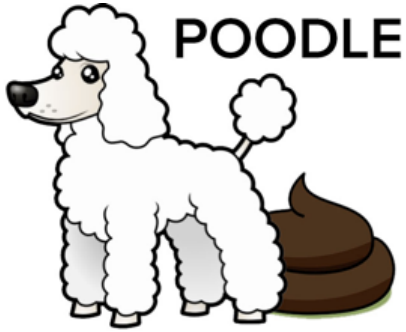
Here's where we are with open source security

… and here's what we need to do about it

Nicko van Someren, Core Infrastructure Initiative

2014 was a bad year for FOSS security

Shellshock




amazon web services
Bloomberg
CISCO
DELL
facebook
FUJITSU
Google
HITACHI
Inspire the Next
hp
HUAWEI
IBM
intel
Microsoft
The Linux Foundation created the Core Infrastructure Initiative with support from 19 Industry Giants
NEC
NetApp
rackspace
salesforce
QUALCOMM
vmware
THE LINUX FOUNDATION
CORE INFRASTRUCTURE INITIATIVE

Core Infrastructure Initiative Mission

- The CII aims to substantially improve security outcomes in the FOSS projects that underpin the Internet
- The CII funds work in security engineering, security architecture, tooling, testing and training on key FOSS projects, as well as supporting general development on security-specific projects (such as crypto libraries)

Security Is Hard For Open or Closed Source - These Are Complex Systems

LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE

FOSS Security Is Different

FOSS is not more or less secure, but it ***is*** different

- Typically there are many more people contributing
- Sometimes (often?) there is a culture of “code is more important than specification”
- Processes are often more ad hoc
- There may be less market pressure to put security first

LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE

What if you don’t have enough eyeballs?

Where is FOSS security in 2016

- Things are better than 2014

  … but we still have a long way to go
- Heartbleed, Shellshock, Poodle, ntpd DDoS etc. were a wake-up call to the open source projects as well as for users and the technology industry
- Security has become a higher priority for many projects

The state of affairs is still highly variable

- Some projects have excellent security process and outcomes
- Many are OK
- Some are terrible

- Quite a lot simply don’t have anyone working on them

Identifying potential sources of risk

- Orphan code in deployment is a problem
  - Ubuntu has nearly 50,000 packages recursively dependent on zlib; the last release was in 2013
- Some projects have higher bug densities than others
- Code that runs with privileges is potentially more dangerous
- Code in memory-unsafe languages is more prone to certain types of dangerous bug

Identifying at risk projects

Census Results

Title	Risk index	CVE count	Contributor Count	Popularity
libexpat1	13	5	10	174,625
procmail	13	2	4	153,379
unzip	13	5	2	137,931
libpcre3	12	6	4	174,907
locales	12	47	12	174,513
multiarch-support	12	47	12	148,819
rsync	12	3	15	102,393
bsd-mailx	12	1	1	151,075
libc-bin	12	47	12	168,914

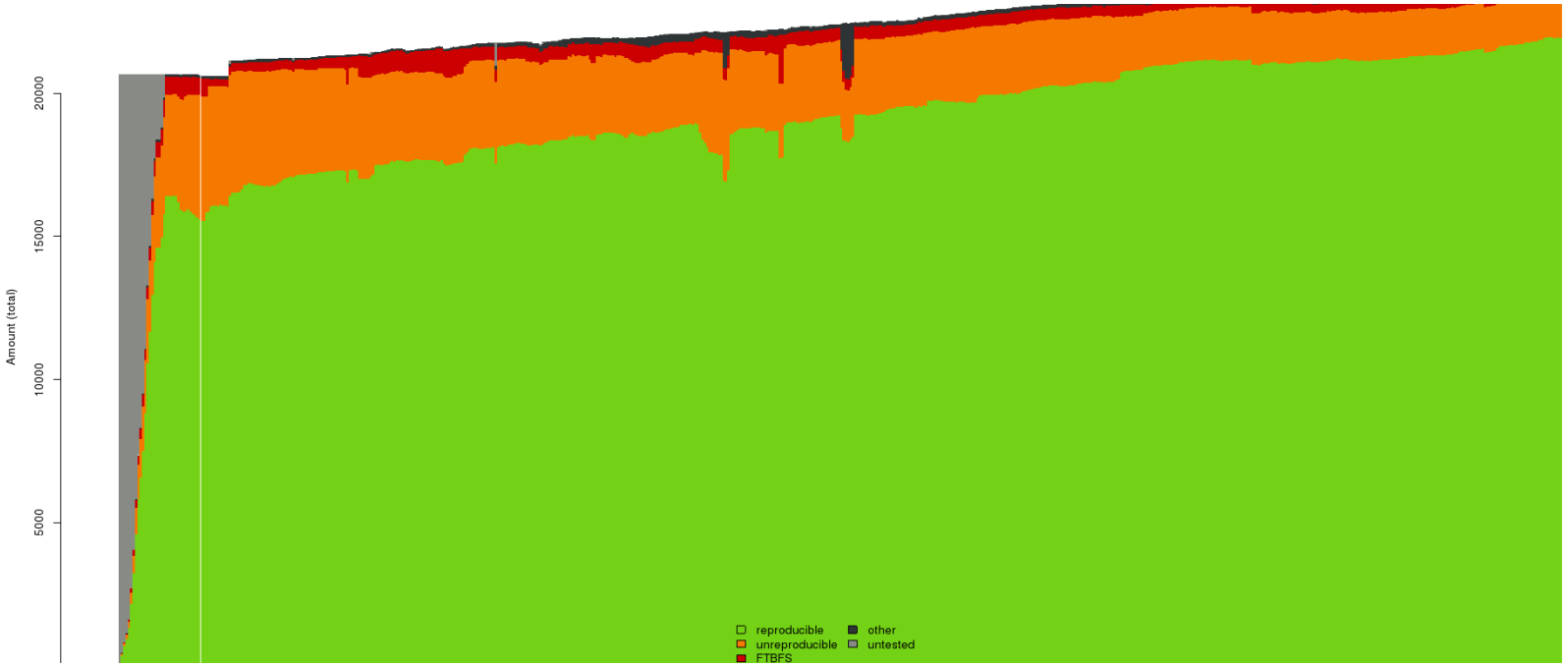
Making progress

- Direct investment has improved bug security process and security outcomes in many projects:
  - OpenSSL, GnuPG, OpenSSH and many more
- NTPSec fork has removed 75% of the code in ntpd without compromising the functionality
- Census project has allowed us to identify and target packages that are at risk
- Reproducible builds is allowing users to check binaries
- Badging has improved security process in 100s of projects

The impact of the CII

- The CII has directly invested in dozens of projects
  - Typical distributions have 20,000+ packages
  - We are only scratching the surface in direct investment
- Some of our projects have very wide reach
  - The Fuzzing Project has tested and reported bugs on hundreds of projects
  - Reproducible Builds has tested ALL 23,931 Debian ‘testing’ source packages

Reproducible Builds: Debian at 91.5%

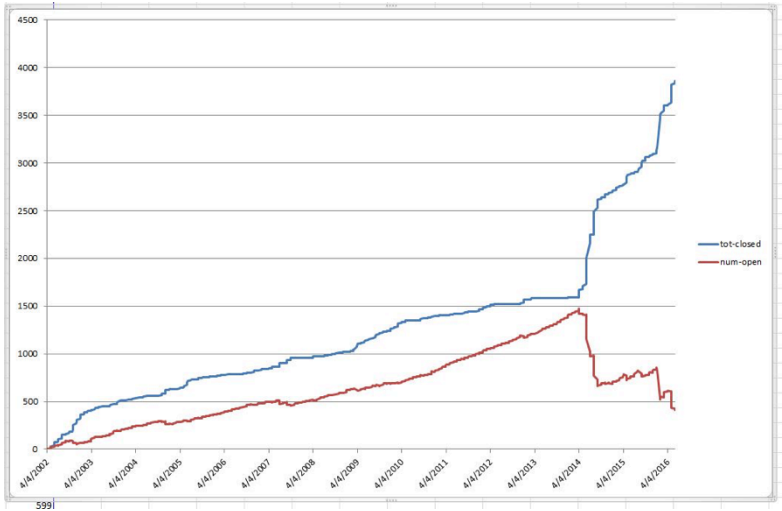


THE LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE

Successes with OpenSSL Governance

- Bugs are found faster **and** closed faster
- More progress on security roadmap items
- New release policies mean security updates are being deployed more quickly

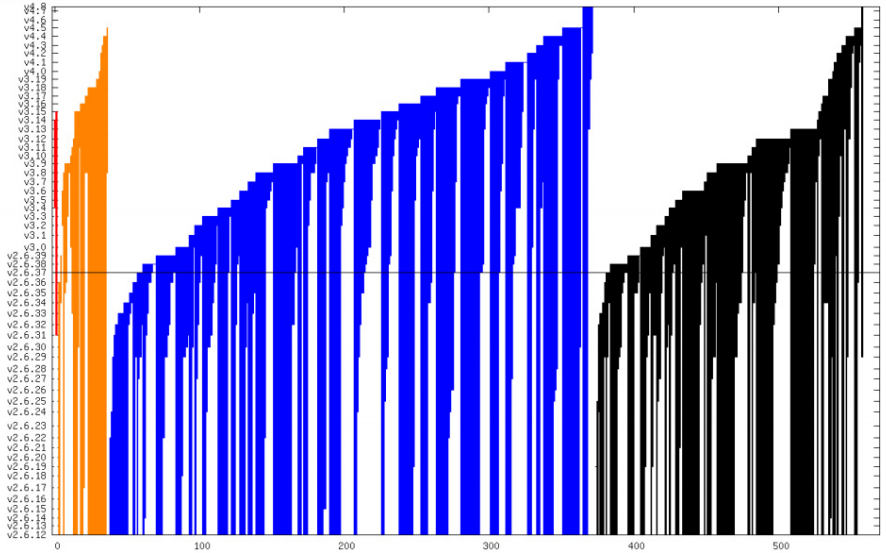


Where do we go next?

- Many bugs are staying unfixed for too long
- Many projects still resist any security improvements that impact performance
- Still too much orphan code in use

Kees Cook's Linux bug time line

Critical and high-severity security bugs in the upstream kernel have lifespans from 3.3 to 6.4 years between commit and discovery.



A cost worth paying

- Many of the well known and well understood ways to mitigate against the impact of security vulnerabilities have performance costs
- Deploying techniques for isolation and self-protection can significantly reduce the risk of harm from whole classes of bug, not just from individual, identified bugs
- Projects (and users) need to realise that these costs are worth paying

Security is a process, not a product

- Projects like the CII Best Practice Badge have been encouraging projects to think more about their security process
- Even mature, well-run projects have been benefitting
- This requires buy-in from the whole project community

Scaling up the impact of the CII

- Tools for testing
  - OWASP ZAP
- Tools for assessment
  - Fuzzing
- Tools for promoting best practices
  - Badging
- Tools for training

The future of FOSS security

- Need to win hearts and minds
- No one size fits all

- Find the projects that matter
- Assess their status
- Work out what they need
- Provide it

Conclusions

- In short, things are getting better but we still have a long way to go
- If Open Source software is to become the dominant force in corporate IT then security ***must*** be a core selling point
- Security ***must*** be something that projects thing about early and often and they ***must*** be willing to prioritise it as highly as other features

Thank you.

https://www.coreinfrastructure.org