Securing Open Source Software Through Strong Governance

Dr. Nicko van Someren
Executive Director, Core Infrastructure Initiative
2014 – Heartbleed

The Linux Foundation forms the Core Infrastructure Initiative
with support from 19 Industry Giants
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)
Ensuring Strong Security Processes

- Think about security at every step of the process: architecture, implementation, testing, documentation, distribution and deployment

- It is not sufficient to have a strong Secure Development Lifecycle (SDLC) policy; you need governance and leadership to ensure that people follow it
Security Is Hard For Open or Closed Source - These Are Complex Systems
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
Linus’s Law: “Given enough eyeballs, all bugs are shallow.”
Why FOSS Security Can Be The Best

- Peer review is one of the best tools available for ensuring code and designs are secure, and FOSS does peer review very effectively
What Does Good Security Governance Look Like?

- Good security governance requires checks and balances
- Security needs to be hard wired into a project, not layered on
- Security should start before coding starts
  - Security is a process, not a discrete feature.
Get All Project Members to Buy Into The Process
Good Security Governance

- Can and should be a living document
  - Consistent coding style makes errors easier to spot
  - Ask all contributors to identify their security assumptions
  - Documentation must describe how to do a secure deployment
    - Yes, you’re going to need documentation!
  - Enforce architecture and code review processes
Setting Security as a Priority

- Most *structural* security failures happen because developers didn’t stop to think about security, not because they thought about it but missed something
- At some level, having a policy about security process and following it is more important than the details of the policy itself
- Making security a priority in the project direction and keeping the issues top of mind helps a huge amount
Multi-party Code Review is Critical

- Most vulnerabilities come about because an attacker found a way to violate assumptions made by the developer
- Design, then design review, coding, then code review helps a great deal at spotting false assumptions
Tracking Code Provenance is Crucial

▪ From a security standpoint, it is very important to know not only who wrote a piece of code but also who reviewed it
▪ Tools for tracking code provenance can also be used for tracking code reviews
▪ Ideally a project should be able to know not only who wrote each line of code but who authorised the pull into the trunk
The Role of Technical Advisory Boards

- As with finding bugs in code, it’s often hard to find bugs in your own processes
  - “We’re used to doing it this way” is all too common
- Your TAB should be constantly reexamining your security process to make sure that it still meets your needs
- TAB members may be have valuable insights about real-world deployment that help improve threat models
- At least one TAB member should be a security maven
The CII Best Practice Badge
More Than 40 Best Practice Badge Holders
CII Best Practice Badge Program

- The CII Best Practice Badge Program is a self-assessment process for checking that your FOSS project has good security practices
- The project is itself open source, both for the code that implements the questionnaire and the set of questions that make up the criteria
- The projects self-assess. The answers are public. The community polices the accuracy of these answers.
CII Best Practice Criteria

- Currently about 70 questions
  - Most are required, some are suggested or marked as future requirements
- Answers filled in on a web form. Private until complete; public once a badge is achieved.
- Much of the form-filling is automated is the code is on GitHub (adding other repositories soon)
- Questions are grouped into categories
CII Best Practice Criteria

- Criteria categories include:
  - Defined contribution policies and guidelines
  - Documentation completeness
  - Change control process and checks
  - Bug and vulnerability reporting
  - Testing, test coverage and quality process
  - Crypto and security-specific design
  - Automated security analysis and testing
OpenSSL: A Governance Case Study

After Heartbleed, CII started funding the OpenSSL team

- Worked with them to improve the security governance
  - Formal code review requirements
  - Formal policies for change control
  - Formal policies on bug handling
  - More collaborative architecture review
  - Efforts to ensure policies were followed
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
Conclusions

▪ It is much easier to achieve good security outcomes with a sound Secure Development Life Cycle in place
▪ The SDLC will only be effective if people are watching to make sure that it is adhered to
▪ The technical leadership of a project needs to set an example and apply pressure when it is not followed
▪ None of this is rocket science!
  ▪ It just needs buy-in from the community