Driving Security Process in your Open Source Project

Nicko van Someren
The Linux Foundation

LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE

Open Source Software has had it's fair share of major security issues

```
$ env x='() { :;}; echo vulnerable'
```

bash
Shellshock
POODLE
DDoS ATTACK
Shift

LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE

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

LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE

FOSS Security Is Different Though

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.”

But what if you don’t have enough eyeballs?

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)

- The CII is a project run by the Linux Foundation

CII is a non-profit, funded by membership donations, largely from the tech industry

Multiple pillars to the CII’s approach

- Find where the risky projects are
- Help them fix their own code
- Support the development of better OSS security tools
- Teach developers to use security tools
- Encourage developers to make security a priority within their projects

What can we do to improve the security of Open Source Software?

We can do all the same things as we do when building commercial software

The big difference is that we have to do it collaboratively, without having a top-down mandate demanding it

LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE

Security is a process, not a product

- Think about security early. Think about security often.
- This requires buy-in from the whole project community

  Fostering a culture of security within your open source project is the single most important thing that you can do to improve your security outcomes
- Security needs to be given equal weight with scalability, performance, usability and all the other design factors that matter to your users

Applying “Best Practice” to FOSS

- There are a great many widely known and widely used techniques that have been shown to improve security outcomes
- The CII Best Practice Badge program aims to get projects to actually use them!
- The Best Practice Badge web application is an open source project

  … as is the set of criteria that it applies

Security design

- Build a threat model and keep it up to date
  - Threat modelling doesn’t need to be hard or complex
  - **Tool**: Elevation of Privilege Threat Modelling Card Game
- Don’t use weak crypto
  - And definitely don’t try to design your own crypto!
- Know your dependencies
  - Fix known broken things

Change control

- Tracking who proposed changes, who reviewed those changes and who released them is critical to security.
  - This is often more complex in collaborative OS projects
  - Failures with this are how Heartbleed made it into OpenSSL
- As soon as your project has one or more people coding you need a policy for how code will get reviewed

Change control

- Use a version-controlled source repository
  FOSS **Tool**: git, Mercurial, bazaar
- Make code publicly visible between major releases
  - Public code review before final release is valuable
- Change logs are a must
  - If other people rely on your code, you can break ***their*** security by changing things in ***your*** code

Quality testing

Not all bugs represent vulnerabilities

… but all vulnerabilities are bugs, and…

It’s often very hard to tell the difference (at least until someone publishes an exploit!)

Quality testing

- Writing comprehensive tests is far less fun than writing new code to solve new and interesting problems
- But it’s a hell of a lot more fun than dealing with bugs after they get released
- Measure your test coverage and require collaborators to write tests for all contributed code

FOSS **Tools**: gcov (C/C++), CodeCover (Java), CodeCoverage (Python), and many more…

Security analysis tools

- Fancy commercial static analysis tools are expensive

  … Switching all of your compiler warnings on is not!
- Use linters, code complexity checkers, fuzzers and other analysis tools where you can; they all can help
- Some commercial tools are free for open source projects
- The earlier in the project you start using these the less you will have to deal with “low signal to noise ratio”

FOSS **Tools**: SonarCube, FramaC, AFL & many more

Bug reporting: Closing the SDL loop

- Bugs happen; you need a process for dealing with them
- Users need a way to report security vulnerabilities that doesn’t broadcast them to the whole world!
- Take reports of security vulnerabilities seriously
  - Just because you can’t work out how to exploit a bug doesn’t mean that it can’t be exploited

Tools: Bugzilla, Trac, **GitHub Issues**

None of this is rocket science

- I suspect that most of what I have just outlined is not new to you
    - So why aren’t you doing it all? ☺
- The CII Best Practice Badge is ‘just a checklist’
    - To date we’ve had over 800 projects start the process and only 10% have passed
- Checklists don’t teach you new things to do, they remind you to do things that you should be doing

https://bestpractices.coreinfrastructure.org

Can the CII directly support my project?

Maybe..

- The CII can provide direct support to your OSS project if it meets certain criteria
- It needs to meet at least one of these:
  - Is your project “Core Infrastructure”?
  - Does your project aim to improve the security of other OSS projects?
  - Are you working to improve the security processes in OSS projects?

Getting support from the CII

- If you are working on a project that can impact the security of open source and you would like help then please apply!

https://applications.coreinfrastructure.org

LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE

Conclusions

- Projects ***must*** think about security early & often and they ***must*** be willing to prioritise it as highly as other features
- A strong security process can help to avoid security bugs from creeping in in the first place and helps make it easier and safer to deal with them if they do happen
- Most of the ways that we can make open source software more secure are common industry “best practices”. It is simply a matter of choosing to adopt them.

Thank you.

https://www.coreinfrastructure.org

LINUX FOUNDATION

CORE INFRASTRUCTURE INITIATIVE