Security is a Feature

Presented by Keith Hoodlet
Code Security Architect
GitHub

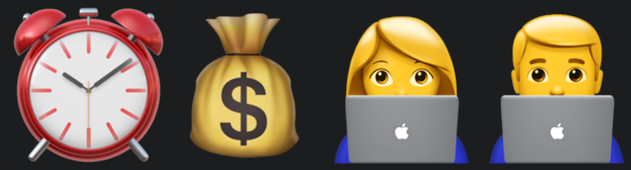
@securingdev

$ vim biography.js

```
1   var Keith_Hoodlet = {
2     title: “Code Security Architect”,
3     company: “GitHub”,
4     miscellaneous: {
5        previously: “Director of Application Experience, Thermo Fisher Scientific”,
6        former_podcast: “Host, Application Security Weekly (eps. 0-55)”,
7        former_hacking: “Top 100 Security Researcher on the Bugcrowd platform”,
8        editor: “Visual Studio Code”,
9        coding: “Python, Bash, and JavaScript”,
10       projects: “infosec_mentors_project & secure-functions”,
11       hobbies: “Reading, writing, gaming, traveling, and learning new languages”
12     }
13  }
```

What makes something a Feature?

Customers ask for Features



Resources are allocated to Features

Features are regularly tested

Features are enhanced over time

@securingdev

So why isn’t Security considered a Feature?

🚰 Water

Customers won’t ask you for it

⚡ Electricity

Costly to fix if not built properly

📱 Cellular

It’s often tested, but not reliably

What happens when you don’t treat Security as a Feature

Vulnerabilities continue to pile up

- Example: Capital One
- Lost staff due to cultural challenges of (in)security
- Cost to the company: $80 Million USD Fine

Vulnerabilities lead to bad publicity

- Example: Uber
- Bounty-as-Data-Breach
- CSO was fired & charged with wire fraud
- $148 Million USD Fine

Your (in)security shapes opinions

- Example: Zoom
- End-to-End Encryption
- The company had to rush to fix its reputation & many security flaws

It doesn’t have to be this way!

Security as Linting:

- Broken Windows Theory applied to Coding
- Linting for Dangerous Functions & Patterns
- Coding Standards

Security as Unit Testing:

- Checking your Recipe & your Ingredients
- Static Analysis
- Dependency Checking

Security as Integration Testing:

- Throwing spaghetti at the wall to see if it sticks
- Dynamic Analysis
- Fuzzing

@securingdev

The benefits of treating Security like a Feature

Attack complexity increases

- Why phishing attacks are (still) so prevalent
- Example: payouts in the Zero Day market

It differentiates the project / product

- Example: Apple products vs others
- Example: React vs. AngularJS (early days)

It saves time and money

- Allowing you to spend it on new features, products, and other technical debt

Where to Get Started

Speed of Deployment

Breadth of Testing

Mean Time to Resolution (MTTR)

Allocating Time for Technical Debt

@securingdev

Free Tools to Start With

Static Analysis

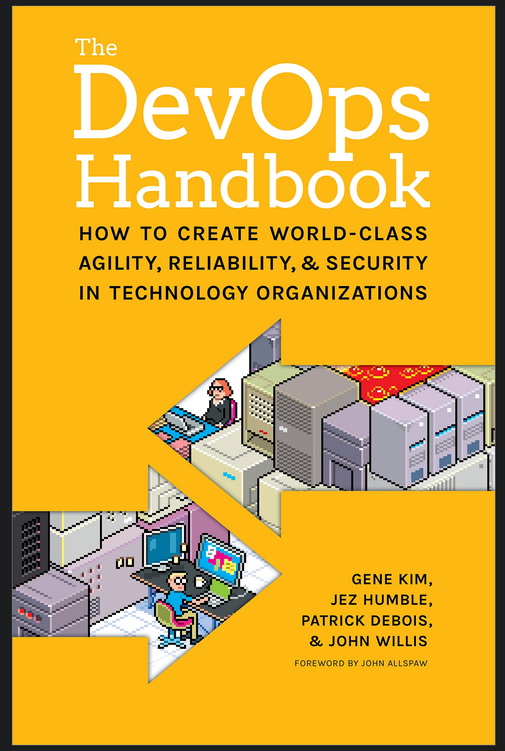
Dynamic Analysis

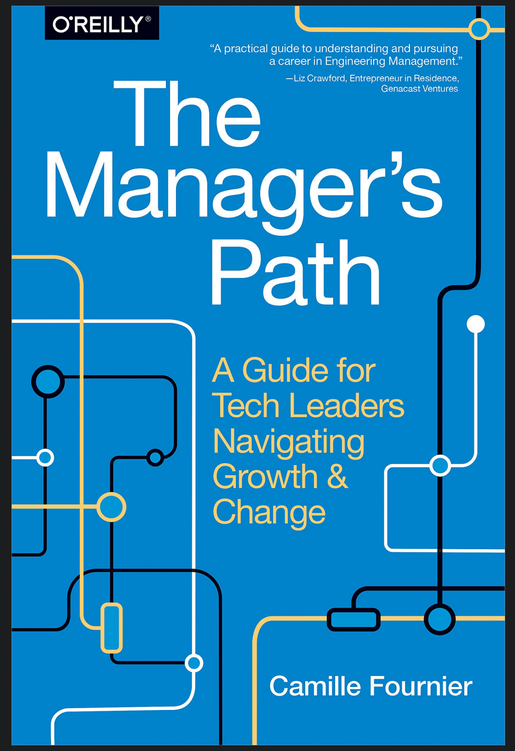
Dependency Checking

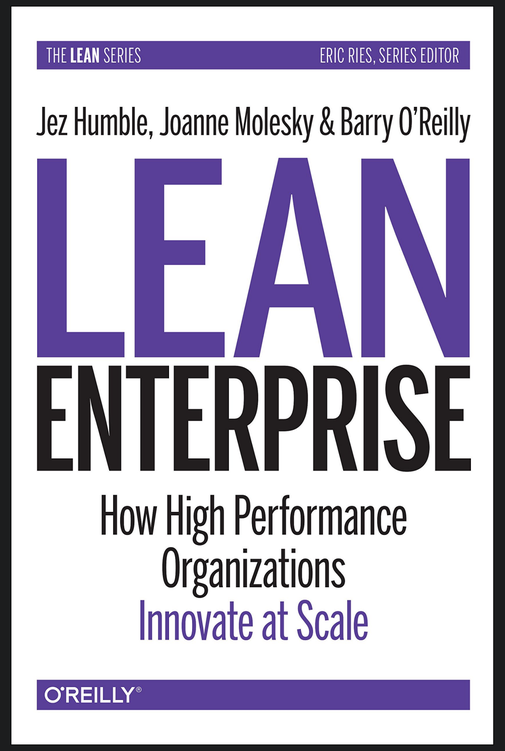
Secrets Management

@securingdev

Further Reading







@securingdev

Special Thanks

@securingdev

Q&A

@securingdev