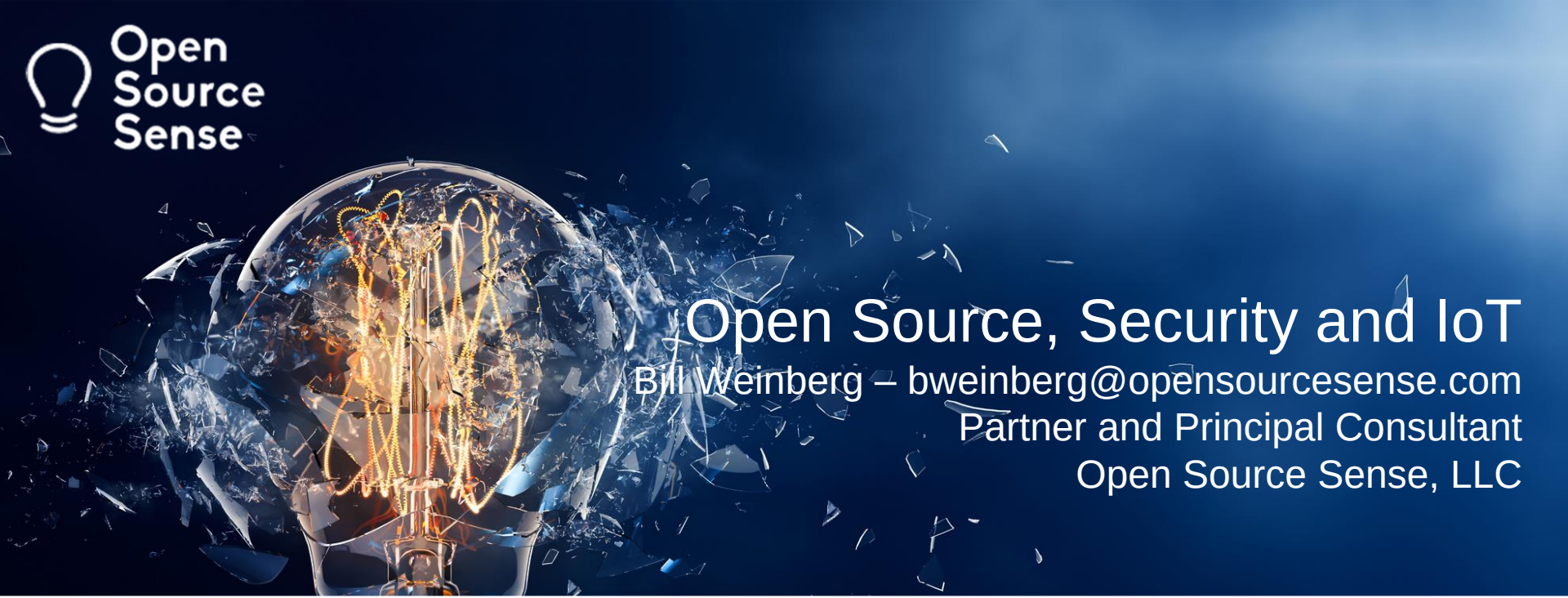
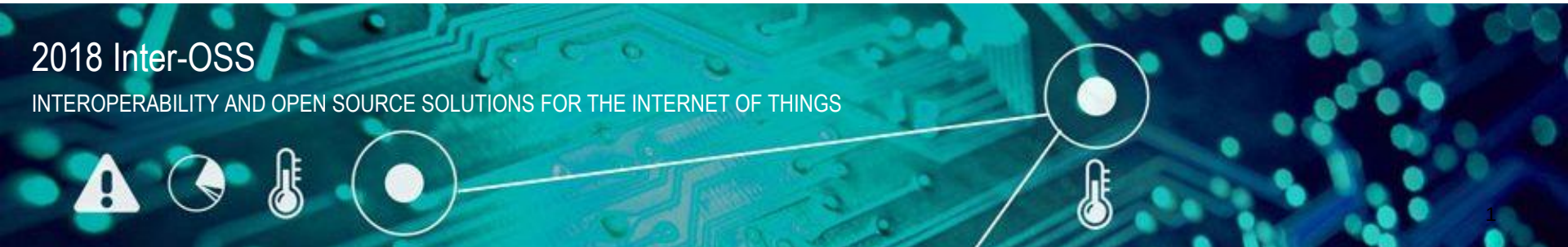


Open Source, Security and IoT

Bill Weinberg – bweinberg@opensourcesense.com
Partner and Principal Consultant
Open Source Sense, LLC



2018 Inter-OSS

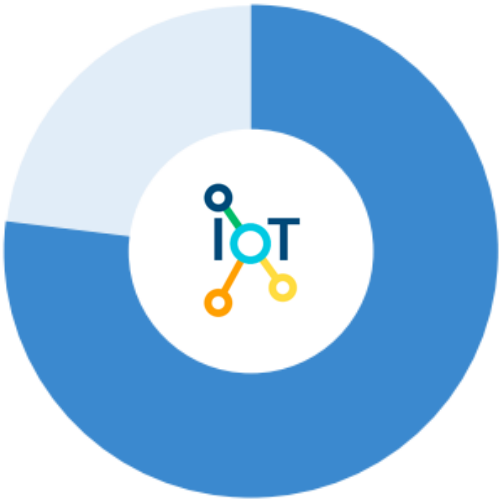
INTEROPERABILITY AND OPEN SOURCE SOLUTIONS FOR THE INTERNET OF THINGS

Abstract

The talk highlights the role open source in the ongoing IoT build-out, across device and cloud ecosystems and within the various IoT project stacks and frameworks. Building upon this vision, the talk examines conventional wisdom and real-world statistics for the security of open source software, and how its presumed resiliency and known (and unknown) vulnerabilities impact security at the edge and in the cloud that hosts IoT apps and infrastructure. Finally, it reviews community and commercial efforts to build more secure code with open source and highlight best practices for choosing and integrating into IoT projects and products.

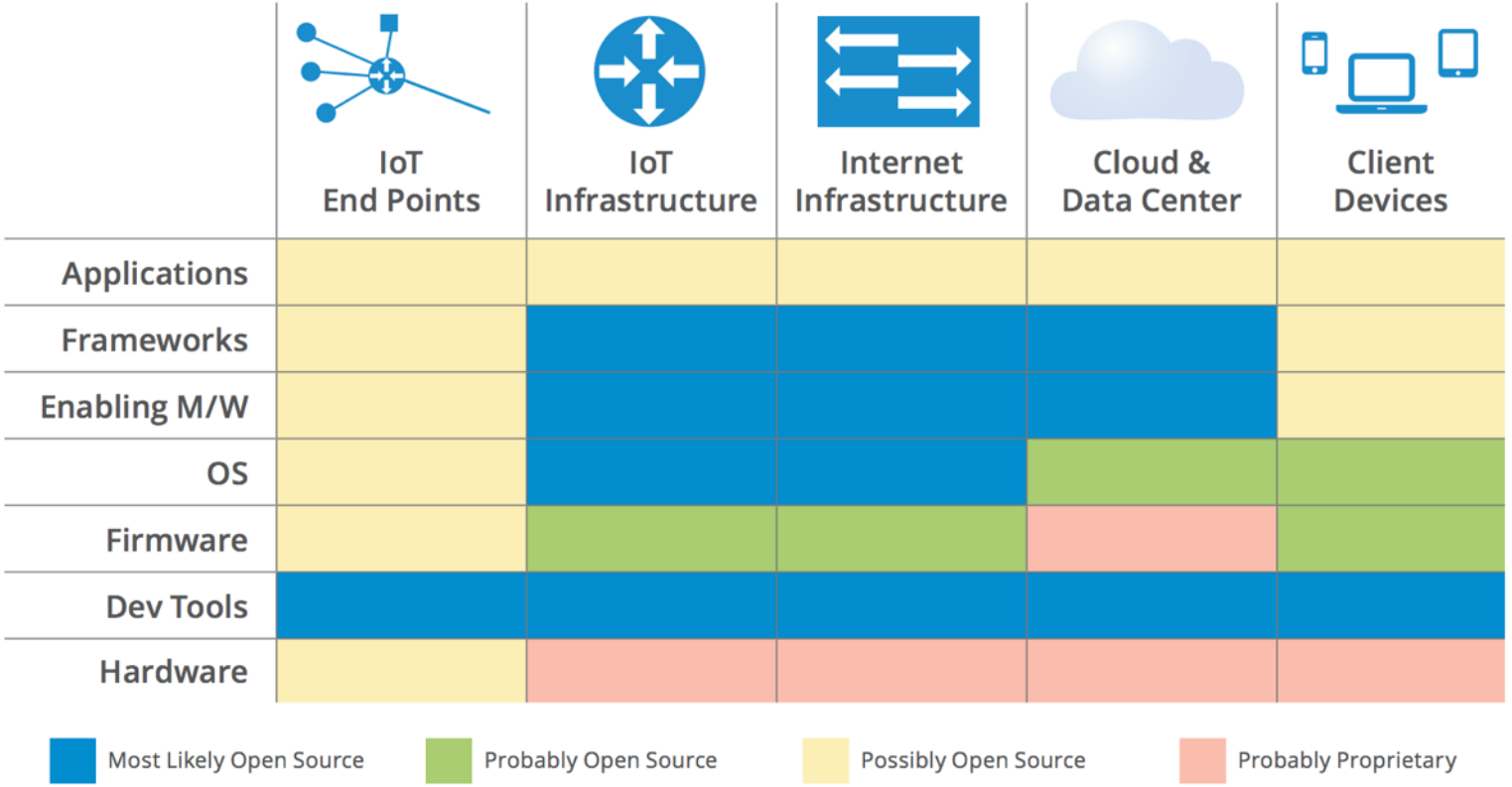
Open Source and IoT

- Open Source is ubiquitous; IoT is or soon will be, too
- The Cloud is built on Open Source; IoT value-added is cloud-based



77% of code in scanned IoT applications
was found to be open source – Black Duck Software

Open Source Across IoT Deployments

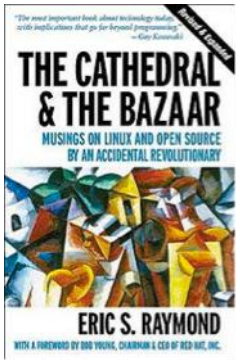


Open Source and Security - Perceptions

- Good
  - Developers view OSS as much more secure than closed source
  - Many security tools are developed as open source
  - Ethical hacking aligned with community-based development
  - Eschew security-by-obscurity
- Less Good
  - Lay-people fear that black hats can more easily exploit open source

Linus' Law

Given enough eyeballs, all bugs are shallow



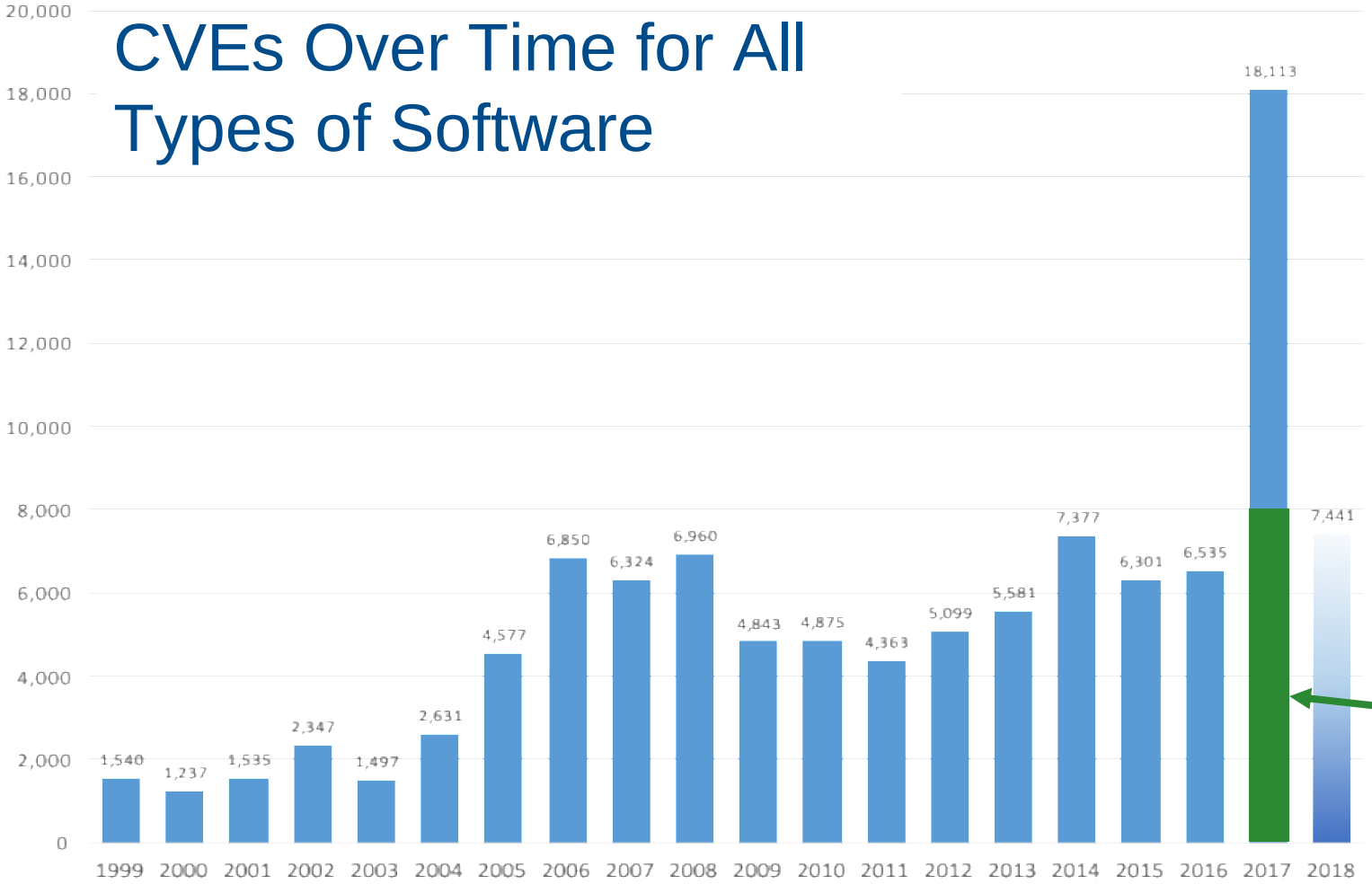
Open Source and Security - Reality

- Open Source is slightly more secure than proprietary s/w
  - 40-45% of CVEs logged against OSS code; 55-60% for other s/w
  - Researchers scrutinize Open Source more; also easier to remediate
- Vulnerabilities in Open Source are publicly disclosed
  - Short-term risk from documentation
- Exploitable OSS makes headlines
  - Heartbleed, Shellshock, Cpython, Poodle, Ghost, SwaggerParser, Spring Web Flow . . .





CVEs Over Time for All Types of Software



NVD
NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Through May 15, 2018

open source

8,000+

Perceptions and Practices around Open Source Security

46%

GIVE OSS FIRST CONSIDERATION AMONG SECURITY TECHNOLOGIES.

55%

SAID OPEN SOURCE DELIVERS SUPERIOR SECURITY

HOWEVER,

67%

DON'T MONITOR OPEN SOURCE CODE FOR SECURITY VULNERABILITIES.

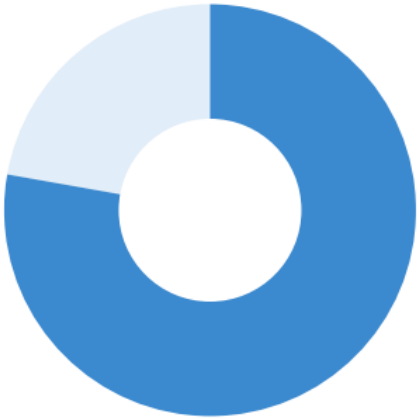
Open Source Sense

Future of Open Source Survey

Application Scanning Results

Black Duck On-Demand audits found open source components in **96%** of the applications scanned, with an average **257** components per application.

The average percentage of codebase that was open source was **57%** vs. **36%** last year. Many applications now contain more open source than proprietary code.



78% of the codebases examined contained at least one vulnerability, with an average **64** vulnerabilities per codebase.

CVEs Logged against Open Source in 2017

Platforms

- Android 1347
- BSD 98
- Linux 832

Web Tech

- Httpd 37
- Chrome 166
- Firefox 118
- Mozilla 21

Data Store

- Hadoop 15
- MariaDB 3
- MySQL 122

Cloud

- Kubernetes 3
- OpenStack 39
- OpenShift 4

Languages

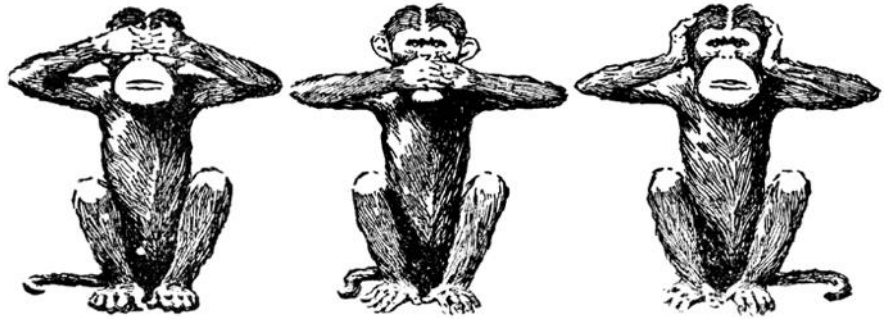
- glibc 20
- Golang 8
- Java 591
- PHP 1416
- Python 40
- Ruby 78
- Swift 2

IoT Security is a Vast Challenge

IoT presents unique security & privacy challenges

- Myriad device types built with varying degrees of security expertise
- A rich mix of public and private data
  - Private data, semi-public meta-data and public aggregate data
- Potential to disrupt operation of industrial and energy systems, and of life-critical connected devices

Security by Obscurity



- Device manufacturers historically rely upon security-by-obscurity (and by simplicity), but practices have been changing.
- OEMs face a combination of evolving pressures:
  - Burgeoning attacks on devices and the networks to which they attach
  - Increase in enterprise-type software in current generation devices
  - Customer and market requirements for more secure device operation, better defense around proprietary/private data on those devices

Mono-function and Limited-function End-Points Present Different Attack Surfaces

- **Physical Attack** – end-point devices can be commandeered, stolen, re-provisioned (re-flashed) with malware, and re-deployed
- **Exposed Ports** – devices present exposed serial, network, USB and other physical interfaces, and can be “cooked” to induce failure modes
- **Spoofing** – of LWANs / mesh networks and man-in-the-middle attacks
- **Simplicity** – Simple devices less stateful, but still include poorly implemented interfaces and authentication, buffer overflow exploits, developer back doors, etc.
- **Life-cycle** – Low-cost devices deployed once and never updated, and so unable to benefit from security updates - \
- **Bandwidth** – “last mile” IoT networking not suited to deploying update images

IoT Attack Surfaces

Ecosystem Access Control	Device Memory	Device Physical Interfaces
Device Web Interface	Device Firmware	Device Network Services
Administrative Interface	Local Data Storage	Cloud Web Interface
Ecosystem Communication	Vendor Backend APIs	Third-party Backend APIs
Update Mechanism	Mobile Application	Vendor Backend APIs
Network Traffic		

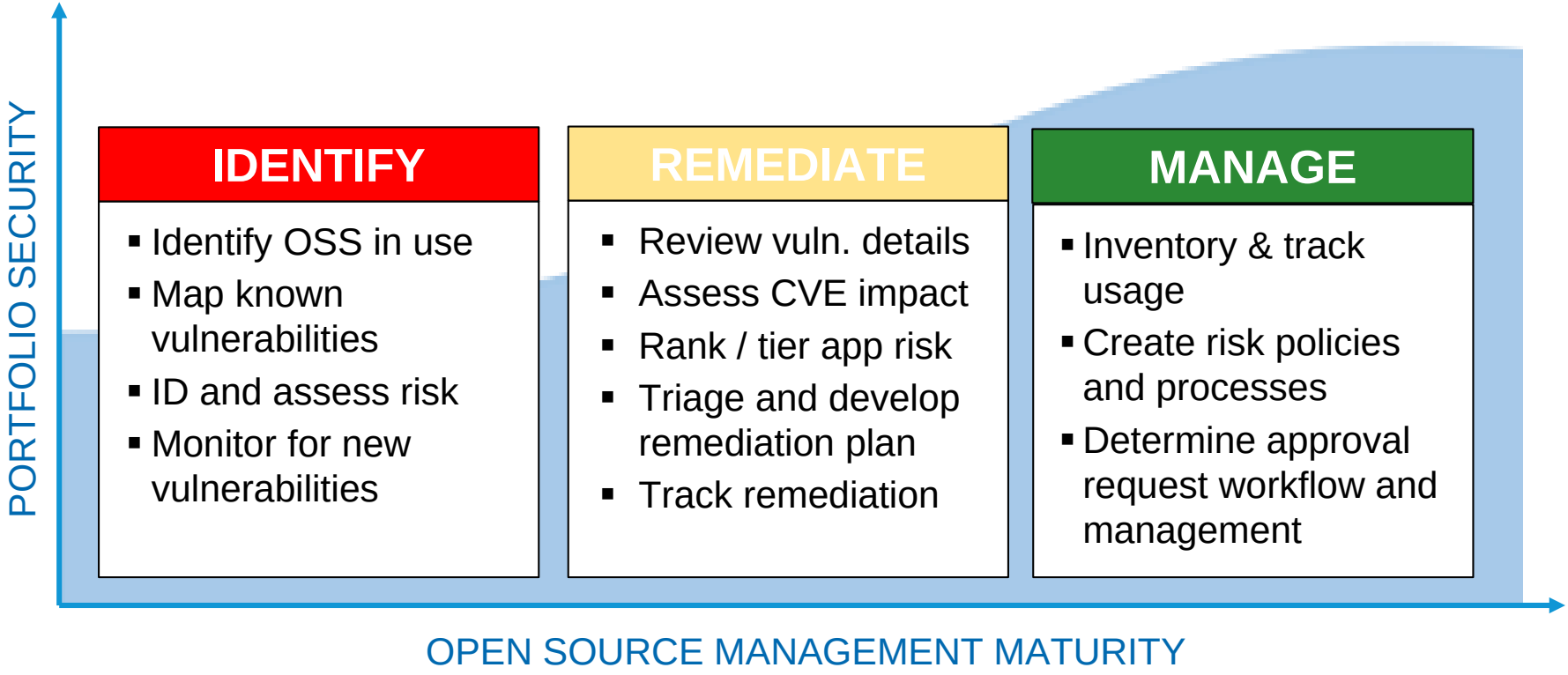
Daniel Miessler, DEFCON 2015

Managing Open Source

Vulnerability Management - Open Source Hygiene

Open Source Hygiene is the practice of cross referencing the open source portfolio of a company or product software stack, module by module, version by version, with databases of known vulnerabilities of those software components.

Open Source Hygiene – Best Practices



OSS Hygiene – Better with Automation

	Manual Procedure 	Automated Process 
Speed	Slow	Faster
Timeliness	Seldom	Automatic
Accuracy	Low	High
Comprehensiveness	With Difficulty	Configurable
Latency	Weeks / Months	Hours
Workflow Impact	Disruptive	Transparent
Repeatable / Traceable	Almost Never	Always
Remediation	Subjective	Policy-based
Cost	FTEs	CapEx / OpEx

Open Source Management Tools
Software Component Analysis and Beyond

Community Security & Management Initiatives

collaborative, pre-emptive approach for strengthening cyber security through auditing, badging and funding project remediation

builds trust in open source by making compliance simpler and more consistent, predictable, understandable and efficient for participants of the software supply chain

Best Practices for Open Source Security

- Employ Open Source Hygiene
- Don’t fork – stay current with project upstream versions
- Clearly segregate private / value-added code from OSS
- Enumerate use cases to understand which code is “outward-facing”
- Understand your organization’s and product’s risk tolerance
- Employ ”triage” process to streamline remediation

Conclusion

- IoT Cybersecurity is a multi-faceted challenge
- Open Source code comprises a large swatch of IoT implementation, with accompany security challenges
- Applying best practices for Open Source security to IoT stacks and applications yields strong ROI and improved security

Key Take-Aways

Identify use of Open Source in IoT Ecosystem

Leverage automation to assess risk from vulnerabilities

Understand use cases for vulnerable code

Build policies to help prioritize remediation

Design, build and maintain updateable Devices and apps

Practice Open Source Hygiene