Vulnerability Management in Software: Before Patch Tuesday

KYMBERLEE PRICE

BUGCROWD

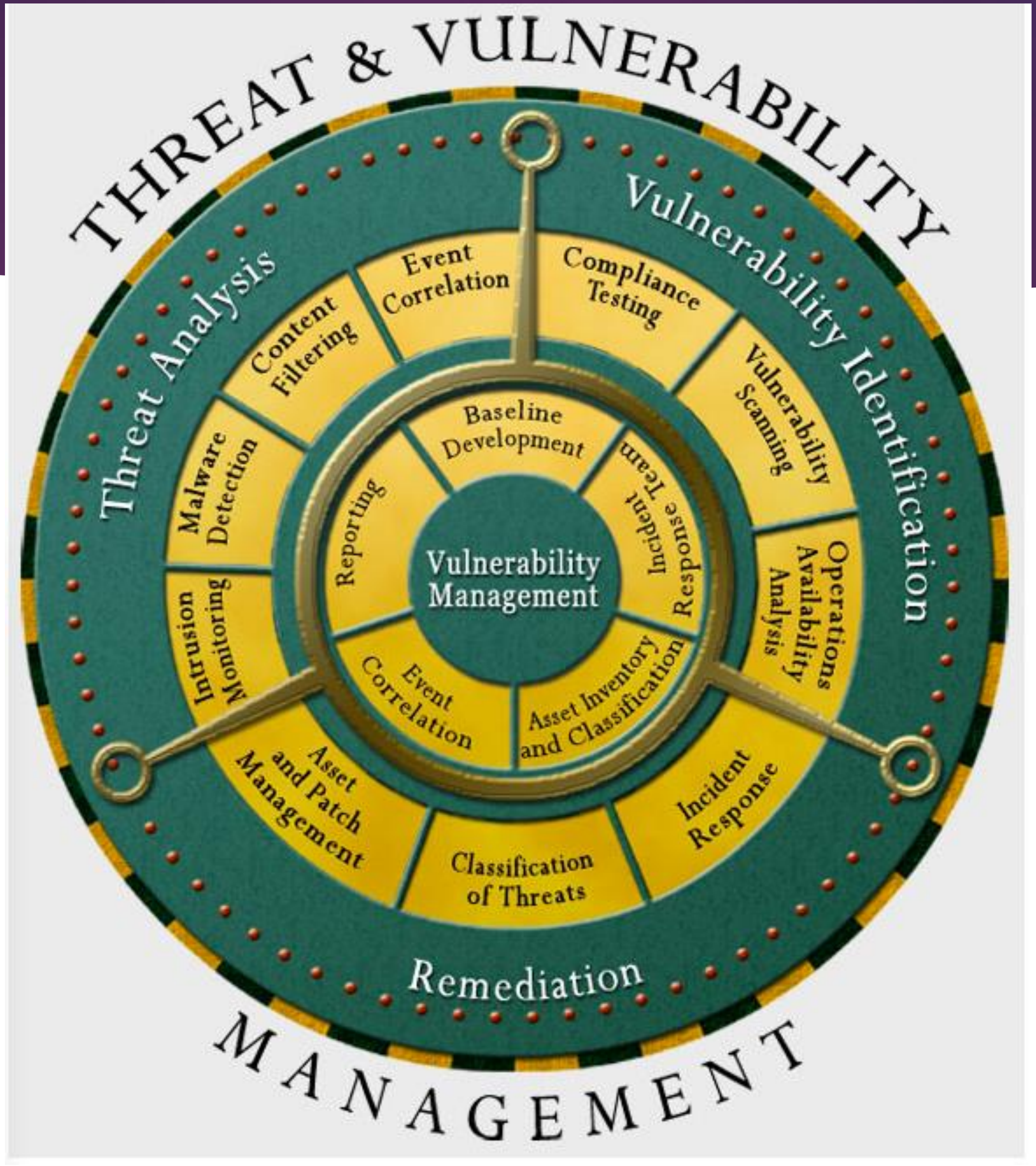
whoami?

- Senior Director of a Red Team
- PSIRT Case Manager
- Data Analyst
- Internet Crime Investigator
- Security Evangelist
- Behavioral Psychologist

@kym_possible

Vulnerability Management

- **Vulnerability management** is the "cyclical practice of identifying, classifying, remediating, and mitigating vulnerabilities", especially in software and firmware. Vulnerability management is integral to computer security and network security.
- Vulnerabilities can be discovered with a vulnerability scanner, which analyzes a computer system in search of known vulnerabilities, such as open ports, insecure software configuration, and susceptibility to malware. Unknown vulnerabilities, such as a zero-day attack may be found with fuzz testing, which can identify certain kinds of vulnerabilities, such as a buffer overflow exploit with relevant test cases. Such analyses can be facilitated by test automation. In addition, antivirus software capable of heuristic analysis may discover undocumented malware if it finds software behaving suspiciously (such as attempting to overwrite a system file).
- Correcting vulnerabilities may variously involve the installation of a patch, a change in network security policy, reconfiguration of software (such as a firewall), or educating users about social engineering.

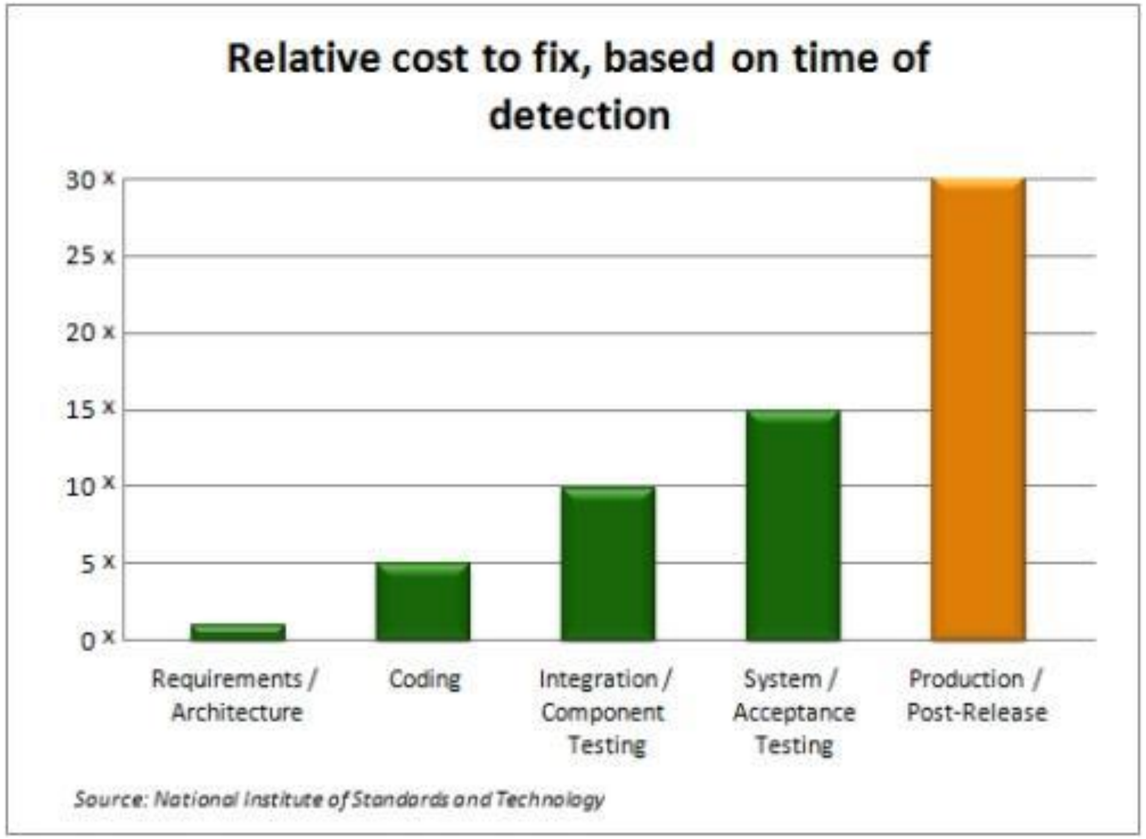

THREAT & VULNERABILITY
MANAGEMENT
Threat Analysis
Vulnerability Identification
Remediation
Event Correlation
Compliance Testing
Vulnerability Scanning
Operations Availability Analysis
Incident Response
Classification of Threats
Asset and Patch Management
Intrusion Monitoring
Malware Detection
Content Filtering
Baseline Development
Incident Response Team
Asset Inventory and Classification
Event Correlation
Reporting
Vulnerability Management

So Vuln Mgmt is A NetSec Issue!

Cost to Fix Vulnerabilities

The National Institute of Standards and Technology (NIST) estimates that code fixes performed after release can result in 25+ times the cost of fixes performed during the design phase.

tl;dr: Pay me now or pay me later… with interest.



“Fix vulnerabilities as early as is practical, resulting in fewer vulnerabilities to patch at the most expensive time - late in the development cycle.”

The Goal of Vulnerability Management

Easy, right?

Security versus…

99 LITTLE BUGS IN THE CODE, 99 LITTLE BUGS
FIX ONE BUG, RUN IT AGAIN, 117 LITTLE BUGS IN THE CODE!
memegenerator.net

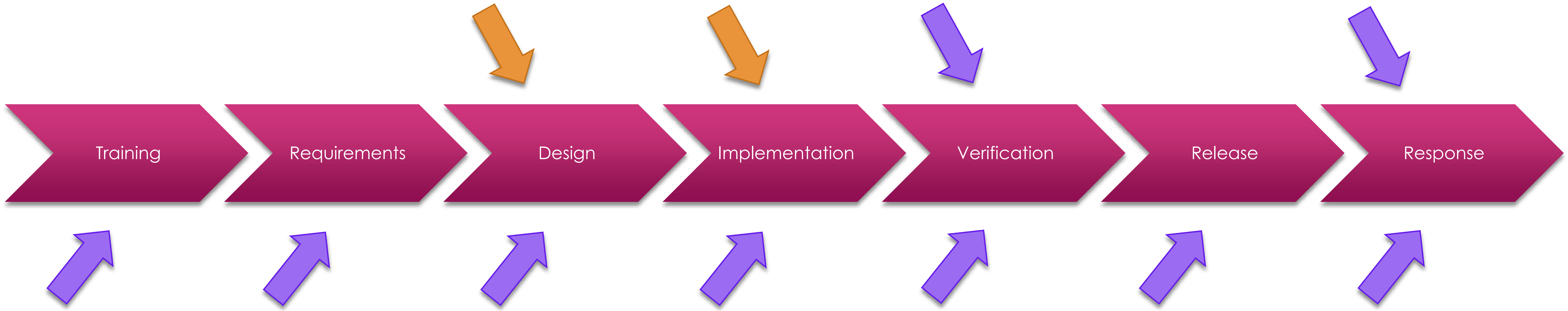
Performance

Usability

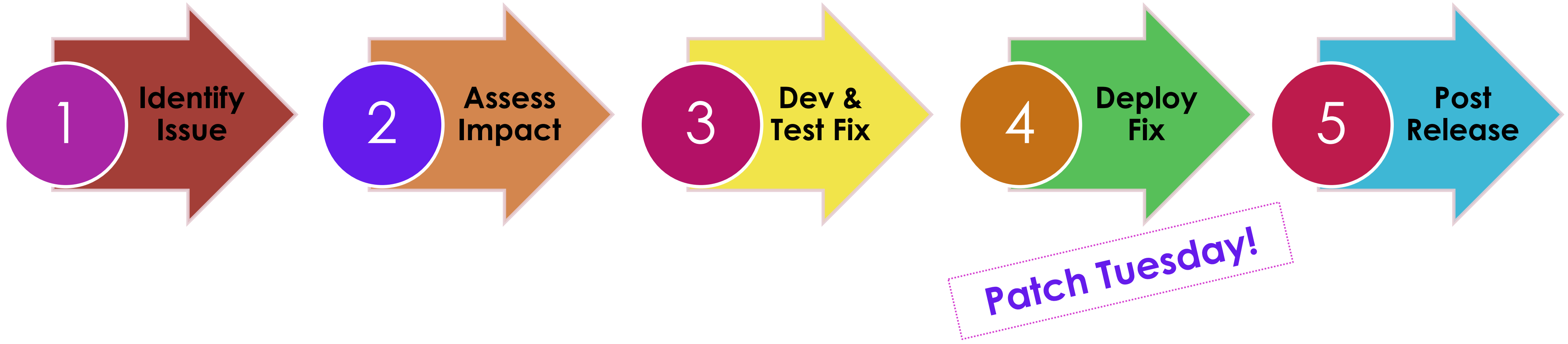
Functionality

Development cost & time

Secure Development Lifecycle



Vulnerability Management Process



Identify Issues

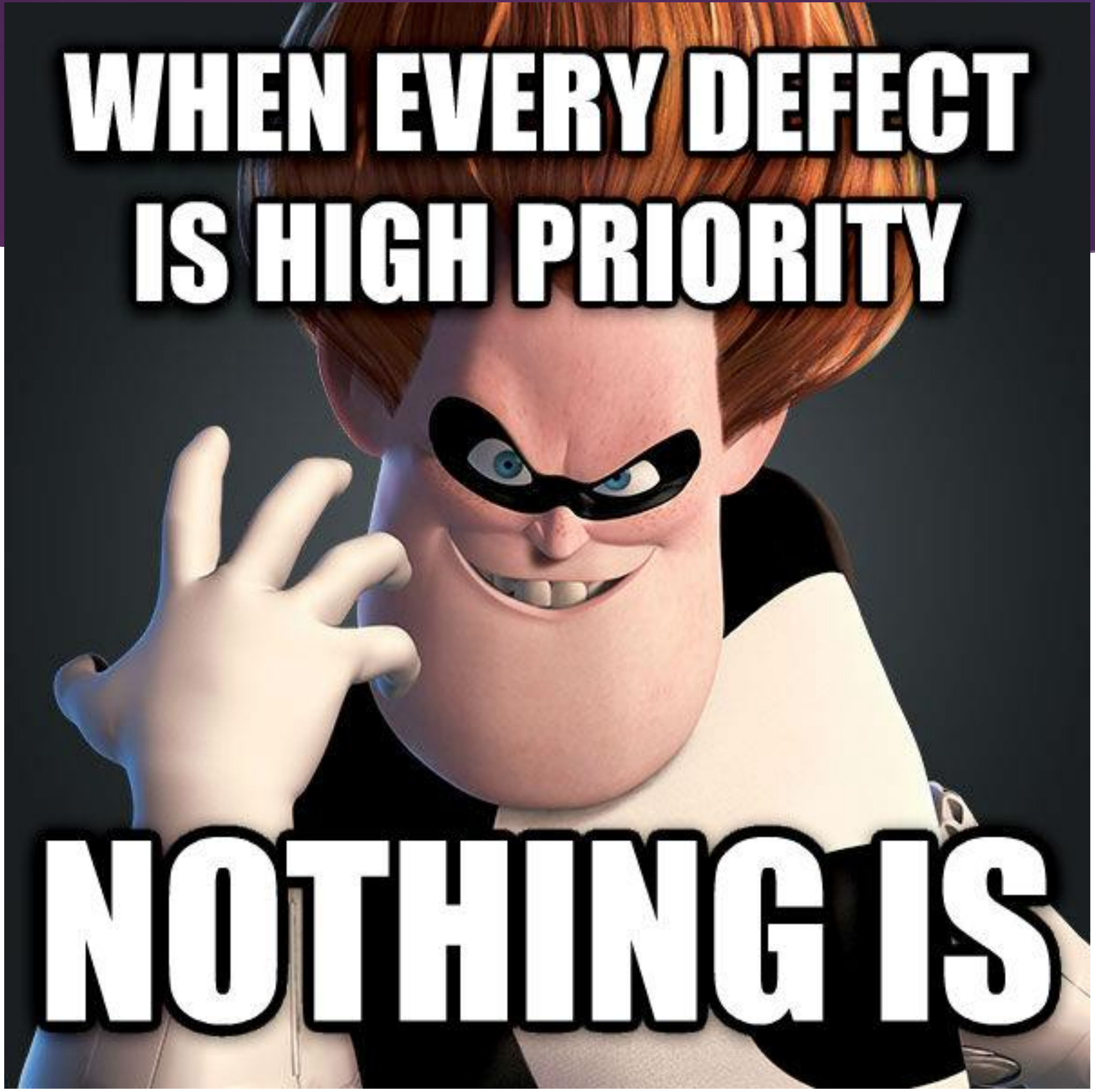
- Internal Security Research Team, Consultants – pre-release vuln assessments
- External Security Researchers – post release incident response, bug bounties
- Third Party Libraries/OSS Disclosures – both pre and post release
- Automated Tools & Analysis
- Crash log analysis

- Vulnerability Management
- Lots of vulnerabilities to manage

Assess Impact: Prioritization Matters

2 **Assess Impact**

- You have 150 vulnerabilities open with CVSS 7.5 and above
- Your inbound new vulnerabilities average 15 dev tasks per week, from both internal and external sources
- What do you fix first?
  - Highest CVSS Score?
  - FIFO? LIFO?
  - Externally known issues?
  - Issues with Exploit Presence in Metasploit?
- **Intelligent prioritization reduces risk**


WHEN EVERY DEFECT IS HIGH PRIORITY
NOTHING IS

Dev & Test Fix

Now lets go write some $\overset{\text{secure}}{\wedge}$ code!

3 Dev & Test Fix

- "Just ship it, we can patch that later" is not cost effective, but becomes more likely the closer you get to release dates
- Vulnerabilities are inevitable. Choose those that you fix pre-release and those you postpone to post-release carefully.
- Don't put off fixing the complicated vulnerabilities – they won't get easier once the product is in customer hands
- Sustainment planning is not just for post-release – you will have to patch vulnerabilities in perfectly functional code before RTM

The Numbers: 2007-2014

Library	Vuln Count	Vulns Per Year	Releases Per Year	Average CVSS
OpenSSL Cryptography and SSL/TLS Toolkit	71	10	4-5	5.49
the FreeType Project	52	8	2	8.02
libpng	28	4	3	6.65
Apache Tomcat	97	14	6	4.99
	387	55	9	8.52
Java *2010-2014	481	120	4-5	7.36

Take Aways

Vulnerability Management in SDL

Requirements

- Define guiding Security principles
- Define prioritization model and sustainment plan

Design

- Design for security and reduce attack surface
- Evaluate vuln trends in libraries as part of selection criteria

Implementation

- Automated static analysis tools
- Deprecate unsafe functions
- Code scanning tools to monitor all third party libraries – know what you use and where

Verification

- Automated static and dynamic analysis tools, fuzzing
- Manual pen testing & attack surface review
- Update 3rd party libraries regularly

How do you report on Vuln Mgmt?

- Analysis of vulnerability trends to predict future workload
- How many vulnerabilities are identified per month?
- What are their sources?
- What are the vulnerability types? Is dev training indicated?
- How many have been fixed, and how quickly is the backlog growing (or shrinking)?
- What is your Time To Fix?
- Success (or not) of SLA's based on resolution of vulnerabilities

Network Admins

- Ask potential software vendors about their SDL program and vulnerability trends
- Monitor the third party libraries being used in software you deploy and press vendors for security fixes

- Make it clear security is a priority

Discussion

Kymberlee Price
@kym_possible

Senior Director of Researcher Operations
Bugcrowd