

Software Security Assurance for DevOps

Mike Pittenger, VP Security Strategy, Black Duck Software
Lucas v. Stockhausen, Sr. Product Manager HPE Fortify



Agenda

- Challenges impacting application security in DevOps
- Strategies for overcoming these challenges
- 5 Things you can do tomorrow

Why We Partnered

- Organizations today manage application security for both custom and open source code
- HPE Security Fortify is a market leader in the application security space for customer code; Black Duck is a market leader in the application security space for open source
- Together, we allow customers to manage security risk in custom and open source code, through a single interface

Application Security Challenges

GROWING ATTACK SURFACE



Web, Mobile, Cloud, IoT

- Which apps are people using?
- How do I set internal policy requirements for app security?
- Is my private / sensitive data exposed by apps?
- Who is developing the apps?

NEW DEPLOYMENT MODELS



Containers, IT and Small Security Teams

- How do we prioritize the work for the resources I have?
- What do we test and how do we test it?
- How do we staff and improve skills and awareness?

OPEN SOURCE



Increasing Portion of Code Base

- What policies are in place for open source use?
- How are those policies enforced?
- Who is tracking usage for new vulnerabilities

Changing Attack Surface

- Web applications
- Cloud applications and services
- IoT

IoT Units Installed Base by Category (Millions of Units)

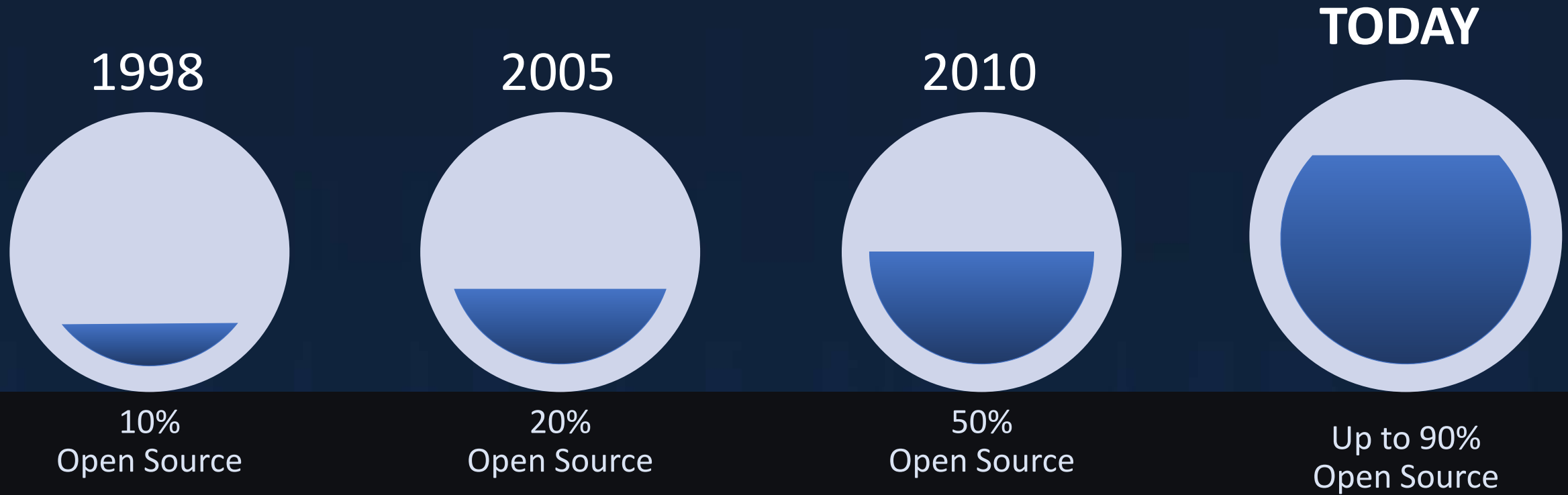
Category	2014	2015	2016	2020
Consumer	2,277	3,023	4,024	13,509
Business: Cross-Industry	632	815	1,092	4,408
Business: Vertical-Specific	898	1,065	1,276	2,880
Grand Total	3,807	4,902	6,392	20,797

Source: Gartner (November 2015)

“If perimeter control is to remain the paradigm of cybersecurity, then the number of perimeters to defend in the Internet of Things is doubling every 17 months.”

Dan Geer
In-Q-Tel
RSA 2016

Open Source is the Foundation of Modern Applications



Open source Use has Outpaced Process Maturity

Everybody is using open source, but many organizations still do not have adequate processes or tools in place to manage it.

GROWING OPPORTUNITY FOR POLICIES & PROCEDURES

@FUTUREOFOSS #FUTUREOSS



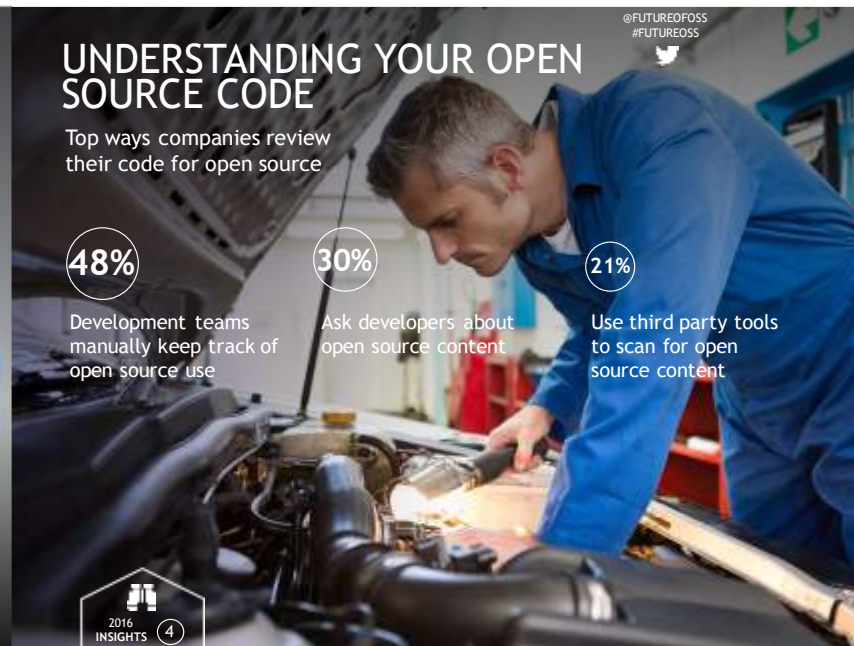
RULES
Nearly
50%

2016 INSIGHTS 4

UNDERSTANDING YOUR OPEN SOURCE CODE

@FUTUREOFOSS #FUTUREOSS

Top ways companies review their code for open source



48% Development teams manually keep track of open source use

30% Ask developers about open source content

21% Use third party tools to scan for open source content

2016 INSIGHTS 4

HOW ARE COMPANIES HANDLING KNOWN OPEN SOURCE VULNERABILITIES?

@FUTUREOFOSS #FUTUREOSS



Nearly
1/3
of companies have no process for identifying, tracking or remediating known open source vulnerabilities

2016 INSIGHTS 4

How Well Do Manual Processes Work?

2017 Open Source Security and Risk Analysis

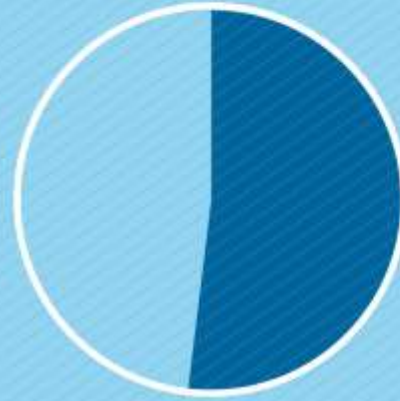
67%

Applications that contained known open source vulnerabilities



52%

Known open source vulnerabilities in each application that were rated as 'severe'



7%

Applications that still contained Heartbleed, Poodle, Freak, or Drown vulnerabilities



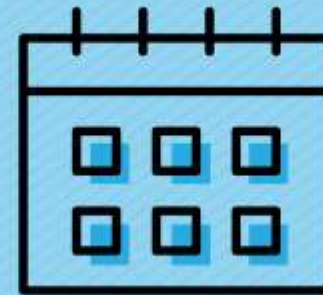
147

Average number of unique open source components per application



27

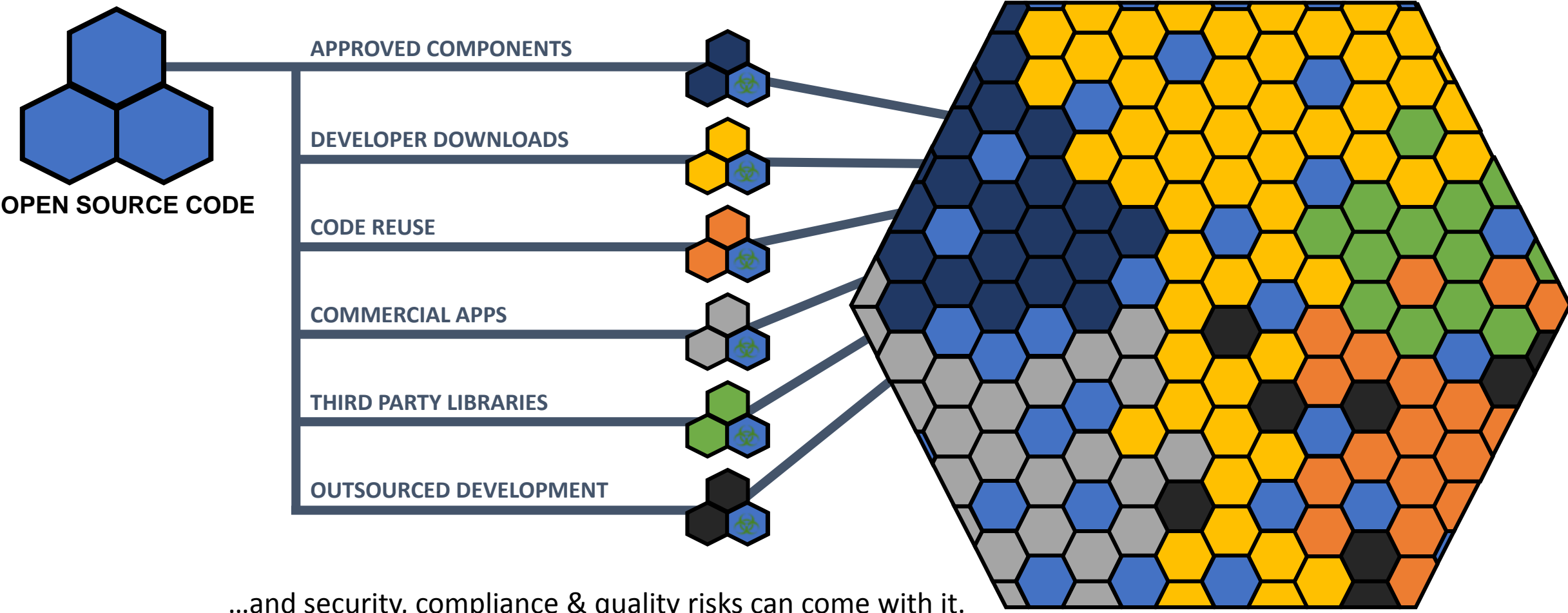
Average number of known open source vulnerabilities in per application



4 YEARS

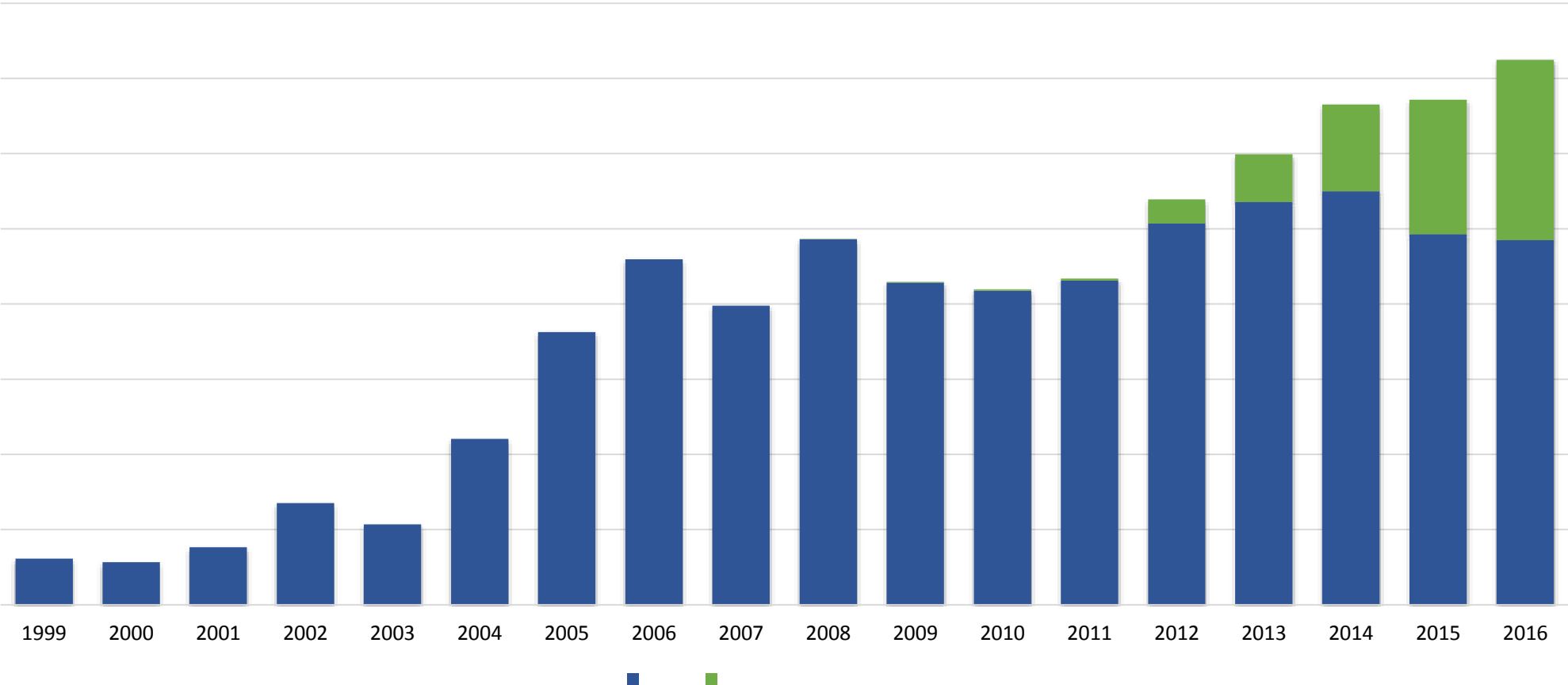
Average age of vulnerabilities found in each application

Open Source Enters Your Code in Many Ways...



...and security, compliance & quality risks can come with it.

Open Source Vulnerabilities are Increasing



Reference: Black Duck Software knowledgebase, NVD, VulnDB



FREAK!
SSL, TLS Vulnerability



Static Analysis Does Not Help With Open Source

- Automated testing finds common vulnerabilities in the code you write
 - They are good, not perfect
 - Different tools work better on different classes of bugs
 - Many types of bugs are undetectable except by trained security researchers

All possible
security vulnerabilities



FREAK!

Four Factors That Make Open Source Different

Used Everywhere



Easy access to code



Vulnerabilities are public



Exploits readily available



Who's Responsible for Open Source Security?

Commercial Code



The screenshot shows a .NET Blog post from May 2015. The title is "May 2015 .NET Security Updates". The author is "The .NET Fundamentals Team" and the date is "12 May 2015 10:00 AM". The post content includes a "Microsoft Security Bulletin MS15-044 - Critical, Vulnerability in .NET Framework Could Allow Remote Code Execution (8057110)". It describes a vulnerability in the .NET Framework that could allow remote code execution. The post is rated "Critical" and mentions affected versions of the .NET Framework and Microsoft Windows.

- Dedicated security researchers
- Alerting and notification infrastructure
- Regular patch updates

Dedicated support team with SLA

Open Source Code



The screenshot shows an email announcement from MediaWiki. The subject is "[MediaWiki-announce] MediaWiki Security and Maintenance Releases: 1.25.2, 1.24.3, 1.23.10". The sender is "Chad innocentkiller@gmail.com" and the date is "Mon Aug 10 21:54:44 UTC 2015". The email content includes a "Security Fixes" section and mentions the release of MediaWiki 1.25.2, 1.24.3, and 1.23.10. It states that these releases fix three security issues in core, in addition to other bug fixes. Several extensions have also had security issues fixed. Download links are given at the end of the email. A link to a Phabricator ticket is provided: <https://phabricator.wikimedia.org/T106803>.

- “Community”-based code analysis
- Monitor newsfeeds yourself
- No standard patching mechanism

Ultimately, you are responsible

Bad Guys Have Quotas Too (Non-Targeted Attacks)

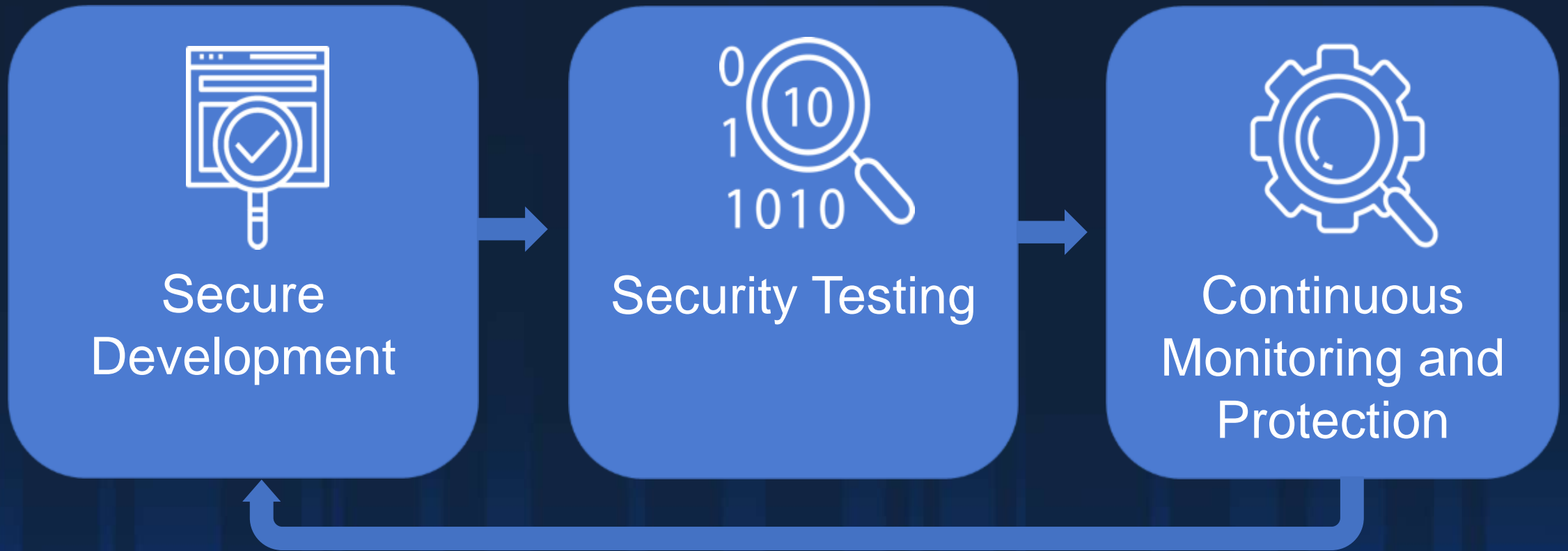
Rational Choice Theory

- Criminals make a conscious, rational choice to commit crimes
- Behavior is a personal choice made after weighing costs and benefits of available alternatives
- The path of least resistance will be taken



Integrating Application Security in DevOps

Application Security Testing for the New SDLC



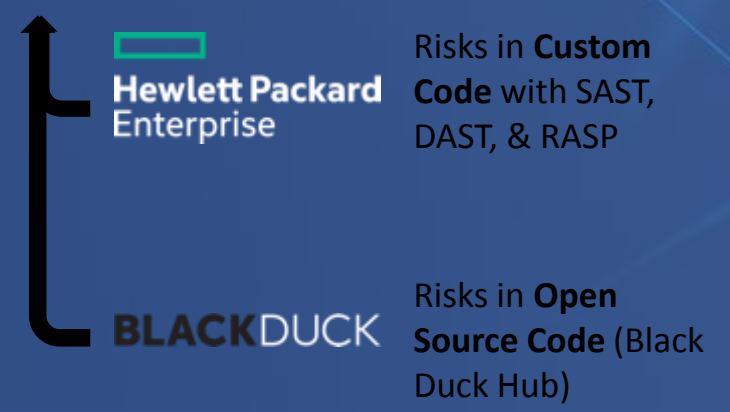


HPE Security Fortify and Black Duck Integration

Black Duck Integration with HPE Security Fortify SSC

- HPE Security Fortify + Black Duck Technology Alliance Partnership
- Address pervasive, rapidly-growing Security & Compliance risks with Open Source
- Gain visibility on risks across Custom Code and Open Source Code
- Integrate governance and remediation as part of Software Security Assurance

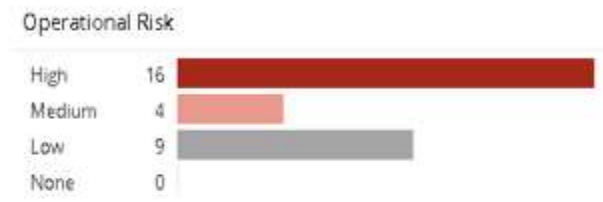
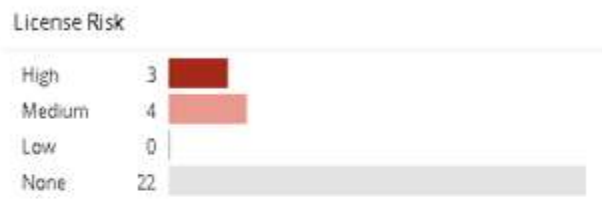
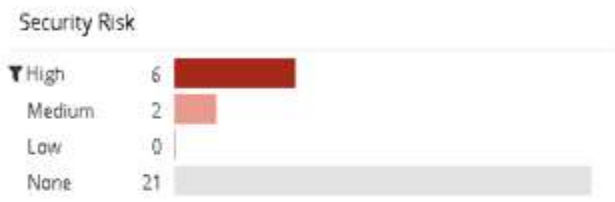
Manage Risks in Open Source as part of HPE Security Fortify SSC



Open Source Vulnerabilities – Black Duck

📦 Hub Internal Projects
Bill Payment Processor ▸ 1.0
unknown Versions: 2 | Phase: In Planning | Distribution: External

☰ Components 🛡️ Security 📁 Files 📊 Reports ⚙️ Settings



[+ Add Component](#)

Security risk: High ✕
▾ Filter components...
[Add Filter ▾](#)

Component	Match Count	Match Type	Usage	License	Security Risk	Operational Risk
⊗ Apache Commons BeanUtils 1.8.3	📄 1 Match	Exact	Dynamically Linked	Apache-2.0	1	High
⊗ Apache Tomcat 7.0.55	📄 1 Match	Exact	Dynamically Linked	Apache-2.0	1 8	Medium
⊗ libcurl3-gnutls 7.19.4	📄 10 Matches	Files Modified	Dynamically Linked	curl License	1 11	High
⊗ libxml2 2.9.1+dfsg1	📄 2 Matches	Exact	Dynamically Linked	libxml2 License and 1 more...	5 25 1	Low
⊗ Net-SNMP 5.7.1	📄 16 Matches	Exact	Dynamically Linked	BSD-3-Clause	1 3 1	High
⊗ OpenSSL 1.0.2a	📄 1 Match	Exact	Dynamically Linked	M OpenSSL and 1 more...	8 17 4	Low

Displaying 1-6 of 6

Overview Shows Black Duck Results Within HPE Security Fortify

Hewlett Packard Enterprise Dashboard

Bill Payment Processor 1.1 Audit

Group by Analysis Type Filter by Select attributes

0 of 1109 issues selected

Issue Name	Primary Location	Analysis Type	Criticality	Tagged
3rd Party Component	Apache Commons Codec:1.6:1.0	BLACK DUCK SOFTWARE	Critical	
3rd Party Component	Apache Commons Collections:3.2.1:1.0	BLACK DUCK SOFTWARE	Critical	
3rd Party Component	Apache Commons Collections:3.2.1:1.0	BLACK DUCK SOFTWARE	Critical	
3rd Party Component	Apache Santuario:1.4.2:1.0	BLACK DUCK SOFTWARE	Critical	

BLACK DUCK SOFTWARE - (0 / 4)

Issue Name	Primary Location	Analysis Type	Criticality	Tagged
Poor Error Handling: Overly Broad Catch	AbstractLesson.java:420	SCA	Low	Reliability Issue
System Information Leak	AbstractLesson.java:422	SCA	Low	
Poor Logging Practice: Use of a System Output Stream	AbstractLesson.java:422	SCA	Low	Bad Practice
System Information Leak	AbstractLesson.java:423	SCA	Low	

SCA - (626 / 1105)

Open Source vulnerabilities (3rd Party Components) from Black Duck analysis

Custom Code vulnerabilities from Fortify SCA analysis

Detailed View of Black Duck Results Within HPE Security Fortify

3rd Party Component Apache Commons FileUpload:1.2.1 BLACKDUCK High

BLACKDUCK Comments & History Attachments

▼ Vulnerability description (modified)

The MultipartStream class in Apache Commons Fileupload before 1.3.2, as used in Apache Tomcat 7.x before 7.0.70, 8.x before 8.0.36, 8.5.x before 8.5.3, and 9.x before 9.0.0.M7 and other products, allows remote attackers to cause a denial of service (CPU consumption) via a long boundary string.

Vulnerability Id (CVE number)	CVE-2016-3092
Vulnerability source	NVD
Published on	08/24/2016
Updated on	11/28/2016
Analysis	Not Set

Details

Project name	solrWar2
Project version	4.10.4
Component name	Apache Commons FileUpload
Component version	1.2.1
Exploitability	10
Vulnerability impact	6.9
CVSS score Base score	7.8
URL	http://web.nvd.nist.gov/view/vuln/detail?vulnid=CVE-2016-3092



Automating Security Testing in a DevOps Environment

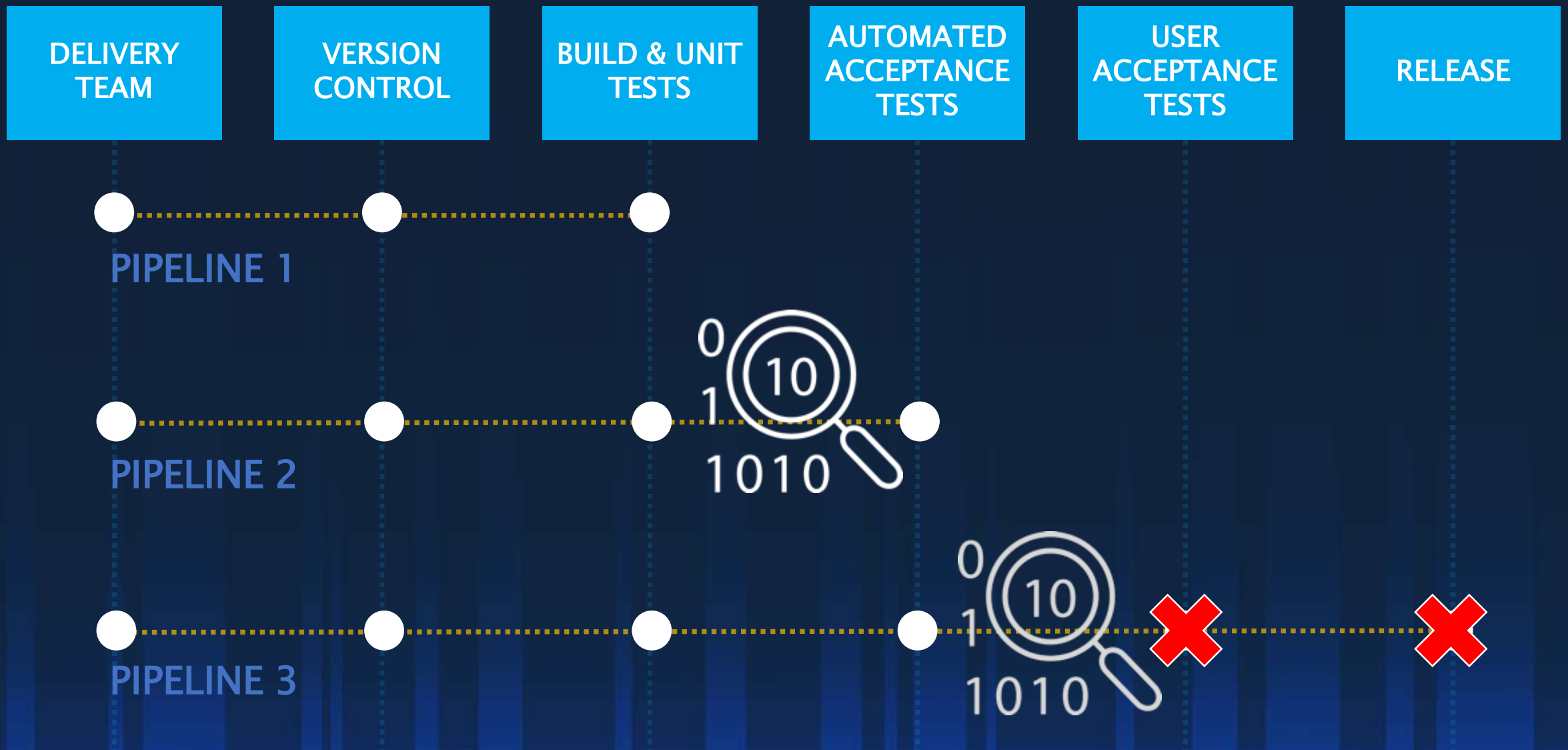
Continuous Integration Environment



Continuous Integration Environment



Automation Differs Between Apps



What Can You Do Tomorrow?

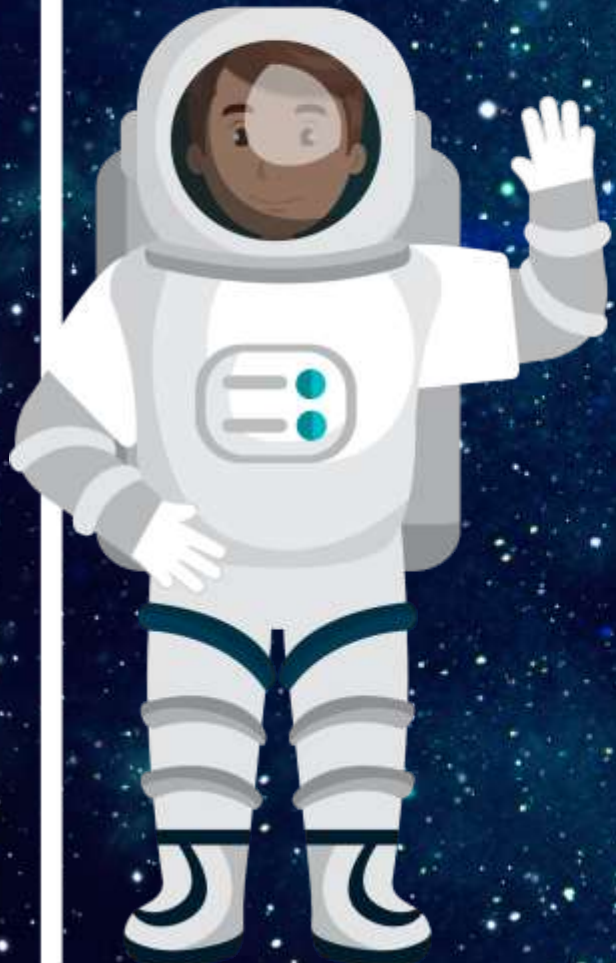
- Speak with your heads of application security and software development and find out...
 - What policies exist for managing open source?
 - Is there a list of components used in all applications?
 - How are they creating the list?
 - What controls do they have to ensure nothing gets through?
 - How are they tracking vulnerabilities for all components over time?
 - How do they account for the different testing requirements for custom code v. open source?
 - What is the best security automation strategy for your organization?



Questions

Contact: hpe@blackducksoftware.com

Visit: <http://www.blackducksoftware.com/hpe>



BLACKDUCK 2017 FLIGHT

November 7-9, 2017 | Boston, MA

Join us in Boston

REGISTER NOW

Or Click Here: <http://blkduk.io/2vnikhH>