MICRO FOCUS

Application Security in an Ever Changing Digital Landscape

Trace Hollifield
Security Architect
CISSP, GSLC, CISM, C|CISO, ITIL
Trace.Hollifield@microfocus.com

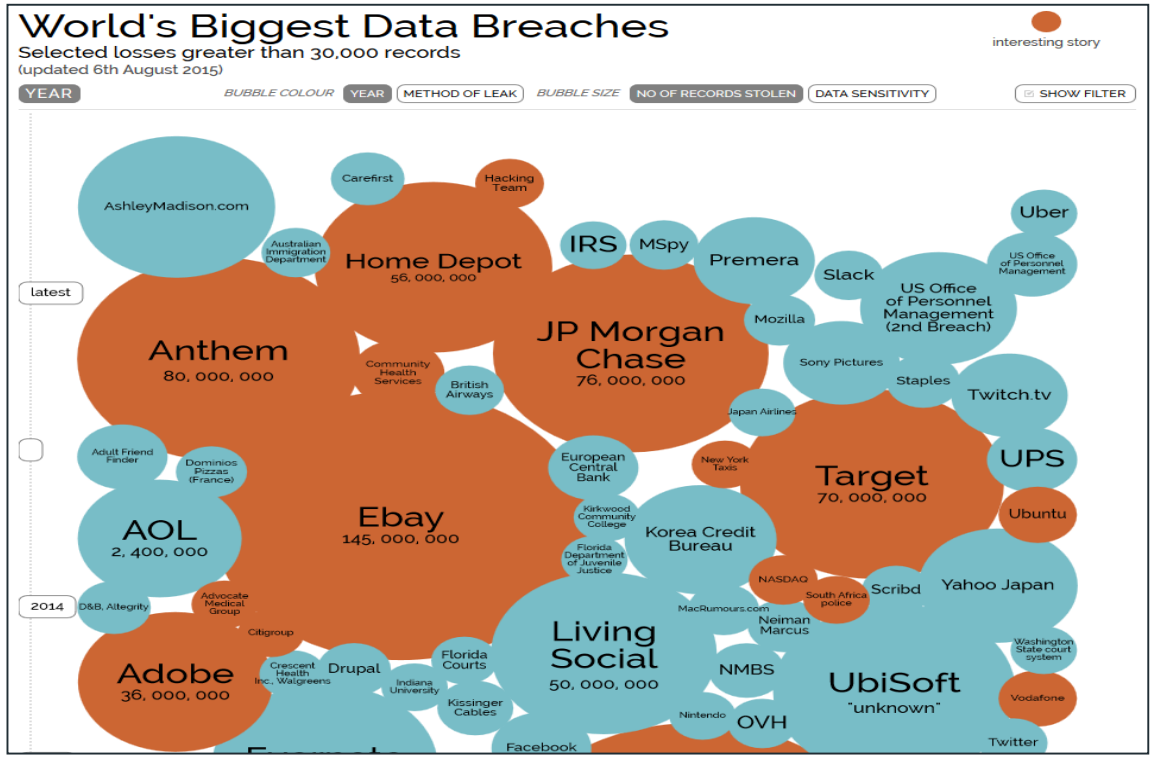
Agenda

- Security Landscape
- Specific Application Security (App Sec) Challenges
- Establishing an AppSec Program
- Why AppSec is Hard
- Software Security Research Results
- Conclusion

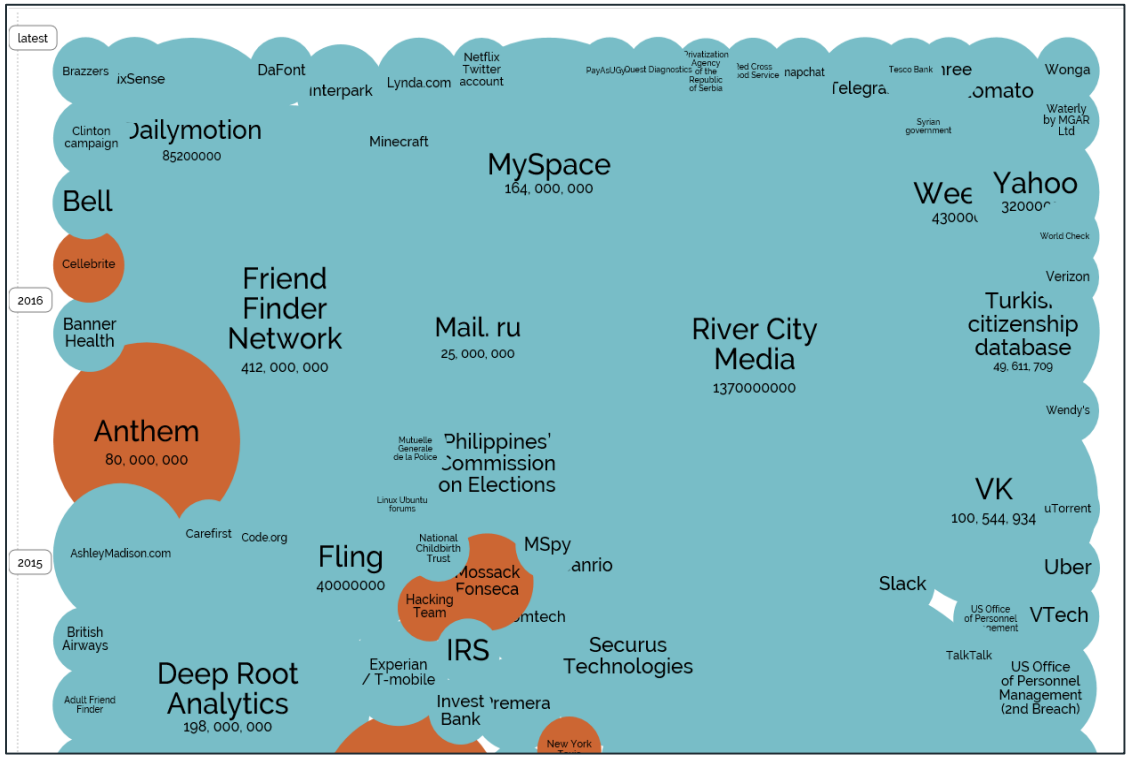
Security Challenges

Breaches Around the world

July 2015



July 2017



145 million
people affected

77 days
time to detect

500 million
people affected

600 days
time to detect

148 million
people affected

229 days
time to detect

http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/

Homeland Security

Software is essential to the operation of the Nation's critical infrastructure. Software vulnerabilities can jeopardize intellectual property, consumer trust, and business operations and services. In addition, a broad spectrum of critical applications and infrastructure, from process control systems to commercial application products, depends on secure, reliable software.

The Software Engineering Institute estimates that 90 percent of reported security incidents result from exploits against defects in the design or code of software. Ensuring software integrity is key to protecting the infrastructure from threats and vulnerabilities and reducing overall risk to cyber attacks. To ensure system reliability, integrity, and safety, it is critical that provisions be included for built-in security of the enabling software.

"90 percent of reported security incidents result from exploits against defects in the software."

Software Engineering Institute

Carnegie Mellon

Got Vulnerabilities ?

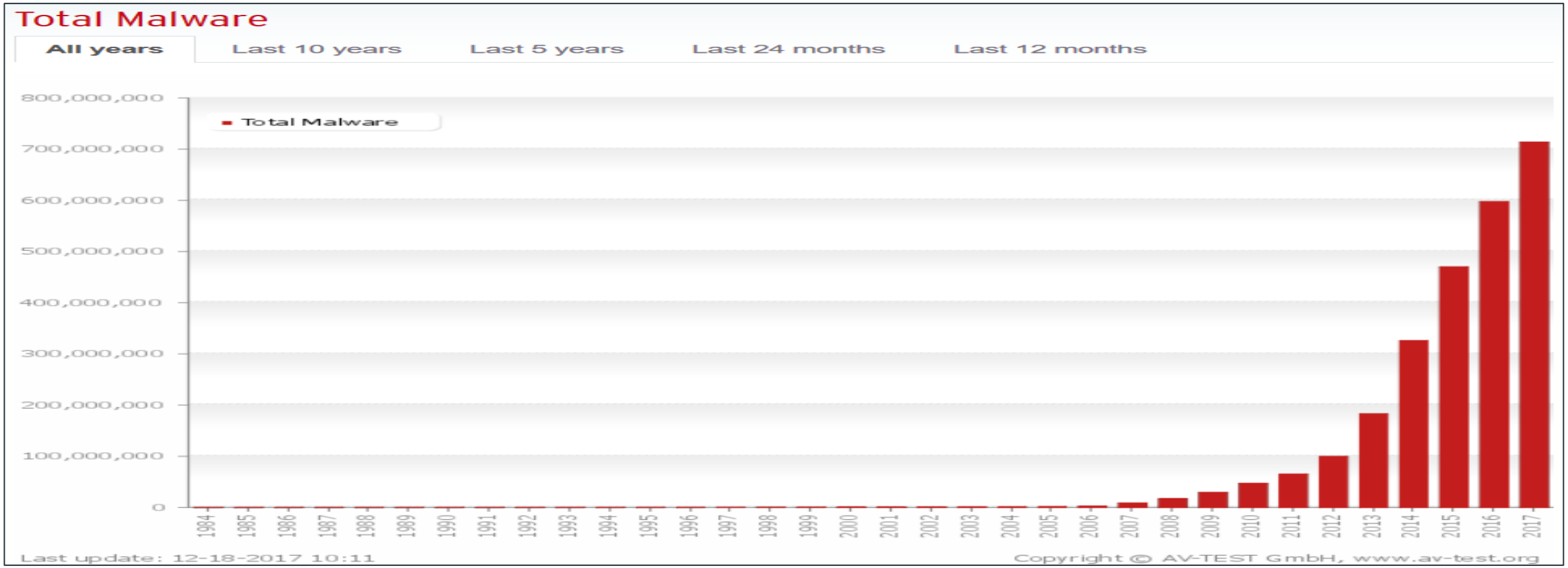
Since 1999, over 147k vulnerabilities have been registered

41% of all vulnerabilities are from recent years

7% are from 2019 (Only 63 days in)

https://cve.mitre.org/data/downloads/index.html

Malware

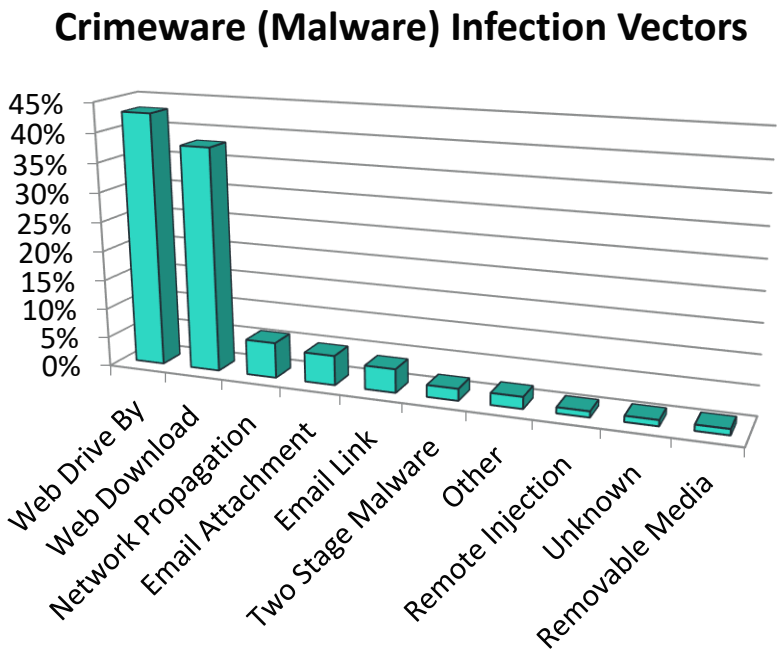


- Malware continues to grow and morph as new vulnerabilities are discovered and old vulnerabilities are not remediated.

- ~50 Million variants in the last 60 months

https://www.av-test.org/en/statistics/malware/

Malware – Attack Vectors and Cost



Malware Cost

Trojans :

- Keylogger Detective 2.3.2 (w/ hidden installation). Price: **US $3**
- Web Browser Based (Opera, Mozilla Firefox, Chrome, Safari, etc). Price: **US $8**
- Remote Access / Backdoor. Price: **US $25**
- Spider Keylogger Pro v. 1.2.4. Price: **US $50**

Ransomware:

- Winlocker Source Code. Price: **US $8**
- Winlocker Activiation Key. Price: **US $10-20**

Source – Trendmicro "Russian Underground 101"

Malware Highlights

- **Low Cost** … The cost of various malware is extremely low for the possible greater rewards
- **Accuracy by Volume** … The attacker only needs to be right once while defense has to be right all the time
- **Go Where the People Are** … Web attacks are the most popular vector to infect hosts due to people's habits, advertising conduits, and rapid changes in the environment which create challenges for data security defenders.

Ransomware Jackpot

FBI Received Over 2,600 Ransomware Complaints in 2016 Costing $1.3 B

- About 2673 complaints were submitted according to IC3's report. The number is just the tip of the iceberg, though, when compared to the 298,728 cybercrime-related complaints received overall last year, 2016. Losses connected to such cybercriminal activities is reported to be around $1.3 B.

WannaCry: Encrypts 176 file types, including database, multimedia, and archive files, as well as Microsoft Office documents.

Petya: Encrypts the system's files, overwrites its Master Boot Record, and locks users out with a blue screen of death.

Locky: Encrypts over 130 file types, including those on removable drives and unmapped network shares.

Hidden Tear: Is an open-source ransomware that allowed cybercriminals to create their own versions which were themselves reworked into more spinoffs. One variant can encrypt up to 2,783 file types.

Cerber: Sold as a ransomware as a service (RaaS), which means cybercriminals can customize its encryption behavior and ransom demands; it's been recently spotted to be capable of stealing from Bitcoin wallets and evading machine learning.

https://cointelegraph.com/news/fbi-received-over-2600-ransomware-complaints-in-2016-costing-13-bln

https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/2017-ransomware-recap

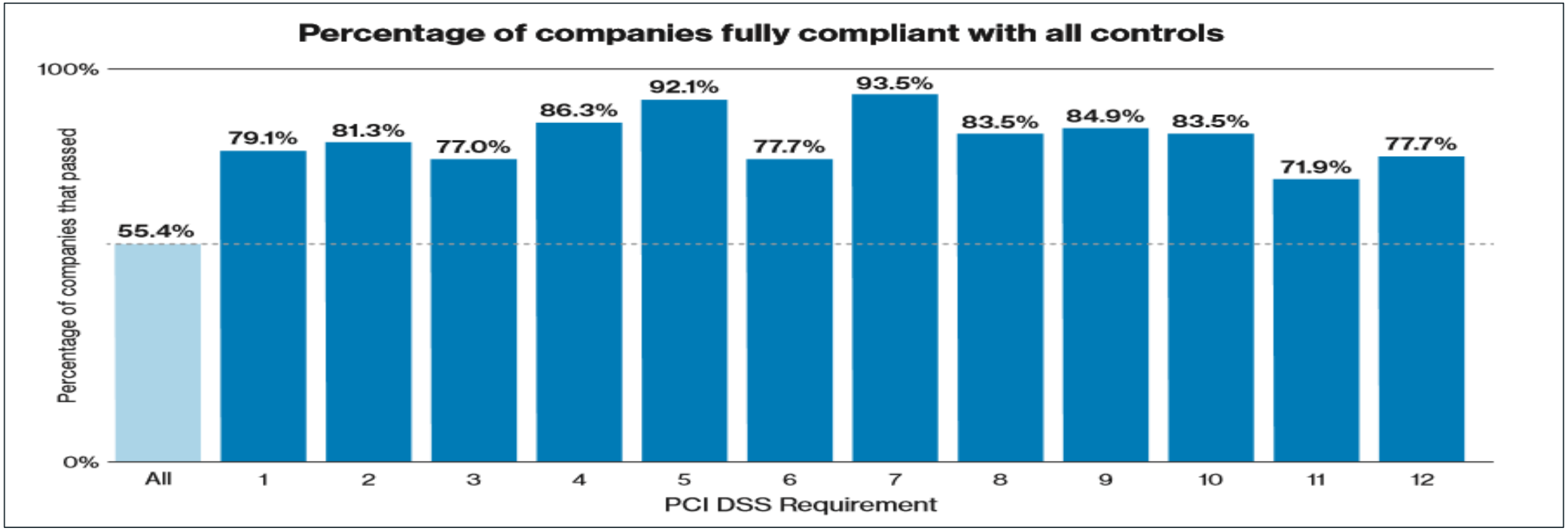
Data Security Standards / Frameworks

- Several to pick from: ISO 27000, PCI-DSS, NIST, COBIT, SANS 20 and more
- Purpose of the Frameworks: To provide guidance on how to protect sensitive data
- Payment Card Industry – Data Security Standards (PCI-DSS) started in 2004 with the following requirements:

Goals	Requirements
Build & Maintain a Secure Network	1. Install and maintain a firewall configuration to protect cardholder data 2. Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	3. Protect stored cardholder data 4. Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	5. Use and regularly update anti-virus software or programs 6. Develop and maintain secure systems and applications
Implement Strong Access Control Measures	7. Restrict access to cardholder data by business need to know 8. Assign a unique ID to each person with computer access 9. Restrict physical access to cardholder data
Regularly Monitor and Test Networks	10. Track and monitor all access to network resources and cardholder data 11. Regularly test security systems and processes
Maintain an Information Security Policy	12. Maintain a policy that addresses information security for all personnel

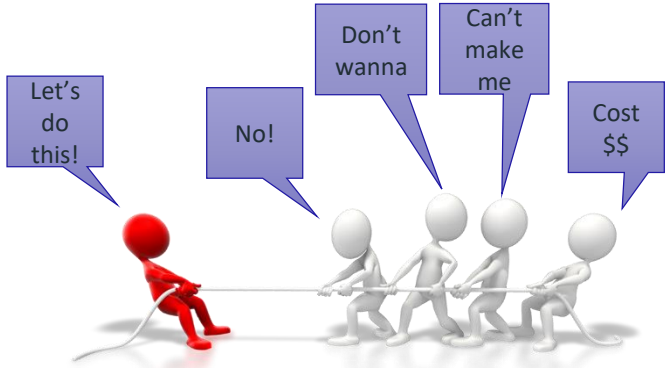
https://www.pcisecuritystandards.org/document_library

Being fully PCI Compliant…



Key Points

- Not 100% across the board
- Compensating controls (duct tape) fill in the blanks
- Qualified Security Assessors (QSAs) work with shared data and limited time engagements
- Compliance testing is only once a year



http://www.verizonenterprise.com/verizon-insights-lab/payment-security/2017/

But we are PCI compliant… Not So Much.

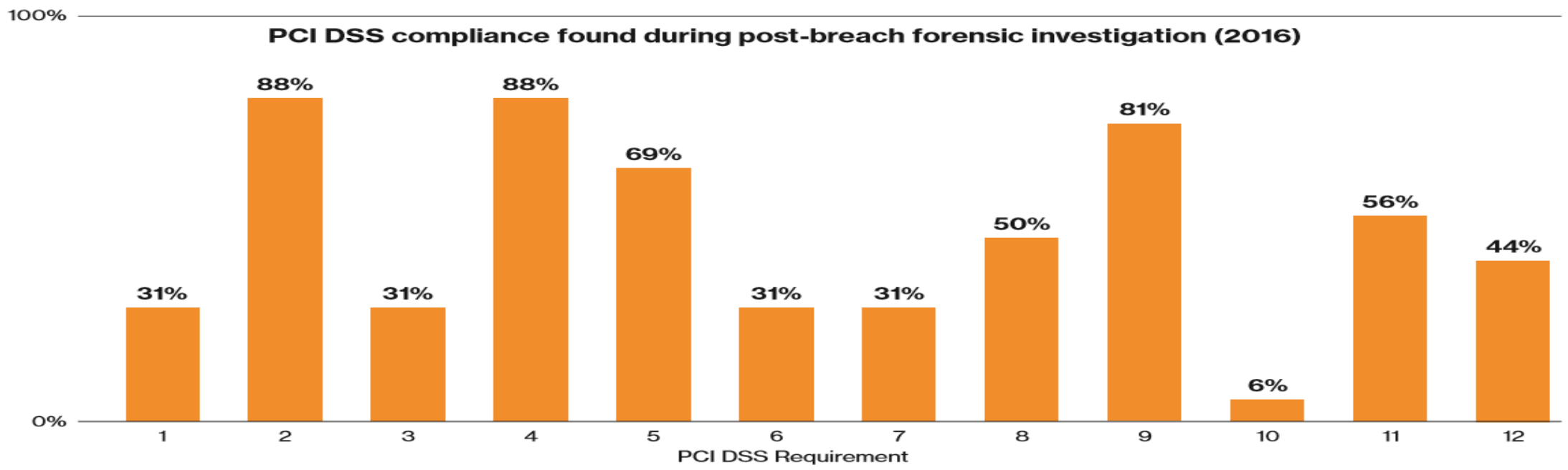


Fig 15. QSA versus PFI, 2016*

(PFI – PCI Forensic Investigator)

Large Misses

- Requirement 10: Logging and Log Review
- Requirement 1: Network Environment & Changes
- Requirement 3: Data at Rest Protections
- Requirement 6: Vulnerability Management & Patching
- Requirement 7: Need to Know Access & Permissions
- Requirement 12: Policies & Training

http://www.verizonenterprise.com/verizon-insights-lab/payment-security/2017/

Challenge Summary

Bad Actors – External

- Deep Pockets & Cheap Tools
- Creative & Innovative
- Successful at extorting funds from a business

Bad Actors – Internal

- Many verticals lack compliance maturity
- Many verticals struggle with existing compliance requirements let alone new and evolving requirements
- Lack of structured Policies & Processes combined with enforcement
- Confirmed immaturity for Logging and Log Review, Data at Rest Protections, **Vulnerability Management**, Need to Know Access & Permissions
- Governments are jumping in on the action with new compliance regulations and fines

Business Impact

- Reputation and Confidence Loss
- Lawsuits
- Compliance Fines
- Combination of the three results in a decline of revenue while increasing business expenses

Application Security Challenges

Tsunami of Apps

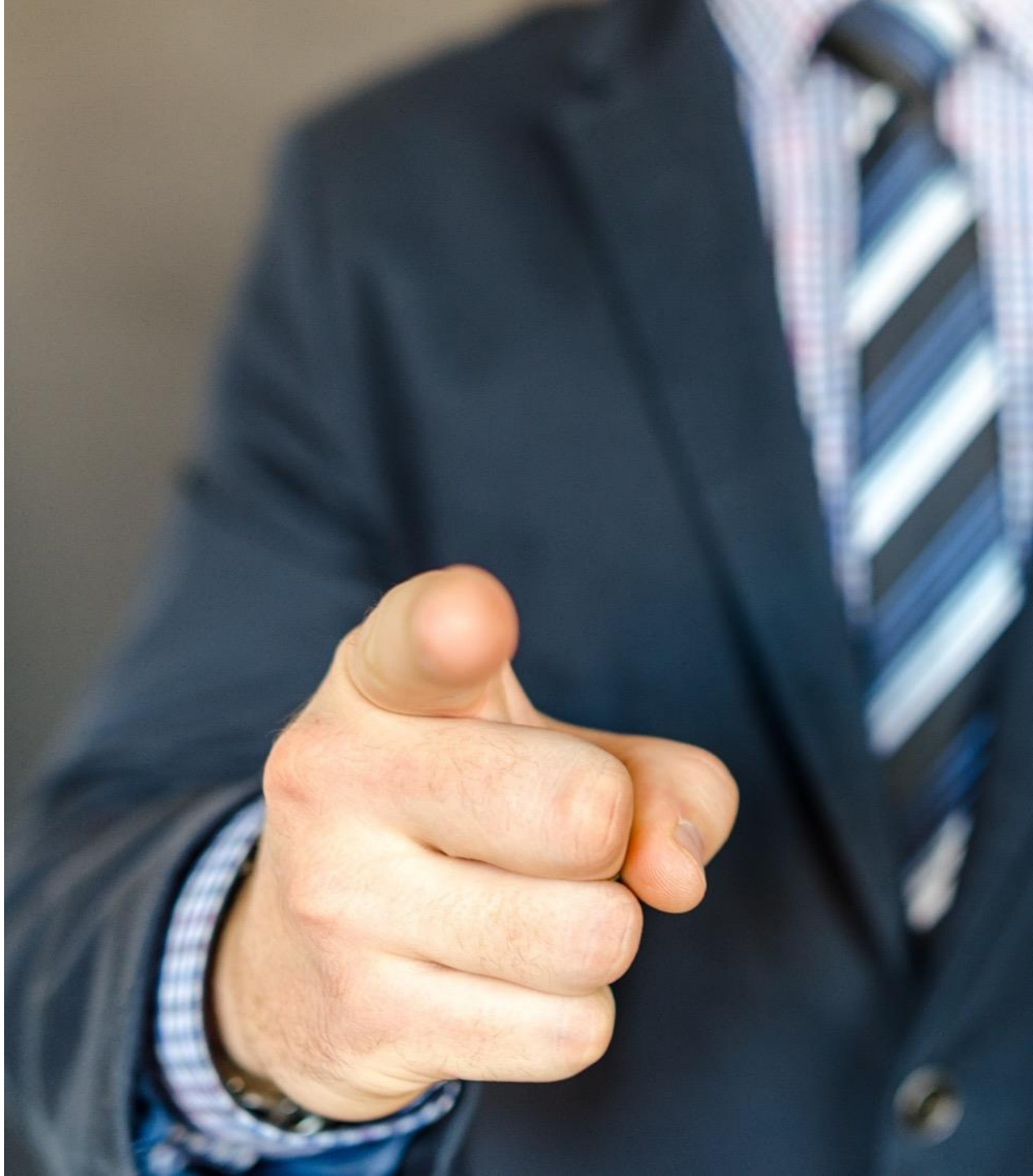
1000 applications and counting…

Speed vs Depth

“I want 5 minute scans with no false positives.”

Developer User Story

We have seen the AppSec team
AND IT IS YOU! (the developer)



Establishing an AppSec Program

Goals and benefits of an Application Security Program

The mitigation of application security risks is not a one time exercise; rather it is an ongoing activity that requires paying close attention to emerging threats and planning ahead for the deployment of new security measures to mitigate these new threats. This includes the planning for the adoption of new application security activities, processes, controls and training.

Source: "Application Security Guide for CISOs," OWASP, 2013

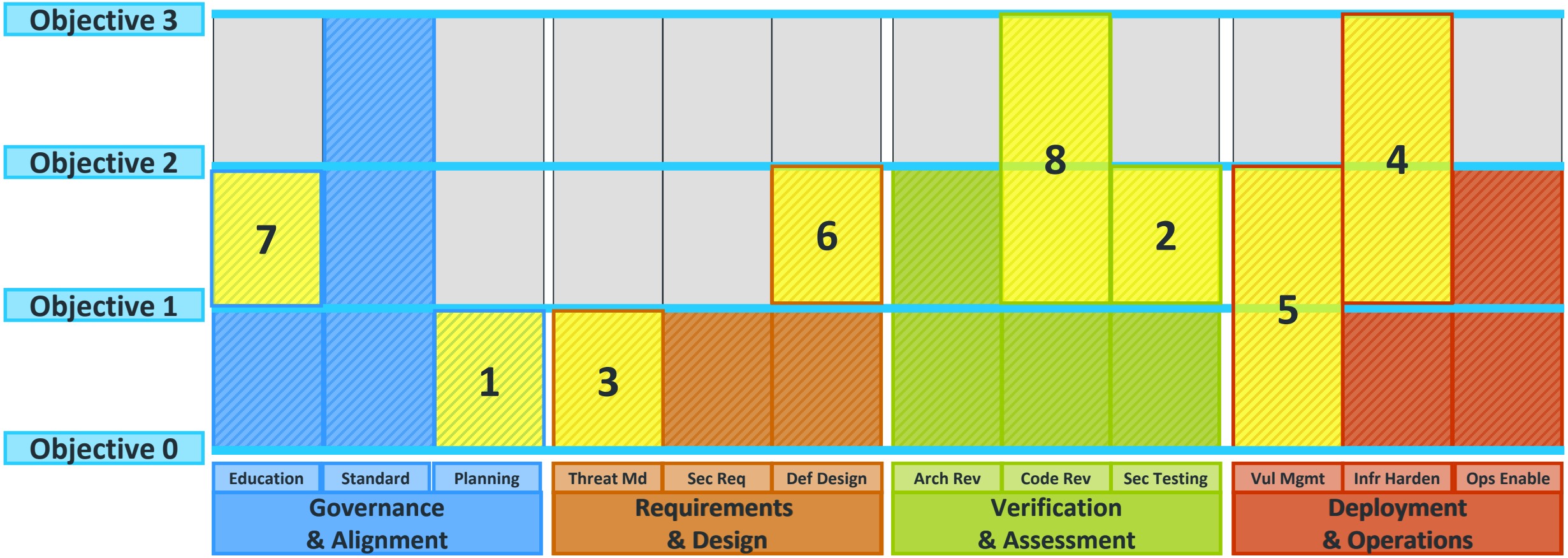
A successful applications security programs need to:

- Map security priorities to business priorities
- Assess the current state and target state using a security program maturity model
- Seamlessly integrate into development processes and tool chains

Evolution of Capability for Application Security

Baseline and periodic maturity assessments key

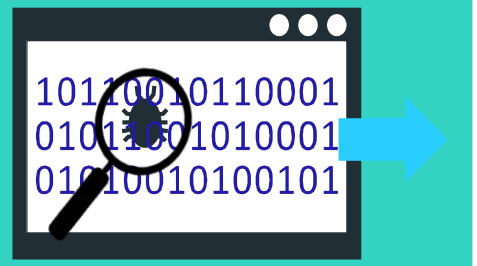
Blank Scorecard → Industry Best Practices → Enterprise Scoring → Prioritized Roadmap



Building Security Into the SDLC

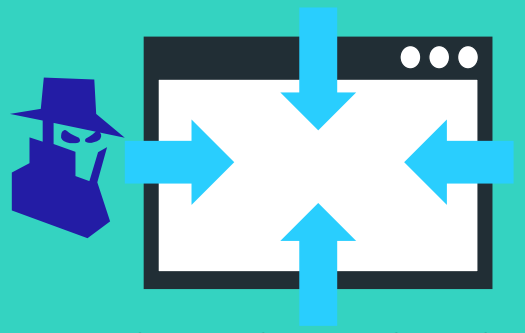
Value of automated and manual analysis

Static Analysis



- 100% code coverage
- Pinpoint and prioritize violations within code authoring environment
- Root cause of vulnerabilities with line-of-code detail
- Language specific remediation strategies

Dynamic Analysis



- Stimulate application through automated, external security attacks
- Identifies, crawls and attacks application attack surface
- QA or production environments
- Doesn’t require code
- Real world attack simulation

Manual Review

- Threat modeling
- Requirements verification
- Security Architecture Reviews
- Business logic verification
- Reduce false positives

Building an AppSec Program – Major Milestones

Project Planning

Solution Roadmap

SOC Process & Procedures

Initial Monitoring Capability

Assess / Design | Development | Implementation | Mature / Operate | Operate / Transition

Architecture design

Workflow Creation

Deliver KPIs and Metrics

Business Process Integration
Activate development and SDL
Rollout of Processes and Procedures (documentation and training)

Maturity Assessment

AppSec Maturation
Enhancements, monitoring & system tuning, process and technology adjustments

Staff Planning
Role definition
Interviews
Hiring

Processes & Procedures
Business
Operational
Technical
Analytical

Data Onboarding

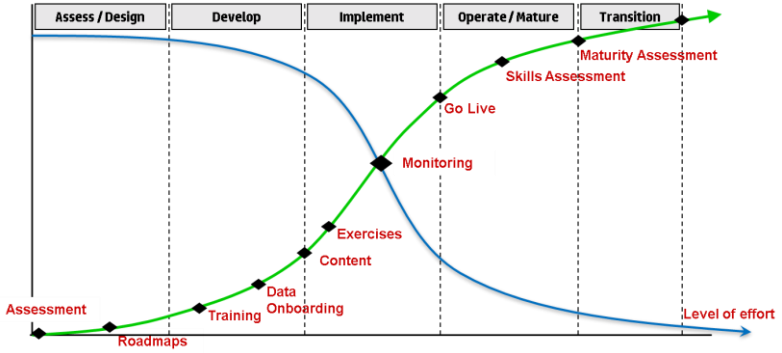
Update Roadmap
Advanced Analysis
Integrated Hunt Operations
ML / Analytics
ODS

Analyst Development Plan
Hiring + Training

Assess
Workshop
Baseline Assessment
Requirements
Data Readiness
Roadmap

Deployment
Toolset Deployment
Toolset integration

AppSec Operations go-live



Source: Fortify Professional Services, 2018

Measure to demonstrate success

- % of security defects identified by sprint/phase
- % of security defects whose risk has been accepted vs. % fixed
- % of security defects per project over time (between quarter to quarter)
  - Vulnerability density (security defects/LOC)
- Average time required to fix/close security defects during design, coding, and testing
- Average time to fix security defects by defect type
- Average time to fix security defects by application size/code complexity

Why Application Security is Hard

Development Point of View: Challenges/Concerns

Security Gets Involved at Later Stages in the Dev Cycle

- Traditionally, static or dynamic scans are run before releasing the app.
- This means either developers get dozens of issues to fix in a very short time or they'll release the app with these issues.
- It is painful to go back to a project that you've already finished and fix things.

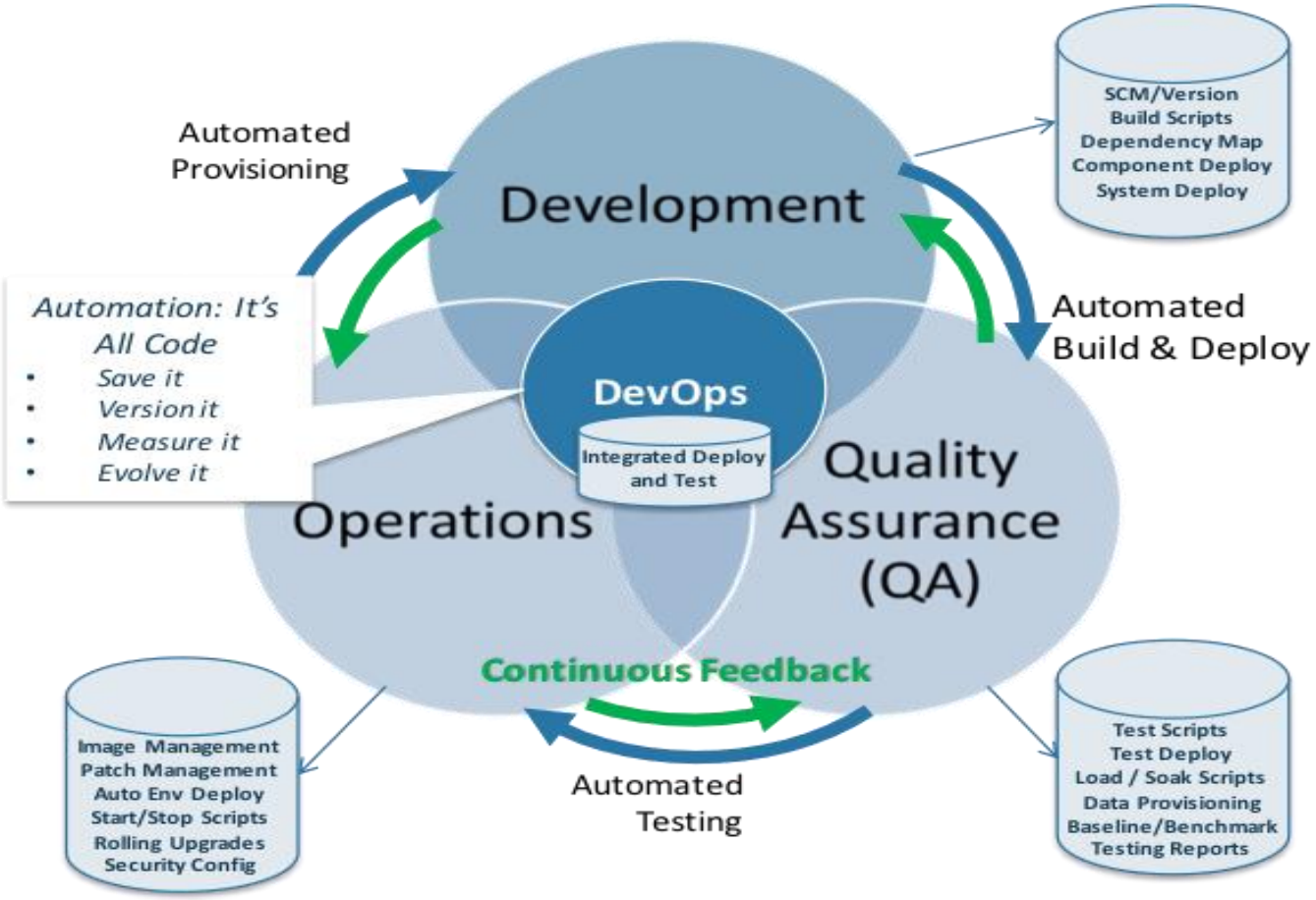
Full Scans Take Too Much Time!

- When scans are initiated, developers don't get results in days in some cases weeks.
- Scanning the entire code base and auditing can take time.
- Developers get security issues way later than they would like.

Audit Process Takes Too Much Time!

- Auditing is still the #1 bottleneck for all application security efforts.
- Even if scans are completed in minutes, human auditors work using FIFO queues and they're outnumbered.
- Audit results are challenged by developers and cause friction/time loss.

Where does Security fit in DevOps?

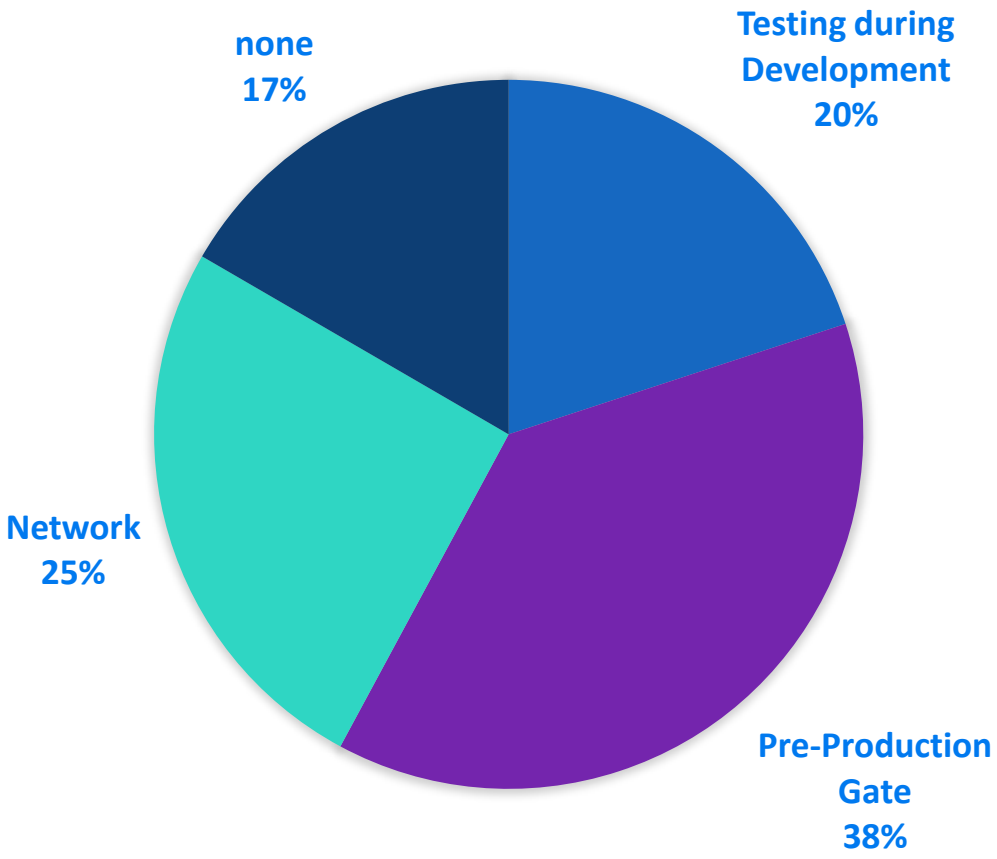


Security??

Promise vs Reality of Security in DevOps

Where does security currently fit?

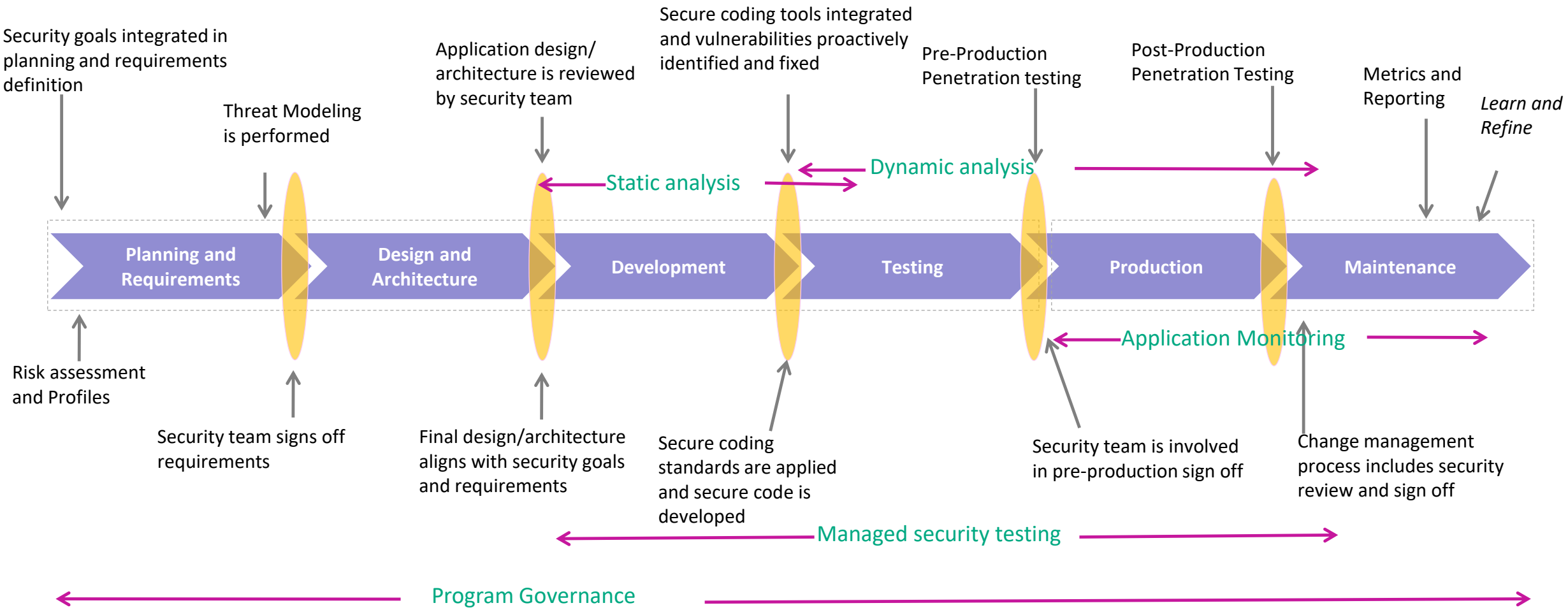
99% of those surveyed agreed that DevOps is an opportunity to improve application security



But only 20% perform application security testing during development. Most wait until late in the SDLC – or not at all!

Reference: Application Security and DevOps Report 2016

Classic integration of security touchpoints can't keep pace



Risk-based application of security activities

DevOps Toolbox

AST Integration can be a challenge given tool diversity

IDE's

eclipse, IntelliJIDEA, Visual Studio Team Foundation Server

Requirements & issues

JIRA, ca, RALLY, Quality Center, DATADOG, Bugzilla, IBM, JUnit, FogBugz, New Relic, Bamboo, Visual Studio Team Foundation Server

Security tools including Open Source

FORTIFY, BLACKDUCK, tripwire, splunk>, Sonatype

Containers

docker

Communication/ ChatOps

HipChat, Visual Studio Team Foundation Server, SharePoint

Code repositories & apps

git, mercurial, Bitbucket, Visual Studio Team Foundation Server, JIR, GitHub

Build servers & Build tools

Bamboo, TeamCity, Jenkins, cucumber, Visual Studio Team Foundation Server, PowerShell, maven, APACHE ANT

Configuration automation

puppet labs, bladelogic, CHEF

Cloud

amazon webservices EC2, servicenow, openstack, cloudera, Microsoft Azure

Modern Application Security Programs Need to Adapt

New languages

Software Supply Chain

Mobile Apps

Internet of Things

Open Source Software Components

Skills

Agility

Micro services
and containers

Bimodal IT

Continuous Integration

Cloud

Automation

DevOps

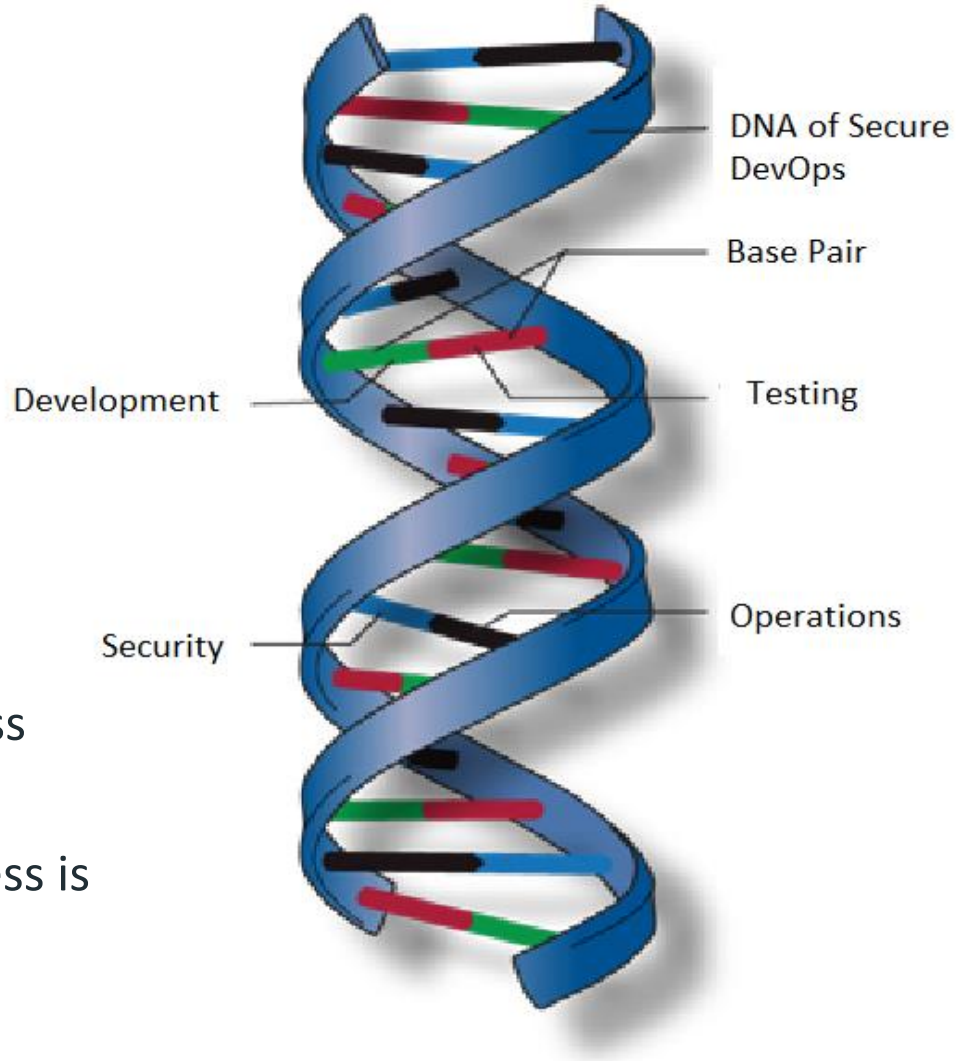
More robust and dynamic apps

Continuous Development

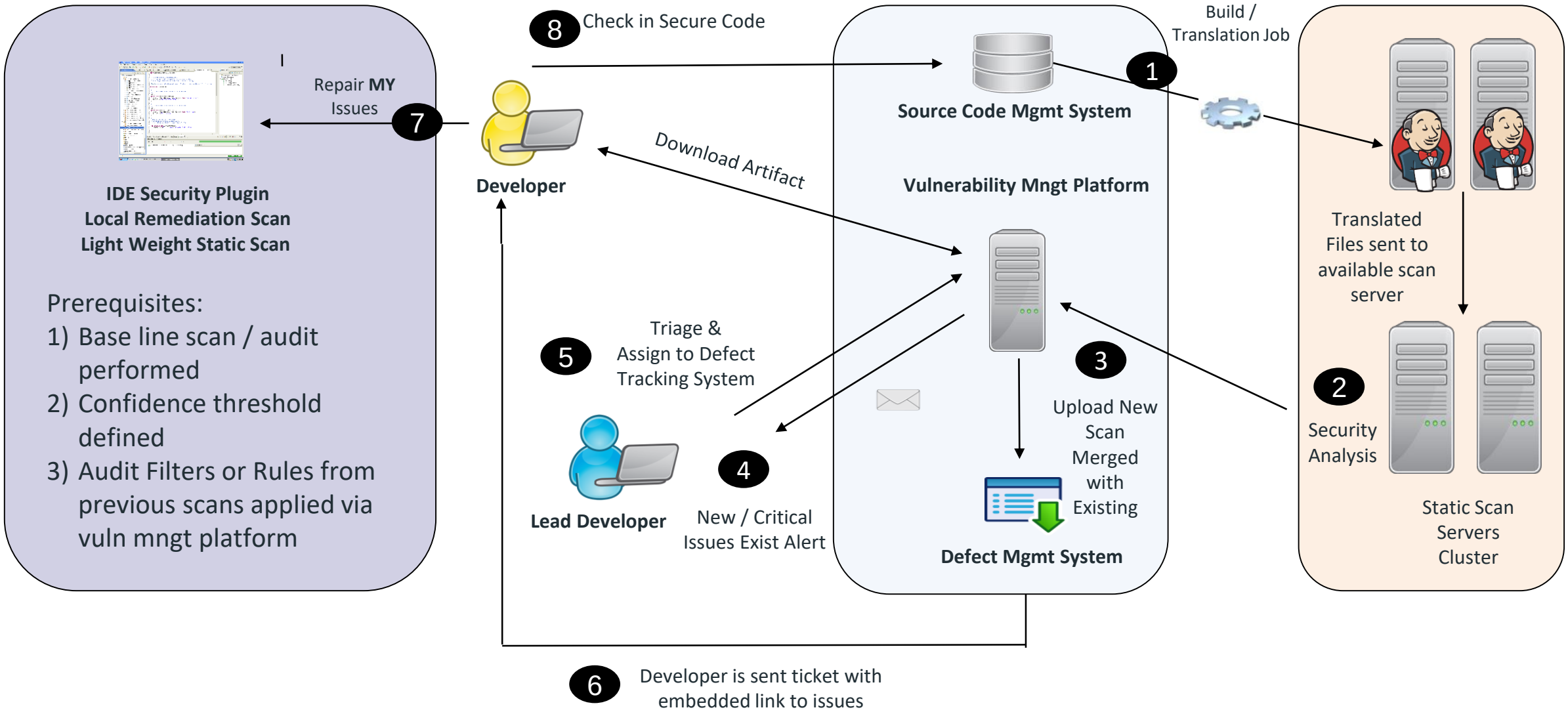
Best practices for integrating security w/DevOps

Security should be part of the DNA of DevOps

- Make security part of the value stream
- Identify skilled early adopters
- Work in small consumable steps
- Standardize on toolset
- Early visible wins
- Focus more on the process than defect totals
- Begin with a loose security policy and tighten as process matures
- Mark builds as unstable but don’t fail builds until process is mature



Effective/High Velocity DevOps with Security built-in



Prerequisites:

1) Base line scan / audit performed
2) Confidence threshold defined
3) Audit Filters or Rules from previous scans applied via vuln mngt platform

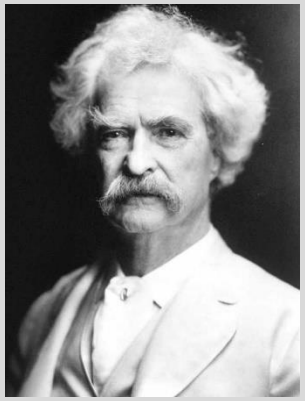
Security **CAN** be part of the DNA of DevOps

Software Security Research Results

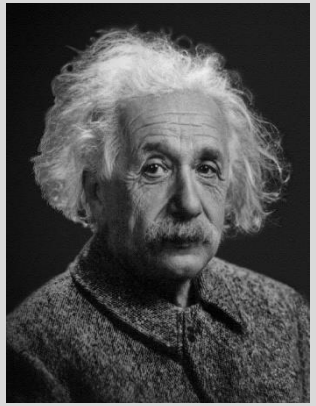
Annual Application Security Research Report

Software Security Research team

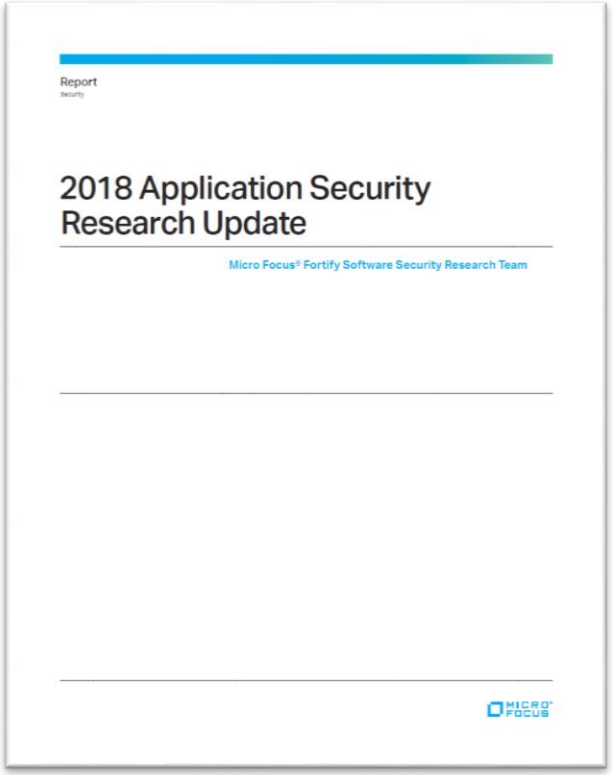
Data



Insight



Meaningful conclusions


Report

2018 Application Security Research Update

Micro Focus® Fortify Software Security Research Team

Data from managed application security platform:

- Anonymized and sanitized vulnerability data collected over a year (Nov 2016 – Oct 2017)
- 7,800+ Web applications & 700+ mobile applications

https://www.microfocus.com/media/report/application_security_research_update_report.pdf

Three themes from the 2018 AppSec risk report

1

Analysis shows broad vulnerability in apps

The majority of web or mobile applications analyzed had at least one critical or high severity issue

2

OWASP Top 10 is a starting point

1 out of 2 apps had critical or high vulnerabilities not covered by the OWASP Top 10 2017

3

GDPR is forcing strong protection of customer data

GDPR strongly hints at the use of encryption and pseudonymization as acceptable approaches to protect personal data; applications are a potential weak link.

2 OWASP Top 10 is an industry best practice

OWASP Top 10 is a powerful awareness document for web application security.

- represents a **consensus about the most critical security risks to web applications**

Some standards reference OWASP Top 10:

- MITRE
- PCI DSS
- United States Federal Trade Commission

Learn more here → https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

T10 OWASP Top 10 Application Security Risks – 2017

6

A1:2017-Injection	Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2:2017-Broken Authentication	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
A3:2017-Sensitive Data Exposure	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
A4:2017-XML External Entities (XXE)	Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
A5:2017-Broken Access Control	Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
A6:2017-Security Misconfiguration	Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.
A7:2017-Cross-Site Scripting (XSS)	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A8:2017-Insecure Deserialization	Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
A9:2017-Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
A10:2017-Insufficient Logging & Monitoring	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

OWASP Top 10 is a starting point, but is not all-inclusive.

Many of the top reported security weaknesses in web application didn't make the list, and it doesn't include vulnerabilities to other attack surfaces of the organization.

90% of applications have at least one issue outside of the OWASP Top 10.

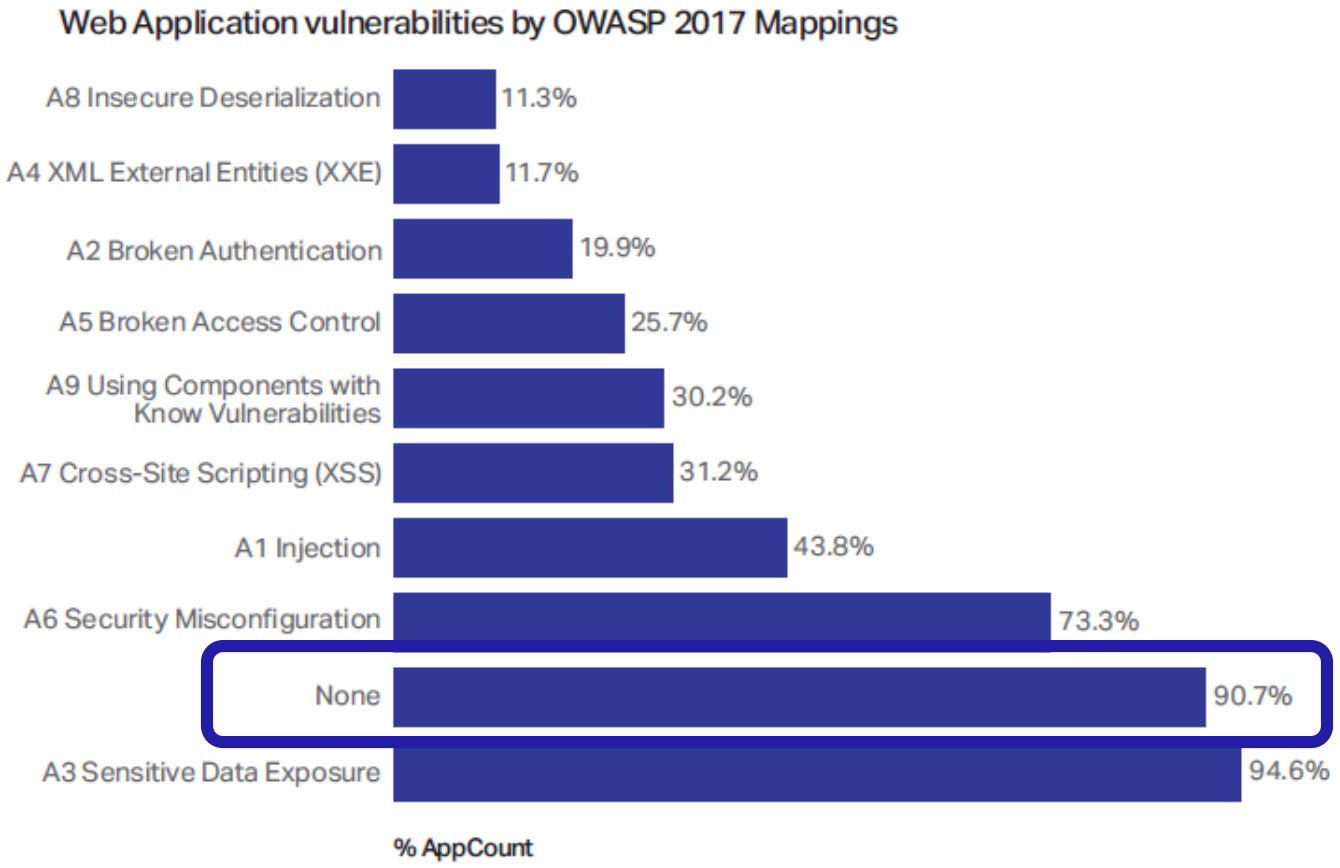


Figure 16. Web application vulnerabilities grouped by OWASP 2017 mappings

Categories outside of OWASP Top 10 (= None) are not lower in severity

1 out of 2 apps have critical or high vulnerabilities **not covered** by the OWASP Top 10 2017

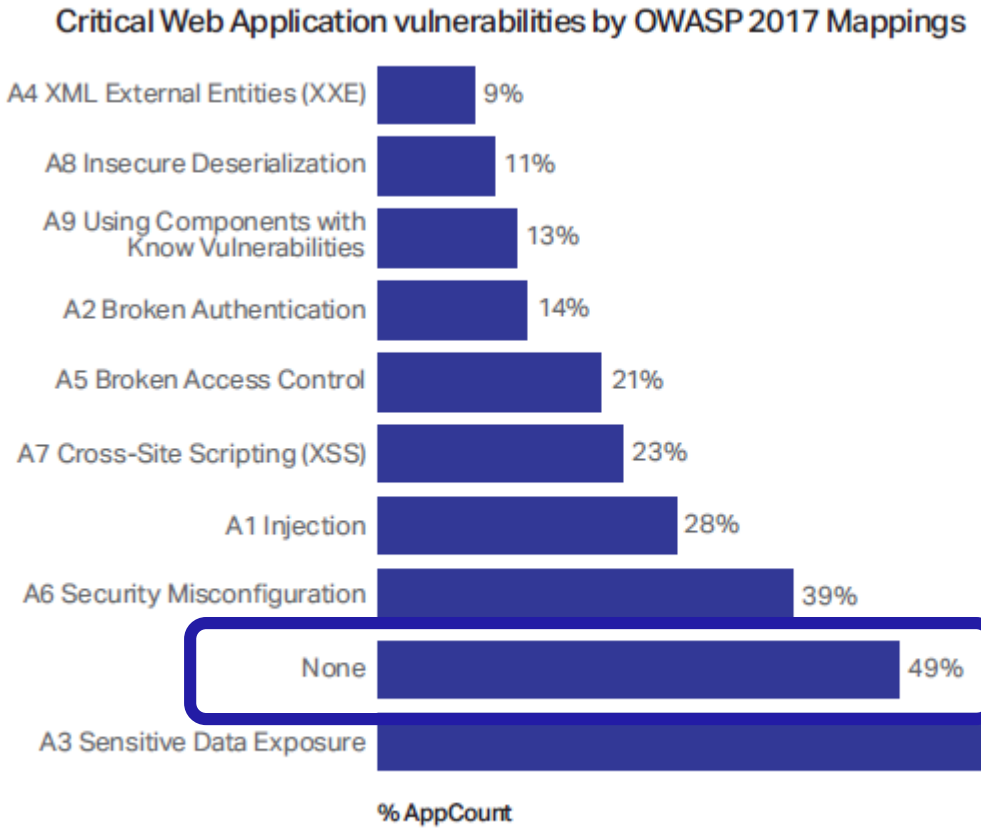
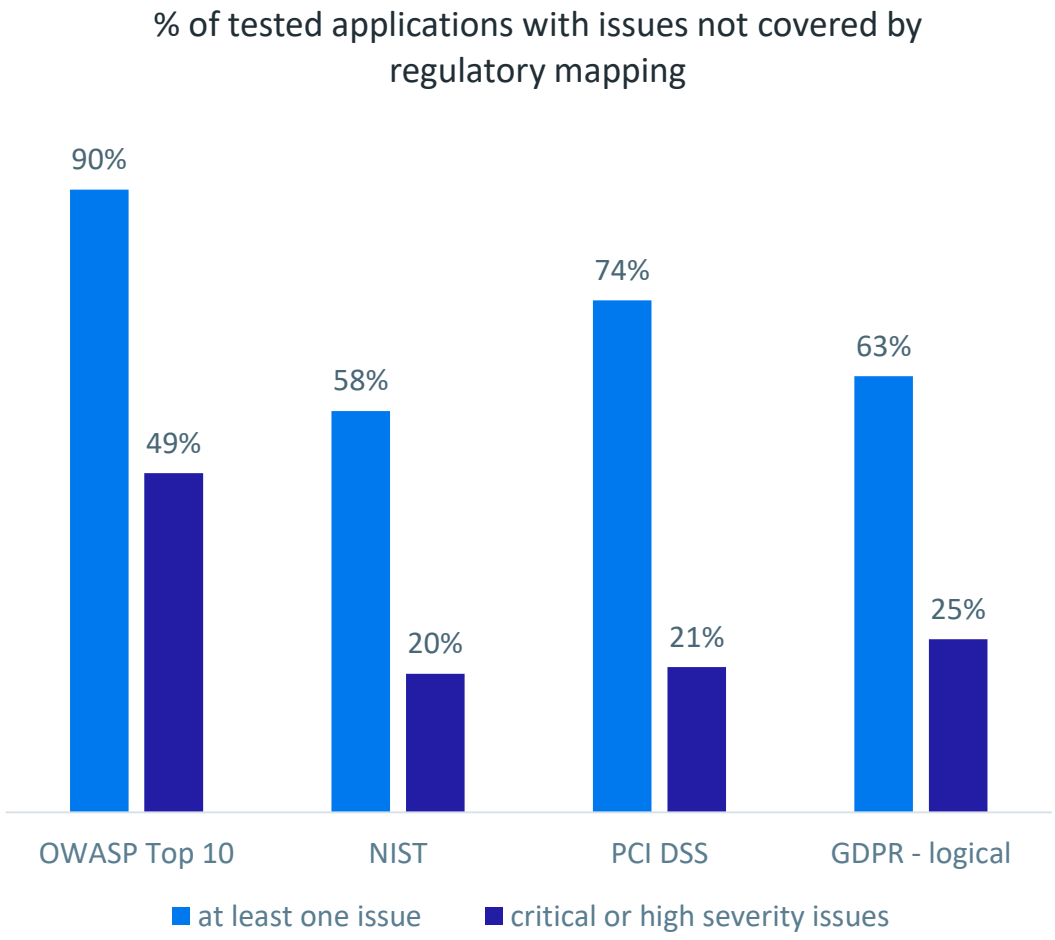


Figure 17. Critical and high-severity web application vulnerabilities grouped by OWASP 2017 mappings

Regulatory compliance with standards and best practices leave out issues across the board

Compliance with standards like OWASP Top 10, NIST, PCI DSS, or GDPR is a great place to start, but **all standards have critical and high severity weaknesses issues that aren't covered**.

Depending on the standard, anywhere from 1/5 to 1/2 apps have critical or high vulnerabilities not covered by the regulatory mapping.



Summary

Making Application Security Matter

- Adapt and catch the wave!
- Establishing an application security capability is an evolutionary journey
- Modern application security programs need to adapt
- Security should be part of the DNA of DevOps
- OWASP Top 10 is a starting point for application security, but 1 out of 2 apps had critical or high vulnerabilities not covered by the OWASP Top 10 2017

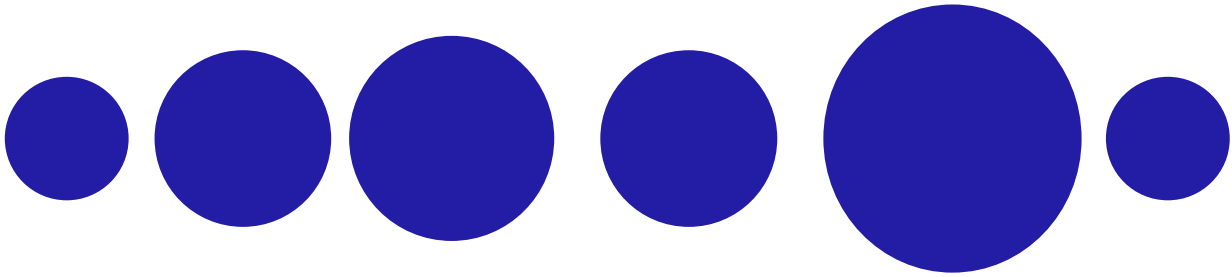
2018 Application Security Research Update

https://www.microfocus.com/media/report/application_security_research_update_report.pdf

Final thought

This stuff doesn’t happen overnight

Do small



Thank You.