We DevOps'd – Experience and Lessons Learned Securing the SDLC

Sherly Abraham, PhD., Excelsior College

Din Cox, PhD., CISSP, ISSAP, ISSMP, CSSLP, CISA, CISM, CRISC, CEH, etc.,

Medical Science and Computing, LLC

Sherly Abraham, Ph.D.

- Excelsior College
  - Program Director for Cybersecurity
- Research Interests
  - Software Security
  - Information Security Training
  - Corporate Governance

Presentation Objectives

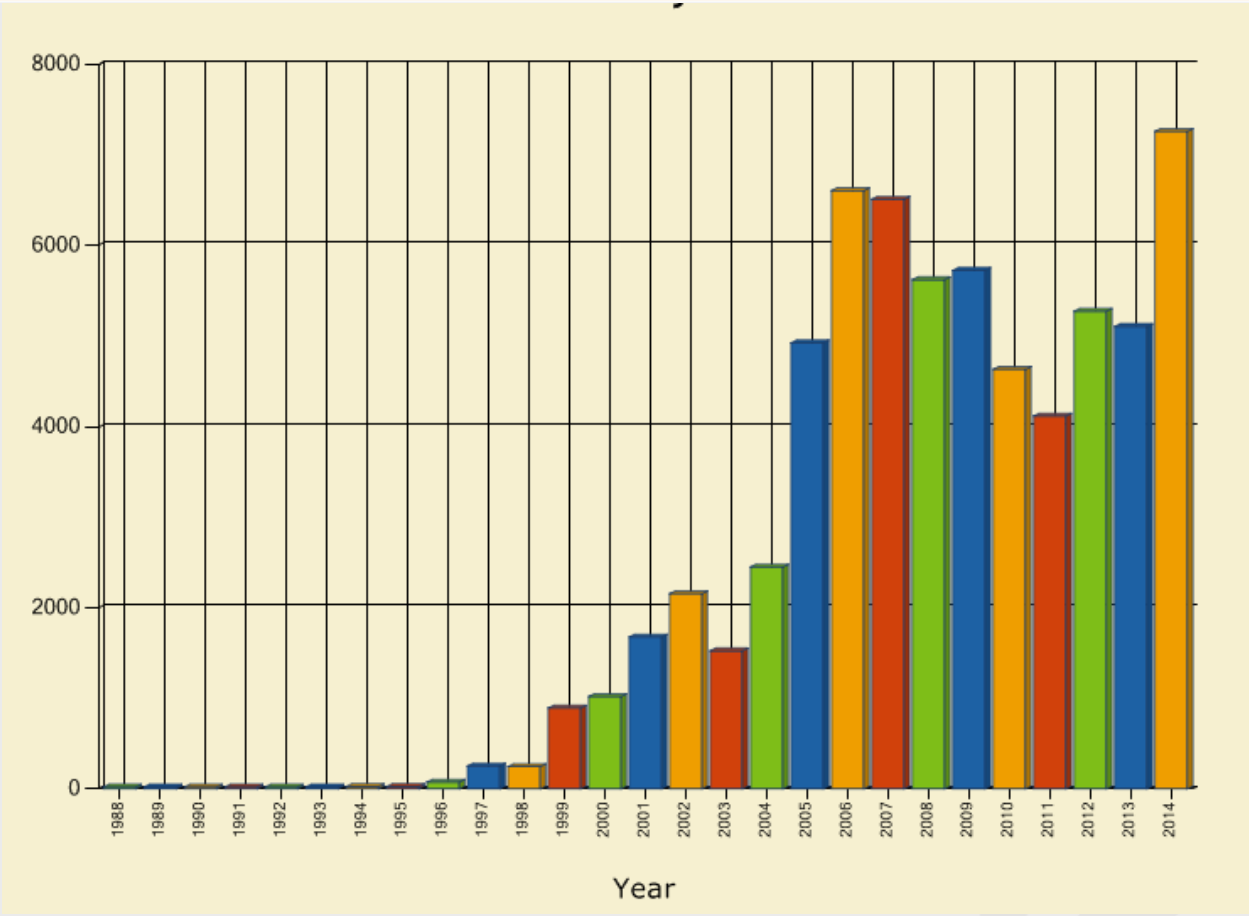
- Software Security
- Challenges in enterprise software security
- What is DevOps
- DevOps Foundations
- Relevance of DevOps to Secuirty
- Lessons learned from application of DevOps
- Recommendations and Resources

2014-2015 Software Bugs

- Heart Bleed
- Shellshock
- Poodle
- Gotofail

Growth Software Vulnerabilities

Number of Vulnerabilities caused by Software Flaws



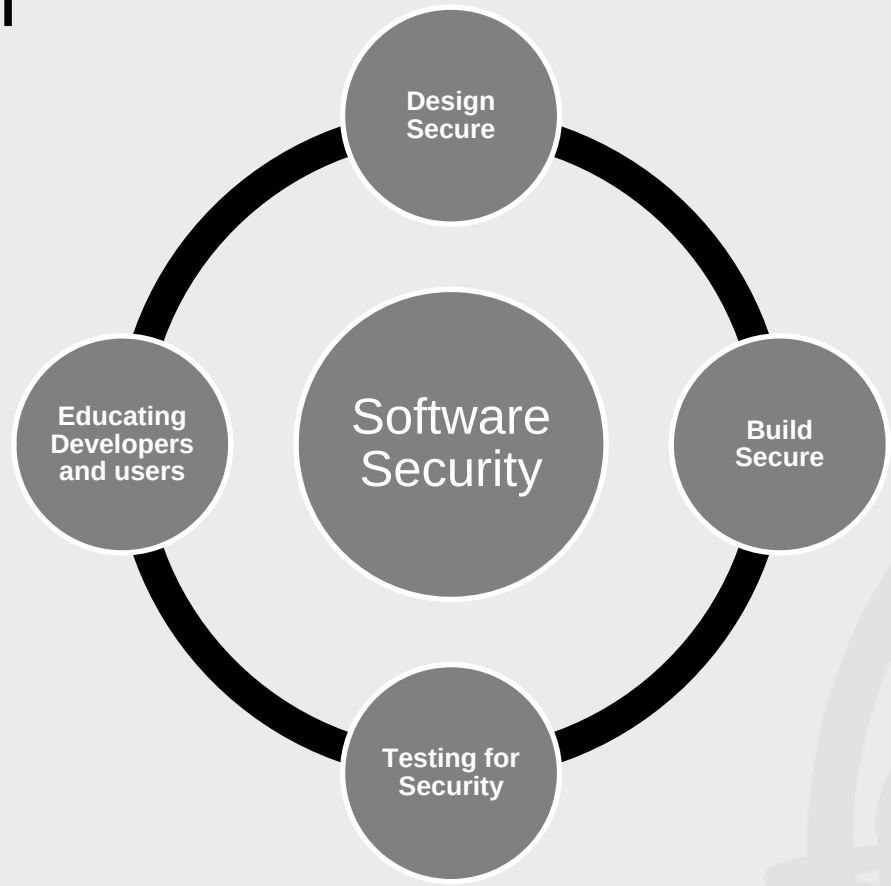
Source: National Vulnerability Database

Software Security Issues

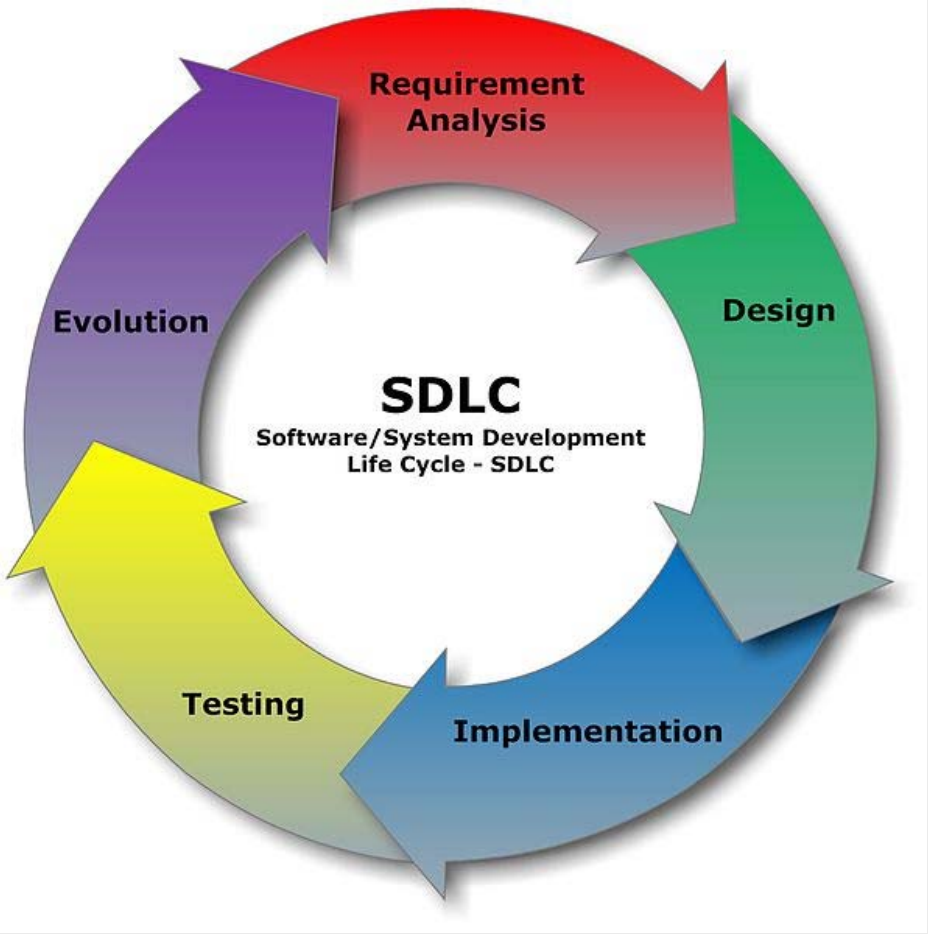
- Defects
- Bugs
  - Eg. Buffer overflow
- Design Flaws
  - Inconsistent error handling
- Maintenance Hooks
  - Backdoors

Software Development Security

- Requires a “holistic” and “proactive” approach



Software Development Life Cycle



Reference: WikiCommons, http://commons.wikimedia.org/wiki/File:SDLC_-_Software_Development_Life_Cycle.jpg

Software Development Models

- Linear Sequential
  - Waterfall model
- Incremental
- Prototyping
  - RAD
- Iterative
  - Spiral
- Agile
  - Teamwork, Iterative and Incremental

Challenges: Enterprise Software Security

- Security not built-in
- Disconnect between developers, business owners, end users and quality assurance
- Configuration Management
- No established metrics and continuous improvement
- Complexity and diversity of development tools, programming languages, and platforms

What is DevOps

- Lean and Agile methods
- Narrow the disconnect between development and business drivers
- Strong collaboration between developers, operations, business, security, and quality assurance teams
- Continuously incorporate feedback from customers and business owners

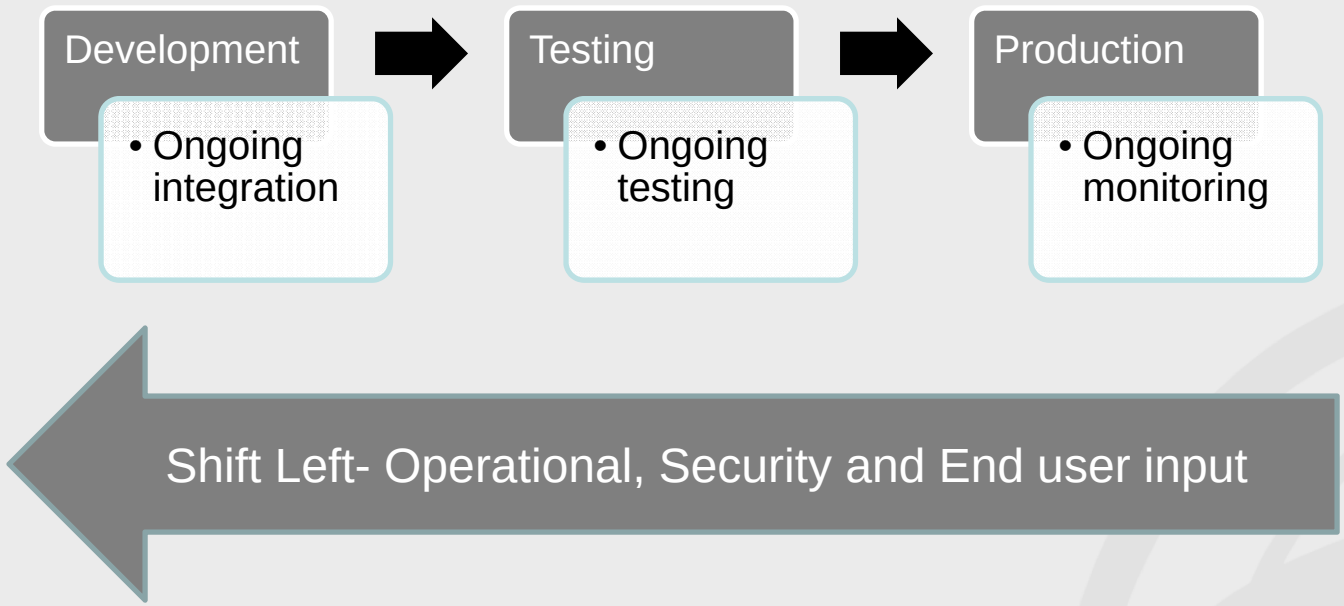
Foundations of DevOps

- Shift Left Concept
  - Address operational issues earlier
  - Test with systems that behave like production
- Agile and Iterative Approach
  - Continuous, automated deployment and testing
- Metrics and evaluation of quality
  - Measure and test effectiveness earlier in the development cycle
- Facilitate feedback from all stakeholders
  - Enable all stakeholders to communicate and provide feedback

DevOps Focus

- Rapid incorporation of customer feedback
- Faster Delivery Process
- Collaboration between disparate teams
- Continuous release and deployment
- Continuous testing
- Ongoing evaluation

DevOps Architecture



What DevOps is not

- Another Software development model
- Everything runs and tested in production
- Blurs the line between developers, system administrators, security
- Tool specific
- A specific job title for DevOps

Relevance of DevOps to Security

- Integration of security in the early stages of development
- Security testing in early stages of development
- Strong Cross functional integration
- Configuration management

Din Cox, Ph.D

- Medical Science and Computing, LLC
  - Application Security Focus
- Research Interests
  - Mobile and Application Security
  - Biometrics
  - Machine Learning
- SynAck Red Team Security Researcher
  - Bug hunter

State of Affairs

https ://puppetlabs.com/sites/default/files/2014-state-of-devops-report.pdf

Who Took the Survey

Overview

While the majority of survey participants work in the technology and web software industries (23 percent and 11 percent, respectively), there were strong showings from other industries as well. Education (8 percent), banking and finance (7 percent) and entertainment and media (7 percent) are just a few of the industries represented in our survey.

Organizational Characteristics

Respondents came from organizations of all sizes: from tiny startups to 10,000-employee companies, from shops with fewer than 100 servers to large enterprises with more than 10,000 servers under management.

In terms of organization size, the plurality of responses — 27 percent — came from companies with 500 to 9,999 employees. With respect to the number of servers managed, the majority of respondents (51 percent) said their infrastructure included fewer than 500 servers. Just 13 percent said their organizations had more than 5,000 servers.

Industry

Technology	22.7%
Web Software	10.9%
Education	7.5%
Finance/Banking	7.4%
ENTMT/Media	6.8%
Consulting	5.9%
Telecomunications	5.7%
Government	4.5%
Retail	3.7%
Healthcare	3.0%
All Others	21.9%

Company Size by # of Employees:

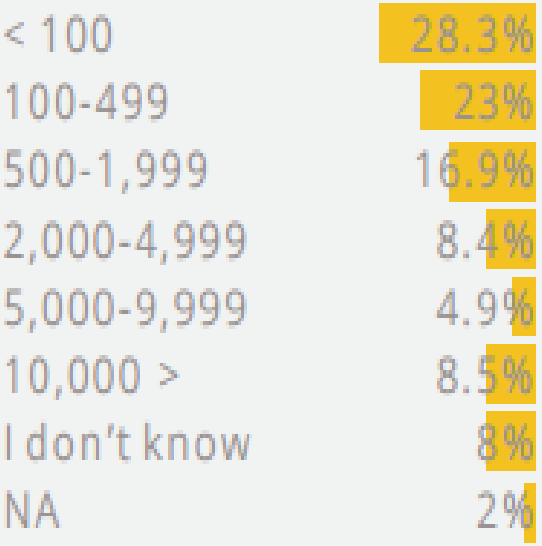
1-4	5.8%
5-9	3.6%
10-19	5.8%
20-99	17.1%
100-499	21.8%
500-9,999	26.8%
10,000+	15.8%
I don't know	2.1%
Not applicable	1.3%

https://puppetlabs.com/sites/default/files/2014-state-of-devops-report.pdf

Demographics

In the 2012 survey, 57 percent of respondents said they worked in IT operations, and more than 33 percent were in development/ engineering. Most remaining respondents fell into the "other" category. In the 2013 survey, we grouped the most common "other" responses to better understand the job roles of our respondents, and how IT and development functions are structured in organizations.

Size of IT infrastructure by # of servers



Results

Key Findings

- Companies with high-performing IT organizations are twice as likely to exceed their profitability, market share and productivity goals.
- IT performance improves with DevOps maturity, and strongly correlates with well-known DevOps practices.
- Culture matters. The cultural practices of DevOps are predictive of organizational performance.
- Job satisfaction is the No. 1 predictor of performance against organizational goals.

https://puppetlabs.com/2014-devops-report

TODAY'S DEVOPS

PART DEV — PART OPS

Application Performance
Modern developers use APM tools to decrease latency, have complete visibility into code, databases, caches, queues and third part services.

End User
A great developer understands end users have the best feedback and analytics play an enormous part of understanding users. Developers re constantly monitoring end user latency and checking performance by devices and browsers.

Quality Code
Developers need to ensure their deployments and new releases don't implode or degrade the overall performance.

Code-Level Errors
When you have a large distributed application it is vital to lower MTTR by finding the root cause of errors and exceptions.

Application Availability
The applicants need to be up and running and it's Ops responsibility to ensure up time and SLAs are in order.

Application Performance
Classic Ops generally rely on infrastructure metrics – CPU, memory, network and disk I/O, etc. Modern Ops correlate all of those metrics with application metrics to solve problems 10 x faster.

End User Complaints
The goal is to know about and fix problems before end users complain, reduce the number of support tickets and eliminate false arts.

Performance Analytics
Automatically generated baselines of metrics help Ops understand what has changed and where to focus their troubleshooting efforts. Alerts based upon deviation from observed baselines improves alert quality and reduce alert noise.

http://www.s-sa.co.uk

Organizational Context

- Current Project – Rugged DevOps
  - Integrate and promote secure coding practices in SDLC across the organization – Agile, Waterfall.
  - 700+ developers – Geographically dispersed
  - Multiple languages and frameworks (Java, PHP, Django, Python, Angular, ColdFusion, Ruby, etc.) + Mobile (iOS, Android)
  - Training and Education

Success Factors

- Cultural change – i.e. view of software security
- Clear repeatable processes
  - Software must be scanned before going to production
  - Policy alignment – remediation timeframe
- Fault detection automation
- Continuous integration – automating unit testing and deployment of software

Success Factors

- Security standard adoption for software development
  - Ability to balance security risks with software development agility.
- Improve effectiveness of public facing applications
  - Usage patterns, break/fix

EXCELSIOR COLLEGE.

DevOps Tools

Application Servers

- JBoss
- Tomcat
- Jetty
- Glassfish

Web Servers

- nginx
- Apache
- IIS

Test and Build Systems

- Jenkins
- Maven
- Ant
- Gradle

Monitoring, Alerting, and Trending

- New Relic
- Nagios
- Icinga
- Graphite
- Ganglia
- Cacti
- PagerDuty
- Sensu

Databases

- Percona Server
- MySQL
- PostgreSQL
- OpenLDAP
- MongoDB
- Cassandra
- Redis
- Oracle
- MS SQL

Operating Systems

- Linux (RHEL, CentOS, Ubuntu, Debian)
- Unix (Solaris, AIX, HP/UX, etc.)
- Windows
- Mac OS X

Infrastructure as a Service

- Amazon Web Services
- Rackspace
- Cloud Foundry
- Azure
- OpenStack

Virtualization Platforms

- VMware
- KVM
- Xen
- VirtualBox
- Vagrant

Secure SDLC

- Security requirements need to be defined as early as possible during the SDLC

Accomplishments

- Agile Testing (security)
- Secure Coding + Operations + Collaboration
- Developer training and education
- Rapid communication on vulnerability intelligence
- Quicker patch cycles/remediation of vulnerabilities
- Collaboration between Development and Operation

Security Automation

- SAST (Static Application Security Testing)
  - Source code, byte code or application binaries for conditions indicative of a security vulnerability
  - Leverage tools – statics analysis, etc.
- DAST (Dynamic Application Security Testing)
  - Black-box (Functional and non-functional), White-box, and Defect-based tests.
  - Examine application at runtime to identify vulnerabilities
  - Robustness testing (i.e. fuzz testing) or fault-injection
- Integrate with build and code repositories
  - GIT, Bamboo, Jenkins, etc.

Realized Benefits

- Identify problems early
- Continuous integration
- Infrastructure automation
- System stability and uptime
- Monitoring
- Deployment
- Continuous delivery – testing

Challenges

- Misaligned tools and processes
- Competing interests (development vs operation)
- Infighting – who's at fault when something happens
- Documentation
- Varying views of security and roles

Lessons Learned

- Require resources – People
- Cannot be done in a vacuum, dynamic
- Align IT with the business
- Leverage internal talent
- Visibility of applications – Customer experience, including components (server, DB, etc.)
- Training and education

Recommendations

- Start at the Top
  - Organization buy-in and support
- Measure Success – metrics
  - Deployment frequency
  - Mean time to recover (MTTR)
- Identify system failures / waste
  - Automate where possible (puppet, etc.)
- Decompose system components into modules

- Identify a champion in each department
- Establish a center of excellence

Resources

- http://www.rackspace.com/blog/enterprise-cloud-forum-recap-prepare-for-devops-success/
- http://www.isaca.org/knowledge-center/research/researchdeliverables/pages/devops-overview.aspx
- https://puppetlabs.com/2013-state-of-devops-infographic
- https://puppetlabs.com/sites/default/files/2014-state-of-devops-report.pdf