Secure software development

An overview of building security into application software systems

Chris Horn

April 2019

SECURE DECISIONS
A DIVISION OF APPLIED VISIONS, INC.

About Chris Horn

Principal consultant,
Product strategy & development

Experience

- 18 years in research, software systems, and new product development
- Principal Investigator at Secure Decisions, an R&D division of Applied Visions
- Focused on developing technologies to improve application security

We provide software development, security research, and application security management

Applied Visions, Inc.

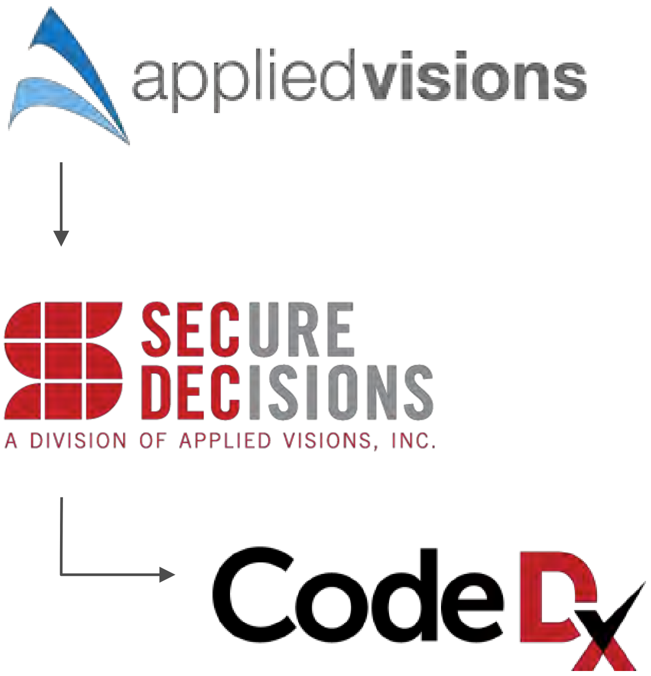
- Software development since 1987
- Primarily develops business applications

dba, Secure Decisions

- Cyber R&D
- Primarily serving DHS and DoD, including DARPA, ONR, AFRL; some intel and commercial projects

Code Dx, Inc.

- Application security spin-out based on SBIR funded by DHS S&T in 2011



Outline of today’s talk

Baseline

- Systems engineering
- Security

DevSecOps

- Overview

for each stage of DevSecOps

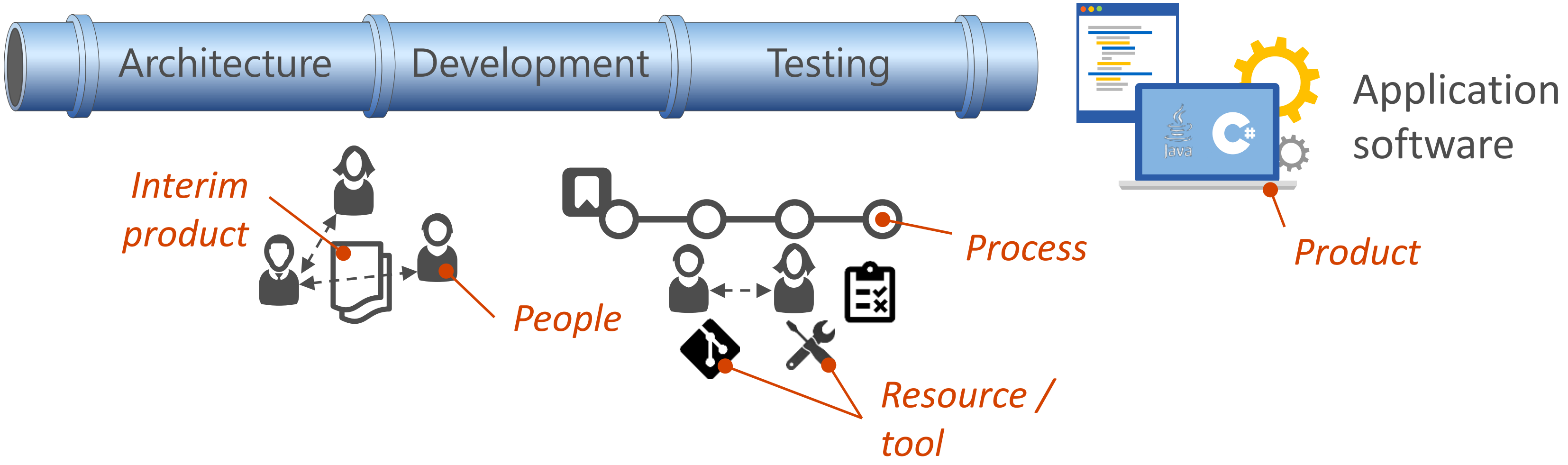
- Security practices

Three key takeaways from today's talk

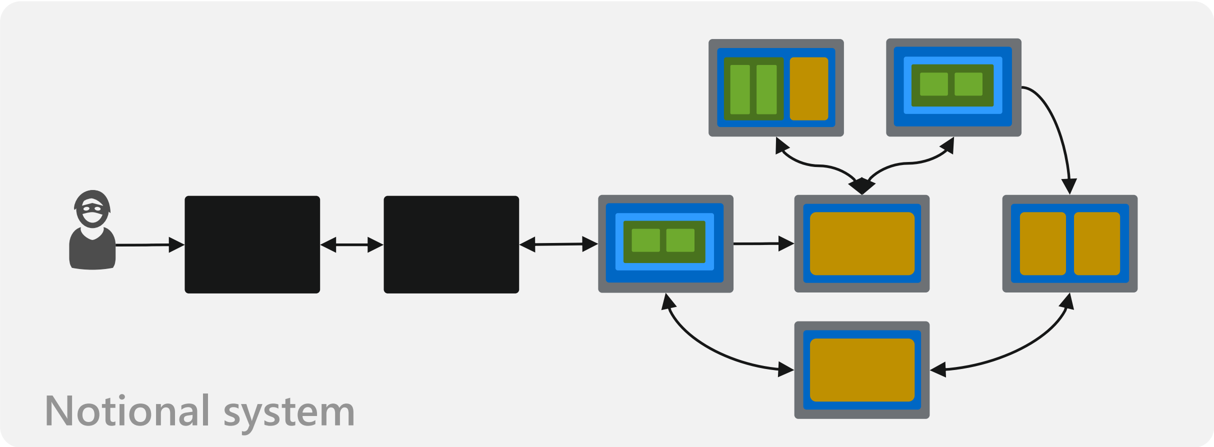
1. Security is perfectly compatible with DevOps
2. There is no silver bullet to achieve security
   - Results from hundreds of smaller decisions and actions
   - Coordinated application of people, process, and technology
3. There are many great public resources to support learning
   - Citations & links included

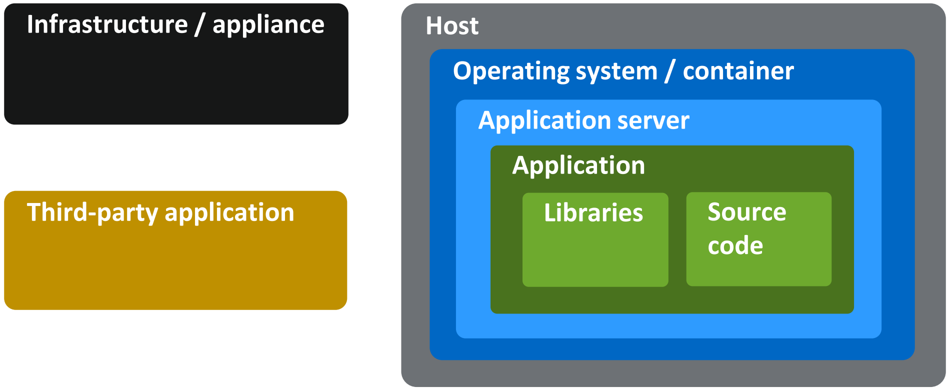
Baseline

People apply thought, process, and technology to create application software



Application systems are composed of components that each have their own security needs





Infrastructure / appliance

Third-party application

Host

Operating system / container

Application server

Application

Libraries

Source code

Systems engineering integrates multiple skillsets

The art and science of guiding the end-to-end creation of systems

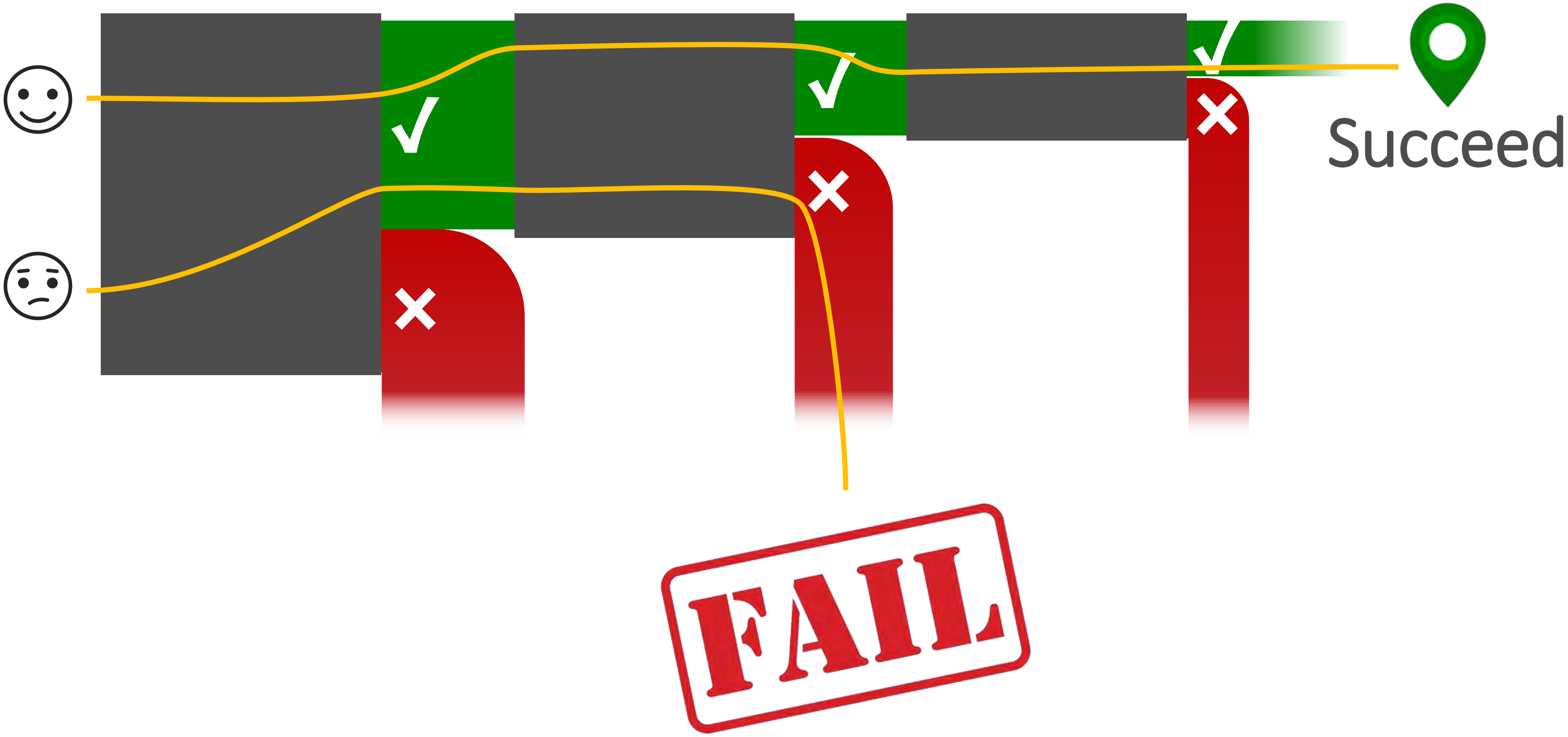
- Art because it involves extensive people skills and leadership
- Science because it requires rigorous applications of tools & methodologies

Interdisciplinary approach governing the total technical and managerial effort required to transform a set of stakeholder needs, expectations, and constraints into a solution and to support that solution throughout its life.

Jolly, Steve. "Systems Engineering: Roles and Responsibilities." presented at the NASA Principal Investigator Forum, Annapolis, MD, July 27, 2011. https://www.nasa.gov/pdf/580677main_02_Steve_Jolly_Systems_Engineering.pdf.

"Glossary: Systems Engineering." Computer Security Resource Center. Accessed April 2, 2019. https://csrc.nist.gov/glossary/term/systems-engineering.

There are many ways a system can fail



Security feels good

The state of being free from danger or threat

Freedom from those conditions that can cause loss of assets with unacceptable consequences

An *asset* is an item of value to stakeholders that may be:

- Tangible (e.g., a physical item such as hardware, firmware, computing platform, network device, or other technology component)
- Intangible (e.g., data, information, software, trademark, copyright, patent, intellectual property, image, or reputation)

Ross, Ron, Michael McEvilley, and Janet Carrier Oren. "Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems." National Institute of Standards and Technology, November 2016. https://doi.org/10.6028/NIST.SP.800-160.

Security is an emergent system property

Emergent properties are the opposite of one-and-done feature development

- Result from tens and hundreds of smaller decisions

Other emergent system properties include:

- Availability
- Usability
- Safety
- Security
- Maintainability
- Resilience
- Reliability
- Agility
- Survivability
- Recoverability
- Supportability
- Durability

DevSecOps

DevSecOps is a framework

DevSecOps is a collection of ideas and approaches that represent an evolution in how we develop systems

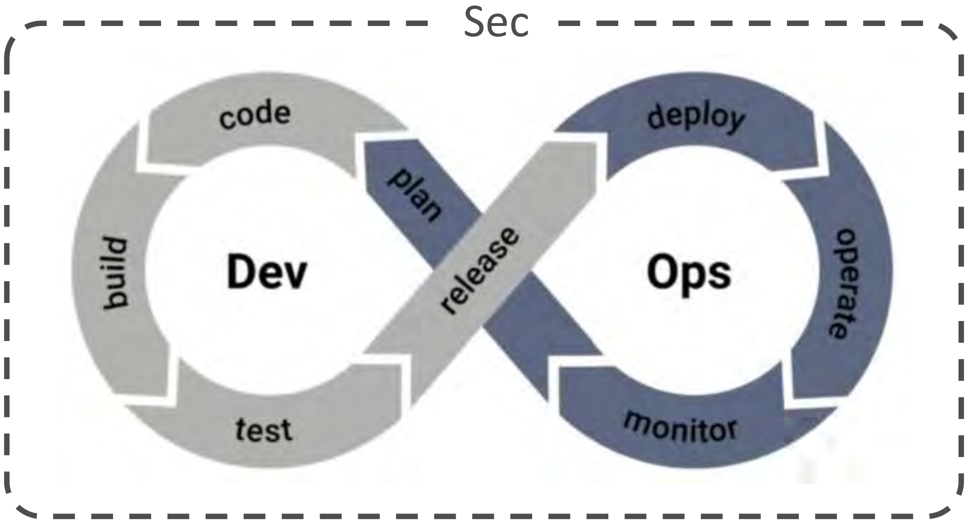
DevSecOps is about culture

DevSecOps values, beliefs, attitudes, and behaviors include:

- Iterative value delivery
- Shared responsibility
- Autonomous teams
- Automation
- Measurement
- Learning & experimentation

Wilsenach, Rouan. “DevOpsCulture.” martinfowler.com. Accessed April 2, 2019. https://martinfowler.com/bliki/DevOpsCulture.html.

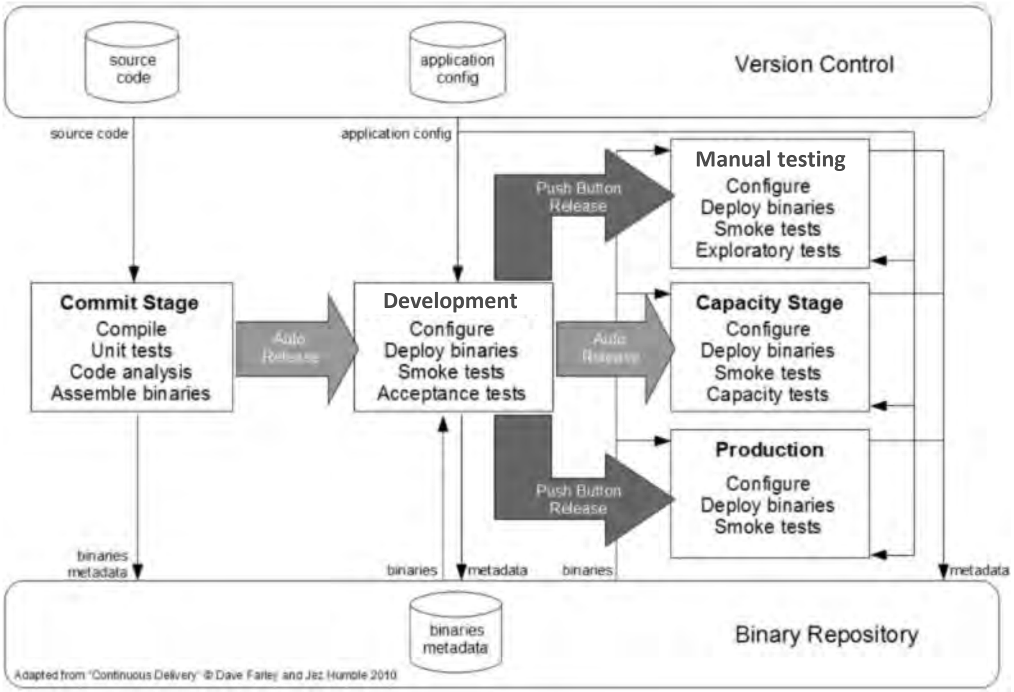
DevSecOps is about process



DevSecOps is about technical approaches

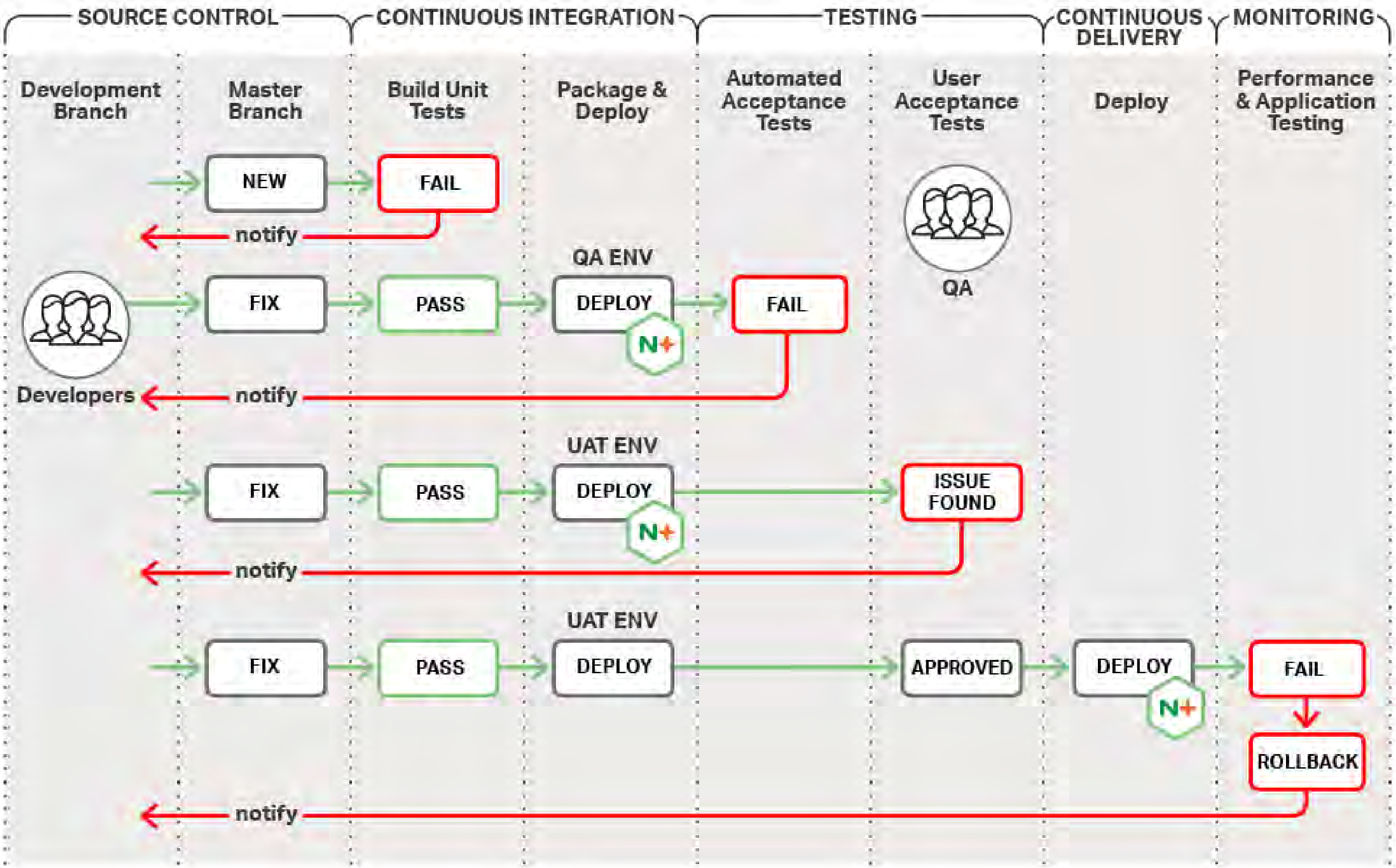
Continuous integration (CI) & continuous deployment (CD)

- Cloud technologies
- Everything as code
- APIs
- Automated testing
- Rollback / recovery



Azeri, Izzy. "What Is CI/CD?" DZone. Accessed April 2, 2019. https://dzone.com/articles/what-is-cicd.

DevSecOps is about review processes



"Continuous Integration/Continuous Delivery with NGINX and NGINX Plus." NGINX, May 31, 2017. https://www.nginx.com/blog/introducing-cicd-with-nginx-and-nginx-plus/.

DevSecOps is a framework

DevSecOps is an evolution to how we develop systems

Most practices from other frameworks carry through

- Systems engineering
- SDLC
- Risk management
- Cybersecurity

DevSecOps is how we apply those practices

Resources for learning about DevSecOps

“Devops Security Checklist.” Sqreen. Accessed April 1, 2019. https://www.sqreen.com/checklists/devops-security-checklist.html.

Synopsys. *Build a DevSecOps Culture with Automated and Integrated Security Tools*, 2018. https://vimeo.com/285843713.

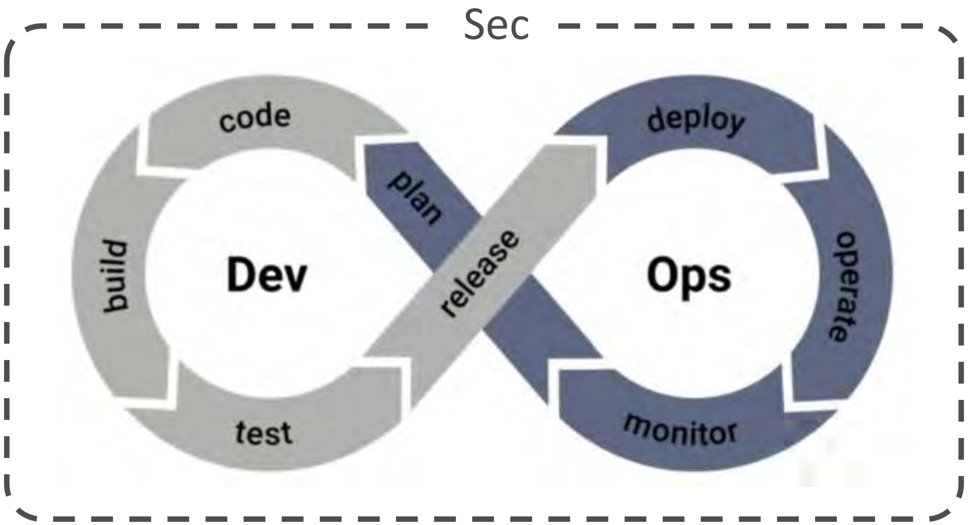
“Common Security Challenges in CI/CD Workflows.” DZone. Accessed April 1, 2019. https://dzone.com/articles/common-security-challenges-in-cicd-workflows.

“Introduction to DevSecOps, Refcard #267.” DZone. Accessed April 1, 2019. https://dzone.com/refcardz/introduction-to-devsecops.

Security practices in DevSecOps

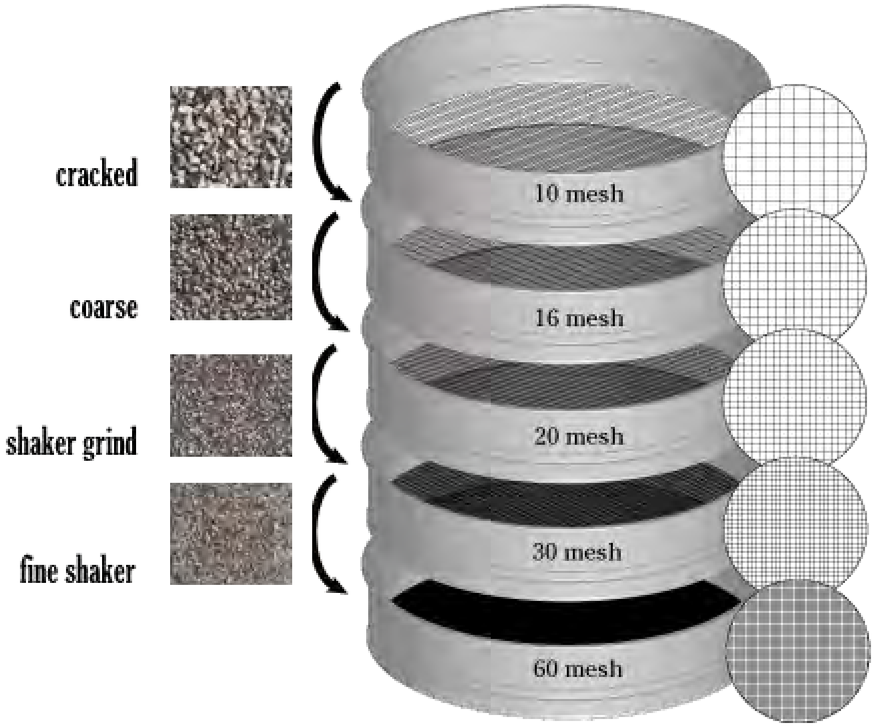
DevSecOps

1. Plan & design
2. Code
3. Build
4. Test
5. Release
6. Deploy
7. Operate
8. Monitor



SECURE DECISIONS
A DIVISION OF APPLIED VISIONS, INC.

The development pipeline is like a series of screens



Early in pipeline

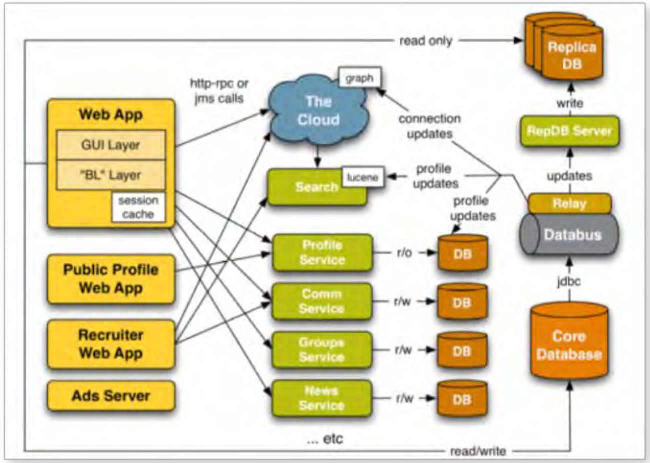
Identify & resolve major, overarching issues

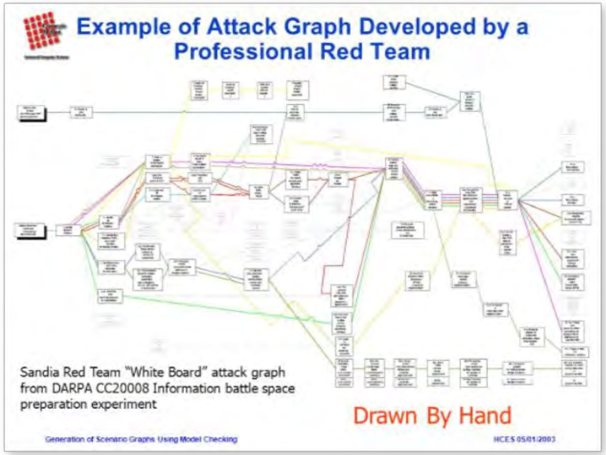
Late in pipeline

Identify & resolve relatively small, detailed issues

Aim to detect issues as early as possible

Plan & design



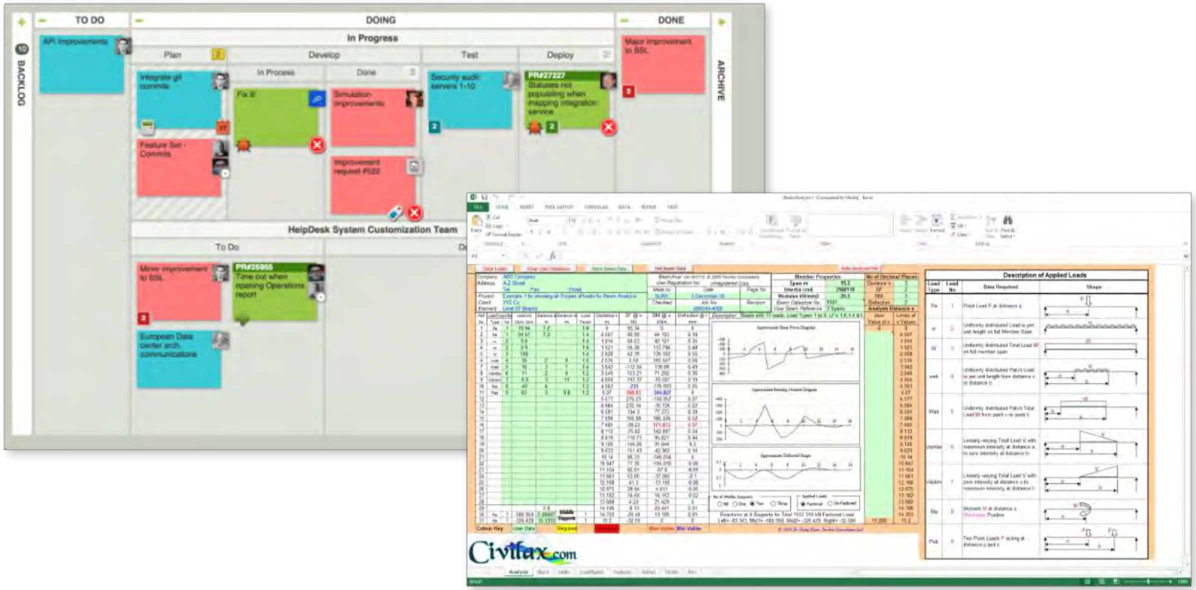

Example of Attack Graph Developed by a Professional Red Team

Sandia Red Team "White Board" attack graph from DARPA CC20008 Information battle space preparation experiment

Drawn By Hand

Generation of Scenario Graphs Using Model Checking

HCES 05/01/2003



Important design principles, patterns, & methods

Principles of secure design

- Compartmentalization
- Minimize attack surface
- Defense in depth
- Economy of mechanism (KISS)
- Authenticate, then authorize
- Least privilege
- Separation of privilege
- Graceful degradation
- Safe defaults
- Audit trails & logging
- Open design
- …more

Design patterns

Threat modeling

- Abuse cases
- Attack graph/map
- Fault trees

Attacker mindset

“Design Principles.” US-CERT. Accessed April 1, 2019. https://www.us-cert.gov/bsi/articles/knowledge/principles/design-principles.

“Security by Design Principles.” OWASP. Accessed April 1, 2019. https://www.owasp.org/index.php/Security_by_Design_Principles.

Arce, Iván, Neil Daswani, Jim DelGrosso, Danny Dhillon, Christoph Kern, Tadayoshi Kohno, Carl Landwehr, et al. “Avoiding the Top 10 Software Security Design Flaws.” *IEE Center for Secure Design*, 2014, 32. https://ieeecs-media.computer.org/media/technical-activities/CYBSI/docs/Top-10-Flaws.pdf.

Microsoft. “Cloud Design Patterns.” Microsoft Azure Architecture Center. Accessed April 2, 2019. https://docs.microsoft.com/en-us/azure/architecture/patterns/.

Gerber, Joe, Jay Reynolds, Chris Wells, and Christian Price. “Building Patterns for Secure Micro Services, an Approach and Pattern Zero Candidate.” presented at the Rocky Mountain Information Security Community 2018, Denver, CO, May 9, 2018. https://drive.google.com/file/d/1qF70eYdWhueNmmdmy2fuYXDOr1fkXZZd/view.

Healthy paranoia requires knowing the threats

Types of problems

- Physical damage
- Loss of essential services
- Technical failure
- Function compromise
- Information compromise

Degrees of intent

- Deliberate
- Negligence
- Accidental
- Environmental

Human adversaries

- National governments
- Terrorists
- Industrial spies
- Organized crime groups
- Hacktivists
- Hackers
- Insiders
  - Disgruntled
  - Financially-motivated

"Threat (Computer)." *Wikipedia*, March 27, 2019. https://en.wikipedia.org/w/index.php?title=Threat_(computer)&oldid=889722177.

"Cyber Threat Source Descriptions." U.S. Federal Government. Cybersecurity and Infrastructure Security Agency (CISA). Accessed April 1, 2019. https://ics-cert.us-cert.gov/content/cyber-threat-source-descriptions.

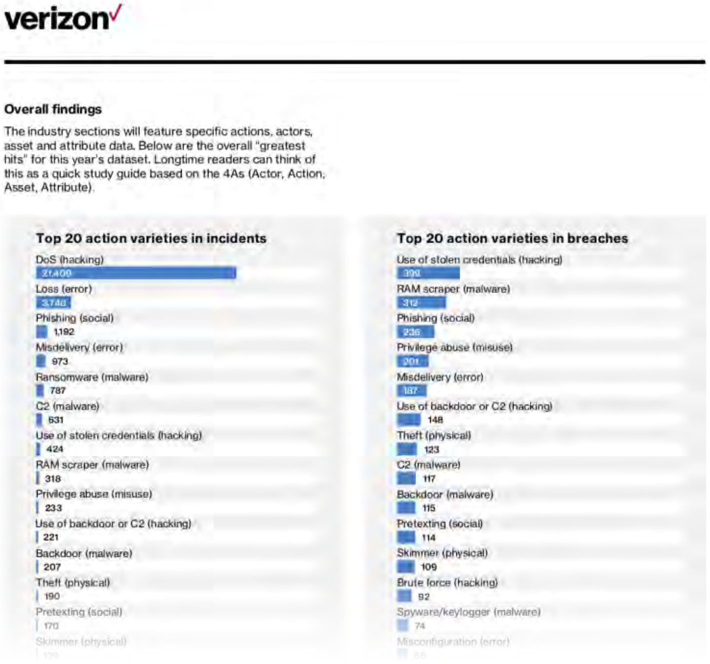
Threat catalogs are useful input to threat modeling

Check out the results from our first round of ATT&CK Evaluations at attackevals.mitre.org!

MITRE ATT&CK™ is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world — by bringing communities together to develop more effective cybersecurity. ATT&CK is open and available to any person or organization for use at no charge.

ATT&CK Matrix for Enterprise



Overall findings

The industry sections will feature specific actions, actors, asset and attribute data. Below are the overall "greatest hits" for this year's dataset. Longtime readers can think of this as a quick study guide based on the 4As (Actor, Action, Asset, Attribute).

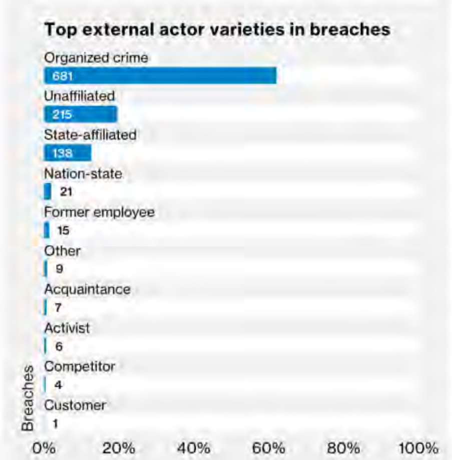


Figure 6. Top external actor varieties within confirmed data breaches (n=1,097)

"2018 Data Breach Investigations Report, 11th Edition." Research Report. Verizon Enterprise, 2018. https://enterprise.verizon.com/resources/reports/DBIR_2018_Report.pdf.

"MITRE ATT&CK^{TM}." Accessed April 1, 2019. https://attack.mitre.org/.

"HITRUST Threat Catalogue." HITRUST, January 31, 2017. https://hitrustalliance.net/threat-catalogue/.

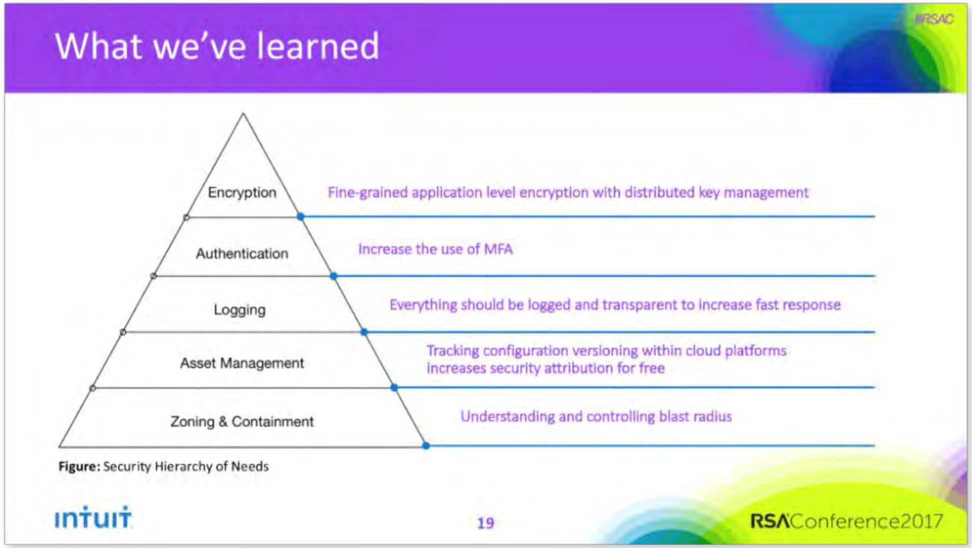
Control catalogs can save on design time

CIS Controls

OWASP ASVS

NIST 800-53 Rev. 4

DoD Instruction 8500.2



#RSAC

What we've learned

Encryption	Fine-grained application level encryption with distributed key management
Authentication	Increase the use of MFA
Logging	Everything should be logged and transparent to increase fast response
Asset Management	Tracking configuration versioning within cloud platforms increases security attribution for free
Zoning & Containment	Understanding and controlling blast radius

Figure: Security Hierarchy of Needs

intuit 19 RSAConference2017

"CIS Controls." Center for Internet Security (CIS). Accessed April 2, 2019. https://learn.cisecurity.org/20-controls-download.

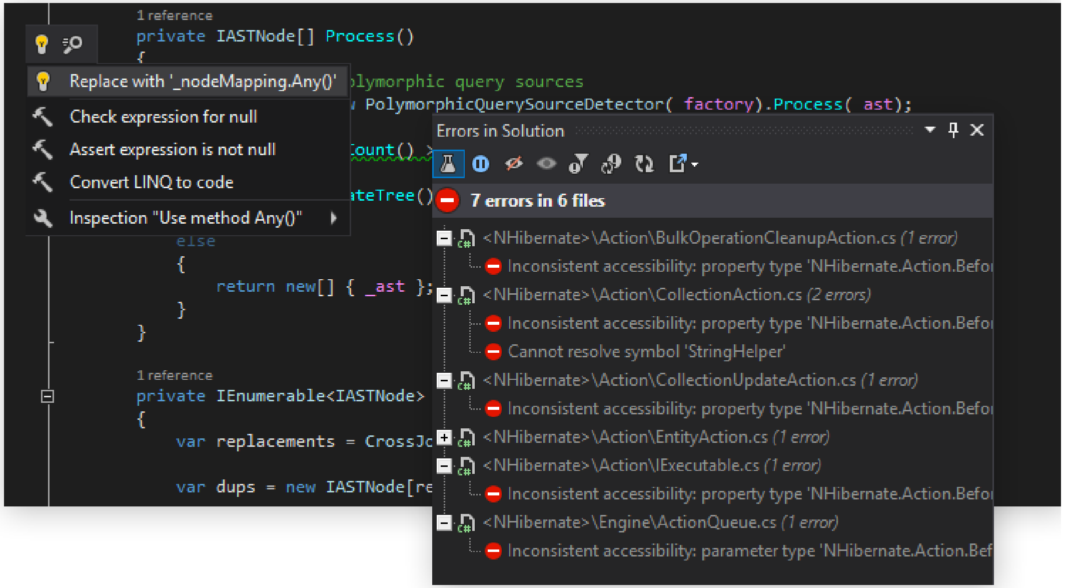
"OWASP Application Security Verification Standard Project." Open Web Application Security Project (OWASP). Accessed April 2, 2019. https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project.

Joint Task Force Transformation Initiative. "Security and Privacy Controls for Federal Information Systems and Organizations." NIST Special Publication. Gaithersburg, MD: National Institute of Standards and Technology, April 2013. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf.

"Complete 8500 Control List." STIG Viewer | Unified Compliance Framework®. Accessed April 2, 2019. https://www.stigviewer.com/controls/8500.

Iacovone, Michele. "Securely Moving Data to the Cloud with Confidence and Customer Focus." presented at the RSA Conference 2017, San Francisco, February 13, 2017. https://published-prd.lanyonevents.com/published/rsaus17/sessionsFiles/4864/CSV-R10F-Securely-Moving-Data-to-the-Cloud-with-Confidence-and-Customer-Focus.pdf.

Code & build



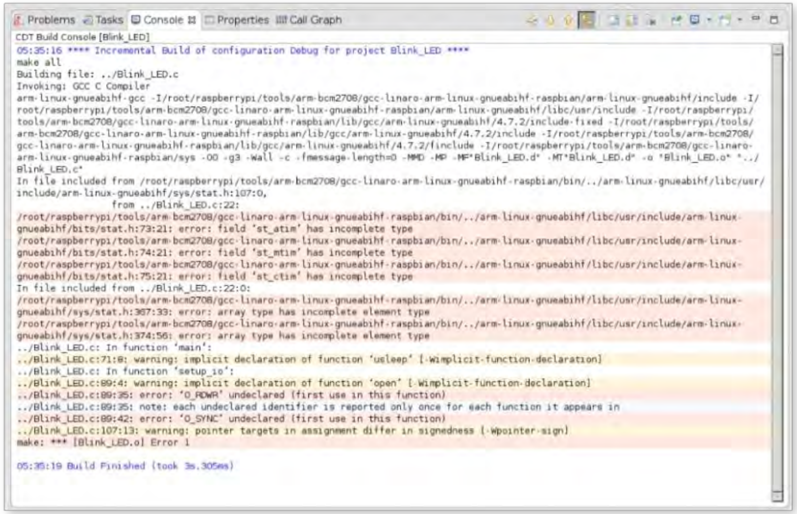
Replace with '_nodeMapping.Any()'
Check expression for null
Assert expression is not null
Convert LINQ to code
Inspection "Use method Any()"

```
1 reference
private IASTNode[] Process()
{
                        lymorphic query sources
                        PolymorphicQuerySourceDetector( factory).Process( ast);
                        ount() >
                        ateTree()
          else
          {
              return new[] { _ast };
          }
    }

1 reference
private IEnumerable<IASTNode>
{
    var replacements = CrossJo

    var dups = new IASTNode[re
```

Errors in Solution

7 errors in 6 files

- <NHibernate>\Action\BulkOperationCleanupAction.cs *(1 error)*
  - Inconsistent accessibility: property type 'NHibernate.Action.Befo
- <NHibernate>\Action\CollectionAction.cs *(2 errors)*
  - Inconsistent accessibility: property type 'NHibernate.Action.Befo
  - Cannot resolve symbol 'StringHelper'
- <NHibernate>\Action\CollectionUpdateAction.cs *(1 error)*
  - Inconsistent accessibility: property type 'NHibernate.Action.Befo
- <NHibernate>\Action\EntityAction.cs *(1 error)*
- <NHibernate>\Action\IExecutable.cs *(1 error)*
  - Inconsistent accessibility: property type 'NHibernate.Action.Befo
- <NHibernate>\Engine\ActionQueue.cs *(1 error)*
  - Inconsistent accessibility: parameter type 'NHibernate.Action.Bef



CDT Build Console [Blink_LED]

```
05:35:16 **** Incremental Build of configuration Debug for project Blink_LED ****
make all
Building file: ../Blink_LED.c
Invoking: GCC C Compiler
arm-linux-gnueabihf-gcc -I/root/raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/arm-linux-gnueabihf/include -I/
root/raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/arm-linux-gnueabihf/libc/usr/include -I/root/raspberrypi/
tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/lib/gcc/arm-linux-gnueabihf/4.7.2/include-fixed -I/root/raspberrypi/tools/
arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/lib/gcc/arm-linux-gnueabihf/4.7.2/include -I/root/raspberrypi/tools/arm-bcm2708/
gcc-linaro-arm-linux-gnueabihf-raspbian/lib/gcc/arm-linux-gnueabihf/4.7.2/finclude -I/root/raspberrypi/tools/arm-bcm2708/gcc-linaro-
arm-linux-gnueabihf-raspbian/sys -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"Blink_LED.d" -MT"Blink_LED.d" -o "Blink_LED.o" "../
Blink_LED.c"
In file included from /root/raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/../arm-linux-gnueabihf/libc/usr/
include/arm-linux-gnueabihf/sys/stat.h:107:0,
                 from ../Blink_LED.c:22:
/root/raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/../arm-linux-gnueabihf/libc/usr/include/arm-linux-
gnueabihf/bits/stat.h:73:21: error: field 'st_atim' has incomplete type
/root/raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/../arm-linux-gnueabihf/libc/usr/include/arm-linux-
gnueabihf/bits/stat.h:74:21: error: field 'st_mtim' has incomplete type
/root/raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/../arm-linux-gnueabihf/libc/usr/include/arm-linux-
gnueabihf/bits/stat.h:75:21: error: field 'st_ctim' has incomplete type
In file included from ../Blink_LED.c:22:0:
/root/raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/../arm-linux-gnueabihf/libc/usr/include/arm-linux-
gnueabihf/sys/stat.h:367:33: error: array type has incomplete element type
/root/raspberrypi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin/../arm-linux-gnueabihf/libc/usr/include/arm-linux-
gnueabihf/sys/stat.h:374:56: error: array type has incomplete element type
../Blink_LED.c: In function 'main':
../Blink_LED.c:71:8: warning: implicit declaration of function 'usleep' [-Wimplicit-function-declaration]
../Blink_LED.c: In function 'setup_io':
../Blink_LED.c:89:4: warning: implicit declaration of function 'open' [-Wimplicit-function-declaration]
../Blink_LED.c:89:35: error: 'O_RDWR' undeclared (first use in this function)
../Blink_LED.c:89:35: note: each undeclared identifier is reported only once for each function it appears in
../Blink_LED.c:89:42: error: 'O_SYNC' undeclared (first use in this function)
../Blink_LED.c:107:13: warning: pointer targets in assignment differ in signedness [-Wpointer-sign]
make: *** [Blink_LED.o] Error 1

05:35:19 Build Finished (took 3s.305ms)
```

Checklists and analyzers improve code quality

Apply commonly needed practices

- Sanitize/validate all inputs
- Parametrize queries
- Prevent XML external entities
- Prevent cross-site scripting (XSS)

Use static analyzers

- Enable compiler warnings
- Adhere to a style guide
- Find analyzers that fit your needs with Kompar (https://kompar.tools)

Belk, Mark, Matt Coles, Cassio Goldschmidt, Michael Howard, Kyle Randolph, Mikko Saario, Reeny Sondhi, Izar Tarandach, Antti Vähä-Sipilä, and Yonko Yonchevv. "Fundamental Practices for Secure Software Development." Edited by Stacy Simpson. Software Assurance Forum for Excellence in Code (SAFECode), February 8, 2011. https://safecode.org/publication/SAFECode_Dev_Practices0211.pdf.

OWASP Cheat Sheet Series. 2018. Reprint, OWASP, 2019. https://github.com/OWASP/CheatSheetSeries.

"OWASP Top Ten Project." Open Web Application Security Project (OWASP), November 2017. https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

"CWE/SANS Top 25 Most Dangerous Software Errors." SANS Institute, June 27, 2011. https://www.sans.org/top25-software-errors.

Code reviews are an effective way to catch bugs

Code reviews answer two questions:

- Is it the correct code?
- Is the code correct?

Typical workflow

1. Developer edits code
   - Usually in a feature branch
   - Not for longer than a couple of weeks
2. Initiates a pull request
3. Someone else reviews code
4. Request changes, or approves merge

“What Are the Best Code Review Tools?” Quora. Accessed April 2, 2019. https://www.quora.com/What-are-the-best-code-review-tools.

“Best Practices for Code Review.” SmartBear Software, 2019. https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/.

“Trunk-Based Development or Pull Requests - Why Not Both?” Jimmy Bogard, December 6, 2017. https://jimmybogard.com/.

Secure configuration

Third-party component configuration guidance

- CIS Benchmarks
- DISA Security Technical Implementation Guides (STIGs)

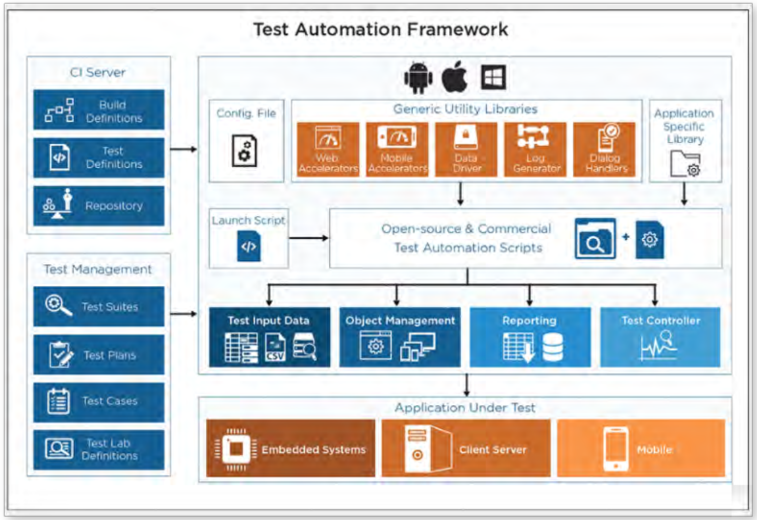
Manage components with known vulnerabilities

- Software composition analysis
  - Dependency-Track, Black Duck, WhiteHat, CA Veracode, etc.
- Apply vendor security updates

“CIS Benchmarks™.” Center for Internet Security (CIS). Accessed April 2, 2019. https://www.cisecurity.org/cis-benchmarks/.

“Complete STIG List.” STIG Viewer | Unified Compliance Framework®. Accessed April 2, 2019. https://www.stigviewer.com/stigs.

Test



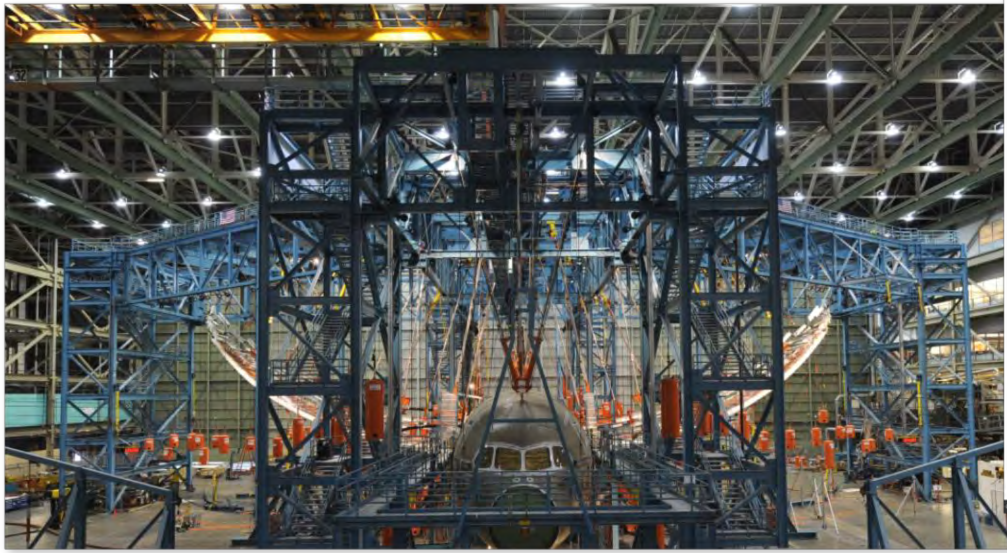
Projects » Jenkins

Bulk Operations for the 7,281 matching findings

Displaying all findings

Finding count 7,281 / 7,281

Displaying 1 to 50 of 7,281 Findings



Paur, Jason. “Boeing 787 Passes Incredible Wing Flex Test.” *Wired*, March 29, 2010. https://www.wired.com/2010/03/boeing-787-passes-incredible-wing-flex-test/.

There are multiple types of testing to apply

Unit

Integration

Static analysis

Dynamic/interactive application security testing

Abuse cases

Fuzz/robustness

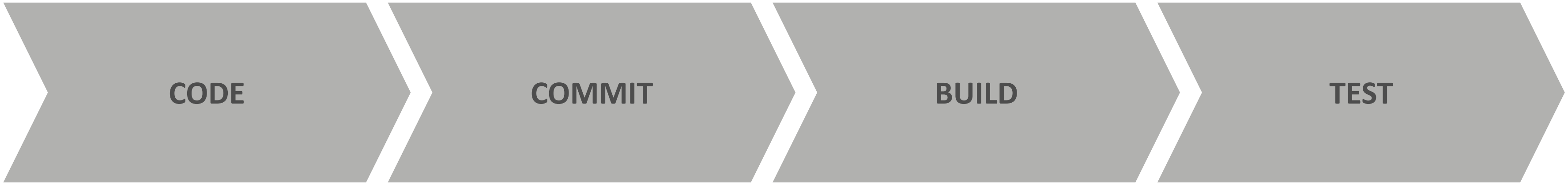
Infrastructure security

Penetration

Atlassian. “The Different Types of Testing in Software.” Atlassian. Accessed April 1, 2019. https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing.

Secure Decisions. “OWASP Attack Surface Detector Project.” Open Web Application Security Project (OWASP), October 2018. https://www.owasp.org/index.php/OWASP_Attack_Surface_Detector_Project.

Selectively apply testing by pipeline stage



CODE

COMMIT

BUILD

TEST

- Lightweight static analysis (linters, style checkers, etc.)
- Unit tests

Automated

- Lightweight static analysis (linters, style checkers, etc.)

Manual

- Code review

Synchronous

- Unit tests
- Integration tests
- Midweight static analysis
- Software composition analysis (SCA)

Synchronous

- Configuration compliance
- Abuse case

Asynchronous

- Dynamic/interactive application security testing
- Heavyweight static analysis
- Fuzz/robustness
- Penetration testing

> Synchronous tests should complete in a reasonable amount of time and the results used to decide whether to break the build. Do not go live with a build if it has serious quality errors based on synchronous tests, *but* many organizations will go live with known security vulnerabilities in their applications.
>
> Cornell, Dan. "Effective Application Security Testing in DevOps Pipelines." Denim Group, December 14, 2016. https://www.denimgroup.com/resources/blog/2016/12/effective-application-security-testing-in-devops-pipelines/.

Application security management systems consolidate testing findings

Application security testing will generate lots of data in different formats and nomenclatures

- Collect, normalize, and deduplicate with a management system
  - E.g., Code Dx, ThreadFix, Defect Dojo, Kenna Security, etc.

Release & deploy

Deployment is a high privilege

Limit who is authorized to deploy builds

Audit all deployment activity

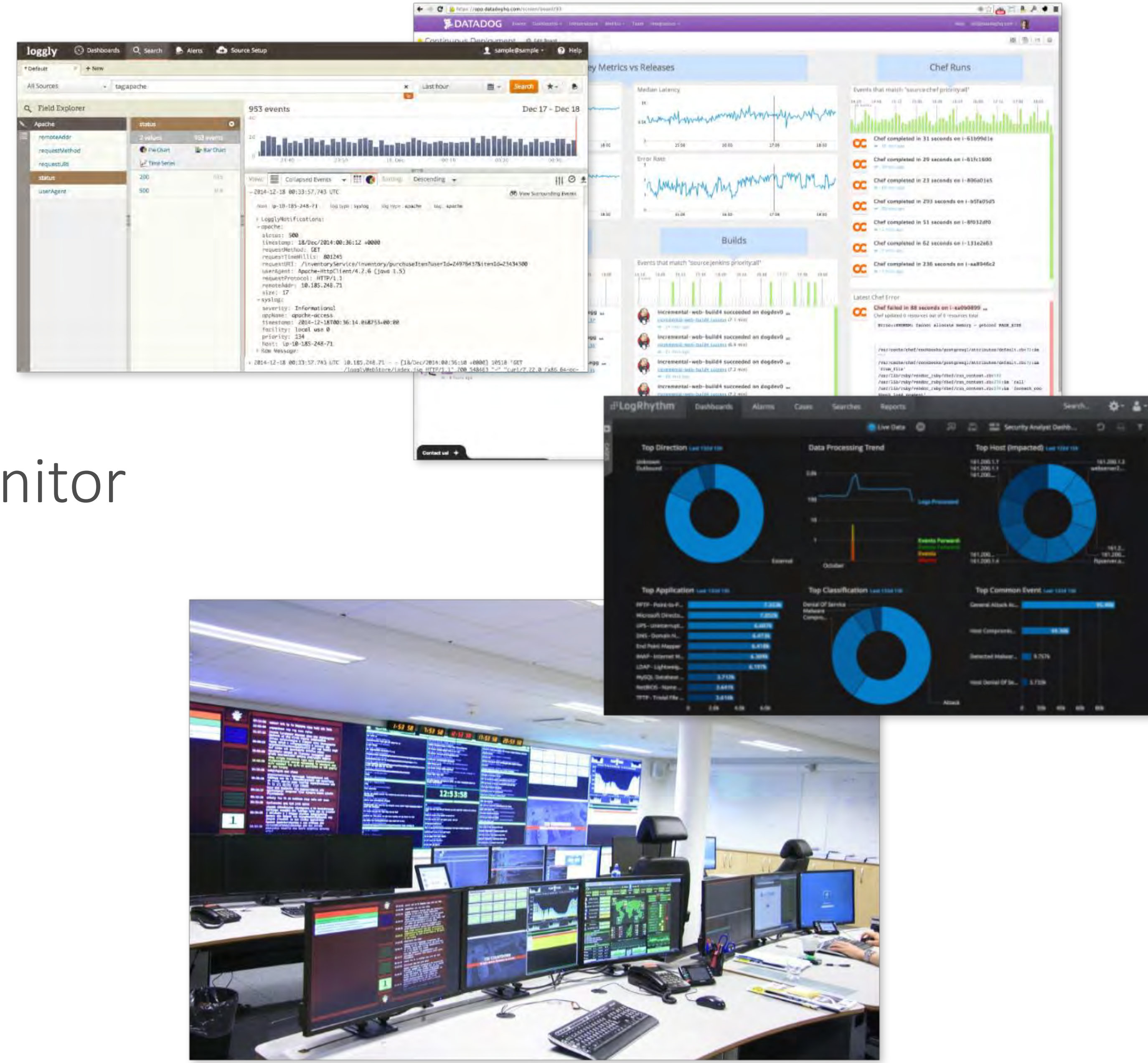
Consider additional technical controls

- Binary authorization

"Deploy Only What You Trust: Introducing Binary Authorization for Google Kubernetes Engine." Google Cloud Blog. Accessed April 2, 2019. https://cloud.google.com/blog/products/identity-security/deploy-only-what-you-trust-introducing-binary-authorization-for-google-kubernetes-engine/.

SECURE DECISIONS
A DIVISION OF APPLIED VISIONS, INC.

Operate & monitor



```
- 2014-12-18 00:33:57.743 UTC
  host: ip-10-185-248-71   log type: syslog   log type: apache   tag: apache
  > LogglyNotifications:
  - apache:
    status: 500
    timestamp: 18/Dec/2014:00:36:12 +0000
    requestMethod: GET
    requestTimeMillis: 801245
    requestURI: /inventoryService/inventory/purchaseItem?userId=24978437&itemId=23434300
    userAgent: Apache-HttpClient/4.2.6 (java 1.5)
    requestProtocol: HTTP/1.1
    remoteAddr: 10.185.248.71
    size: 17
  - syslog:
    severity: Informational
    appName: apache-access
    timestamp: 2014-12-18T00:36:14.068753+00:00
    facility: local use 0
    priority: 134
    host: ip-10-185-248-71
  > Raw Message:
> 2014-12-18 00:33:57.743 UTC 10.185.248.71 - - [18/Dec/2014:00:36:18 +0000] 10518 "GET
    /logglyWebStore/index.jsp HTTP/1.1" 200 548463 "-" "curl/7.22.0 (x86_64-pc-
```

Events that match "source:chef priority:all"

Chef completed in 31 seconds on i-61b99d1e

Chef completed in 29 seconds on i-61fc1600

Chef completed in 23 seconds on i-806a01e5

Chef completed in 293 seconds on i-b5fa05d5

Chef completed in 51 seconds on i-8f012df0

Chef completed in 62 seconds on i-131e2e63

Chef completed in 236 seconds on i-aa8946c2

Chef failed in 88 seconds on i-ea0b0899

Chef updated 0 resources out of 0 resources total

Events that match "source:jenkins priority:all"

incremental-web-build4 succeeded on dogdev0

incremental-web-build4 succeeded on dogdev0

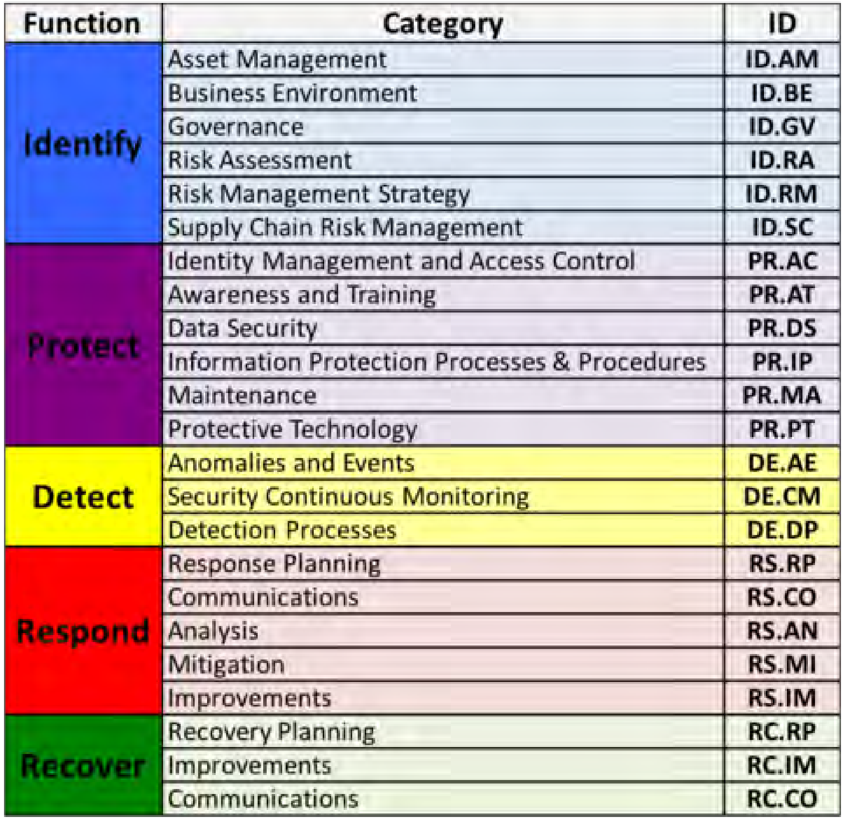
incremental-web-build4 succeeded on dogdev0

incremental-web-build4 succeeded on dogdev0

NIST decomposes cybersecurity into five functions

Operations focuses on

- Detect
- Respond
- Recover



Function	Category	ID
Identify	Asset Management	ID.AM
	Business Environment	ID.BE
	Governance	ID.GV
	Risk Assessment	ID.RA
	Risk Management Strategy	ID.RM
	Supply Chain Risk Management	ID.SC
Protect	Identity Management and Access Control	PR.AC
	Awareness and Training	PR.AT
	Data Security	PR.DS
	Information Protection Processes & Procedures	PR.IP
	Maintenance	PR.MA
	Protective Technology	PR.PT
Detect	Anomalies and Events	DE.AE
	Security Continuous Monitoring	DE.CM
	Detection Processes	DE.DP
Respond	Response Planning	RS.RP
	Communications	RS.CO
	Analysis	RS.AN
	Mitigation	RS.MI
	Improvements	RS.IM
Recover	Recovery Planning	RC.RP
	Improvements	RC.IM
	Communications	RC.CO

Keller, Nicole. “An Introduction to the Components of the Framework.” NIST Cybersecurity Framework, February 6, 2018. https://www.nist.gov/cyberframework/online-learning/components-framework.

Three key takeaways from today’s talk

1. Security is perfectly compatible with DevOps
2. There is no silver bullet to achieve security
   - Results from hundreds of smaller decisions and actions
   - Coordinated application of people, process, and technology
3. There are many great public resources to support learning
   - Citations & links included

Contact information

Chris Horn
Principal consultant,
Product strategy & development

chris.horn@securedecisions.com

https://avi.com
https://securedecisions.com
https://codedx.com

appliedvisions

SECURE DECISIONS
A DIVISION OF APPLIED VISIONS, INC.

CodeDx