RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: **CSV-T07**

THE EMERGENT CLOUD SECURITY TOOLCHAIN FOR CI/CD

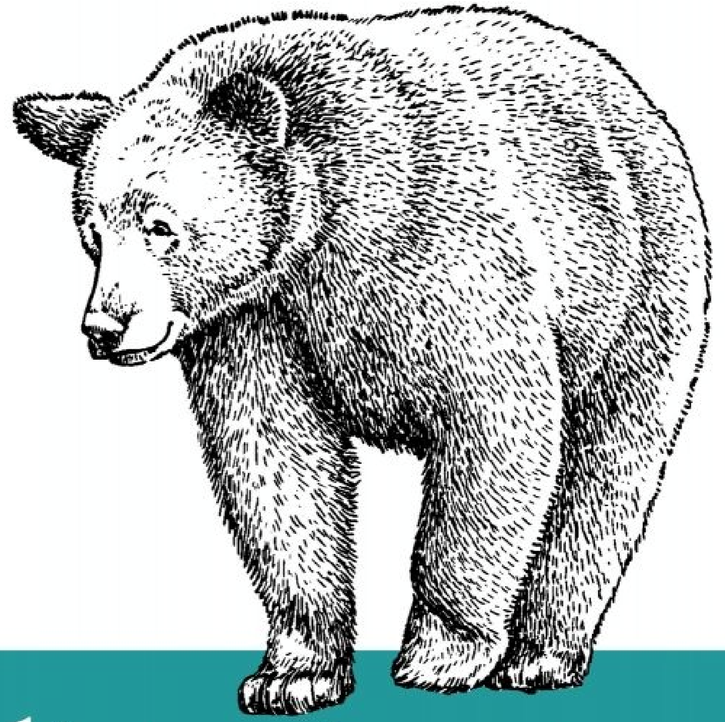
James Wickett

Head of Research
Signal Sciences
@wickett

@WICKETT

- Head of Research @ Signal Sciences
- Organizer of DevOps Days Austin
- lynda.com author on DevOps and Security Courses
- Blog at theagileadmin.com and labs.signalsciences.com

Paying $1,500 to browse Twitter and hang out on Slack



Half-listening to Conference Talks

In Depth

Signal Sciences

RSAConference2018

O RLY?

@ThePracticalDev

Get the slides now!

james@signalsciences.com

Questions on my Mind

- Can Security as an Industry Rise to the Demands of DevOps?
- Is the DevOps culture able to handle security and all of our baggage?
- Will security destroy the DevOps Culture?

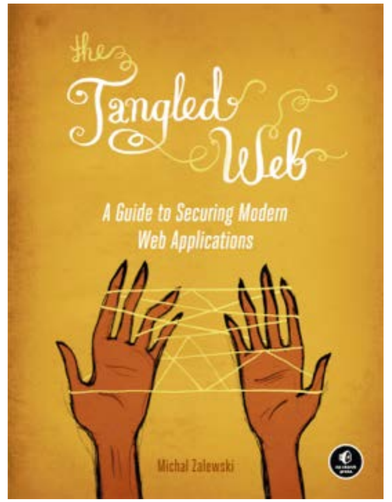
SECURITY IS IN CRISIS

This may be hard to see at RSA, all looks well and good

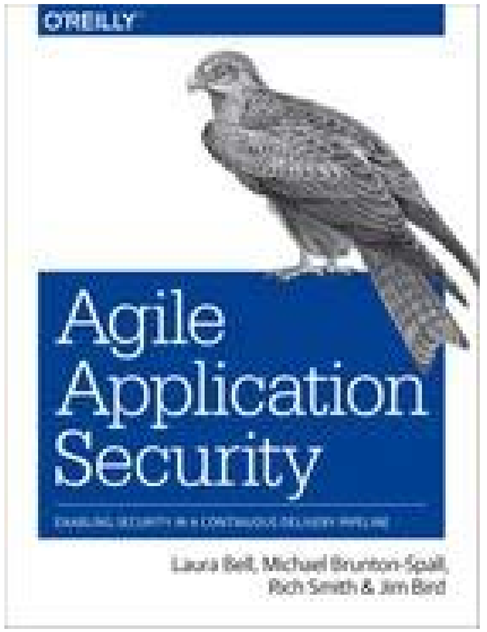
Companies are spending a great deal on security, but we read of massive computer-related attacks. Clearly something is wrong.
The root of the problem is twofold:
we're **protecting the wrong things**,
and **we're hurting productivity** in the process.

THINKING SECURITY, STEVEN M. BELLOVIN 2015

[Security by risk assessment] introduces a dangerous fallacy: that structured inadequacy is almost as good as adequacy and that underfunded security efforts plus risk management are about as good as properly funded security work



many security teams work with a worldview where their goal is to inhibit change as much as possible



“SECURITY PREFERS A SYSTEM POWERED OFF AND UNPLUGGED”

- DEVELOPER

“…THOSE STUPID DEVELOPERS”

- SECURITY PERSON

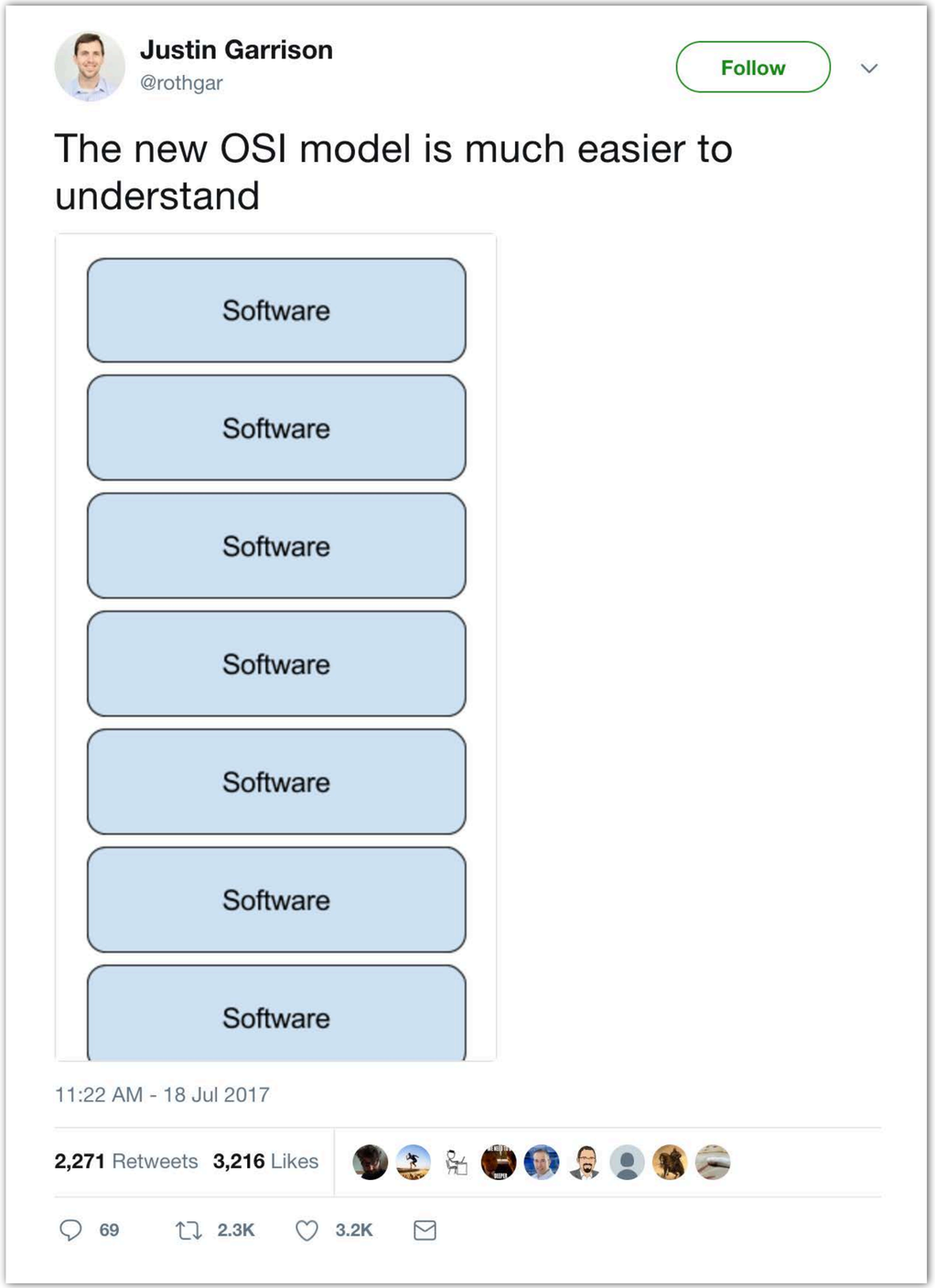
Security must Change or Die

A large percentage of the companies on the expo floor will not there in 5 years @rmogull #RSAC2017

5:51 PM - 14 Feb 2017

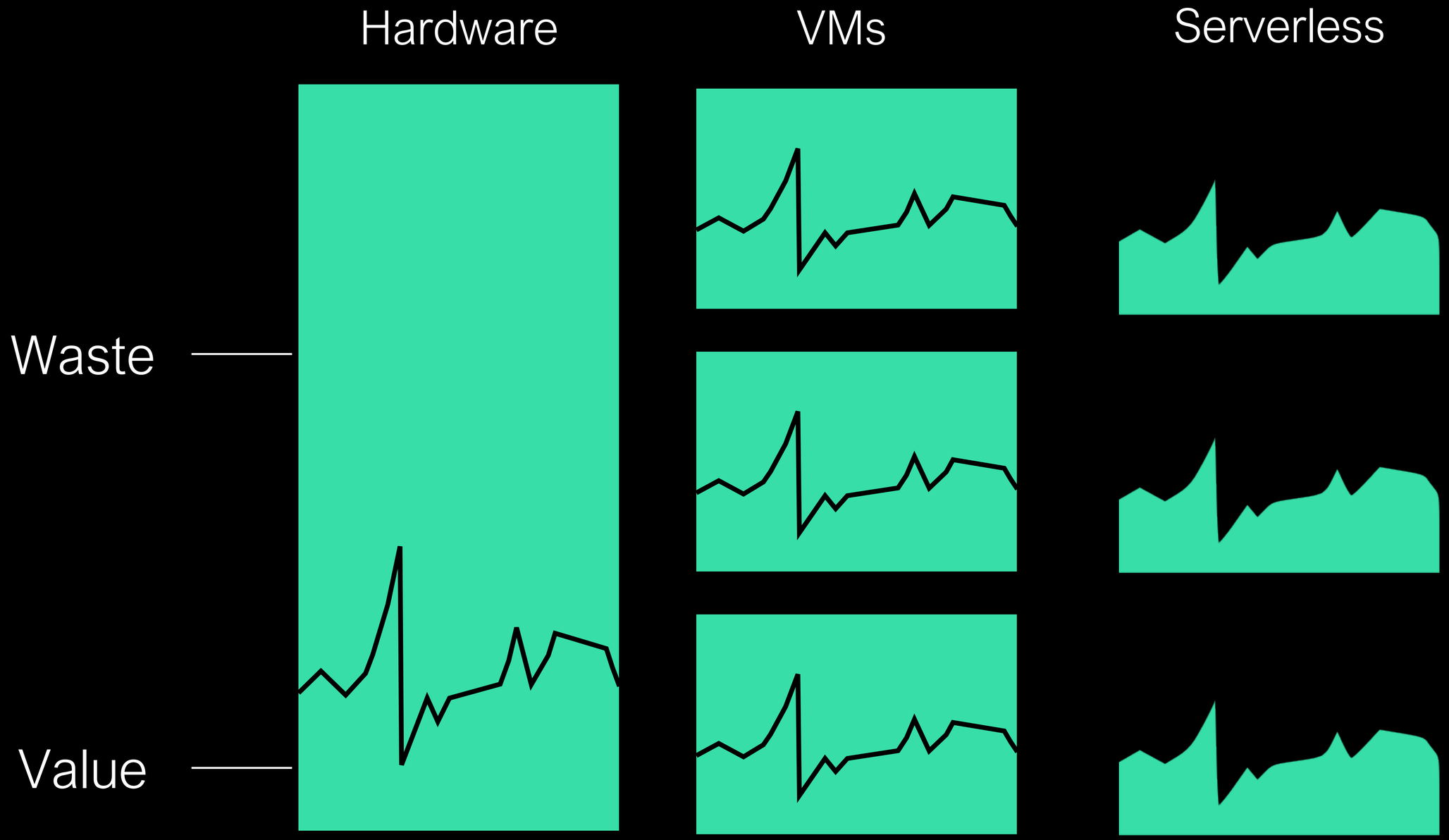
2

THE WORLD HAS CHANGED


Justin Garrison
@rothgar
Follow
The new OSI model is much easier to understand
Software
Software
Software
Software
Software
Software
Software
11:22 AM - 18 Jul 2017
2,271 Retweets 3,216 Likes
69
2.3K
3.2K

Serverless encourages functions as deploy units and run as one-time*, read-only containers*, coupled with third party services that allow running end-to-end applications without worrying about system operation.

* - yes, we know there is container reuse and writability



Signal Sciences

Inspiration from @adrianco

Read-only containers and serverless shift the security story to almost 100% application security

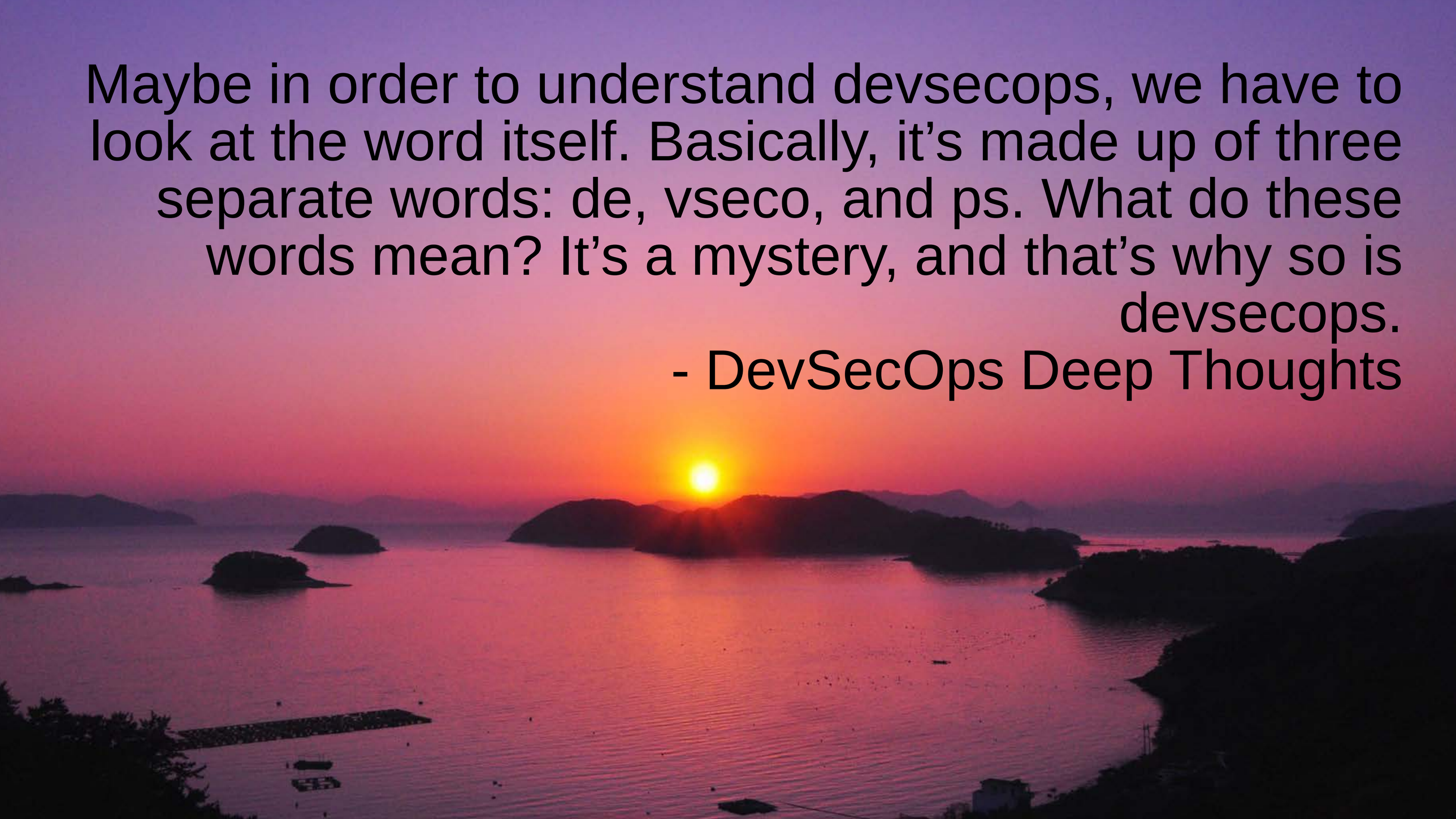
DEVSECOPS TO THE RESCUE!

Signal Sciences

@WICKETT

What is DevSecOps

DevSecOps Deep Thoughts



Maybe in order to understand devsecops, we have to look at the word itself. Basically, it’s made up of three separate words: de, vseco, and ps. What do these words mean? It’s a mystery, and that’s why so is devsecops.

- DevSecOps Deep Thoughts

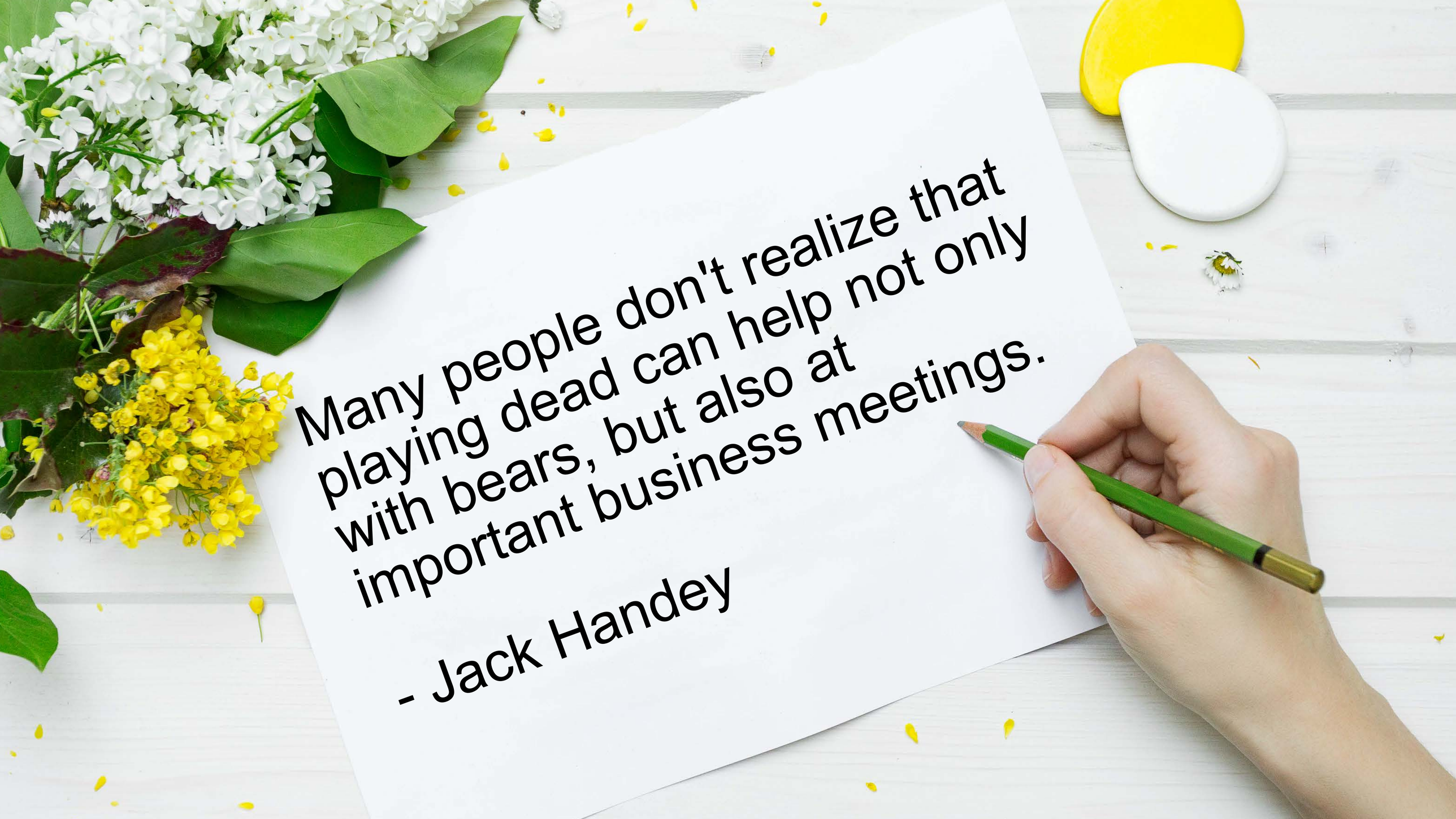
Whenever someone asks me to define devsecops, I usually think for a minute, then I spin around and pin the guy's arm behind his back. NOW who's asking the questions?

- DevSecOps Deep Thoughts

Shoutout to the @TheJewberwocky the original DevOps Deep Thoughts

The original DevOps Deep Thoughts were created by the hilarious and awesome Josh Zimmerman (@TheJewberwocky) as Not Jack Handey which is parody of Deep Thoughts by Jack Handey.

These DevSecOps Deep Thoughts are not nearly as funny nor deep, but hey what do you expect of a parody of a parody?


Many people don't realize that playing dead can help not only with bears, but also at important business meetings.
- Jack Handey

High performing orgs achieve quality by incorporating security (and security teams) into the delivery process

2016 State of DevOps Report

Signal Sciences

@WICKETT

DevSecOps is a cultural movement that furthers the movements of Agile and DevOps into Security

CULTURE IS THE MOST IMPORTANT ASPECT TO DEVOPS SUCCEEDING IN THE ENTERPRISE

- PATRICK DEBOIS

Dev:Ops

10:1

Dev:Ops:Sec

100:10:1

4 Keys to Culture

- Mutual Understanding
- Shared Language
- Shared Views
- Collaborative Tooling

A security team who embraces openness about what it does and why, spreads understanding.
- Rich Smith

EMERGING PATTERNS FOR SECURITY IN A CI/CD WORLD

Signal Sciences

@WICKETT


OLD PATH VS. NEW PATH
Embrace Secrecy
Create Feedback Loops
Just Pass Audit!
Compliance adds Value
Enforce Stability
Create Chaos
Build a Wall
Zero Trust Networks
Slow Validation
Fast and Non-blocking
Certainty Testing
Adversity Testing
Test when Done
Shift Left
Process Driven
The Paved Road

SECURITY TOOLCHAIN FOR CI/CD

Signal Sciences

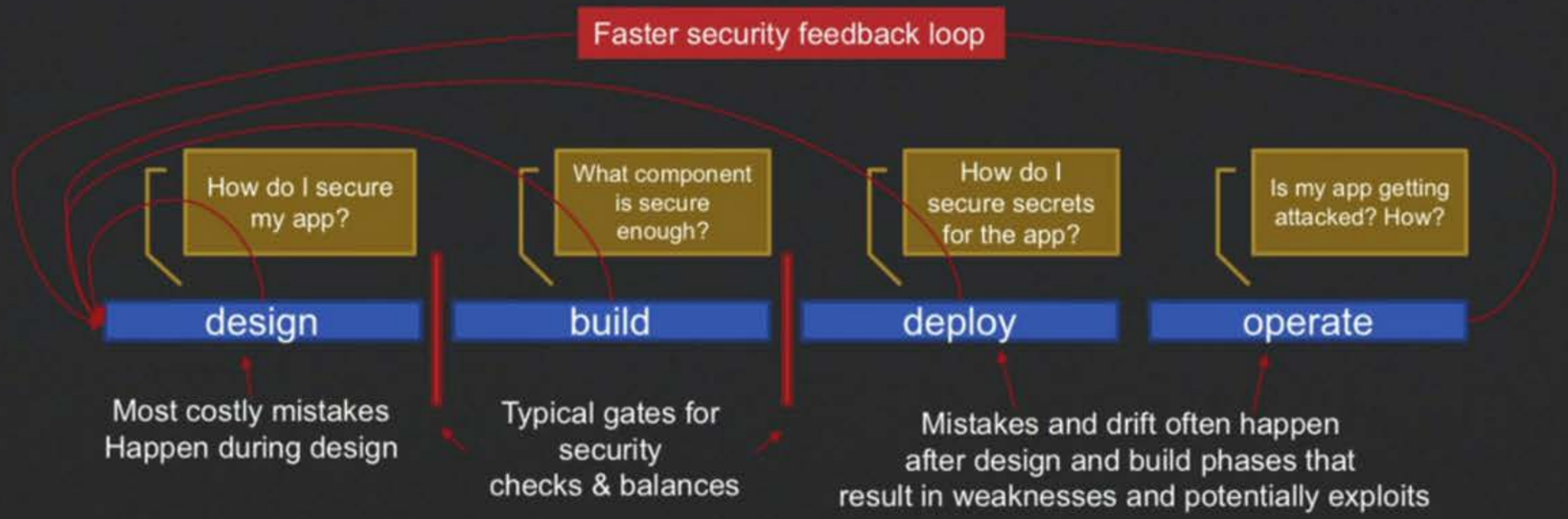
@WICKETT

Software Delivery Pipeline

Develop | **Inherit** | **Build** | **Deploy** | **Operate**

Secure Software Supply Chain

1. Gating processes are not Deming-like
2. Security is a design constraint
3. Decisions made by engineering teams
4. It's hard to avoid business catastrophes by applying one-size-fits-all strategies
5. Security defects is more like a *security "recall"*



Secure Software Supply Chain presented by Shannon Leitz at DevOps Days Austin 2016.

Develop | Inherit | Build | Deploy | Operate

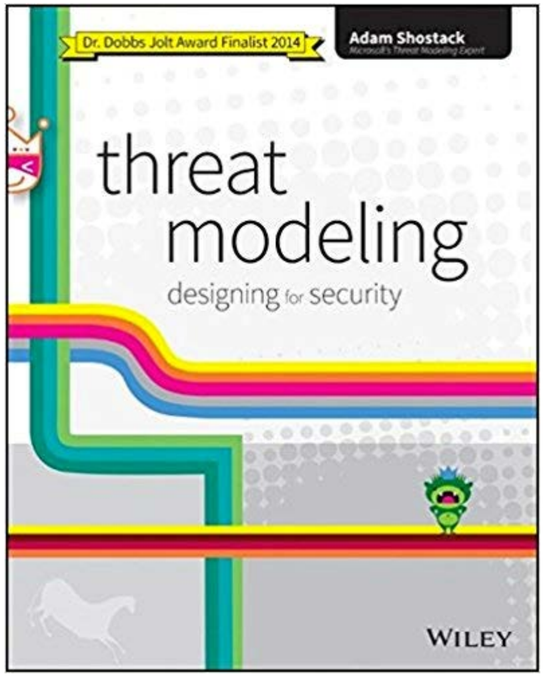
The design and development of an application and its features. Including all the development practices like version control, sprint planning, unit-testing.

Security Activities and Considerations

- **Threat Modeling**
- **Security Stories**
- **Authentication to Push**
- **Development Standards**
- **Peer Review**
- **Static Code Analysis**
- **Unit Tests for Security**

Threat Modeling and Security Stories

- **The Threat Modeling Book by Adam Shostack**
- **OWASP App Threat Modeling Cheat Sheet**
- **Evil User Stories (link)**
- **OWASP Application Security Verification Standard**
- **Mozilla Rapid Risk Assessment (link)**



Develop | Inherit | Build | Deploy | Operate

Development Standards

- **Pre-commit Hooks for Security**
- **Coding Standards (Security and otherwise)**
- **Peer Review**
- **Single Mainline Branch**
- **Linting and Code Hygiene**

Code Standards and Team Tooling

- git-secrets Prevents you from committing passwords and other sensitive information to a git repository. From awslabs. (**link**)

- git-hound Hound is a Git plugin that helps prevent sensitive data from being committed into a repository by sniffing potential commits against PCRE regular expressions. (**link**)

- **gometalinter** or whatever your language of choice (this is a golang example, you will need one for your language)
- **gofmt** formats the code automatically and makes everything look the same, easier for everyone to grok (again, this is specific to lang)

Develop | Inherit | Build | Deploy | Operate

Code Standards and Team Tooling is run on developer laptops and systems, but verified by CI system.

Static Code Analysis!

- Not unfamiliar territory for security!

- Static Application Security Testing (SAST)

- IDE Plugin if Possible

- **Open Source:** Brakeman (Ruby), FindSecurityBugs (Java), Phan (PHP), Go AST (golang)
- **Paid:** Brakeman Pro, Veracode, Fortify, …

Unit Testing for Security

- Unit Testing is the currency of Developers

- JUnit, Rspec, Testing (golang), ….

- Goal is to have security tests being written with other unit tests or whatever testing patterns you use: TDD, BDD, ATDD, …

Questions to Ask

Are the developers testing for security locally before it gets to CI system?

Do we practice good hygiene and coding practices?

Are we developing as a team in trunk with few branches?

Develop | **Inherit** | Build | Deploy | Operate

This is an overlooked phase because it is the most invisible as software dependencies get bundled in and inherited in our own code and upstream.

Develop | **Inherit** | Build | Deploy | Operate

Security Considerations

- This is your real LOC count!

- The Software Delivery Supply Chain

- Publish a Bill of Materials and trace back

- This is not just application dependencies and libraries, but also OS-level (remember shellshock, heartbleed, ..)

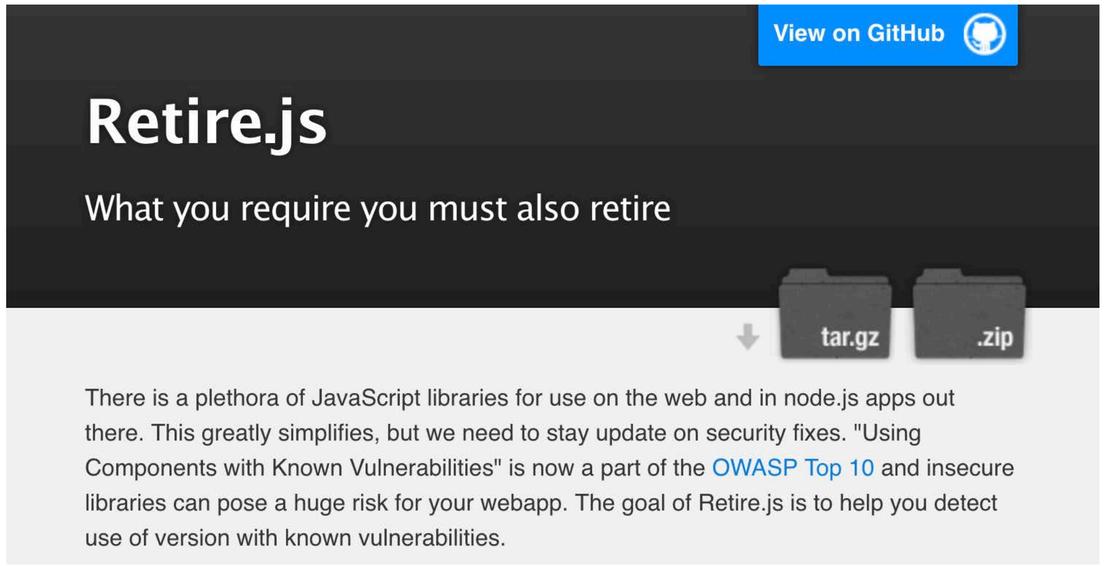
Language Tooling

- bundler-audit - checks for vulnerable versions of gems in your ruby code (link)

- nsp - node security platform (link)

- **Paid options:** Sonatype, BlackDuck, JFrog

- **Retire.js** - known vuln JS libs (link)



Develop | **Inherit** | Build | Deploy | Operate

Containers!

- Over 30% of containers in Docker Hub have high sev vulns (source)

- Open Source: Docker Bench, Clair

- Paid Options: aqua, twistlock

Questions to Ask

What have I bundled into my app that is making vulnerable?

Am I publishing a Bill of Materials with my application?

Develop | Inherit | **Build** | Deploy | Operate

This phase is where the CI build system runs all the build steps and does acceptance testing. Previous testing and tooling gets verified here.

Develop | Inherit | **Build** | Deploy | Operate

Security Considerations

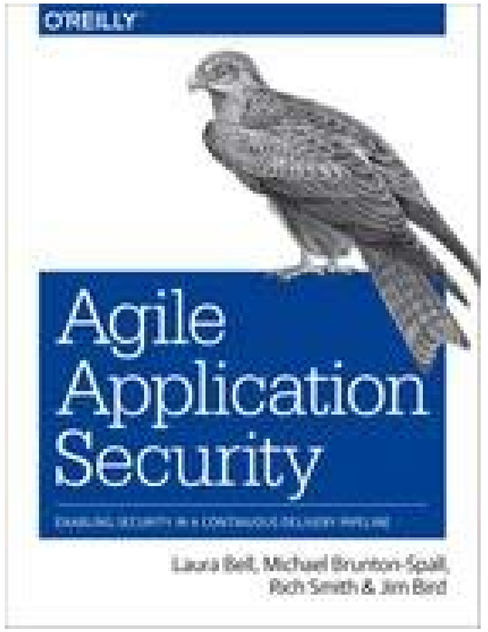
- **Outside-In Security Testing**
- **Infra as Code (Testing)**
- **Dynamic Application Security Testing (DAST)**
- **Compliance on every build!**
- **Cloud provider config as code**
- **Using containers**

Dynamic Application Security Scanners

- These all require tuning and can be difficult to integrate into build pipelines.
- Application Security scanners: Nikto, Arachni, ZAP, sqlmap, xsser, …

- Other - SSLyze, nmap, ssh_scan
- See Kali Linux
- Paid: Qualys, AppScan, BurpSuite, …

The goal should be to come up with a set of automated tests that probe and check security configurations and runtime system behavior for security features that will execute every time the system is built and every time it is deployed.



GAUNTLT

BE MEAN TO YOUR CODE AND LIKE IT

Framework with Security testing written in a natural language that developers, security and operations can understand.

Gauntlt wraps security testing tools but does not install tools

Gauntlt was built to be part of the CI/CD pipeline

Open source, MIT License,

gauntlt.org

Gauntlt Example

What?

Given

When

Then

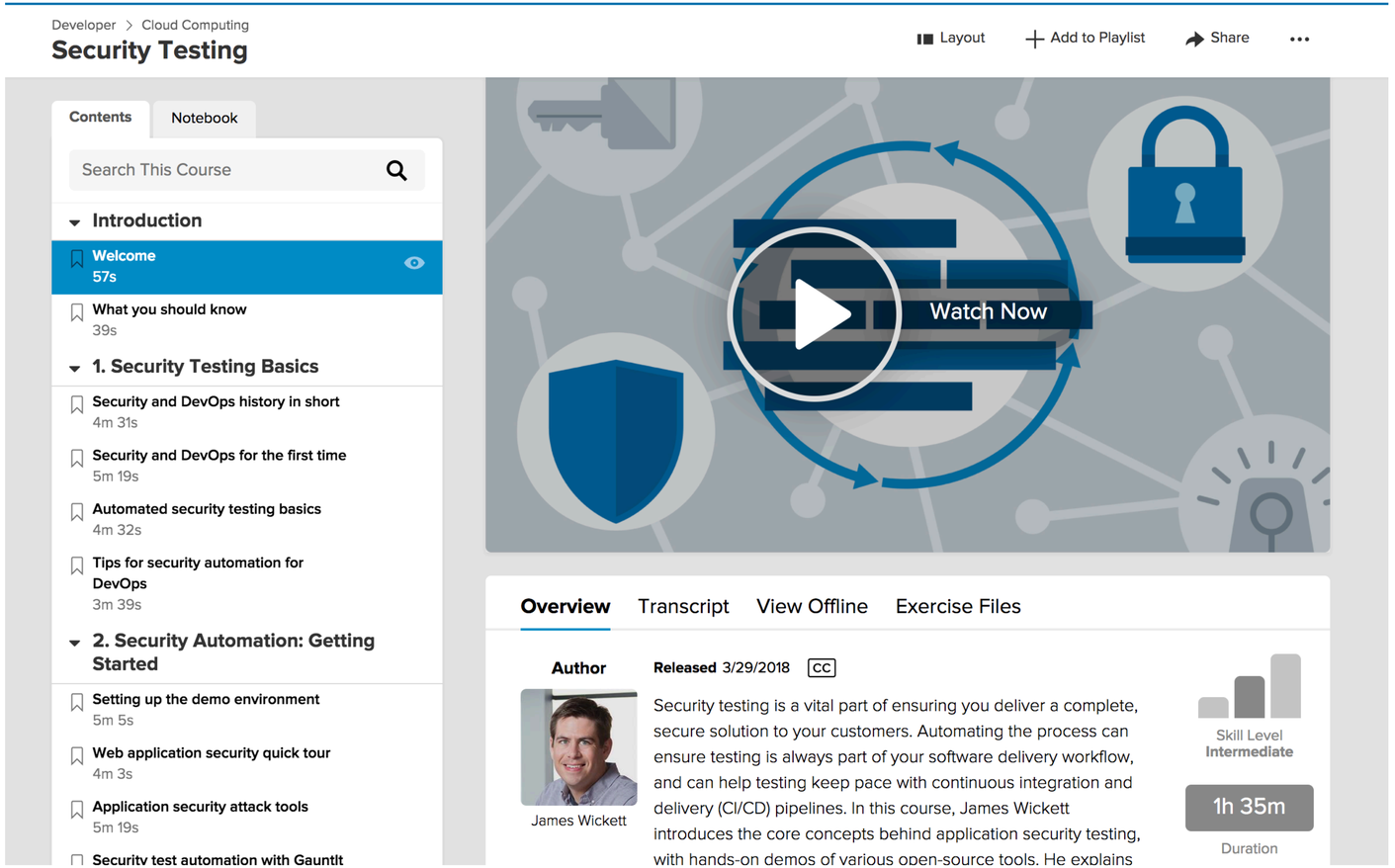
```
@slow @final
Feature: Look for cross site scripting (xss) using arachni
against a URL

Scenario: Using arachni, look for cross site scripting and verify
no issues are found
  Given "arachni" is installed
  And the following profile:
     | name                  | value                          |
     | url                   | http://localhost:8008          |
  When I launch an "arachni" attack with:
  """
  arachni –check=xss* <url>
  """
  Then the output should contain "0 issues were detected."
```

"We have saved millions of dollars using Gauntlt for the largest healthcare industry project."

- Aaron Rinehart, UnitedHealthCare

A Whole Course on Security Testing with Gauntlt



https://www.lynda.com/Software-Development-tutorials/Security-Testing/667367-2.html

Infrastructure and Compliance

- **Test Kitchen** - https://kitchen.ci/
- **Serverspec** - http://serverspec.org/
- **Chef InSpec** - Continuous Compliance Testing (https://www.chef.io/inspec/)

- **Cloud Provider is Infrastructure too**
- **Version and test Cloud Config (e.g. CloudFormation for AWS)**

Questions to Ask

Am I testing for security low hanging fruit?

Am I arming my pipeline with attack tools to exercise my application?

Have I validated the previous two phases of testing in secure build environment?

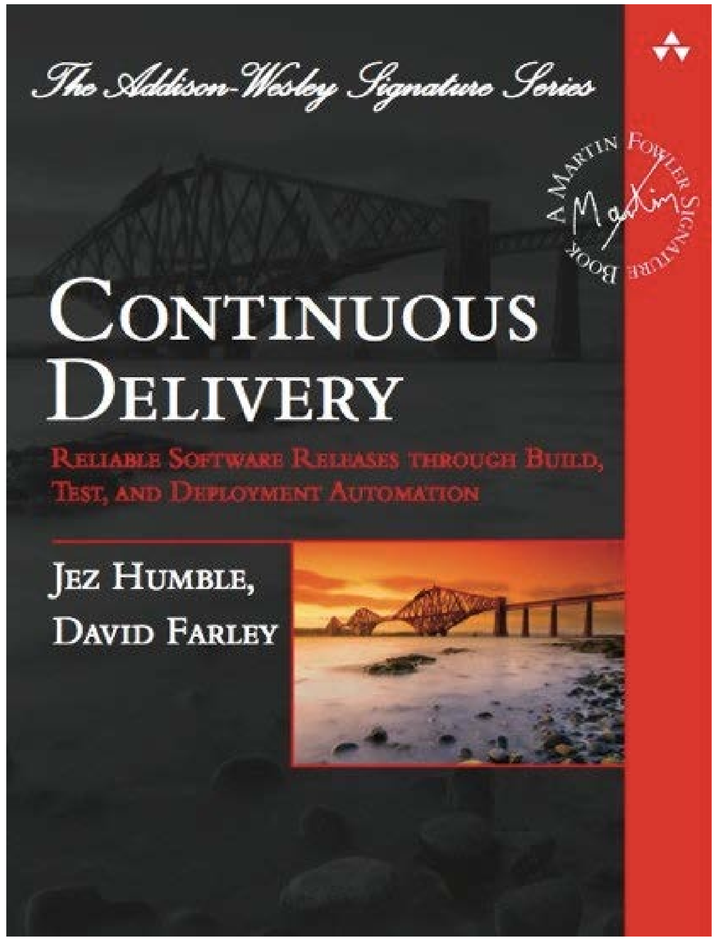
Develop | Inherit | Build | **Deploy** | Operate

The phase where software moves from our testing to where customers are able to operate it for the first time.

Develop | Inherit | Build | **Deploy** | Operate

Security Considerations

- **Watch out for Compliance**
- **Secrets Management**
- **Deploy Accountability**
- **Authorization and Logging**
- **Monitoring Deploys**
- **Infra as Code (Execution)**
- **Repeatable Execution**


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Martin
CONTINUOUS DELIVERY
RELIABLE SOFTWARE RELEASES THROUGH BUILD, TEST, AND DEPLOYMENT AUTOMATION
JEZ HUMBLE,
DAVID FARLEY

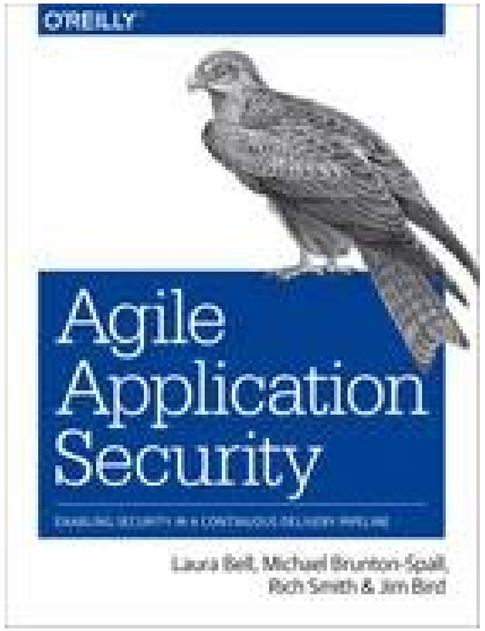
Currently, at Signal Sciences we do about 15 deploys per day

Roughly 10,000 deploys in the last 2.5 yrs

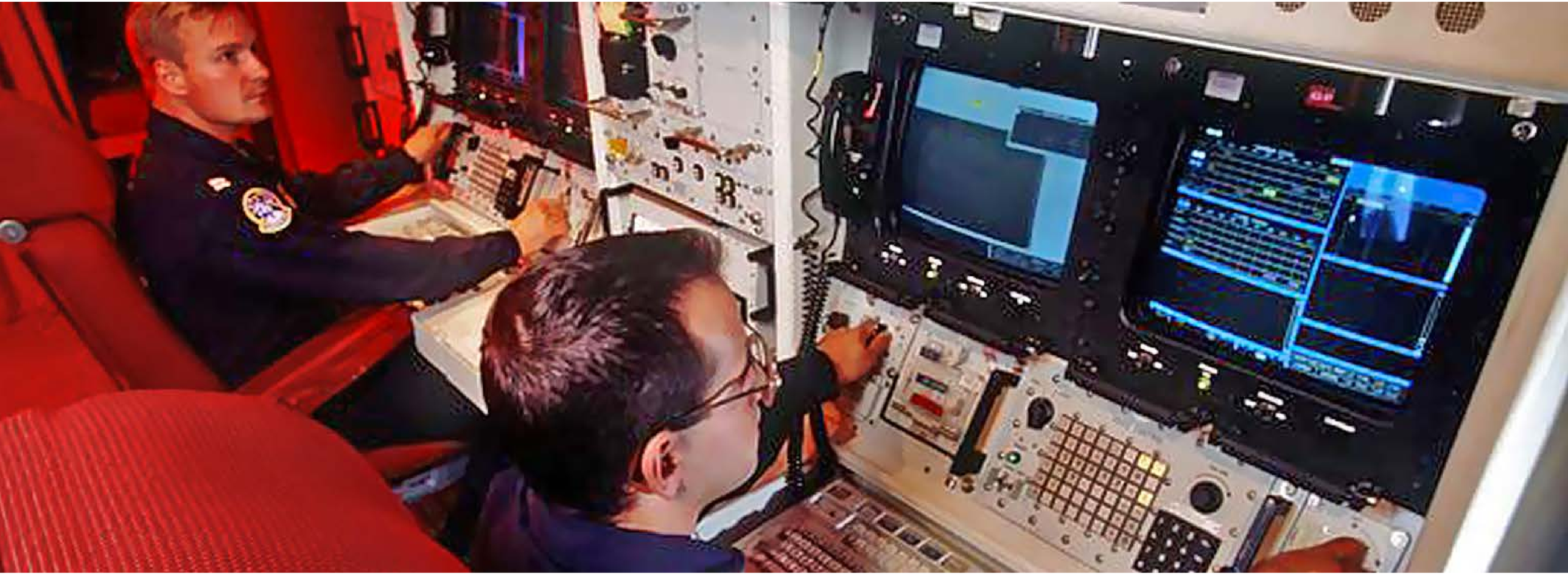
CD is how little you can deploy at a time

We optimized for cycle time—the time from code commit to production

[Deploys] can be treated as standard or routine changes that have been pre-approved by management, and that don’t require a heavyweight change review meeting.



Separation of Duties Considered Harmful



Check out DevOps Audit Defense Toolkit

https://cdn2.hubspot.net/hubfs/228391/Corporate/DevOps_Audit_Defense_Toolkit_v1.0.pd

Dear Auditor,

a love letter to auditors from devops, where we promise to make life better

View My GitHub Profile

Download **ZIP File** | Download **TAR Ball** | View On **GitHub**

Hosted on GitHub Pages — Theme by orderedlist

Dear Auditor,

We realize that we have been changing things in a rapid fashion from Agile and DevOps to Cloud and Containers. Yes, we have been busy, and are having great success delivering faster than ever, with better quality and supporting the business response to competitive pressures. This isn't just icing on the cake, the only sustainable advantage in our industries is the ability to meet customer demands faster, more reliably than our competitors.

With all this growth, we made a mistake, we forgot to bring you along for the ride. That is totally our bad, but we want to make it right. We want to make some new commitments.

- We will bring you along
- We will be fully transparent about our development process
- We do realize that we own the risks
- We will maintain an open channel of discussion to demonstrate to you how we manage risks with our modern development practices

For example, you have told us that you are concerned about "Separation of Duties" in agile and DevOps practices, and we heard you! We think we have a better way to manage this and risks now. Having everything in version control, enforcing peer review for every change, releasing via a secure pipeline, restricting production access, and monitoring unauthorized changes in production systems should address your concern.

The DevOps community has been experimenting quite a bit over the last number of years and common practice represents the collective wisdom across many companies, industries, and countries.

We have compiled a list of audit concerns and documented them in a DevOps Risk Control Matrix with lot of details around the controls, our practices and evidences that are collected to support the control. We hope this matrix provides a way to collaborate.

Please don't misinterpret that we are backing down from speed and providing value, but we are really excited to move forward, together.

XOXO,

The DevOps Community

Monitoring Cloud Config

- Paid Cloud Config security:

Evident.io, ThreatStack,

AlienVault

- Cloud Provider: AWS

CloudTrail, Inspector, GuardDuty

Questions to Ask

What secrets are needed to move my application from development into production?

Am I testing for Compliance on each and every deploy?

Is there a repeatable mechanism to push changes to production?

Develop | Inherit | Build | Deploy | **Operate**

The runtime state of the application, where users interact with or consume the application. Our application in production.


VAYAGIF.COM

Develop | Inherit | Build | Deploy | **Operate**

Security Considerations

- Chaos Engineering and creating stability through instability
- Circuit Break Pattern in use
- Instrumentation and Visualization

- **Application security and service abuse and misuse**
- **Bug Bounties**
- **Red Teaming as a Service**

Now with user-generated content!

Essential

'); DROP TABLE animals;--

O RLY?

@ThePracticalDev

real point of this research. To maximize value from yo

- WHITEPAPER FROM AN UNDISCLOSED WAF VENDOR

Detect what matters

Account takeover attempts

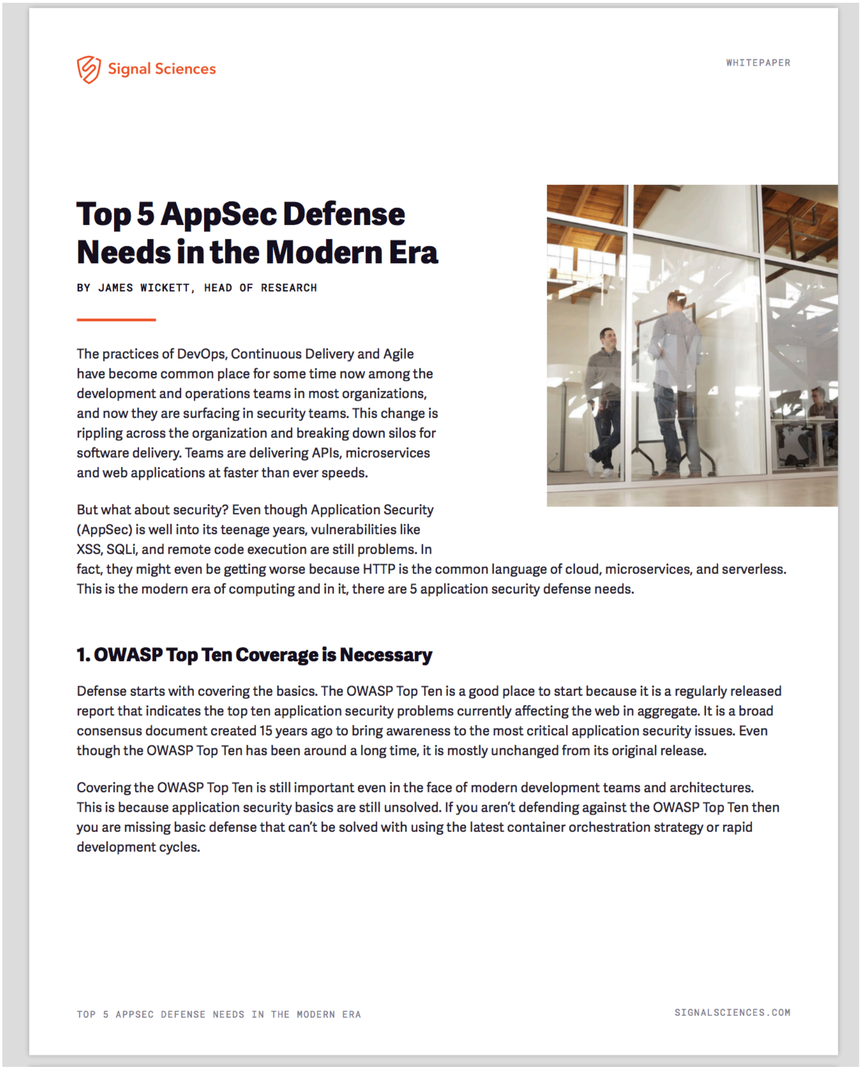
Areas of the site under attack

Most likely vectors of attack

Business logic flows

Abuse and Misuse signals

Free
Guidebook
on AppSec in
Modern Era



Signal Sciences

WHITEPAPER

Top 5 AppSec Defense Needs in the Modern Era

BY JAMES WICKETT, HEAD OF RESEARCH

The practices of DevOps, Continuous Delivery and Agile have become common place for some time now among the development and operations teams in most organizations, and now they are surfacing in security teams. This change is rippling across the organization and breaking down silos for software delivery. Teams are delivering APIs, microservices and web applications at faster than ever speeds.

But what about security? Even though Application Security (AppSec) is well into its teenage years, vulnerabilities like XSS, SQLi, and remote code execution are still problems. In fact, they might even be getting worse because HTTP is the common language of cloud, microservices, and serverless. This is the modern era of computing and in it, there are 5 application security defense needs.

1. OWASP Top Ten Coverage is Necessary

Defense starts with covering the basics. The OWASP Top Ten is a good place to start because it is a regularly released report that indicates the top ten application security problems currently affecting the web in aggregate. It is a broad consensus document created 15 years ago to bring awareness to the most critical application security issues. Even though the OWASP Top Ten has been around a long time, it is mostly unchanged from its original release.

Covering the OWASP Top Ten is still important even in the face of modern development teams and architectures. This is because application security basics are still unsolved. If you aren't defending against the OWASP Top Ten then you are missing basic defense that can't be solved with using the latest container orchestration strategy or rapid development cycles.

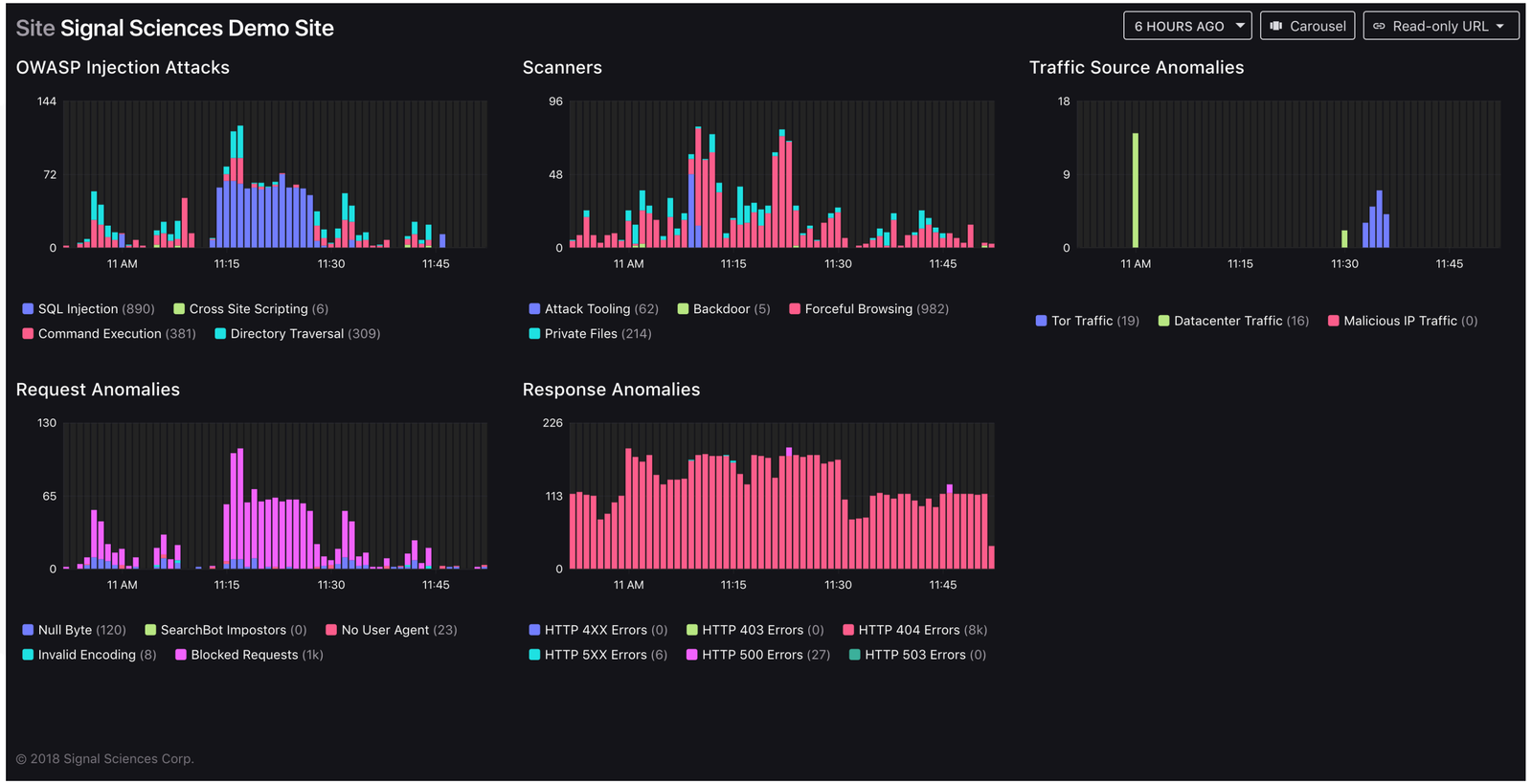
TOP 5 APPSEC DEFENSE NEEDS IN THE MODERN ERA

SIGNALSCIENCES.COM

https://info.signalsciences.com/appsec-defense-needs-top-five

Which is better application attack feedback?

```
se.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:10.0) Gecko/20100101 Fi
refox/10.0" - - - [redacted] - - - - - - - 16951
- - - - [20/Feb/2012:22:32:10 +0000] "GET /images/sprites/buttons-master.png HTT
P/1.1" 304 - "http://[redacted]assets/dist/88166671/css/
modules/buttons-new.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:10.0)
Gecko/20100101 Firefox/10.0" - - - [redacted] - - - - - -
 - 12156
- - - - [20/Feb/2012:22:32:10 +0000] "GET /images/spinners/spinner16.gif HTTP/1.
1" 304 - "http://[redacted]/assets/dist/88166671/css/base
.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:10.0) Gecko/20100101 Fire
fox/10.0" - - - [redacted] - - - - - - - 18810
- - - - [20/Feb/2012:22:32:10 +0000] "GET /assets/dist/88166671/js/convos/thread
s.js HTTP/1.1" 200 61743 "http://[redacted]/conversations?re
f=si_con" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:10.0) Gecko/20100101
Firefox/10.0" - - - [redacted] - - - - - - - 834687
- - - - [20/Feb/2012:22:32:10 +0000] "GET /assets/dist/88166671/js/bootstrap/com
mon.js HTTP/1.1" 200 127238 "http://[redacted]conversations
?ref=si_con" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:10.0) Gecko/201001
01 Firefox/10.0" - - - [redacted] - - - - - - - 928201
- - - - [20/Feb/2012:22:32:11 +0000] "GET /assets/dist/88166671/js/overlays/exte
rnal-link.js HTTP/1.1" 200 487 "http://[redacted]/conversati
ons?ref=si_con" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.7; rv:10.0) Gecko/201
```



Runtime Defense Tooling

- Open Source: modSecurity + ELK To gain application insight monitoring.

- Paid NGWAF / RASP Options: Signal Sciences, Contrast, Prevoty

- **Pro-tip: Avoid adding appsec defense at the CDN**

Red Team Mondays at Intuit

Shannon Lietz

Signal Sciences

@WICKETT

Bug Bounties

- **Roll your own!**
- **Paid Options:** HackerOne, BugCrowd, Synack

Logging Security Telemetry

- **Log All The Things**
- **ELK Stack for Open Source**
- **Paid Options:** Splunk, SumoLogic

The Pragmatic Programmers

Release It!

Second Edition

Design and Deploy
Production-Ready Software

Questions to Ask

Do you know if you are under attack at this current moment?

Do you know what the attackers are going after?

Can I turn on and off services independently if being attacked?

Are we doing Chaos experiments?

SECURITY'S NEW CORE IDEOLOGY

Signal Sciences

@WICKETT

The New Ways

- Empathy and Enablement
- Be Fast and Non-Blocking
- Don’t slow delivery
- Join with continuous testing efforts
- Security testing automated in every phase
- Penetration Testing alongside the Pipeline
- Security provides value through making security normal

Apply What You Have Learned Today

- Next week you should:
  - Identify the who/where/what of your CI/CD Pipeline
- In the first three months following this presentation you should:
  - Create a plan around the five phases and security tooling and practices
  - Implement 1-2 tools in the pipeline
- Within six months you should:
  - Have security in all five phases of the pipeline
  - Answer the maturity questions for each phase

Want the slides and referenced links?

james@signalsciences.com