#RSAC

RSAConference2017

Singapore | 26–28 July | Marina Bay Sands

POWER OF OPPORTUNITY

SESSION ID: LAB-W02

Overcoming the Challenges of Automating Security in a DevOps Environment

Murray Goldschmidt
Chief Operating Officer
Sense of Security
@ITsecurityAU

Michael McKinnon
Director, Commercial Services
Sense of Security
@bigmac

Learning Lab Agenda

Item	Description	Duration
DevSecOps Lab Intro	Introduction and attack demonstration of a DevSecOps Lab Environment on Amazon AWS	30 minutes
App Sec Automation	Challenges with securing first party and third party code, and static and dynamic code scanning	30 minutes
Monitoring & Self-Healing	Implementing continuous monitoring and self-healing	30 minutes
Mitigations & Conclusions	Overview of how the AWS attack we demonstrated could be successfully mitigated.	30 minutes

Automation is Everywhere

WIRED

This Robot Barista Makes a Dang Good Latte

SUBSCRIBE

BUSINESS | CULTURE | DESIGN | GEAR | SCIENCE | SECURITY | TRANSPORTATION

DAVID PIERCE GEAR 01.30.17 8:00 AM

THIS ROBOT BARISTA MAKES A DANG GOOD LATTE

Source: https://www.wired.com/2017/01/cafe-x-robot-barista/

Adversaries are using Automation

C:\Users\[redacted]\DotRansomwareBuilder.exe

```
Ransomware Builder v1.0.1

Menu

1. Set Bitcoin Address
2. Set Encryption Mode
3. Set Attacked Extensions
4. Set Default Decryption Price
5. Set Special Decryption Price For Country
7. Print Current Ransomware Build Settings
8. Download Ransomware Core
9. Load Ransomware Core
0. Build Ransomware
```

ZDNet

SECURITY NBN AI CXO STORAGE HARDWARE INNOVATION MORE NEWSLETTERS

MUST READ JUSTICE DEPT. DROPS PLAYPEN CHILD PORN CASE TO PREVENT RELEASE OF TOR HACK

New dark web scheme lets wannabe cybercriminals get in on ransomware - for free

Ransomware operation provides malicious software in exchange for a slice of successful scores.

By Danny Palmer | March 6, 2017 -- 13:09 GMT (00:09 AEDT) | Topic: Security

f 44 in 97

C:\Users\[redacted]\DotRansomwareBuilder.exe

```
Ransomware Builder v1.0.1

Menu

1. Set Bitcoin Address
2. Set Encryption Mode
3. Set Attacked Extensions
4. Set Default Decryption Price
5. Set Special Decryption Price For Country
7. Print Current Ransomware Build Settings
8. Download Ransomware Core
9. Load Ransomware Core
0. Build Ransomware
```

The service offers step-by-step instructions on building ransomware.

RELATED STORIES

Security
Brazil readies first satellite launch

Security
FCC blocks rules that stop internet providers from secretly selling your data

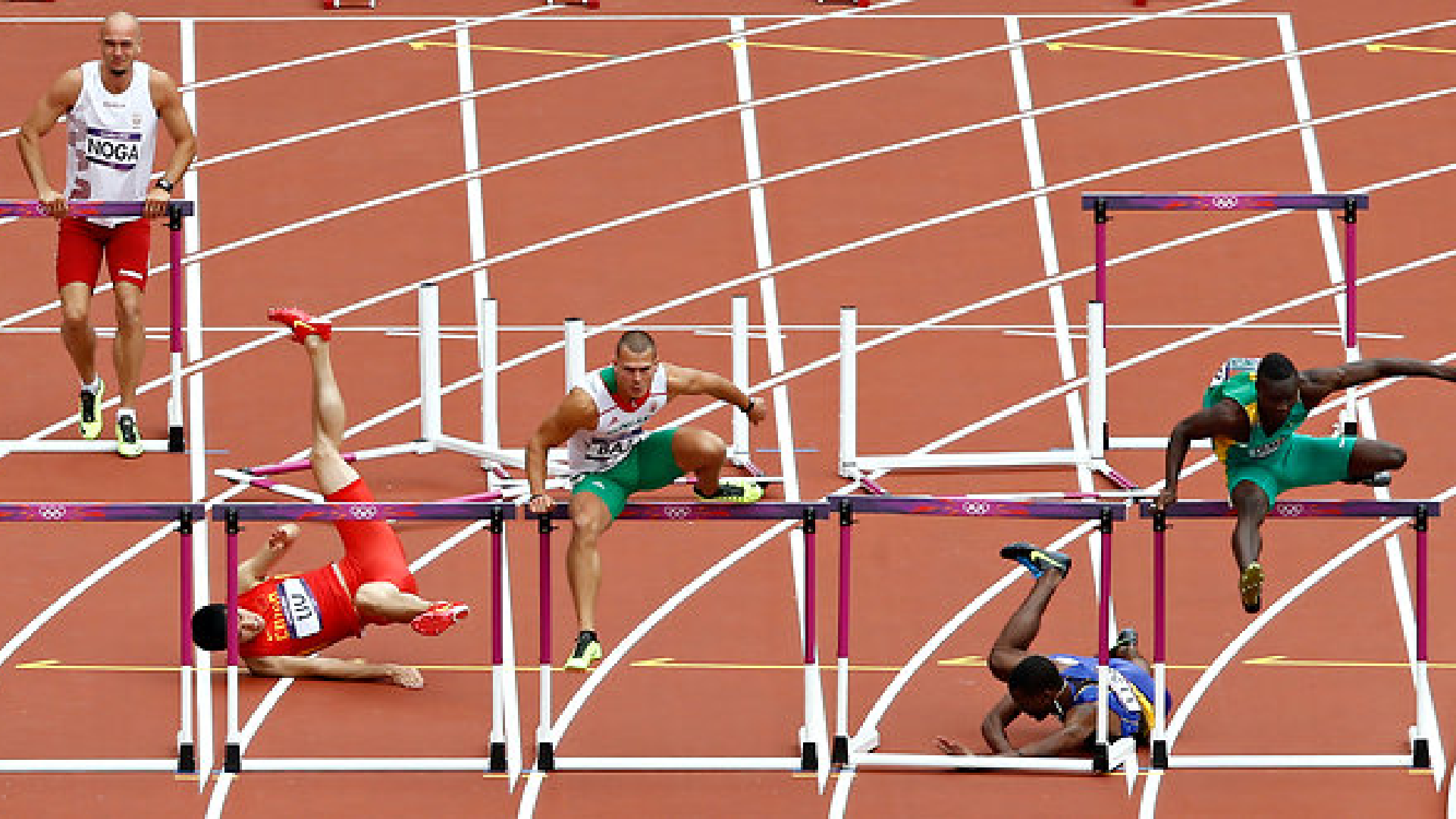
Security
Did Russia's election hacking break international law? Even the experts aren't sure

Source: http://www.zdnet.com/article/new-dark-web-scheme-lets-wannabe-cybercriminals-get-in-on-ransomware-for-free/

Silos Don’t Work

Developers	Operations	Security
		
		

Why does Automation matter?


NOGA

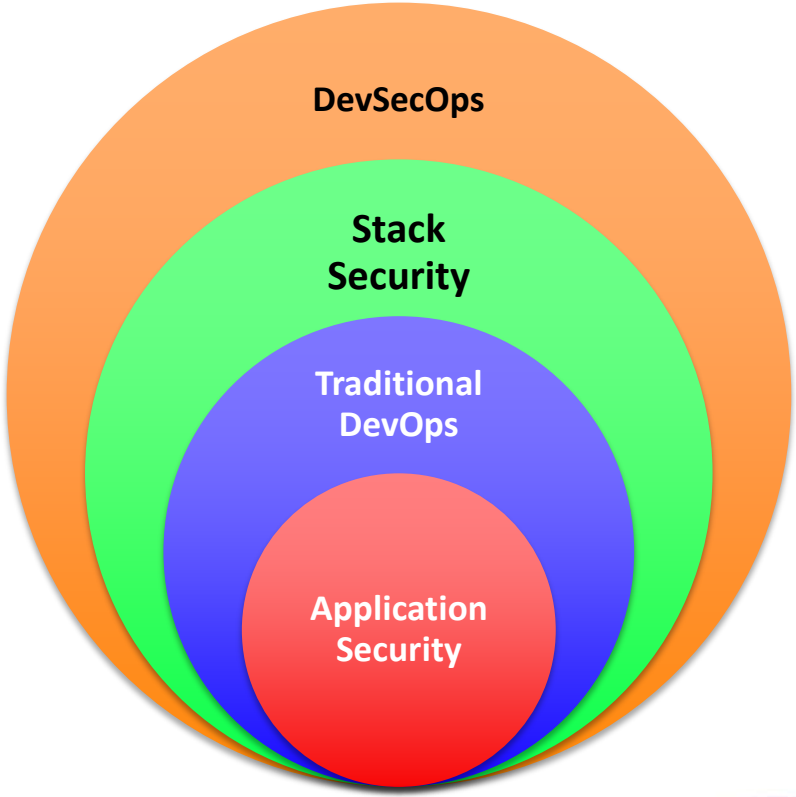

8

Security Automation in DevOps

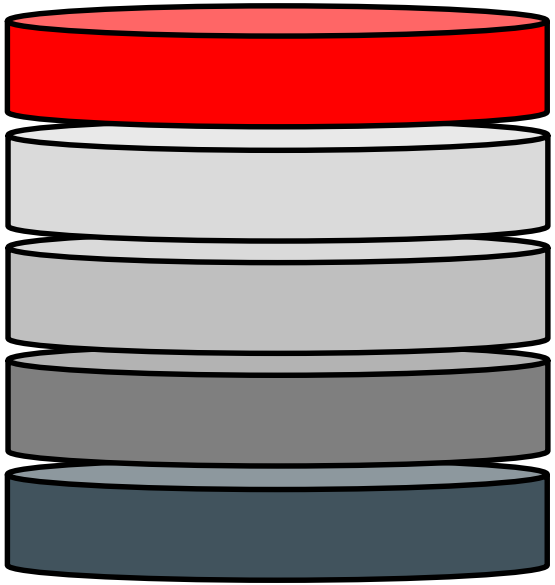
We look at a generic development pipeline…

1. Development Environment
2. Source Code Repository
3. Build Platform (CI)
4. Deployment Process (CD)
5. Staging / Production Hosting Environment

DevSecOps – Securing the Stack



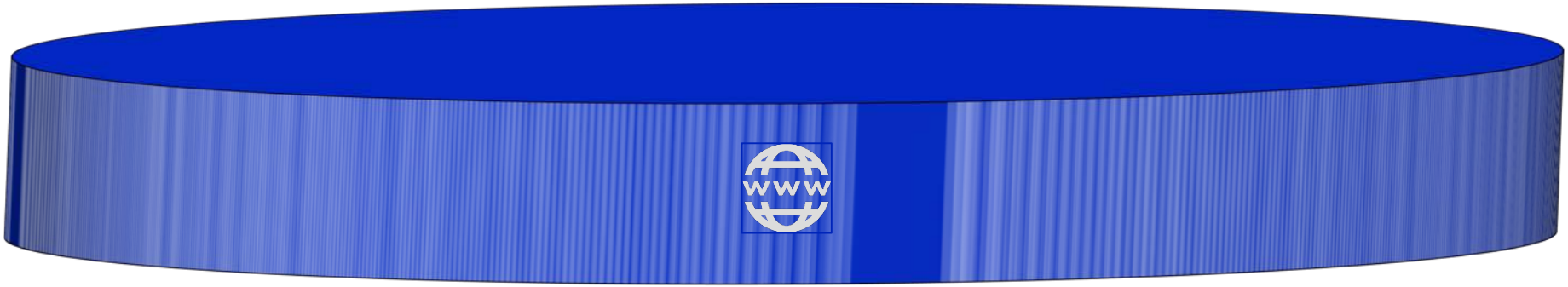
DevOps Coverage: Speed & Timing



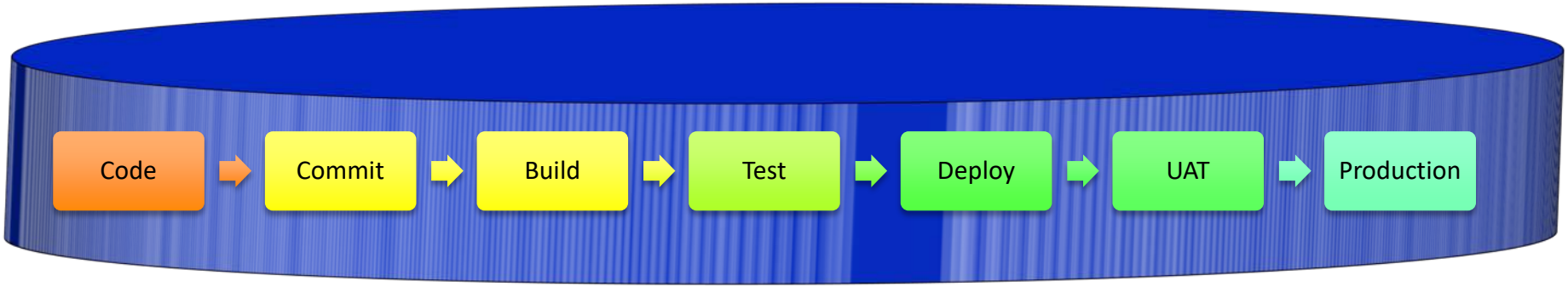
- Custom Application (1ST party code, 3rd party libraries, etc.)
- Application Framework (Tomcat, Apache, .Net, IIS etc.)
- Network & OS (Linux, Windows, etc.)
- Cloud Platform (Amazon RDS, S3, Lambda, etc.)
- Core Infrastructure (Fabric Functions: AWS IAM, EC2, Azure)

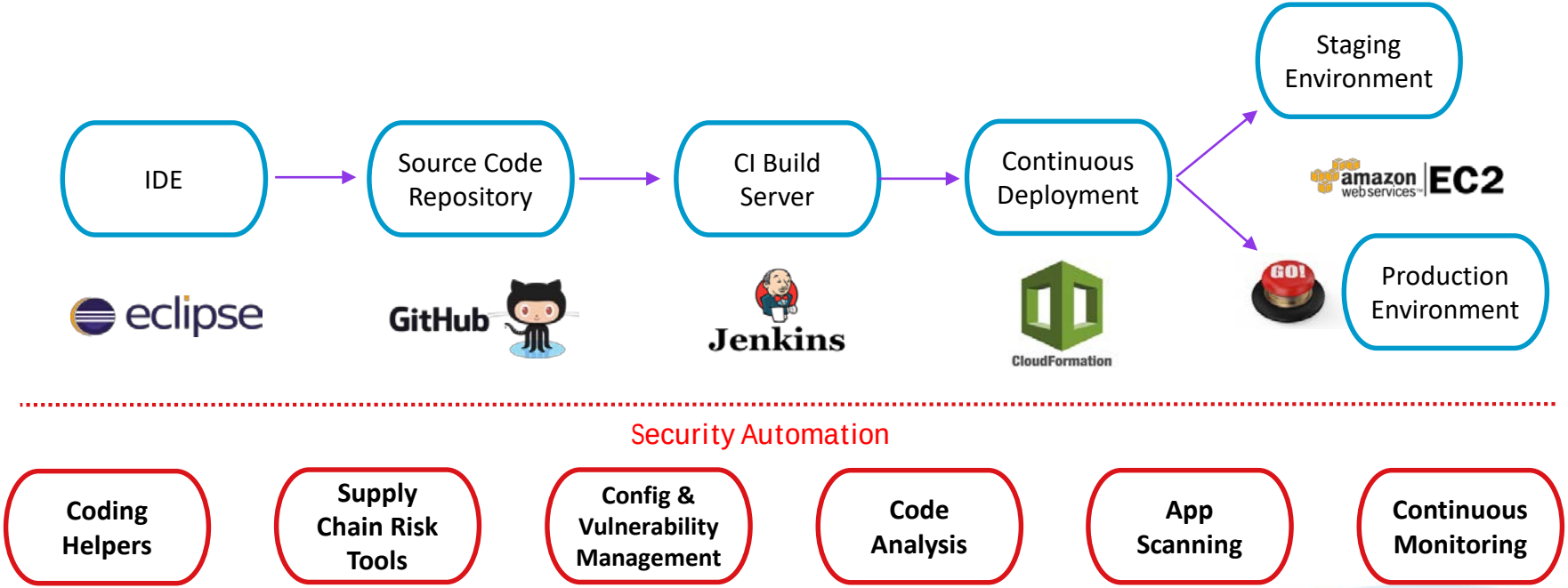
Collapse the Vertical Plane

Stretch into Horizontal Plane

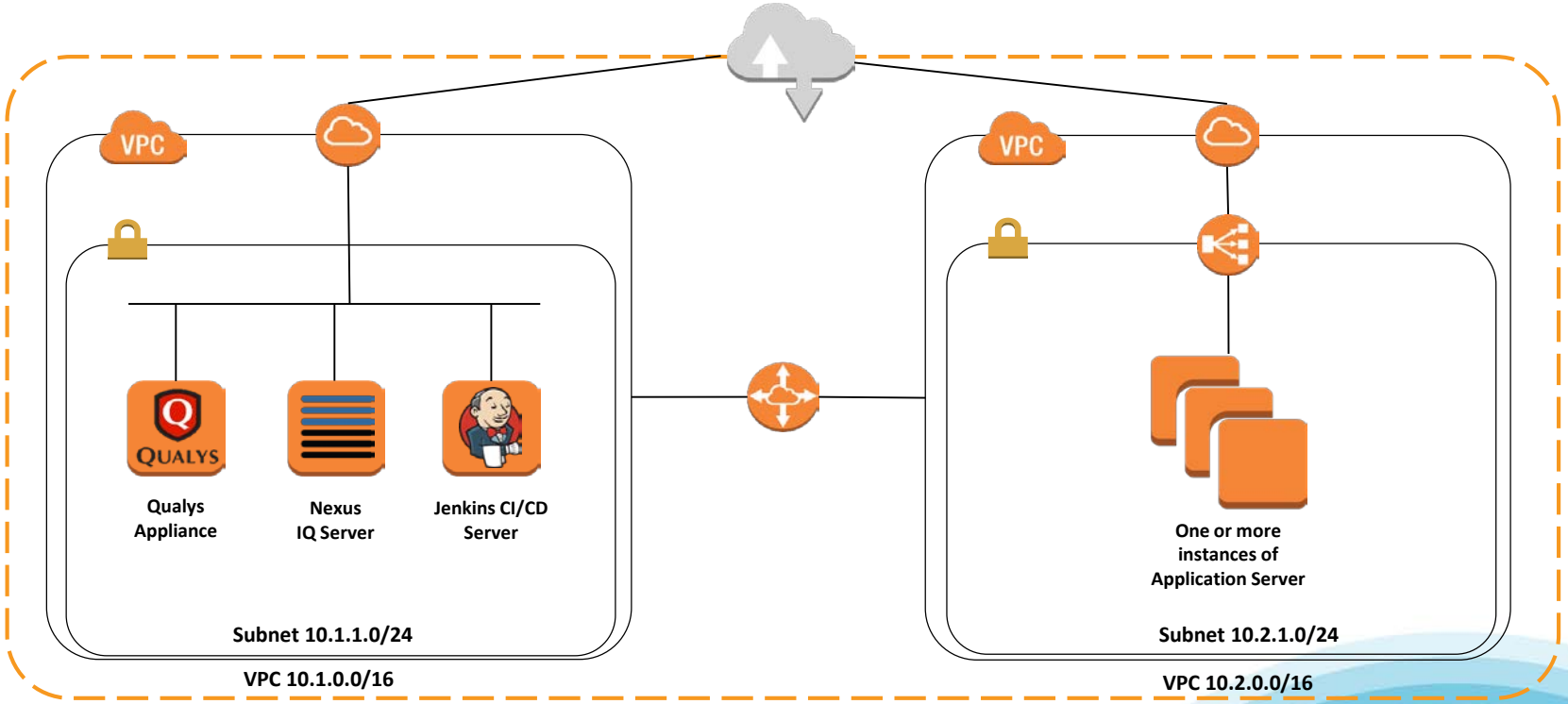


Produces the DevOps Pipeline




IDE
Source Code Repository
CI Build Server
Continuous Deployment
Staging Environment
amazon web services EC2
Production Environment
eclipse
GitHub
Jenkins
CloudFormation
GO!
Security Automation
Coding Helpers
Supply Chain Risk Tools
Config & Vulnerability Management
Code Analysis
App Scanning
Continuous Monitoring

Welcome to our DevSecOps Lab



DevSecOps Lab AWS Kill Chain Attack

We played this video during the learning lab:

https://www.youtube.com/watch?v=fm4CqlxqQfs

RSA Conference 2017
Singapore

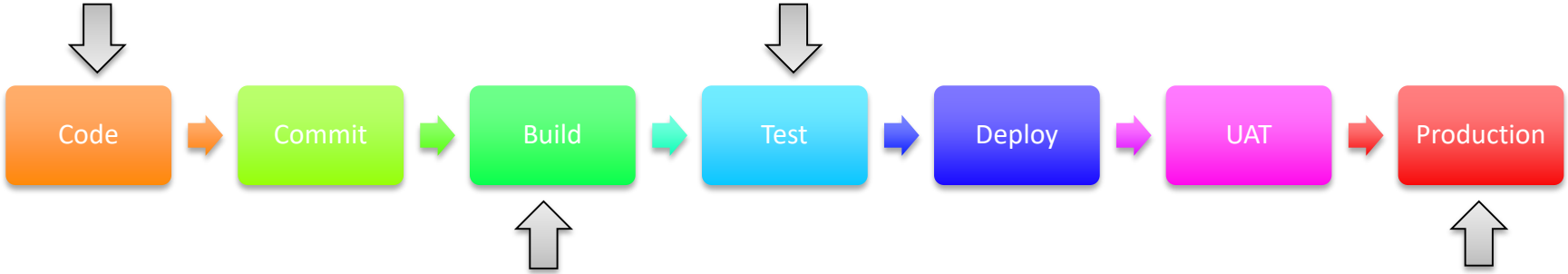
Application Security Automation

1. Addressing the need to identify defects earlier.
2. Writing and testing your in-house “first party” code.
3. Testing and inspecting libraries and “third party” code.

Defense in Depth

Layer #1 – The developer has an opportunity to avoid introducing a security vulnerability in their IDE.

Layer #3 – Automated dynamic scanning of the application detects the same vulnerability if it gets this far.



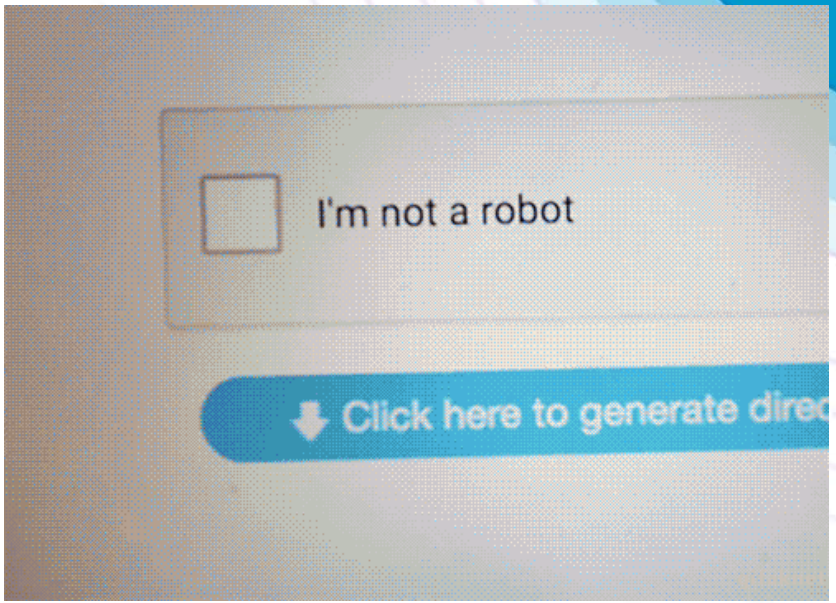
Layer #2 – Static code analysis triggered by the code commit action identifies the vulnerability – build fails.

Layer #4 – Continuous Monitoring & Vulnerability Management detects the exposed vulnerability. Add comprehensive Manual Pen Test.

RSA Conference 2017

Singapore

Automating Quality Code



Good Quality Code – Problem Statement

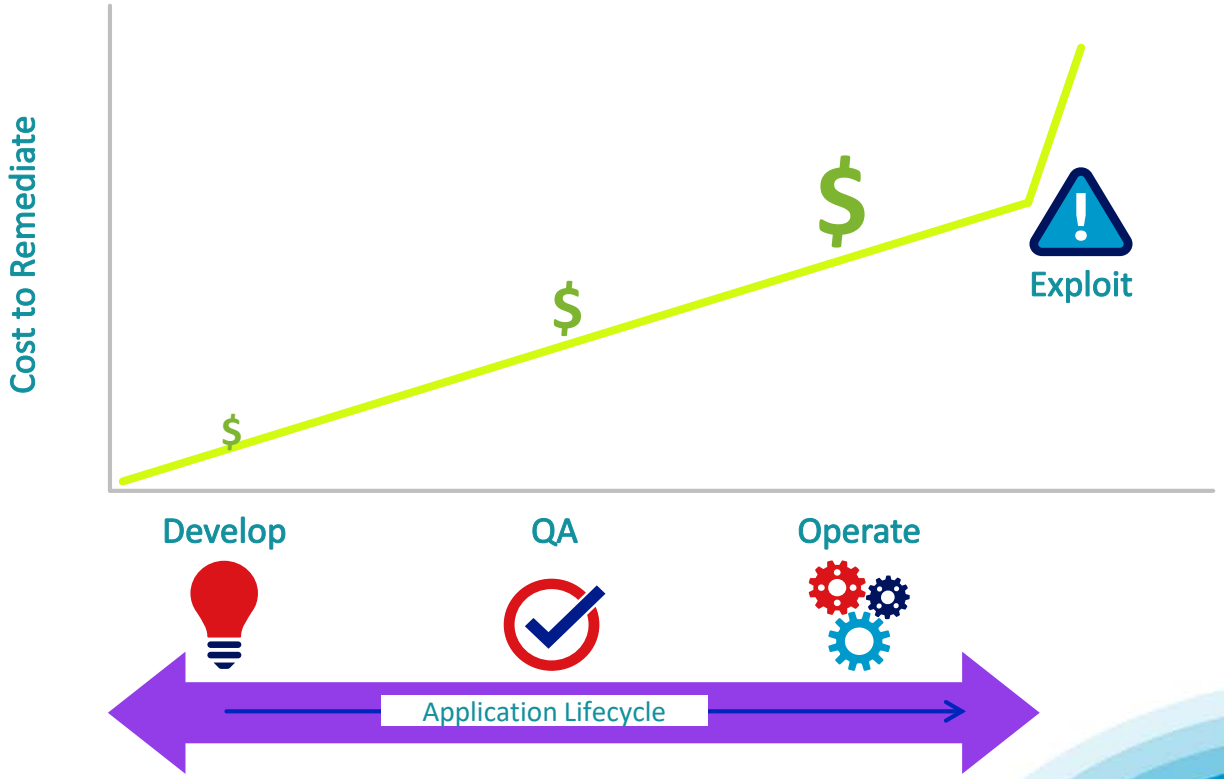
Why do you need to address code quality?

- Vulnerabilities caused by coding may lead to **unacceptable risk**.
- Well written code **performs better**
  - If well understood, has less risk of being vulnerable.
  - Likely to have better bottom line results on the final application.

Pop Quiz!

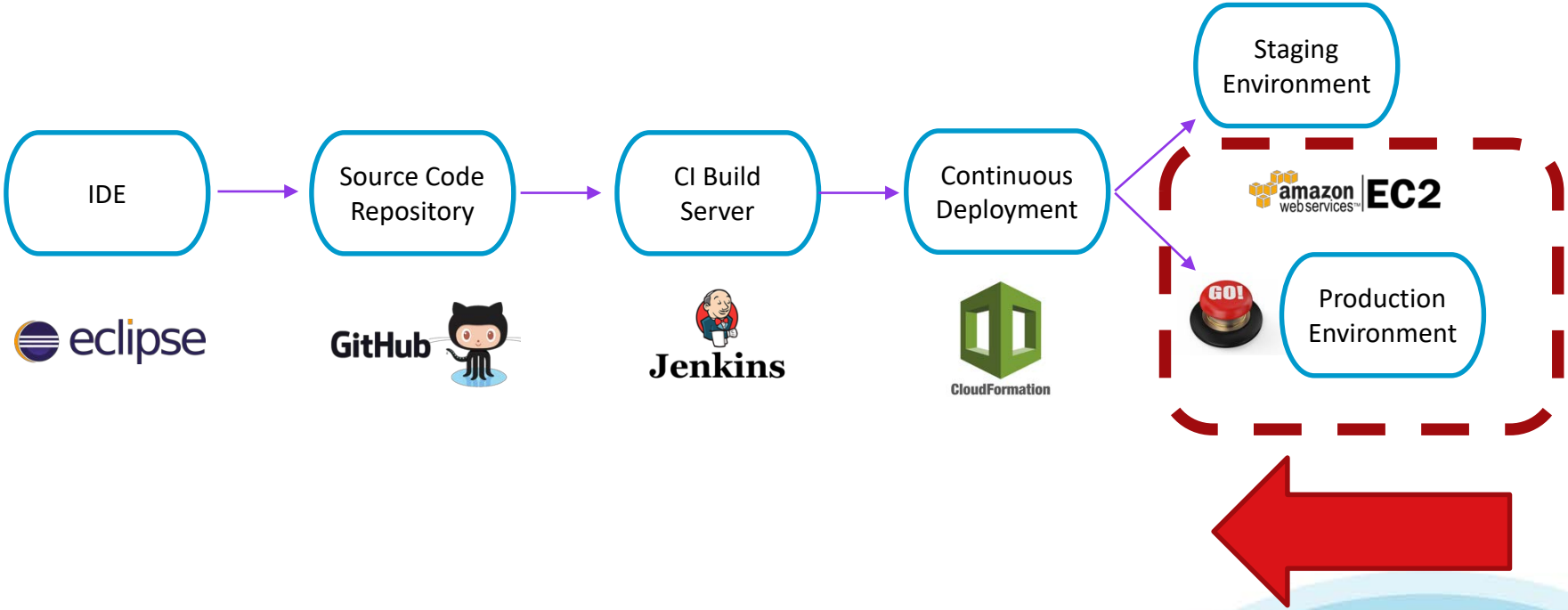
When is the best time to address coding defects?

Identify Defects As Soon As Possible



Source: Veracode

Shifting Left



Guardrail Approach

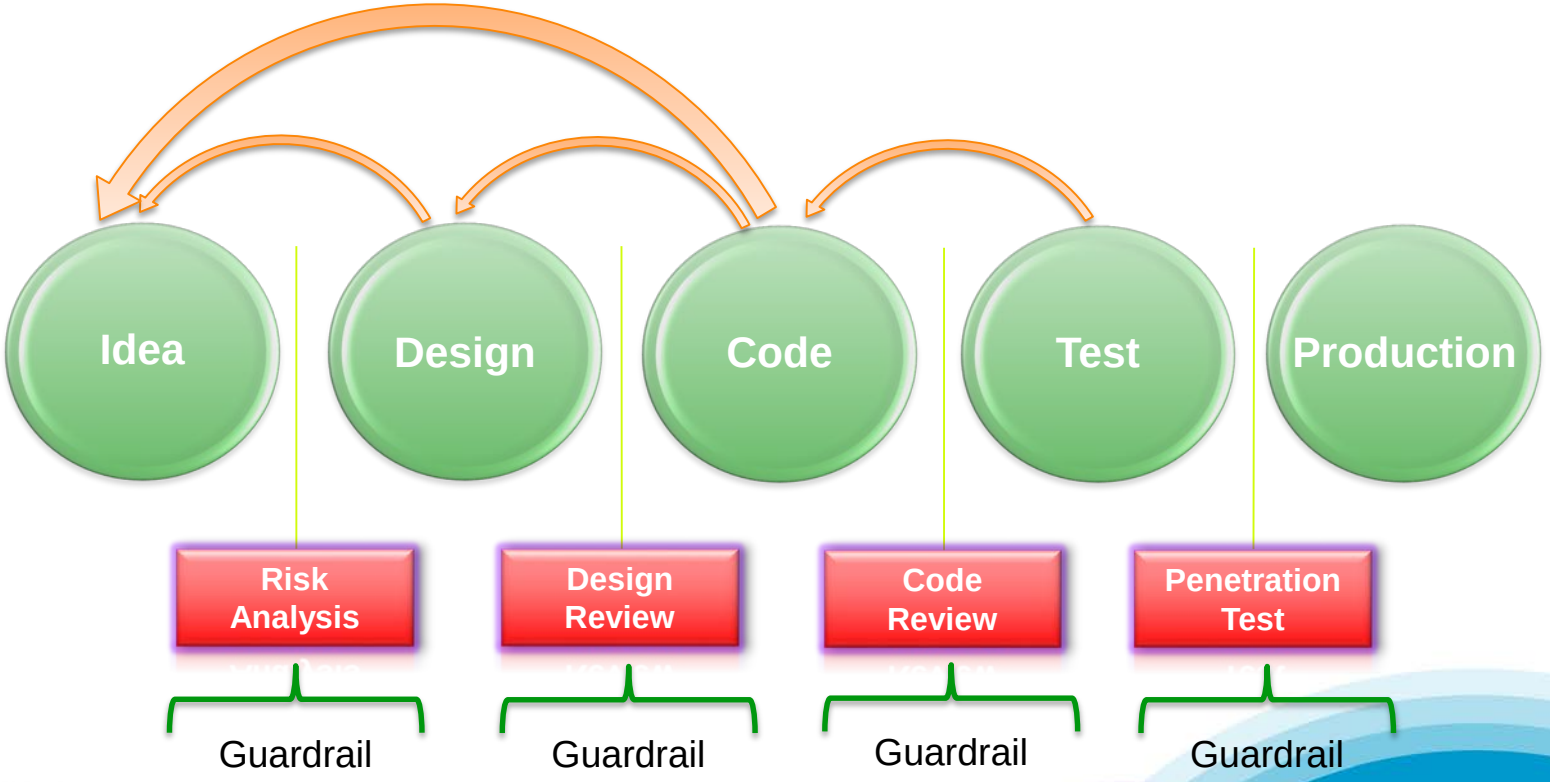


Image adapted from: Michael Brunton-Spall

Scanning Code at the IDE

Markers | Properties | Servers | Data Source Explorer | Snippets | Veracode Greenlight

Security Flaws Found: 2 High 6 Medium 2 Low

VERACODE

Best Practices: 0 CWEs Protected Against

com.badapp.servlet took 10 seconds to scan

Severity	Issue	CWE ID	Filepath	Line	Last Scanned	Details	Ignore
High	SQL Injection: Improper Neutraliza...	89	/BadWebApp/src/com/badapp/servlet/login.java	61	5 seconds ago	Details	Ignore
High	SQL Injection: Improper Neutraliza...	89	/BadWebApp/src/com/badapp/servlet/search.java	62	5 seconds ago	Details	Ignore
Medium	Basic XSS: Improper Neutralization...	80	/BadWebApp/src/com/badapp/servlet/search.java	67	5 seconds ago	Details	Ignore
Medium	Basic XSS: Improper Neutralization...	80	/BadWebApp/src/com/badapp/servlet/search.java	68	5 seconds ago	Details	Ignore
Medium	Use of Hard-coded Password	259	/BadWebApp/src/com/badapp/servlet/login.java	57	5 seconds ago	Details	Ignore
Medium	Use of Hard-coded Password	259	/BadWebApp/src/com/badapp/servlet/search.java	57	5 seconds ago	Details	Ignore
Medium	Session Fixation	384	/BadWebApp/src/com/badapp/servlet/login.java	63	5 seconds ago	Details	Ignore
Medium	Trust Boundary Violation	501	/BadWebApp/src/com/badapp/servlet/login.java	64	5 seconds ago	Details	Ignore
Low	J2EE Bad Practices: Direct Manag...	245	/BadWebApp/src/com/badapp/servlet/login.java	57	5 seconds ago	Details	Ignore
Low	J2EE Bad Practices: Direct Manag...	245	/BadWebApp/src/com/badapp/servlet/search.java	57	5 seconds ago	Details	Ignore

Findings (10) | Ignored (0) | Best Practices (0)

Early Dev, Mid Dev & Build Coverage on Commit

Automating Security at the Deploy Layer

Preventing a deployment if something fails.

```
Using Scan 1218389
Checks Failed
POST BUILD TASK : FAILURE
END OF POST BUILD TASK: 0
ESCALATE FAILED POST BUILD TASK
TO JOB STATUS
Build step ‘Post build task’
changed build result to FAILURE
Finished: FAILURE
```

RSA Conference 2017

Singapore

Third Party Libraries

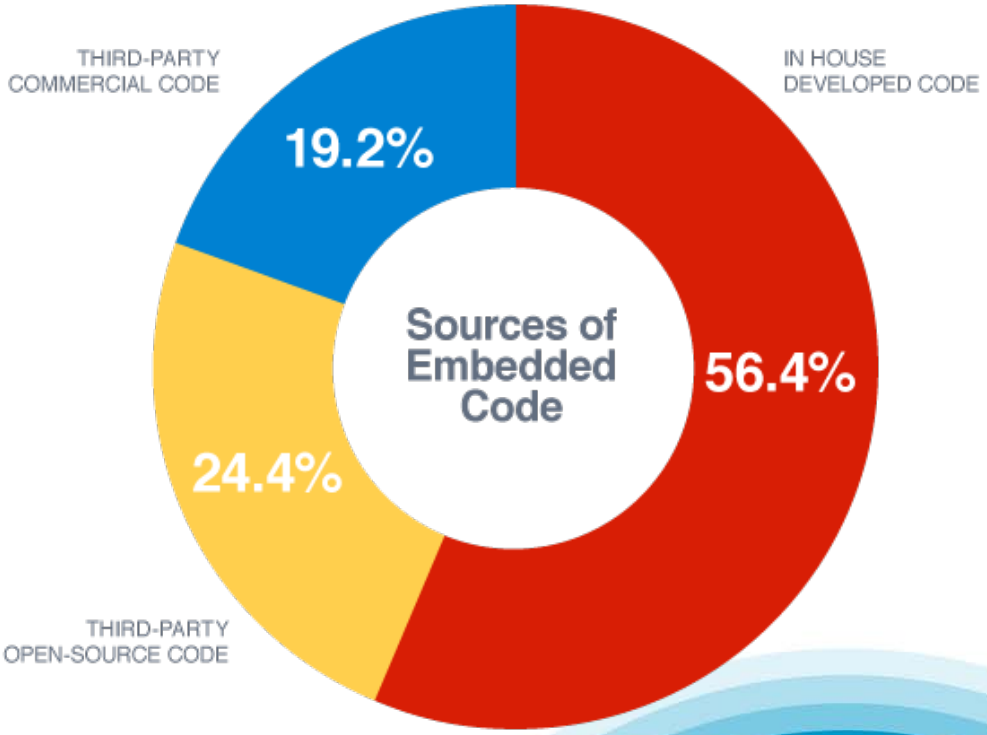
Third Party Code – Problem Statement

Why do you need to address third party library risk?

- Embedding third party code in your application has huge advantages, but comes at the risk of **latent exposure to vulnerabilities**.
- Many open source library repositories have little or no vetting of contributors, meaning **third party code cannot be trusted** blindly.
- When vulnerabilities are discovered in a shared library, it is important to **quickly identify your exposure**.

The Software Supply Chain Problem

44% of applications contain critical vulnerabilities in an open source component.
~ Veracode

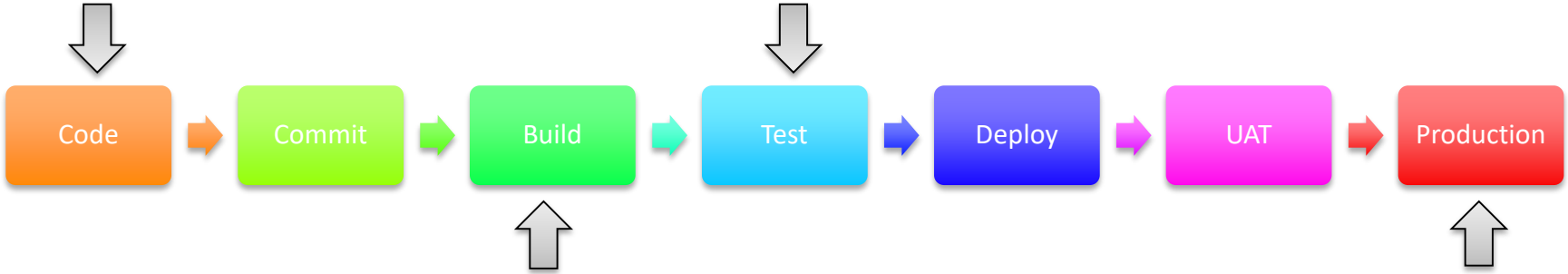


Source: https://www.grammatech.com/

Defense in Depth

Layer #1 – The developer has an opportunity to avoid introducing a security vulnerability in their IDE.

Layer #3 – Automated dynamic scanning of the application detects the same vulnerability if it gets this far.



Layer #2 – Static code analysis triggered by the code commit action identifies the vulnerability – build fails.

Layer #4 – Continuous Monitoring & Vulnerability Management detects the exposed vulnerability. Add comprehensive Manual Pen Test.

RSA Conference 2017
Singapore

Monitoring & Self-Healing

1. Cloud environments require proper configuration management.
2. Visibility is key to knowing if your DevOps stack is secure.
3. Self-healing is a growing trend and worth implementing.

RSA Conference 2017

Singapore

Cloud Configuration

Configuration Monitoring – Problem Statement

Why is your cloud environment configuration important?

- Complex environments have **complex and diverse configurations**.
- Cloud configurations **aren't always visible**, and we need that visibility to understand the real configuration.
- We **need to have assurance** that our configuration standard is being enforced and is compliant.

RSA Conference 2017

Singapore

Continuous Monitoring

It’s all about Visibility

Custom Application (1ST party code, 3rd party libraries, etc.)

Application Framework (Tomcat, Nginx, Apache, etc.)

Network & OS (Linux, Windows, etc.)

Cloud Platform (Amazon RDS, S3, Lambda, etc.)

Core Infrastructure (Fabric Functions: AWS IAM, EC2, Azure, etc.)

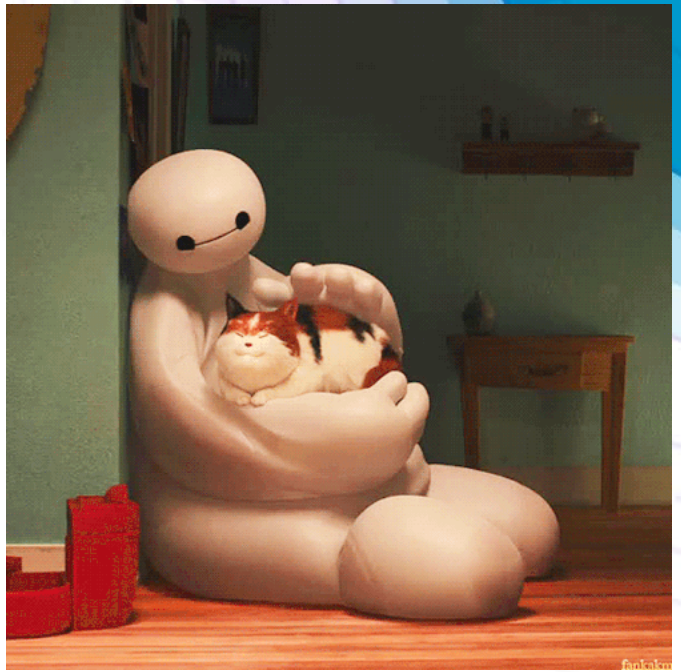
Sense of Security

Continuous Monitoring

RSA Conference 2017

Singapore

Self-Healing



Self-Healing - Problem Statement

Why is **Self-Healing** important?

- Respond to changes in your environment immediately, reverting changes- malicious or accidental.
- Assurance that your stack configuration is compliant to your risk appetite at all times.
- Alert you to take action for improvement if it does detect unwanted changes (or alert of a security incident).

Event Driven Security, Self-Healing, or RASP

The techniques we're about to look at in our lab are all known by different names:

- Event Driven Security – responding to events
- RASP – Runtime Application Self Protection
- Self-Healing – we think this describes it nicely!

There may be subtle difference in implementation, but for the large part we consider they all do the same thing.

We're Going Serverless!

*"Serverless computing solutions execute logic in environments with no visible VM or OS. Services such as Amazon Web Services Lambda are disrupting many cloud development and operational patterns. Technology and service provider product managers must prepare for the change." - **Gartner***

AWS Lambda

- It's "Serverless"
- A stateless, programmatic function that responds to events based on triggers.
- Other Platforms:
  - Microsoft Azure: "Azure Functions"
  - Google Cloud Platform: "Google Cloud Functions"

AWS Lambda

Event Driven Security / Self-Healing

To implement automated self-healing using a serverless solution we generally need a few things:

1. A well defined “event” that we can respond to (i.e. an open port, or a new user account being created)
2. A near real-time source of logging data to listen for the event.
3. Something to do if the event is triggered.

Demo

Demo Lambda locking a user out after they try to create another user account.

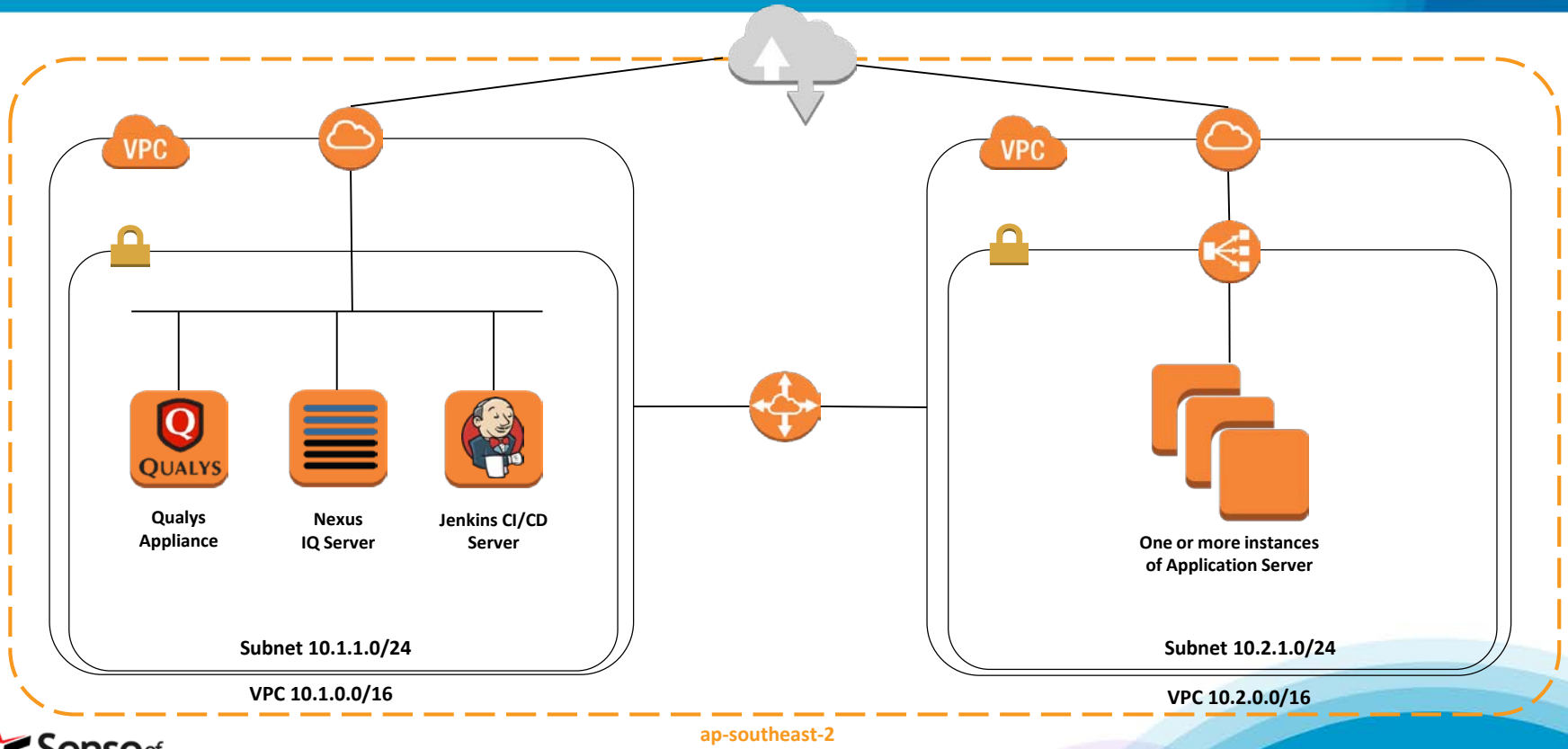
Or disable user without 2-factor?

RSA Conference 2017

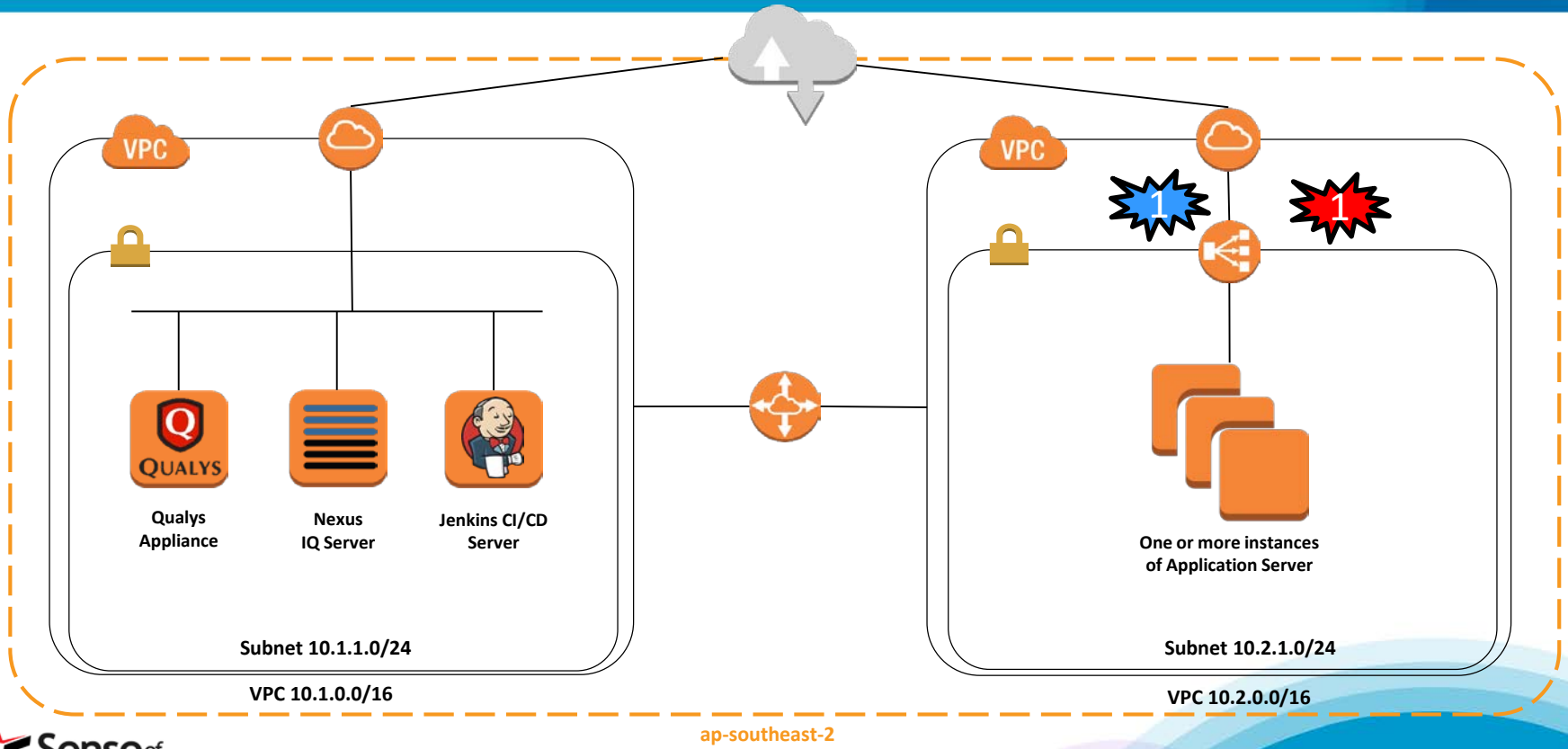
Singapore

AWS Kill Chain Mitigations

DevSecOps Lab – AWS Kill Chain



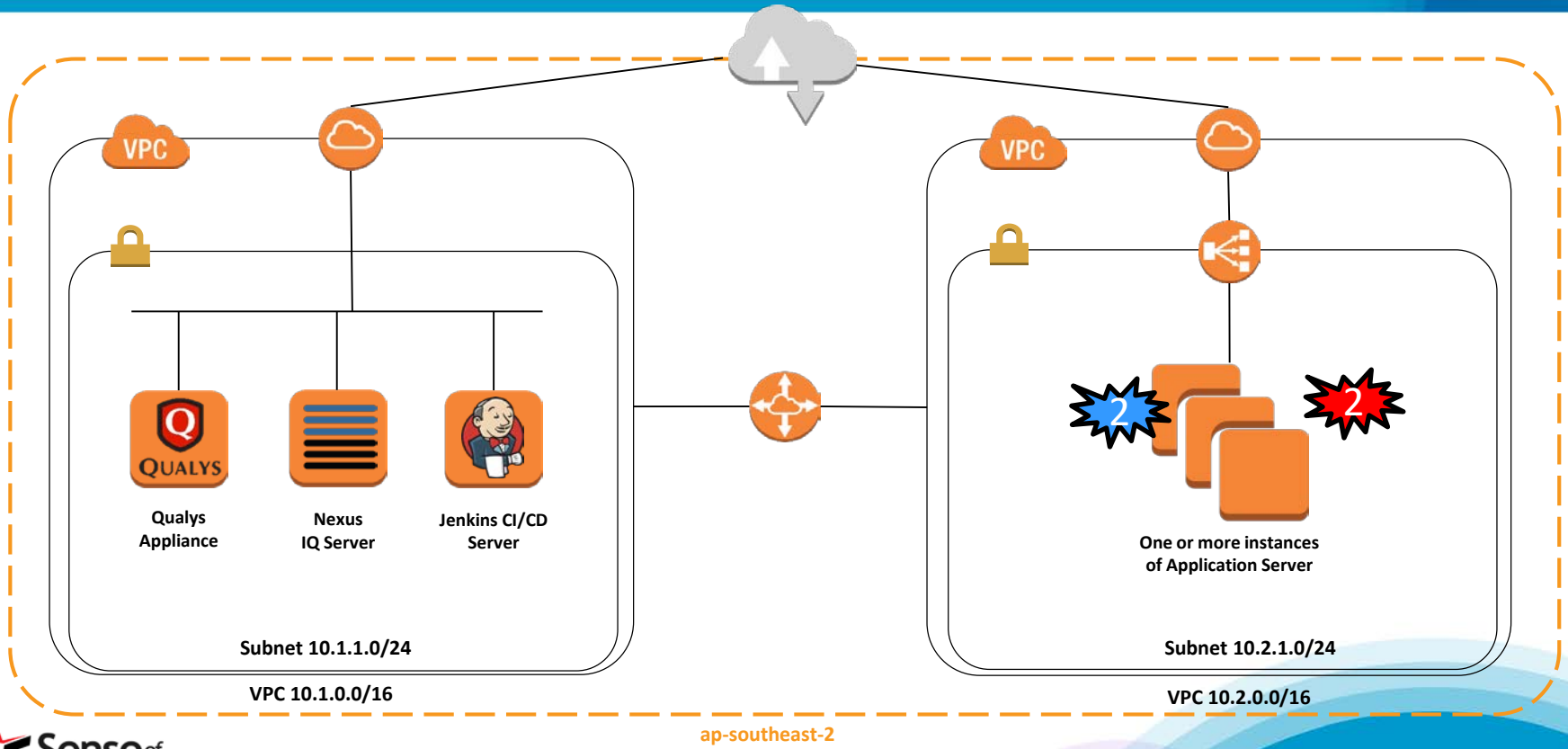
DevSecOps Lab – AWS Kill Chain



Time Line

ID	Attack	Countermeasure Process	Countermeasure Technology
1	Vulnerability Identification	External Vuln Scanning Automation – extend to Continuous Monitoring	Qualys (VM + Cont Mon, WAS) Veracode (Dynamic)
1	Vulnerability Prevention (OS, Framework, Environment etc.)	Config Mgt Patch Mgt	Active: • IPS Passive: • Qualys (VM, Policy Compliance)
1	Vulnerability Prevention (First Party Code)	Security in SDLC	Active WAF RASP (e.g. Veracode) SDLC Veracode (Greenlight, Static)
1	Vulnerability Prevention (3rd Party Code)	Security in SDLC	Veracode (SCA) Sonatype

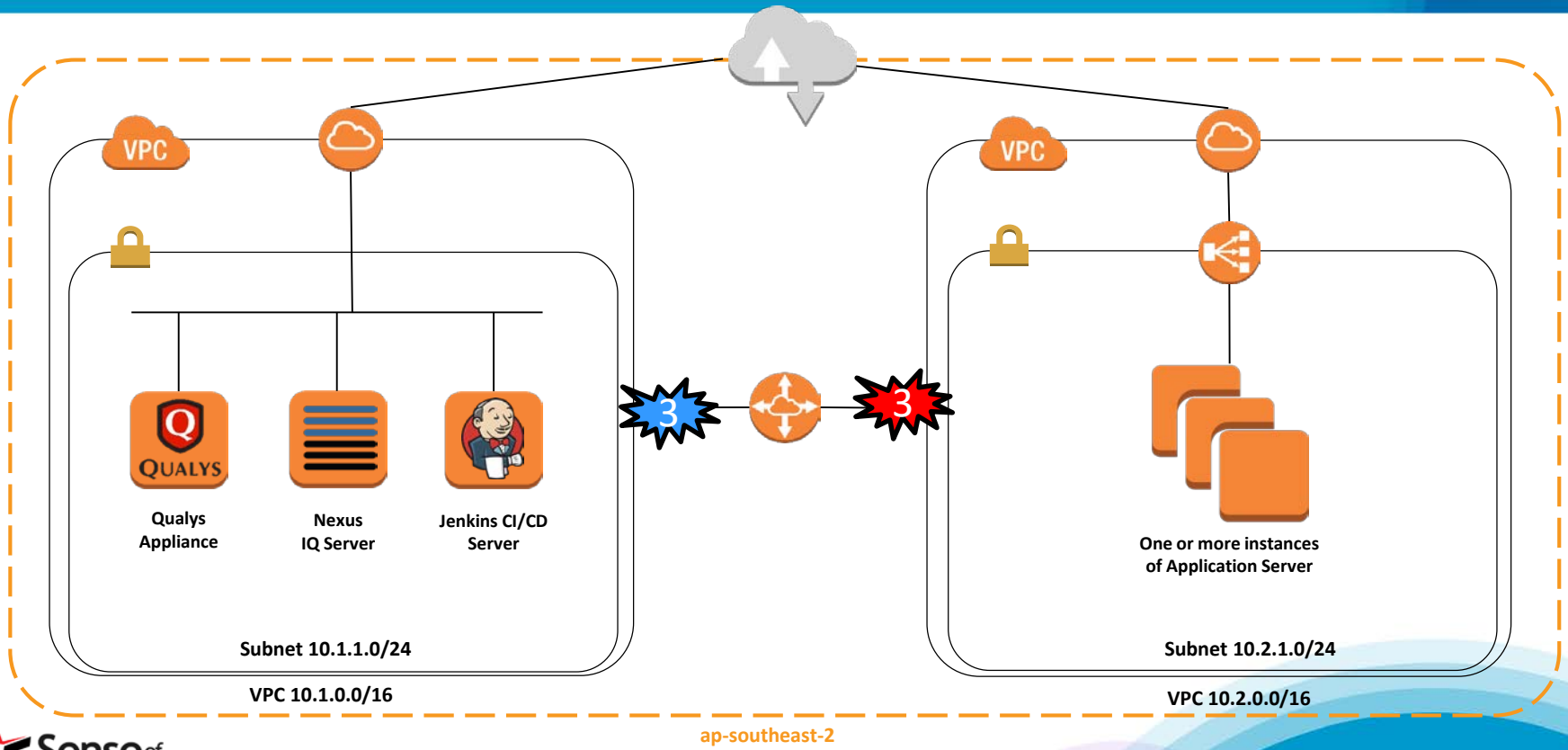
DevSecOps Lab – AWS Kill Chain



Time Line

ID	Attack	Countermeasure Process	Countermeasure Technology
2	Vulnerability Prevention (3rd Party Code)	Security in SDLC	Veracode (SCA) Sonatype
2	Shell Binding, Tools Download etc.	Restrict unsolicited outbound access	• Self-Healing / Tamper Resistance • Application Whitelisting • AWS Lambda Functions (DIY) • Dome9 Clarity Diagram • Dome9 Clarity VPC Log Review
2	Vulnerability Prevention	Configuration Management Patch Management	• IPS • Qualys (VM, Policy Compliance)
2	Vulnerability Prevention (First Party Code)	Security in SDLC	WAF RASP (e.g. Veracode) Veracode (Greenlight, Static)

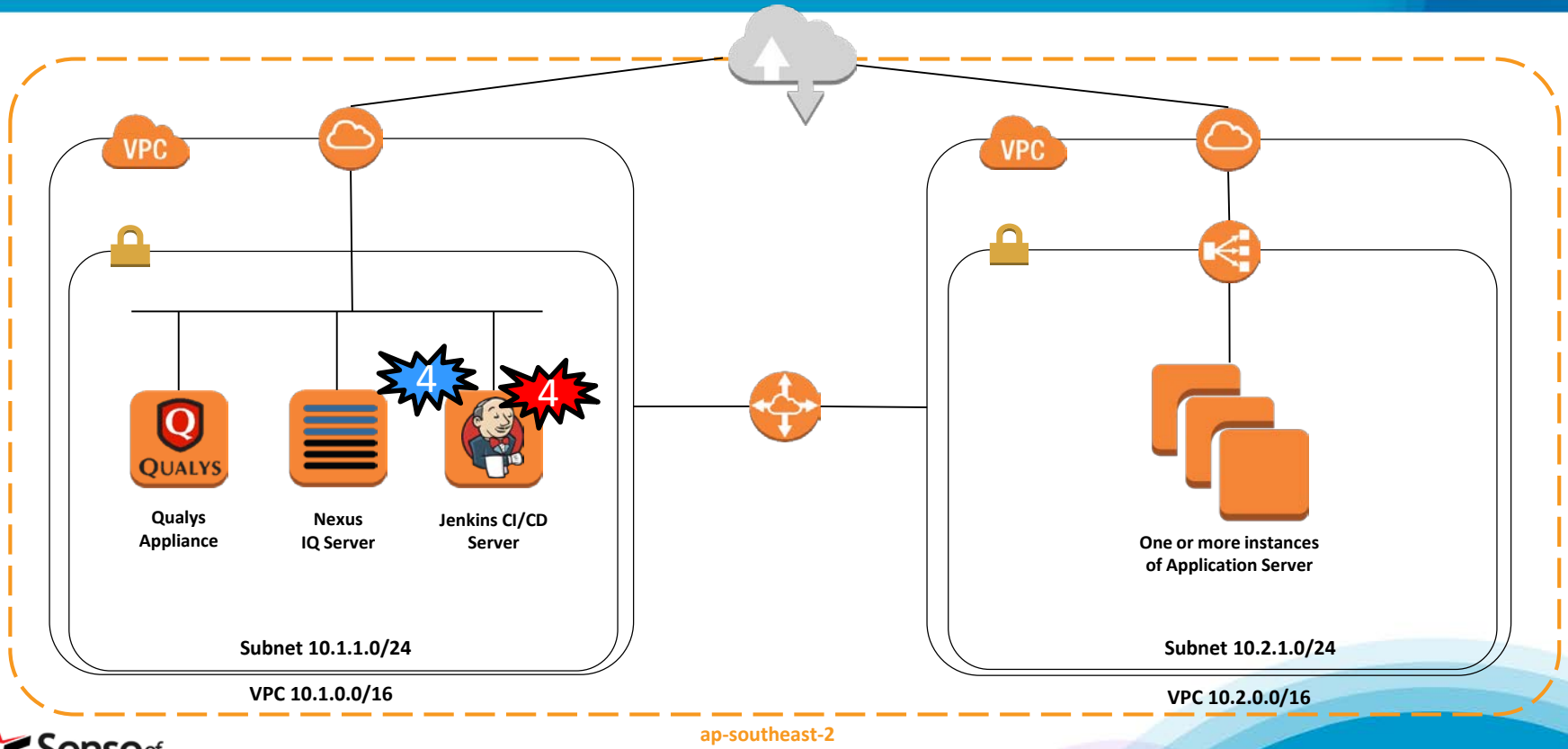
DevSecOps Lab – AWS Kill Chain



Time Line

ID	Attack	Countermeasure Process	Countermeasure Technology
3	Pivot, Vuln Identification	Restrict unsolicited traffic intra-VPC, intra-Account, VPC-WAN etc.	Active Automation • Dome9 AWS Security Group Rule Tamper Resistance Visual • Dome9 Clarity Diagram • Dome9 Clarity VPC Log Review • Passive • Qualys VM + Cont Mon

DevSecOps Lab – AWS Kill Chain

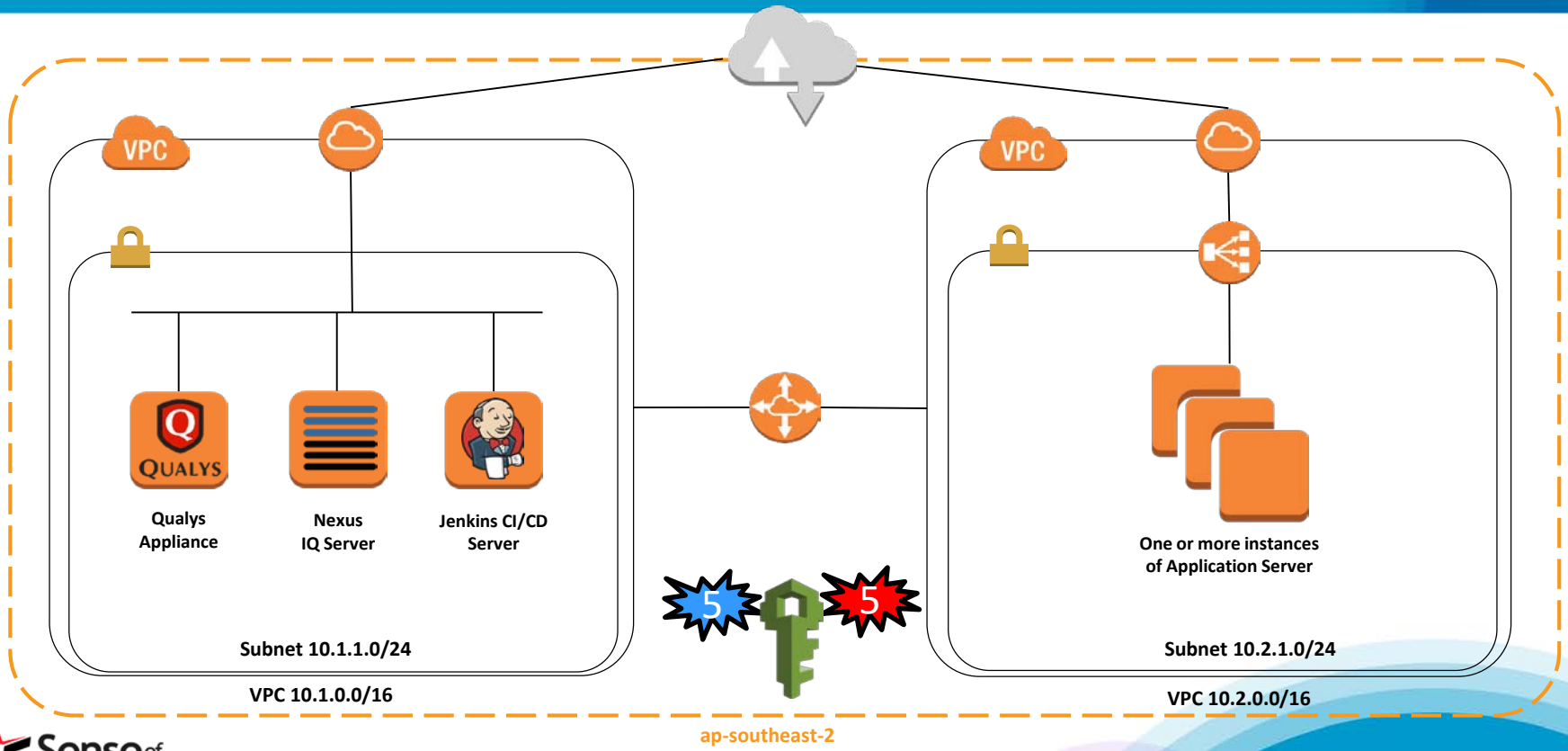


Time Line

ID	Attack	Countermeasure Process	Countermeasure Technology

4	Vulnerability Prevention (OS, Framework, Environment etc.)	As Per Previous
		• Depends on Vuln Type:
		  • Config Mgt
		  • Patch Mgt
		  • Security in SDLC
			Active:
			  • IPS
			Passive:
			  • Qualys (VM, Policy Compliance)
			SDLC
			  • Veracode, Sonatype etc

DevSecOps Lab – AWS Kill Chain



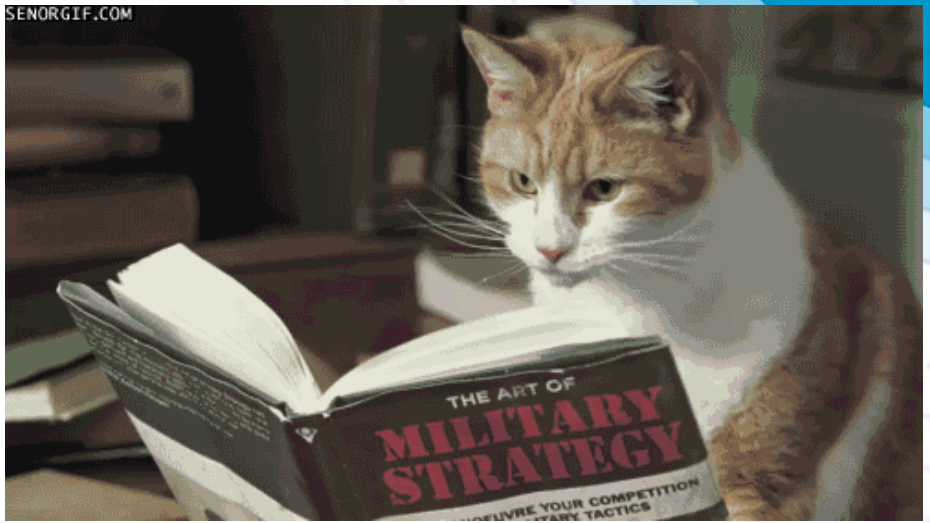
Time Line

ID	Attack	Countermeasure Process	Countermeasure Technology
5	Cloud, Account Creation, Priv Escalation, Priv Abuse	Access Controls and Permissions • RBAC • Permissions on business need to know/use	Active • Dome9 IAM Protection • AWS Lambda Functions (DIY)

RSA Conference 2017

Singapore

Applying Security Automation



Applying Security Automation in DevOps

- Look for opportunities in your SDLC to automatically identify defects earlier in the pipeline – i.e. “Shift Left”
- Examine all your security tools and investigate whether exposed API’s can be leveraged to provide automated control/feedback.
- Review your cloud based architecture for opportunities to apply automated checking of configuration and continuous monitoring.
- Remember to protect the “full stack” of tools, processes and technology in your DevOps pipeline. It’s not just about the output!

RSA Conference 2017

Singapore

Sense of Security Pty Ltd

www.senseofsecurity.com.au

info@senseofsecurity.com.au

Office: +61 2 9290 4444