black hat USA 2019

DevSecOps What Why and How?

Anant Shrivastava
@anantshri
NotSoSecure Global Services

NOT SO SECURE
A Claranet Group Company

About: Anant Shrivastava

Director NotSoSecure Global Services

Sysadmin / Development / Security

Project Owner: AndroidTamer, Codevigilant

Contributor : OWASP, null, G4H and more

https://anantshri.info (@anantshri on social platforms)

NotSoSecure Global Services (a Claranet group company)

Boutique Consulting firm specialized in training and consulting

Agenda

- What is DevSecOps?
- Why do we need DevSecOps?
- How do we do DevSecOps?
- Integrate Security in Pipeline
- Tools of Trade
- Sample Implementation
- Case Studies

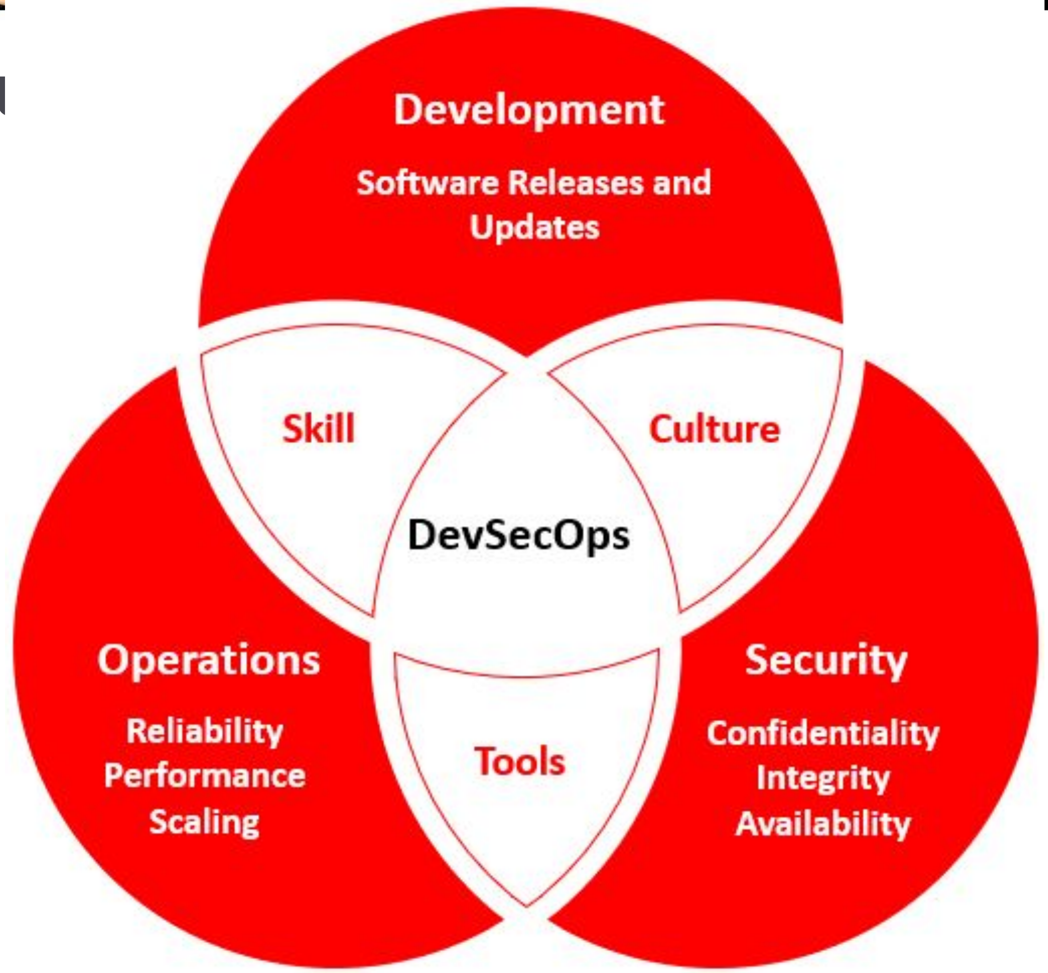
Disclaimer

- I will be listing a lot of tools, It’s not an exhaustive list
- I don't endorse or recommend any specific tool / vendor
- Every environment is different: Test and validate before implementing any ideas

What is DevSecOps?

Effort to strive for "Secure by Defa

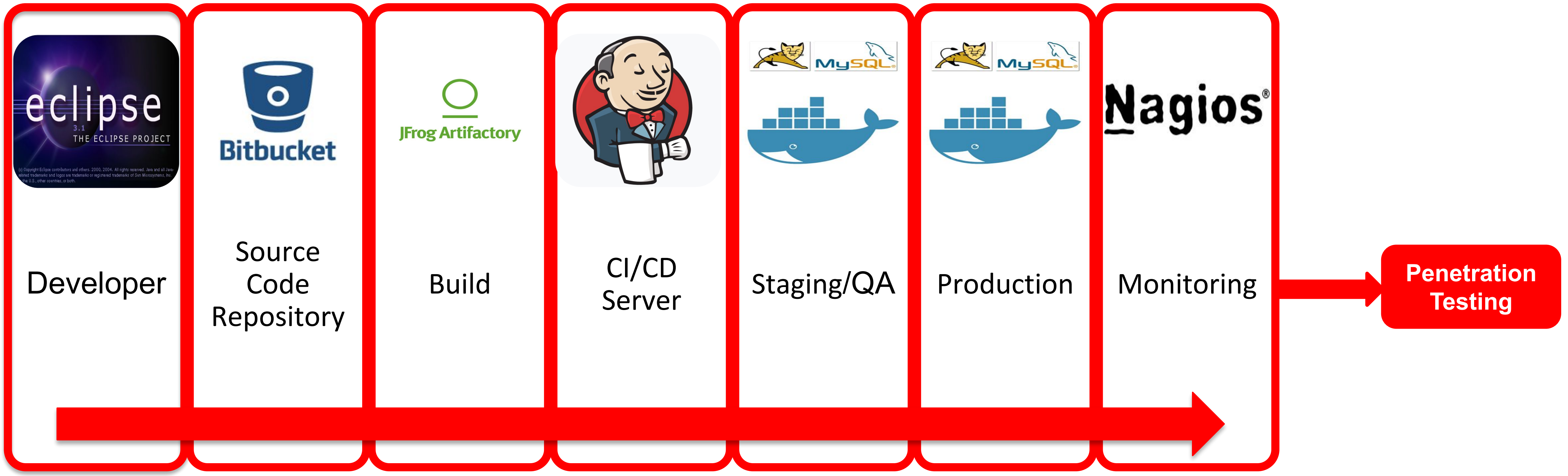
- Integrate Security via **tools**
- Create Security as Code **culture**
- Promote cross **skilling**



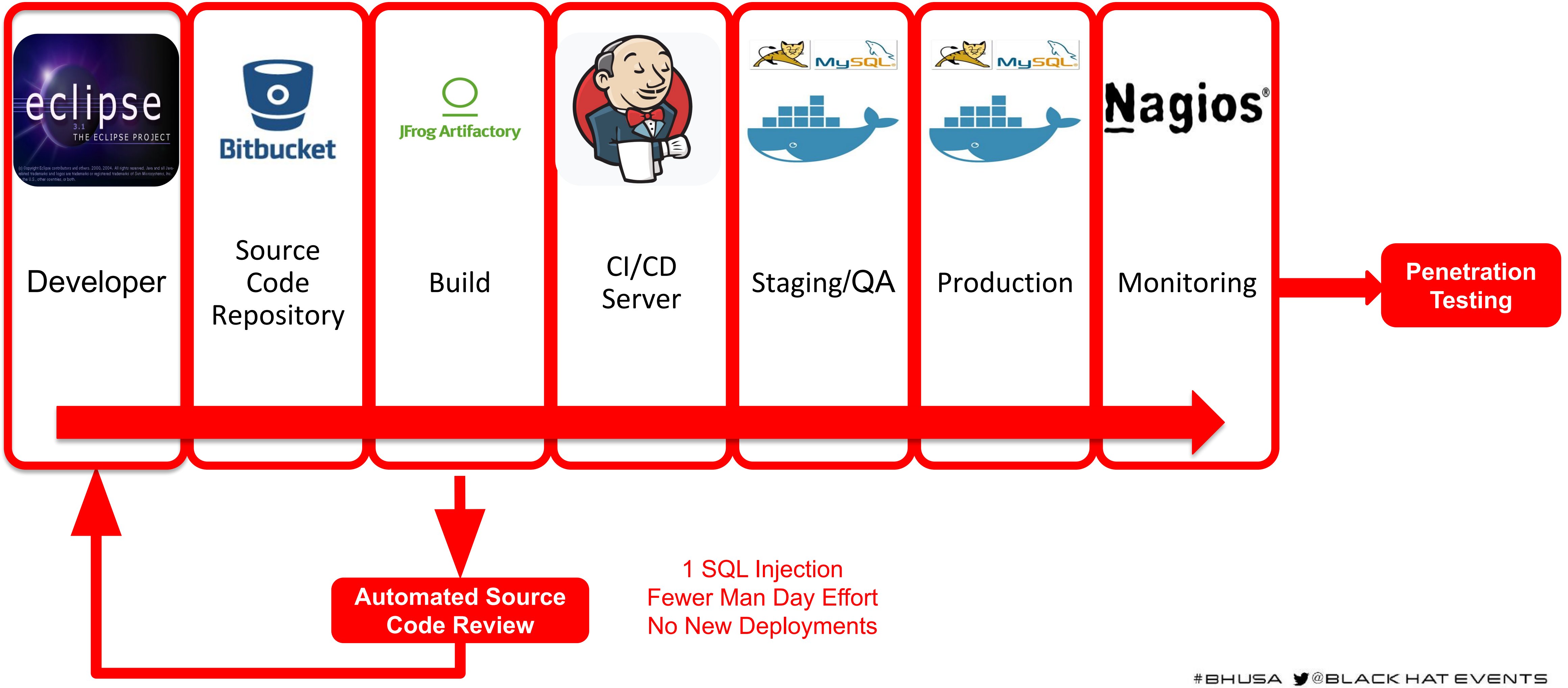
Why do we need DevSecOps?

- DevOps moves at rapid pace, traditional security just can’t keep up
- With rapid pace of development and large scale of application devsecops makes it easier to manage
- DevSecOps allows for much smoother scaling of process
- Security as part of process is the only way to ensure safety

Shifting Left saves cost & time



Shifting Left saves cost & time

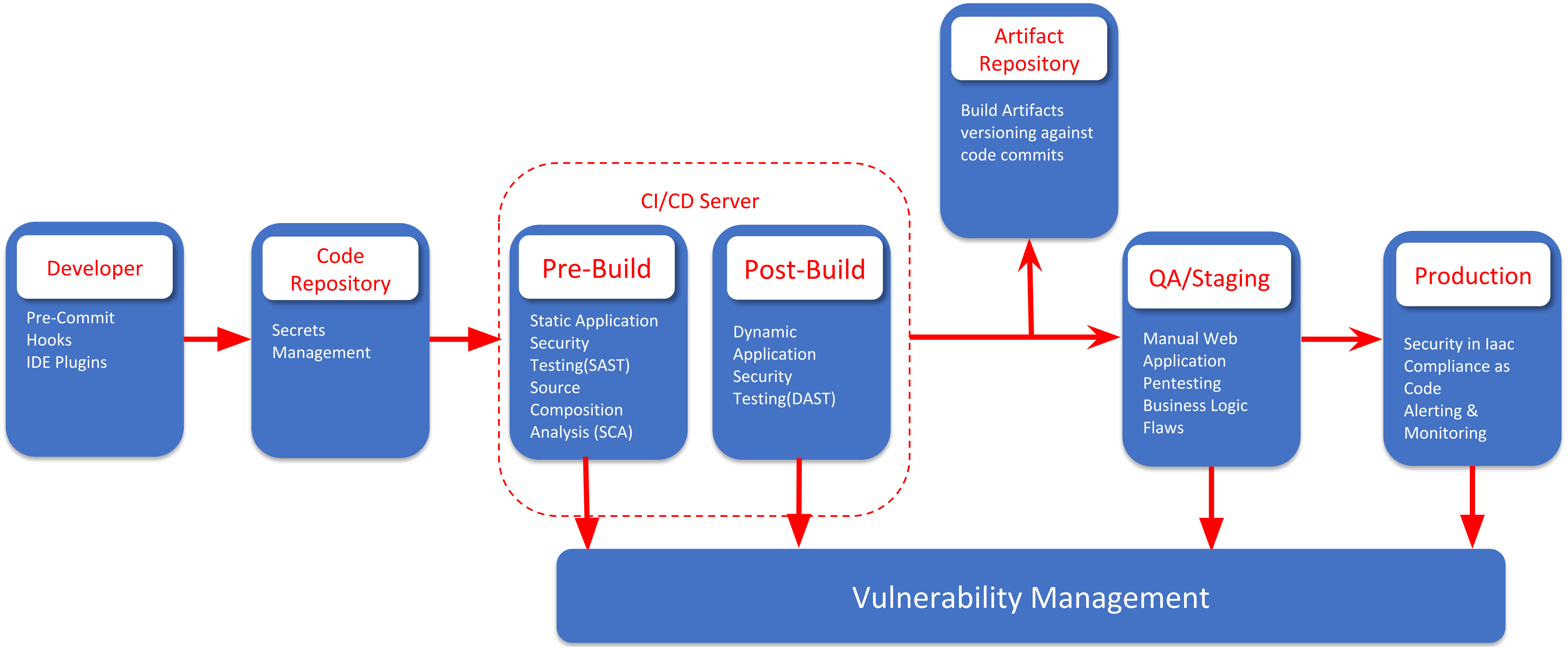


1 SQL Injection
Fewer Man Day Effort
No New Deployments

How do we do DevSecOps?

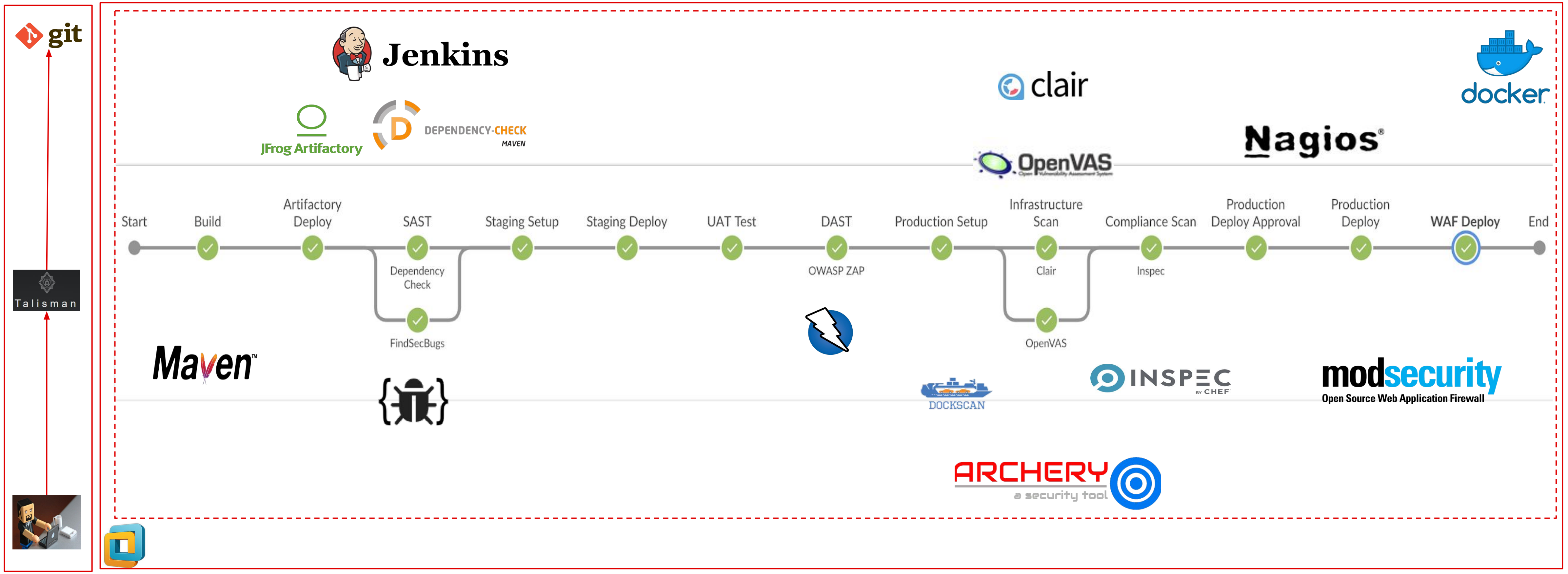
- DevSecOps is Automation + Cultural Changes
- Integrate security into your DevOps Pipeline
- Enable cultural changes to embrace DevSecOps

Injecting Sec in DevOps



Sample Implementation

- A simplistic flow of DevSecOps Pipeline incorporating the stages mentioned earlier



black hat USA 2019

Tools of The Trade

Threat Modelling Tools	THREAT PLAYBOOK · ThreatSpec. · Microsoft Threat Modeling Tool
Pre-Commit Hooks	Talisman · git-secret · truffleHog · Git Hound
Software Composition Analysis	DEPENDENCY-CHECK · Requires.io · Retire.js
Static Analysis Security Testing (SAST)	Bandit · BRAKEMAN · RIPS · sonarqube · PMD DON'T SHOOT THE MESSENGER
IDE Plugins	dev skim · CAT.net
Secret Management	HashiCorp Vault · Keywhiz · Confidant

Preference given to open-source tools; we don't endorse any tool

Tools of The Trade

Vulnerability Management	ARCHERY a security tool; JACK HAMMER; DEFECTDOJO
Dynamic Security Analysis	arachni web application security scanner framework; w3af; Wapiti
Infrastructure Scan	OpenVAS Open Vulnerability Assessment System; anchore; clair; DOCKSCAN; OpenSCAP
Compliance as Code	KitchenCI; INSPEC; DevSec Hardening Framework; Docker Bench for Security
WAF	modsecurity Open Source Web Application Firewall; NAXSI

Preference given to open-source tools; we don't endorse any tool

To be or Not to Be in Pipeline

- API / command line access
- Execution start to final output should be 15 minutes max
- Tools should be Containerized / scriptable
- Minimal licensing limitations (parallel scans or threads)
- Output format parsable / machine readable (no to stdout, yes to json / xml)
- Configurable to counter false negatives / false positives

Does Programming Language Matter

- Different programming languages need different tools for static analysis and software composition analysis
- Some tools support multiple languages like sonarqube
- Others are focused on one language

Language Specific Tools

Languages	Software Composition Analysis	Source Code Static Analysis
JAVA	DEPENDENCY-CHECK, ClearlyDefined, OSS Review Toolkit	{SpotBugs}, sonarqube, graudit
PHP	OSS Review Toolkit, sonatype	sonarqube, graudit
Python	DEPENDENCY-CHECK, Safety, sonatype	Bandit, graudit, sonarqube
.NET	DotNET Retire, SafeNuGet, OSS Review Toolkit	PUMA SCAN, DotNet Security Guard
Ruby/Rails	OSS Review Toolkit, sonatype	Brakeman, sonarqube, graudit
Node JS	ClearlyDefined, OSS Review Toolkit	npm-check, NodeJsScan, sonarqube

Preference given to open-source tools; we don't endorse any tool

What about Cloud

- The Threat Landscape changes
  - Identity and Access Management
  - Billing Attacks

CS Suite

SCOUTSUITE

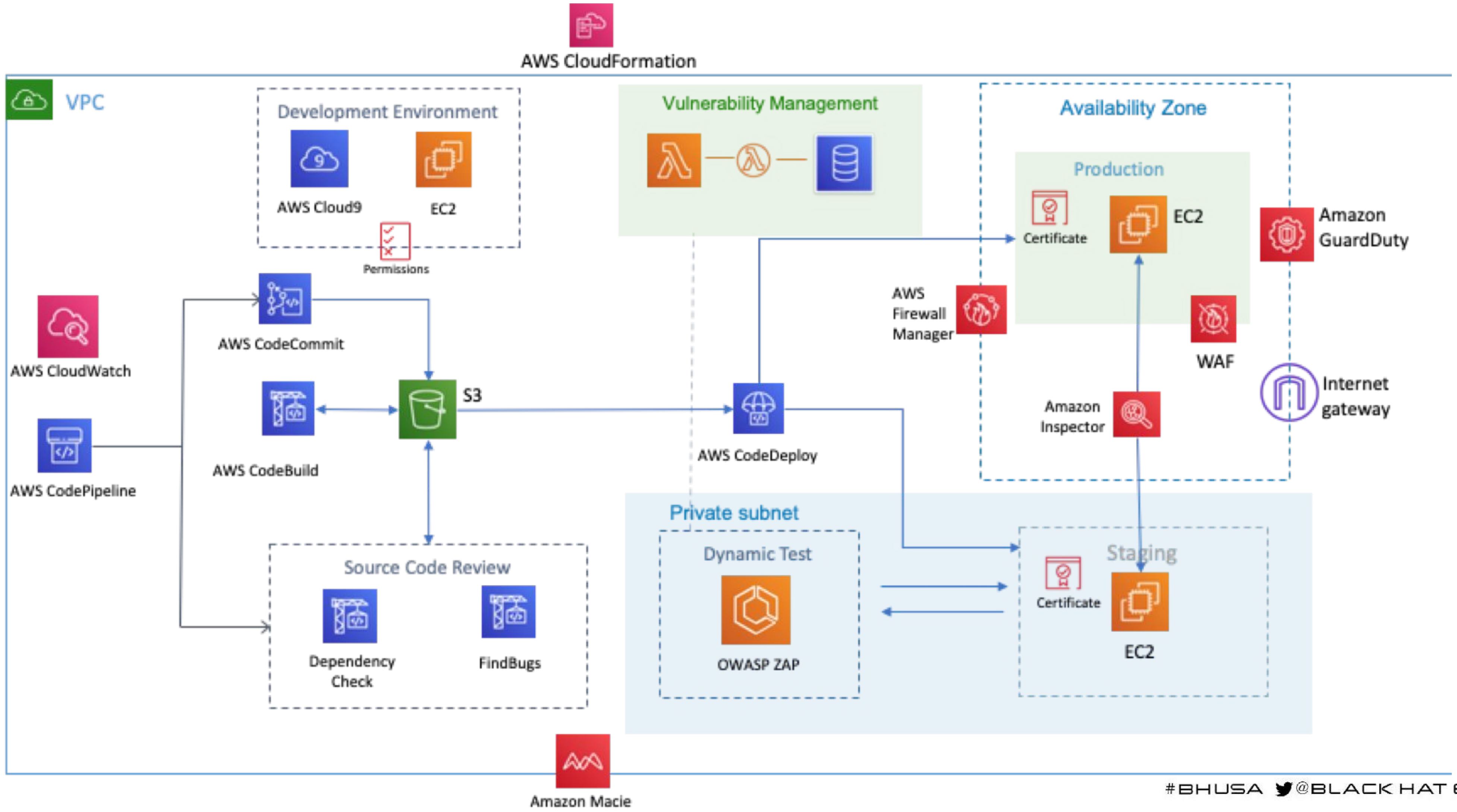
SECURITY MONKEY

- Infrastructure as Code allows quick audit / linting
- Focus more on:
  - Security groups
  - Permissions to resources
  - Rouge /shadow admins
  - Forgotten resources (compromises / billing)

Cloud Native Approach to Security

- Different Service Providers Approach Security Differently
- All of them provide some of the ingredient in-house
- Irrespective of Cloud provider some tools will still need to be sourced
  - Static Code Analysis Tool
  - Dynamic Code Analysis Tool
  - Software Composition Analysis
  - Vulnerability Management Tool

AWS Cloud Native DevSecOps

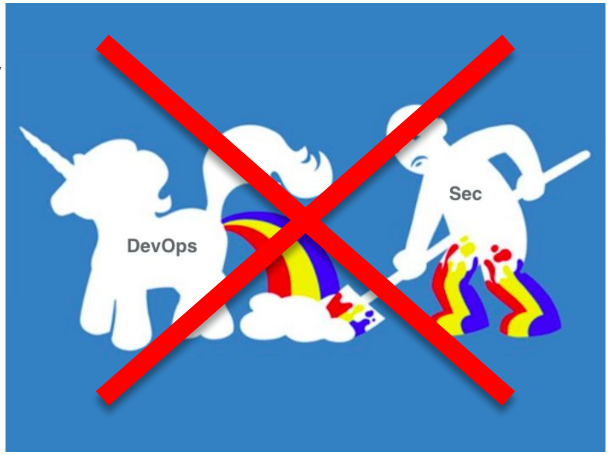


Cloud Native Dev[Sec]Ops

	Conventional Infra	AWS	Azure	GCP
Source Code Management	Bitbucket, Github, Gitlab etc..	AWS CloudCommit	Azure Repos	Cloud Source Repositories
Infrastructure As a Code	Chef, Puppet, Ansible more..	Amazon CloudFormation	Azure DevTest Labs	Cloud Code
CI/CD Server	Jenkins, Bamboo, Gitlab, Travis CI, Circleci more	AWS CodeBuild AWS CodeDeploy AWS CodePipeline	Azure Pipelines, Azure Test Plans	Cloud Build, Tekton
Artifactory Repository	jFrog Artifactory, Sonatype Nexus, more..	Amazon S3	Azure Artifacts	Cloud Firestore
Stg/Prod Servers	VMWare, On-premises servers	EC2 ECS (Elastic Containers) EKS (Elastic Kubernetes)	Virtual Machines, Azure Lab Services, Azure Kubernetes Service (AKS)	Compute Engine, App Engine, Shielded VMs
Monitoring & Alert	Nagios, Graphite, Grafana	AWS CloudWatch	Azure Monitor, Network Watcher	Access Transparency
Firewall	Modsecurity	AWS Firewall Manager, AWS WAF	Azure Firewall	Application Gateway
DLP	MyDLP, OpenDLP	Amazon Macie	Azure Information Protection	Cloud Data Loss Prevention
Threat Detection	Snort, Kismet	Amazon GuardDuty	Azure Advanced Threat Protection	Event Threat Detection (beta)
Vulnerability Scanning	OpenVAS, Nessus	Amazon Inspector	Azure Security Center	Cloud Security Scanner
Secrets Management	Hashicorp Vault, Docker Secrets	AWS Secrets Manager	Azure Key Vault	Secrets management

Cultural Aspect

- Automation alone will not solve the problems
- Focus on collaboration and inclusive culture
- Encourage security mindset specially if it's outside sec team
- Build allies (security champions) in company
- Avoid Blame Game



Security Champion

- Bridge between Dev, Sec and Ops teams
- Build Security Champions
  - **Single Person per team**
  - **Everyone provided with similar cross skilling opportunities**
  - **Incentivize other teams to collaborate with Sec team**
    - **Internal Bug bounties**
    - **Sponsor Interactions (Parties / get-togethers)**
    - **Sponsor cross skilling trainings for other teams**

Security Enablers

People

- Build relationships between teams, don't isolate
- Identify, nurture security conscious individuals
- Empower Dev/ops to deliver better and faster and secure, instead of blocking.
- Focus on solutions instead of blaming

Process

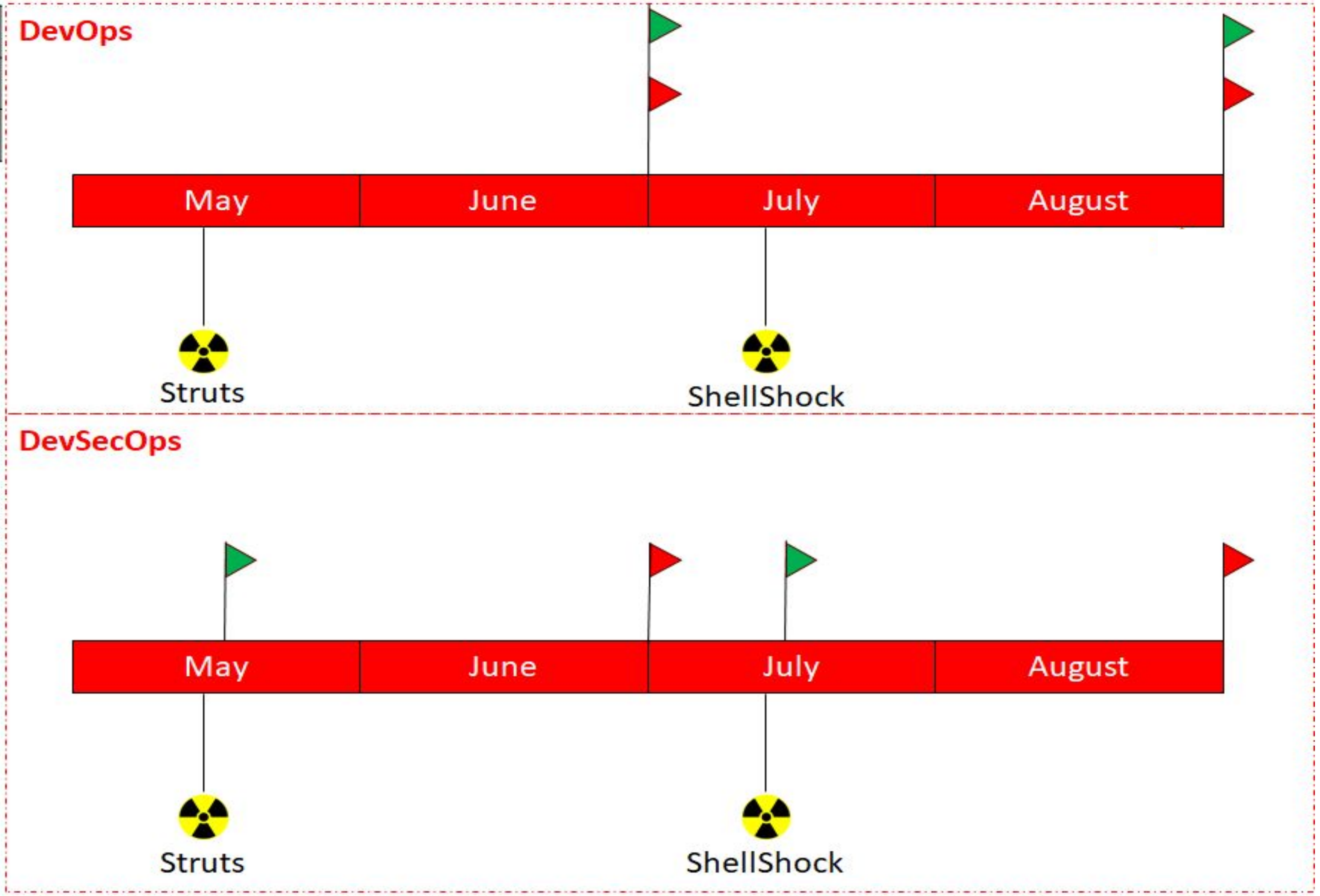
- Involve security from get-go (design or ideation phase)
- Fix by priority, don't attempt to fix it all
- Security Controls must be programmable and automated wherever possible
- DevSecOps Feedback process must be smooth and governed

Technology

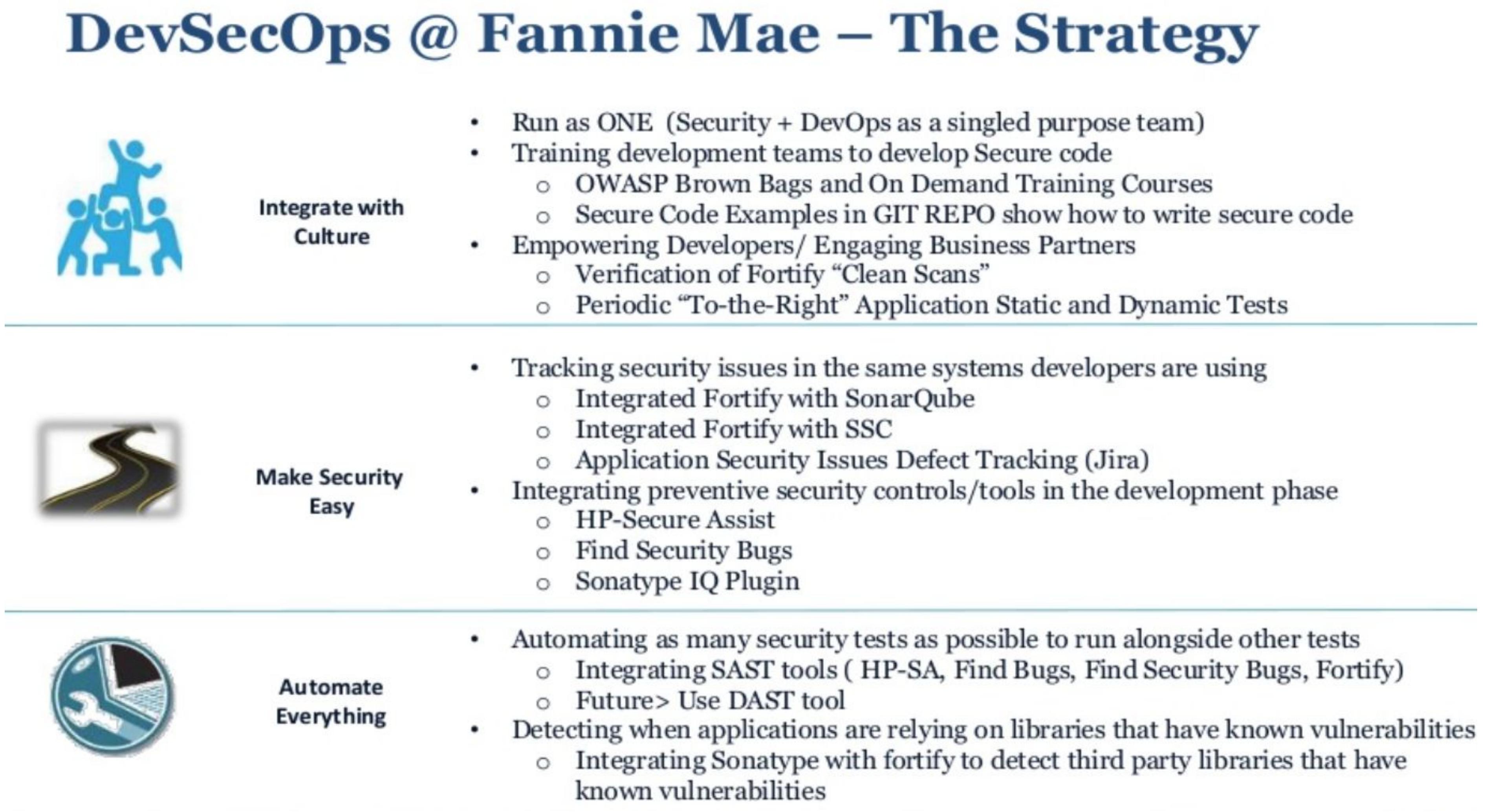
- Templatize scripts/tools per language/platform
- Adopt security to devops flow don't expect others to adopt security
- Keep an eye out for simpler and better options and be pragmatic to test and use new tools

Generic Case Study

▶	Manual Pentest
☢	Zero Day
▶	Zero Day Resolved



Case Studies – Fannie Mae



DevSecOps @ Fannie Mae – The Strategy

Integrate with Culture

- Run as ONE (Security + DevOps as a singled purpose team)
- Training development teams to develop Secure code
  - OWASP Brown Bags and On Demand Training Courses
  - Secure Code Examples in GIT REPO show how to write secure code
- Empowering Developers/ Engaging Business Partners
  - Verification of Fortify "Clean Scans"
  - Periodic "To-the-Right" Application Static and Dynamic Tests

Make Security Easy

- Tracking security issues in the same systems developers are using
  - Integrated Fortify with SonarQube
  - Integrated Fortify with SSC
  - Application Security Issues Defect Tracking (Jira)
- Integrating preventive security controls/tools in the development phase
  - HP-Secure Assist
  - Find Security Bugs
  - Sonatype IQ Plugin

Automate Everything

- Automating as many security tests as possible to run alongside other tests
  - Integrating SAST tools (HP-SA, Find Bugs, Find Security Bugs, Fortify)
  - Future> Use DAST tool
- Detecting when applications are relying on libraries that have known vulnerabilities
  - Integrating Sonatype with fortify to detect third party libraries that have known vulnerabilities

https://www.slideshare.net/DevSecOpsDays/fannie-mae-devsecops-journey-with-chitra-elango-and-john-willis

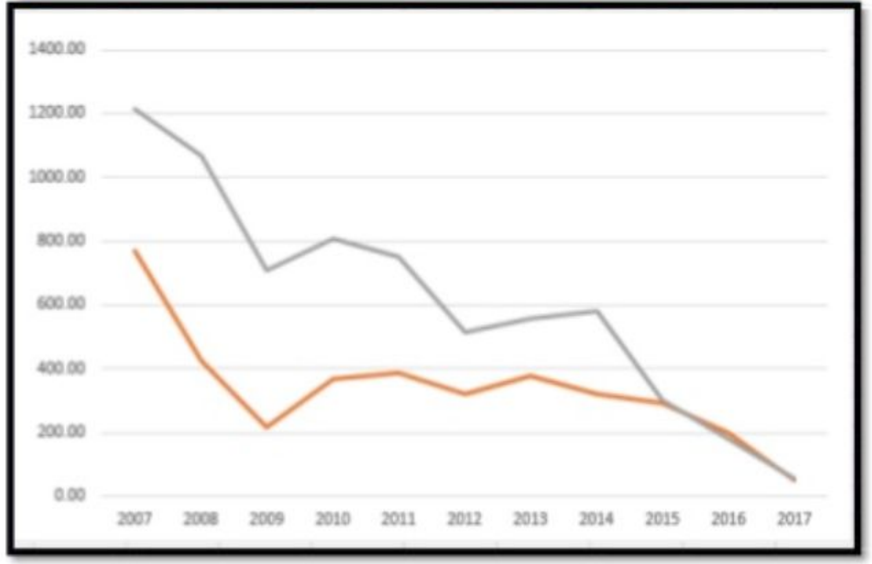
Case Studies – Fannie Mae

DevSecOps @ Fannie Mae – The Results

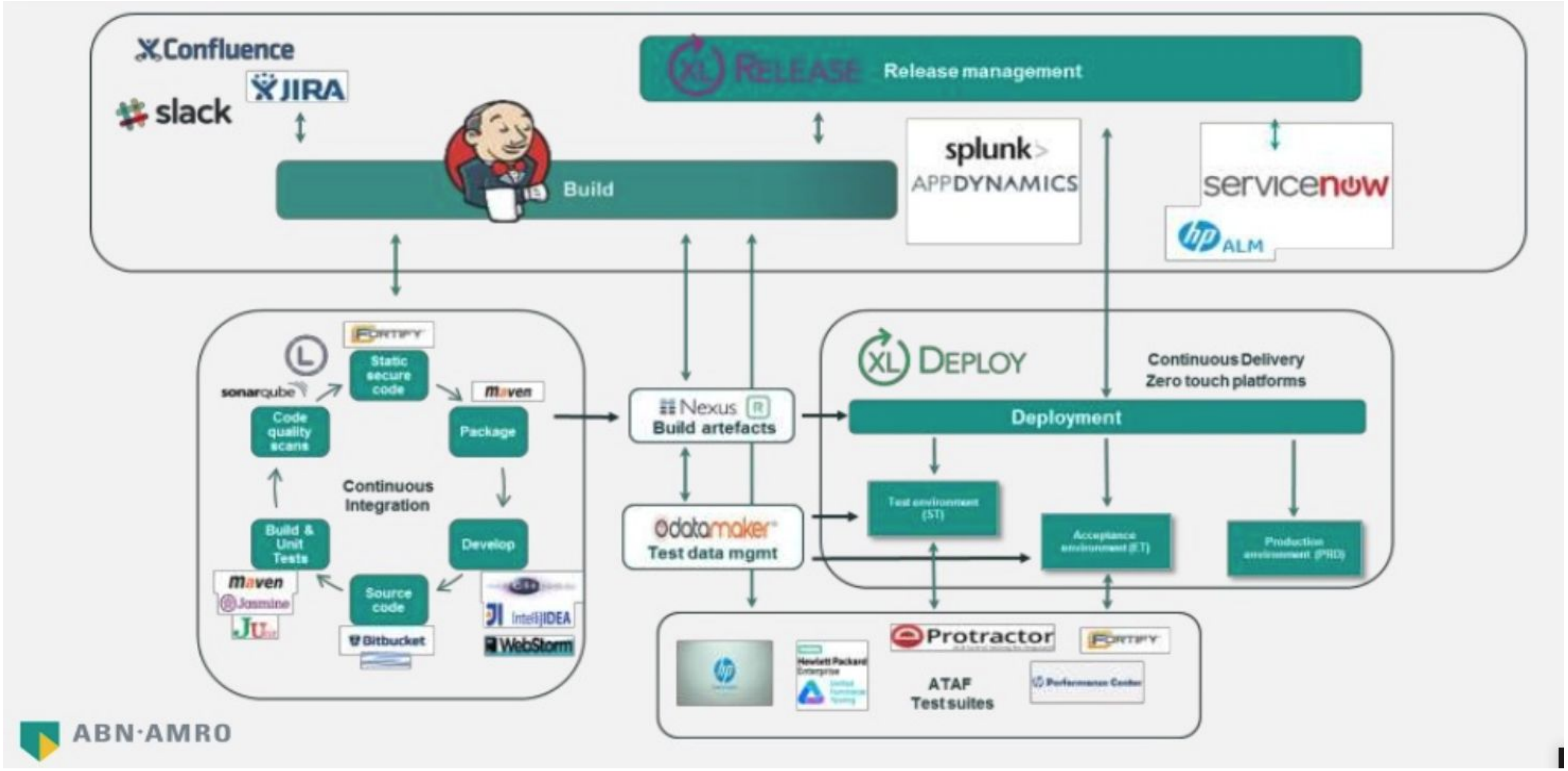
Delivering the Promise

- Average days to close a vulnerability improved by 74%
- Automated code quality scanning shows overall security code scores has increased by 10%
- More than 60% of application teams are performing security tests before release
- Critically vulnerable open source components (CVE 7.5+) downloaded has decreased from 18% to 6.25%
- ~ 55% of technical debt and security defects identified as a result of periodic testing have been dispositioned
- ~ 77% of older technical debt and security defects have been remediated, have a remediation plan in place, or have been addressed through managed retirements of assets

Average Days to Close a Security Vulnerability

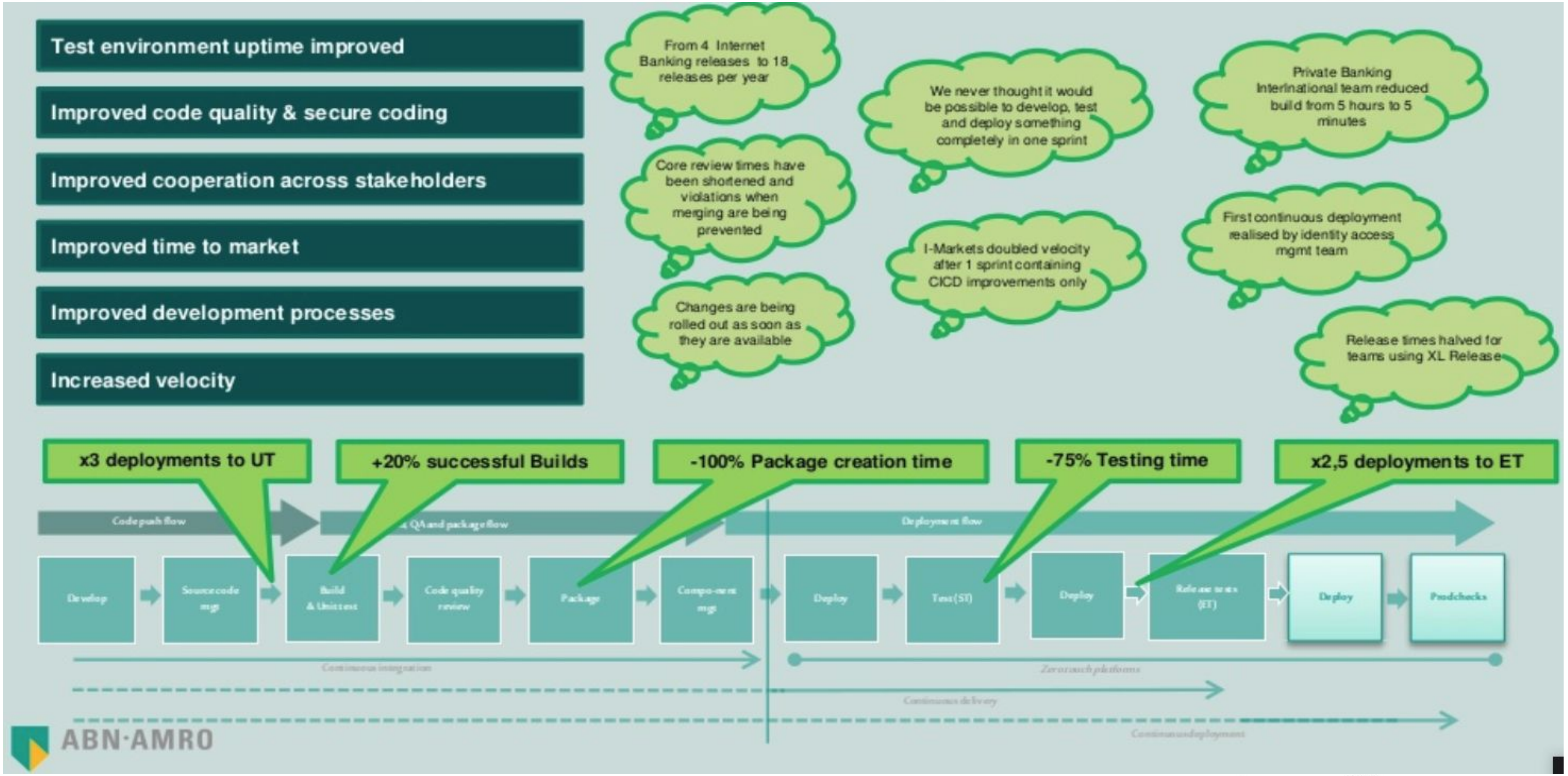


Case Studies – ABN Amro



https://www.slideshare.net/derweeksglobal/abn-amro-devsecops-journey

Case Studies – ABN Amro



Test environment uptime improved

Improved code quality & secure coding

Improved cooperation across stakeholders

Improved time to market

Improved development processes

Increased velocity

From 4 Internet Banking releases to 18 releases per year

Core review times have been shortened and violations when merging are being prevented

Changes are being rolled out as soon as they are available

We never thought it would be possible to develop, test and deploy something completely in one sprint

I-Markets doubled velocity after 1 sprint containing CICD improvements only

Private Banking International team reduced build from 5 hours to 5 minutes

First continuous deployment realised by identity access mgmt team

Release times halved for teams using XL Release

x3 deployments to UT

+20% successful Builds

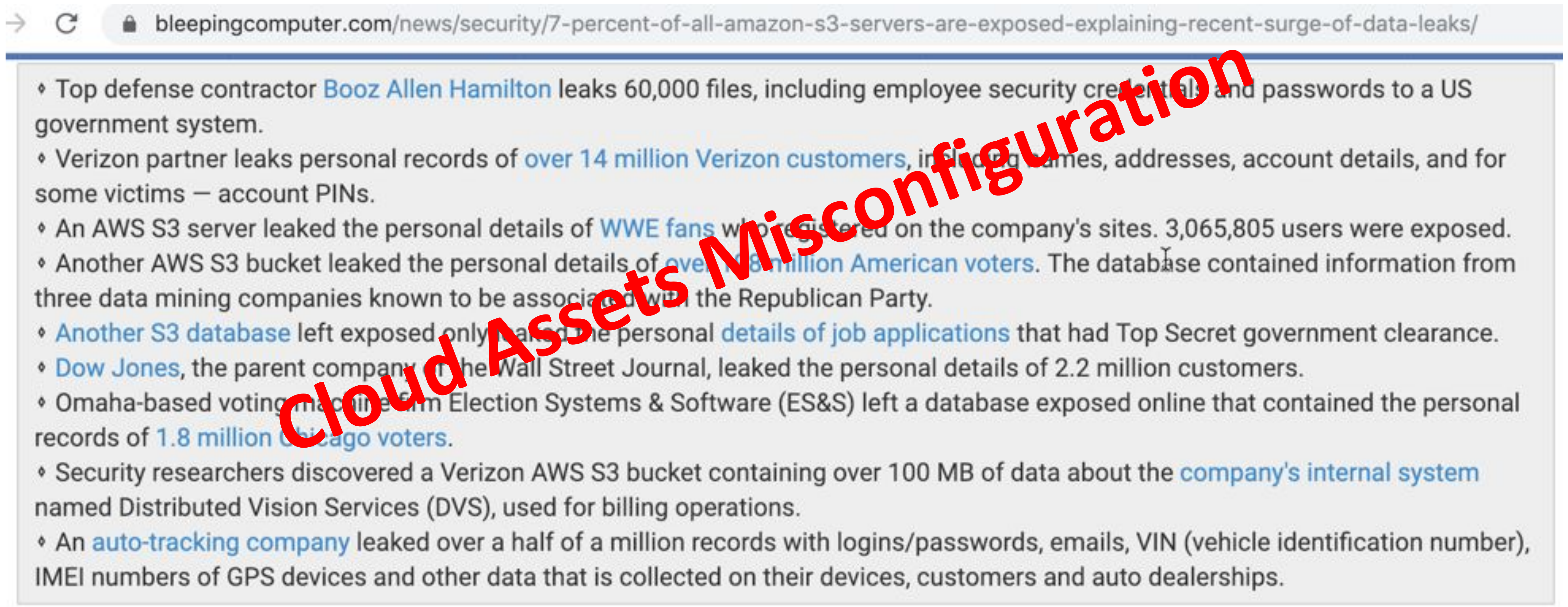
-100% Package creation time

-75% Testing time

x2,5 deployments to ET

https://www.slideshare.net/derweeksglobal/abn-amro-devsecops-journey

Negative Case Studies



bleepingcomputer.com/news/security/7-percent-of-all-amazon-s3-servers-are-exposed-explaining-recent-surge-of-data-leaks/

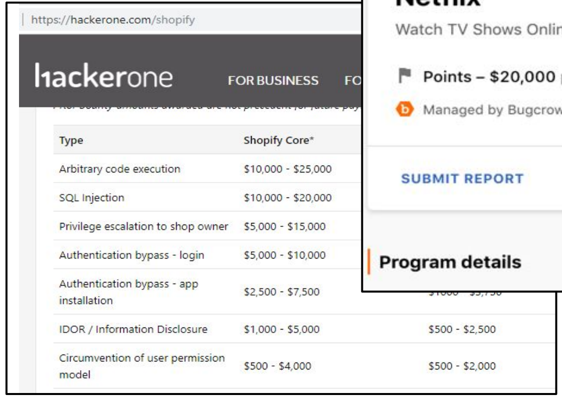
- Top defense contractor Booz Allen Hamilton leaks 60,000 files, including employee security credentials and passwords to a US government system.
- Verizon partner leaks personal records of over 14 million Verizon customers, including names, addresses, account details, and for some victims — account PINs.
- An AWS S3 server leaked the personal details of WWE fans who registered on the company's sites. 3,065,805 users were exposed.
- Another AWS S3 bucket leaked the personal details of over 198 million American voters. The database contained information from three data mining companies known to be associated with the Republican Party.
- Another S3 database left exposed only leaked the personal details of job applications that had Top Secret government clearance.
- Dow Jones, the parent company of the Wall Street Journal, leaked the personal details of 2.2 million customers.
- Omaha-based voting machine firm Election Systems & Software (ES&S) left a database exposed online that contained the personal records of 1.8 million Chicago voters.
- Security researchers discovered a Verizon AWS S3 bucket containing over 100 MB of data about the company's internal system named Distributed Vision Services (DVS), used for billing operations.
- An auto-tracking company leaked over a half of a million records with logins/passwords, emails, VIN (vehicle identification number), IMEI numbers of GPS devices and other data that is collected on their devices, customers and auto dealerships.

Cloud Assets Misconfiguration

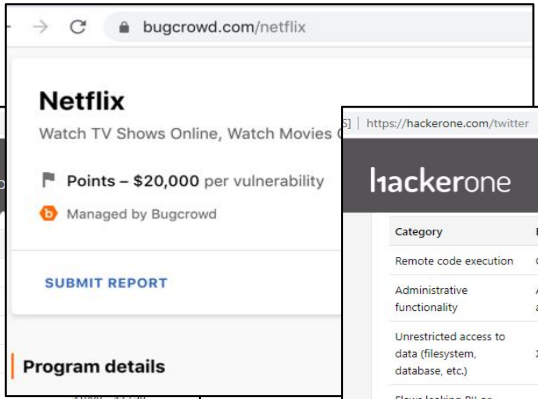
Prevention: Continuous monitoring and review of cloud assets and config

Is it Enough?

- Rite of passage by periodic pen test and continuous bug bounty
- It's not just important to get feedback but to also action on them
- Risk Acceptance Documentation should be the worst case scenario not your first bet


https://hackerone.com/shopify

Type	Shopify Core*	
Arbitrary code execution	$10,000 - $25,000	
SQL Injection	$10,000 - $20,000	
Privilege escalation to shop owner	$5,000 - $15,000	
Authentication bypass - login	$5,000 - $10,000	
Authentication bypass - app installation	$2,500 - $7,500	
IDOR / Information Disclosure	$1,000 - $5,000	$500 - $2,500
Circumvention of user permission model	$500 - $4,000	$500 - $2,000


bugcrowd.com/netflix

Netflix

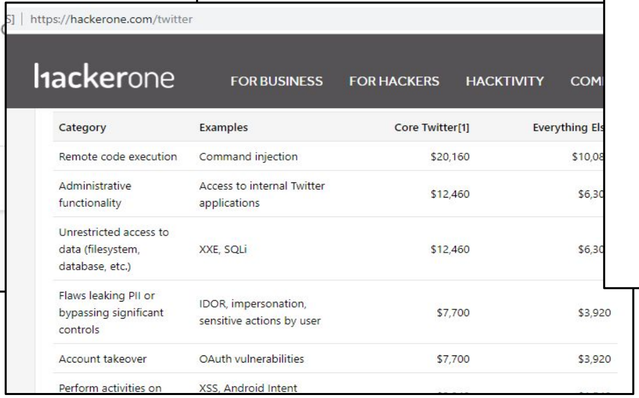
Watch TV Shows Online, Watch Movies

Points – $20,000 per vulnerability

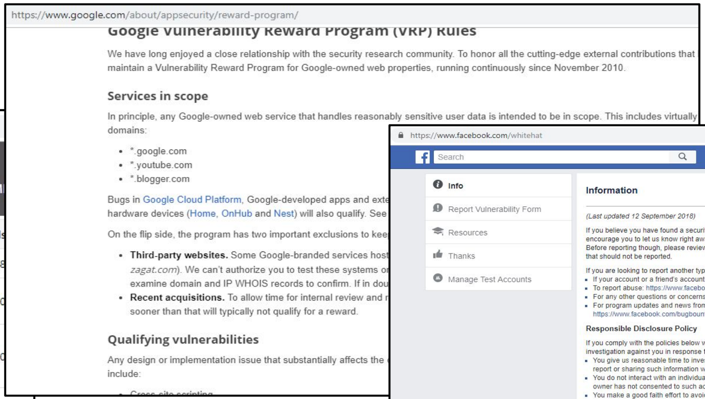
Managed by Bugcrowd

SUBMIT REPORT

Program details


https://hackerone.com/twitter

Category	Examples	Core Twitter[1]	Everything Els
Remote code execution	Command injection	$20,160	$10,08
Administrative functionality	Access to internal Twitter applications	$12,460	$6,30
Unrestricted access to data (filesystem, database, etc.)	XXE, SQLi	$12,460	$6,30
Flaws leaking PII or bypassing significant controls	IDOR, impersonation, sensitive actions by user	$7,700	$3,920
Account takeover	OAuth vulnerabilities	$7,700	$3,920
Perform activities on	XSS, Android Intent		


https://www.google.com/about/appsecurity/reward-program/

Google Vulnerability Reward Program (VRP) Rules

We have long enjoyed a close relationship with the security research community. To honor all the cutting-edge external contributions that maintain a Vulnerability Reward Program for Google-owned web properties, running continuously since November 2010.

Services in scope

In principle, any Google-owned web service that handles reasonably sensitive user data is intended to be in scope. This includes virtually domains:

- *.google.com
- *.youtube.com
- *.blogger.com

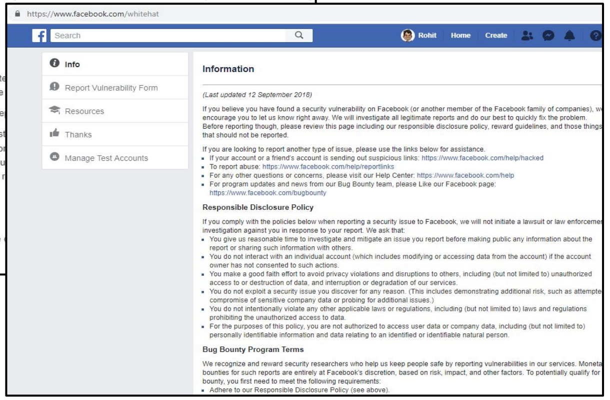
Bugs in Google Cloud Platform, Google-developed apps and exte hardware devices (Home, OnHub and Nest) will also qualify. See

On the flip side, the program has two important exclusions to kee

- **Third-party websites.** Some Google-branded services host *zagat.com*). We can't authorize you to test these systems o examine domain and IP WHOIS records to confirm. If in dou
- **Recent acquisitions.** To allow time for internal review and r sooner than that will typically not qualify for a reward.

Qualifying vulnerabilities

Any design or implementation issue that substantially affects the include:


https://www.facebook.com/whitehat

Info
Report Vulnerability Form
Resources
Thanks
Manage Test Accounts

Information

(Last updated 12 September 2018)

If you believe you have found a security vulnerability on Facebook (or another member of the Facebook family of companies), we encourage you to let us know right away. We will investigate all legitimate reports and do our best to quickly fix the problem. Before reporting though, please review this page including our responsible disclosure policy, reward guidelines, and those things that should not be reported.

If you are looking to report another type of issue, please use the links below for assistance.
- If your account or a friend's account is sending out suspicious links: https://www.facebook.com/help/hacked
- To report abuse: https://www.facebook.com/help/reportlinks
- For any other questions or concerns, please visit our Help Center: https://www.facebook.com/help
- For program updates and news from our Bug Bounty team, please Like our Facebook page: https://www.facebook.com/bugbounty

Responsible Disclosure Policy

If you comply with the policies below when reporting a security issue to Facebook, we will not initiate a lawsuit or law enforcemen investigation against you in response to your report. We ask that:
- You give us reasonable time to investigate and mitigate an issue you report before making public any information about the report or sharing such information with others.
- You do not interact with an individual account (which includes modifying or accessing data from the account) if the account owner has not consented to such actions.
- You make a good faith effort to avoid privacy violations and disruptions to others, including (but not limited to) unauthorized access to or destruction of data, and interruption or degradation of our services.
- You do not exploit a security issue you discover for any reason. (This includes demonstrating additional risk, such as attempte compromise of sensitive company data or probing for additional issues.)
- You do not intentionally violate any other applicable laws or regulations, including (but not limited to) laws and regulations prohibiting the unauthorized access to data.
- For the purposes of this policy, you are not authorized to access user data or company data, including (but not limited to) personally identifiable information and data relating to an identified or identifiable natural person.

Bug Bounty Program Terms

We recognize and reward security researchers who help us keep people safe by reporting vulnerabilities in our services. Moneta bounties for such reports are entirely at Facebook's discretion, based on risk, impact, and other factors. To potentially qualify for bounty, you first need to meet the following requirements:
- Adhere to our Responsible Disclosure Policy (see above).

References

- https://www.blackhat.com/docs/us-17/thursday/us-17-Lackey-Practical%20Tips-for-Defending-Web-Applications-in-the-Age-of-DevOps.pdf
- https://www.sonatype.com/hubfs/2018%20State%20of%20the%20Software%20Supply%20Chain%20Report.pdf
- https://snyk.io/opensourcesecurity-2019/
- https://www.veracode.com/state-of-software-security-report

Key Takeaways

- Security is everyone responsibility
- Embrace security as an integral part of the process, use feedback to refine the process
- DevSecOps is not a one size fit all: your mileage will vary

Questions and Discussion

Questions
&
feedback
devsecops@notsosecure.com