DevSecCon

Why 'Positive Security' is the next software security game changer

Jaap Karan Singh

jaap@scw.io

Co-Founder and Chief Singh, Secure Code Warrior

Singapore | 28 Feb - 01 Mar 2019

> Work for SECURE CODE WARRIOR

We empower developers to write secure code

> Developer >> Pentester >> Developer

> Help organisations build kick-ass training awareness programs

DevSecCon

> Today's challenges with software security

22M

Software developers around the world ~ Evans Data

Source: https://evansdata.com/reports/viewRelease.php?reportID=9

111BN

Lines of code written by developers every year ~ CSO Online

Source: https://www.csoonline.com/article/3151003/application-development/world-will-need-to-secure-111-billion-lines-of-new-software-code-in-2017.html

1 to 4

Exploitable Security Bugs in every 50 000 Lines of Code

Source: StackOverflow

90%

Security incidents result from defects in the design or code ~ DHS

Source: https://www.us-cert.gov/sites/default/files/publications/infosheet_SoftwareAssurance.pdf

21%

Of data breaches caused by software vulnerability ~ Verizon

Source: Verizon, Data Breach Report, 2018 (but in there the last 10 years)

1 in 3

of newly scanned applications had SQL injections over the past 5 yrs ~ Cisco

Source: Cybersecurity as a Growth Advantage, Cisco, 2016

CRY
CRY

DevSecCon

> How did we end up here?

AppSec in 2000

Corporates had a branding website, the Internet was mostly for geeks

> *AppSec was virtually non-existent in corporate world*

> *Hacking was focussed on exploiting infrastructure vulnerabilities (bof, race conditions, fmt str*)*

> *Research on first web app weaknesses*

> *OWASP started and Top 10 released!*

> Penetration testing was black magic

We’ve got bigger problems (Y2K) than worrying about Application Security

DevSecCon

AppSec in 2010

Companies started offering web-based services; Web 2.0 and Mobile are new

> Penetration testing was THE thing

> Web Application Firewalls will stop everything

> Paper-based secure coding guidelines

> Static Code Analysis Tools (SAST) emerge

Monthly data breaches,
Hackers everywhere,
Privacy, GDPR, PCI-DSS, HIPAA
Putin

AppSec in 2019

Everything runs on software.
Cybersecurity & AppSec are hot topics.

> Pen-testing is still here…

> Static Code Analysis Tools (SAST) is still here…

> Runtime Application Security Protection (RASP)

> Dynamic Application Security Testing (DAST)

> Interactive Application Security Testing (IAST)

> Crowd-Sourced Security Testing *(CSST?)*

> **DevSecOps** is getting traction
- Shift left
- Containerisation
- Integrating security and ops into dev
- Security pipelining

AppSec in 2019

Challenge - Pen-testing mostly sucks

Security Experts

Developers

DevSecCon

BUILDERS

Know their code

Do not speak “security”

- JAVA Spring
- Constructors
- SWIFT
- Angular.JS

vs

BREAKERS

Always pointing out problems

Not developers

- SQL Injections
- Object Deserialization
- XSS
- IDOR

AppSec in 2019

Challenge - AppSec is often a bottleneck

DevSecCon

Software Developers (Agile)

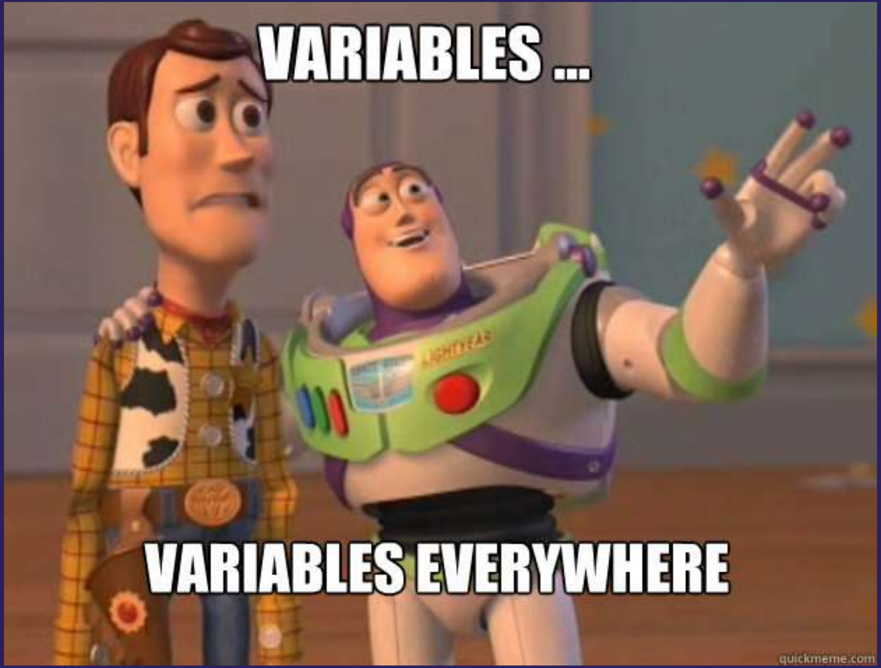
200

Application Security Experts

1

AppSec in 2019

Challenge - Security Pipelining is in its infancy



AppSec in 2019

Challenge - Tools mostly suck

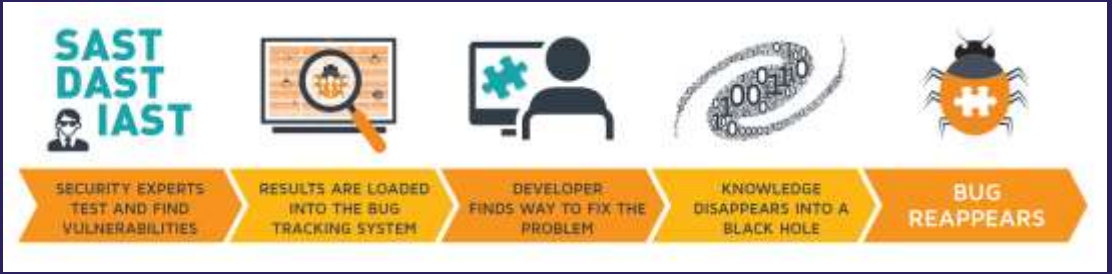
> SAST - Expertise, false positives, slow, framework support

> I/DAST - Expertise, false negatives, slow

> RASP - WAF++, nobody uses block mode, tech specific

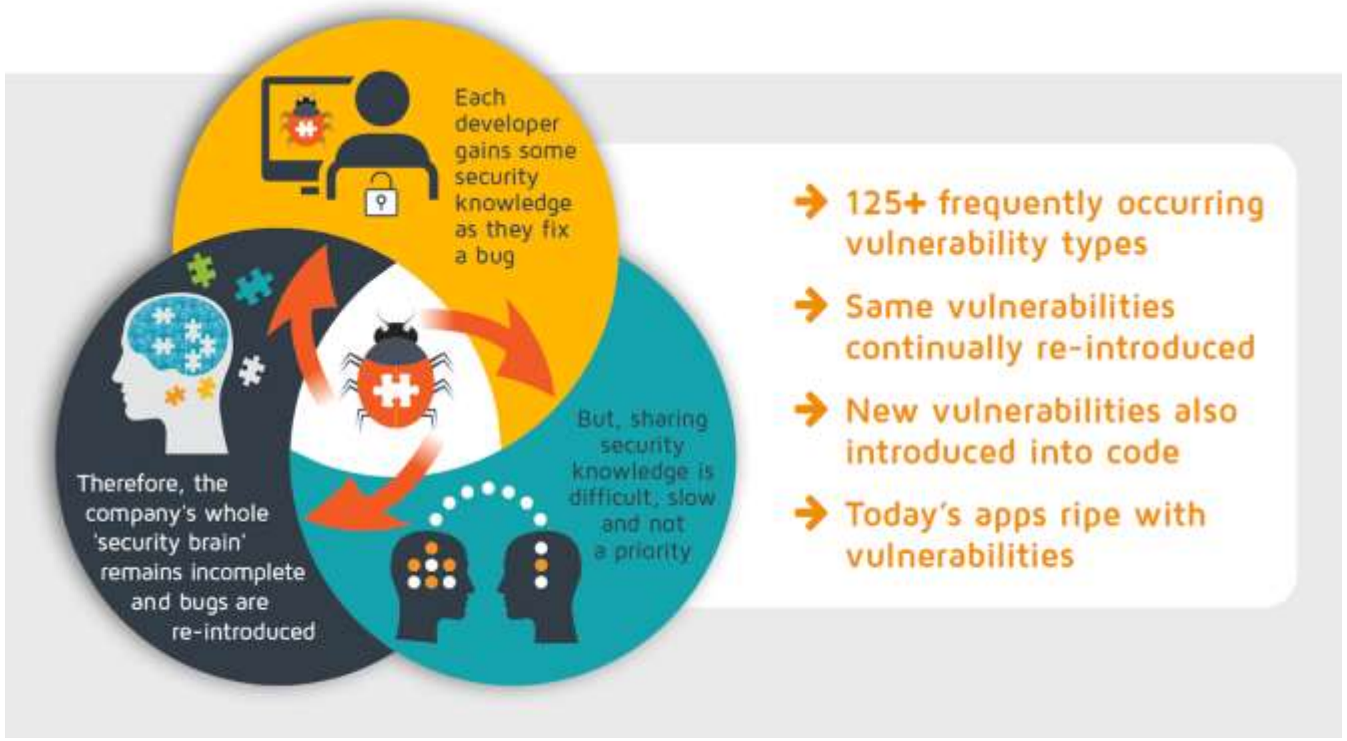
> Testing tools spit out long, mostly inaccurate reports with often useless advice

Challenge - “Black Hole” of security knowledge


SAST
DAST
IAST
SECURITY EXPERTS TEST AND FIND VULNERABILITIES
RESULTS ARE LOADED INTO THE BUG TRACKING SYSTEM
DEVELOPER FINDS WAY TO FIX THE PROBLEM
KNOWLEDGE DISAPPEARS INTO A BLACK HOLE
BUG REAPPEARS

AppSec in 2019

We’re failing to learn from our mistakes


Each developer gains some security knowledge as they fix a bug
But, sharing security knowledge is difficult, slow and not a priority
Therefore, the company's whole 'security brain' remains incomplete and bugs are re-introduced

- 125+ frequently occurring vulnerability types
- Same vulnerabilities continually re-introduced
- New vulnerabilities also introduced into code
- Today’s apps ripe with vulnerabilities

AppSec @ Work

DevSecCon

> ~~SHIFT~~ START left

Scale and Make an Impact as an AppSec Pro

Singapore | 28 Feb - 01 Mar 2019

~~SHIFT~~ START left

Solution – Better Pen-Testing

> Bobby'; DROP TABLE pentesting_attitude;

> Provide a FIX more than input_validation();

> Create a JIRA ticket with advise/fix

> Create a pull request (wishful thinking)

> Lessons Learned to dev teams to distribute knowledge

Less finding problems, more security engineering

~~SHIFT~~ START left

Solution – Weaknesses vs Controls

OWASP

OWASP Top 10 - 2017

The Ten Most Critical Web Application Security Risks

https://owasp.org

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

creative commons

OWASP

OWASP PRO Active CONTROLS

FOR DEVELOPERS

2018 v 3.0

10 Critical Security Areas That Software Developers Must Be Aware Of

PROJECT LEADERS

[illegible]

Solution – Distribute Knowledge

Application Security

1

Secure Coding Guidelines

e.g.

- Ensure application logging (Where, What, When, Who, Why)
- Use context encoding on untrusted user input

DevSecCon

Solution – Distribute Knowledge

Secure Coding Guidelines

1. *Ensure application logging (Where, What, When, Who, Why)*
2. *Use context encoding on untrusted user input*

200

Project X - Secure Coding rules for
<insert your favourite coding framework>

1. Use SecureLogger log_object;
2. Don’t use GetParameter(), Use LibSafe_GetParam()

Solution – Distribute Knowledge

Secure Coding Guidelines

1. *Ensure application logging (Where, What, When, Who, Why)*
2. *Use context encoding on untrusted user input*

Project X - Secure Coding rules for
<insert your favourite coding framework>

1. Use SecureLogger log_object;
2. Don’t use GetParameter(), Use LibSafe_GetParam()

200

Upon Commit

1. Your code violates security rules: You shall not pass!
2. Your code violates security rules: Fill in your get out of jail card (JIRA ticket)
3. Points++ for delivering secure code

Solution – Learn from Mistakes

Application Security

1

Security Vulnerabilities

- *Sensitive data not transported securely*

Developer fixes issue

- Use TLS() for any sensitive data

Solution – Learn from Mistakes

Security Vulnerabilities
- *Sensitive data not transported securely*

Developer fixes issue
- Use TLS() for any sensitive data

200

Project X - Secure Coding rules for
<insert your favourite coding framework>

1. Use SecureLogger log_object;
2. Don’t use GetParameter(), Use LibSafe_GetParam()
3. *Use TLS() for any sensitive data*

DevSecCon

DevSecCon

> Build a positive security culture

Break down “us” vs “them” culture

Positive

Security Culture

Create a brand

> People remember a memorable brand

> Make it fun and geeky!

> AppSec are not marketing experts, get help from Security Awareness

GAME OF CODES

Answer the “why”

> Teachable moments

> Make it personal

Positive

Security Culture

Positive
Security Culture

Build a community

> Special interest group for those interested in AppSec and cyber security

> Not a one-time event, self-sustaining community that carries the culture forward

> Fun events and competitions – write your best phishing email, lock picking, hack internal applications

Security Champions

Jane Doe

> Interested in AppSec

> Great grasp of security concepts

> coding_skills++ - best coder in the team

> Well respected by peers

> Not part of other communities

Works with AppSec doing security engineering

John Smith

> Interested in AppSec

> Good grasp of security concepts

> Good coding skills

> Well liked by peers

> Part of internal communities

Helps spread the word and drive behaviour change

Positive Security Culture

Reward good behaviour

> Cash prize - reward developers for finding security bugs you would pay pen-tester for

> Level up program

> Peer and executive recognition

> Speeding pass - prove security awareness, introduce security pipelining and skip manual security checks

Positive
Security Culture

Remember – it's not easy!

> Crawl…walk…RUN

> Visible management buy-in

> Harder to change mindset of existing employees, easier to introduce to new starters

If at first you don't succeed, try again

Takeaways:

- Demand better outcomes in security testing
- Distribute knowledge to scale AppSec
- Build a positive security culture
- Reward good behaviour

Secure Developers Are Superheroes

DevSecCon

> the_end();

Jaap Karan Singh

jaap@scw.io

Co-Founder & Chief Singh, Secure Code Warrior

Singapore | 28 Feb - 01 Mar 2019