SECURE CODE WARRIOR

Why 'Positive Security' is the Next Security Game Changer

Jaap Karan Singh

jaap@scw.io

Co-Founder & Chief Singh, Secure Code Warrior

> 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

SECURE CODE WARRIOR

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

SECURE CODE WARRIOR

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

SECURE CODE WARRIOR

Statistics can prove anything?

Yes, but they can’t all be THAT wrong

> How did we end up here?

SECURE CODE WARRIOR

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

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.

> 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
- Containerisation
- Integrating security and ops into dev
- Security pipelining

> **SHIFT Left**

http://www.redkid.net/generator/soup/

Challenge:

Right to Left is Backwards

“We want to provide a service that ***transports stuff from A to B****”*

Civil Engineering

Software Engineering

Customers

Security Experts

Secure Dev

30x

more expensive to fix vulnerabilities at the end of the development cycle

Source: National Institute of Standards and Technology (NIST), www.peoplesec.org

SECURE CODE WARRIOR

Challenge:

Security vs Development is an unfair game

Developers without security skills
BIG problem!

Security experts without coding skills
SMALL chance of success.

Software Developers (Agile)

A
B
A
B
A
B
B
A
A
B
A
B
100

Application Security Experts

1

Developers and Security speak different languages

SECURE CODE WARRIOR

BUILDERS

Know their code

Do not speak security

- JAVA Spring
- Constructors
- SWIFT
- Angular.JS

BREAKERS

Always points out problems

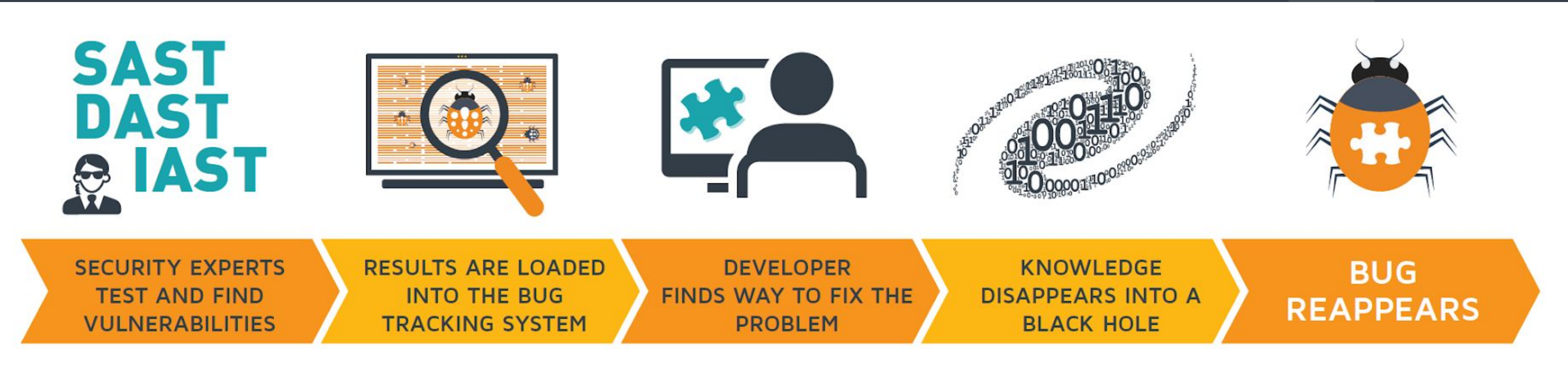
Not Developers

- SQL Injections
- Object Deserialization
- XSS
- IDOR

SECURE CODE WARRIOR

Challenge:

“Black Hole” of security knowledge



SECURITY EXPERTS TEST AND FIND VULNERABILITIES

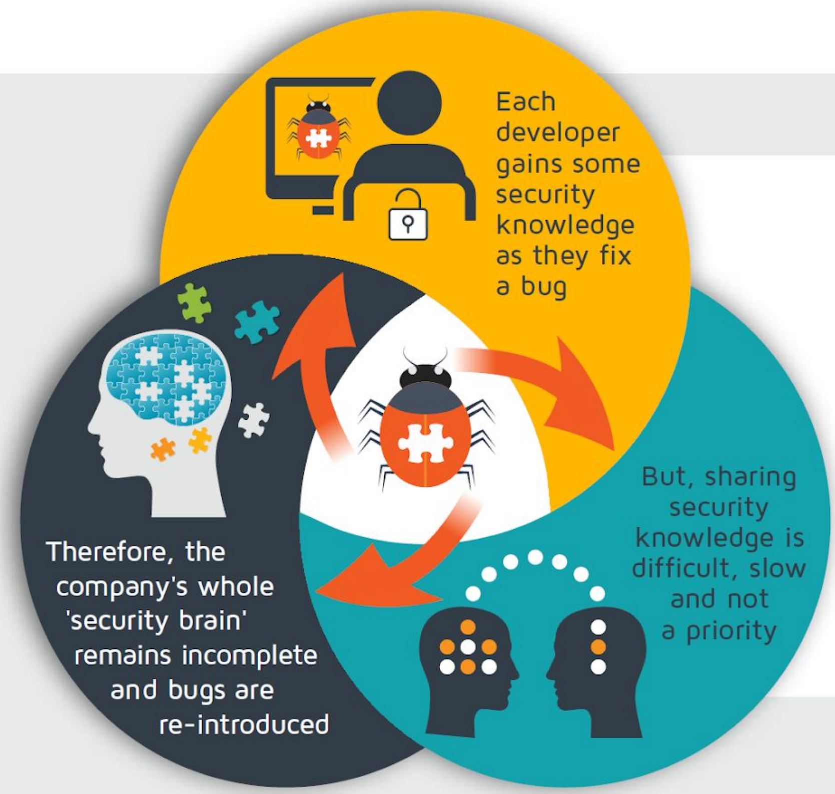
RESULTS ARE LOADED INTO THE BUG TRACKING SYSTEM

DEVELOPER FINDS WAY TO FIX THE PROBLEM

KNOWLEDGE DISAPPEARS INTO A BLACK HOLE

BUG REAPPEARS

We’re failing in Learning 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

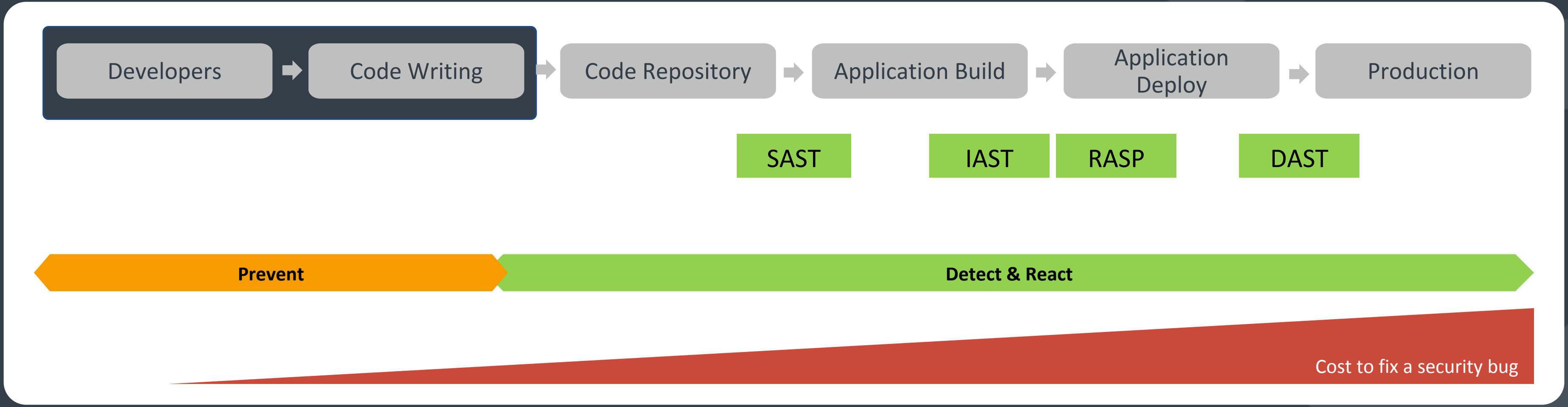
- 125+ frequently occurring vulnerability types
- Same vulnerabilities continually re-introduced
- New vulnerabilities also introduced into code
- Today’s apps ripe with vulnerabilities

AppSec @ Work

Solution

Empower developers to code securely

Shift security to the left



Developers → Code Writing → Code Repository → Application Build → Application Deploy → Production

SAST

IAST

RASP

DAST

Prevent

Detect & React

Cost to fix a security bug

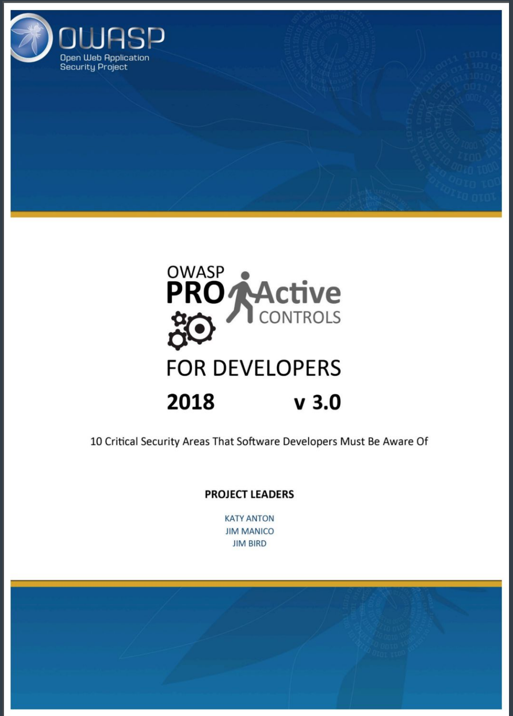
Weaknesses vs Controls

OWASP

OWASP Top 10 - 2017

The Ten Most Critical Web Application Security Risks

https://owasp.org



OWASP
Open Web Application
Security Project

OWASP
PRO Active
CONTROLS
FOR DEVELOPERS
2018 v 3.0

10 Critical Security Areas That Software Developers Must Be Aware Of

PROJECT LEADERS

KATY ANTON
JIM MANICO
JIM BIRD

OWASP
Open Web Application
Security Project

Application Security Verification Standard 3.0

October 2015

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

Distribute Knowledge

Secure Coding Guidelines

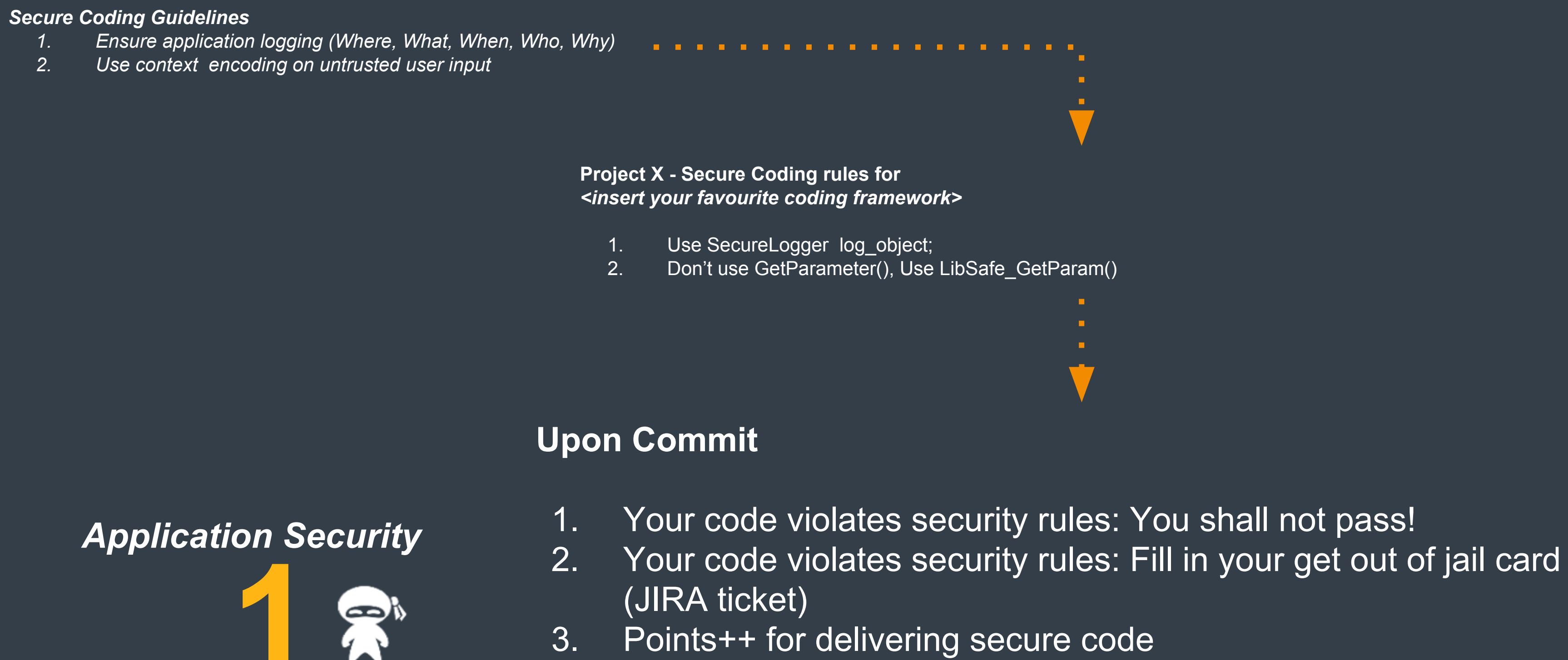
1. *Ensure application logging (Where, What, When, Who, Why)*
2. *Use context encoding on untrusted user input*

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()

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()

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

Application Security

1

Learn from Mistakes

Application Security

1

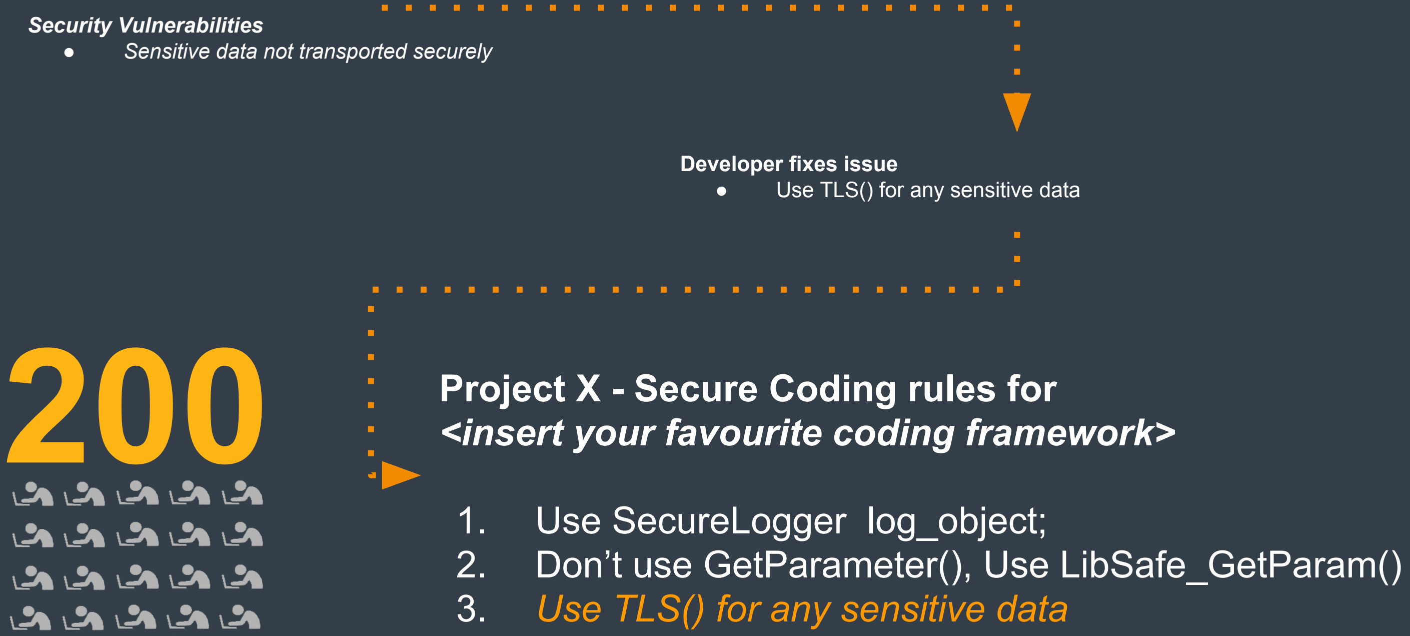
Security Vulnerabilities

- *Sensitive data not transported securely*

Developer fixes issue

- Use TLS() for any sensitive data

Learn from Mistakes



Security Vulnerabilities

- *Sensitive data not transported securely*

Developer fixes issue

- Use TLS() for any sensitive data

200

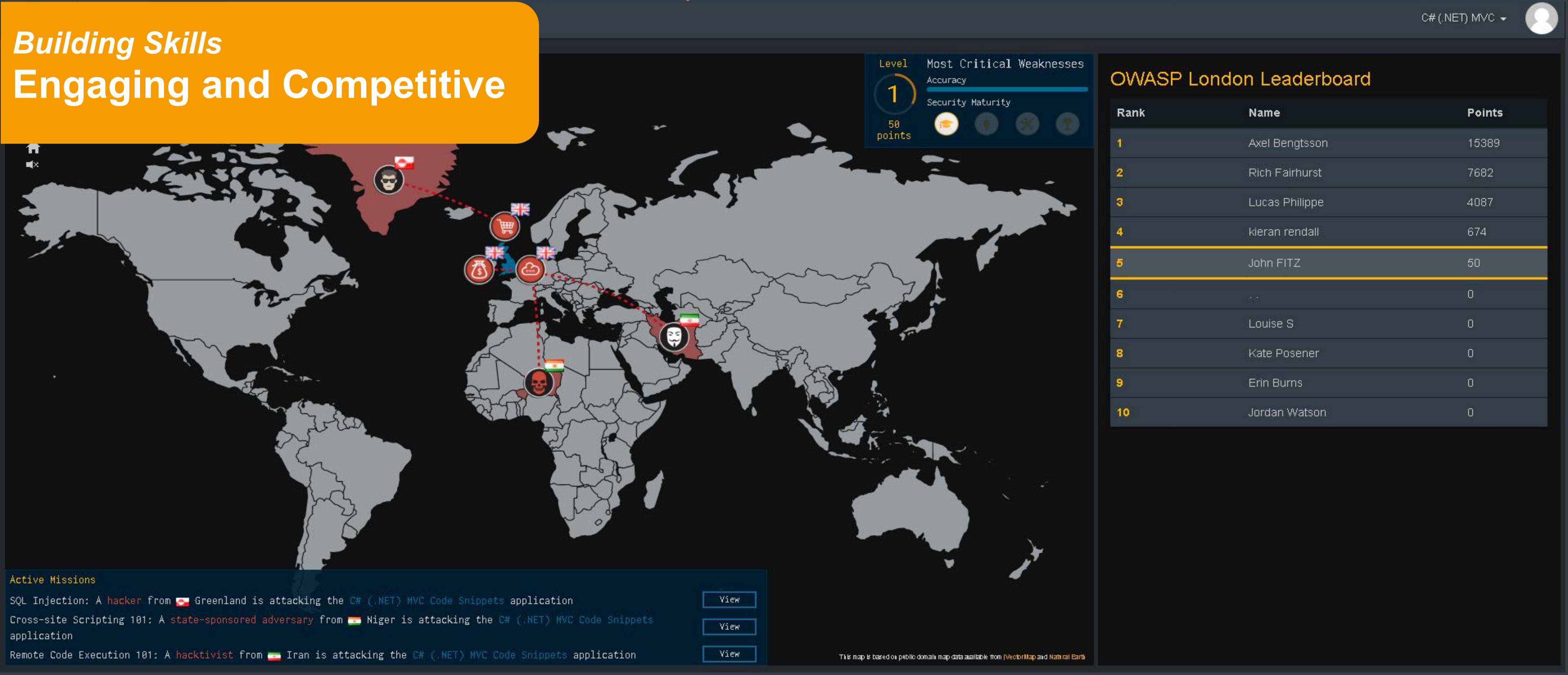
Project X - Secure Coding rules for
<insert your favourite coding framework>

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

SECURE CODE WARRIOR

Building Skills

Engaging and Competitive



C# (.NET) MVC
This map is based on public domain map data available from jVectorMap and Natural Earth

Level

1

50 points

Most Critical Weaknesses

Accuracy

Security Maturity

OWASP London Leaderboard

Rank	Name	Points
1	Axel Bengtsson	15389
2	Rich Fairhurst	7682
3	Lucas Philippe	4087
4	kieran rendall	674
5	John FITZ	50
6	. .	0
7	Louise S	0
8	Kate Posener	0
9	Erin Burns	0
10	Jordan Watson	0

Active Missions

SQL Injection: A hacker from Greenland is attacking the C# (.NET) MVC Code Snippets application View

Cross-site Scripting 101: A state-sponsored adversary from Niger is attacking the C# (.NET) MVC Code Snippets application View

Remote Code Execution 101: A hacktivist from Iran is attacking the C# (.NET) MVC Code Snippets application View

Working hard (or hardly working!)

SECURE CODE WARRIOR

Takeaways:

- Focus on positives such as security fundamentals
- Distribute knowledge to scale AppSec
- Define good patterns and re-use
- Put some fun into everything

Secure Developers Are Heroes

SECURE CODE

WARRIOR

SECURECODEWARRIOR.COM

INSIGHTS.SECURECODEWARRIOR.COM

@SECCODEWARRIOR

LINKEDIN.COM/COMPANY/SECURE-CODE-WARRIOR

FACEBOOK.COM/SECURECODEWARRIOR/

Jaap Karan Singh

jaap@scw.io

Co-Founder & Chief Singh, Secure Code Warrior