

Keto AppSec: It's All About the FATS

David Lindner, Director, Application Security

March 27, 2020

© 2020 CONFIDENTIAL



WHO IS THIS GUY?

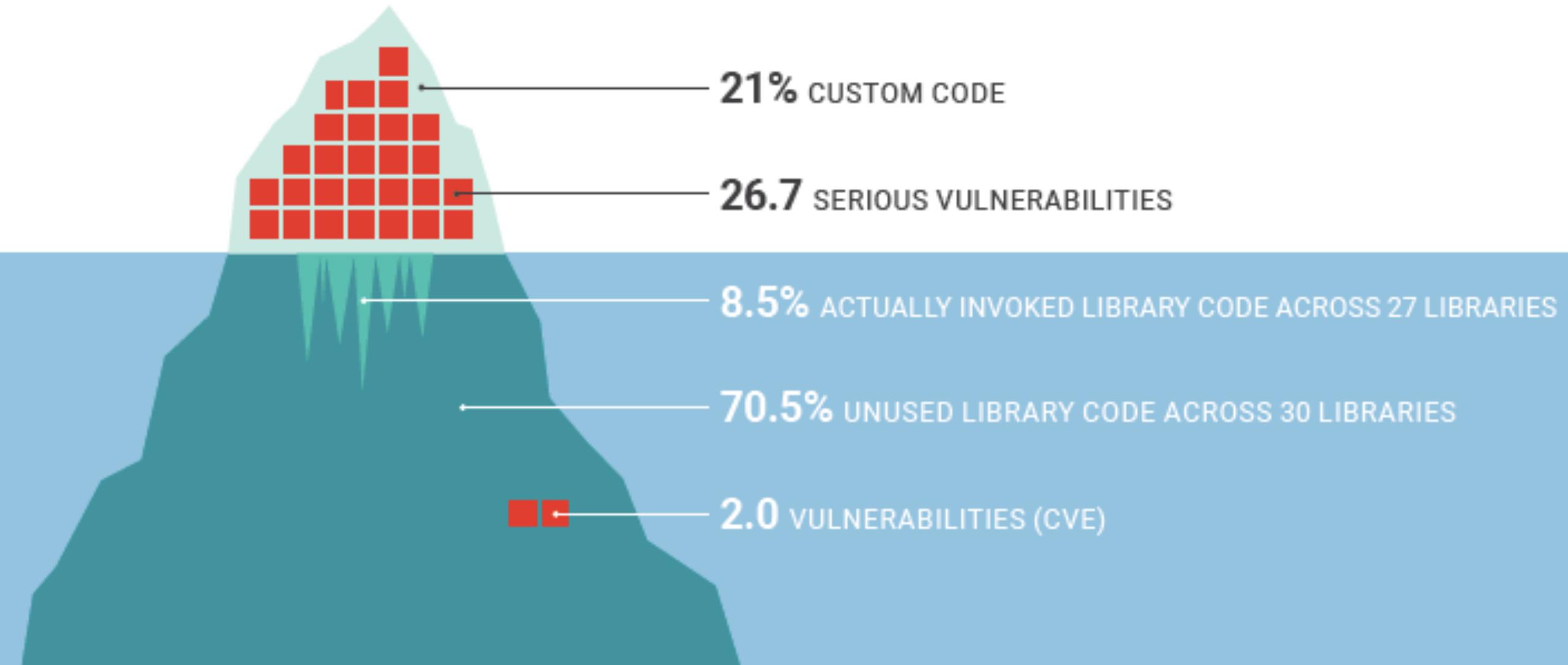
```
class Speaker {  
    let name = "David Lindner"  
    let title = "Director, Application Security"  
    let company = "Contrast Security"  
    let twitter = "@golfhackerdave"  
    var hobbies = ["Dadding", "Golfing", "IoT/Mobile",  
    "Fishing", "Hawkeyes"]  
}
```



Not so breaking news

Software Applications are vulnerable...and they are being attacked.

The Average Application



Software Security Crisis

A system message window with a blue header bar containing the text "System message" and a red close button with a white "X". The main content area displays the text "Average serious vulnerabilities:" followed by a large, bold number "26.7" in black. In the bottom right corner of the content area, there is a rectangular button with the word "OK" in black text.

2000

Average serious
vulnerabilities:

26.7

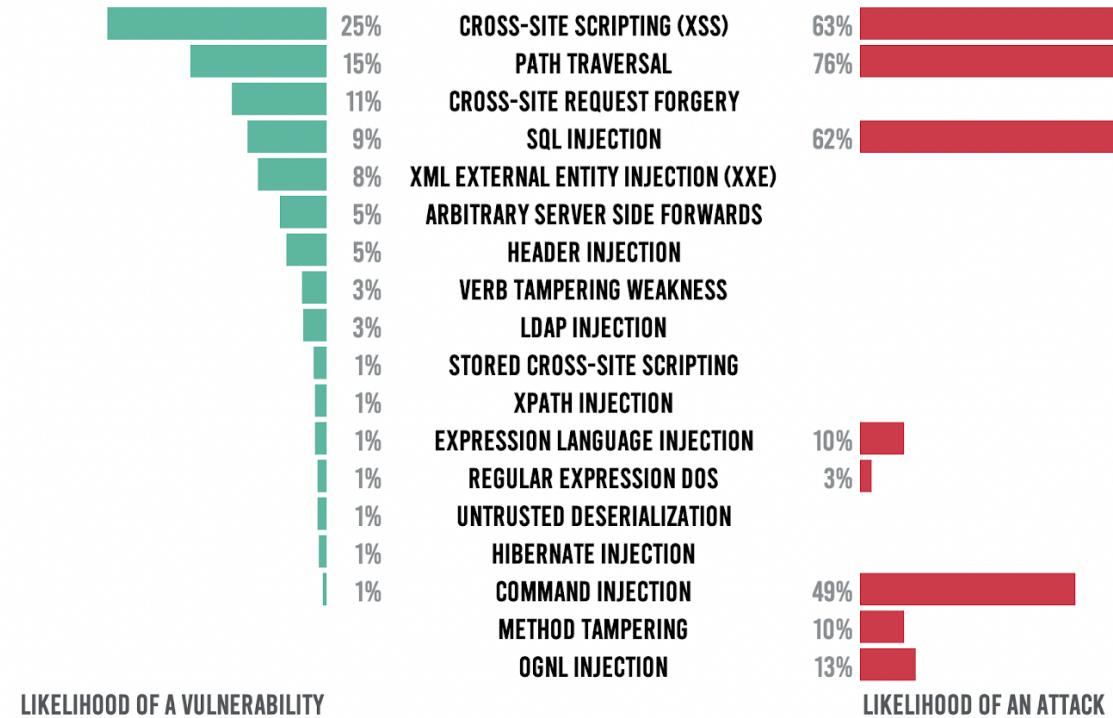
OK

2019

Formal Methods	SW CMM	Vulnerability Disclosure	Common Criteria	Dynamic Scanning	Compliance	Developer Training	DevOps	Shift Left								
1970	1985	1991	1997	1998	1999	1998	2001	2002	2002	2002	2003	2005	2009	2013	2016	2017
TCSEC DOD 5200.28		SSE-CMM		Penetration Testing		OWASP T10		Static Analysis		WAF		BSIMM OpenSAMM		DevSecOps		

Likelihood your application/api gets attacked

Jan/Feb 2020



Hackers are Exploiting the Lag... How Fast Can you Respond?

March 7
CVE-2017-5638
Disclosed, Apache
releases fixed version

Mid-May
Equifax
breach
occurs

July 29
Equifax
learns of
breach

Sept 7
Equifax discloses,
Four more Struts2
CVEs disclosed

No Updates

No Detection

Disaster

March 8
Widespread attack
probes observed

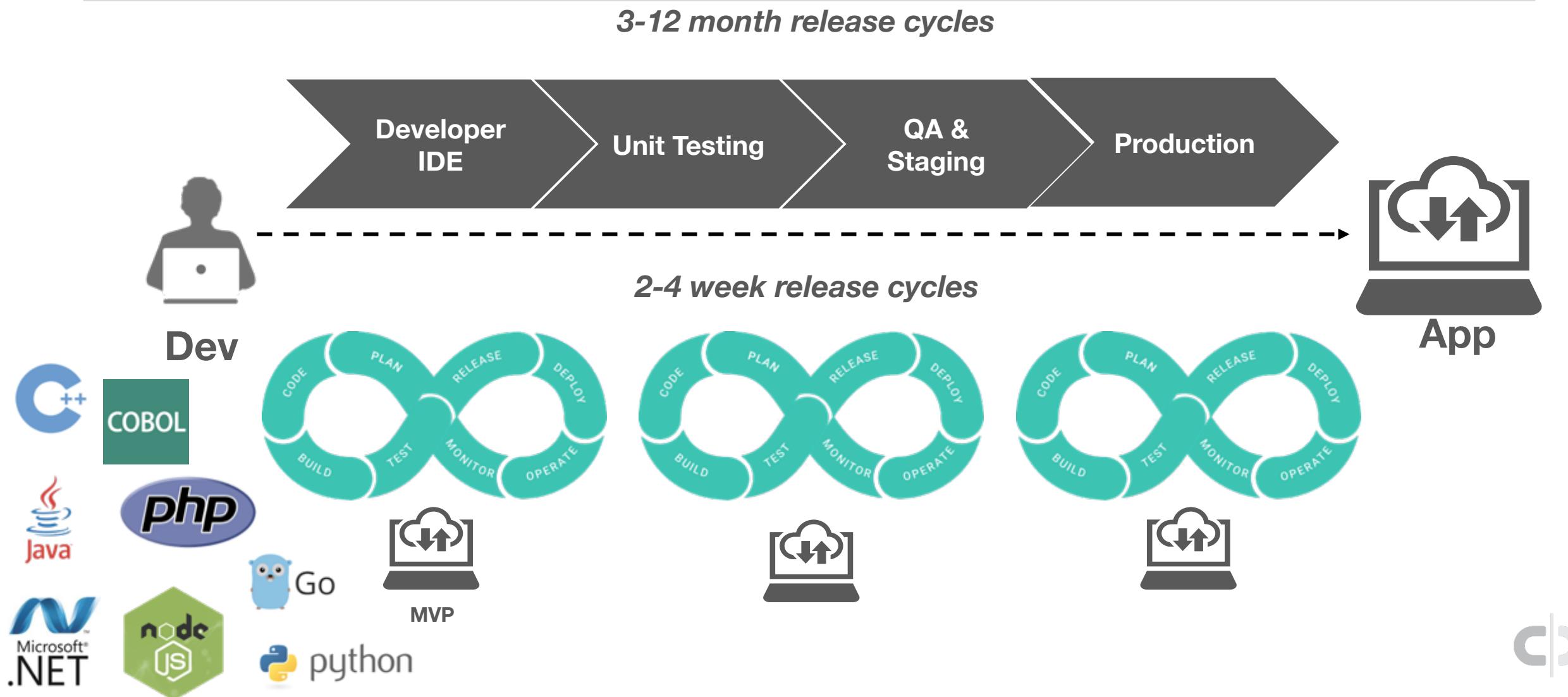


A dynamic photograph of a runner in mid-stride on a track. The runner is wearing a white top and dark shorts, with blue and yellow running shoes. The background is blurred, suggesting speed, and shows a stadium with spectators.

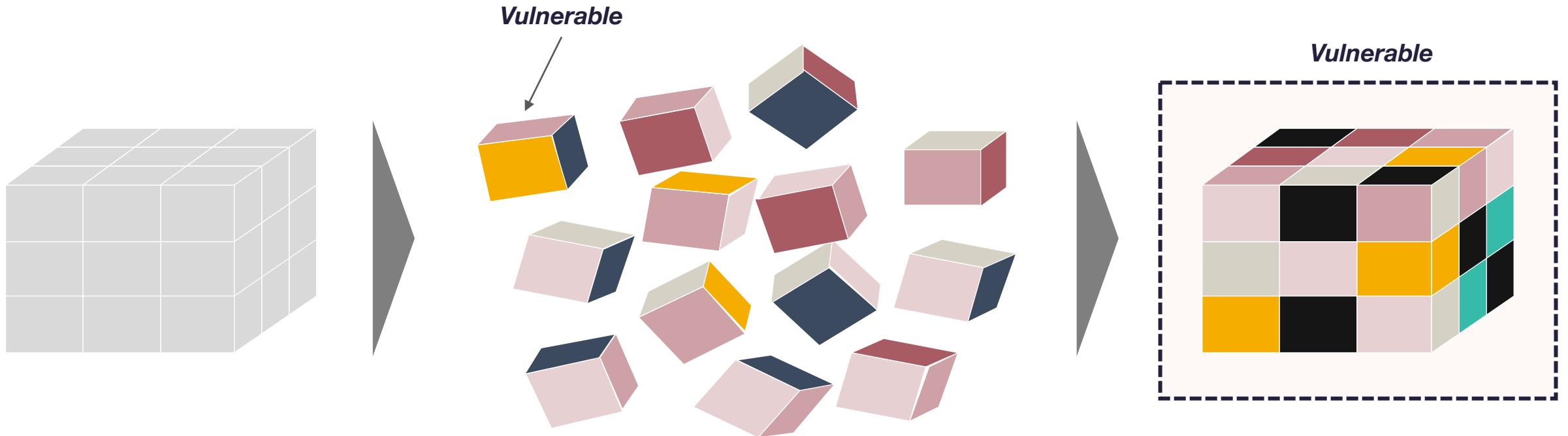
Software development continues to accelerate...and leverage new approaches.

Not so breaking news #2

Fundamental change on the path to an application



The (other) reality of software today



Vulnerable components = exposed software = higher (inherited) risk



Software
Approaches
Change
Rapidly

Agile; DevOps

Microservices;
APIs

Containers;
PaaS; Cloud
Native

CI/CD

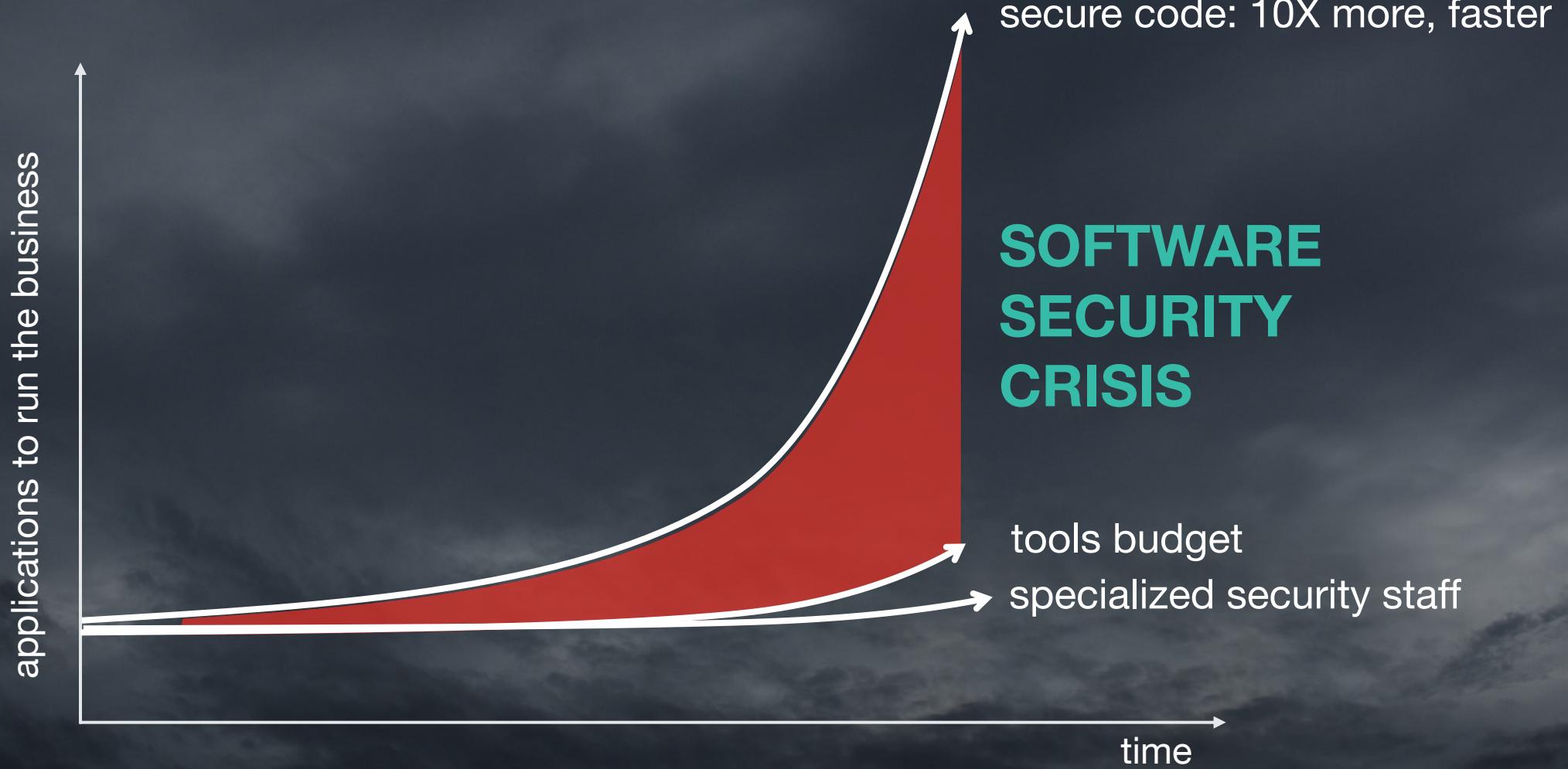


Out of 122 BSIMM 10 organizations



- 1.37% software security group to developers
- That's 1 software security professional to every 73 developers

IMPOSSIBLE ECONOMICS: SOFTWARE SECURITY



A photograph of a waterfall in a lush, green landscape. The waterfall flows down a steep, rocky cliff covered in vibrant green moss and vegetation. A bright rainbow arches across the scene from the bottom left towards the center. The water is white and turbulent as it falls.

The OLD way

The OLD way



I have 1500 applications



THEY ALL MUST BE
TESTED

Manual tests
SAST/DAST



Stop development
while we find
problems

Where could
we go wrong
with this?



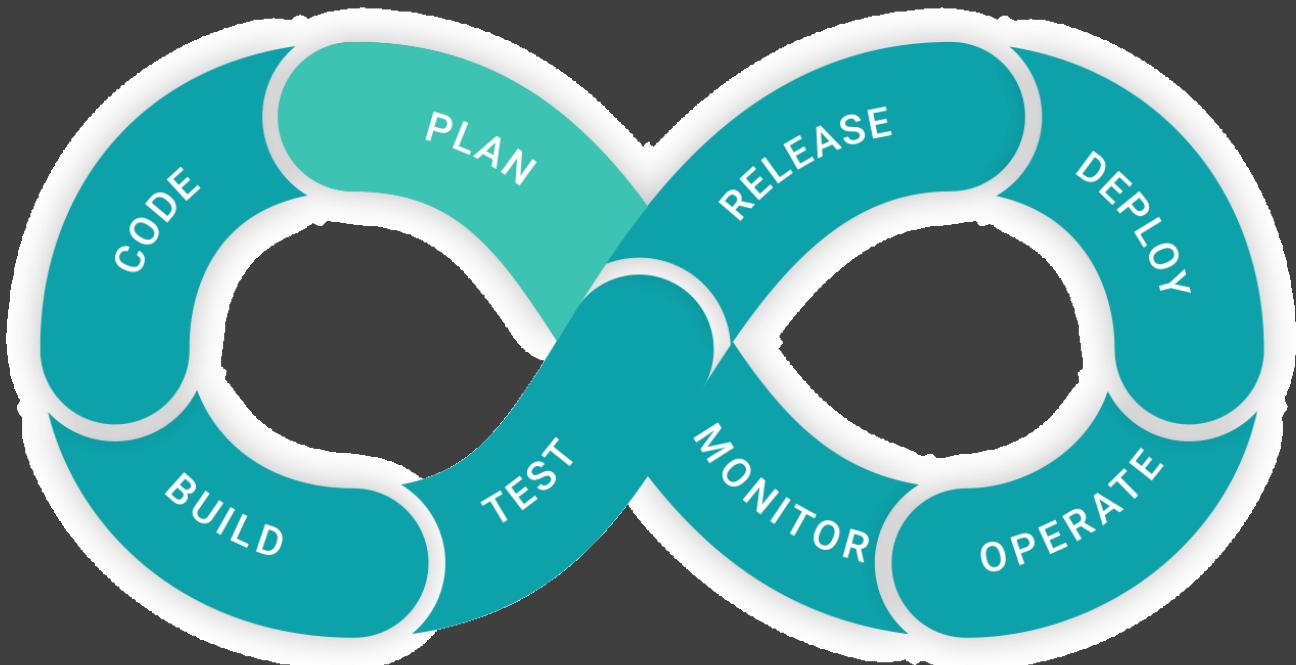


The OLD way

- Security is seen as a blocker
- Everything is fought
- Reports go “missing”
 - Security debt increases
- Security posture is unknown except when report was created



The OLD way





Result of the OLD way - Gating



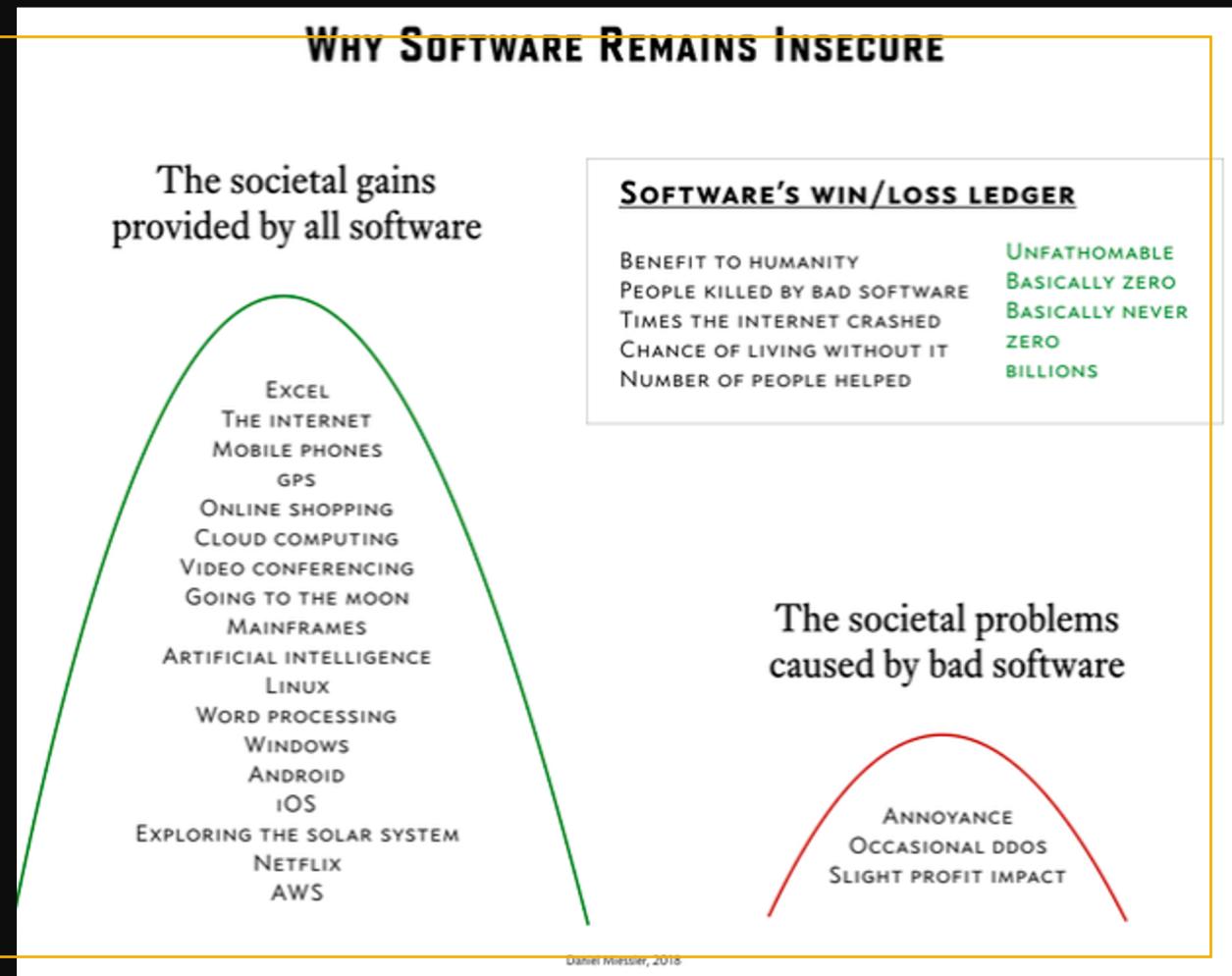
Security has the exact problems development had

- **Problem:** **software** is poor quality, late, slow, and doesn't provide business value.
- **Proven Approach:** DevOps
- **Result:**
 - 5x lower change failure rate
 - 96x faster MTTR service
 - 2x likely to exceed business goal
- **Problem:** **security** is poor quality, late, slow, and doesn't provide business value.
- **Possible Approach:** DevSecOps
- **Desired Result:**
 - 10x increase in portfolio coverage?
 - 80% reduction in vulns to prod?
 - 0x increase in time to market



The existence of insecure software has so far helped society far more than it has harmed it.

- Daniel Miessler
- <https://danielmiessler.com/blog/the-reason-software-remains-insecure/>





How can we deliver 10X
more secure code that
protects the integrity of
the business?



How can we focus and do things securely faster?



FATS

“F is for Frameworks”



What do frameworks do for us?

- REACT and Angular
 - XSS Protections by default
- Node
 - helmet
 - csurf
- Ruby on Rails
 - protect_from_forgery with: :exception
 - Mass assignment is pretty much fixed
 - devise or authlogic password storage
- Go
 - gorilla/csrf
 - gorilla/securecookie
 - crypto.random
 - html/template
- PHP Laravel
 - Salted/hased passwords with bcrypt
 - Prepared Statements
 - Mass Assignment protections



What does it mean?



- Know your frameworks
- Customize your testing
- Focus on the things that you aren't protected from



FATS

“A is for Automation”



LEGACY TOOL QUAGMIRE

Disparate, static, disconnected, inaccurate; requires an army of specialists to interpret results



IDE
Spellcheckers



SCA



Manual
Code
Review



SAST
Full Scan



Manual
Pentesting



NGWAF



DAST



SAST
Quick Scan



Fuzzing



WAF



IPS

Modern software tools work differently

Collaborative, real-time, full lifecycle, integrated, and social

YOUR SOFTWARE PROCESS AND PIPELINE

BUILD

GitHub



QUALITY



PERFORMANCE



APPDYNAMICS

COLLABORATION

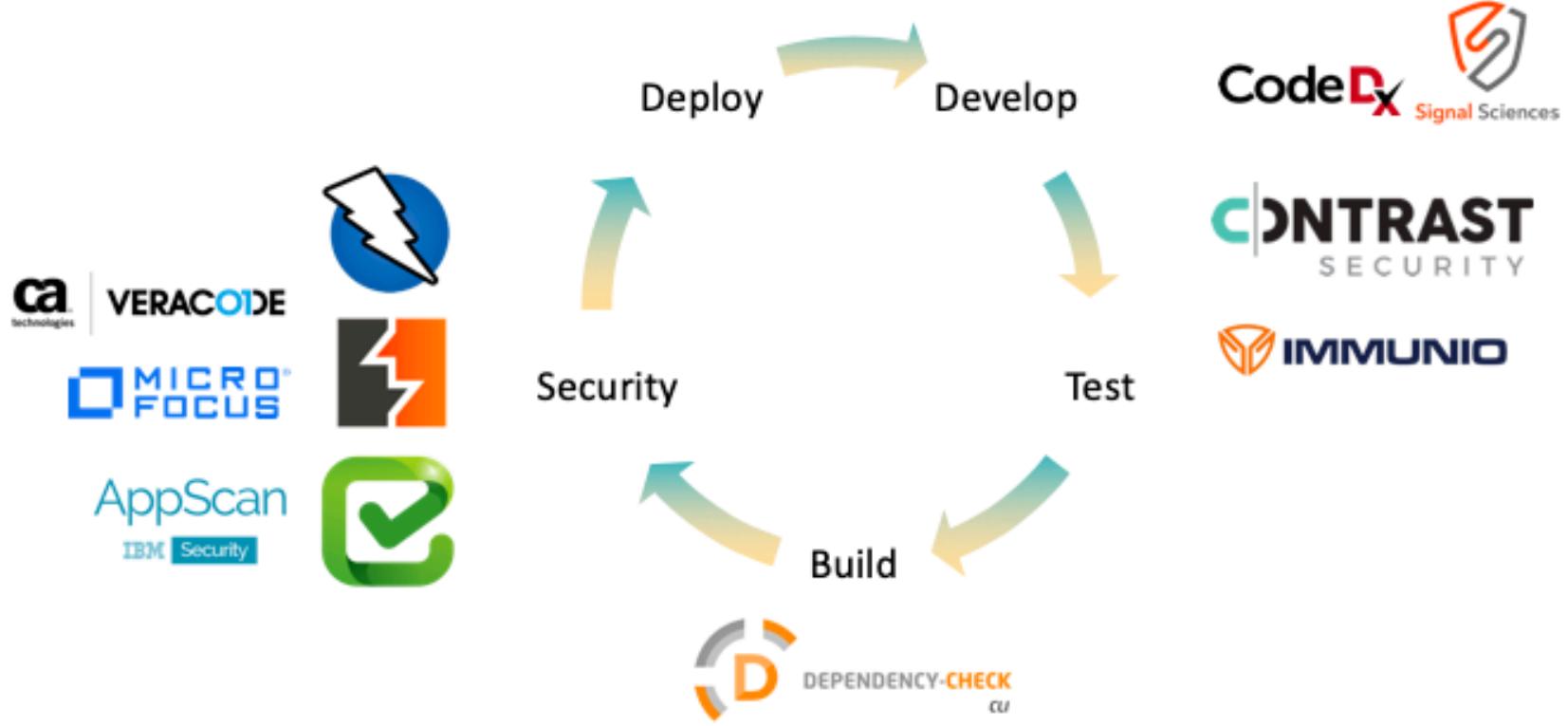


ANALYTICS



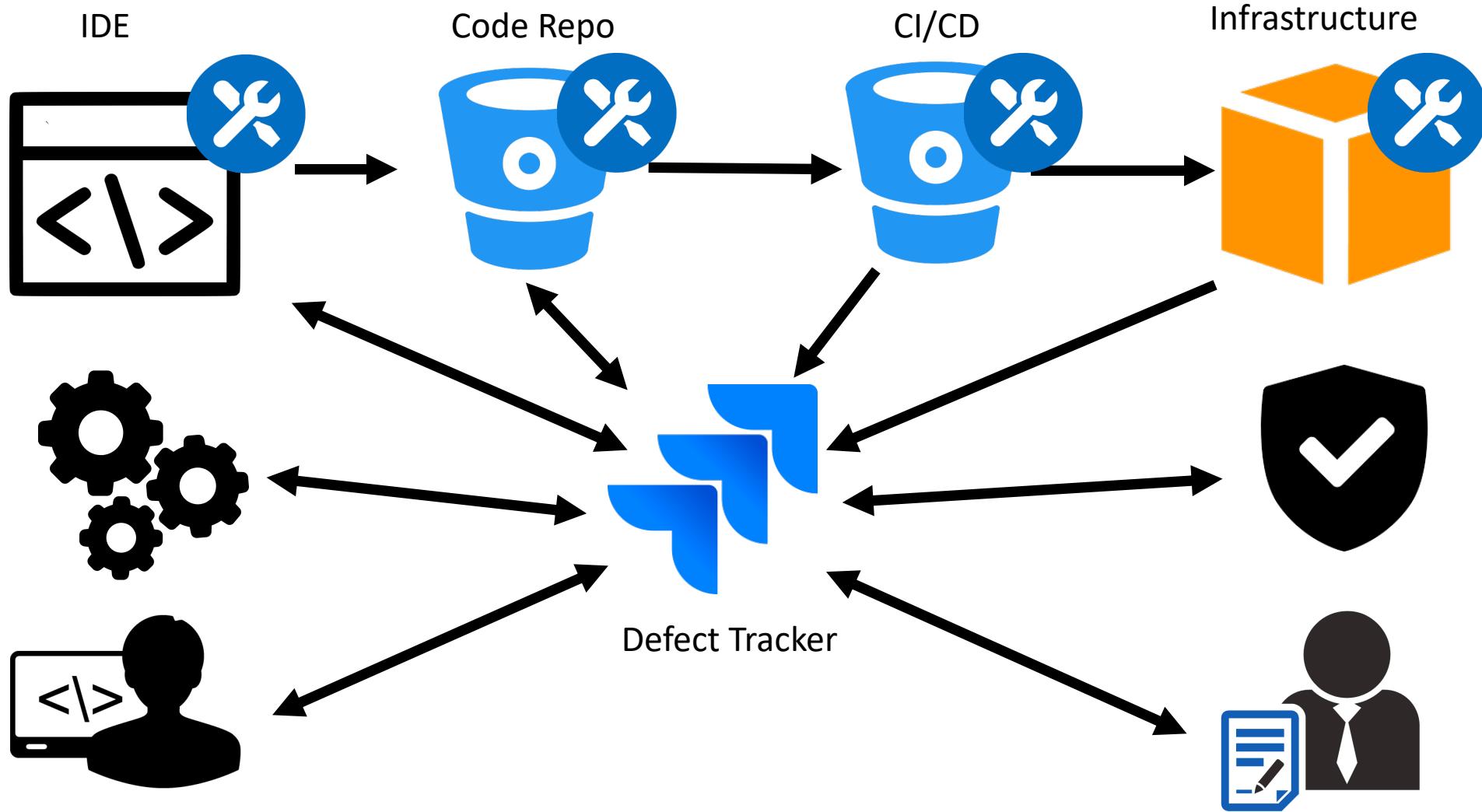
SOFTWARE
SECURITY

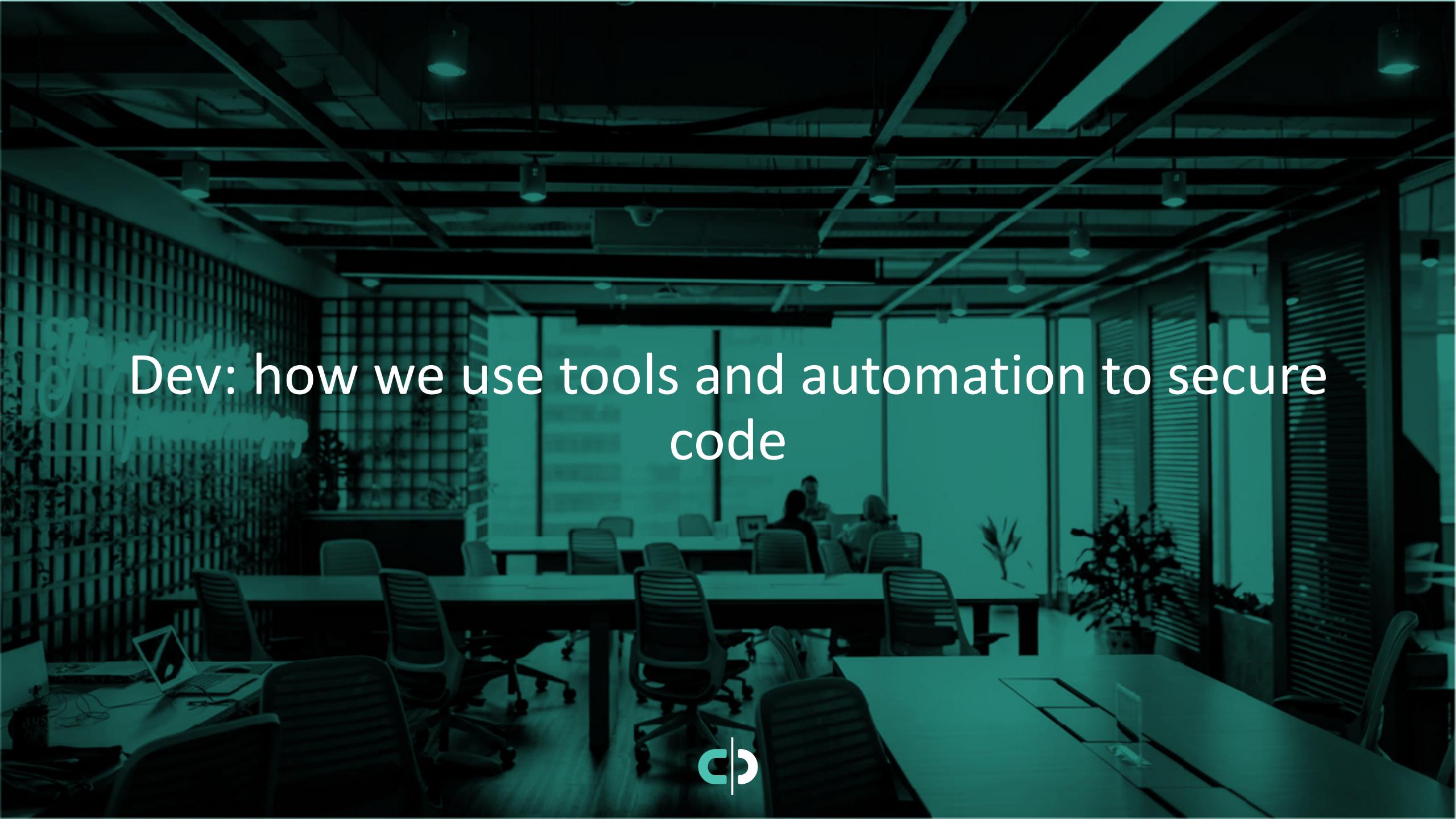




Lots of security tooling

Need to Streamline





Dev: how we use tools and automation to secure
code



Our dev setup

PHILOSOPHY

- Automated from the Beginning, Keep it Small, Boring Releases

TOOLING

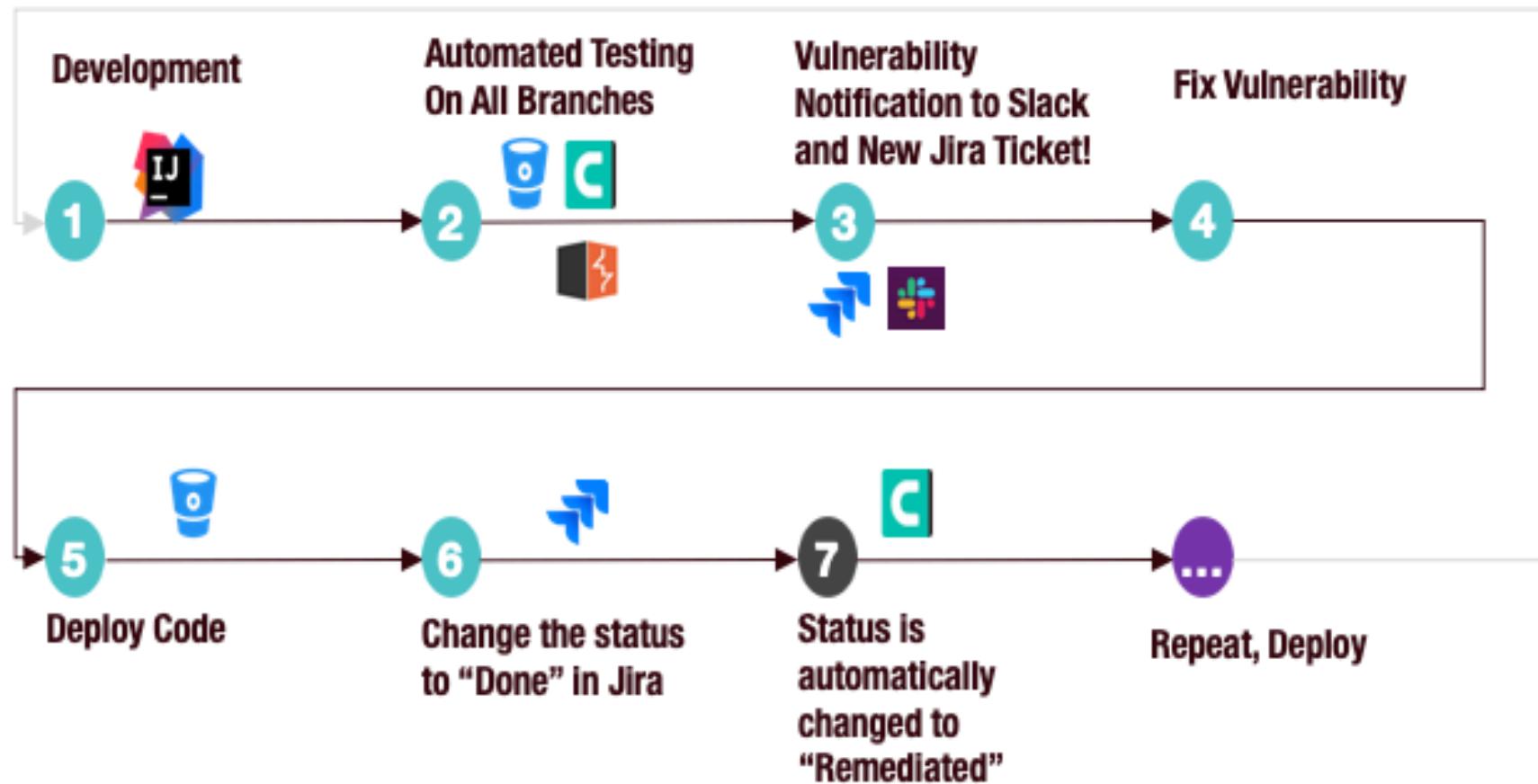
- IDE: IntelliJ
- CI/CD/SCM: Bitbucket Pipelines
- Artifact Repository: Artifactory
- ChatOps: Slack
- Automated Scans: Burp Enterprise (COMING SOON)
- Bug tracking: JIRA
- AppSec: Contrast Assess and Protect

SETUP

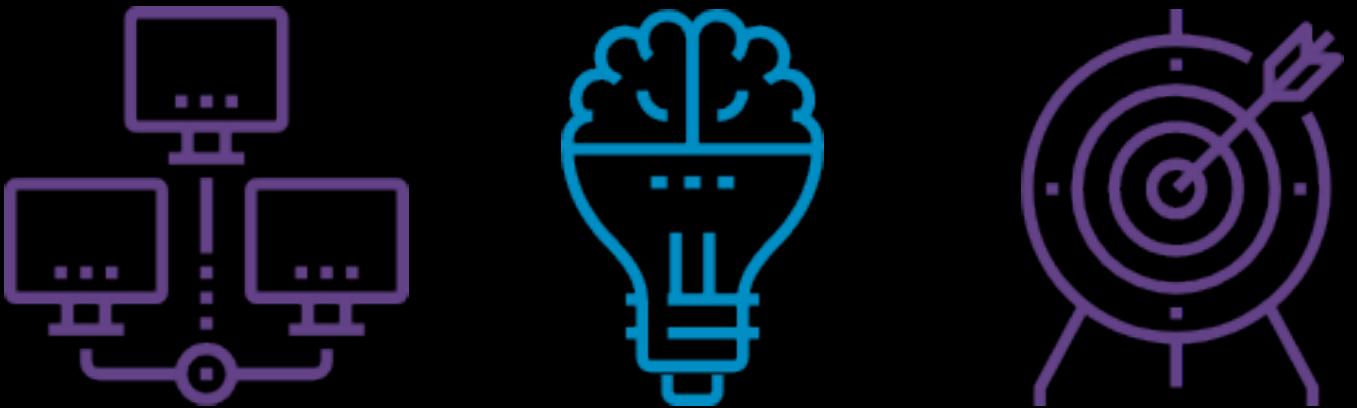
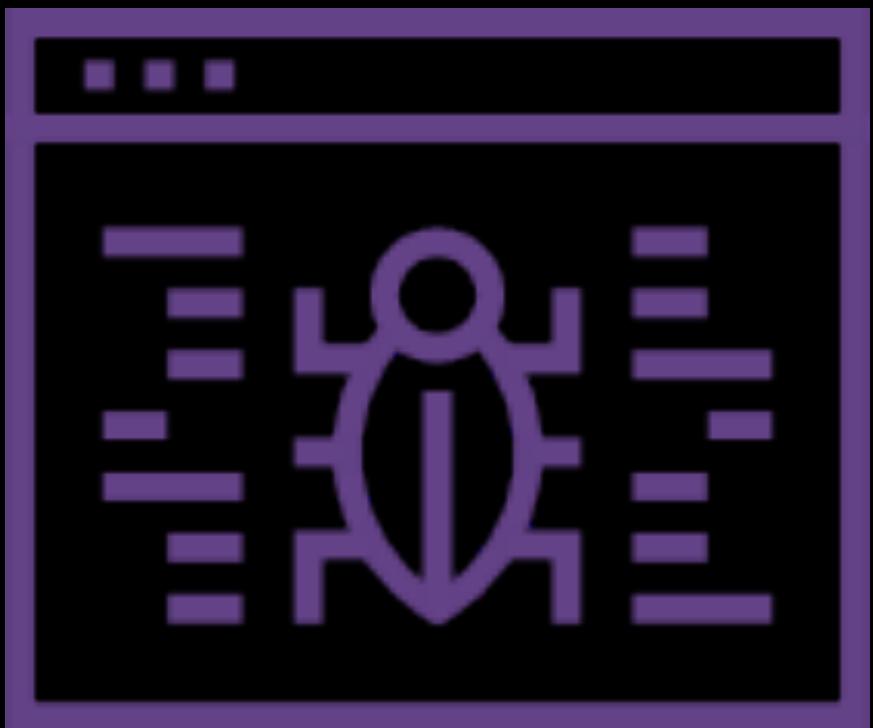
- Run pipeline on all branches; Maven plugin to configure and include Java Agent



Our dev workflow



What are we trying to achieve



- Identify vulnerabilities at DevOps speed
- Accurate results
- Simplified and integrated AppSec
- Continuous coverage
- Automation
- Application intelligence
- Smart response and 0-day protection
- Security anywhere





What if you could kill entire vulnerability classes with automation?

- XSS
- SQL/NoSQL Injection
- Command Injection
- CSRF
- EL/OGNL Injection
- Untrusted Deserialization
- XXE
- Padding Oracle

Empower Development to Weave Security In

- Expanding Security beyond the InfoSec/AppSec team
 - Security Champions within Development, Architecture & Leadership
- Keys to Success:
 - Self-selection
 - Visibility & Praise
 - Incentives





FATS

“T is for Threats”



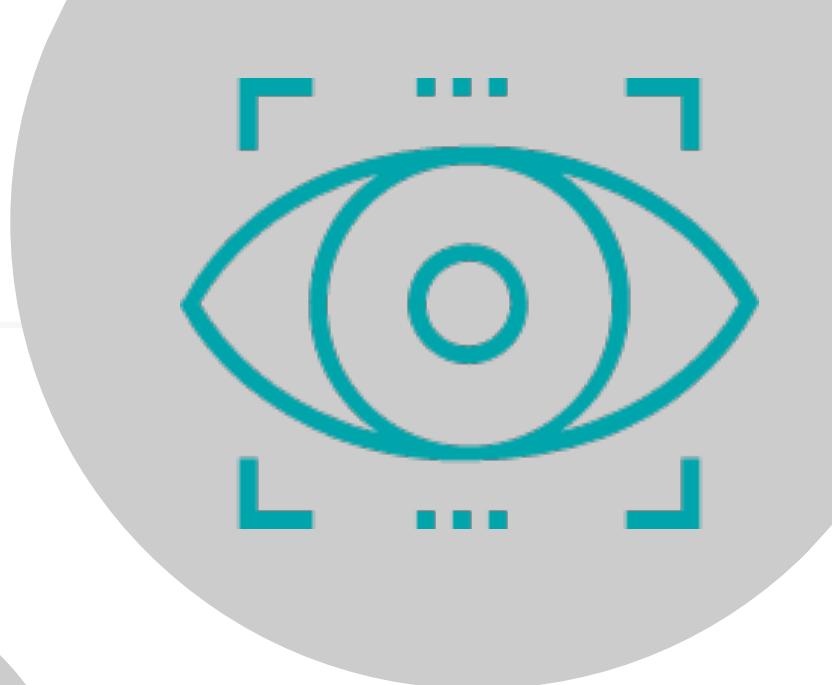
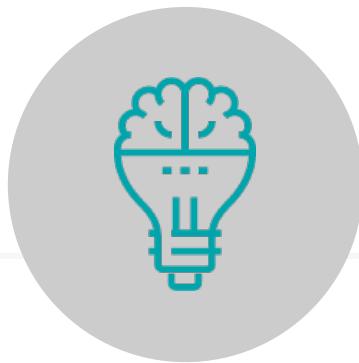
Managing Risks/Threats/Attacks

- Risk rating
- ASVS – helps with rigor
- CVE and issues monitoring
- Threat Intelligence



Software Composition Analysis (SCA)

- Fully automated solution
- Runtime assessment and protection
- Continuous visibility
- Self updating risk intelligence
- Single solution for all your code





Manage REAL Threats

- FUD no more
- Threat Intelligence
 - WAF or RASP data
 - Intelligence tools?
 - Other indicators
- Predictive Analysis
- For instance XSS

What's not FUD?

SQL Injection

- https://en.wikipedia.org/wiki/SQL_injection#Examples

Untrusted Deserialization

- WebLogic RCE - <https://nvd.nist.gov/vuln/detail/CVE-2019-2725>
- Struts 2 - <https://nvd.nist.gov/vuln/detail/CVE-2017-9805>
 - Equifax

OGNL Injection

- Struts 2 - <https://nvd.nist.gov/vuln/detail/CVE-2018-11776>



XSS - .09% CVSS 5 and up

Security Vulnerabilities Related To CWE-79

Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

[Results](#) [Download Results](#)

CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Notes
CVE-2019-1020019	79		XSS	2019-07-29	2019-07-31	4.3	None	Remote	Medium	Not required	Info
			gio-previewer before 1.0.0a12 allows XSS.								
CVE-2019-1020010	79		XSS	2019-07-29	2019-09-05	4.3	None	Remote	Medium	Not required	Info
			key before 10.102.4 allows hijacking a user's token.								
CVE-2019-1020008	79		XSS	2019-07-29	2019-07-31	4.3	None	Remote	Medium	Not required	Info
			ctable.js before 1.0.4 allows XSS.								
CVE-2019-1020007	79		XSS	2019-07-29	2019-07-30	3.5	None	Remote	Medium	Single system	Info
			endency-Track before 3.5.1 allows XSS.								
CVE-2019-1020005	79		XSS	2019-07-29	2019-08-01	3.5	None	Remote	Medium	Single system	Info
			nio-communities before 1.0.0a20 allows XSS.								
CVE-2019-1020003	79		XSS	2019-07-29	2019-08-01	3.5	None	Remote	Medium	Single system	Info
			nio-records before 1.2.2 allows XSS.								
CVE-2019-1010314	79		XSS	2019-07-11	2019-07-12	4.3	None	Remote	Medium	Not required	Info
			3 1.7.2, 1.7.3 is affected by: Cross Site Scripting (XSS). The impact is: execute JavaScript in victim's browser, when the vulnerable repo page is loaded. The component is: repository's descr								
CVE-2019-1010307	79		XSS	2019-07-15	2019-07-18	3.5	None	Remote	Medium	Single system	Info
			GLPI Product 9.3.1 is affected by: Cross Site Scripting (XSS). The impact is: All dropdown values are vulnerable to XSS leading to privilege escalation and executing js on admin. The compo								
CVE-2019-1010287	79		Exec Code XSS	2019-07-17	2019-07-22	4.3	None	Remote	Medium	Not required	Info
			sheet Next Gen 1.5.3 and earlier is affected by: Cross Site Scripting (XSS). The impact is: Allows an attacker to execute arbitrary HTML and JavaScript code via a "redirect" parameter. The c								
CVE-2019-1010261	79		XSS	2019-07-18	2019-07-19	4.3	None	Remote	Medium	Not required	Info
			3 1.7.0 and earlier is affected by: Cross Site Scripting (XSS). The impact is: Attacker is able to have victim execute arbitrary JS in browser. The component is: go-get URL generation - PR to								
CVE-2019-1010247	79		XSS	2019-07-19	2019-08-23	4.3	None	Remote	Medium	Not required	Info
			rtZone IAM mod_auth_openidc 2.3.10.1 and earlier is affected by: Cross Site Scripting (XSS). The impact is: Redirecting the user to a phishing page or interacting with the application on bel								
CVE-2019-1010237	79		Exec Code XSS	2019-07-22	2019-10-09	4.3	None	Remote	Medium	Not required	Info
			5.3 before 5.3.12; 5.2 before 5.2.21 is affected by: Cross Site Scripting (XSS) - CWE-79 Type 2: Stored XSS (or Persistent). The impact is: Execute code in the victim's browser. The compo								
CVE-2019-1010235	79		XSS	2019-07-22	2019-07-23	3.5	None	Remote	Medium	Single system	Info
			CMS 1.1 is affected by: Cross Site Scripting (XSS). The impact is: Cookie stealing, Alert pop-up on page, Redirecting to another phishing site, Executing browser exploits. The component is:								
CVE-2019-1010207	79		XSS	2019-07-23	2019-07-29	4.3	None	Remote	Medium	Not required	Info
			technosolutions Pie Register 3.0.15 is affected by: Cross Site Scripting (XSS). The impact is: Stealing of session cookies. The component is: File: Login. Parameters: interim-login, wp-lang, an								



SQL Injection - 97% CVSS 6 and up

Security Vulnerabilities Related To CWE-89

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By: CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-1010259	89	Sql		2019-07-18	2019-08-13	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
SaltStack Salt 2018.3, 2019.2 is affected by: SQL Injection. The impact is: An attacker could escalate privileges on MySQL server deployed by cloud provider. It leads to RCE. The component is: The mysql.user_chpass function from the MySQL module for Salt. The attack vector is: specially crafted password string. The fixed version is: 2018.3.4.														
2	CVE-2019-1010248	89	Sql		2019-07-18	2019-07-23	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Synetics GmbH I-dolt 1.12 and earlier is affected by: SQL Injection. The impact is: Unauthenticated mysql database access. The component is: Web login form. The attack vector is: An attacker can exploit the vulnerability by sending a malicious HTTP POST request. The fixed version is: 1.12.1.														
3	CVE-2019-1010201	89	Sql		2019-07-23	2019-07-24	4.0	None	Remote	Low	Single system	Partial	None	None
Jeesite 1.2.7 is affected by: SQL Injection. The impact is: sensitive information disclosure. The component is: updateProcInsIdByBusinessId() function in src/main/java/com.thinkgem.jeesite/modules/act/ActDao.java has SQL Injection vulnerability. The attack vector is: network connectivity,authenticated. The fixed version is: 4.0 and later.														
4	CVE-2019-1010191	89	Sql		2019-07-24	2019-07-29	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
marginalia < 1.6 is affected by: SQL Injection. The impact is: The impact is: an injection of any SQL queries when a user controller argument is added as a component. The component is: Affects users that add a component that is user controller, for instance a parameter or a header. The attack vector is: Hacker inputs a SQL to a vulnerable vector(header, http parameter, etc). The fixed version is: 1.6.														
5	CVE-2019-1010153	89	Sql		2019-07-23	2019-07-24	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
zzcms 8.3 and earlier is affected by: SQL Injection. The impact is: sql inject. The component is: zs/subzs.php.														
6	CVE-2019-1010148	89	Exec Code Sql		2019-07-23	2019-07-24	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
zzcms version 8.3 and earlier is affected by: SQL Injection. The impact is: zzcms File Delete to Code Execution.														
7	CVE-2019-1010104	89	Sql		2019-07-18	2019-07-23	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
TechyTalk Quick Chat WordPress Plugin All up to the latest is affected by: SQL Injection. The impact is: Access to the database. The component is: like_escape is used in Quick-chat.php line 399. The attack vector is: Crafted ajax request.														
8	CVE-2019-1010034	89	Sql		2019-07-15	2019-08-21	4.0	None	Remote	Low	Single system	Partial	None	None
Deepwoods Software WebLibrarian 3.5.2 and earlier is affected by: SQL Injection. The impact is: Exposing the entire database. The component is: Function "AllBarCodes" (defined at database_code.php line 1018) is vulnerable to a boolean-based blind sql injection. This function call can be triggered by any user logged-in with at least Volunteer role or manage_circulation capabilities. PoC : /wordpress/wp-admin/admin.php?page=weblib-circulation-desk&orderby=title&order=DESC.														
9	CVE-2019-17429	89	Sql		2019-10-10	2019-10-11	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Adhouma CMS through 2019-10-09 has SQL Injection via the post.php p_id parameter.														
10	CVE-2019-17419	89	Sql		2019-10-09	2019-10-10	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
An issue was discovered in MetInfo 7.0. There is SQL injection via the admin/?n=user&c=admin_user&a=doGetUserInfo id parameter.														
11	CVE-2019-17418	89	Sql		2019-10-09	2019-10-10	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
An issue was discovered in MetInfo 7.0. There is SQL injection via the admin/?n=language&c=language_general&a=doSearchParameter appno parameter, a different issue than CVE-2019-16997.														
12	CVE-2019-17319	89	Sql		2019-10-07	2019-10-09	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
SugarCRM before 8.0.4 and 9.x before 9.0.2 allows SQL injection in the Emails module by a Regular user.														
13	CVE-2019-17318	89	Sql		2019-10-07	2019-10-09	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
SugarCRM before 8.0.4 and 9.x before 9.0.2 allows SQL injection in the pmse_Inbox module by a Regular user.														
14	CVE-2019-17298	89	Sql		2019-10-07	2019-10-09	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
SugarCRM before 8.0.4 and 9.x before 9.0.2 allows SQL injection in the Administration module by a Developer user.														
15	CVE-2019-17297	89	Sql		2019-10-07	2019-10-09	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
SugarCRM before 8.0.4 and 9.x before 9.0.2 allows SQL injection in the Quotes module by a Regular user.														
16	CVE-2019-17296	89	Sql		2019-10-07	2019-10-09	6.5	None	Remote	Low	Single system	Partial	Partial	Partial

CMD Injection 95% CVSS 6 and up

Security Vulnerabilities Related To CWE-77

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Total number of vulnerabilities : **489** Page : [1](#) (This Page) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

[Copy Results](#) [Download Result](#)

Security Vulnerabilities Related To CWE-78

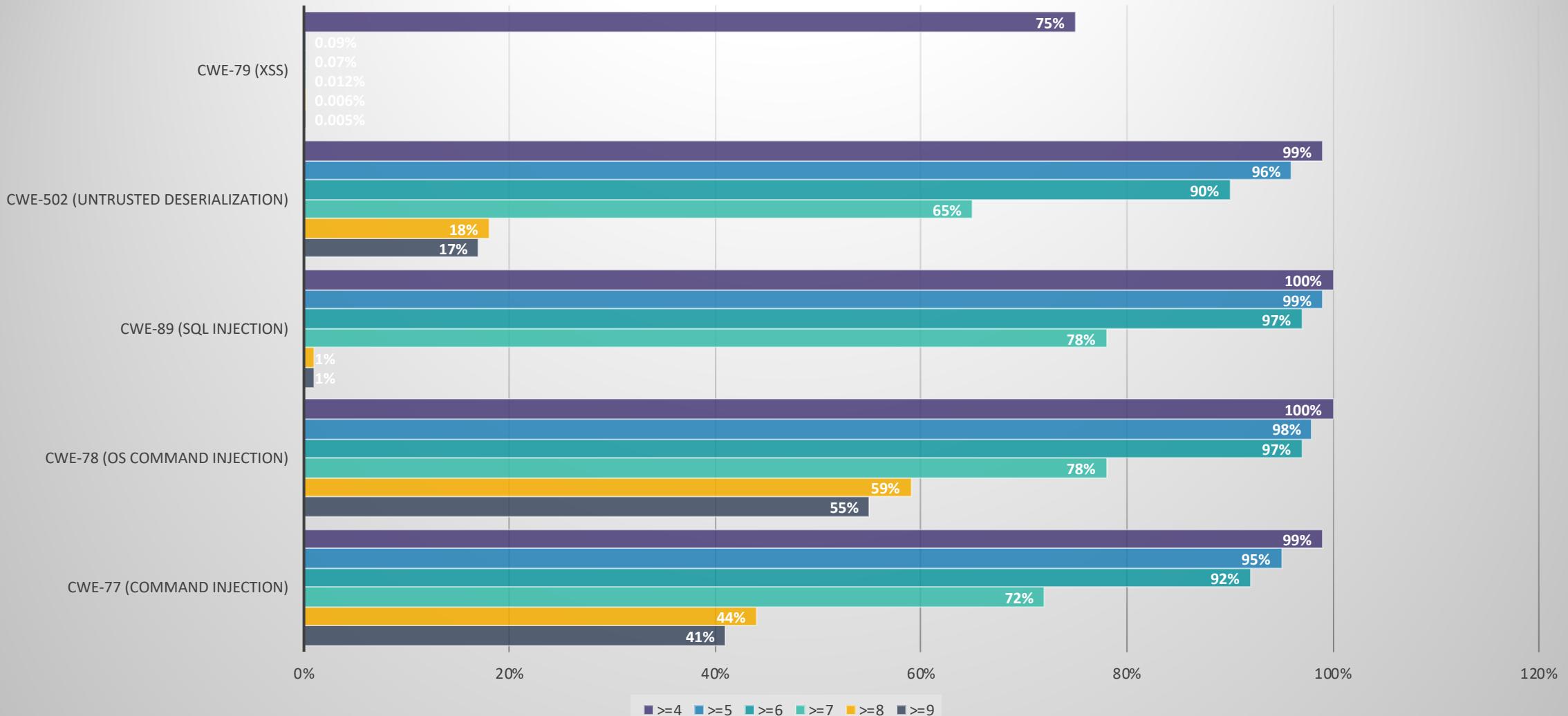
CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

Total number of vulnerabilities : 788 Page : 1 (This Page) 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

[Copy Results](#) [Download Results](#)



% of CVE by CVSS Score

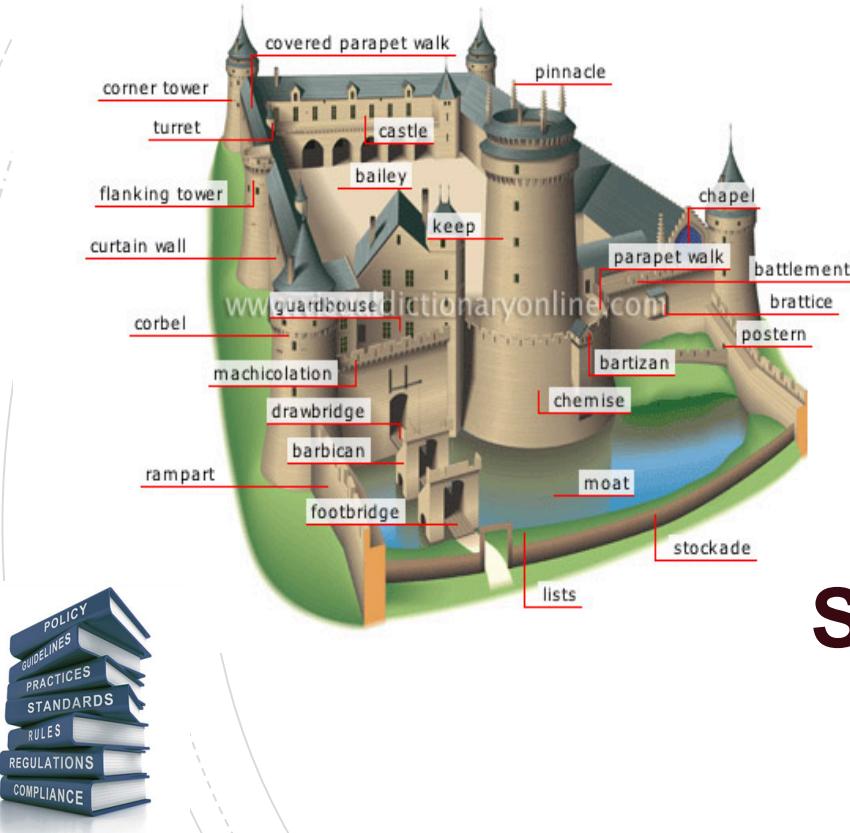




FATS
“S is for Speed and Sophistication”



Don't blame speed



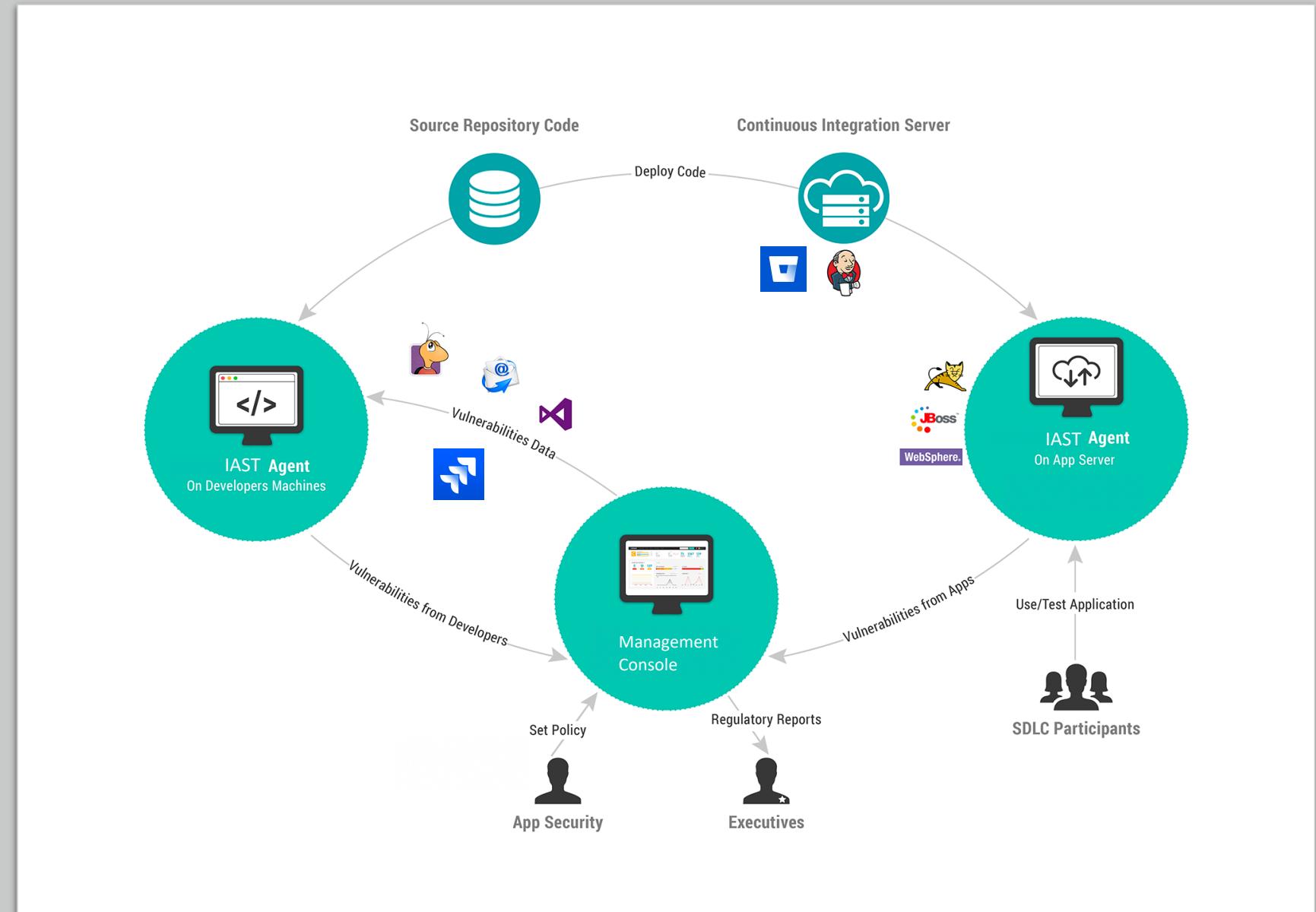
**SECURING FAST-CHANGING
THINGS IS DIFFERENT**



**DevOps speed is held back
by a 15-year-old, scan and perimeter-
based software security model**

Built for the pre-cloud era

Continuous Integration and Delivery



Time for Sophisticated Things

- Cyber talent shortage is real
- Frameworks won't solve everything
- Tools can't find everything, especially the extremely complex or custom security controls
- So focus on the hard stuff with your expensive cyber security assets



Examples of Sophisticated things

 Authorization

 Integrations

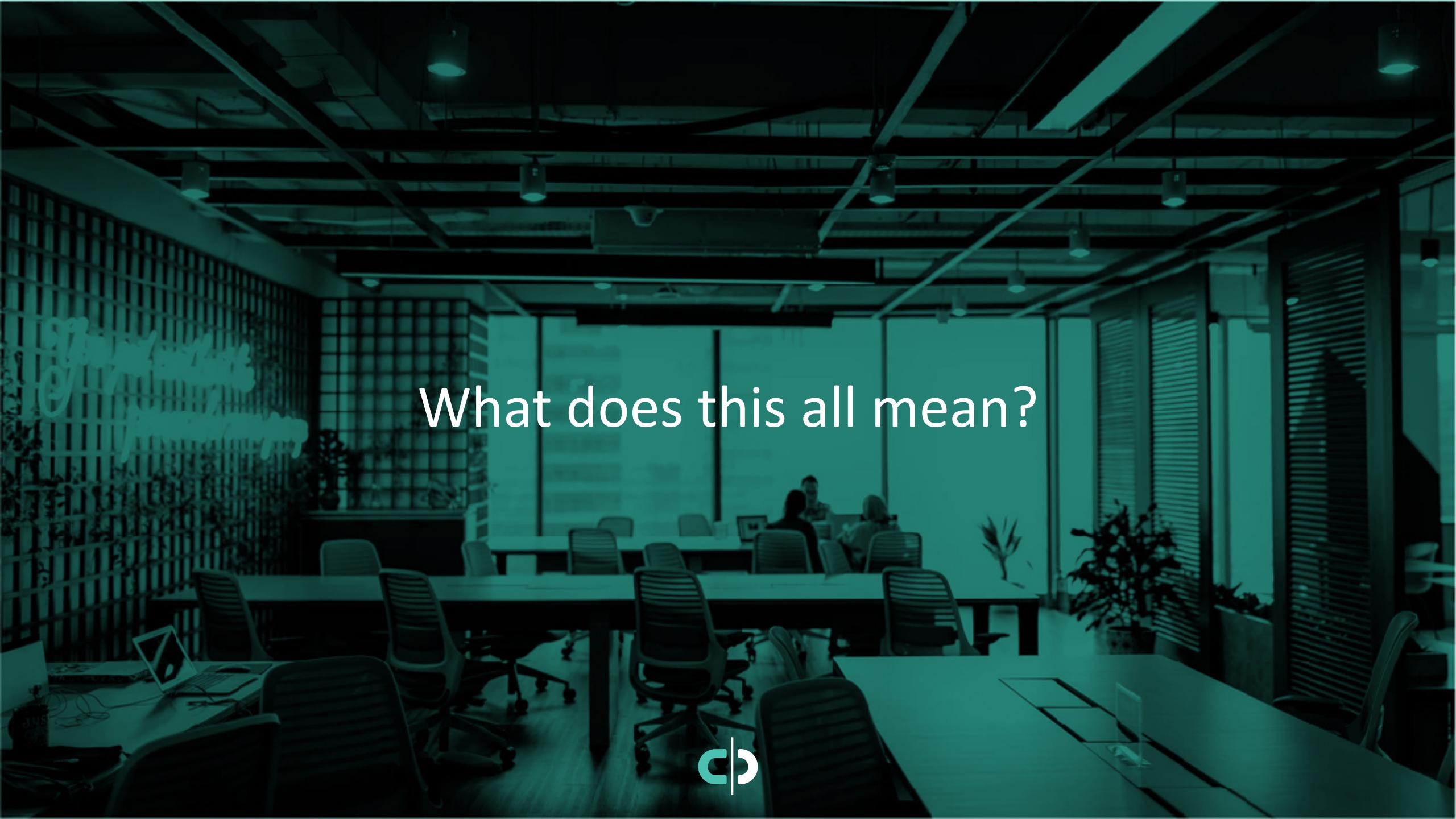
 Cloud Native / Micro Services / Serverless

 Threat modeling

 Secure Design

 When to train/when to do





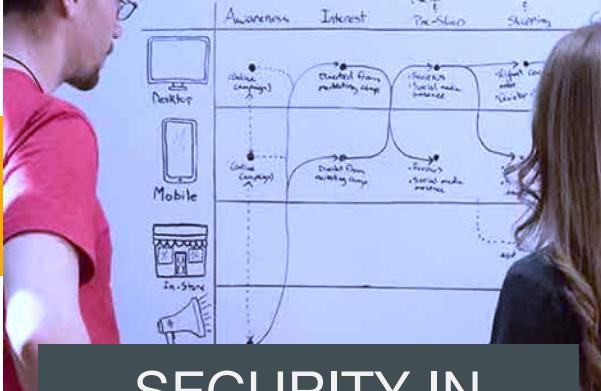
What does this all mean?



Shift EXTEND left, right, and everywhere!



SECURITY IN
DEVELOPMENT



SECURITY IN
INTEGRATION



SECURITY IN
OPERATIONS

EMPOWER

- Have some trust in frameworks
- Real time test 1st party and 3rd party code
- Realtime feedback through my tools
- Don't slow me down

ASSURE

- Don't slow down my builds
- Integrate with my testing tools
- Critical vulns break my build

PROTECT

- Tell me who is attacking and how
- Stop vulnerabilities from being exploited
- Don't create alert fatigue



APPSEC IN THE MODERN DEV WORLD

Evolve Tools to Secure Modern Software

Automated application security distributed across all software development and delivery pipelines; assess and protect microservices/APIs; native support for cloud-native apps

Enable Developer Self Sufficiency

Self-service application security integrated into the developer workflow; developers automatically find and fix vulnerabilities, without reliance on security experts

Automate Open Source Risk Management

Open source security and compliance controls automatically embedded in CI/CD; teams stay on top of risk introduced by use of open source; always on monitoring and protection

Accelerate Digital Transformation; Protect Legacy Portfolio

A single solution to secure on-premises, cloud and hybrid apps at scale; drive cloud adoption and app modernization, while defending your legacy application portfolio

Optimize Penetration Testing

Strategically focus investment in manual penetration testing on complex security weaknesses; Increase fidelity and action-ability of results

Ensure Continuous Visibility

Real-time visibility into security posture across the enterprise; continuous monitoring and intelligence across the SDLC; streamlined compliance



A landscape photograph of mountains at sunset or sunrise. The sky is filled with dramatic, colorful clouds in shades of orange, yellow, and purple. Sunbeams are visible through the clouds. In the foreground, there's a silhouette of a city skyline. A large, stylized black outline of a hand making a 'shaka' or 'hang loose' gesture is positioned on the right side of the frame.

Thanks! Ask me anything!

David Lindner - @golfhackerdave