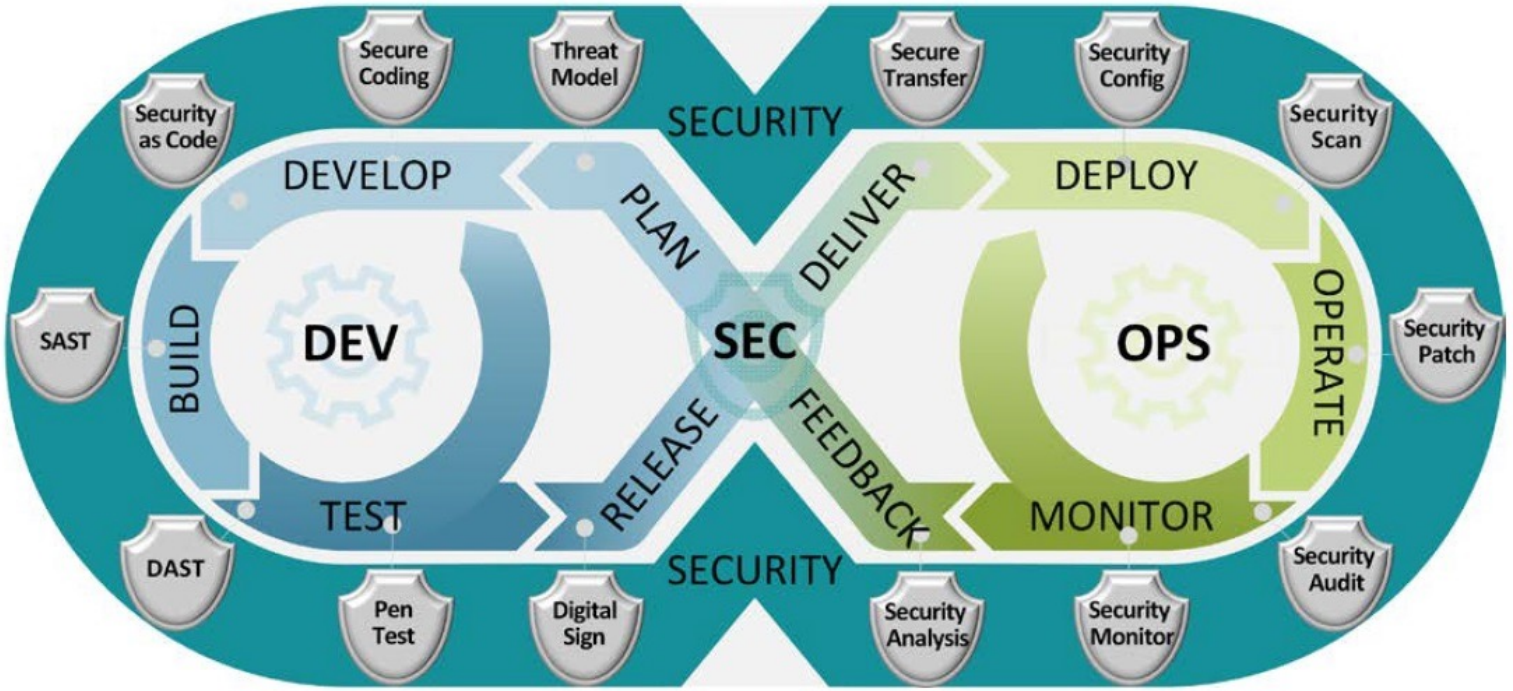
HCLSoftware
Federal

Turning Security from Foe to Friend

Rob Cuddy
HCLSoftware Federal
Rob.Cuddy@hclfederal.com

August 2024

We Know Security Is A Necessity



Source: https://public.cyber.mil/devsecops/

So If We Are Doing It, Then How Does This Keep Happening?

CNBC

Search quotes, news & videos | WATCHLIST | SIGN IN | CREATE FREE ACCOUNT

MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV INVESTING CLUB PRO MAKE IT SELECT USA · INTL WATCH LIVE

AIRLINES

Delta CEO says CrowdStrike-Microsoft outage cost the airline $500 million

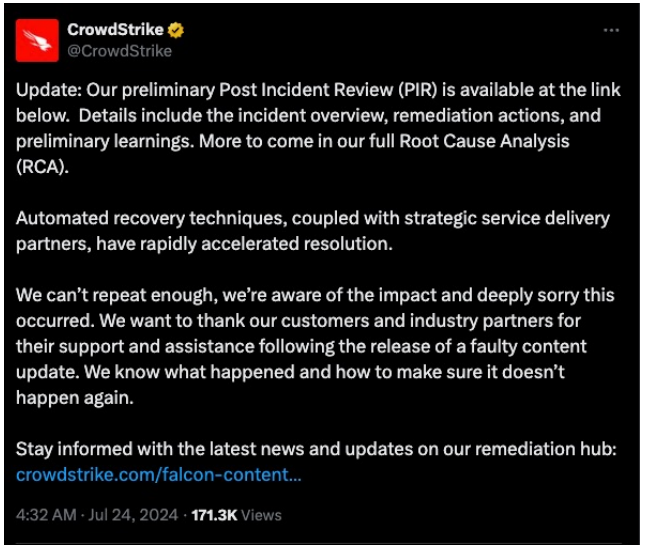
PUBLISHED WED, JUL 31 2024·7:19 AM EDT | UPDATED WED, JUL 31 2024·11:17 AM EDT

Leslie Josephs @LESLIEJOSEPHS | Ece Yildirim @IN/ECEDYILDIRIM/

SHARE

KEY POINTS

- Delta CEO Ed Bastian said the massive IT outage earlier this month that stranded thousands of customers will cost it $500 million.
- The airline canceled more than 5,000 flights in the wake of the outage, which was caused by a botched CrowdStrike software update and took thousands of Microsoft systems around the world offline.
- Bastian said the carrier would seek damages from the disruptions, adding, "We have no choice."



CrowdStrike
@CrowdStrike

Update: Our preliminary Post Incident Review (PIR) is available at the link below. Details include the incident overview, remediation actions, and preliminary learnings. More to come in our full Root Cause Analysis (RCA).

Automated recovery techniques, coupled with strategic service delivery partners, have rapidly accelerated resolution.

We can't repeat enough, we're aware of the impact and deeply sorry this occurred. We want to thank our customers and industry partners for their support and assistance following the release of a faulty content update. We know what happened and how to make sure it doesn't happen again.

Stay informed with the latest news and updates on our remediation hub: crowdstrike.com/falcon-content...

4:32 AM · Jul 24, 2024 · **171.3K** Views

Source: https://www.cnbc.com/2024/07/31/delta-ceo-crowdstrike-microsoft-outage-cost-the-airline-500-million.html
Source: https://x.com/CrowdStrike/status/1816073893645287481

From Public Crowdstrike Blog Posting & Post Incident Review Document:

On Friday, July 19, 2024 at 04:09 UTC, as part of regular operations, CrowdStrike released a content configuration update for the Windows sensor to gather telemetry on possible novel threat techniques.

These updates are a regular part of the dynamic protection mechanisms of the Falcon platform. The problematic Rapid Response Content configuration update resulted in a Windows system crash.

…

Rapid Response Content

Rapid Response Content is used to perform a variety of behavioral pattern-matching operations on the sensor using a highly optimized engine. Rapid Response Content is a representation of fields and values, with associated filtering. This Rapid Response Content is stored in a proprietary binary file that contains configuration data. It is not code or a kernel driver.

Rapid Response Content is delivered as "Template Instances," which are instantiations of a given Template Type. Each Template Instance maps to specific behaviors for the sensor to observe, detect or prevent. Template Instances have a set of fields that can be configured to match the desired behavior.

In other words, Template Types represent a sensor capability that enables new telemetry and detection, and their runtime behavior is configured dynamically by the Template Instance (i.e., Rapid Response Content).

Rapid Response Content provides visibility and detections on the sensor without requiring sensor code changes. This capability is used by threat detection engineers to gather telemetry, identify indicators of adversary behavior and perform detections and preventions. Rapid Response Content is behavioral heuristics, separate and distinct from CrowdStrike's on-sensor AI prevention and detection capabilities.

Source: https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/

OK … So Why Didn't Their Testing Catch It??

Template Type Stress Testing: On March 05, 2024, a stress test of the IPC Template Type was executed in our staging environment, which consists of a variety of operating systems and workloads. The IPC Template Type passed the stress test and was validated for use.

Template Instance Release via Channel File 291: On March 05, 2024, following the successful stress test, an IPC Template Instance was released to production as part of a content configuration update. Subsequently, three additional IPC Template Instances were deployed between April 8, 2024 and April 24, 2024. These Template Instances performed as expected in production.

What Happened on July 19, 2024?
On July 19, 2024, two additional IPC Template Instances were deployed. Due to a bug in the Content Validator, one of the two Template Instances passed validation despite containing problematic content data.

Based on the testing performed before the initial deployment of the Template Type (on March 05, 2024), trust in the checks performed in the Content Validator, and previous successful IPC Template Instance deployments, these instances were deployed into production.

When received by the sensor and loaded into the Content Interpreter, problematic content in Channel File 291 resulted in an out-of-bounds memory read triggering an exception. This unexpected exception could not be gracefully handled, resulting in a Windows operating system crash (BSOD).

Source: https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/

But... But … Where Was the AppSec??!!

Now Notice What They Identified In ”How We Prevent” Especially the FIRST Thing

How Do We Prevent This From Happening Again?

Software Resiliency and Testing

•Improve Rapid Response Content testing by using testing types such as:

- Local developer testing
- Content update and rollback testing
- Stress testing, fuzzing and fault injection
- Stability testing
- Content interface testing

•Add additional validation checks to the Content Validator for Rapid Response Content. A new check is in process to guard against this type of problematic content from being deployed in the future.

•Enhance existing error handling in the Content Interpreter.

Source: https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/

Top Research Also Agrees

- The Web Application layer is a top threat action for incidents & breaches[1]
  - **40% of breaches, 50% of incidents**
  - **55% of System Intrusion**
- Average Data breach cost: **USD $4.88M**[2]
  - **Stolen / compromised credentials 16%**
  - **Phishing 15%.**
  - **Cloud misconfiguration 12%**

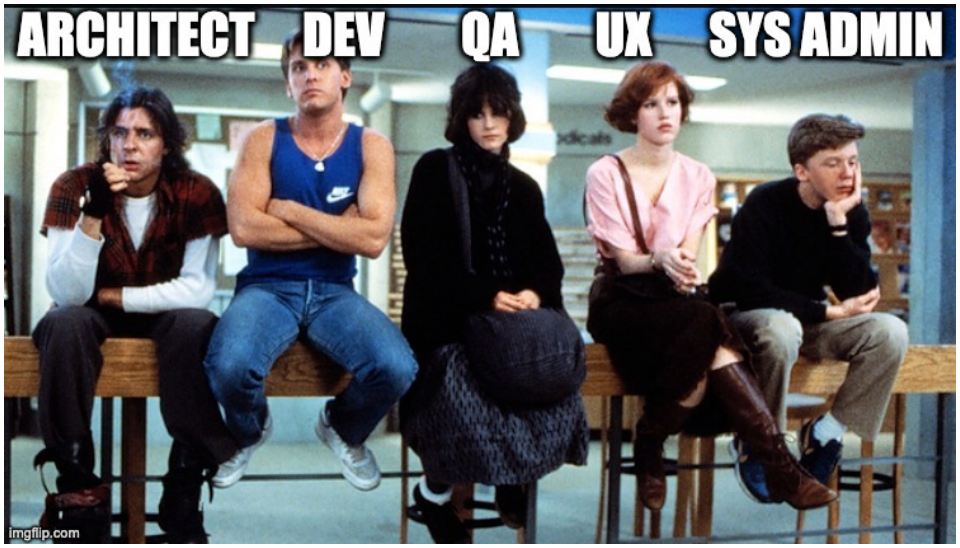
1. Verizon data breach report: https://www.verizon.com/business/resources/reports/dbir/
2. IBM Cost of a Data Breach 2024 report: https://www.ibm.com/security/data-breach

70% of dev teams

admit to skipping security steps

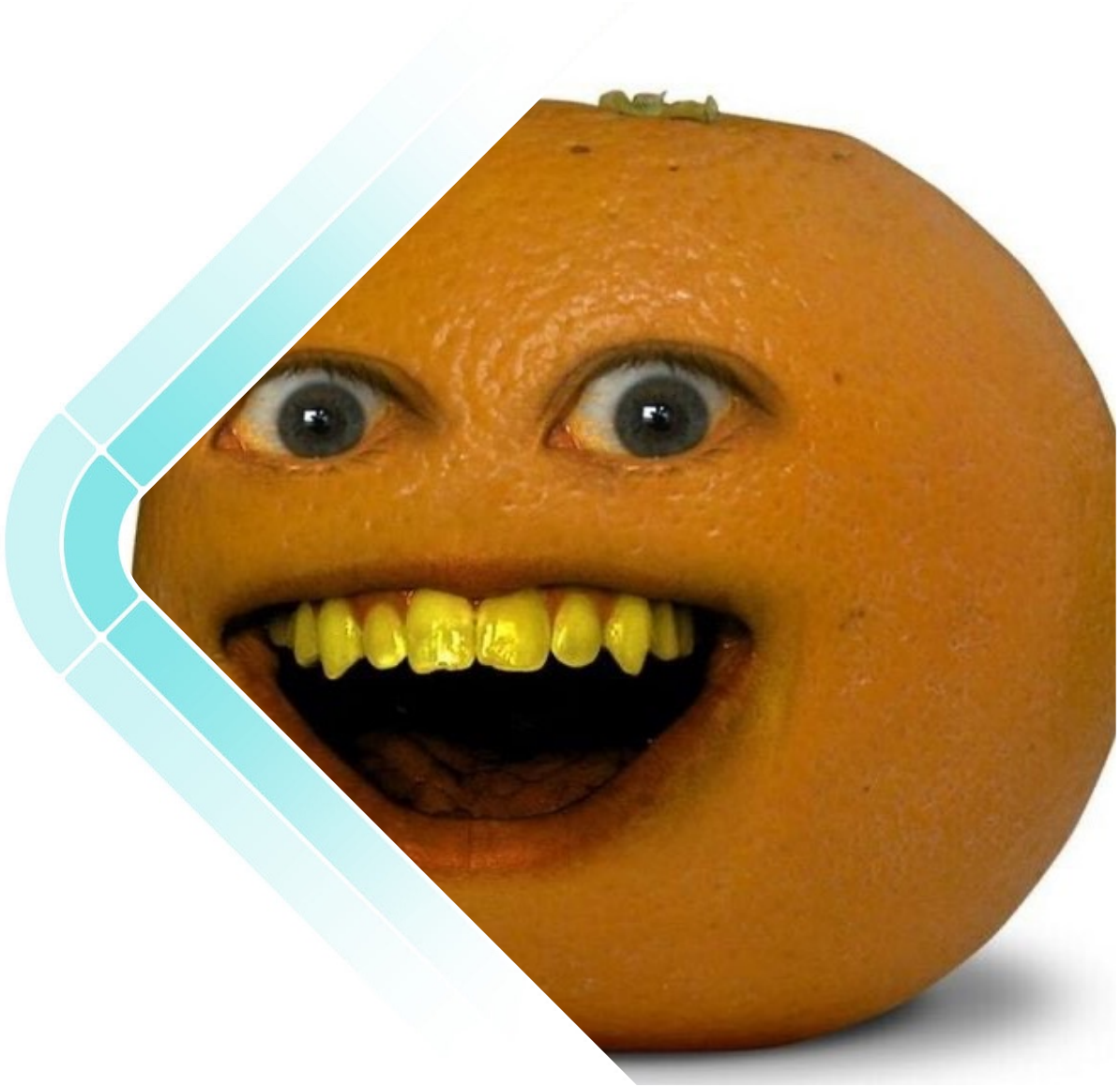
- Study by Invicti Security, venturebeat.com

How NOT To Be A Necessity



HCLSoftware

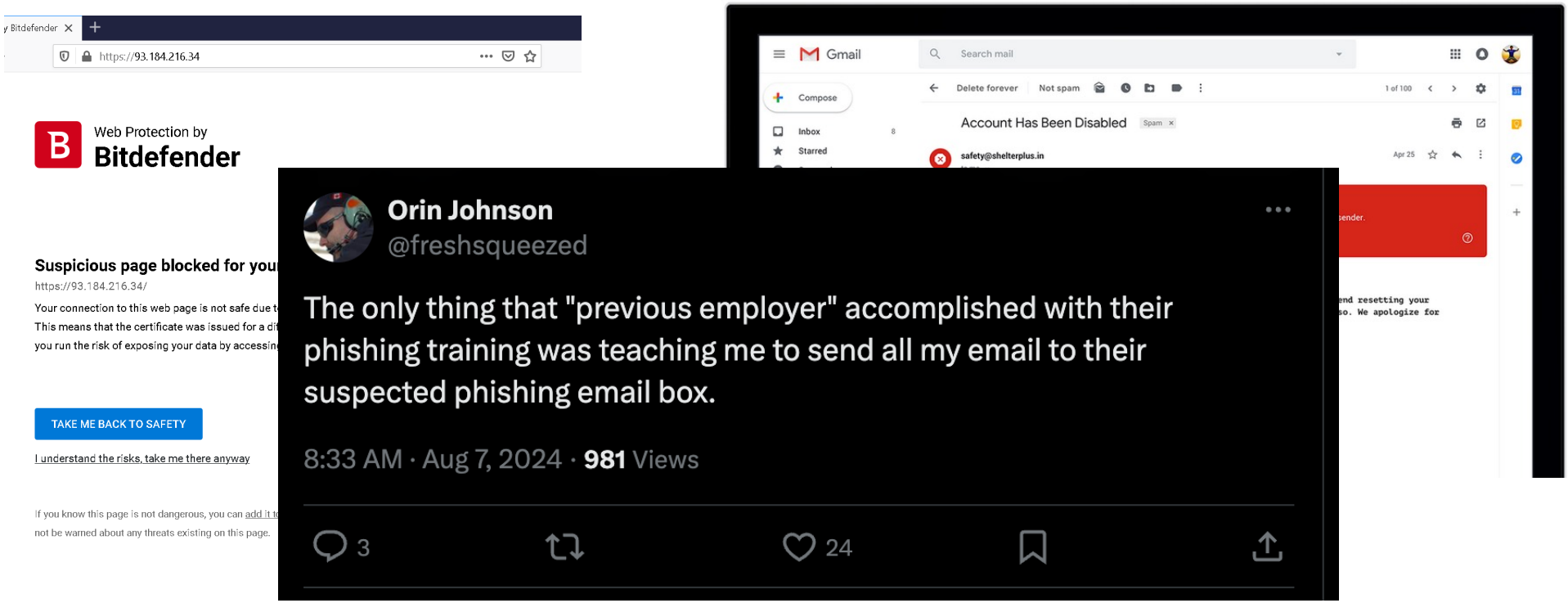
First Step: Don’t Be a Nusiance



What is A Nuisance?

- Only shows up when it is a crisis or problem.
- Expects “drop everything and fix” because issue is critical
- Little to no real AppSec / Secure Coding training
- Compliance is more punitive than proactive

We Know Things Like This Don’t Help



What Does It Look Like Specific to AppSec?
Excerpts from "DevSecOps Worst Practices" by Tanya Janca

- **Reporting Bugs in A Different System**
  - Just makes it harder to know what is really resolved and being addressed
  - And how do you effectively manage the backlog?
- **Unrealistic expectations from SLAs**
  - A critical bug found in production has likely been there for a long time. It is going to take time to resolve it
- **Gates in the wrong places**
  - Failing a build after getting code from a repository, coding, and submitting, deploying and only THEN find out it was vulnerable.
- **Failing on a False Positives**
  - Need to continually fine tune testing tools/policies and eliminate as many of these as possible

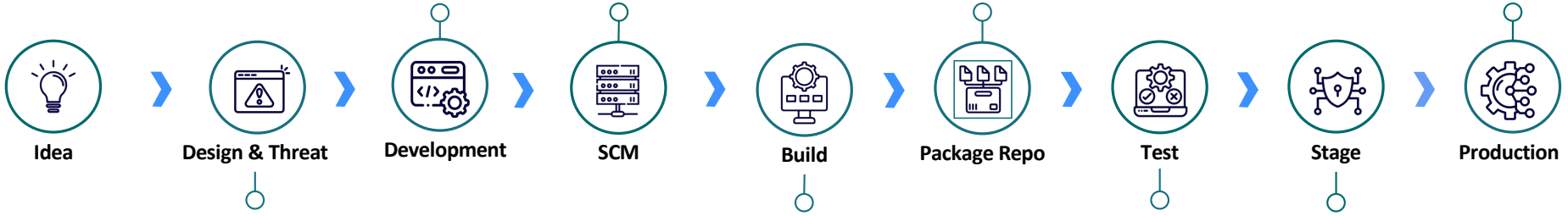
Source for DevSecOps Worst Practices: https://www.youtube.com/watch?v=rTekArq5ZzA

HCLSoftware

How to Not Be A Nuisance

Come Alongside

What Coming Alongside Looks Like: Building Continuous Security In Your Pipe

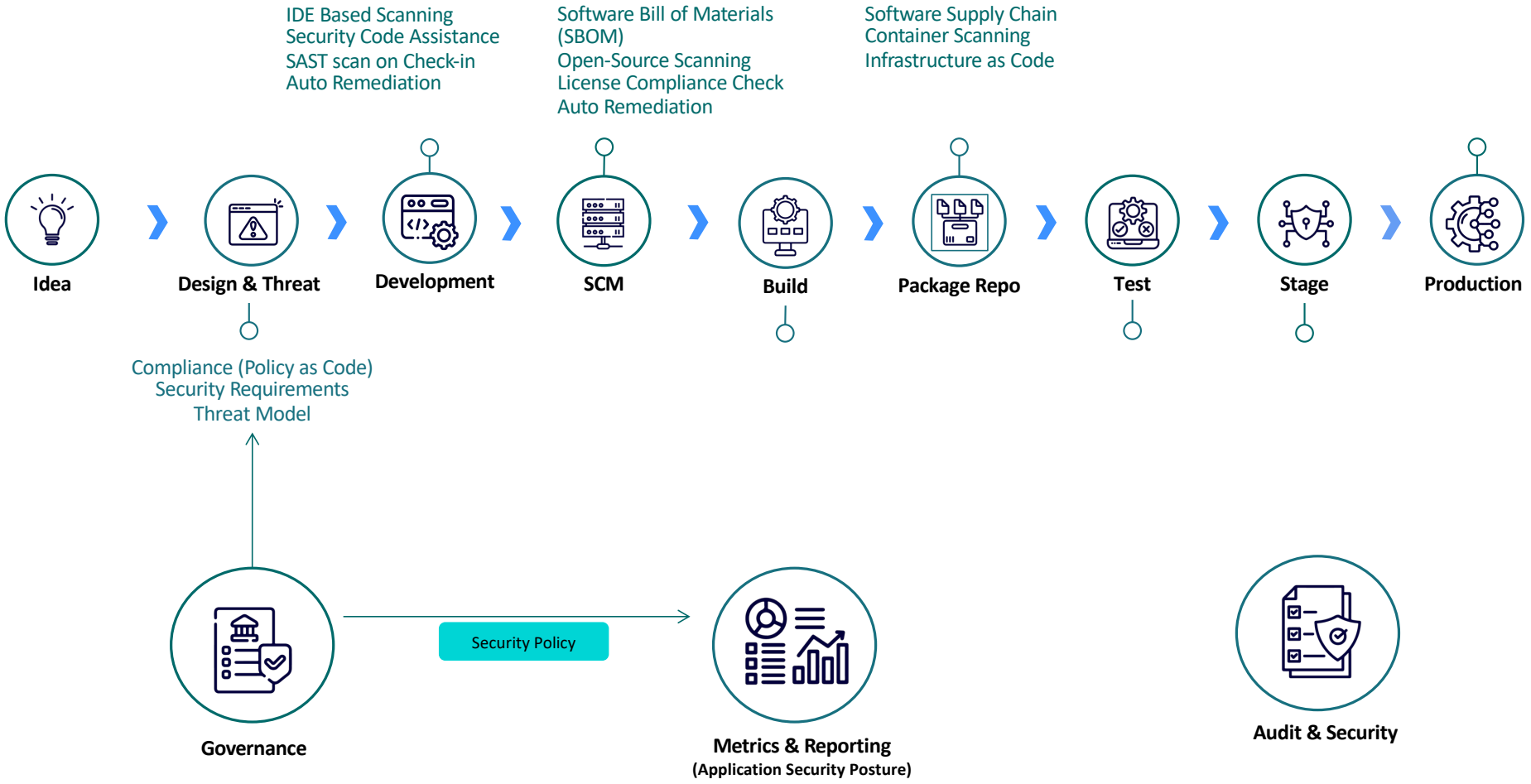


Governance

Metrics & Reporting
(Application Security Posture)

Audit & Security

What Coming Alongside Looks Like: Building Continuous Security In Your Pipe



Quick Wins to Come Alongside

Sounds basic but make sure to compare apples to apples

- Devs Need Environments That Look Like Production

Integrate into their ways of working

- Submit vulnerabilities in dev systems (JIRA, Azure DevOps, Gitlab, etc) to allow for better prioritization and balance
- Devs should at least be able to do SAST and SCA easily – ideally in their IDE
- Verify open source at time of consumption/collection
  - GitHub actions, quick scans, etc

Speak their language

- Treat security as part of quality
- Real secure coding training – budget for it
  - OWASP events, conference workshops
- Passive checking as code is saved is great

Another Great Way to Come Alongside: Threat Modeling

Include Developers as part of the threat modeling process

Keep it Simple

- The goal isn't a perfect model. It's to get people THINKING more about security
- Focus on the biggest threats and exposures

No idea is bad

- Avoid focusing only on the technical aspects – where could humans go wrong by mistake?

Remember this will be new to many

- Encourage with examples
- A little silence is OK – might just be thinking

Threat Modeling Manifesto: https://www.threatmodelingmanifesto.org/

Can AI Help Here? I Figured I'd Try - Copilot

COPILOT | Work | Web | NOTEBOOK

You

draw me a threat model for the Altoro Mutual application

Copilot

Sure, I can help with that! Here's a basic threat model for the Altoro Mutual application:

Threat Model for Altoro Mutual Application

1. Assets

- **Customer Data**: Personal information, account details, transaction history.
- **Financial Data**: Account balances, transaction records.
- **Application Code**: Source code, APIs.
- **Infrastructure**: Servers, databases, network devices.

2. Potential Threats

- **Data Breach**: Unauthorized access to customer and financial data.
- **SQL Injection**: Malicious SQL queries to manipulate the database.
- **Cross-Site Scripting (XSS)**: Injection of malicious scripts into web pages.
- **Denial of Service (DoS)**: Overloading the system to make it unavailable.
- **Man-in-the-Middle (MitM) Attacks**: Intercepting communication between users and the application.
- **Phishing**: Deceptive attempts to obtain sensitive information.

3. Vulnerabilities

- **Weak Authentication**: Inadequate password policies, lack of multi-factor authentication.
- **Unpatched Software**: Outdated software with known vulnerabilities.
- **Improper Input Validation**: Failure to sanitize user inputs.
- **Insecure Communication**: Lack of encryption for data in transit.

But Quickly Found Out We Have A Ways To Go

- **Logging and Monitoring**: Implement comprehensive logging and real-time monitoring.

Would you like more details on any specific part of this threat model?

You

Can you make this into a picture?

Copilot

I'll try to create that.

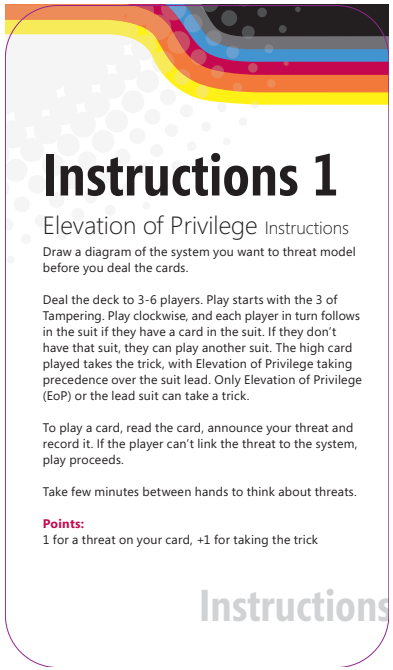
AI-generated content may be incorrect

A threat model for the Altoro Mutual application

Designer | Powered by DALL·E 3 | 98

Thank you! I'm excited to see it. | Take your time, no rush! | Can you add some color?

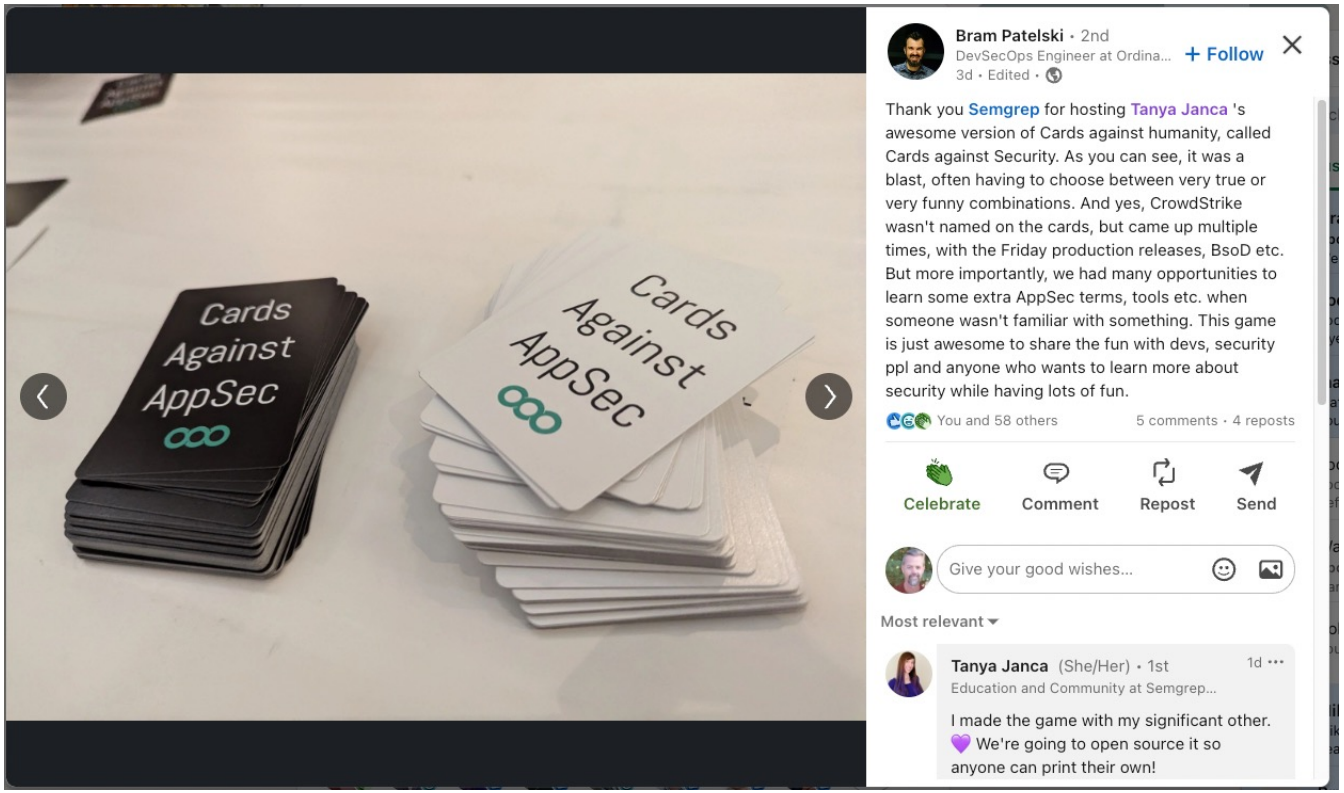
What Does Work Well Here - Making A Game of It
Two ”Classics”



Elevation of Privilege -
Find on Github & Microsoft

Backdoors & Breaches – Find
at BlackHillsInfosec.com

And One Brand New Just Introduced at DefCon!!
From Tanya Janca



HCLSoftware

Second Step: Get Rid Of Noise!

What is Noise?

- Unnecessary or low value activities
- ”Busy work”, “Security Theater”, etc
- Takes people away from higher value efforts

Common Categories

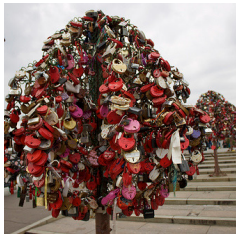
Created Complexity

Caused Confusion

Cryptic Compliance

Cumbersome Configurations

Noise Examples in Application Security



Created Complexity

- Different systems and criteria for vulnerability management (what's "medium"?)
- Vulnerabilities presented without context any context (code, attack vector, etc)

Caused Confusion

- Is this vulnerability real or a false positive? Did anyone check?
- Which critical is most important to fix first?
- How do I get help?

Cryptic Compliance

- How do I even know if my environment is compliant?
- Why can't I do what I need for my role (e.g. download from shared site, update a VM, get an environment)?
- Where can I get reliable code from?

Cumbersome Configurations

- How do I automate testing well when I need multiple logins for the session?
- Choosing tools and plugins and assuming they will integrate well
- Poor, Simplistic or no documentation

HCLSoftware

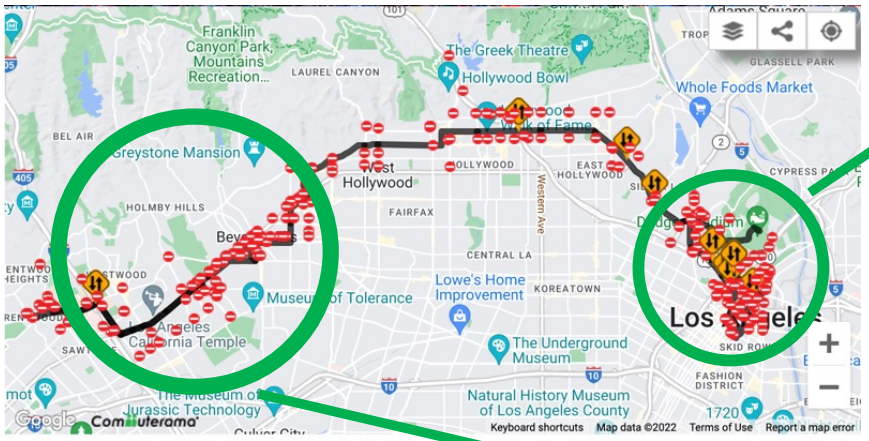
How Not To Be Noisy

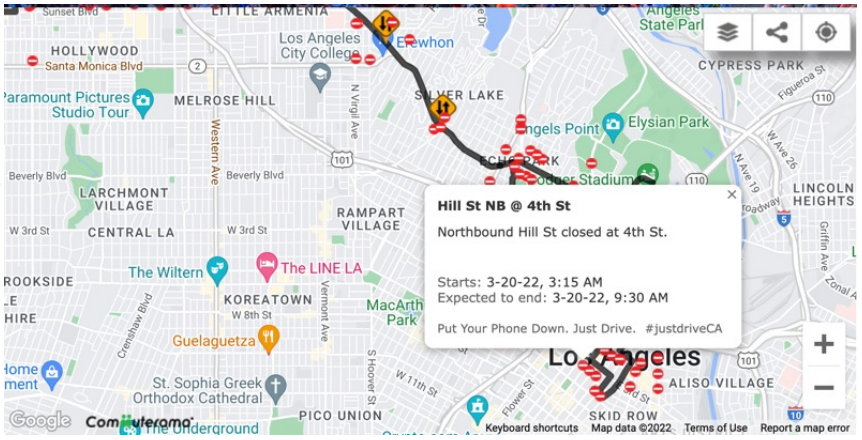
Context Over Content

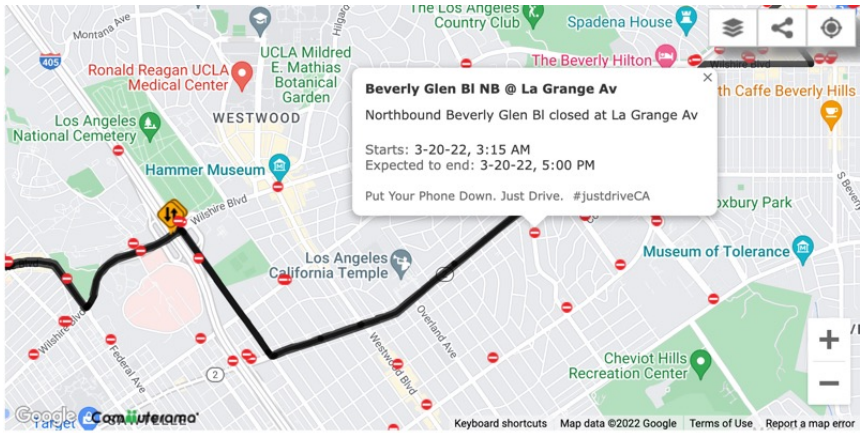
LANDMARKS

1. Dodger Stadium
2. Chinatown Dragon Gate
3. Olvera Street
4. Los Angeles City Hall
5. Little Tokyo
6. Cathedral of our Lady of Angels
7. Dorothy Chandler Pavillion
8. Echo Park Lake
9. Barnsdall Park
10. Pantages Theater
11. Capitol Records Tower
12. Hollywood & Vine
13. Hollywood Walk of Fame
14. Musso & Frank Grill
15. Grauman's Chinese Theater
16. Chateau Marmont
17. Whisky A Go Go
18. The Troubador
19. Beverly Hills City Hall
20. Rodeo Drive
21. Historic Route 66

Source: https://www.mccourtfoundation.org/event/los-angeles-marathon/distances-courses/

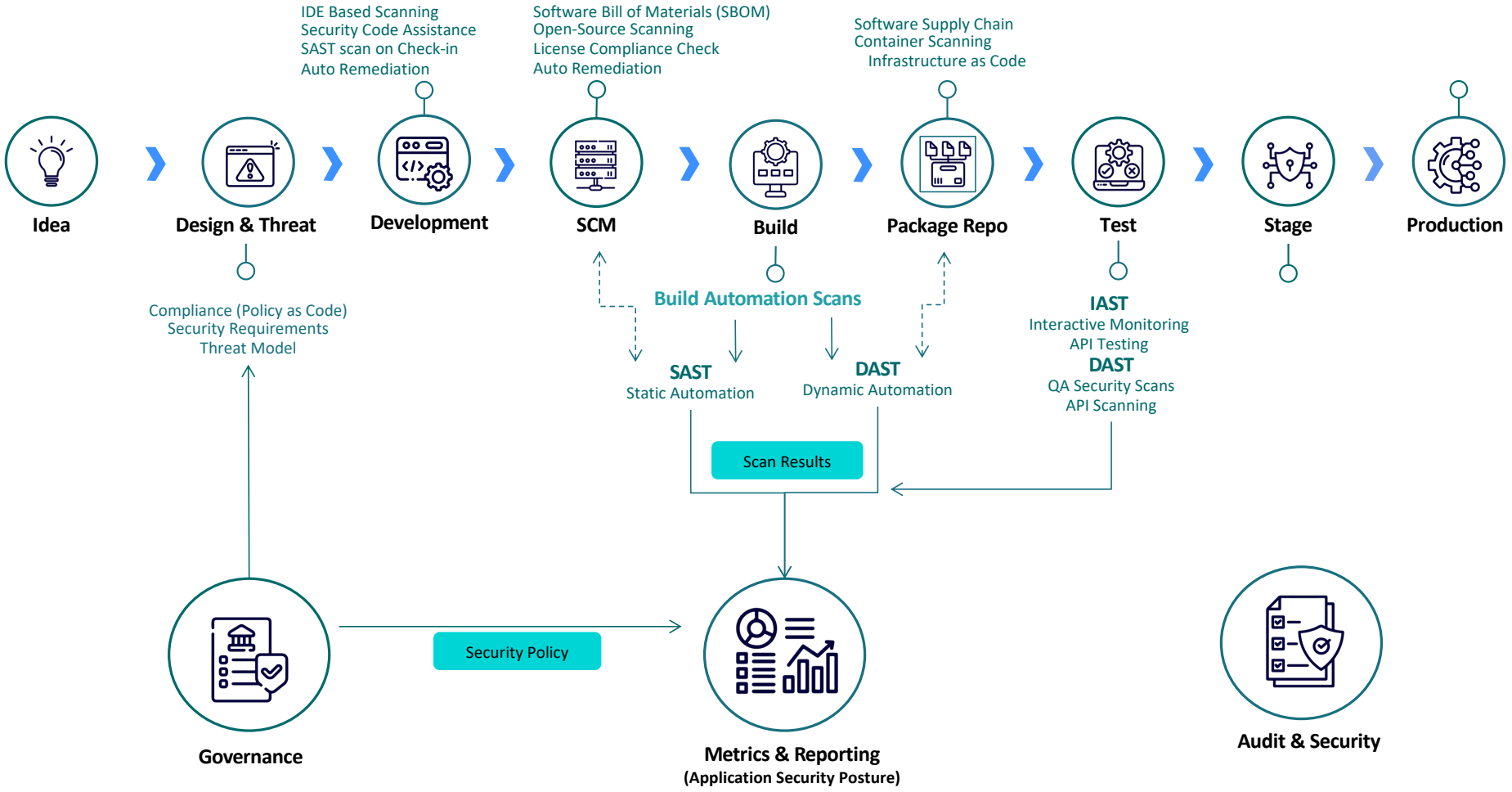






Roads Closed for 6-14+ hrs

Continuous Security Model



What is This??

- A Zebra?
- A Horse?
- A Giraffe?

It’s An Okapi

It has the …

- Head of a Giraffe
- Body of a Horse
- Legs & Stripes of a Zebra

Also On ICUN endangered list

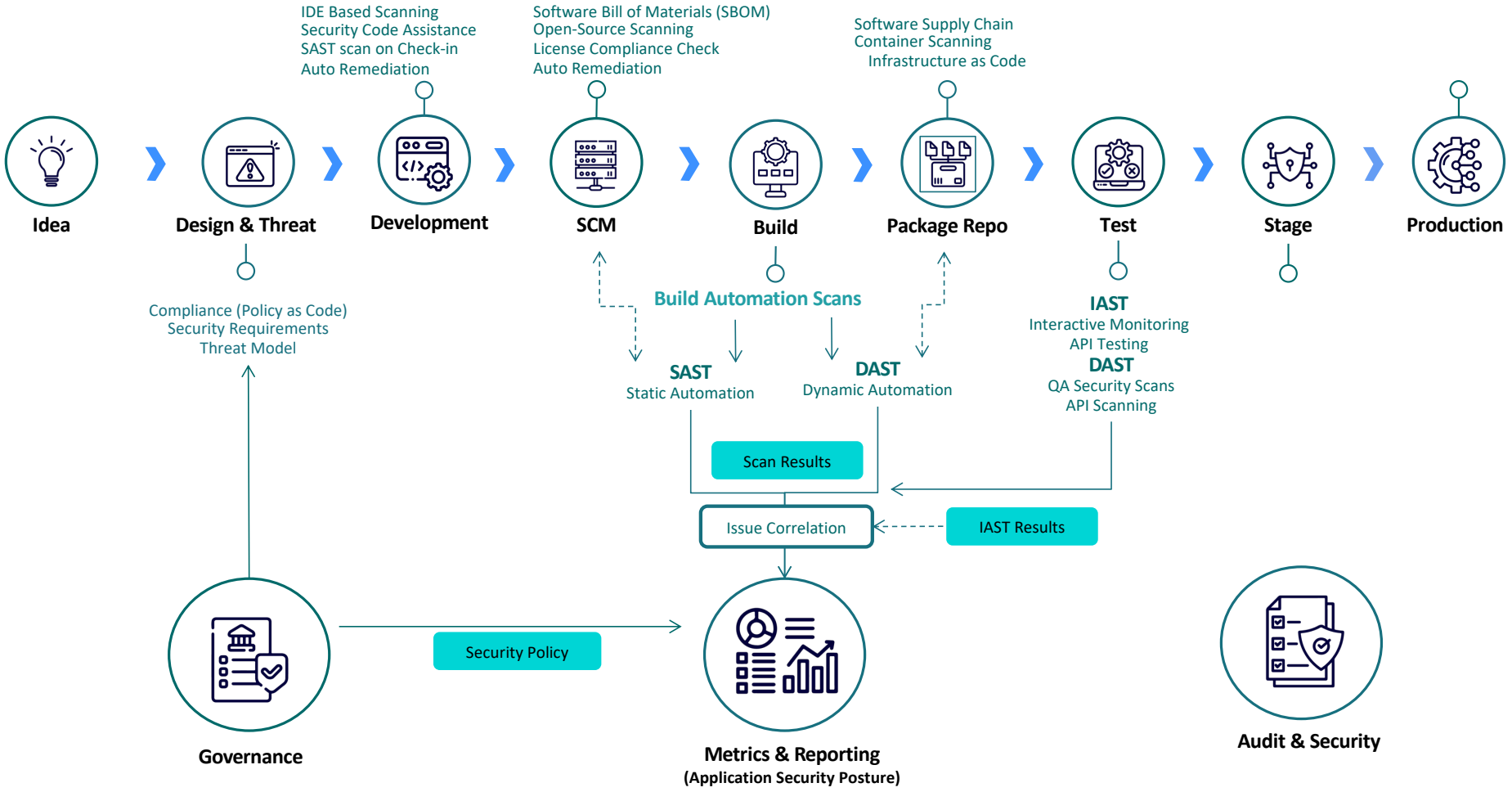
It is most closely related to the giraffe.

Now Consider A Typical Application Security Program

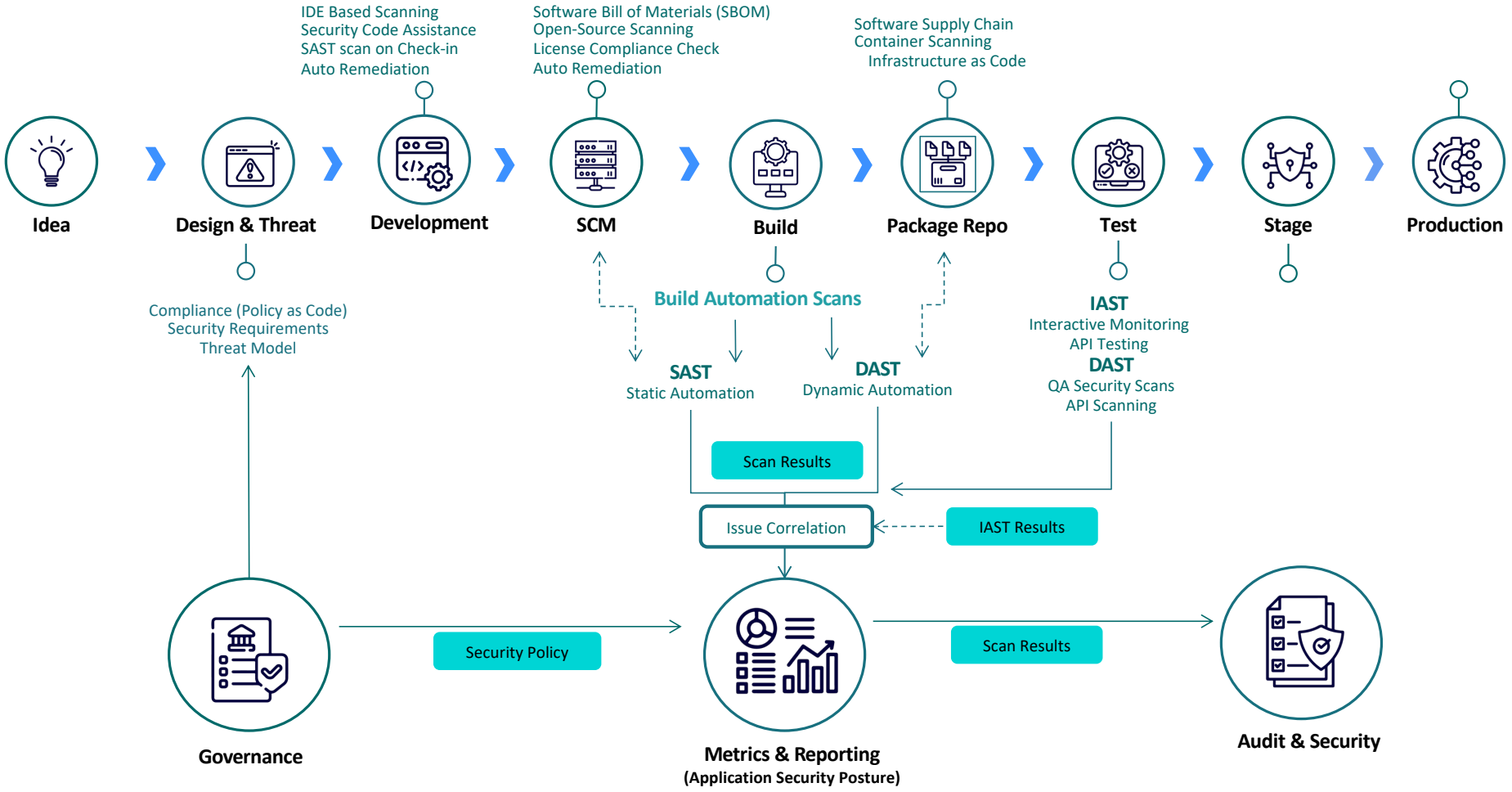
Separate Tools/Vendors for

- Static Application Security Testing
- Dynamic Application Security Testing
- Interactive Application Security Testing
- Software Composition Analysis
- Penetration Testing
- Mobile Security Testing
- Web Application Firewalls / Runtime Application Security Protection
- Infrastructure
- And the CISO Often feels like they’re on endangered list!
  - Average tenure is 26 months
  - Estimated skills gap of 3.5 Million jobs in 2021

Continuous Security Model



Continuous Security Model



Application Security Reducing Noise

Focus on Fixing, not just Finding

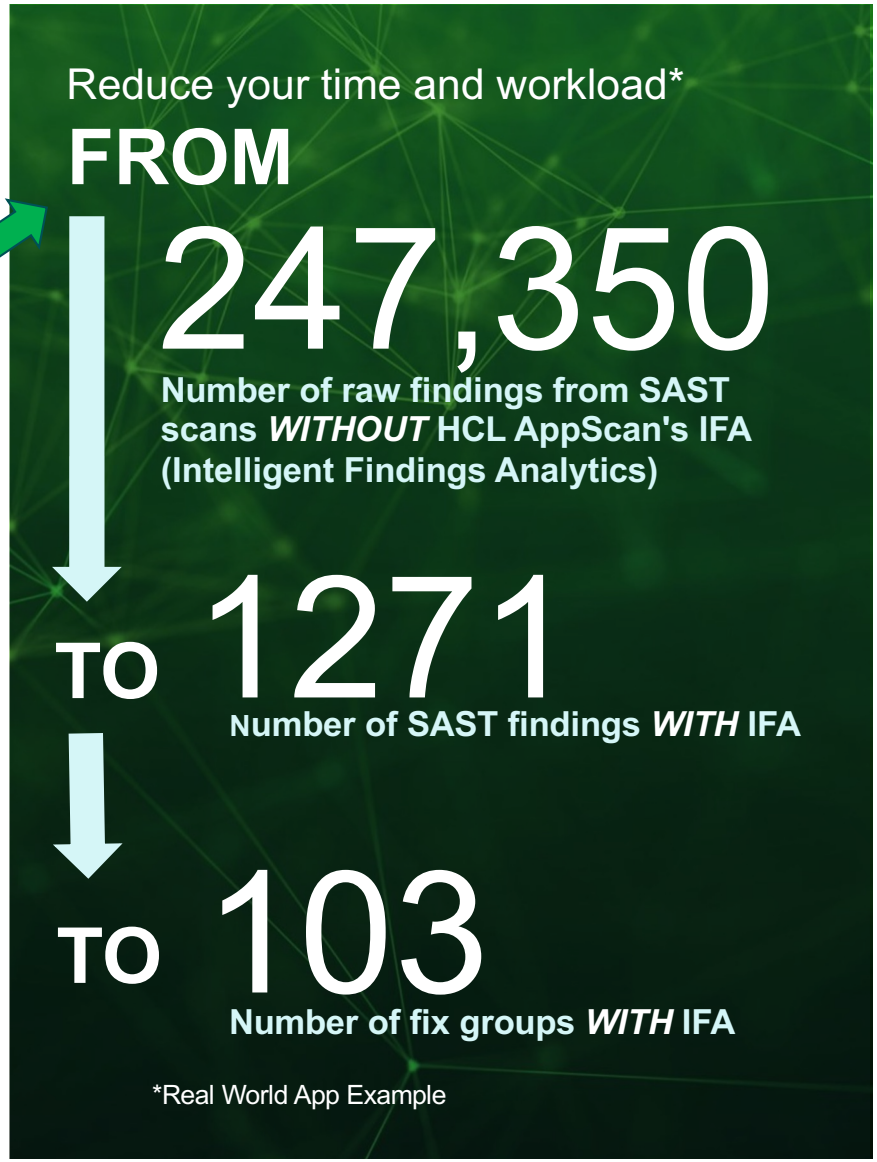
- Auto issue correlation with IAST, DAST, SAST
- Real-time auto-fix capabilities
- Machine Learning for reduced false positives

Security in the SDLC

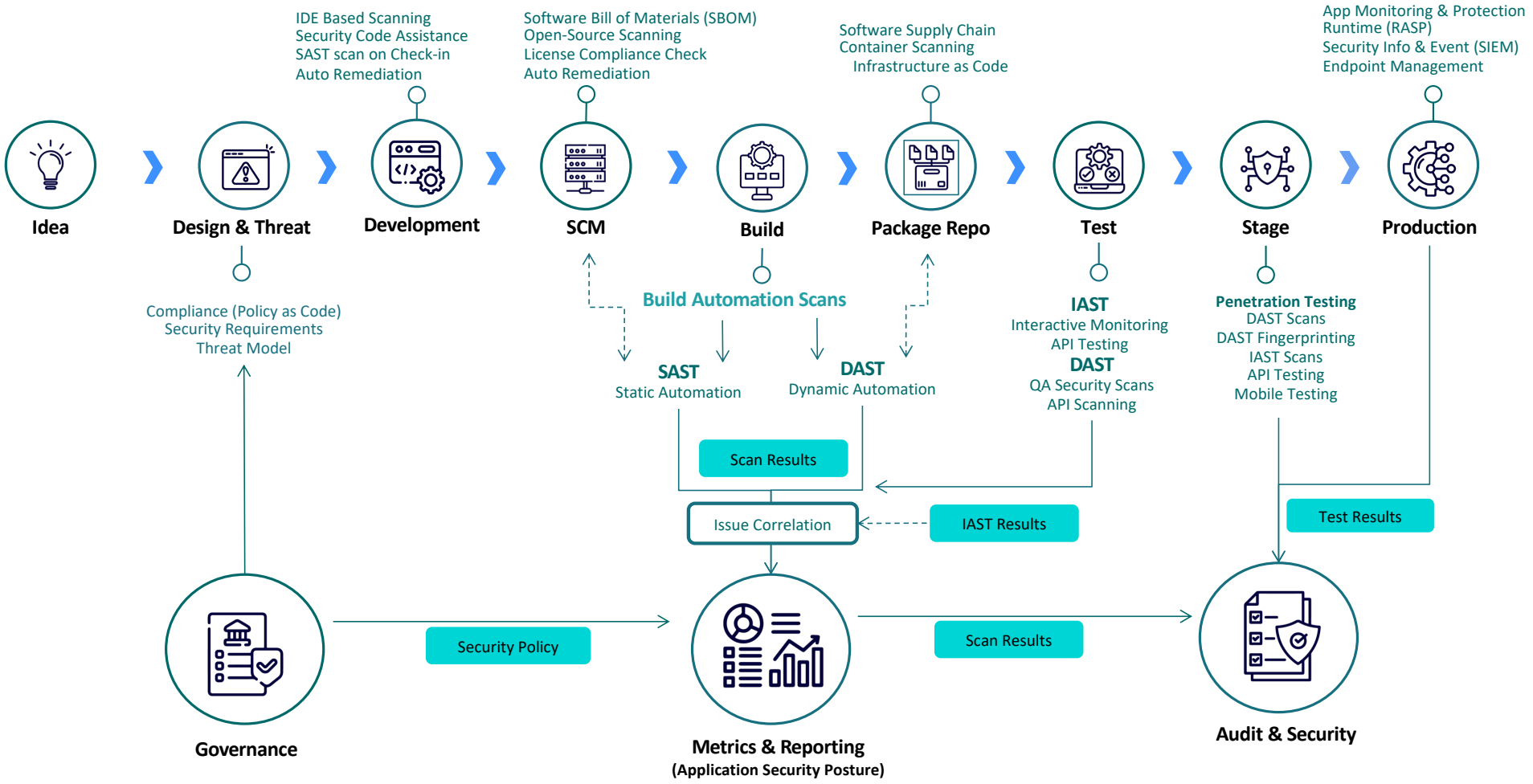
- Developers secure code as they write it.
- Integrations in most popular IDEs and CI/CD pipelines
- Incremental scanning / scan speed and depth control

Communicating

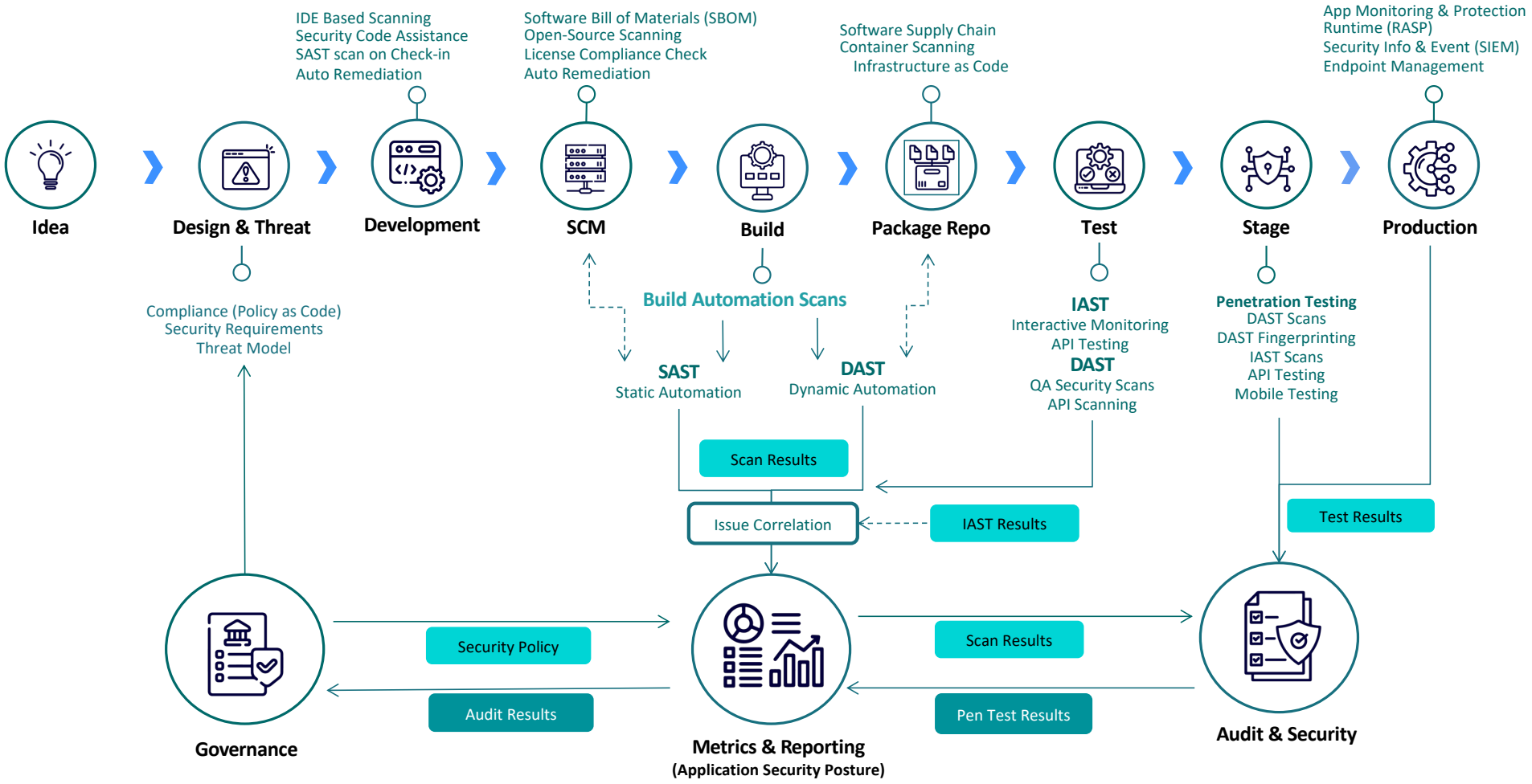
- Centralized dashboards with aggregated scan results
- Customizable lenses for risk posture and compliance
- Efficient vulnerability triage and remediation



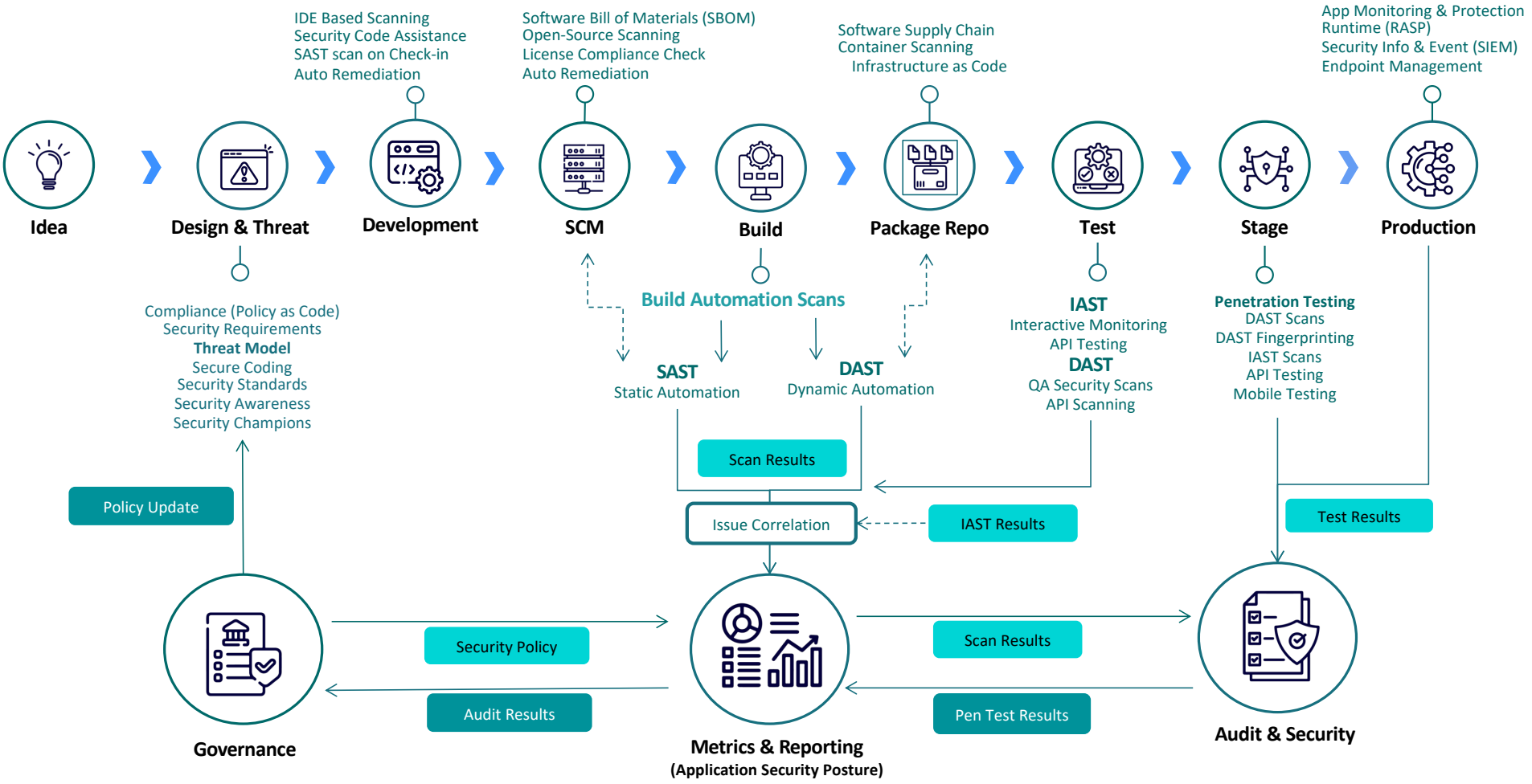
Continuous Security Model



Continuous Security Model



Continuous Security Model



Summary: Reducing Noise Application Security Drivers

MEASURE compliance against both security regulations & Data Regulations

- GDPR
- HIPAA
- PCI
- DISA
- CCPA
- OWASP Top 10
- SANS Top 25
- NIST
- and more.

ENSURE security in all parts of your software development lifecycle

- DAST
- SAST
- SCA
- IAST
- SBOM

FIND AND FIX vulnerabilities faster

- Prove/correlate exploitability
- Reduce false positives
- identify false negatives
- Remediation

USE application security to better implement GRC

- Share polices and measure effectiveness
- Mitigate risks
- Facilitate compliance

HCLSoftware

Final Step: Remember the Humans


Faster isn’t always better

Top Security Soft Skills

Soft Skill	Why It Matters	What It Looks Like
Empathy	Security is Stressful. Burnout is real.	Implementing security in context with their job role. Training vs Blaming.
Perspective	There is a business to run - it needs to for all of us to benefit.	Security finding ways to say ”YES” to the ideas and vision of the business. Building customer trust
Problem Resolution (not avoidance)	If issues aren’t addressed, we get security theater instead of health	Governance that naturally pushes people to do the right things
Managing “Up”	Security is a part of technical debt too – Low risk can become critical given right circumstances (log4j)	Coordination across disciplines DR and BC plans that work (and are tested!)
Engaging and Encouraging	Resolve problems before they become problems	Rewarding things done well more than punishing what isn’t
Prioritizing	Too many tools, dashboards, and alert fatigue leads to confusion in the short term, burnout in the long.	Well managed assets, systems level thinking. Managing overall health.

And If You Are Hiring:
Things Security Leaders Say They Are Specifically Looking For:

Trait	Why
Tenacity / Grit	Security problems are complex and require validation. Often hard to prove a negative or identify root cause.
Curiosity	Think of things others don't, consider ways we could be attacked, finds exploits before others do.
Great Communication	Clear and Concise beats cute and clever. Teammates need to know exactly what to do
Understanding how to make trade-offs	Resources and budgets are limited. Need to find the best combinations of controls, policies, procedures to provide right governance
Detail oriented	Little things that make a little difference that make a big difference (Do we know what we need to patch? Did we get them all?)

Some Other Great Resources

Books:

“Alice and Bob Learn Application Security” by Tanya Janca (Amazon, Indigo, Barnes & Noble, Audible)

“Cybersecurity Career Guide” by Alyssa Miller (Amazon, Manning)

“Relax: A Guide To True Cyber Security” by Ken Fanger (https://ontechnologypartners.com/relax/)

Talks:

Tanya Janca – RSA 2022 “Transforming Security Champions” Link

Jennifer Czaplewski – RSA 2023 “The Psychology fo DevSecOps” Link

Larry Maccerone – RSA 2020 “The Impact of Software Security Practice Adoption Quantified” Link

HCLSoftware

Thank You!

www.hclfederal.com
info@hclfederal.com

Speaker Email:
Rob.Cuddy@hclfederal.com