AppSec is Changing

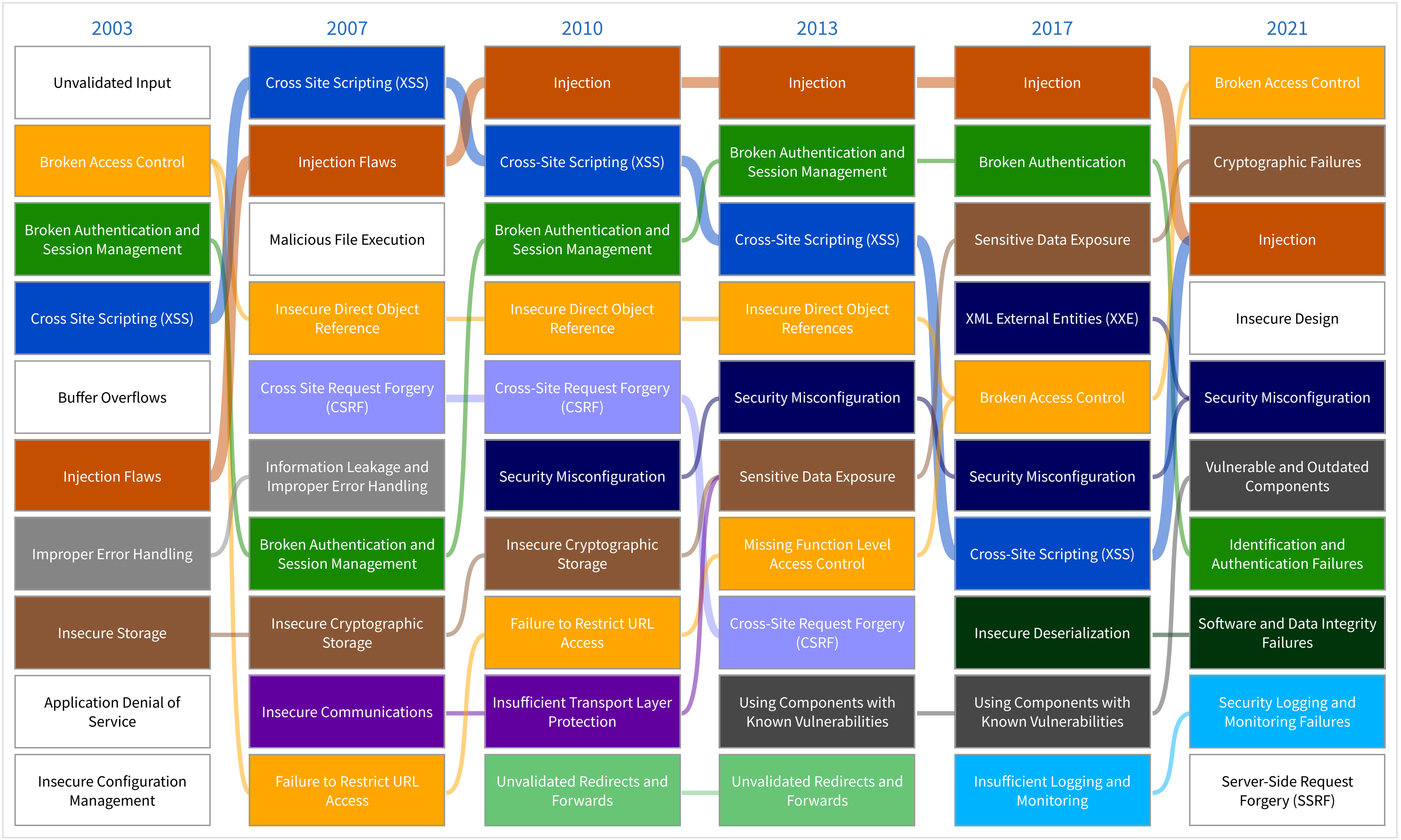
SecAppDev 2024

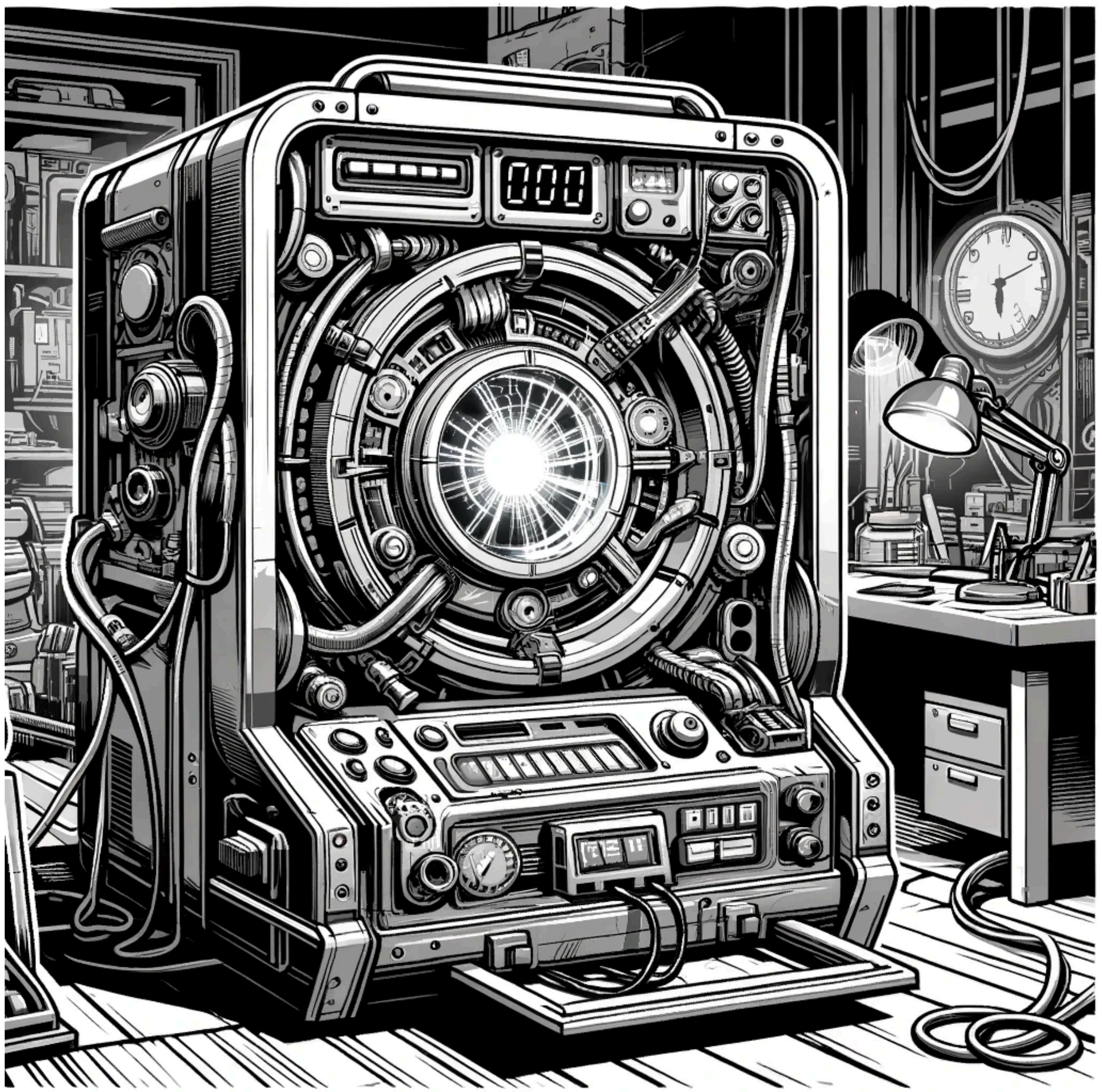
Erlend Oftedal

- Security researcher @ Crosspoint Labs
- Software Developer and Security Engineer/Architect/Tester for 20 years
- Builds and maintains free open source security tools such as retire.js
- OWASP Oslo chapter lead
- @webtonull

Disclaimer

This presentation is being given in the presenter's individual capacity and reflect the subjective views and opinions of the presenter. This presentation is not meant to express any views of Crosspoint Labs or any of its affiliates. The statements contained herein cannot be independently verified and are subject to change.


2003
Unvalidated Input
Broken Access Control
Broken Authentication and Session Management
Cross Site Scripting (XSS)
Buffer Overflows
Injection Flaws
Improper Error Handling
Insecure Storage
Application Denial of Service
Insecure Configuration Management
2007
Cross Site Scripting (XSS)
Injection Flaws
Malicious File Execution
Insecure Direct Object Reference
Cross Site Request Forgery (CSRF)
Information Leakage and Improper Error Handling
Broken Authentication and Session Management
Insecure Cryptographic Storage
Insecure Communications
Failure to Restrict URL Access
2010
Injection
Cross-Site Scripting (XSS)
Broken Authentication and Session Management
Insecure Direct Object Reference
Cross-Site Request Forgery (CSRF)
Security Misconfiguration
Insecure Cryptographic Storage
Failure to Restrict URL Access
Insufficient Transport Layer Protection
Unvalidated Redirects and Forwards
2013
Injection
Broken Authentication and Session Management
Cross-Site Scripting (XSS)
Insecure Direct Object References
Security Misconfiguration
Sensitive Data Exposure
Missing Function Level Access Control
Cross-Site Request Forgery (CSRF)
Using Components with Known Vulnerabilities
Unvalidated Redirects and Forwards
2017
Injection
Broken Authentication
Sensitive Data Exposure
XML External Entities (XXE)
Broken Access Control
Security Misconfiguration
Cross-Site Scripting (XSS)
Insecure Deserialization
Using Components with Known Vulnerabilities
Insufficient Logging and Monitoring
2021
Broken Access Control
Cryptographic Failures
Injection
Insecure Design
Security Misconfiguration
Vulnerable and Outdated Components
Identification and Authentication Failures
Software and Data Integrity Failures
Security Logging and Monitoring Failures
Server-Side Request Forgery (SSRF)


000

Medieval AppSec

Attackers focused on network, servers, and browsers

Little or no understanding of the need for AppSec

"Not my job"

"The department of no"

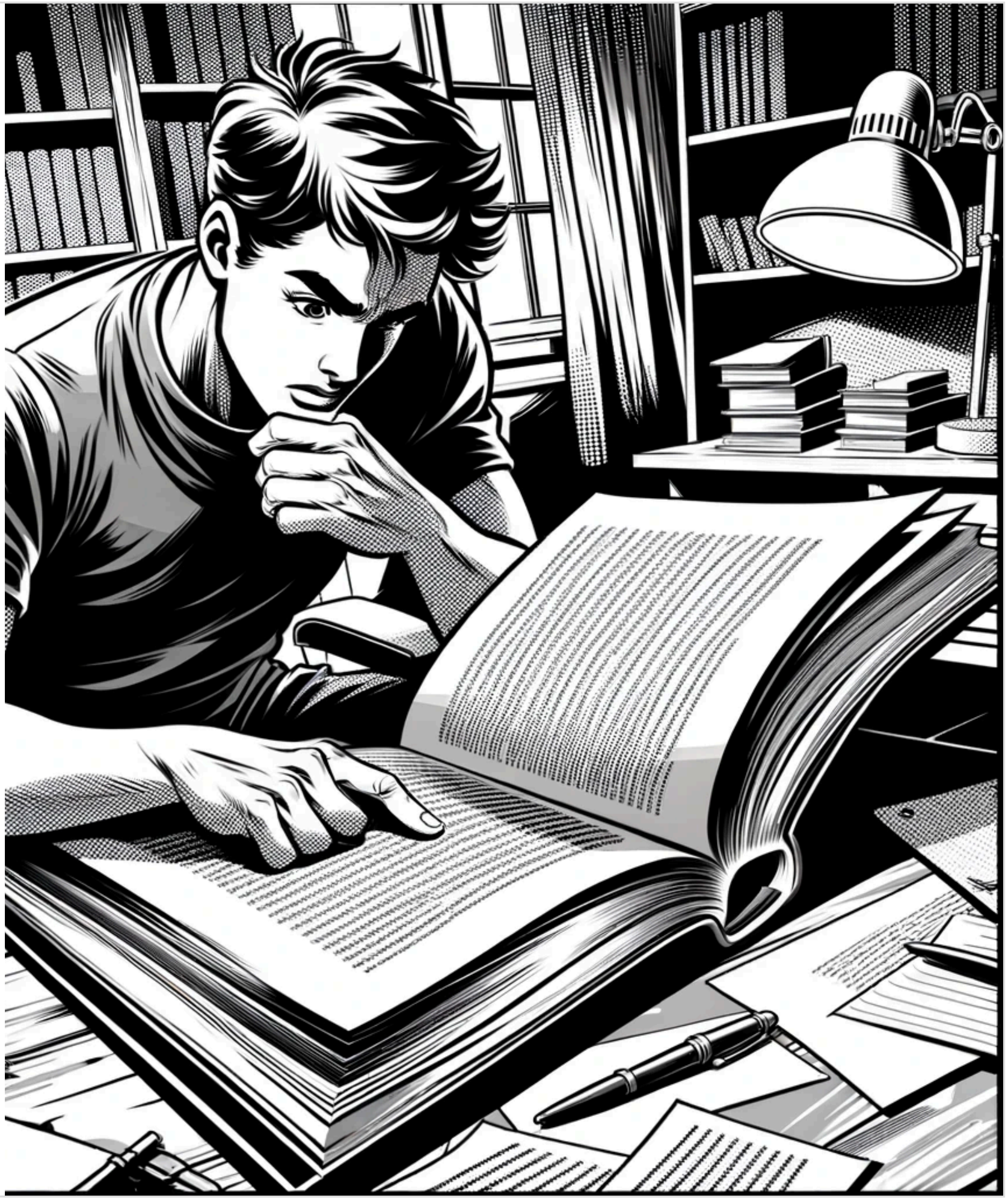
No management buy-in

Ad-hoc processes

"Pen test driven security"

No or unclear security requirements

No or unclear regulations

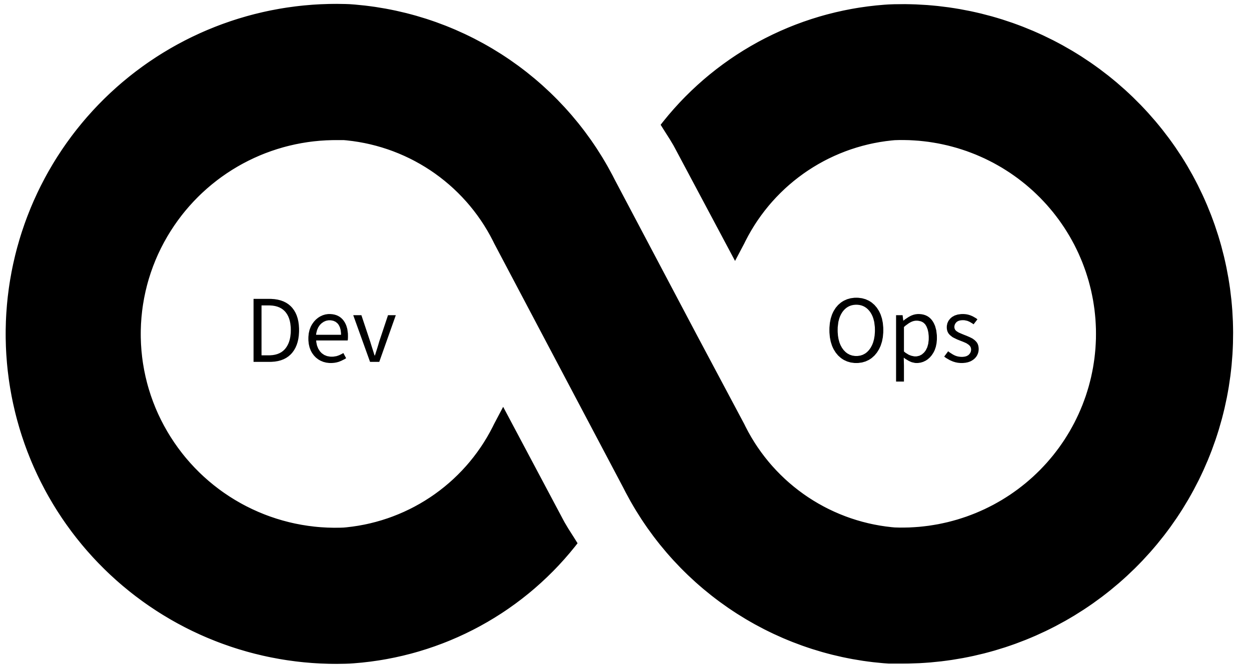


Unsung heroes -
Highly dependant on people

Intermission: XSS Worms

"But most of all, Samy is my hero"

The age of Dev(Sec)Ops

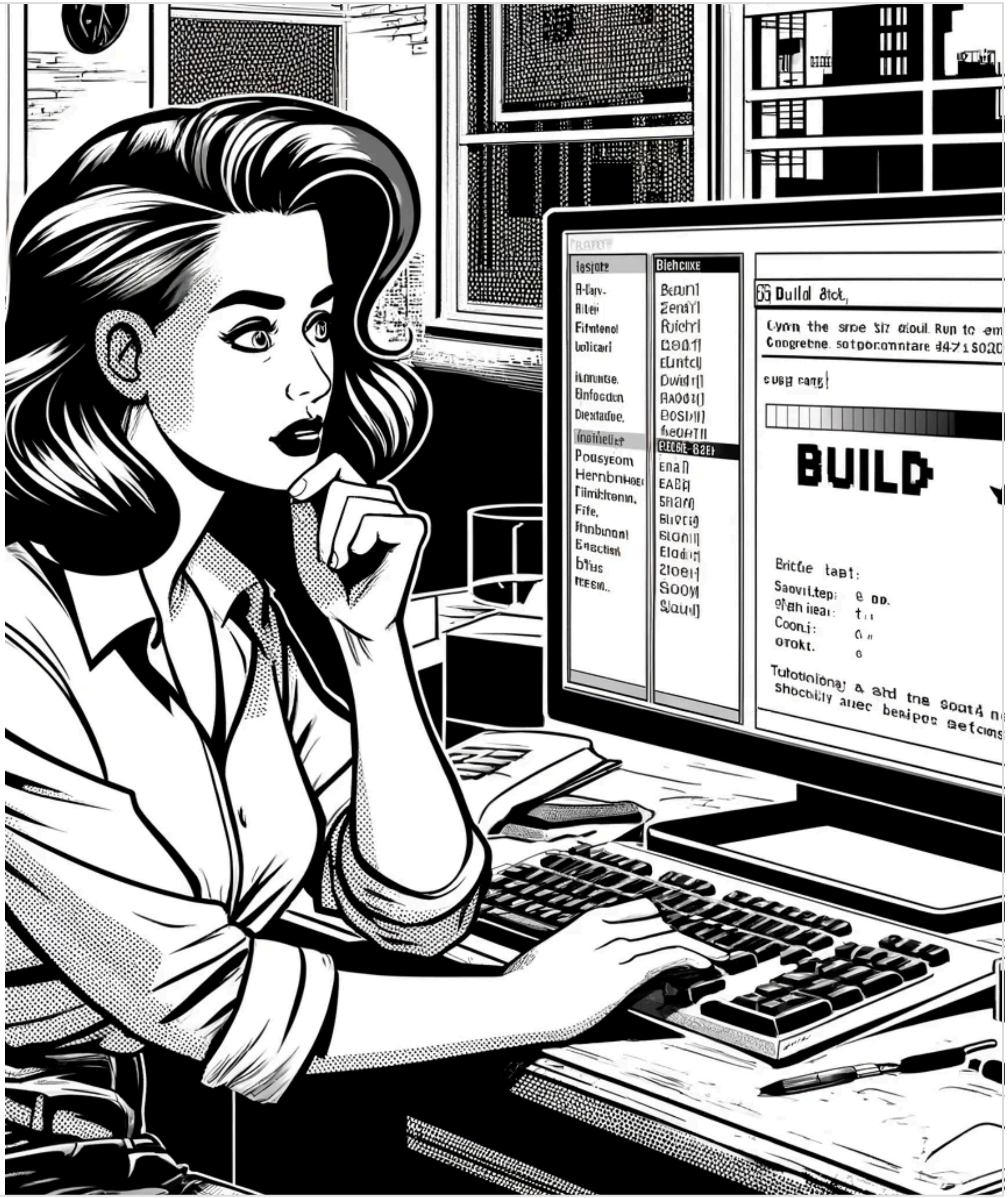

Dev
Ops

“ *Security testing can be automated, but true security assurance cannot be. The only way that we can free up enough cycles to have security assurance, is to automate all the things that can be.*

- Gene Kim

"Shift Left"

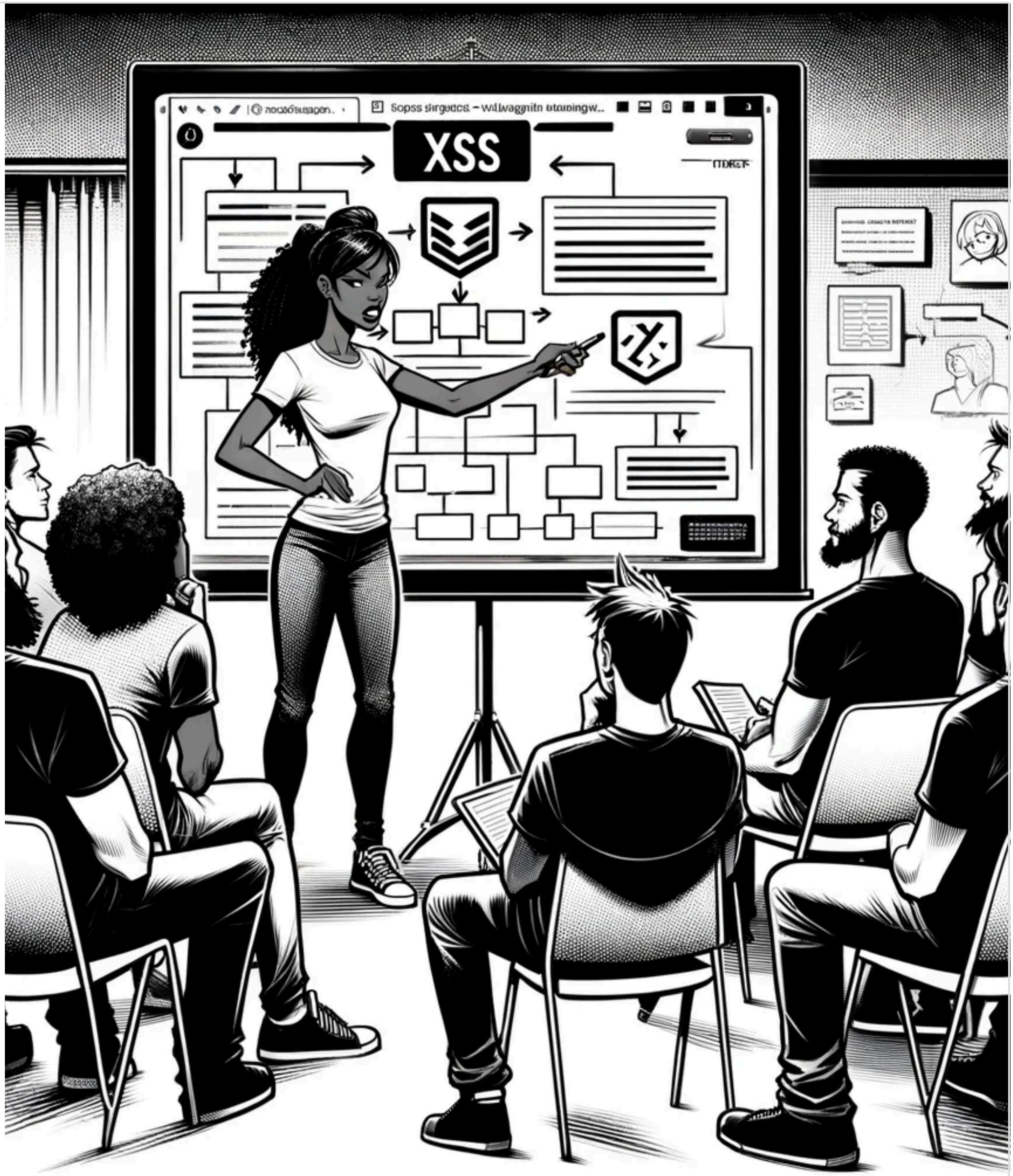
CI/CD and pipelines



Security Test Automation and Cadence

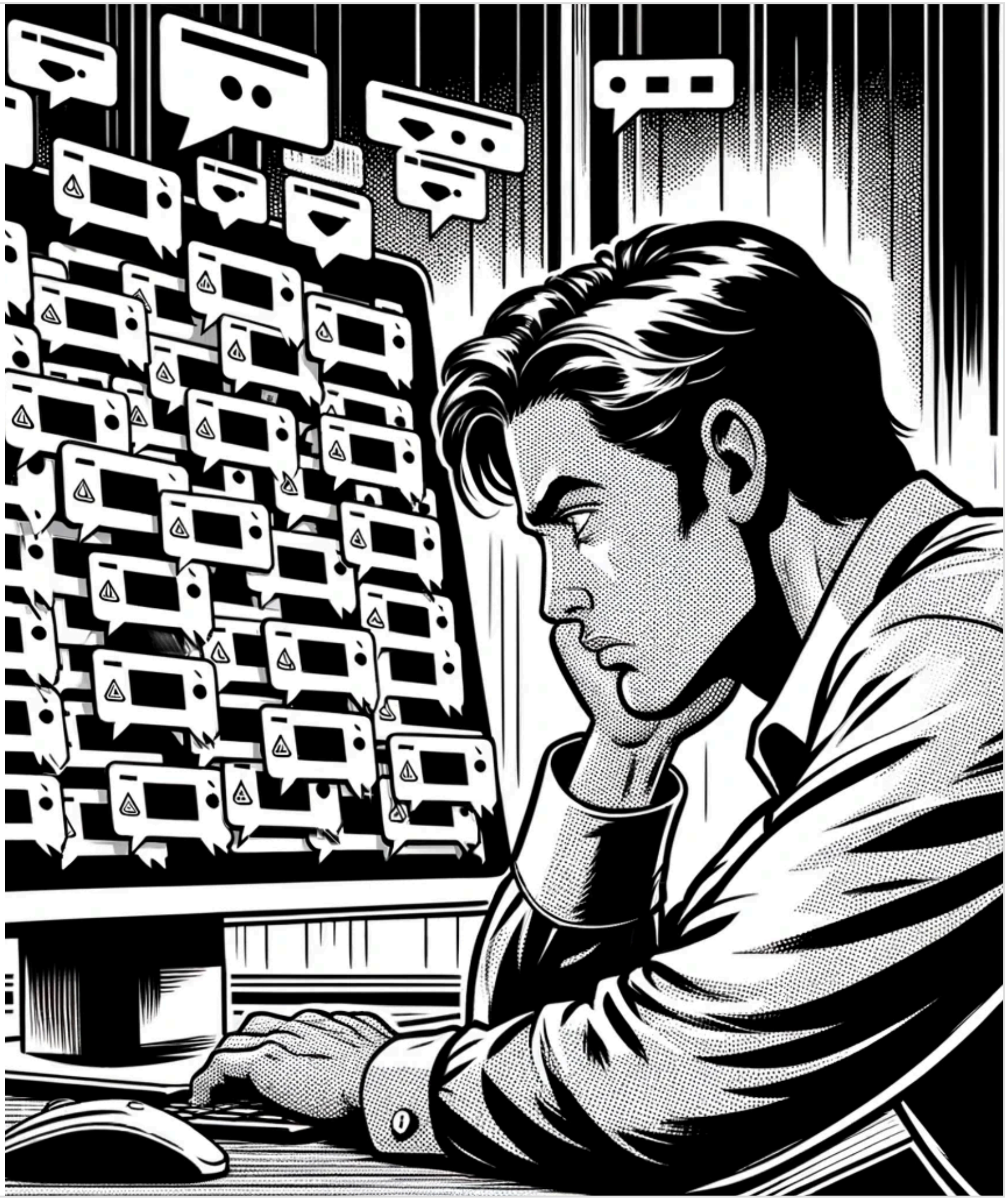
Secure Software Development Lifecycle (SDLC)

Security Champions and Security Engineers



Depedency management / Software Composition Analysis (SCA)

Alert fatigue

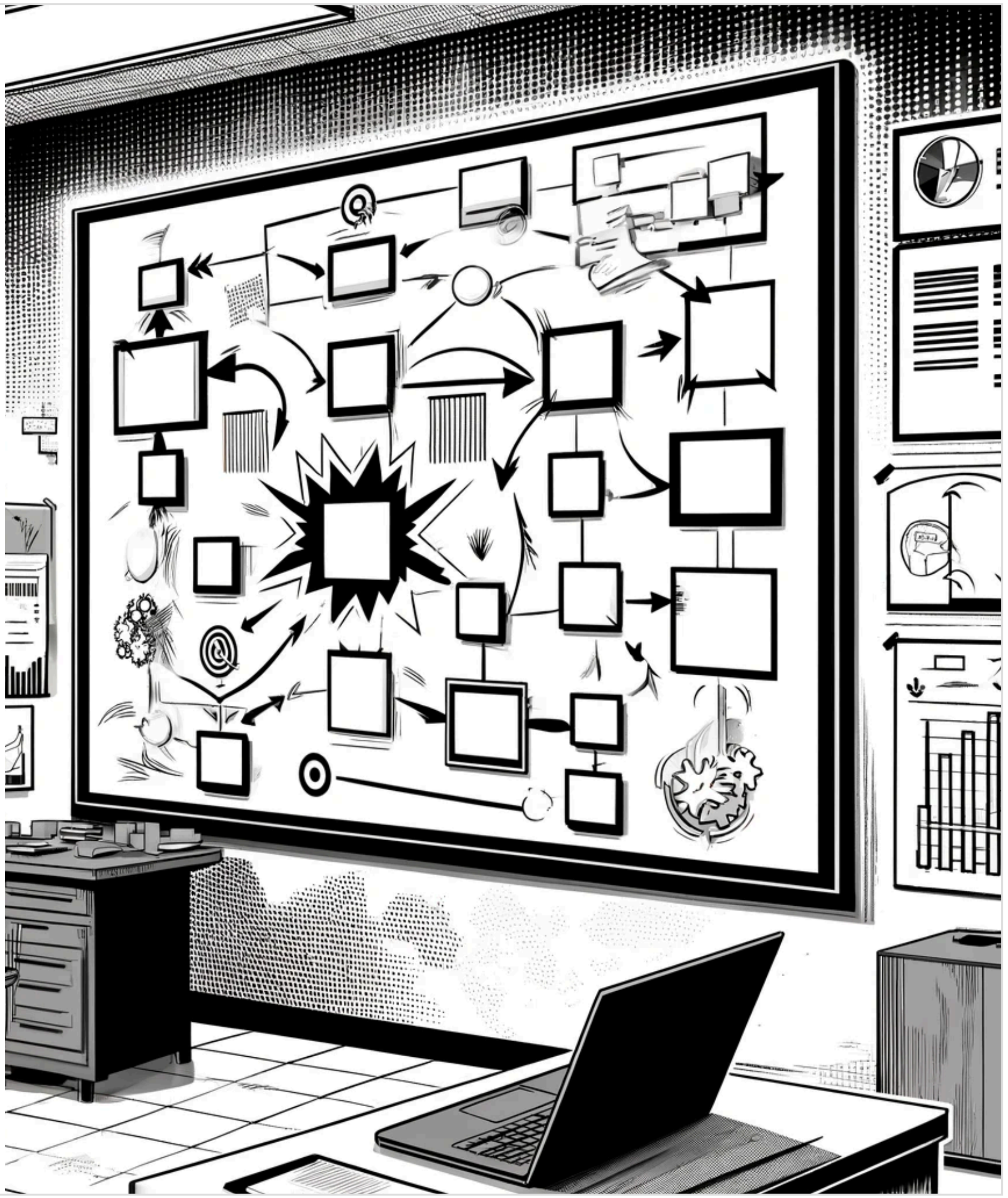


AppSec 2024

"(Application) Security is about making **only** the *expected* thing possible"

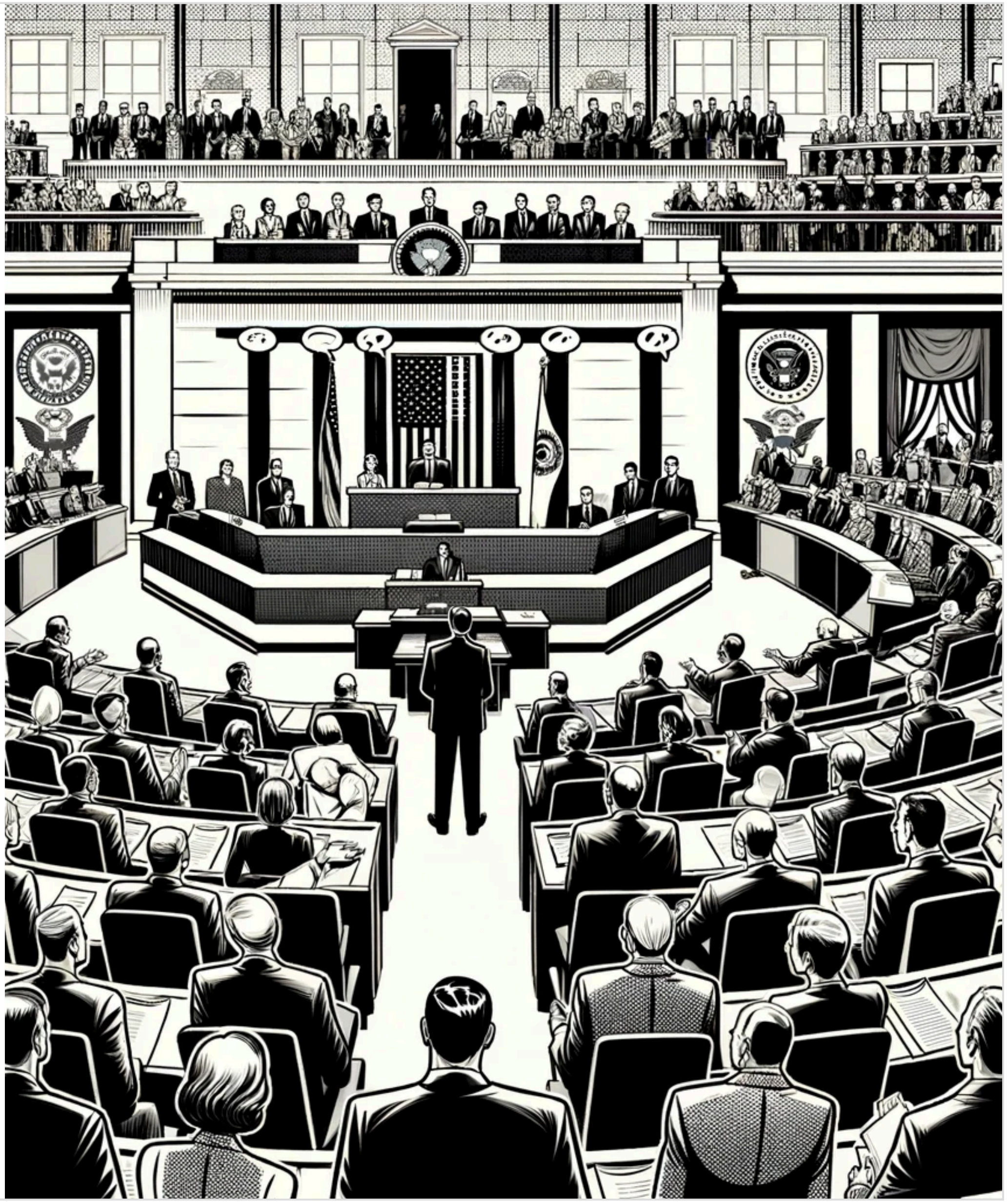
Developers are more involved in security tool selection

Appplication attacks are getting more complex



Accountability and Legislation

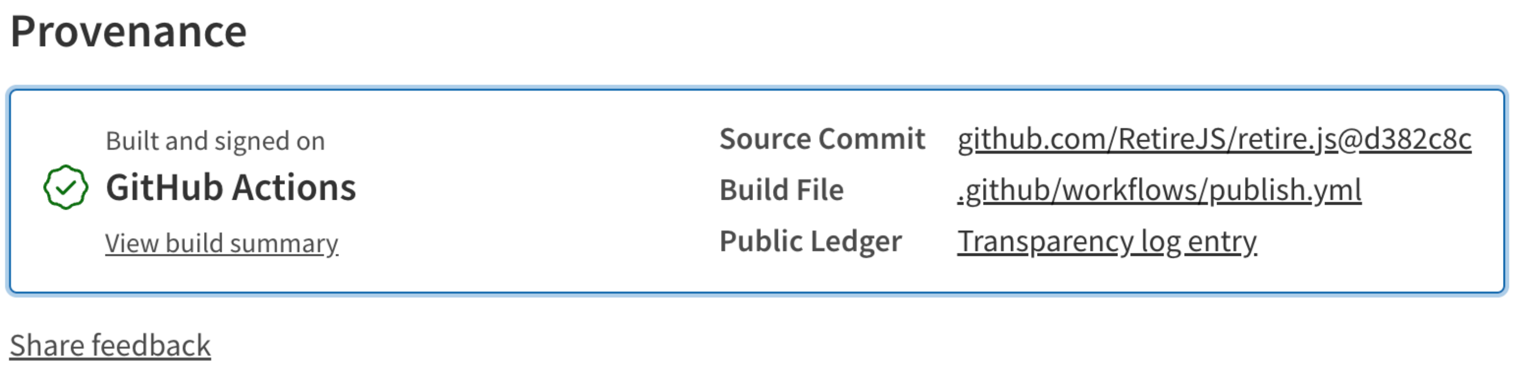
- Executive Order 14028 - Improving the Nation's Cybersecurity
- Cyber Resilience Act
- NIS2



Software Bill of Materials (SBOM) and SBOM management

Software Supply Chain Security (SSCS)

Collect and Analyze
Verify and Trace



Provenance

Built and signed on

GitHub Actions

View build summary

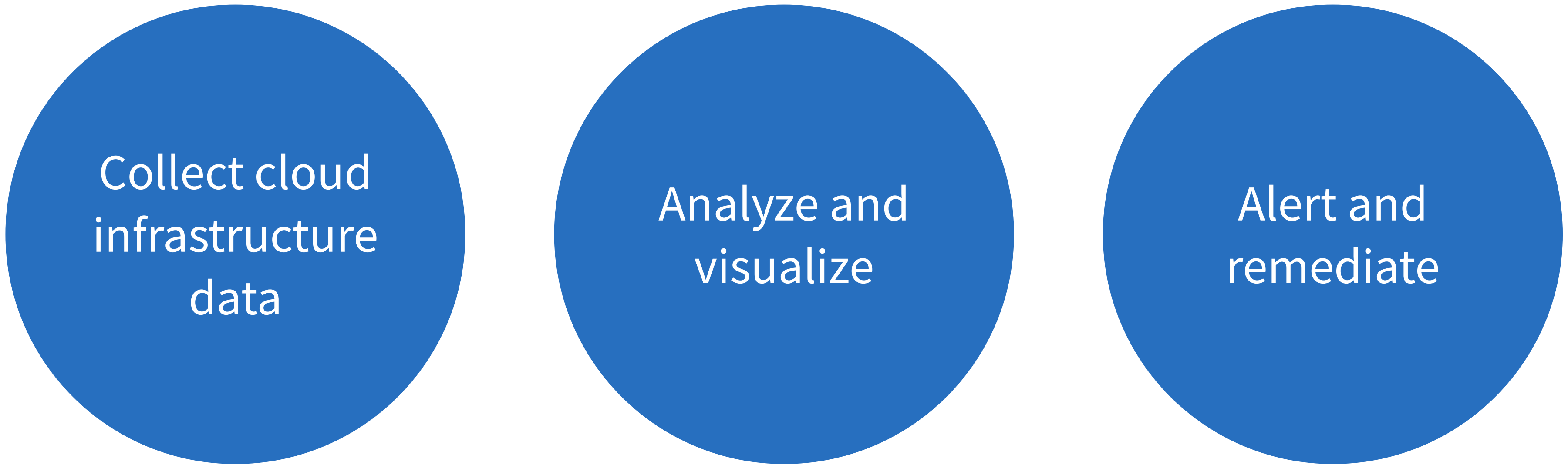
Source Commit	github.com/RetireJS/retire.js@d382c8c
Build File	.github/workflows/publish.yml
Public Ledger	Transparency log entry

Share feedback

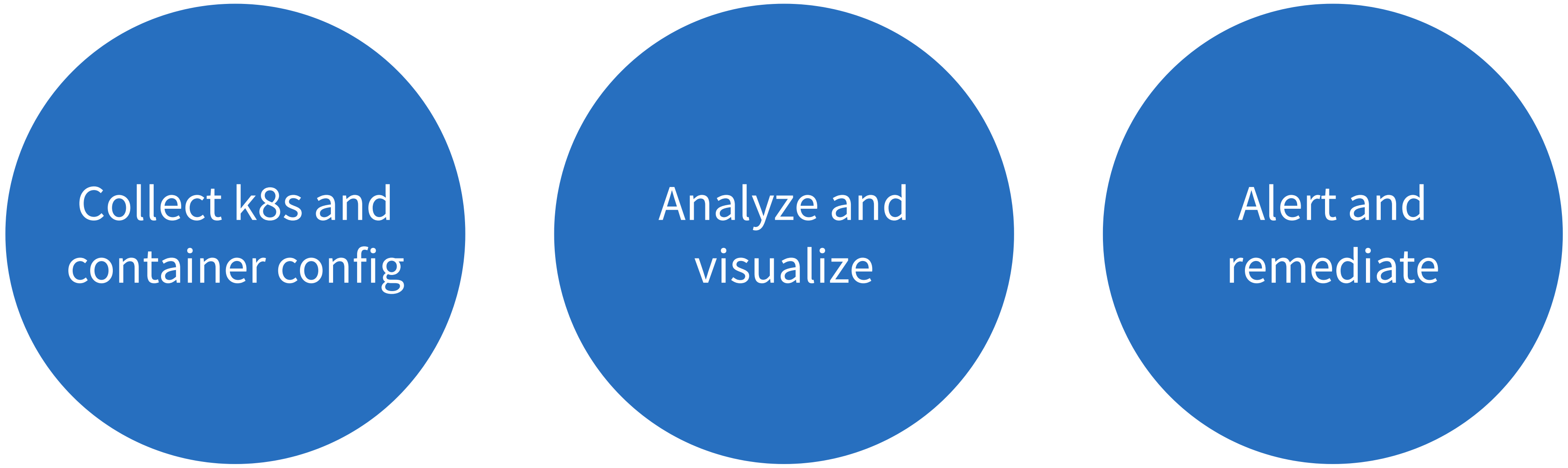
https://www.npmjs.com/package/retire

Posture Management

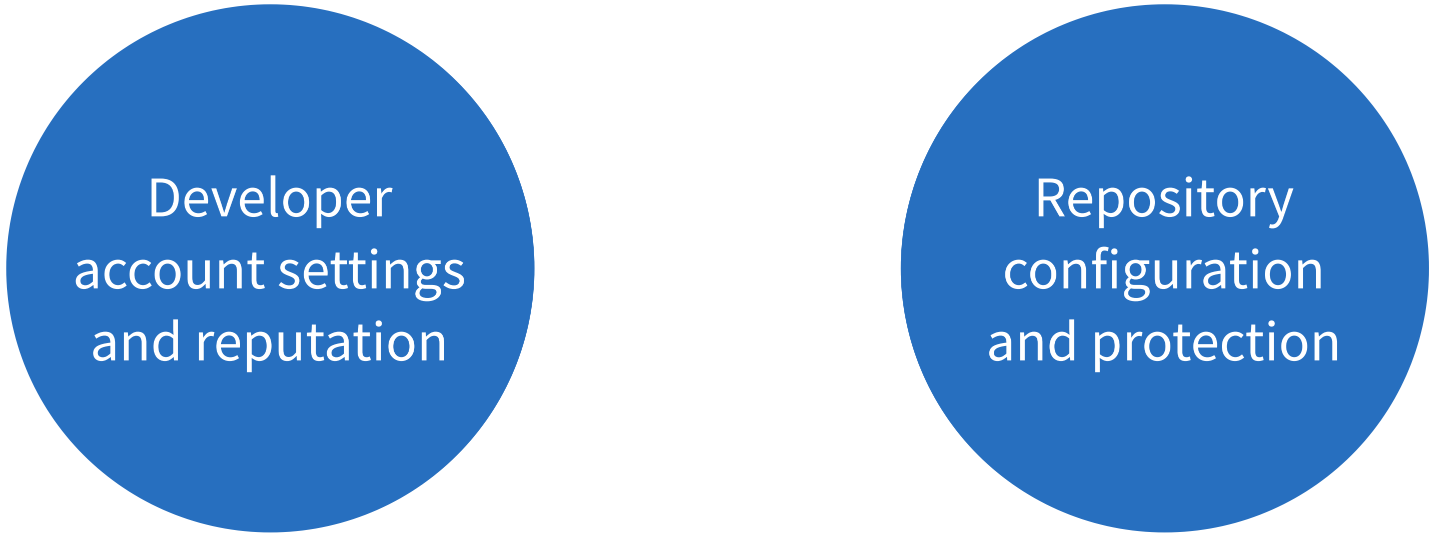
Cloud Security Posture Management (CSPM)



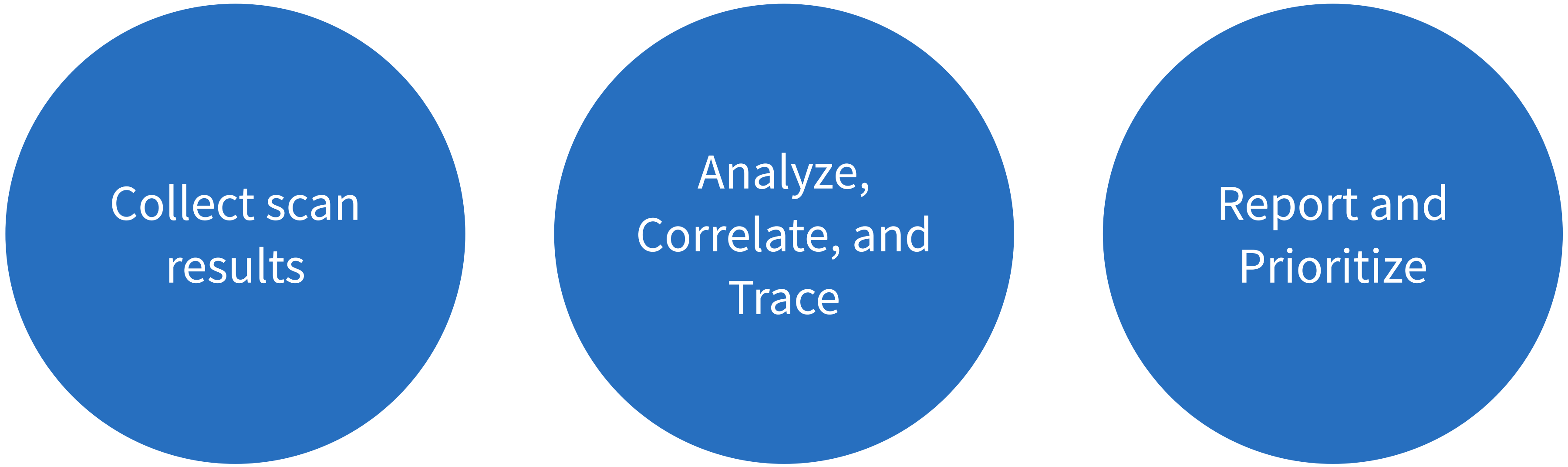
Kubernetes Security Posture Management (KSPM)



Developer/Repository Security Posture Management


Developer account settings and reputation
Repository configuration and protection

Application Security Posture Management (ASPM)



What's next

National Cybersecurity Strategy 2023

STRATEGIC OBJECTIVE 3.3: SHIFT LIABILITY FOR INSECURE SOFTWARE PRODUCTS AND SERVICES

Markets impose inadequate costs on—and often reward—those entities that introduce vulnerable products or services into our digital ecosystem. Too many vendors ignore best practices for secure development, ship products with insecure default configurations or known vulnerabilities, and integrate third-party software of unvetted or unknown provenance. **Software makers are able to leverage their market position to fully disclaim liability** by contract, further **reducing their incentive to follow secure-by-design principles** or perform pre-release testing. **Poor software security greatly increases systemic risk** across the digital ecosystem and leave American citizens bearing the ultimate cost.

We must begin to shift liability onto those entities that fail to take reasonable precautions to secure their software while recognizing that even the most advanced software security programs cannot prevent all vulnerabilities. **Companies that make software** must have the freedom to innovate, but they **must also be held liable when they fail** to live up to the duty...

https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf

The CISA **Secure by Design** pledge

- Increase use of **Multi-factor Authentication**
- Reduce use of **default passwords**
- **Reducing entire classes of vulnerabilities** including - but not limited to - SQL-injection, Cross-Site Scripting (XSS) and memory management issues.
- Increase installation of **Security patches**
- Publish a **Vulnerability disclosure policy**
- Accurate reporting of **CVEs**
- Increase ability for customers to gather **Evidence of intrusions**

https://www.cisa.gov/securebydesign/pledge

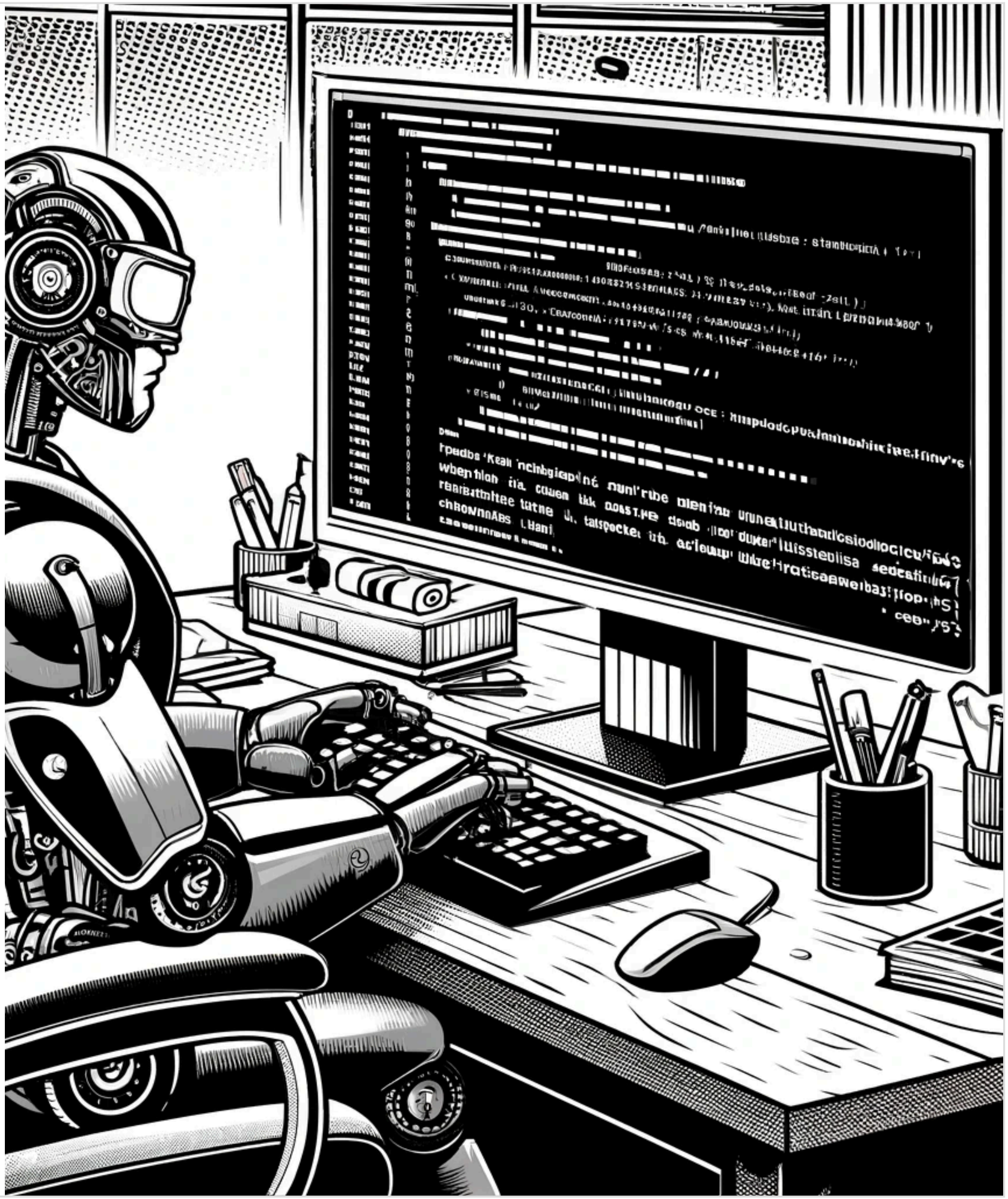
AI

Detecting vulnerabilities with AI

Detecting misconfigurations with AI

Detecting attacks with AI

Fixing code with AI



AppSec for AI applications

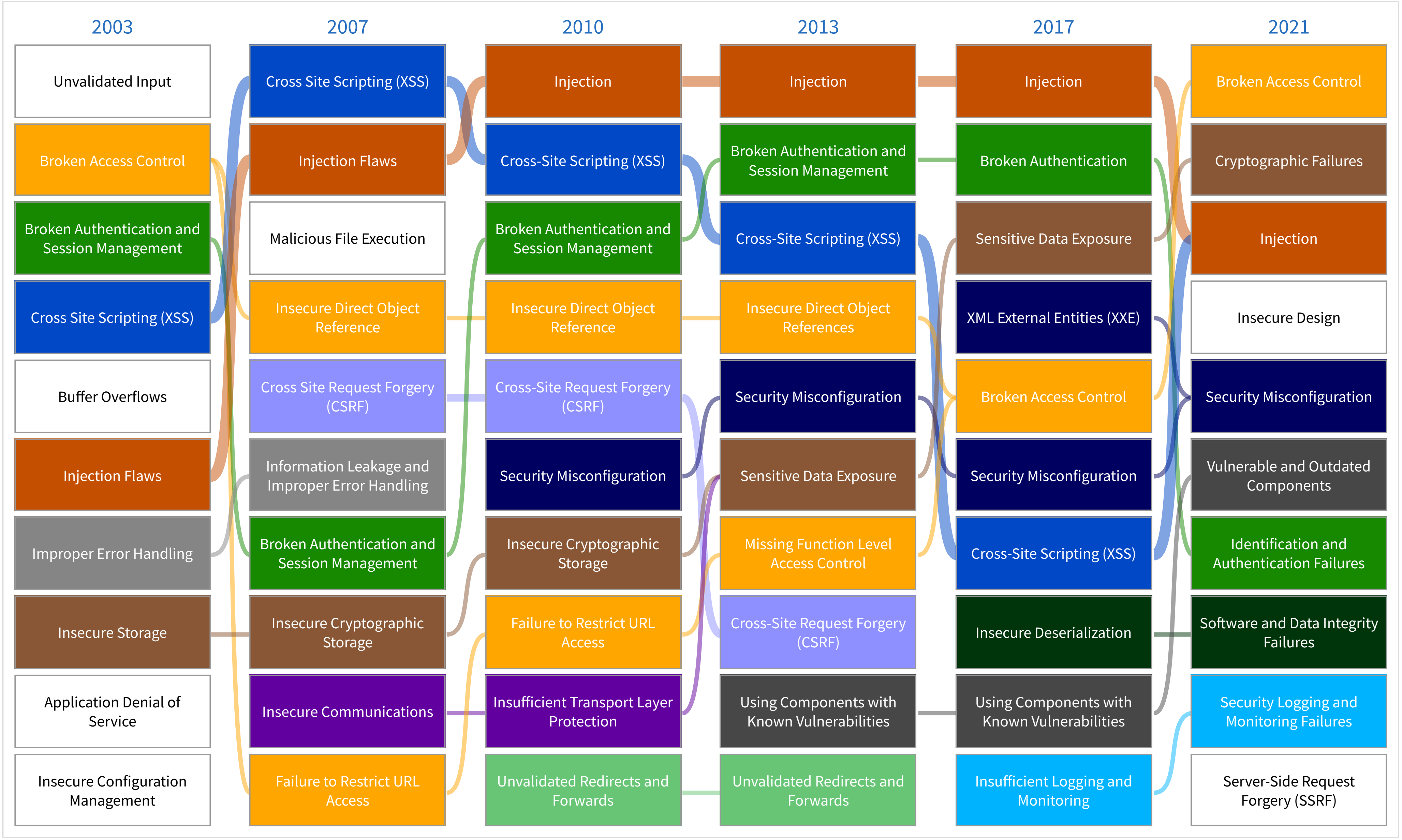
AI doing the wrong thing

In summary

The focus on AppSec is increasing

The AppSec culture is maturing

Interesting times ahead and still a lot to learn



2003
Unvalidated Input
Broken Access Control
Broken Authentication and Session Management
Cross Site Scripting (XSS)
Buffer Overflows
Injection Flaws
Improper Error Handling
Insecure Storage
Application Denial of Service
Insecure Configuration Management
2007
Cross Site Scripting (XSS)
Injection Flaws
Malicious File Execution
Insecure Direct Object Reference
Cross Site Request Forgery (CSRF)
Information Leakage and Improper Error Handling
Broken Authentication and Session Management
Insecure Cryptographic Storage
Insecure Communications
Failure to Restrict URL Access
2010
Injection
Cross-Site Scripting (XSS)
Broken Authentication and Session Management
Insecure Direct Object Reference
Cross-Site Request Forgery (CSRF)
Security Misconfiguration
Insecure Cryptographic Storage
Failure to Restrict URL Access
Insufficient Transport Layer Protection
Unvalidated Redirects and Forwards
2013
Injection
Broken Authentication and Session Management
Cross-Site Scripting (XSS)
Insecure Direct Object References
Security Misconfiguration
Sensitive Data Exposure
Missing Function Level Access Control
Cross-Site Request Forgery (CSRF)
Using Components with Known Vulnerabilities
Unvalidated Redirects and Forwards
2017
Injection
Broken Authentication
Sensitive Data Exposure
XML External Entities (XXE)
Broken Access Control
Security Misconfiguration
Cross-Site Scripting (XSS)
Insecure Deserialization
Using Components with Known Vulnerabilities
Insufficient Logging and Monitoring
2021
Broken Access Control
Cryptographic Failures
Injection
Insecure Design
Security Misconfiguration
Vulnerable and Outdated Components
Identification and Authentication Failures
Software and Data Integrity Failures
Security Logging and Monitoring Failures
Server-Side Request Forgery (SSRF)

Thank you!

All images generated by ChatGPT