

# Epic fails in AppSec: How not to set an AppSec program?

Iqbal Singh

AppSec Day

5<sup>th</sup> Sep 2025

# Who am I?



CyberSec Handyman

Java Developer

AppSec Engineer

Cloud Security

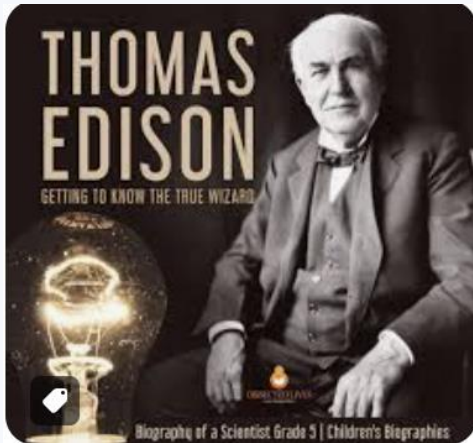
Security Architect

SME AppSec

2010 - 2025

*Upcoming talk: Sh\*t left Security at AISA CyberCon, Australia 15-17 Oct 2025*

# Failures – that become cornerstone of success

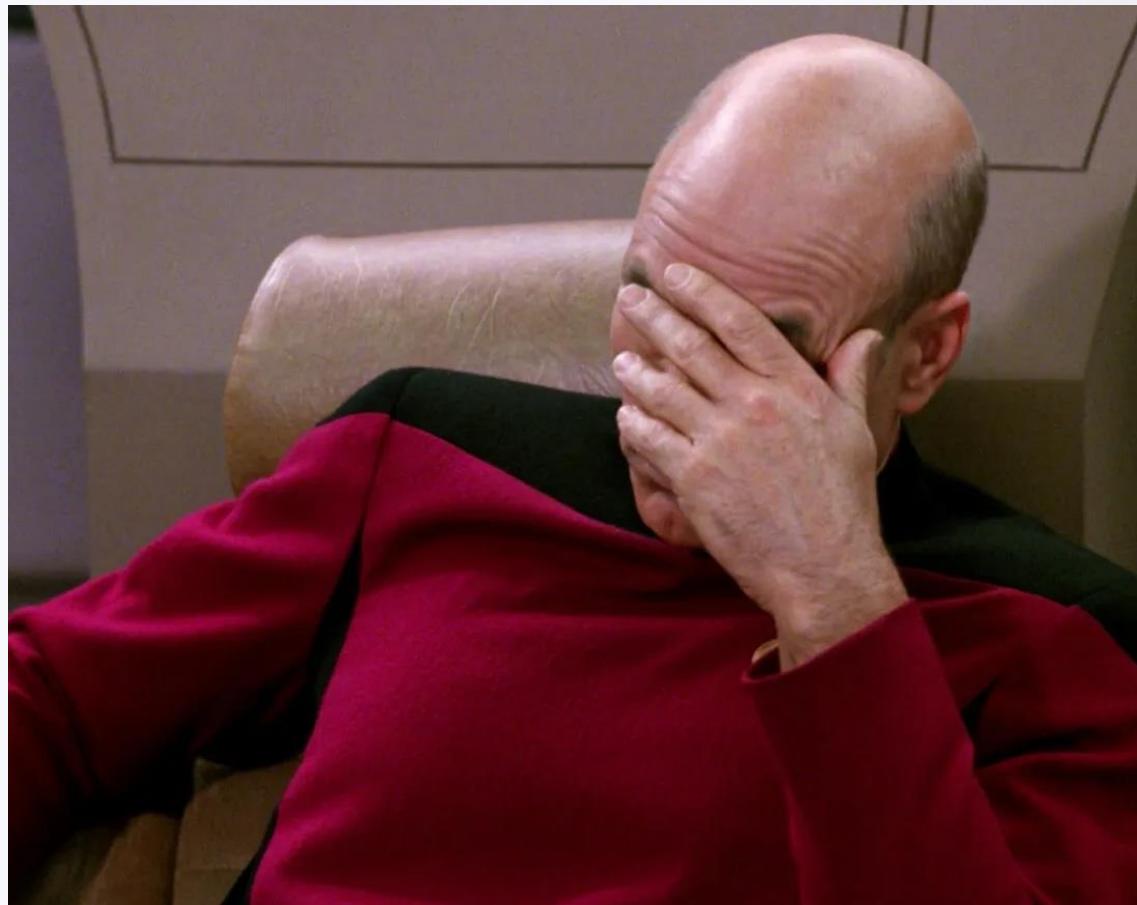


Failed thousands of times inventing the lightbulb, famously saying: “I have not failed 10,000 times—I’ve successfully found 10,000 ways that will not work.”

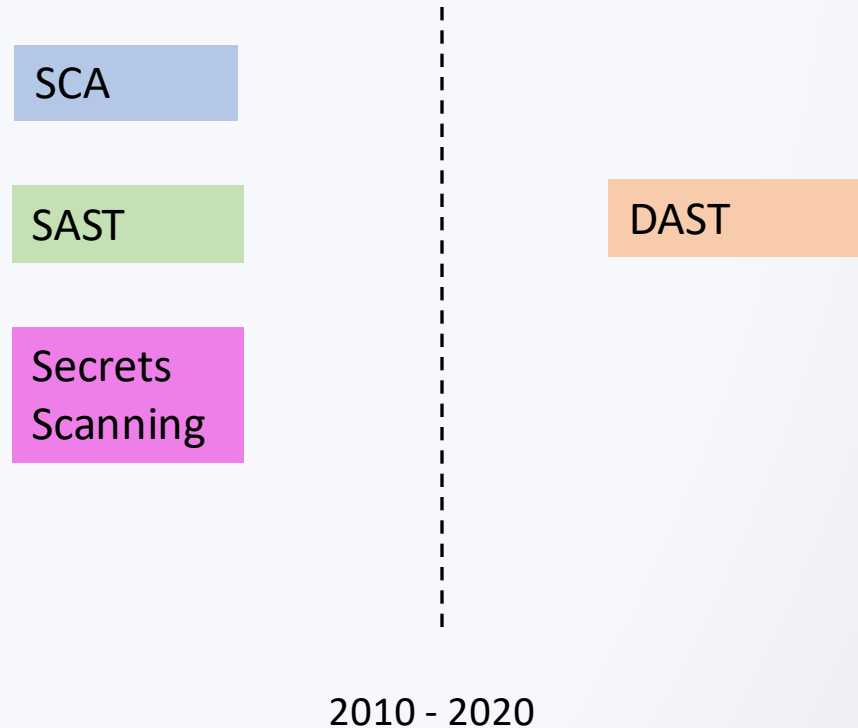
*Failures in CyberSec – Imagine having an incomplete inventory during incidents?*

# Stories

Tooling Chaos  
Scanning only  
CVE Doom Cycle  
AI Impacting SDLC



# AppSec Tooling



# Tooling Chaos

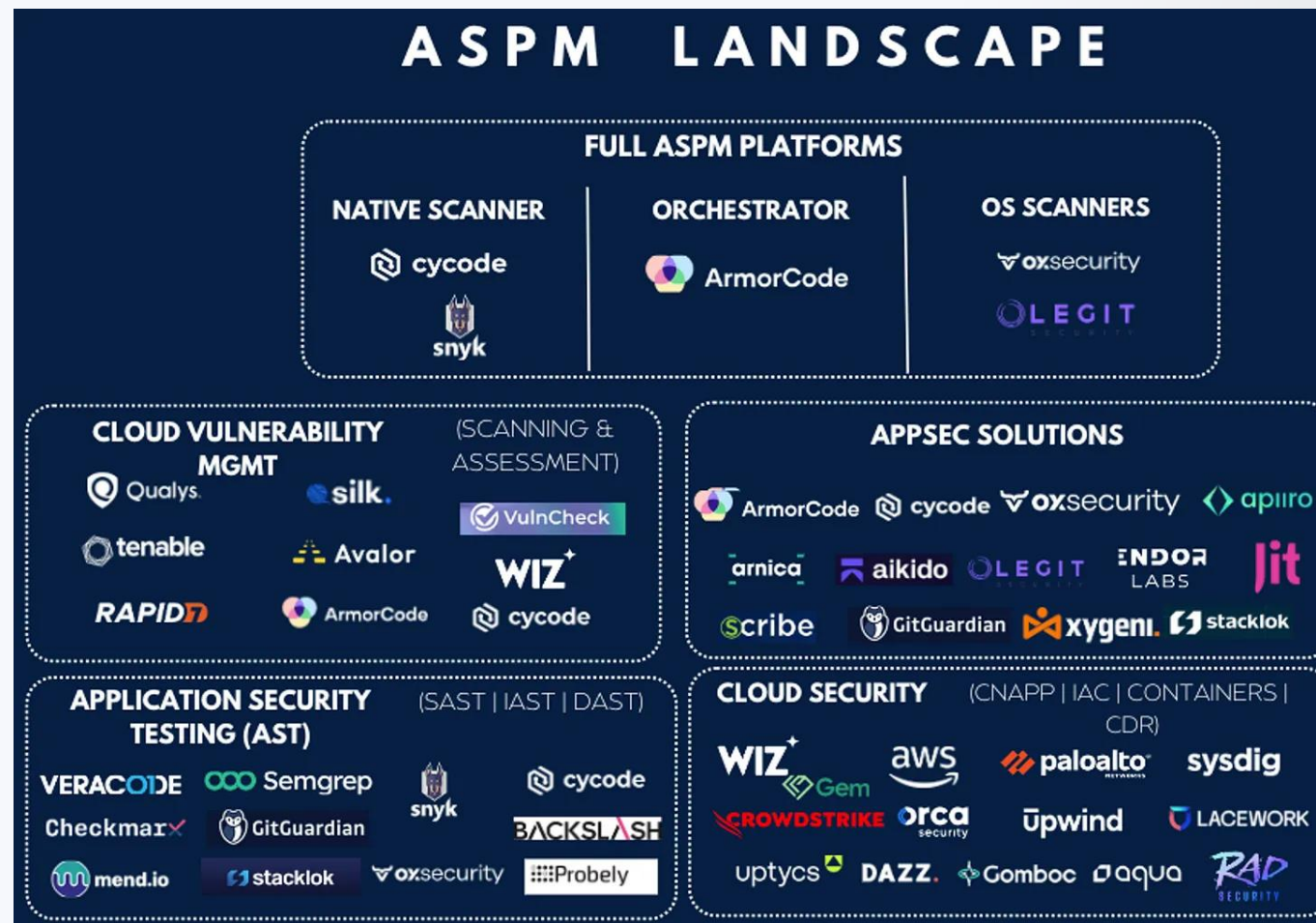
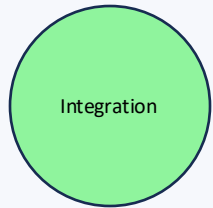


Image source: Resilient Cyber

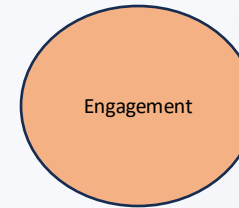
# Tooling Chaos – *scenarios*



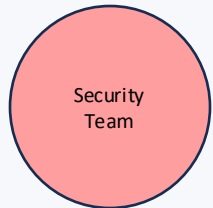
Tools from multiple vendors results in multiple access requests, billing cycles, integrations and training for developers



Excessive tooling results in alert fatigue, no resources to fix



Ineffective tooling creates friction between development & security teams



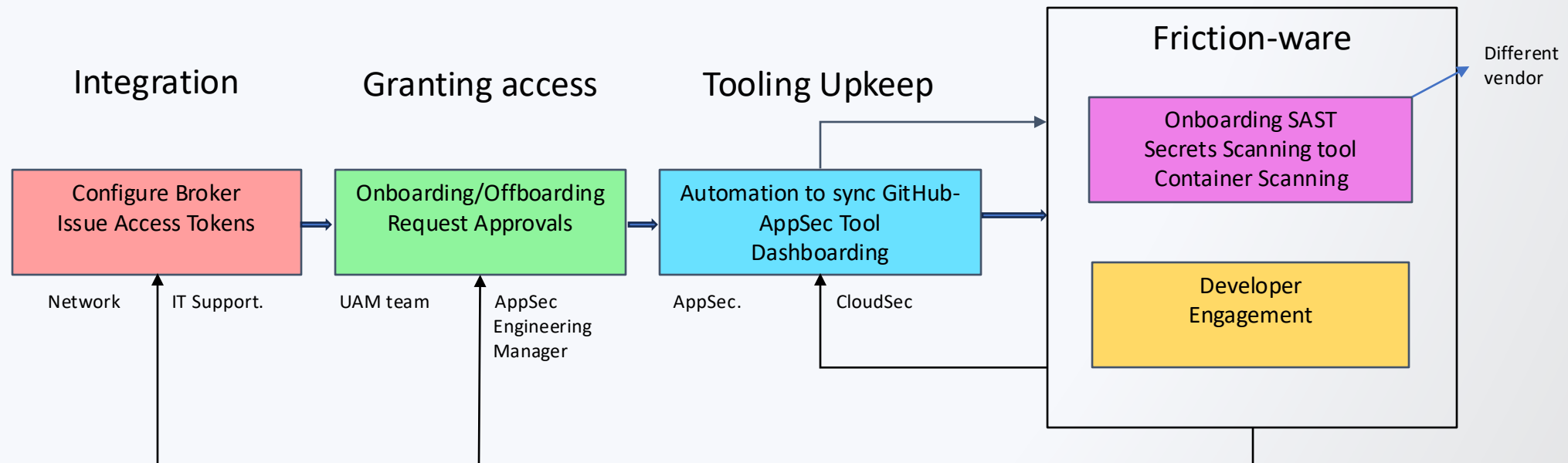
False sense of security - more tools does not always give you better control over attack surface



Those 1 click integrations works well on demo data only

## Let's dive into real scenario

John Smith, Security manager at fintech company procured a top SCA solution, senior management want to burn vulns as fast as possible





# Important things to consider



Simplify  
Onboarding

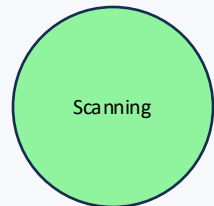


Work with existing vendor

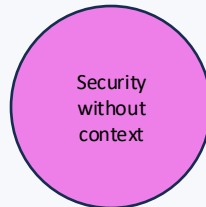


Impact on  
other teams

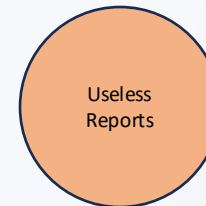
# OnlyScans Approach – *scenarios*



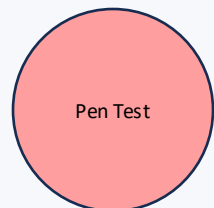
Simply believing that security tooling is a magic bullet and would fix my AppSec vows is guaranteed failure



Security without context is meaning less, sec alerts without application context is meaningless to development teams



One way to become enemy with your development team is to throw automated vulnerability scan reports at them



Penetration testing isn't dying – but still very relevant, AppSec tooling not most of business logic issues



Think about reachability, exploitability, attack graph – it varies with every vendor

# Let's dive into real scenario

There's a mandate from management to enforce AppSec Tooling in block mode to stop security debt accumulation & use auto PRs for upgrades:

## *Fail All builds on new vulnerabilities*

- Block on severity
- Existing CVEs ignored
- Enabled Auto PRs
- Scanning to blocking
- Management is happy  
no more vulns



# Let's dive into real scenario

There's a mandate from management to enforce AppSec Tooling in block mode to stop security debt accumulation & use auto PRs for upgrades:

## *Why Fail All builds on new vulnerabilities backfires*

- Block on severity
- Existing CVEs ignored
- Enabled Auto PRs
- Scanning to blocking
- Management is happy  
no more vulns



**Over-Enforcement** – exemptions, no context

**Baseline Debt** – permanent risk

**Auto PRs myth** – upgrading is hard

**Tool != solution** – scanning not security program

**Back to square one** – what now?

# Another scenario - SAST

*Remember SAST will find issue if insecure line of code exists only*

```
app.post('/api/logout', (req, res) => {  
  // Check if a session exists and if a user is logged in.  
  if (req.session && req.session.userId) {  
    req.session.userId = null;  
    req.session.username = null;  
  
    req.session.save(err => {  
      if (err) {  
        console.error("Error saving session:", err);  
        return res.status(500).json({ message: 'Could not log out, please try again.' });  
      }  
      // Send a success response.  
      res.status(200).json({ message: 'User logged out successfully (session preserved).' });  
      console.log('User data cleared from session.');    });  
  } else {  
    res.status(200).json({ message: 'No active user to log out from.' });  
  }  
});
```

# Another scenario - SCA

AppSec team has been asked to help investigate a zero-day vulnerability for an affected library

## *Scanning during Incidents*

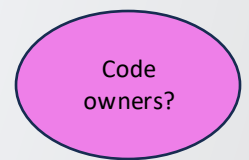
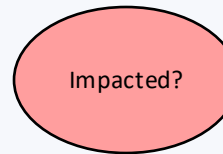
- SCA tool detected multiple instances of affected library in GitHub code
- SCA tool couldn't tell whether affected library is running in prod
- Runtime EDR showed few containers running affected library
- No effective way to connect code to cloud

# Another scenario - SCA

AppSec team has been asked to help investigate a zero-day vulnerability for an affected library

## *Scanning during Incidents*

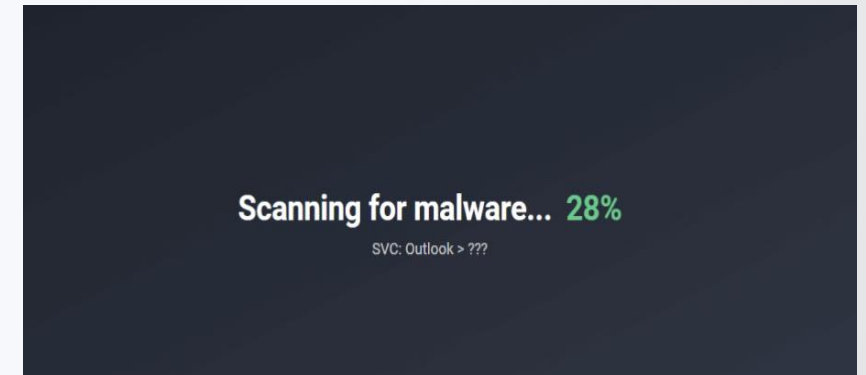
- SCA tool detected multiple instances of affected library in GitHub code
- SCA tool couldn't tell whether affected library is running in prod
- Runtime EDR showed few containers running affected library
- No effective way to connect code to cloud



# Important things to consider



Pace of AppSec Program



Look beyond Scanning



# Important things to consider



Balance security with velocity



Plan for existing issues

# CVE Doom Cycle

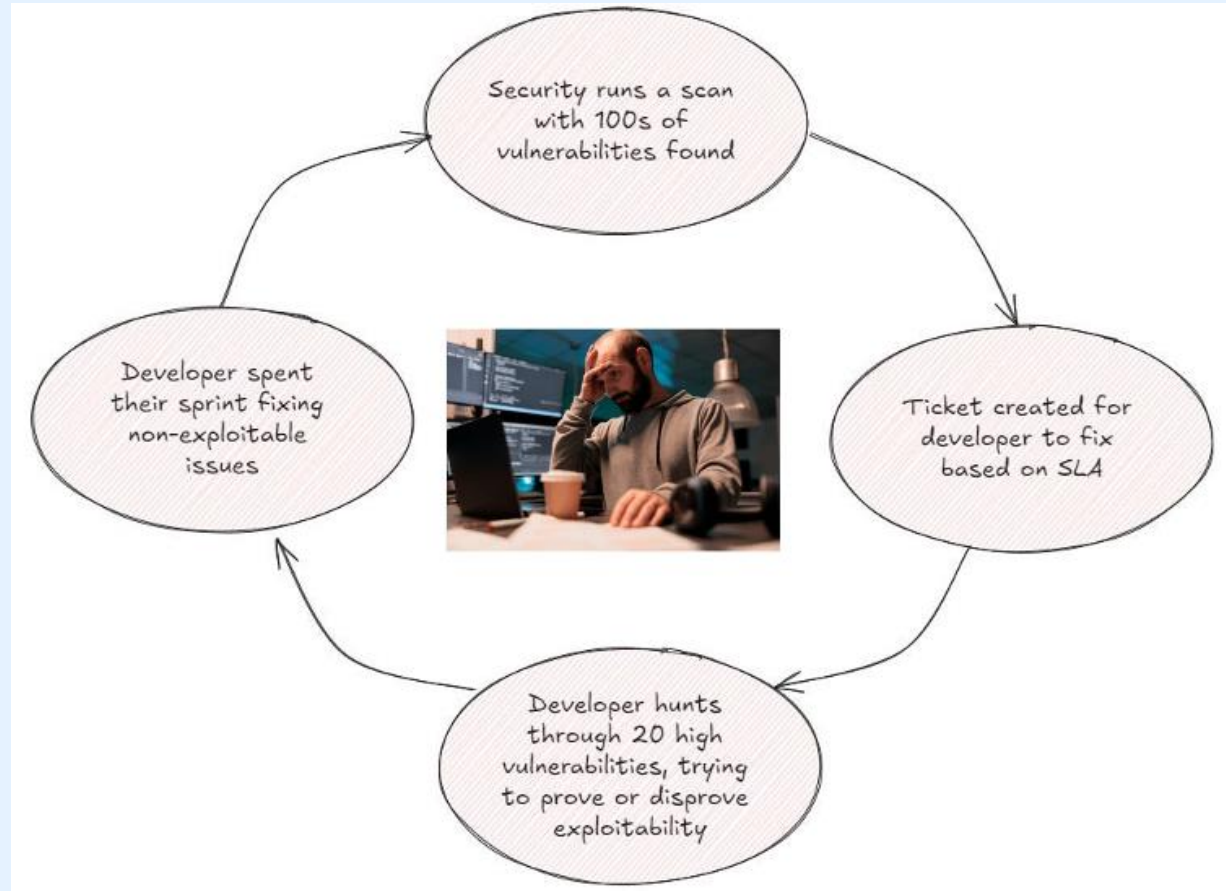



Image Source: Latio Pulse

# CVE Doom Cycle – *how people fall into it?*

## Buy a scanning tool

- Have a complete Inventory
  - Scan code for vulnerabilities
  - Patch Management (Devs)
  - Be Compliant
  - Meet SLAs
- 
- Repeat

# CVE Doom Cycle – *getting out?*

## Buy a scanning tool

- Have a complete Inventory
- Stop Exploits
- Patch Management
- Be Compliant
- Meet SLAs



## Proactive approaches

- Paved road approach
- Distroless container images
- Golden AMIs
- Service Control Policies
- Tagging Policies

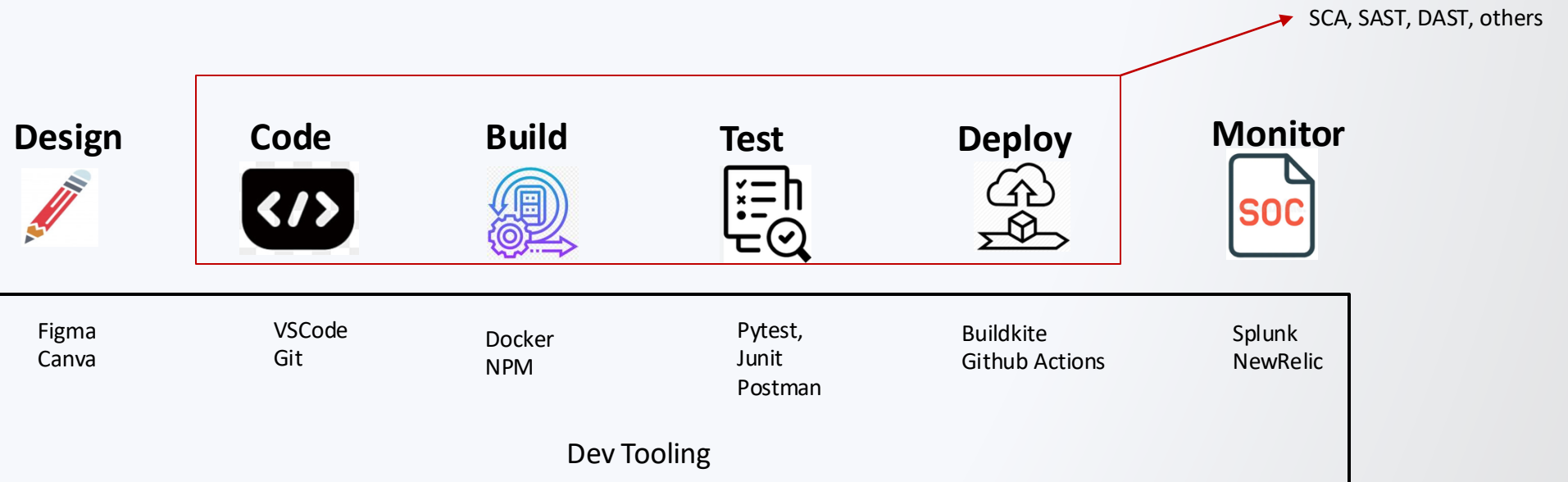
## CVE Doom Cycle – *getting out (be careful)*

- **Get the basics right** before exploring distro-less images—ensure a solid CI/CD platform with regular (nightly/weekly/monthly) deployments
- **Golden images** are valuable, but only if you have resources to maintain them
- **Paved road approaches** often break down over time in small/mid-sized companies.
- **Policy enforcement** (tagging, SCPs, OPA, etc.) must be **frictionless** to succeed

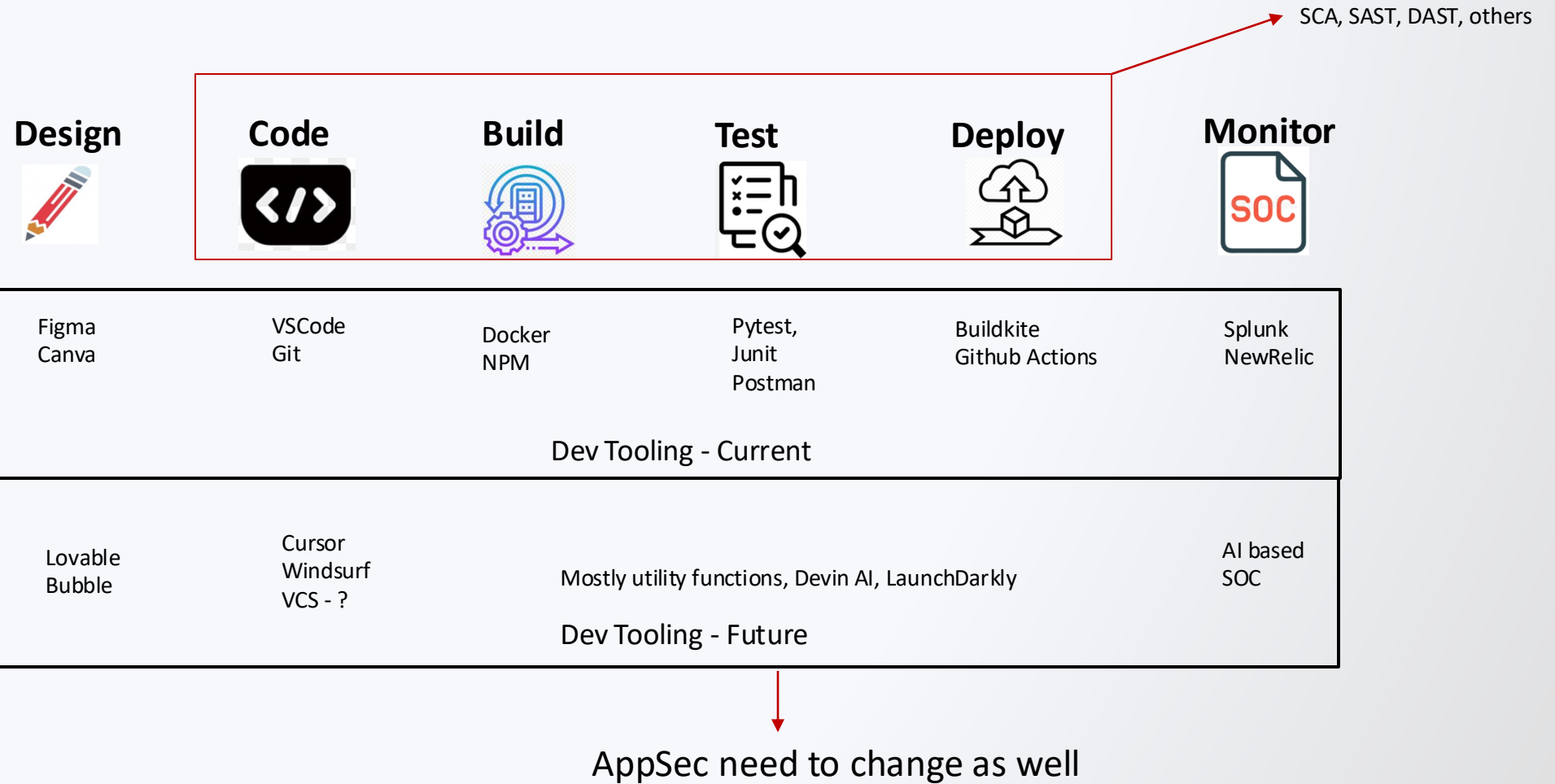
# Vibe Coding

Tech isn't the threat – security is the challenge. Yesterday it was cloud; today, it's AI

# SDLC



# AI SDLC





# AI Powered Development - challenges



Not same level of success as code generation

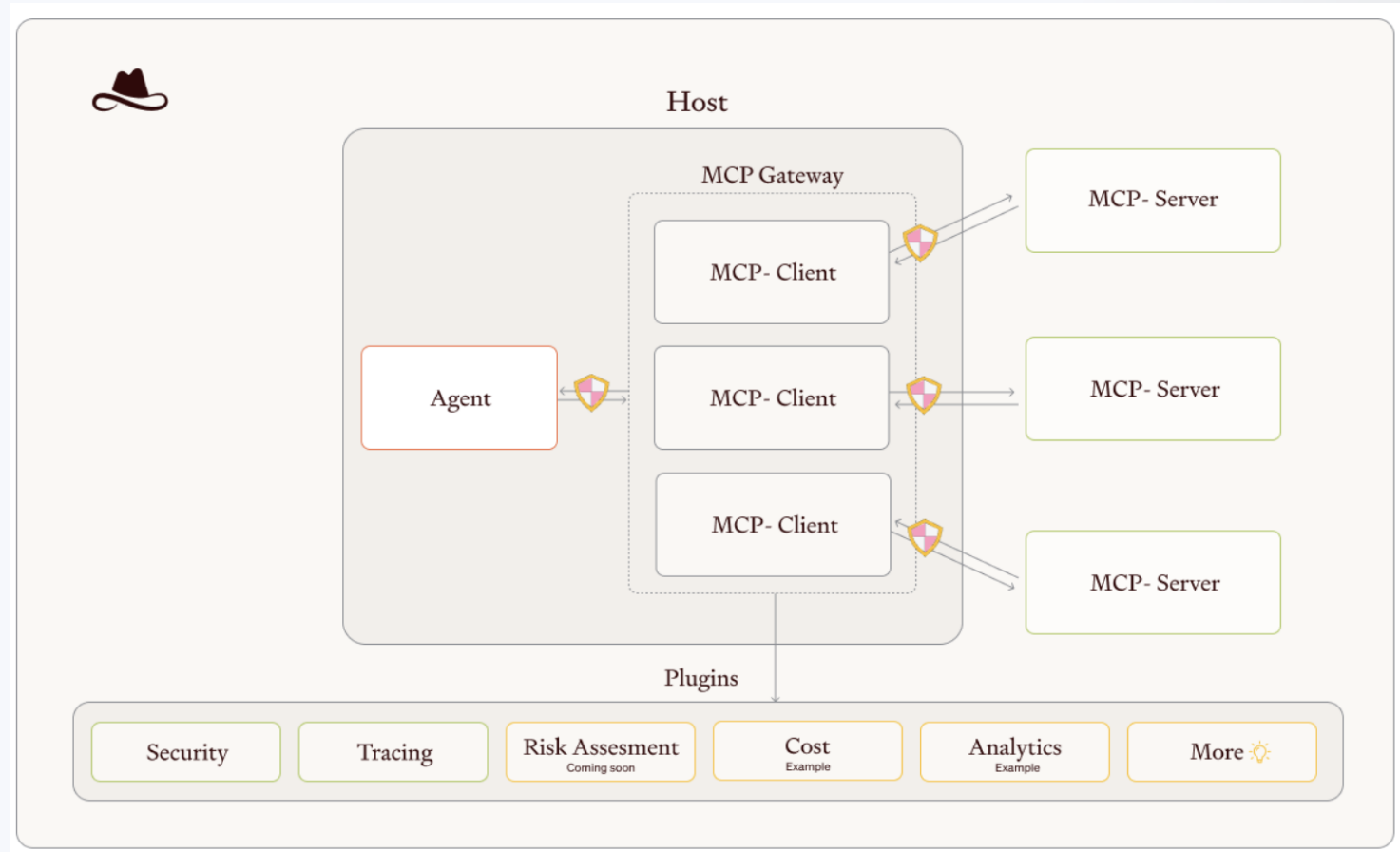


Pull request fatigue



Secure vibing is big challenge

# Consider MCP Gateways



Source: Lasso Security

# Other Stories

- Running AppSec program as a project
- Changing AppSec tooling thinking it'll change my fortune
- Raising JIRA ticket for every alert and realizing BA is required to triage after just 1 week
- Security Metrics present wrong picture

