Pixee

Who’s going to secure the code our army of robots is going to be writing?

Arshan Dabirsiaghi
CTO, Pixee.ai

Hi, I’m Arshan

20 years experience in software security

Code reviews, threat modeling, pen. testing for F100

Spoken at BlackHat, OWASP, BlueHat, others

Authored multiple CVEs, OSS security tools

Co-founded a security unicorn

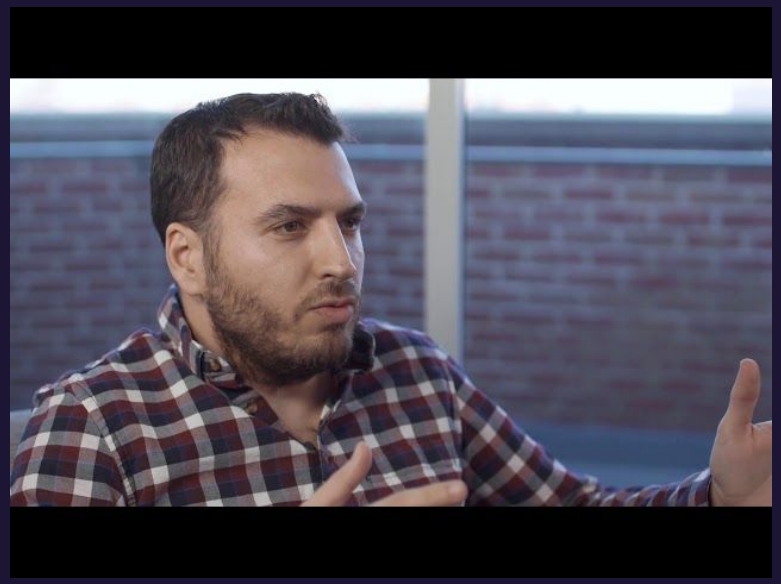
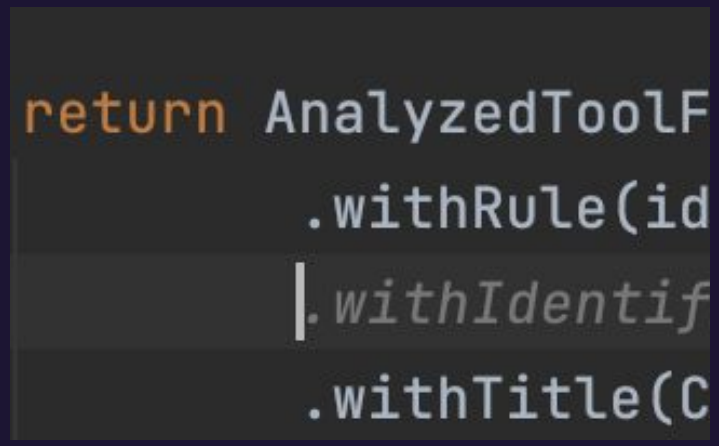


Figure 1: Me absolutely hating making slides like these

The Army of Robots Is Coming

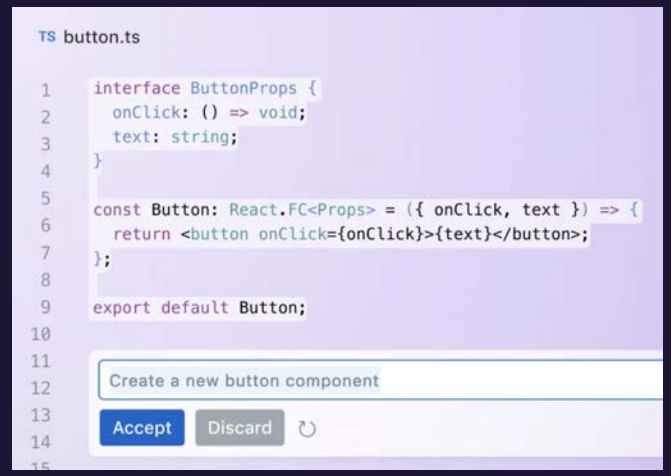
Fill in the middle (autocomplete)



Good adoption (1M+)

25-60% more throughput

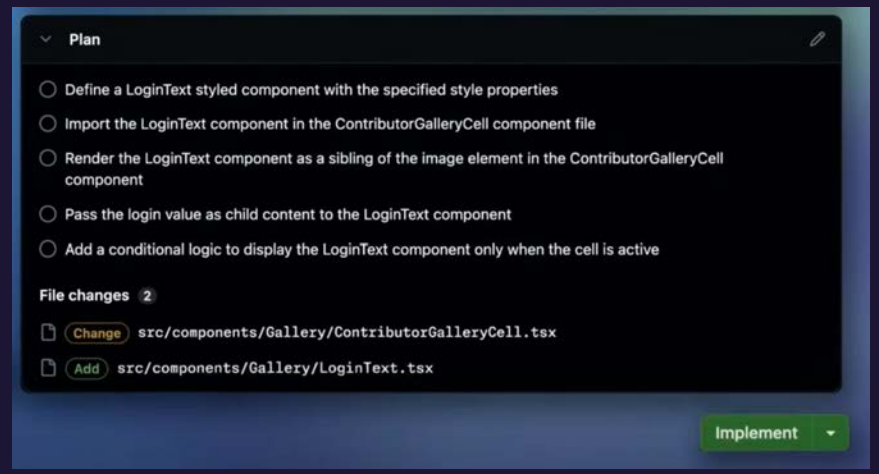
Assistant (code drafting)



Growing adoption (~200K)

?? 100%?

Unguided (full feature development)

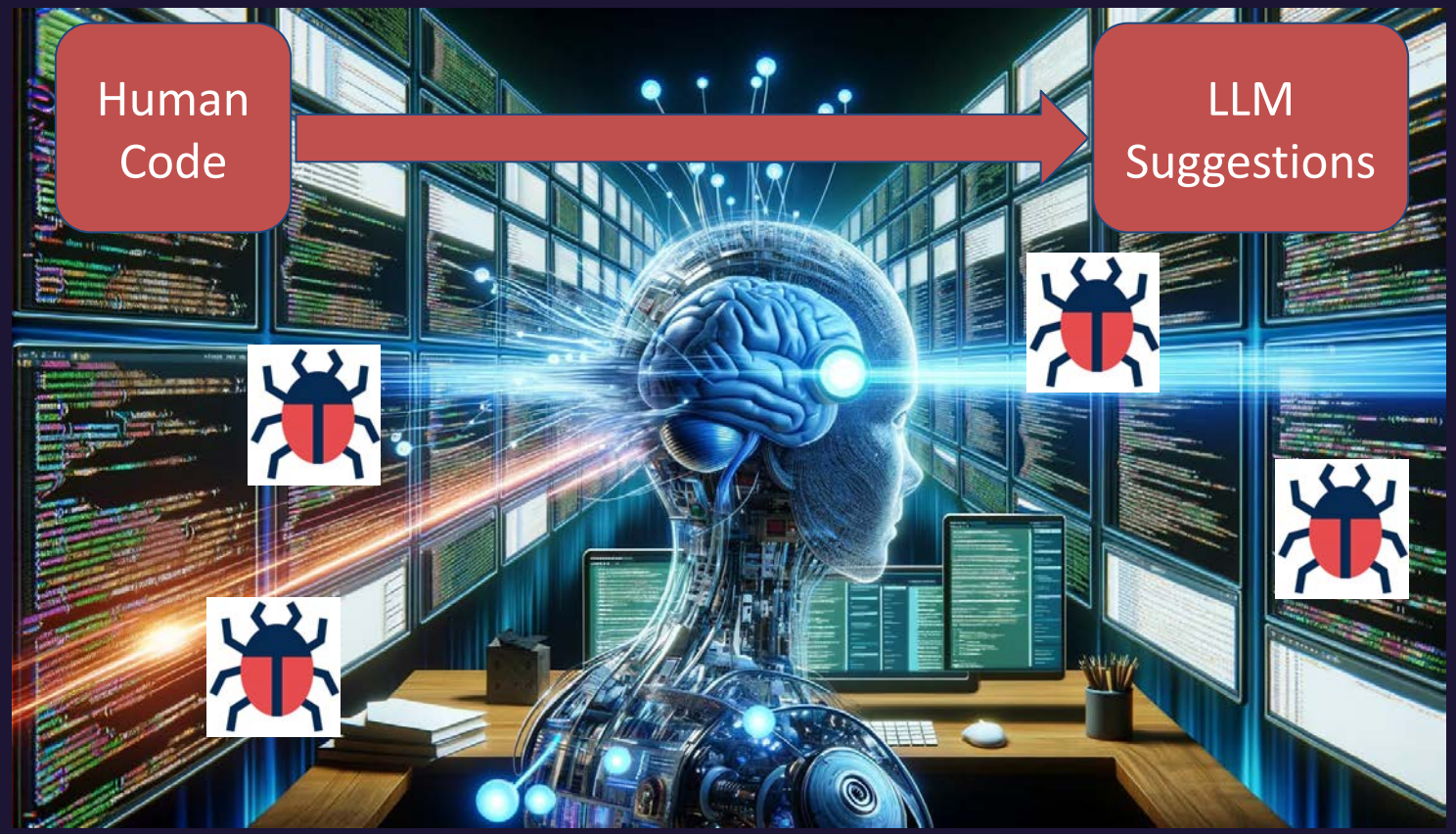


Not publicly released yet

?? 500%?

Sources: McKinsey, Microsoft, MIT

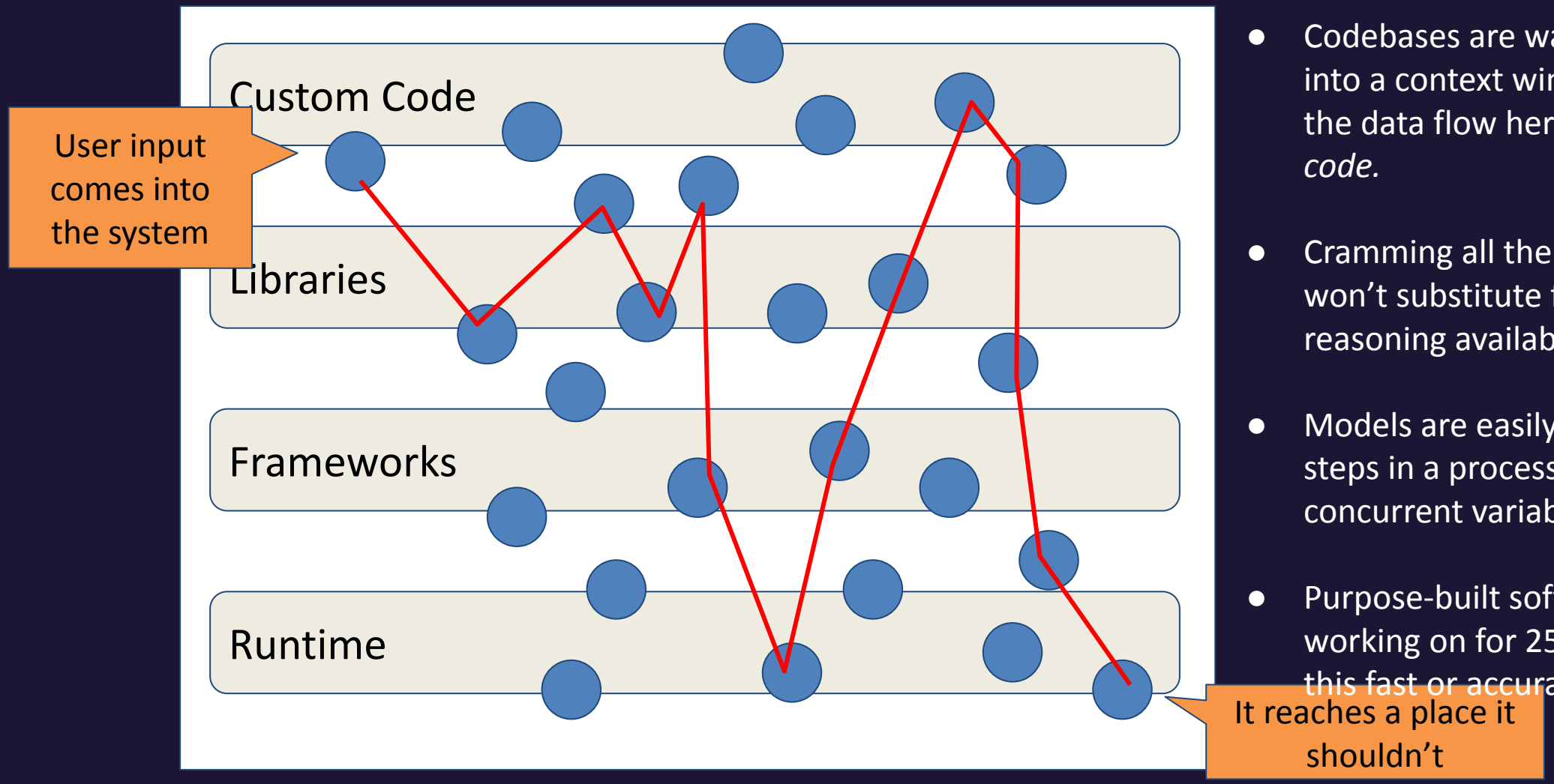
LLMs Write Insecure Code And Then Devs Believe It Isn't



- "Significantly more likely to provide an insecure solution ($p < 0.05$)"
- "...given 89 scenarios, about 40% of the computer programs made with the help of Copilot had potentially exploitable vulnerabilities."
- "participants provided access to an AI assistant were more likely to believe that they wrote secure code than those without access to the AI assistant"

Sources: NYU, Stanford

Can’t the Models Just Generate Secure Code?



- Codebases are way, way too big to fit into a context window. And most of the data flow here *isn’t even in your code.*
- Cramming all the code to embeddings won’t substitute for complicated reasoning available in the context.
- Models are easily confused by more steps in a process and more concurrent variables in play.
- Purpose-built software we’ve been working on for 25 years can’t even do this fast or accurately.

Secure Software Processes Are Very Manual

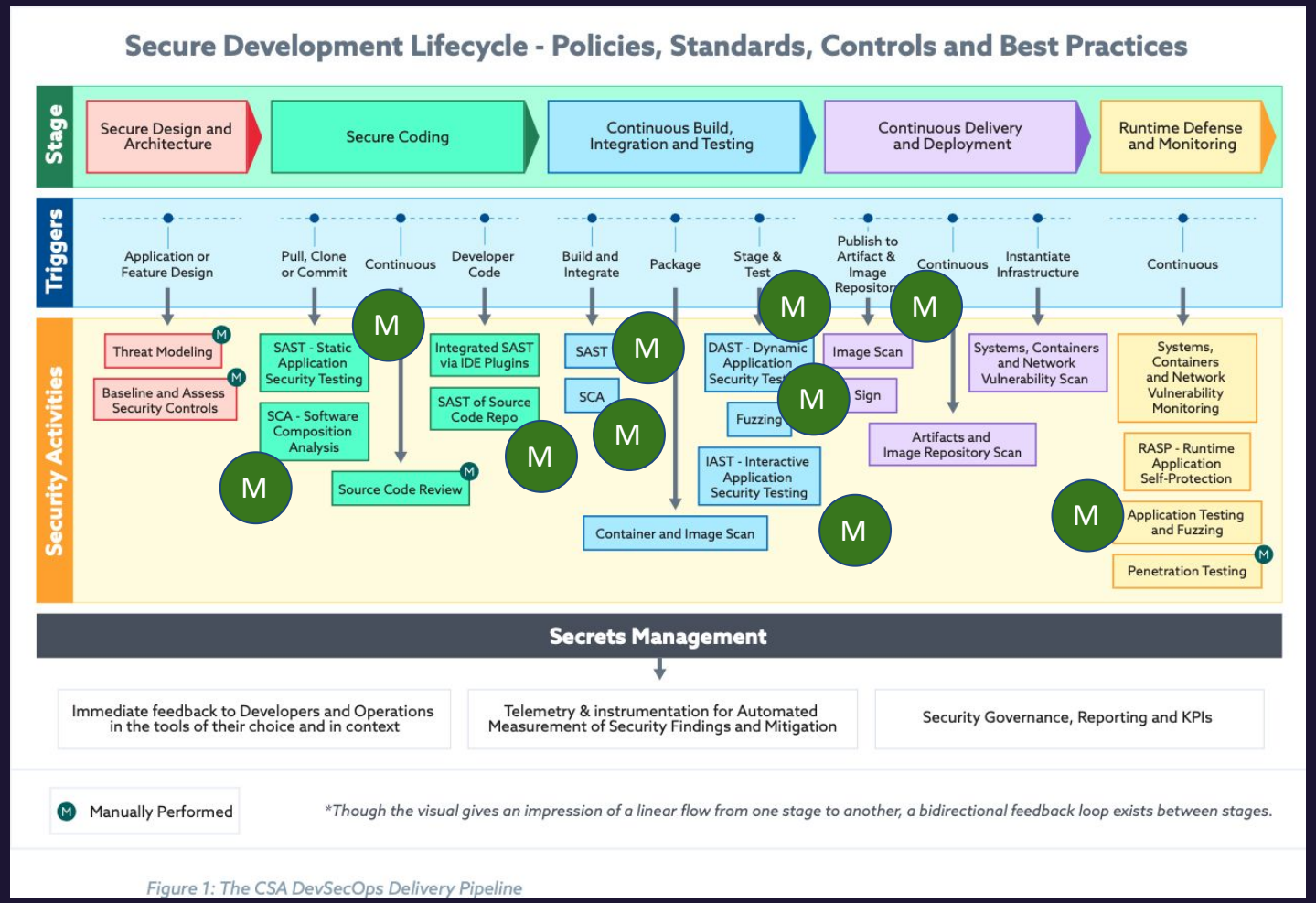


Figure 1: The CSA DevSecOps Delivery Pipeline

The factory requires constant human intervention:

- Triaging results from tools
- Fixing things tools find
- Ticket management
- CYA documentation
- Product tradeoffs

Across these disciplines:

- Risk management
- Software engineering
- Product management
- Compliance
- Security engineering

Source: PagerDuty

Limitations of Our Security Programs Today

Not Enough Humans

- Developers outnumber security 100:1 (my experience is this is drastically worse, the bigger the company)

The Humans We Have Aren't Cross-Skilled

- Security personnel many times don't have hands-on coding skills to pitch in directly or review
- Developers don't have good security skills to efficiently and accurately triage

Reality

- AppSec typically runs many activities only on the most critical applications (internet facing w/ sensitive assets)

Sources: Sonatype

What Can Scale With The Robots?

Solution: Paved Roads

Strategy: <u>Make It Hard To Be Insecure</u>

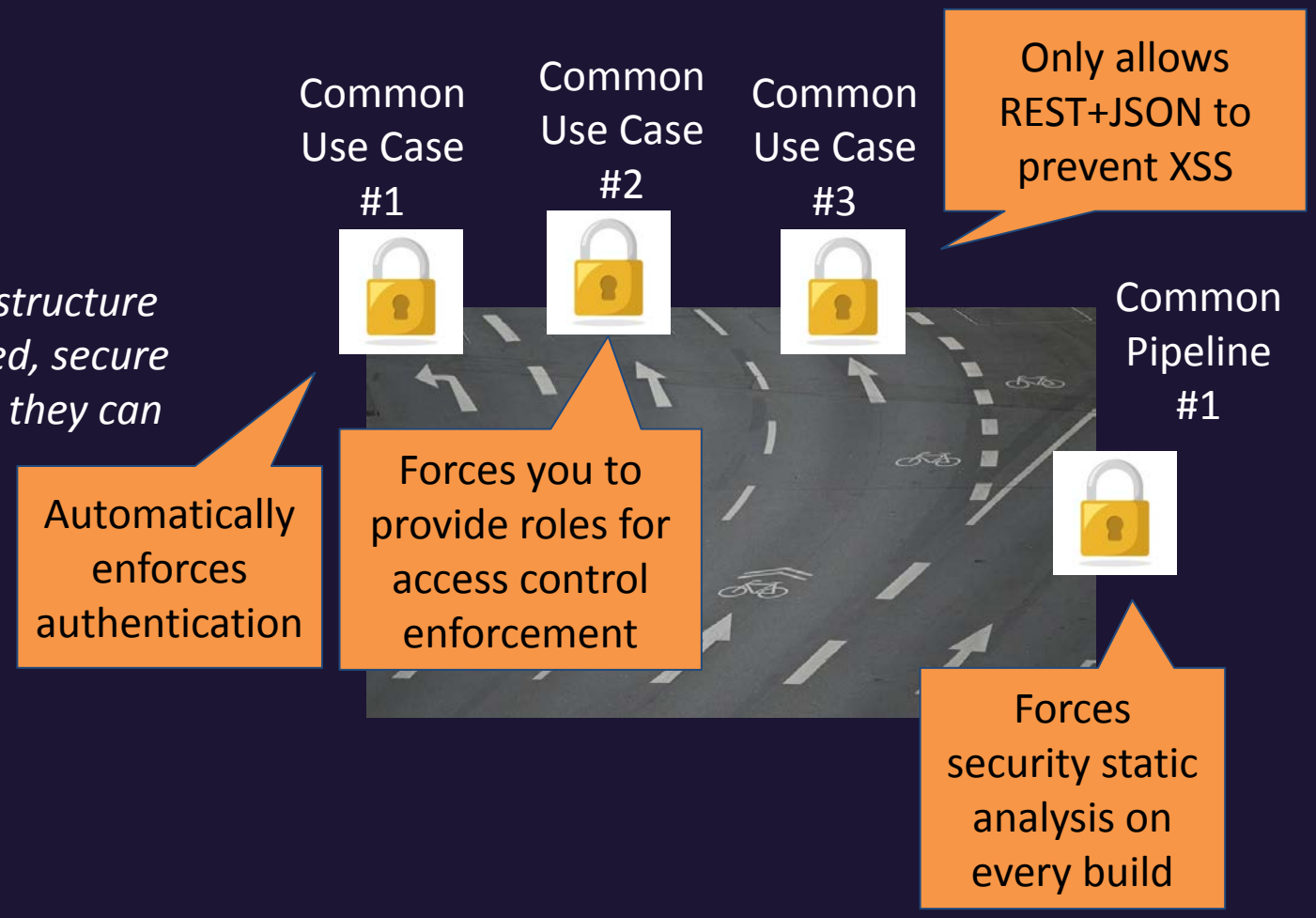
*"Netflix engineering invests in the concept of an Infrastructure and Security **Paved Road**. This provides well-integrated, secure by default central platforms to engineers at Netflix so they can focus on delivering their core business value"*

Requirements

- Strong DevEx / platform teams
- Fewer technology stacks
- Developer Security champions

Help in this area

- Resourcely (vendor)
- BridgeCrew (vendor)
- Spinnaker (OSS tool)

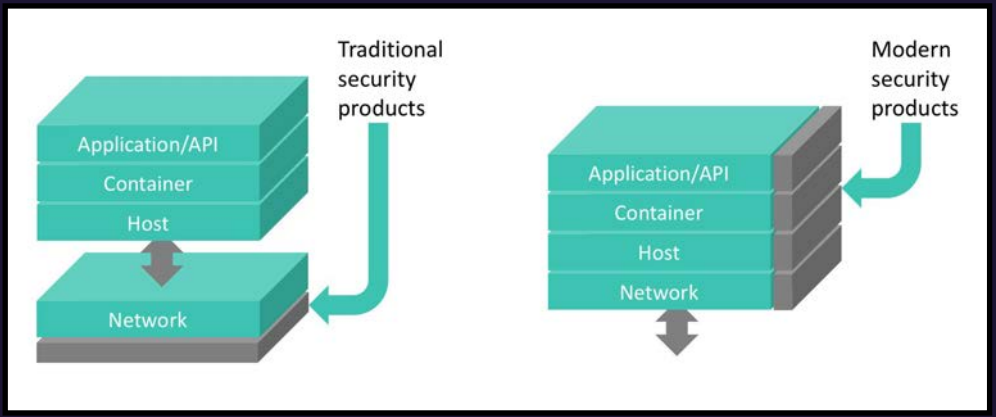


Sources: Netflix

Solution: Better Runtime Protection (with RASP)

Strategy: <u>Make It Hard To Exploit Your Insecure Code</u>

"Traditional security measures are not equipped to deliver protection in the cloud, which means that organizations must craft a new strategy and adopt new tooling, including application-level policies, tools, technologies and rules — chief among them RASP."



Help in this space:

- Contrast Security (vendor)
- DataDog (vendor)
- Imperva (vendor)
- AppDynamics (vendor)

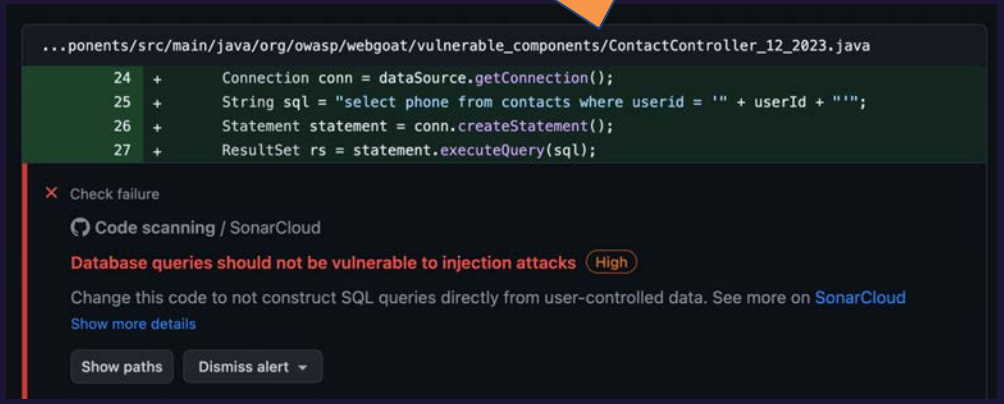
Sources: Contrast Security, Crowdstrike

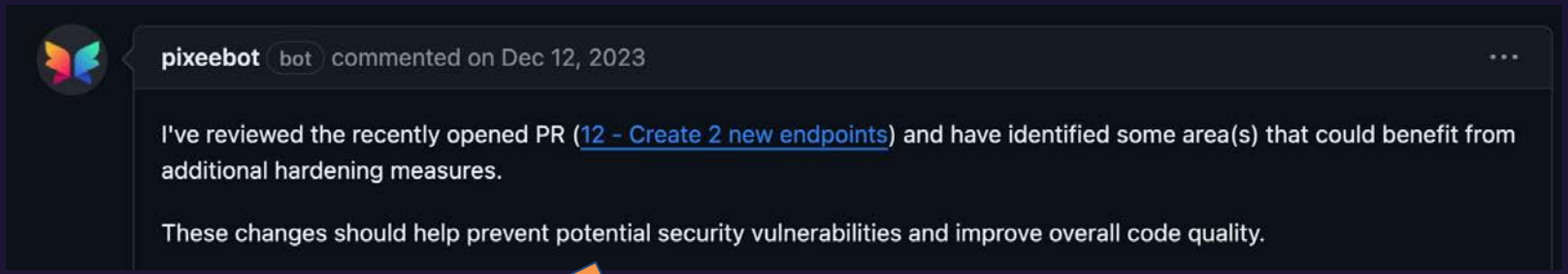
Solution: Security Tool Copilot

Strategy: <u>Eliminate Human Interruptions for Security Tools</u>

The highest spend in secure development is also the one that has the hardest skill to find – triaging and fixing security tool results.

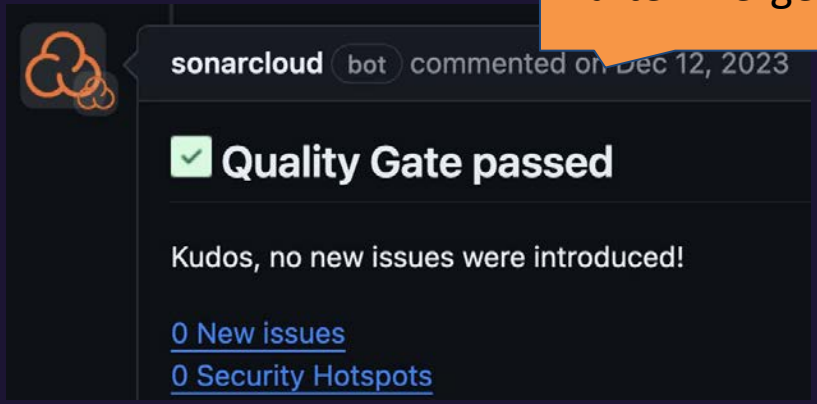
1. Scanner finds something





2. Security Copilot triages and proposes the fix

3. Scanner finds nothing after merge

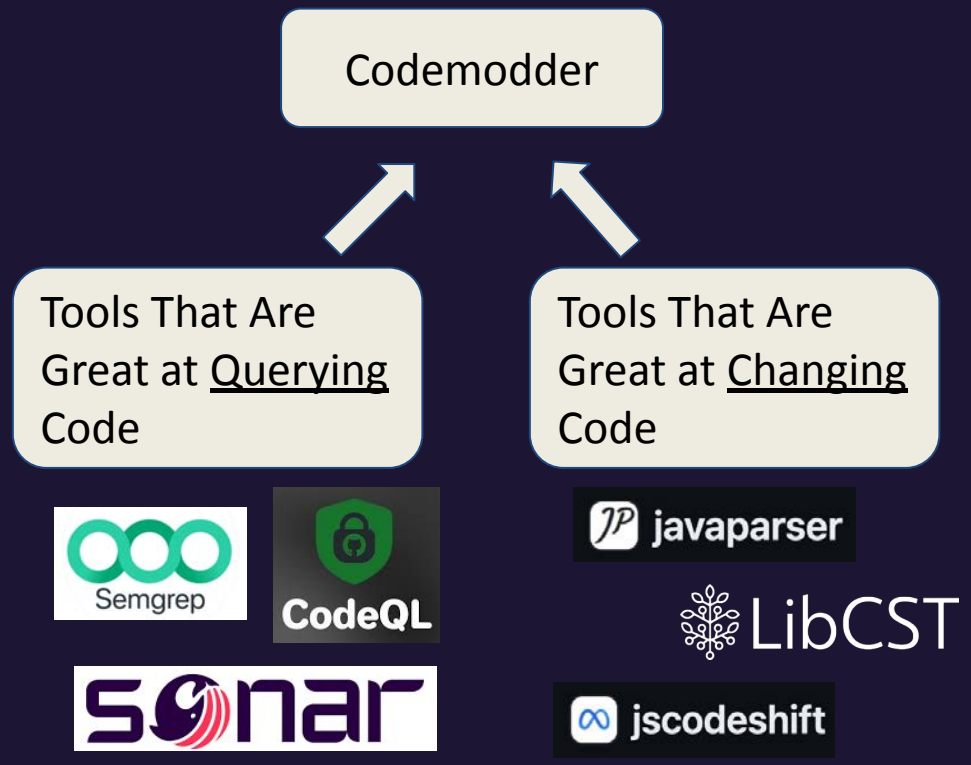


Help in this space:

- Pixee (vendor - me)
- Corgea (vendor)
- renovate (vendor)
- dependabot/renovatebot (vendor)
- Codemodder (OSS library)

Codemodder: A modern, OSS codemod library

A codemod library focused on orchestrating great tools together.




```
class SecureRandom(SemgrepCodemod):
    NAME = "secure-random"
    REVIEW_GUIDANCE = ReviewGuidance.MERGE_WITHOUT_REVIEW
    DESCRIPTION = "Replaces random.{func} with more secure secrets library functions."

    @classmethod
    def rule(cls):
        return """
        rules:
          - patterns:
            - pattern: random.$F(...)
            - pattern-inside: |
                import random
                ...
        """

    def on_result_found(self, original_node, updated_node):
        self.remove_unused_import(original_node)
        self.add_needed_import("secrets")
        return self.update_call_target(updated_node, "secrets.SystemRandom()")
```

https://codemodder.io/
https://github.com/pixee/codemodder-python
https://github.com/pixee/codemodder-java
https://github.com/pixee/cli

Thank You!

@nahsra

arshan@pixee.ai

https://pixee.ai

https://github.com/apps/pixeebot