
SNOWFROC

Climbing App Sec Mountains (and how to summit)

Adam Schaal
March 5, 2020

CONTRAST SECURITY

$ whoami

SNOWFROC

```
class Speaker:
    def __init__(self):
        self.name = “Adam Schaal”
        self.title = “Principal Application Security Researcher”
        self.company = “Contrast Security”
        self.twitter = “@clevernyyyy”
        self.website = “adamschaal.com”
        self.hobbies = [
            “Family Time”,
            “Homelab / Home Automation”,
            “Kernelcon Board Member”,
            “CTF Competitions”
        ]
```

Inspiration

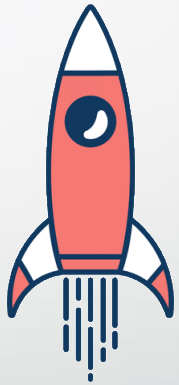
Past work experiences

Berkshire Hathaway Insurance Company

World-wide Data Provider

Cable Services Conglomerate

Current Workplace



Startup Culture

~150 employees when I was hired


Really?
Tell me more about it!

Meet Ned

- Ned is an Application Security Engineer
- Ned works for a Large Enterprise
- Ned is starting an Internal Application Security Team at Large Enterprise
- Ned is happy his company is prioritizing Application Security

Meet Ned

@AppSecNed
#AppSecNed

Ned's Directives

- Ensure all current codebases are secure
  - Including all Open Source Software utilized
- Ensure all future development is secure
  - Shift Left with future development
- Train software developers in secure coding practices
- Minimize attack vectors in external facing applications

Ned's Challenges

2 out of **3** Applications FAIL to pass initial tests on the OWASP Top 10 and SANS 25 standards

171 days Average time to remediate application vulnerability

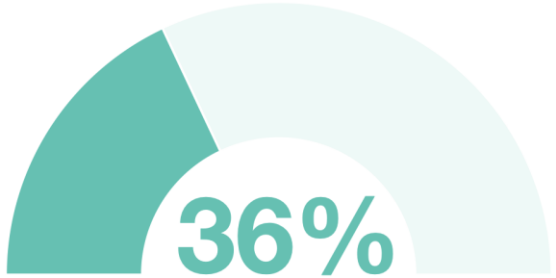
1 to **73** One Software Security Specialist to Every 73 Software Engineers and Developers

Sources: BSIMM and Veracode

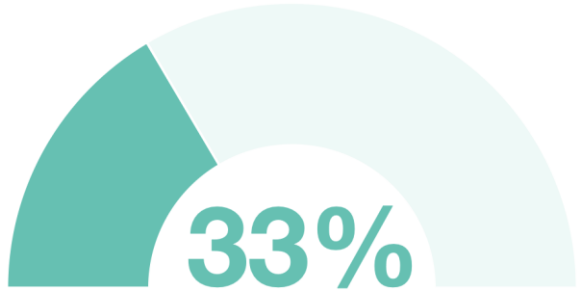
Ned's Challenges

All of the most common vulnerabilities of the last five years are still just as prevalent today.

Sensitive Data Exposure



Security Misconfiguration



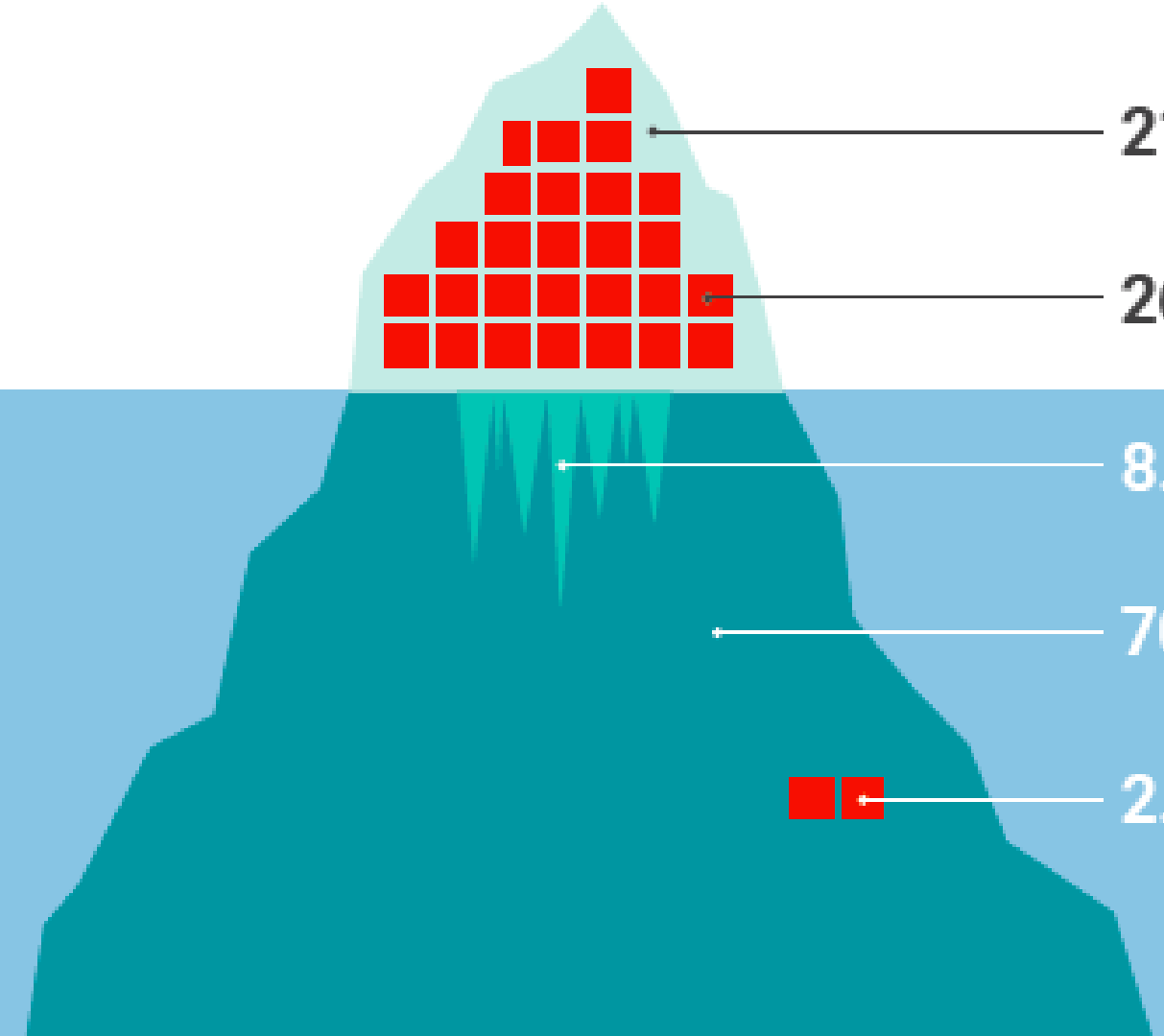
Broken Authentication

Injection

Broken Access Control

Source: Contrast Security

Typical Application



21% CUSTOM CODE

26.7 SERIOUS VULNERABILITIES

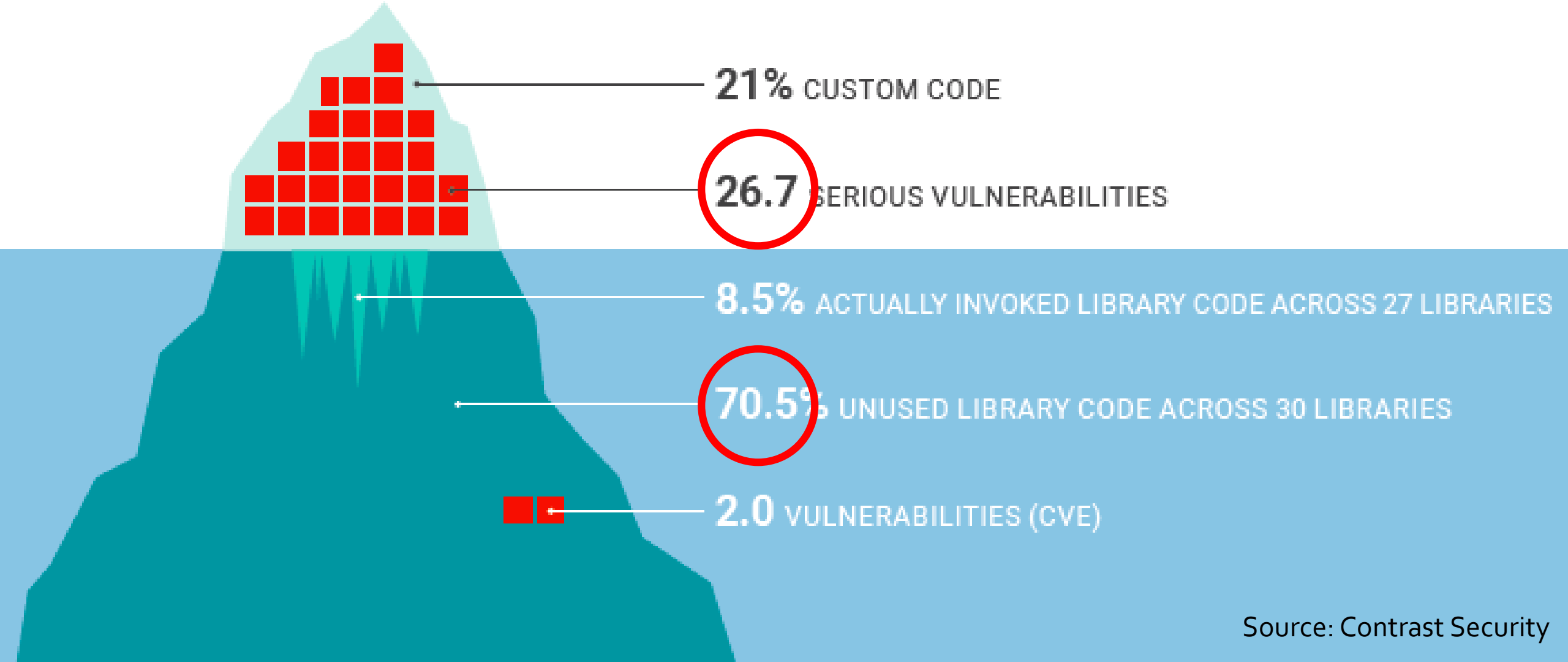
8.5% ACTUALLY INVOKED LIBRARY CODE ACROSS 27 LIBRARIES

70.5% UNUSED LIBRARY CODE ACROSS 30 LIBRARIES

2.0 VULNERABILITIES (CVE)

Source: Contrast Security

Typical Application



Source: Contrast Security


But Ned knew about those challenges…

He Had Some Climbing Gear

Static Application Security Testing

Dynamic Application Security Testing

Web Application Firewall

Compliance

Ned was Prepared for Anything

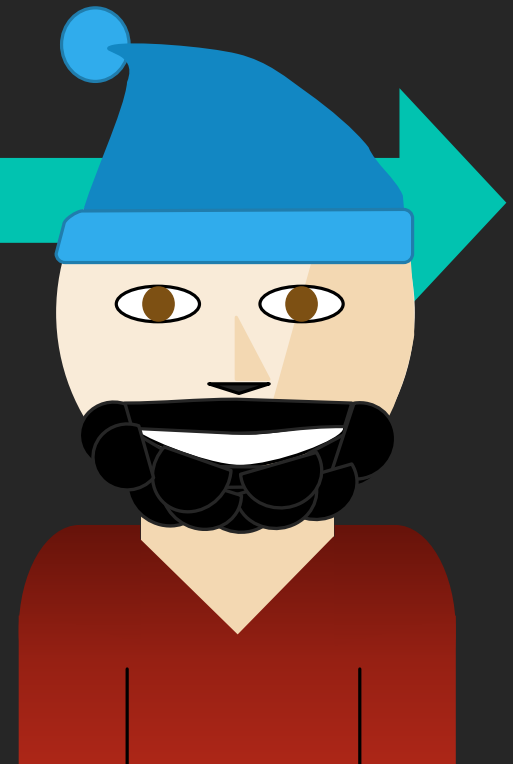

The State of Application Security

Hidden Challenges

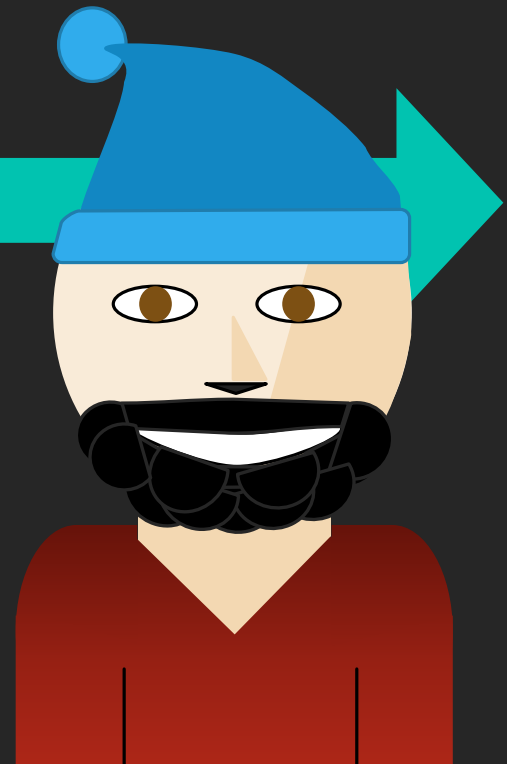
- Old Tooling
- Mergers & Acquisitions
- New Products and New Technologies
- Workplace Shifts
- Inflexible Developers


OLD TOOLS

Old Security Tools



Old Security Tools



Old Security Tools

- Can be slower
  -  Scanning
  -  Remediation
- Can miss vulnerabilities
- Alert Fatigue
- Do not offer the best protection against attacks


Foreshadowing...


MERGERS&ACQUISITIONS

Mergers and Acquisitions

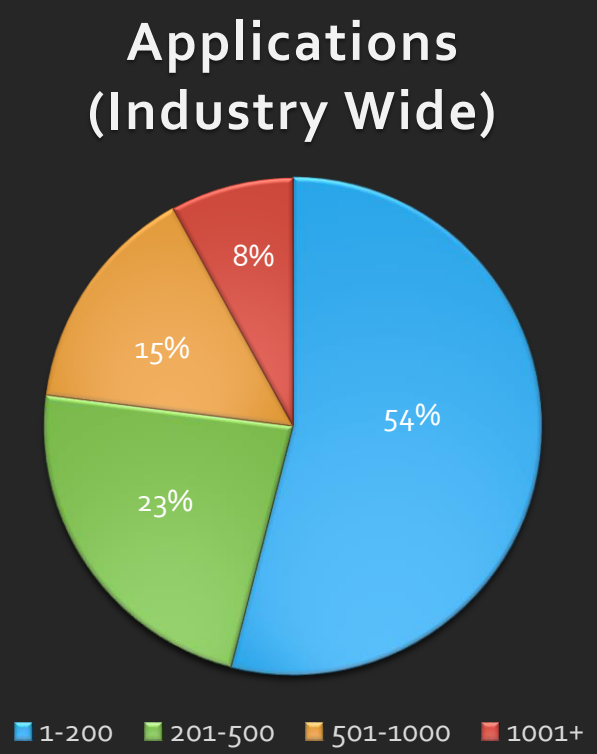
- M & A is a part of big business
- Acquiring new code bases is a part of that

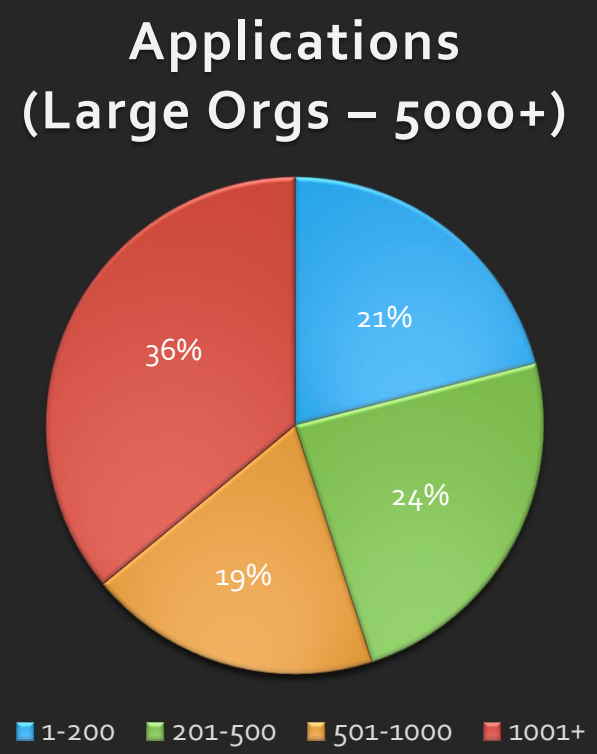
"Every business will be a software business."
- Satya Nadella (2015)

Mergers & Acquisitions

Old Tooling

Application Spread

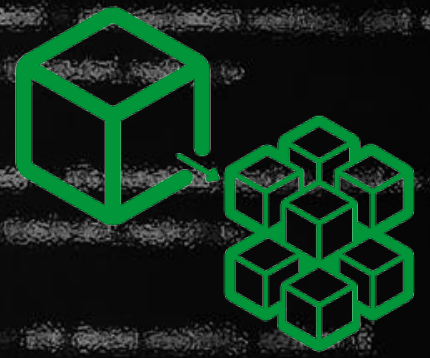




Source: F5 Networks

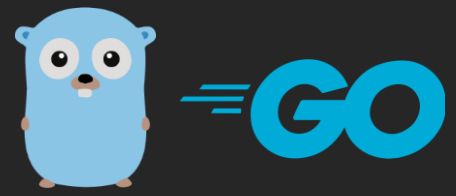

New Products & New Technologies

Software Change is Constant



New Products & New Technologies

- What's the new hotness?

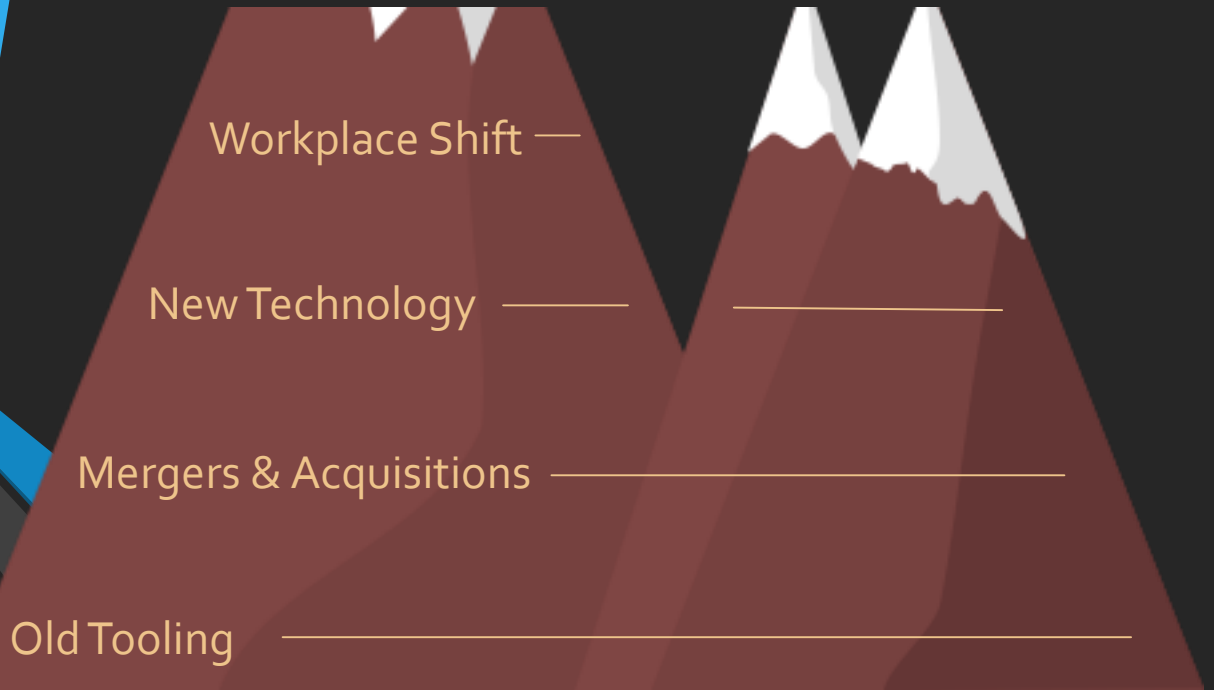


- Chances are, Ned's existing tools don't cover it and his staff doesn't have the experience

Workplace Shift

Workplace Shift

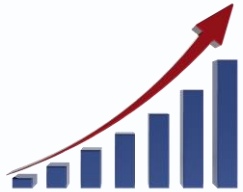
- Train new employees – sometimes in remote locations
- How to retain knowledge of the departed



Inflexible Developers

Inflexible Developers

Security is seen as a blocker



Security and Tech debt grows

Engineering fights to backlog Security

Findings are swept under the rug

That Mountain Is Looking Big


It Seems Impossible…


How to Summit

Ned is Still at Base Camp

But Ned Wants to Climb

Ned
@AppSecNed

I want to CLIMB THAT MOUNTAIN!

7:26 AM · Mar 5, 2020 · Twitter Web App

View Tweet activity

Ned's Directives

- Ensure all current codebases are secure
  - Including all Open Source Software utilized
- Ensure all future development is secure
  - Shift Left with future development
- Train software developers in secure coding practices
- Minimize attack vectors in external facing applications

Shifting Left

Requirements → Design → Code → Test → Acceptance → Production → Maintenance

SHIFT TESTING LEFT

Where your scanning security tools currently sit

WAF
(Is likely in monitor mode)

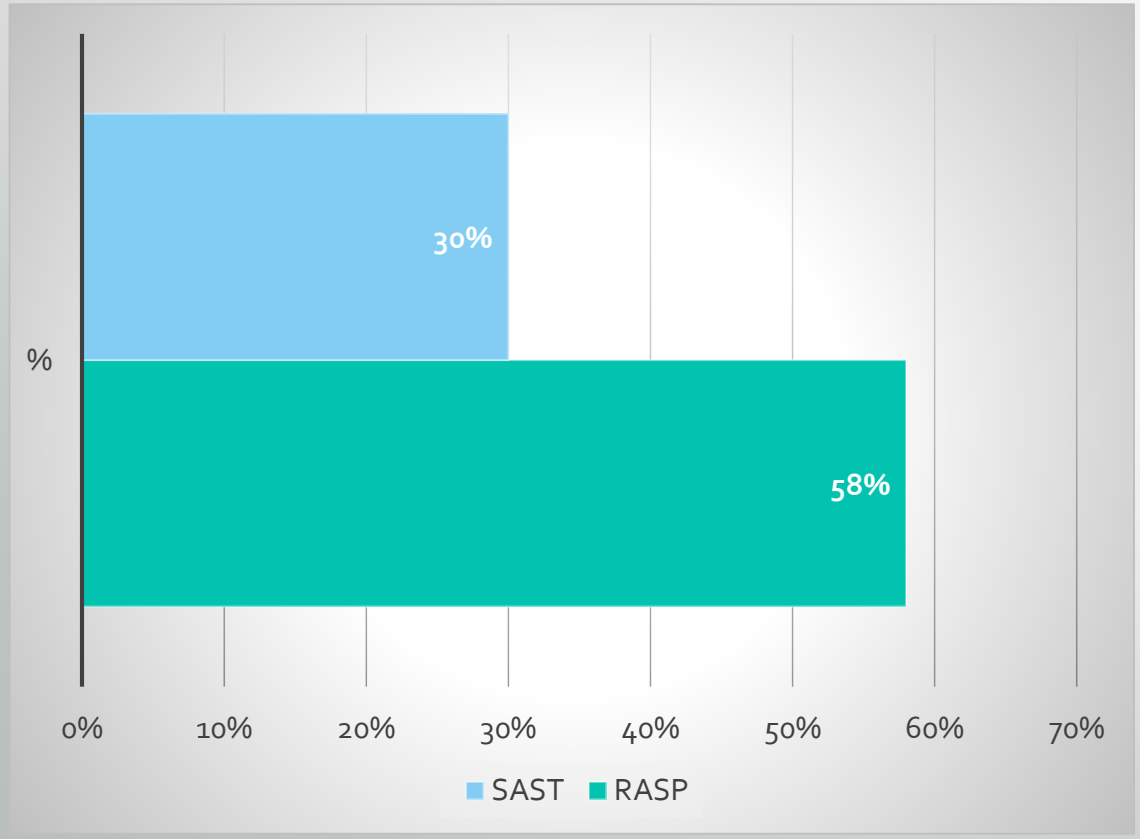
Review Challenges

- Old Tooling
- Mergers & Acquisitions
- New Products and New Technologies
- Workplace Shifts
- Inflexible Developers

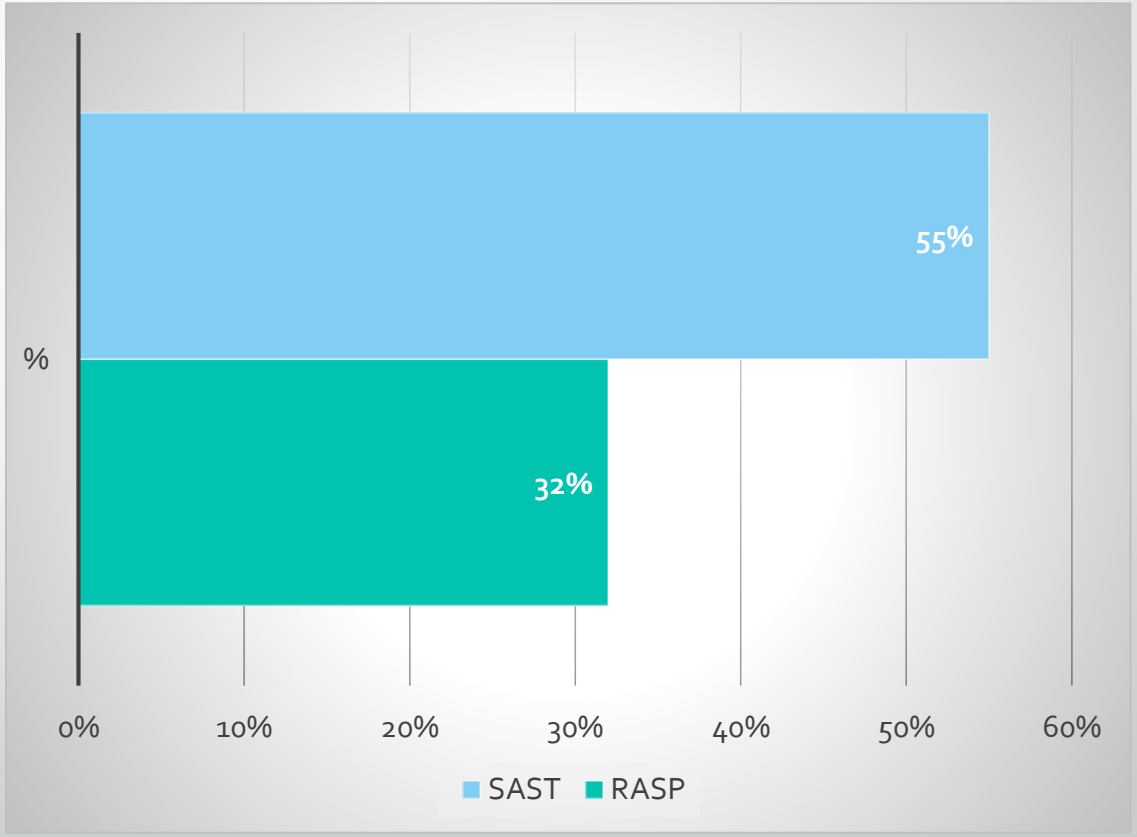
Security Tooling

- Can newer security tooling really help with speeds?
- The answer is a resounding yes!
- Introducing RASP – Runtime Application Self Protection
- Agents instrument the code bases to both:
  - Provide security against attacks
  - Allow for more expedient remediation by pinpointing issues

% of Vulnerabilities CLOSED
within 1 month of discovery



% of Vulnerabilities REMAINING
after 3 months of discovery



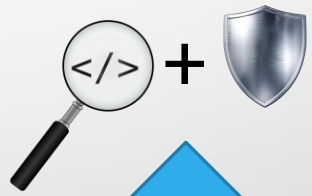
Sources: Contrast Security and Veracode

Shifting Out

Requirements → Design → Code → Test → Acceptance → Production → Maintenance

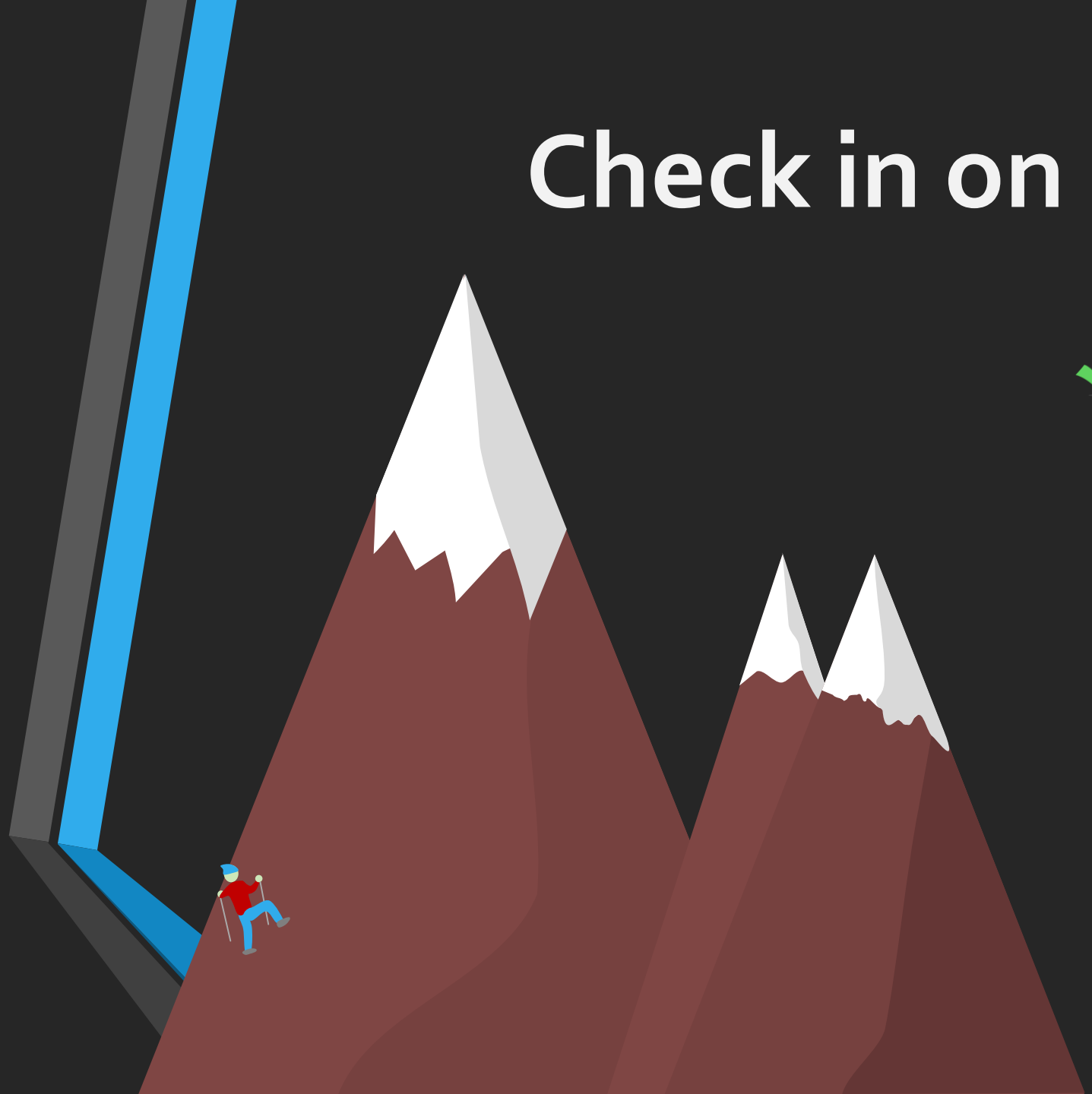
SHIFT LEFT

SHIFT RIGHT



WAF
(In block mode)
+
RASP

Check in on Ned…



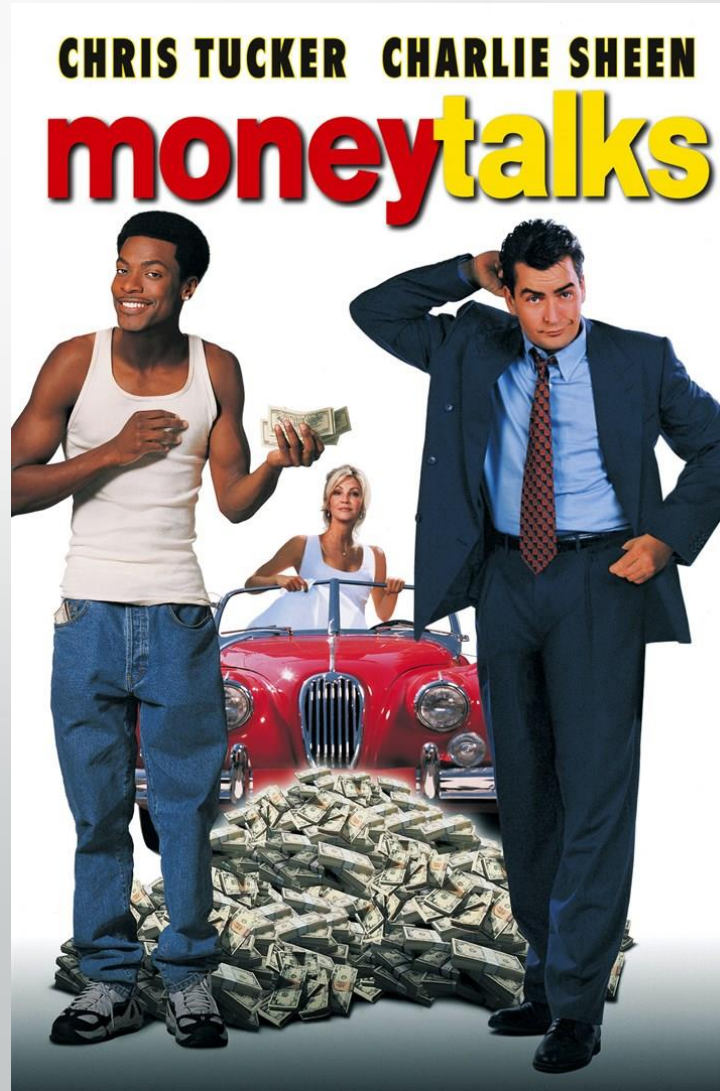
- ✓ Old Tooling
- Mergers & Acquisitions
- New Products and New Technologies
- Workplace Shifts
- Inflexible Developers


Still a long way to go

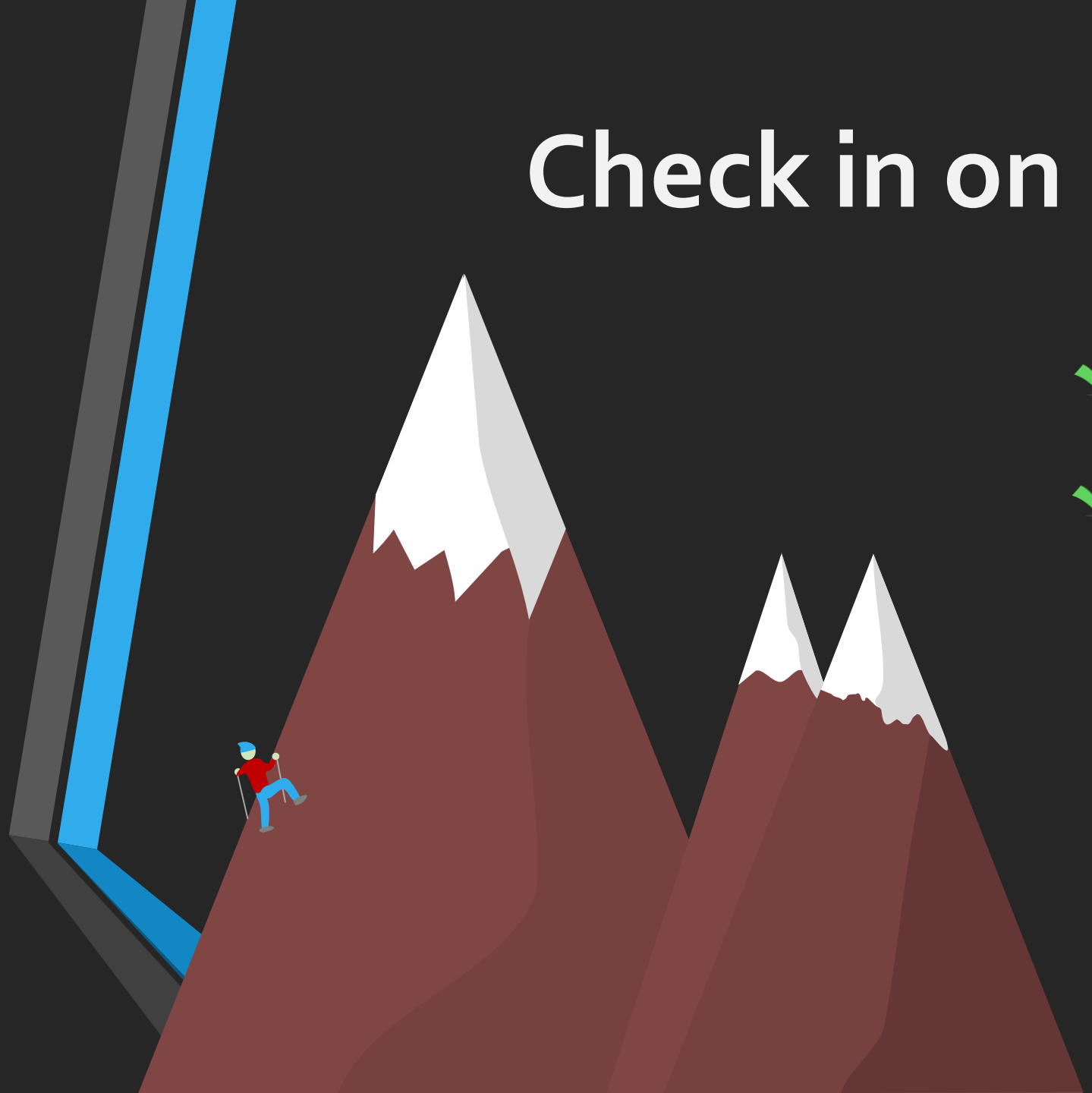
Mergers & Acquisitions

- Ned convinces his executive level staff to get App Sec more involved in M & A discovery

- After all, it can only be positive to involve the Application Security team



Check in on Ned…



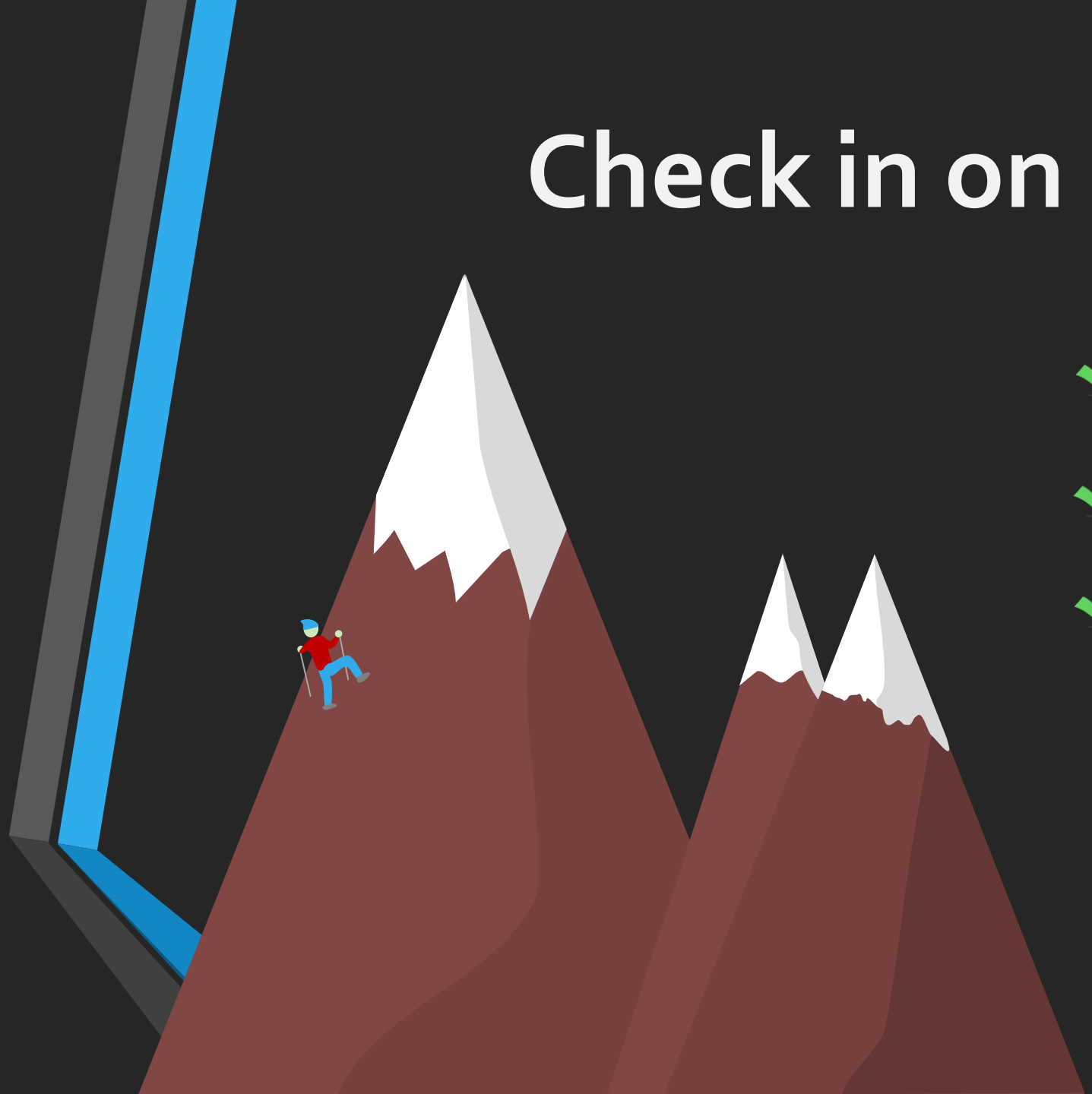
- ✓ Old Tooling
- ✓ Mergers & Acquisitions
- New Products and New Technologies
- Workplace Shifts
- Inflexible Developers

New Products and New Technologies

- Ned does NOT want to discourage INNOVATION
- He only wants to encourage CONSIDERATION

- Ned creates an Acceptable Use Policy
- Adding to the policy is easy, but it does need to be reviewed by an Enterprise Architecture Review Board (EARB)
- Enforceable by compliance

Check in on Ned…



- ✓ Old Tooling
- ✓ Mergers & Acquisitions
- ✓ New Products and New Technologies
  - Workplace Shifts
  - Inflexible Developers

Workplace Shift

The Departed

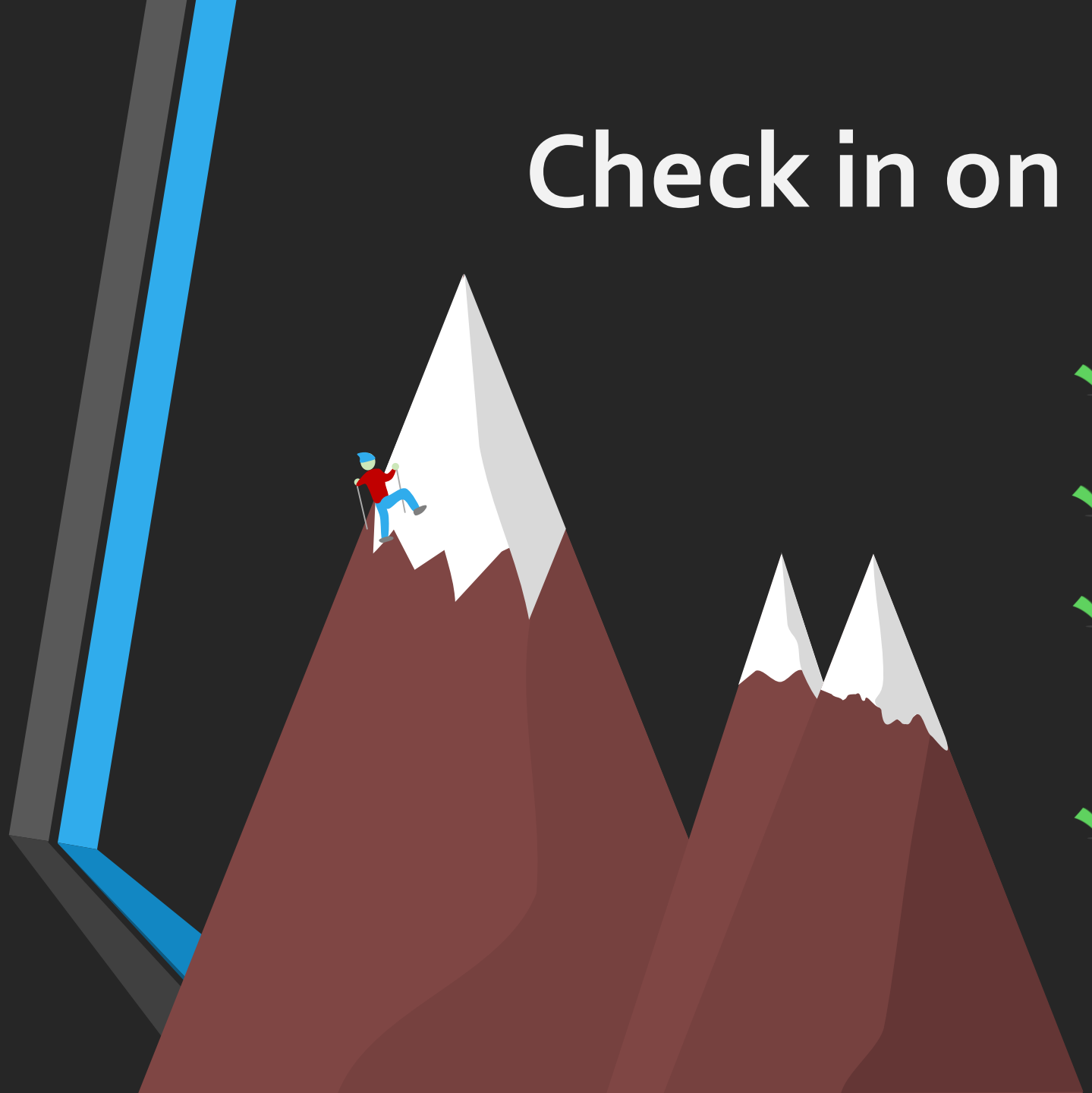
- This happens. It is not always fun and not always with much notice
- If Ned does not have advanced warning of a relocation or layoff there are a few things that he can be prepared with ahead of time
  - Burn Lists
    Remove all user accounts in every application. This can be vast and take significant time, so the sooner Ned can automate it, the better.
  - Default Creds Sweep
    To ensure that not only are the departed's credentials gone from valued applications, Ned also periodically checks his applications against a known list of test credentials developers use in test environments

Workplace Shift

The New Employees

- Ned uses a training platform that works remotely and is easily to distribute. It is important that any new employee go through secure software development training before they touch code.
- Ned also creates documentation around standard best coding practices. His team stubs out authentication, session management, basic API requests and more.

Check in on Ned…



- ✓ Old Tooling
- ✓ Mergers & Acquisitions
- ✓ New Products and New Technologies
- ✓ Workplace Shifts
- Inflexible Developers

High Enough to Appreciate the View

But Not Comfortable Yet

Inflexible Developers

- Application Security fighting Engineering is a fight where everybody loses
- Several things can improve that relationship:
  1. Application Security should be housed outside of the Engineering group, either under a CISO, CIO, or CFO. This helps create a separation of priorities among Engineering management.
  2. It is important to reiterate that Application Security is a resource, not a hinderance – after all, developers don't want to push vuln code!
  3. Application Security Engineers should have some development experience or extensive experience with code.

Endearing Yourself to Developers

Training can be Fun

- Gamified Competitions
  - Capture the Flag (CTF) contests
  - Secure Coding Hackathons
- Lunch-and-Learn Sessions
  - Q & A Sessions
  - Provide Lunch once a month, pizza, etc

Joint Outings

- Include Development in Team outings

Check in on Ned…

- ✓ Old Tooling
- ✓ Mergers & Acquisitions
- ✓ New Products and New Technologies
- ✓ Workplace Shifts
- ✓ Inflexible Developers

Ned's Directives

- ☑ Ensure all current codebases are secure
  - ☑ Including all Open Source Software utilized
- ☑ Ensure all future development is secure
  - ☑ Shift Left with future development
- ☑ Train software developers in secure coding practices
- ☑ Minimize attack vectors in external facing applications

Questions or Comments

@clevernyyyy

Thank You

@clevernyyyy