Vulnerability Management

Steve Schofield : “VA analyst by trade, obsessed w/ Pentesting, with a developer habit”

Who am I? : Steve Schofield

Title : Cybersecurity Analyst

Experience : Varied and a few years ago

Current Role : “DevSecOps Champion” / AppSecDev

Professional Desire / Goals :
Ethical Hacker / Offensive Security / Pentesting

Have Hobbies!

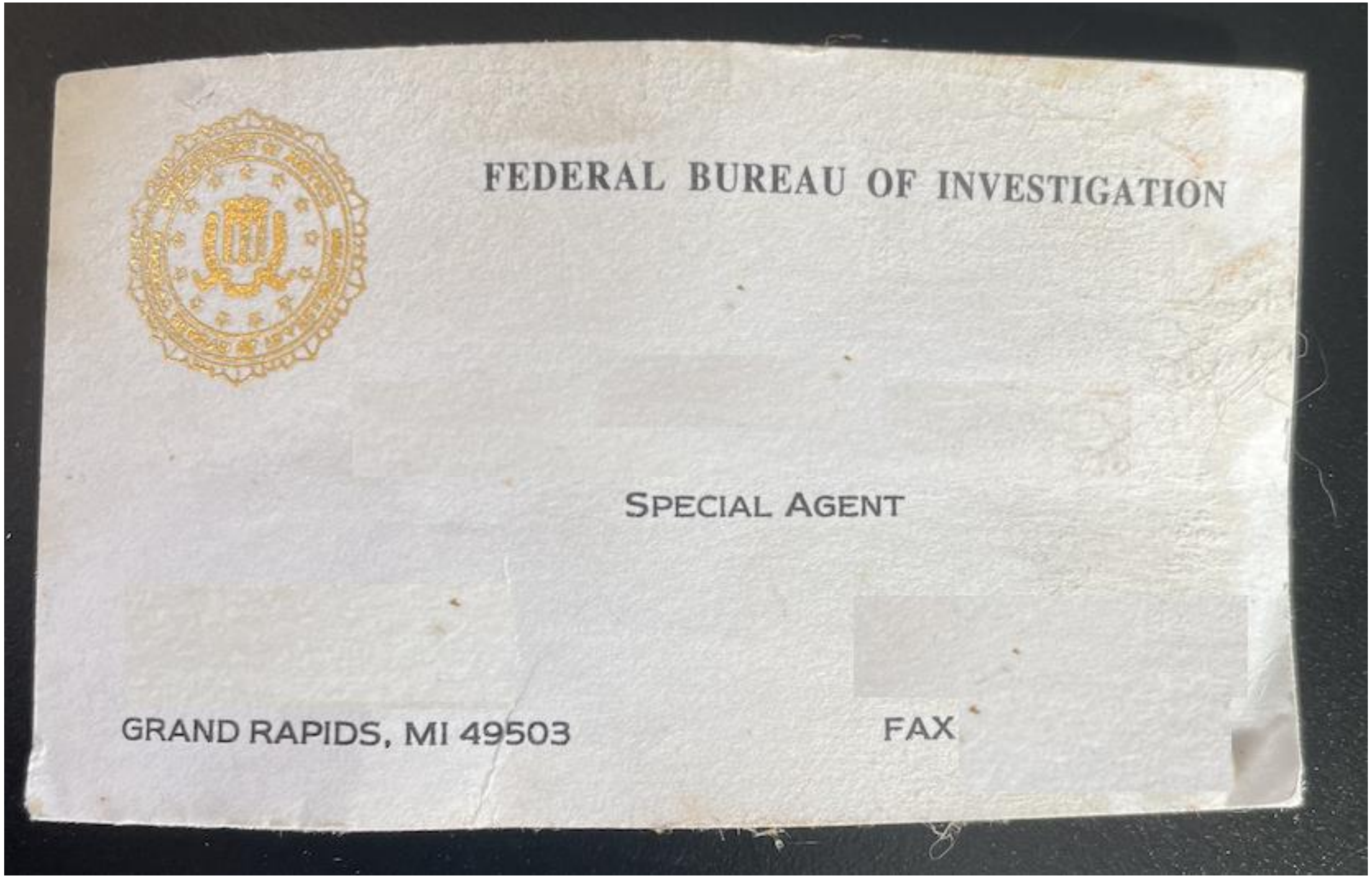
Try things

Have Passion

Keep Learning

1952 F1 Ford with Flathead v8 (106 hp, 3 speed on tree)

How did I get into information Security?



FEDERAL BUREAU OF INVESTIGATION

[redacted]

SPECIAL AGENT

[redacted]

[redacted]

GRAND RAPIDS, MI 49503

FAX [redacted]

You are the decision maker!

Select Three

1. New Features released to customers?

2. Good performance of application?

3. Meeting a date?

4. Security?

When do you want feedback?

1. Design Time? (more findings, earlier in development)

2. Run time? (later in dev, less findings)

3. Blend?

Vulnerability Management

What is a Vulnerability?

Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

credit : https://csrc.nist.gov/glossary/term/vulnerability

Vulnerability Management

- **Define Risks / Standards :** (Governance, Risk and Compliance) - *Leadership sign-off* and *"rules of engagement"*

- **Scan Assets :** find assets through discovery / performing when possible authenticated scans. Non-auth scans are usually port based

- **Report :** determine based on critical, high, medium or low. Refer to Risk standards timeframe

- **Remediate** : Work with teams supporting servers and apps

- **Verify** : have a regular scan schedule to verify status. If vulns found, some way to assign / report to teams to follow-up

Vulnerability “Capabilities / Terms”

Static Code Analysis

- **SCA** (Software Composition Analysis) - 3rd parties' libraries / dependency checks
- **SAST** (static code and infrastructure as code) - custom code

Run Time Analysis

- **IAST** (interactive application security testing)- scan vulnerabilities during run time during lower environment (containers, function-less apps, mobile apps – ipa / apk)
- **DAST** (dynamic application security testing) - web apps, API, infrastructure in pre-production (human driven)
- **Pentesting** (automated or manual)

Observability

- **Scan after patches pushed**
- **monitor for exploits / prevention**
- **Pentesting** (automated or manual)
- **SIEM / Incident Response**

What is DevSecOps?

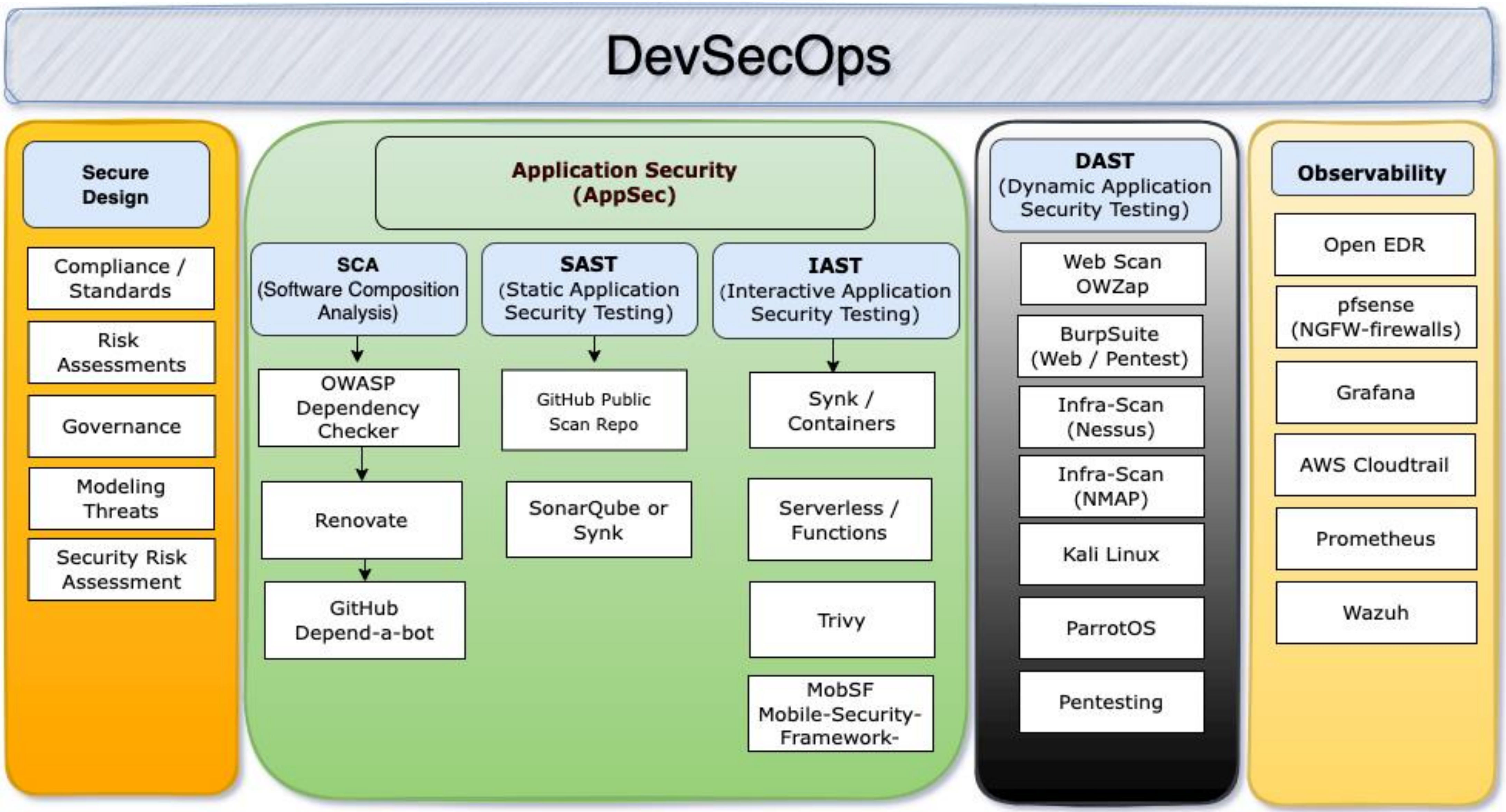
>> People, Process, then Technology <<

*“Integrating security into DevOps to deliver "DevSecOps" requires **partnerships** between Security and Development. Security **cannot force** adoption rather meet teams where they are at.*

*Having **empathy** towards people and processes, security will find **common ground** and achieve *real* collaboration. This will help advance greater DevSecOps **adoption** instead of DevOpsSec.”*

- Steve Schofield
“DevSecOps Champion”

DevSecOps Toolset



DevSecOps

Secure Design

- Compliance / Standards
- Risk Assessments
- Governance
- Modeling Threats
- Security Risk Assessment

Application Security (AppSec)

SCA (Software Composition Analysis)

- OWASP Dependency Checker
- Renovate
- GitHub Depend-a-bot

SAST (Static Application Security Testing)

- GitHub Public Scan Repo
- SonarQube or Synk

IAST (Interactive Application Security Testing)

- Synk / Containers
- Serverless / Functions
- Trivy
- MobSF Mobile-Security-Framework-

DAST (Dynamic Application Security Testing)

- Web Scan OWZap
- BurpSuite (Web / Pentest)
- Infra-Scan (Nessus)
- Infra-Scan (NMAP)
- Kali Linux
- ParrotOS
- Pentesting

Observability

- Open EDR
- pfsense (NGFW-firewalls)
- Grafana
- AWS Cloudtrail
- Prometheus
- Wazuh

Summary

Leadership Support

- Find
- Report
- Remediate
- Verify
- Measure

Repeat…

Questions / Comments

DevSecOps Discovery

- Understand DevSecOps concepts and how relates to your company

- Find a *”reference” customer*

- Document standards, partner with customer

- Setup alias https://devsecops.example.com/ central document repository

- Defined four major areas involved defining Capabilities:
    - Threat Modeling / Planning
    - Static Scanning (SCA / SAST / IAST) - ”building product”
    - Dynamic Scanning (DAST) – “built, running in QA, not in prod”
    - Observability “In Prod”

Problem Statement

*"**AppSec**" is the process of finding, fixing, and preventing security vulnerabilities at the application level as part of the software development processes.*

- The scanning process in AppSec is **complex,** involves many tools.
- **Manual** integration can be time intensive process
- Developers mention on-boarding can be **cumbersome,** results can contain **“noise”** as well as **false positives**.

Goals to achieve by each capability

***"AppSec"** is the process of finding, fixing, and preventing security vulnerabilities at the application level as part of the software development processes.*

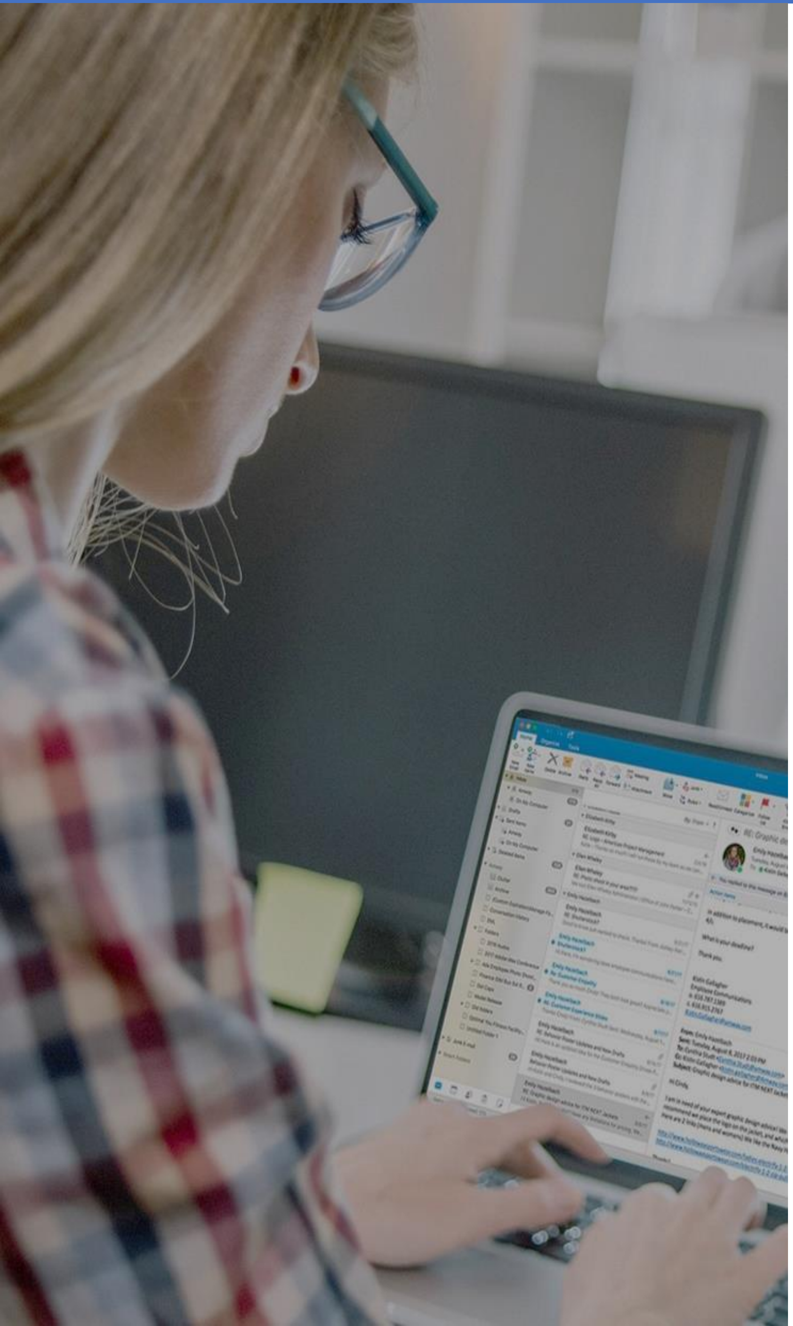
- Include development teams to review tools
- CI/CD integration providing feedback
- Discover and Report *vulnerabilities*
- Help provide Remediation options
- Provide friendly results to help with remediation efforts
- Near Zero Config needed to setup by developers
- Integration into "developer" tools (i.e IDE, GitHub, Jira, Slack)

Secure Design

“Threat modeling is a structured process with these objectives: identify security requirements, pinpoint security threats and potential vulnerabilities, quantify threat and vulnerability criticality, and prioritize remediation methods before starting any development.”

**https://www.synopsys.com/glossary/what-is-threat-modeling.html*

- Legal Review / Vendor Risk Assessments
- Governance, Risk and Compliance review
- Review with Stakeholders application plans / capabilities



Static Scanning - ”before build”

*“**AppSec** is the process of finding, fixing, and preventing security vulnerabilities at the application level, as part of the software development processes.”*

**https://www.checkpoint.com/cyber-hub/cloud-security/what-is-application-security-appsec/*

- SCA (Software Composition Analysis – 3rd party / Open Source packages) - 70% of app stack – Supply Chain...
- SAST (Static Code Analysis)
- IAST (Interactive Application Security Testing)
- MAST (Mobile Application Security Testing)

Dynamic Scanning (DAST) – “After build, not deployed to Prod…yet”

“D.A.S.T is the process of analyzing a web application and through the front-end to find vulnerabilities through simulated attacks and / or authenticated assessments.”

**https://www.microfocus.com/en-us/what-is/dast*

- Web and API scanning
- Container / Infrastructure Scanning
- Unit / Function Testing cases

Observability "In Prod" - Capabilities

"The goal of observability is to understand what's happening across all these environments and among the technologies, so you can detect and resolve issues to keep your systems efficient and reliable and your customers happy."

**https://www.dynatrace.com/news/blog/what-is-observability-2/*

- Cloud Account Configuration
- Device End Point protection
- API Threat Detection and protection
- Continuous and periodic scanning / penetration testing.
- Intelligence / Threat hunting / SOC services

- http://nmap.org **nmap --script vuln scanme.nmap.org**
- https://owasp.org/www-community/Free_for_Open_Source_Application_Security_Tools
- https://www.kali.org/
- https://github.com or https://gitlab.com (source control)
- https://metasploit.com. (Metasploit framework)
- https://www.tenable.com/downloads/nessus. (infra scanner)
- https://www.zaproxy.org/download/ (web)
- BurpSuite - https://portswigger.net/burp/communitydownload (web) and check **Web Academy** lessons for web hacking
- Vuln apps (OWASP WebGoat, NodeJS)
- https://github.com/trimstray/the-book-of-secret-knowledge
- https://trivy.dev/
- https://github.com/mobsf

Tips for mobile devices

- Keep applications update-to-date. Set software updates to automatically run, when possible.
- Enable PIN / Face recognition
- Turn off Bluetooth if not using
- Turn off Hotspot
- Review "App Background refresh" (I disable)
- Review Location services
- Review when an app asks for additional permissions, (camera, mic, location, Bluetooth)
- Remove apps not in-use (review one / twice yearly)

Suggestions / Tips / Tricks

- Learn to script (Python, PowerShell, Go)
- Automation
- Learn Software Development practices
- #Homelab
- Experiment, try, fail, learn, try
- ***Don’t expect perfection. Do something uncomfortable on-purpose***