#NoHobbyists

What's really needed to shift Cyber Security?

Trac Bannon
Senior Principal
Advanced Software Innovation Center

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD

Who am I?

Tracy L. Bannon

- ✓ Senior Principal with the MITRE Corporation
- ✓ Software Architect and Engineer
- ✓ Focused on problem solving using software

/trās/

What are my tags?

Red Cross servers 'were hacked via unpatched flaw'

18 February 2022

California public office admits Covid-19 healthcare data breach

Network cavity blamed for data breach at Japanese candy maker

29 March 2022

SQL injection vulnerability found in Moodle e-learning platform

08 March 2022

Zero-day XSS vulnerability found in Horde webmail client

24 February 2022

Attackers getting faster at latching onto unpatched vulnerabilities

28 March 2022

Nvidia cyber-attack linked to Lapsus$ ransomware gang

28 February 2022

GitLab addresses critical account hijack bug

01 April 2022

Japanese retailer traces breach to third-party hack

04 March 2022

Prison data breaches

UK Ministry of Justice recorded more than 2,000 incidents over 12 months

14 March 2022

Okta investigation

Authentication and identity management giant probes LAPSUS$ gang's compromise claims

22 March 2022

Source: https://portswigger.net/daily-swig

Quick! Shift Left!

Shifting Security “left” means…

Security activities start at design

Extend through out the SDLC

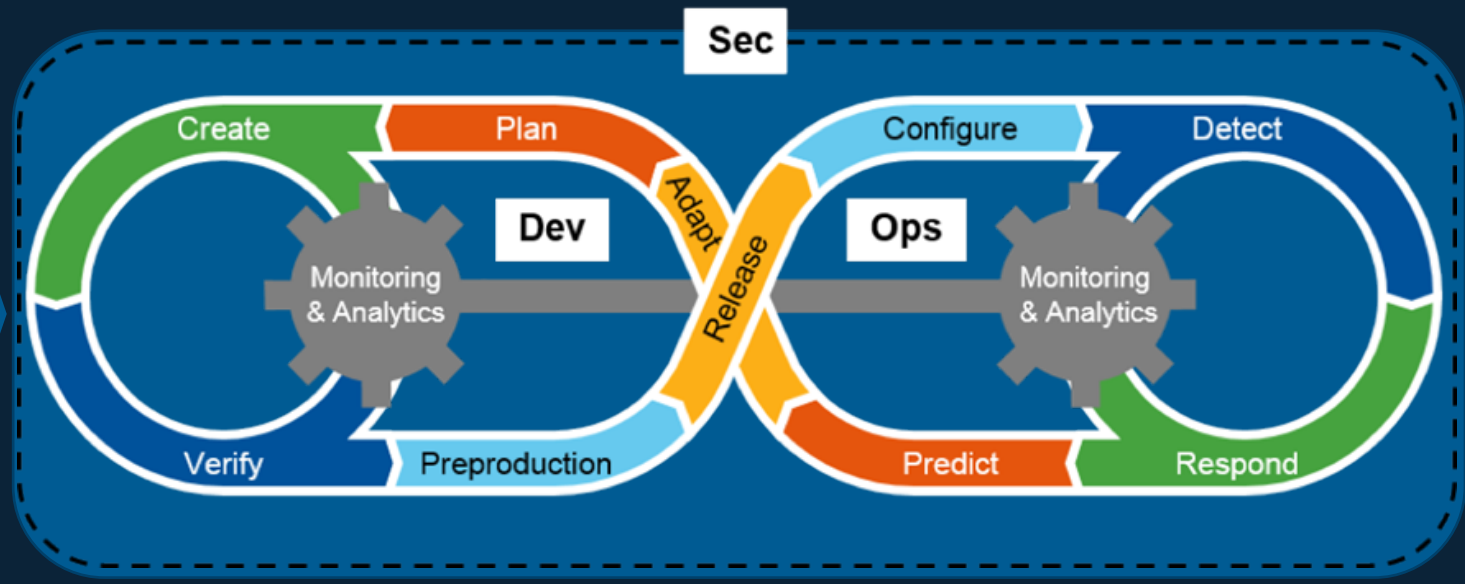
From vision through operate

Continuous feedback at every step

MITRE

Make Cyber Security Ubiquitous

Don’t Shift Left!
Shift Everywhere!



Source: https://tech.gsa.gov/guides/understanding_differences_agile_devsecops/

Security and Cyber Resilience is Everyone’s Responsibility

MITRE

Address the Four Facets!

Any “shifts” will fail unless you are holistic

People – roles, autonomy, upskilling

Process – workflows, ceremonies, domain specific

Technology – new tech insertion, digital platforms, production

Culture – Trustworthiness, psychological safety, collaboration

MITRE

MITRE

Sounds logical, but how?

Start by beginning to build a new culture

Together …

Define unifying principles

Agree on roles and responsibilities

Define your Security Software Development Framework (SSDF)

Work elbow-to-elbow

Demonstrate model behaviors

MITRE

Secure Software Development Framework

Don’t start from scratch

Use industry standards like NIST Special Publication 800-218.

This tells you what to do; you define how.

- Clearly defined roles and responsibilities
- Provide adequate software security training
- Agree on secure software development lifecycle
- Establish secure coding standards
- Build and leverage reusable objects
- Verify security control

MITRE

Four Question Framework[1]

Make threat modeling a team sport

What are we working on?

What can go wrong?

What are we going to do about it?

Did we do a good job?

MITRE

Thread Modeling Manifesto

Why thread model?

> “you begin to recognize what can go wrong in a system. It also allows you to pinpoint design and implementation issues that require mitigation”

Who should threat model?

> “You. Everyone. Anyone who is concerned about the privacy, safety, and security of their system”

https://www.threatmodelingmanifesto.org/

Capture threat information in plain-language

User Story Madlib*

As a _____

I want to _____

So that _____

** I want you to:

Protect _____

From _____

Source: Alyssa Miller

MITRE

Add in critical asset and the possible threat

User Story Madlib

As a Car Driver

I want to Enter a destination name

So that I can navigate w/o an address

** I want you to:

Protect My search history (Critical Asset)

From Being accessed by others (Threat)

MITRE

MITRE

Is it time to code yet?

What is a Security Hobbyist?

- ! More responsibility is being placed on developers
- ! Constant addition of new tools
- ! Training lacks depth
- ! Continued pressure to “go fast”
- ! Reading and experimenting on their own time, if at all

*Developers are **not** experts “waving magic wands to cast spells that can defend against evil hackers in black hoodies.”*[2]

MITRE

#NoHobbyists

From Hobbyist to Pro

How can we protect and enable the developers?

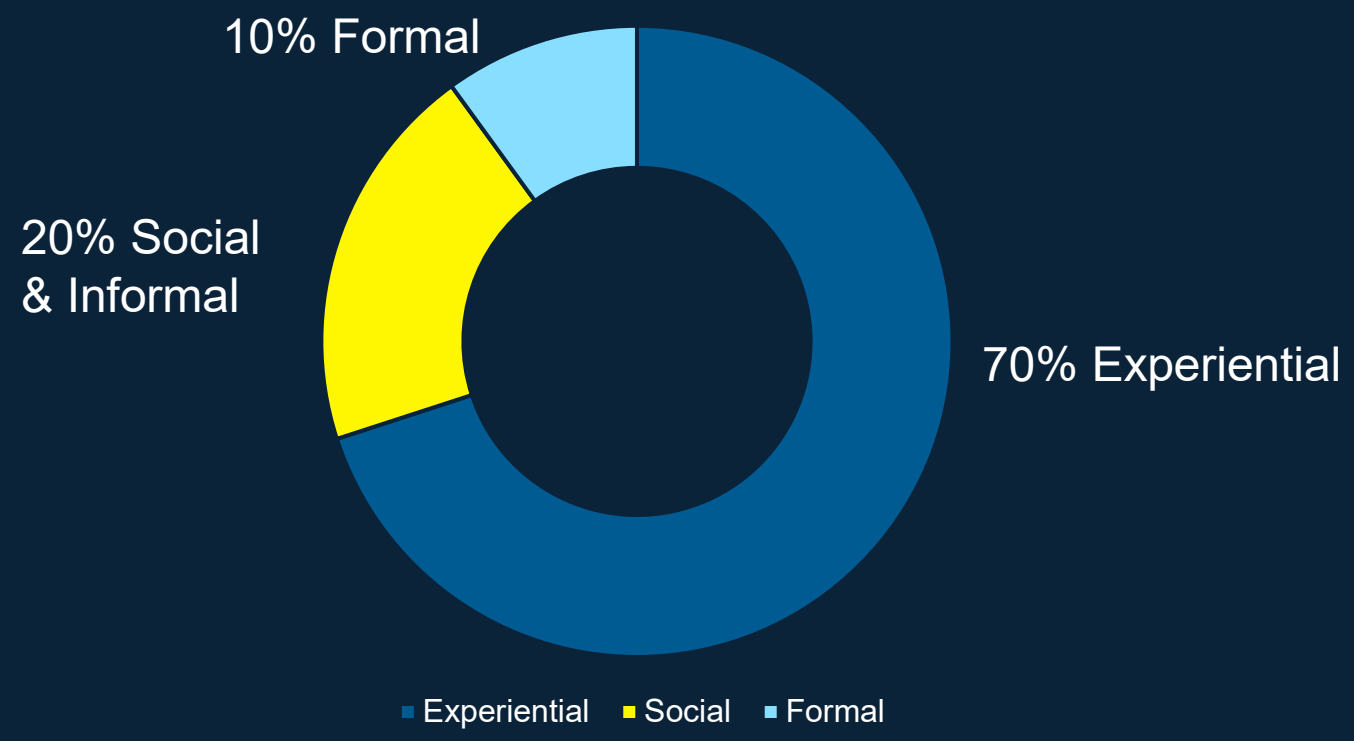
- ✓Training and Education
- ✓Make sure you are secure by design
- ✓Use secure coding standards
- ✓Leverage design patterns
- ✓Encourage experimentation
- ✓Double down on tools to help reduce noise and cognitive overload

Upskill and Train Developers

Use a 70:20:10 Model

Current cybersecurity training for developers lacks depth

Start with core concepts then move to experiences



MITRE

Non-classroom Training Examples

Run a team hack-a-thon using an OWASP project like Juice Shop or Web Goat (https://owasp.org/projects/)

Join a competition at the MITRE Cyber Academy (https://mitrecyberacademy.org/)

Pair program with SAST tools (Static Application Security Testing)

Storm-the-castle lunch events using OWASP ZAPfor DAST (Dynamic Application security testing)

MITRE

MITRE

Is it time to code yet?

MITRE

"Design is the guiding principle for how a system is built and is applicable on all levels, from code to architecture. It includes any activity that involves active decision-making."[3]

Before you code, be secure by design

Security is a **concern**, not a feature

Explicitly **think** about security

Context and Archetype matter

Focus on domain; many security bugs are caught implicitly

MITRE

Secure Coding Practices

Estimated 82% of software vulnerabilities are from coding errors.[2]

Identify secure coding standards as a team

Make sure everyone knows the standard

Run Static Application Security Tools (SAST)

Developers can run SAST before code complete

MITRE

Sample Secure Coding Standards[4]

- CWE and CWE Top 25
- CERT – From Carnegie Mellon's SEI - C, C++, Java, Perl, and Android
- OWASP and OWASP Top 10 – Web apps and APIs
- CVE - Cybersecurity vulnerabilities and exposures
- NVD -
- DISA STIG – DOD's Secure Technical Implementation Guide
- PA-DSS – Payment application systems
- IEC 62443 - Industrial networks

Sample Secure Coding Practices

- Input validation
- Output encoding
- Authentication and password management
- Session management
- Access control
- Error handling and logging
- Data protection
- Communication safety
- System configuration
- Database security
- File management memory management
- . . .

MITRE

Is it time to code yet?

Dedicate time to dependencies

Document package & program dependencies

Align to stream your SSDF

Run Software Composition Analysis (SCA)

Automate SBOM creation

MITRE

MITRE

Learn to break things!

Learn to love Application Security Testing (AST)

Static App Sec Testing (SAST) – Earliest Detection

Dynamic App Sec Testing (DAST) – outside perspective before production

Interactive App Sec Testing (IAST) – agent based runtime grey-box

SCA - Software Composition Analysis

MITRE

MITRE

"There is simply too much code being produced for humans to handle alone from a security perspective."[3]

Tools to Help Developers

(just a TINY sample)

OWASP Threat Dragon

OWASP Threat Model Cookbook

Automated Code Security Remediation

Code Quality & SAST

GitGuardian Secrets Detection

HashiCorp Vault Secrets Management

Dependency Scanning

OWASP ZAP DAST “black box” testing

MITRE

Security is everyone's responsibility.
Everyone needs to be equipped.

MITRE

Let’s code!!

Tracy L. Bannon
TBannon@MITRE.org
TracyBannon@gmail.com
https://www.linkedin.com/in/tracylbannon
@TracyBannon

Disclaimer: The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

References:

[1] https://owasp.org/www-community/Threat_Modeling

[2] https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020/

[3] https://www.securecodewarrior.com/blog/certified-security-awareness-an-executive-order-to-elevate-developers

4 https://www.perforce.com/blog/qac/secure-coding-standards#:~:text=Secure%20coding%20standards%20are%20rules,that%20could%20compromise%20software%20security