

# BIO/WHO AM I?

* 1996-2006	Network Engineer <i>(recovering)</i>	Financial Institution
* 2006-2011	Security Analyst/Engineer	Financial Institution
* 2011-2014	Security Architect	Global Consulting Firm
* 2014-2022	Director of Security & IR	Analytics/Large Retailer
* 2022-2024	CISO/BISO-Americas field	Security/Compliance Vendor
* 2024-Current	Consulting CISO	CBTS





# Secrets Revealed: Launching a Successful DevSecOps Program



Josh Hankins, Consulting CISO  
CISSP | GPCS | GMON | GMOB

LinkedIn Contact



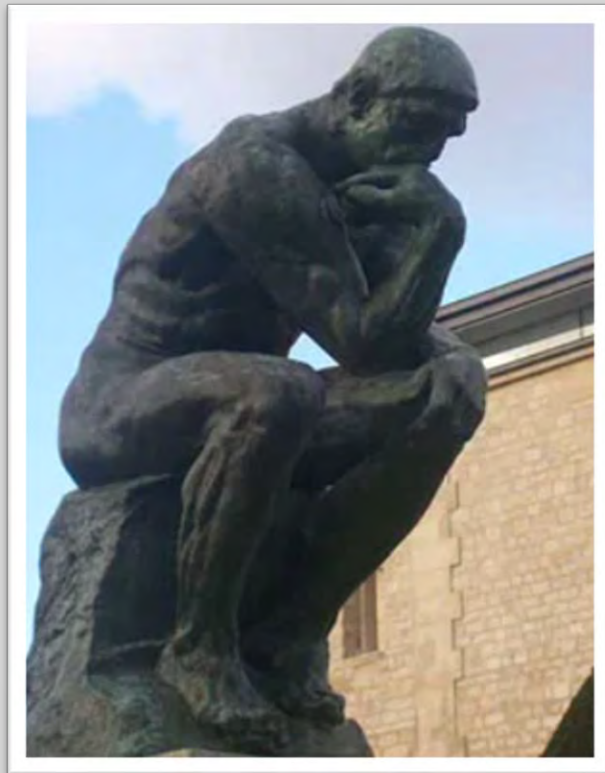


# AGENDA

- 1.) Goals
- 2.) Engagement Strategy
- 3.) Highlights
- 4.) Success Patterns
- 5.) How SBOM Shapes Next Steps



# DevSecOps, WHAT DOES GOOD LOOK LIKE?

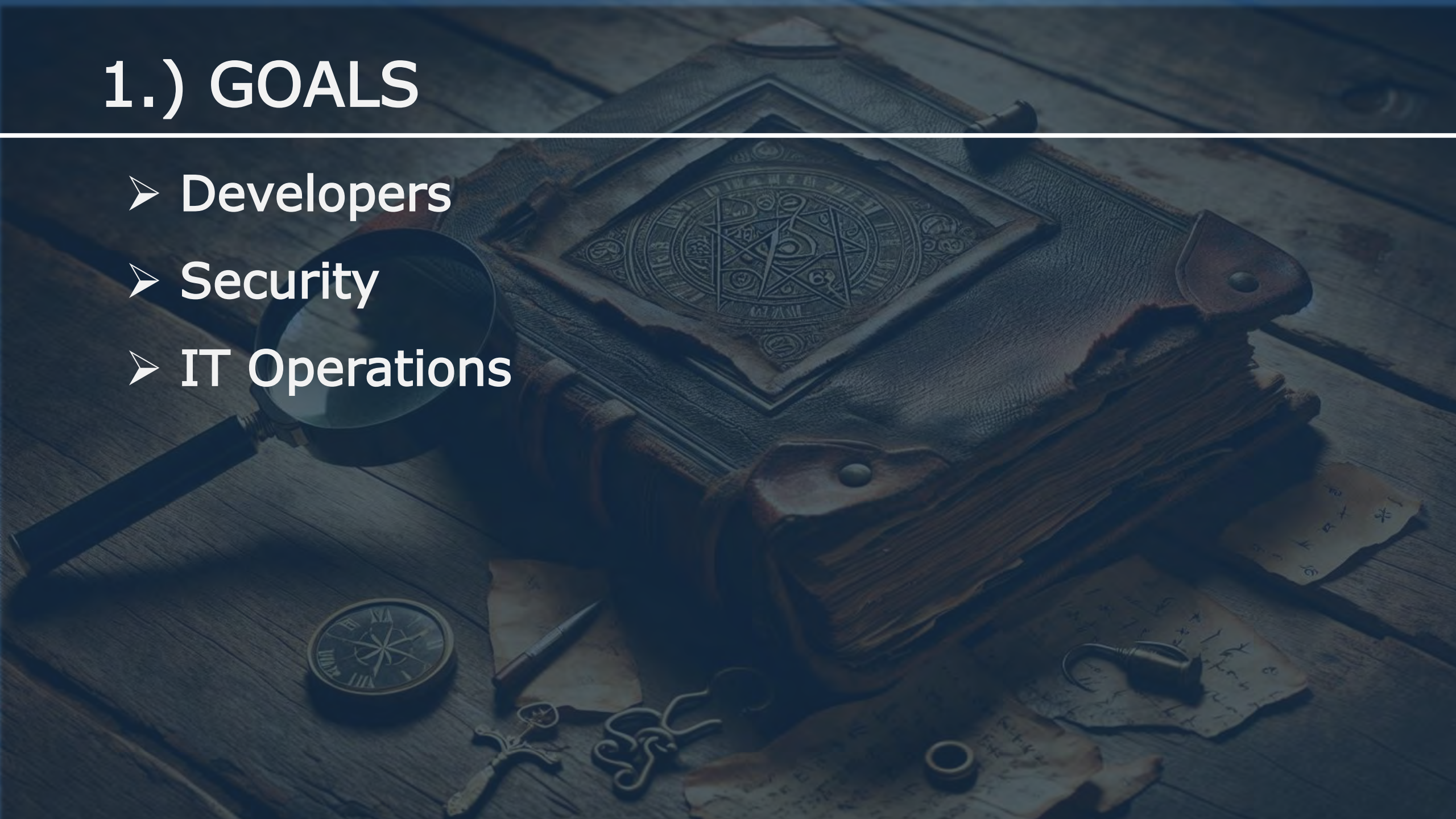


A successful blend of IT operations and security enabling developers to deliver products for the business in a secure manner.



# 1.) GOALS

- Developers
- Security
- IT Operations





# GOALS FOR DEVELOPERS

- ✓ \*Frictionless
- ✓ \*Secure coding IS code quality
- ✓ Prevent rework before deployment
- ✓ Continue to deliver stable & secure products
- ✓ Patterns easily transition to cloud
- ✓ What can be **Automated?** (Now & Future)



# GOALS FOR SECURITY

- ✓ \*Decrease application attack surface
- ✓ Mitigate the chances of breaches & outages with this emphasis on code quality
- ✓ Use a “bend but don’t break” security approach
- ✓ Automate, Automate, Automate



# GOALS FOR IT OPERATIONS

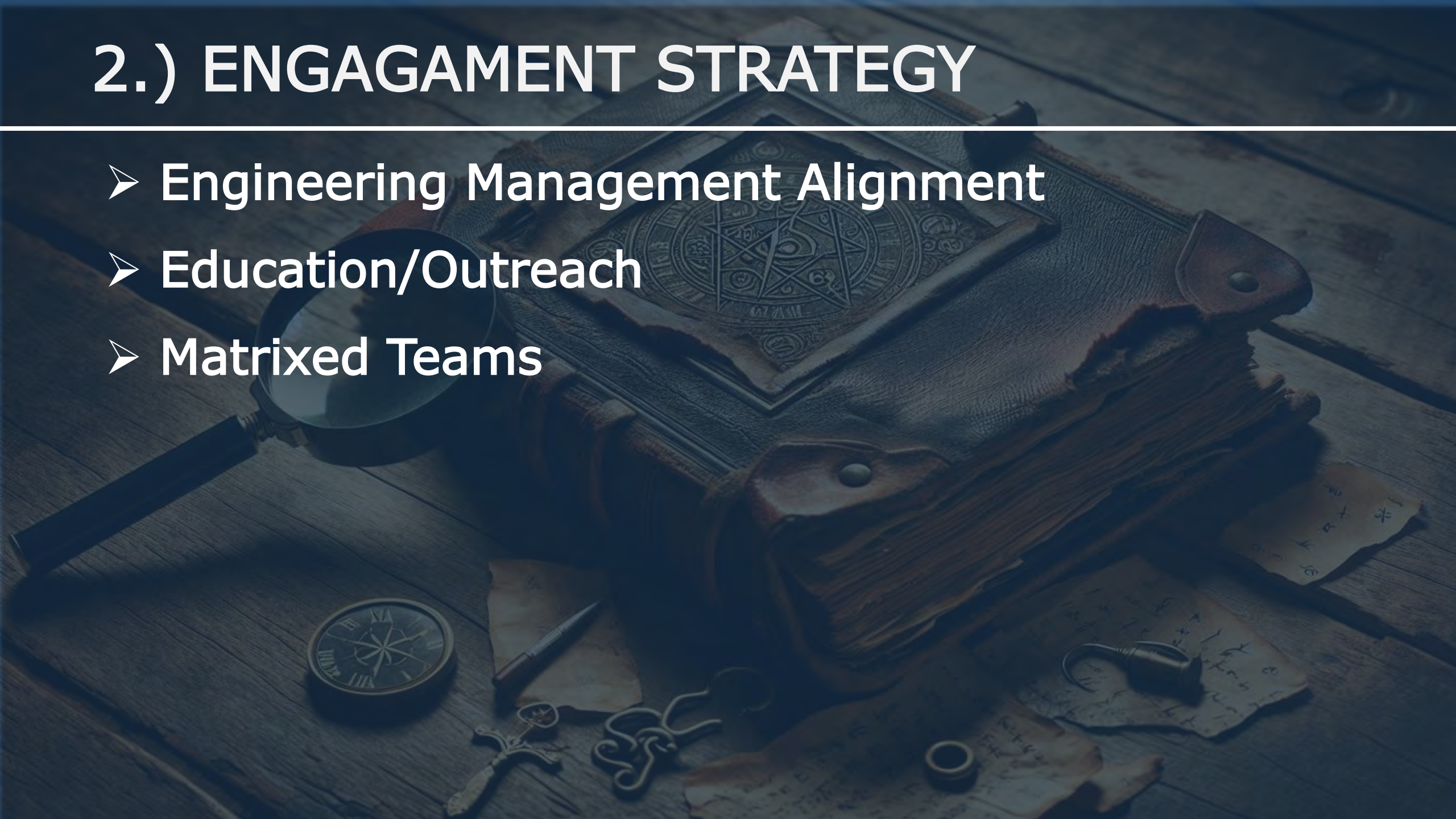
- ✓ Daily operations work – not interrupted
- ✓ Empowerment focus
- ✓ Upskill opportunity
- ✓ Automate, Automate, Automate





## 2.) ENGAGEMENT STRATEGY

- Engineering Management Alignment
- Education/Outreach
- Matrixed Teams



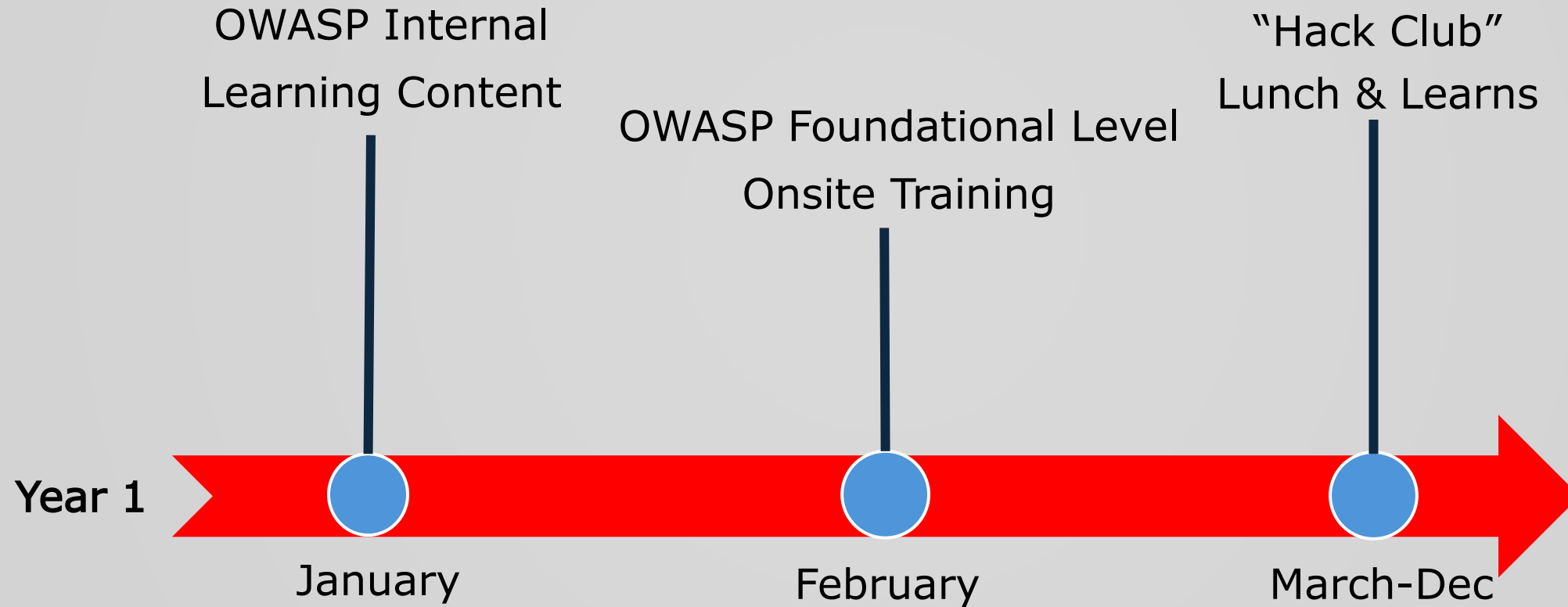
# ENGINEERING MANAGEMENT ALIGNMENT

- ✓ ~~"We don't have time for this DevSecOps initiative." (Priority)~~
- ✓ ~~"We don't have money for this new initiative."~~
- ✓ ~~We don't know how to do this effectively & efficiently".~~

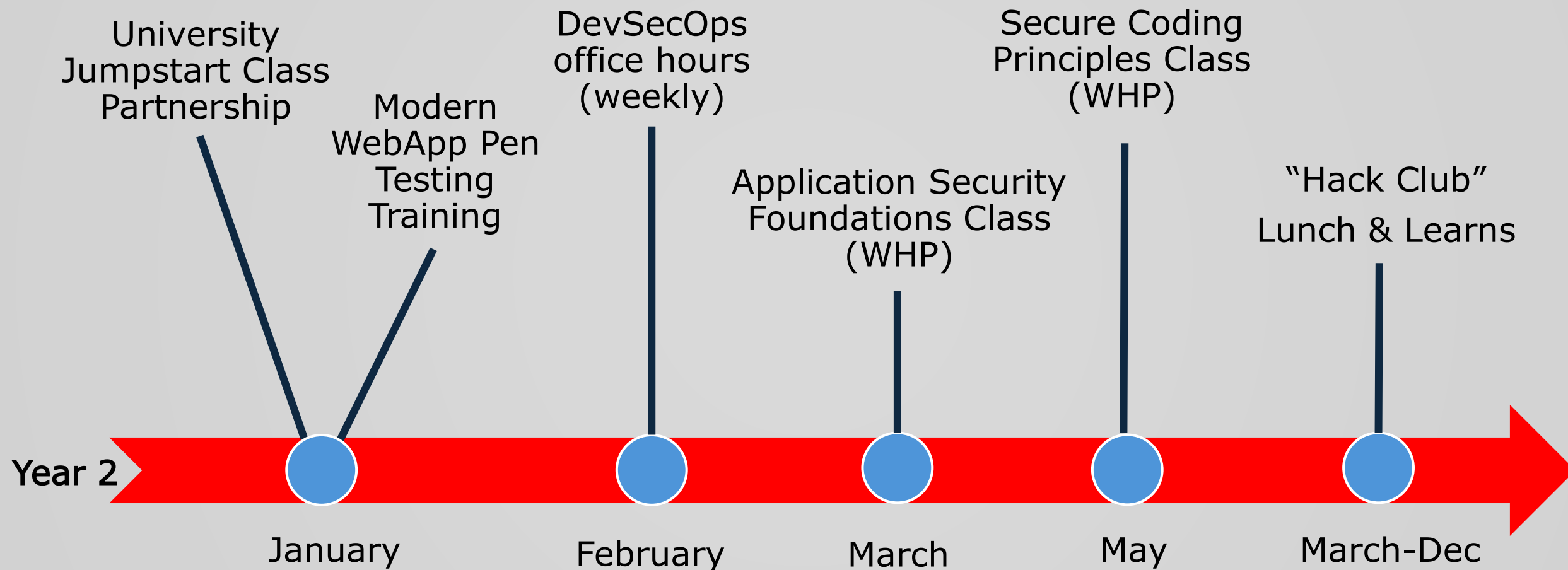




# EDUCATION APPROACH - YEAR 1

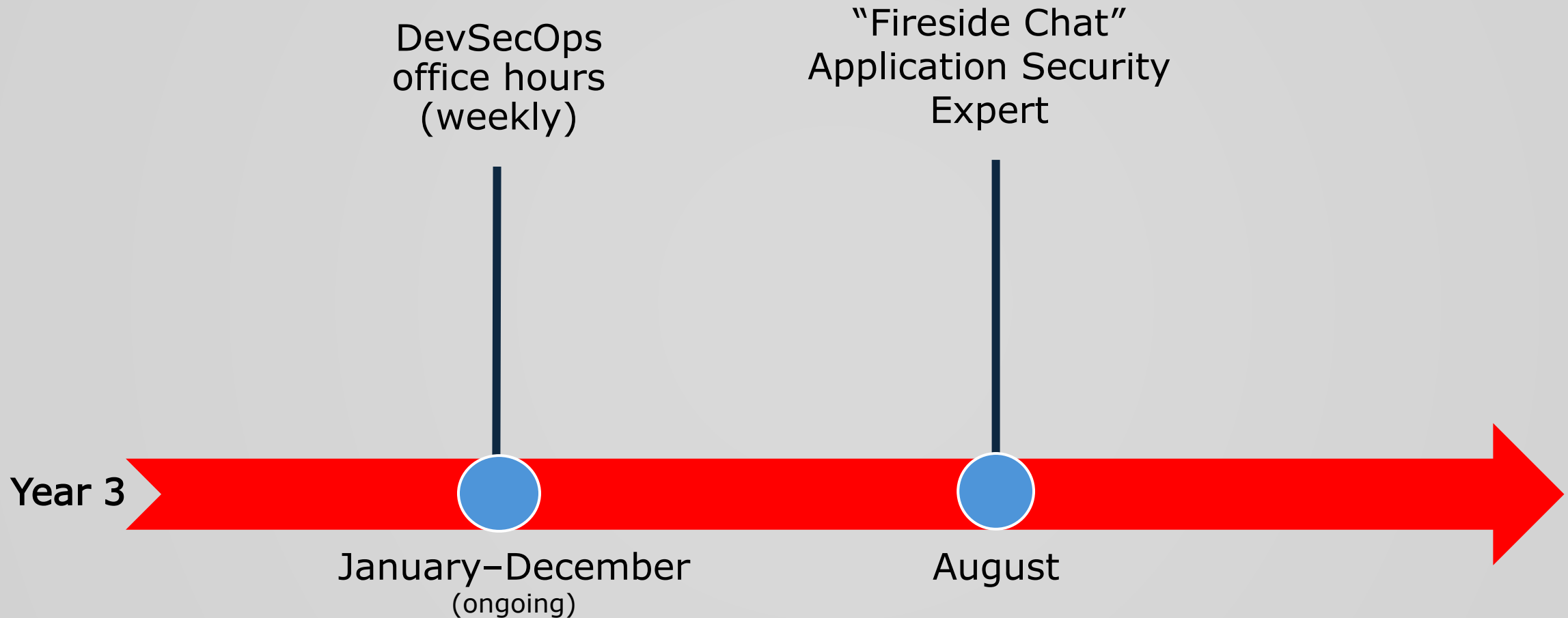


# EDUCATION APPROACH – YEAR 2





# EDUCATION APPROACH – YEAR 3



# MATRIXED TEAMS

## ✓ Mission Statement:

Drive security throughout the development and CI/CD process by leveraging Star Chamber's diverse **skills** and **perspectives** to achieve mutual agreed outcomes for the business.

## ✓ Star Chamber Members:

- Engineering
- Developer Services
- SREs
- Platform
- Security





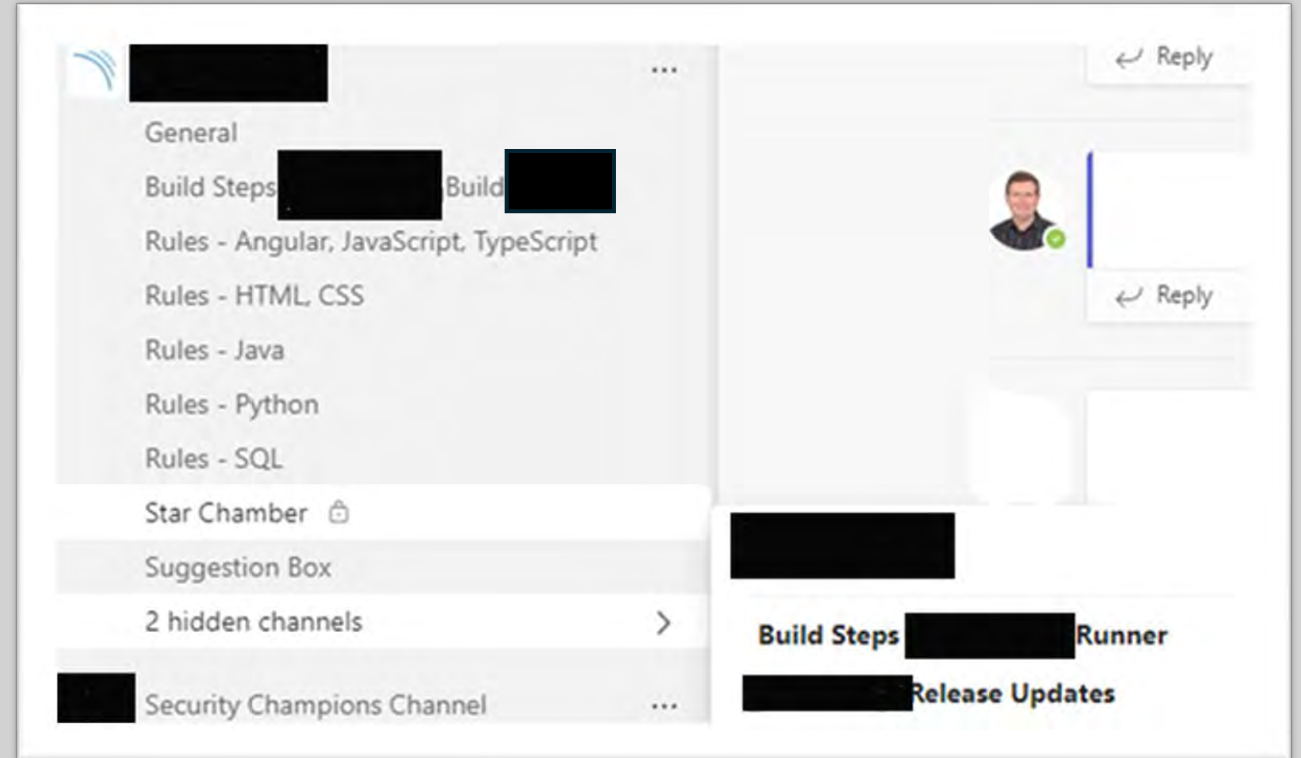
# MATRIXED TEAMS

## ✓ Security Champions Program

- Role: To “champion” security concepts throughout their respective dev workstreams

## ✓ SAST Teams Channel

- Facilitated expedited feedback



## ✓ Developer Services

- Enrich established “Paved Road” patterns w/ these secure coding principles
- Leveraged “office hours” to cross pollinate for learning opportunities

# 3.) PROGRAM HIGHLIGHTS

➤ Year 1

➤ Year 2





# HIGHLIGHTS – YEAR 1

## ✓ Decreased Application Attack Surface

- Influenced RGA/MC dev. programs to adhere to SAST tooling compiling breaker feature
- Result: 100% organization's #1 Revenue Generating Application July Year 1
- Result: >90% organization's #2 Revenue Generating Application Nov Year 1

## ✓ Increased Code Quality

- Helped deliver stable offerings to customers & upstream company
- SAST Dashboard Metrics

## ✓ Matured Processes

- Automatic exception process completed Aug Year 1
- DevSecOps roadmap validated by 3<sup>rd</sup> party Sept Year 1

# HIGHLIGHTS - SAST DASHBOARD





# HIGHLIGHTS – YEAR 2



## Process Improvement

- Compiler breaker compliance for all dev stream programs (operationalized: Q1 Year 2)
- Added **security evaluation points** into the CI/CD pipeline
- DAST solution researched & tested Q1 Year 2



## SAST “Hot Spot Champions” - Q1 Year 2

- Empowered selected developers to review & fix findings:
  - status of code smells
  - issues & security hotspots identified in their teams’ code

## 4.) SUCCESS PATTERNS

- Non-Technical
- Technical





# SUCCESS PATTERNS: NON-TECHNICAL - 1

- ✓ Partnering, Listening, Partnering, Asking Questions, Listening etc.
  - Reoccurring meetings with ALL engineering leadership (monthly/weekly)
  - \***"Secure code is a *form* of code quality."**
  - Seamless experience delivered to developers

- ✓ Training/Outreach
  - Web App Pen testing training → OK
  - Monthly Hack Club meetings & ILMS → Better
  - \*Secure Coding training → Best Approach!

# SUCCESS PATTERNS: NON-TECHNICAL - 2

## ✓ "Secret Sauce"

- Matrixed working groups- Star Chamber; Security Champions; DevSecOps Office Hours
- Word-Marketing - "Security Assessment/Evaluations Points" vs. "Security Toll Gates"
- Training dollars - Sourced by Security
- Measure Sentiment - Gauge & Report program value

## ✓ Manager/Leader Pro-Tips:

- #1 Recognize developers via employee internal recognition mechanisms
- #2 Enrich recognition write-up w/ company values + company mission statement
- #3 Calibration: highlight key developers' contributions to this security space



# SUCCESS PATTERNS: TECHNICAL

## ✓ Static Code Analysis Solution

- Low cost point/low entry point for CI/CD pipeline integration & adoption
- Purchased by Security
- Administered by Platform team

## ✓ Metrics

- Dashboard: DevSecOps program **value clearly articulated** delivered by near real-time status
- **Instant self feedback** led to application remediations

## ✓ Frictionless Experience Delivered

- Leveraged existing ITSM/Asset Management application - **automated & timed** ability to document exceptions to get “code out the door” to **meet business goals**  
(\*Always *be mindful of your organization’s risk tolerance w/ exceptions!*)
- **Feedback quickly** provided via DevSecOps “office hours”; SAST Teams channel; Dev Services “office hours”

# 5.) NEXT STEPS



➤ SBOM & DevSecOps





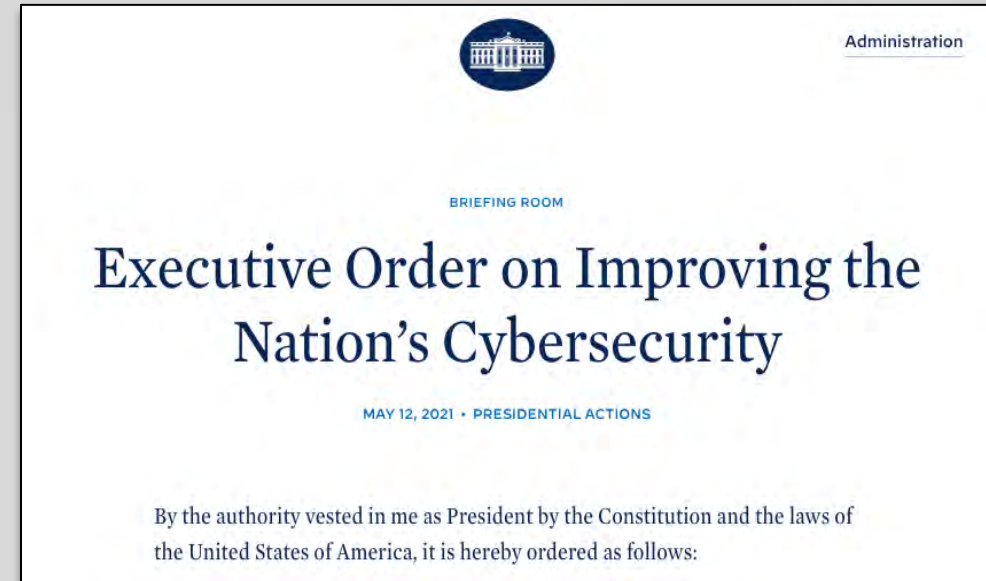
# SBOM, WHAT IS IT?

- ✓ **Software Bill Of Materials-** A structure list of components, libraries, and modules used in a build
- ✓ **Goal:** automated job built into the existing CI/CD pipeline tooling & process

Attribute	 <b>SPDX</b>	 <b>CycloneDX</b>
Author Name	(2.8) Creator:	metadata/authors/author
Timestamp	(2.9) Created:	metadata/timestamp
Supplier Name	(3.5) PackageSupplier:	Supplier publisher
Component Name	(3.1) PackageName:	name
Version String	(3.3) PackageVersion:	version
Component Hash	(3.10) PackageChecksum: (3.9) PackageVerificationCode:	Hash "alg"
Unique Identifier	(2.5) SPDX Document Namespace (3.2) SPDXID:	bom/serialNumber component/bom-ref
Relationship	(7.1) Relationship: DESCRIBES CONTAINS	(Inherent in nested assembly/subassembly and/or dependency graphs)

# DRIVING FORCES FOR SBOM ADOPTION

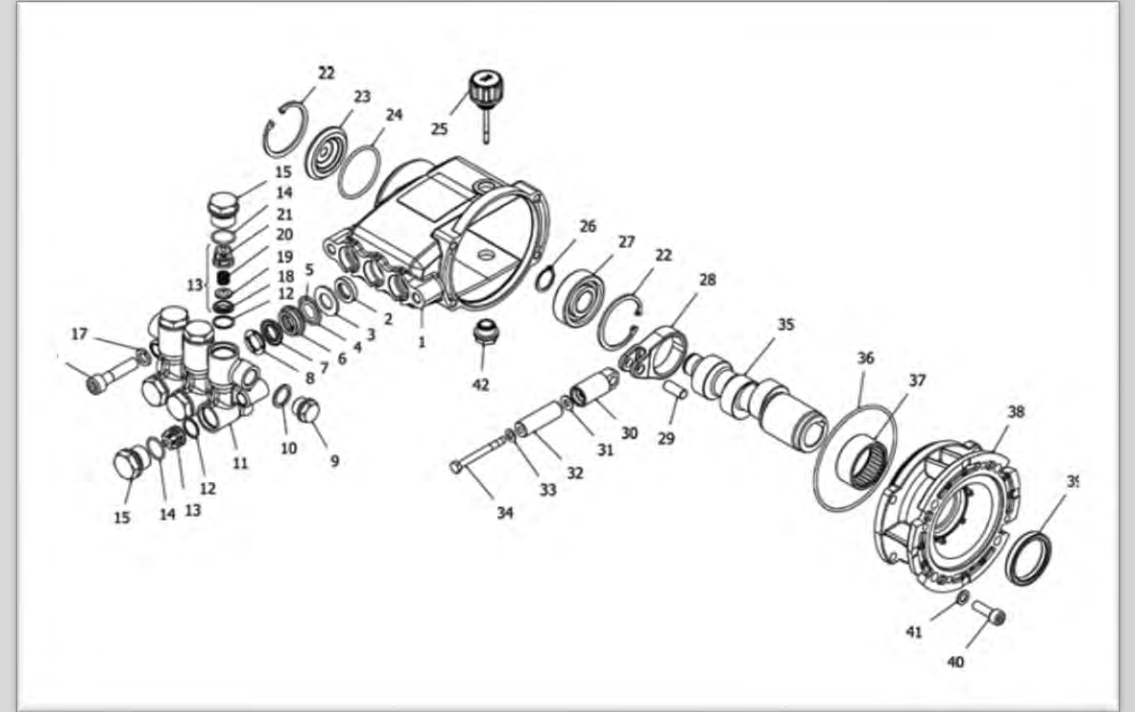
- ✓ Executive Order on SBOM
- ✓ Secure software supply chain
- ✓ Upstream customer expectations





# PREVENT ISSUES WITH SBOM

- ✓ Identify all software assets
- ✓ Detect vulnerabilities early and often
- ✓ Provide transparency to customers & internal teams
- ✓ Provide insights to engineering teams



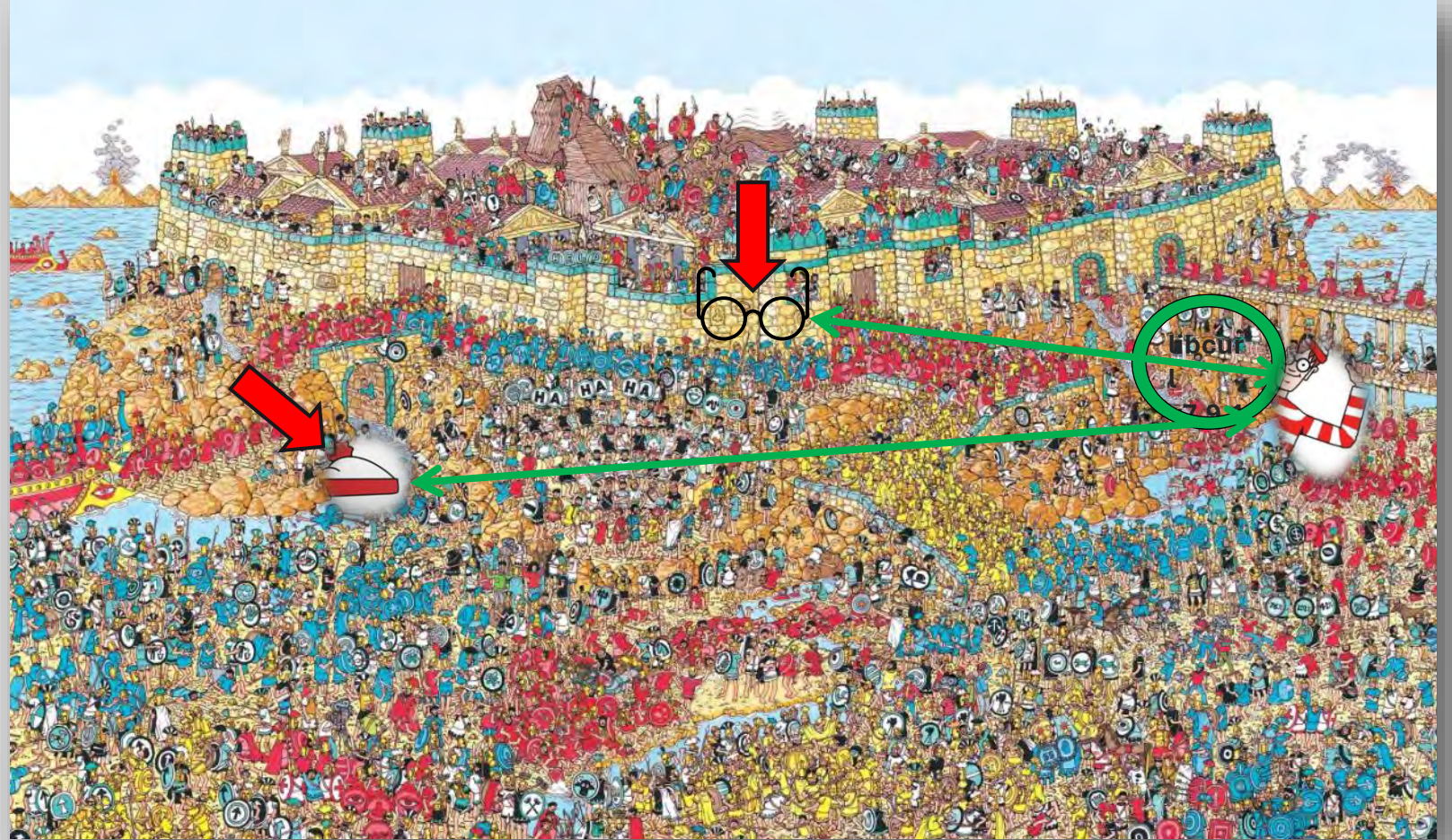
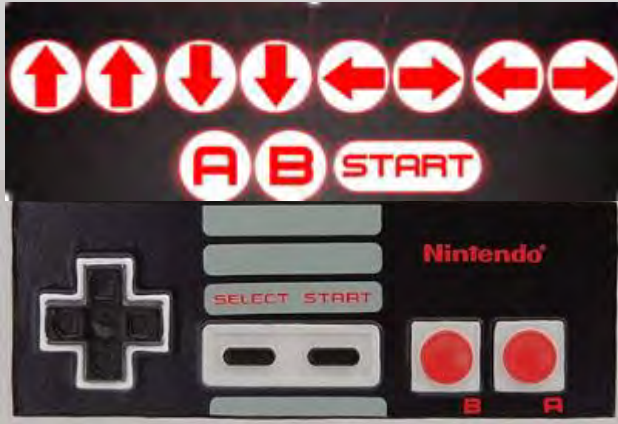


# HOW CAN THIS WORK IN REALITY?

✓ CVE-2023-38545 cURL & libcurl

✓ CVE-2023-38546 libcurl

SBOM IS THE  
CHEAT CODE





# 7 KEY TAKEAWAYS

1.) Identify & Remove Adoption Barriers

2.) Collaboration Centric

3.) "Word-Marketing"

4.) Training/Outreach

5.) Program Value Advertising

6.) Program Improvement Beyond "*What Good Looks Like.*"

7.) Document & Action "Your Future List"



# Thank You



Josh Hankins, Consulting CISO - CBTS  
CISSP | GPCS | GMON | GMOB

LinkedIn Contact

