Jonathan Davis

Cybersecurity Considerations of an Agile Development, Integration, and Test Environment

Agenda

- Introduction
- My Perspective
- The Seesaw
- So What’s DevSecOps?
- Defense-in-depth in-depth
- No one is happy, and that’s a good thing
- So, how?
- Questions

Introduction

Introduction

- Schooling
  - Bachelor’s Degree in Computer Science – IT
  - Working on Master of Science in Information Security Engineering
- Currently working for the MITRE Corporation
  - System Architect
  - DevOps Development Pipeline
    - Live migration to DevSecOps
- Chief Warrant Officer in the Army Reserves
  - 18 Years
  - Cyber Corps
  - 31U > 25U > 255A > 170A

My Perspective

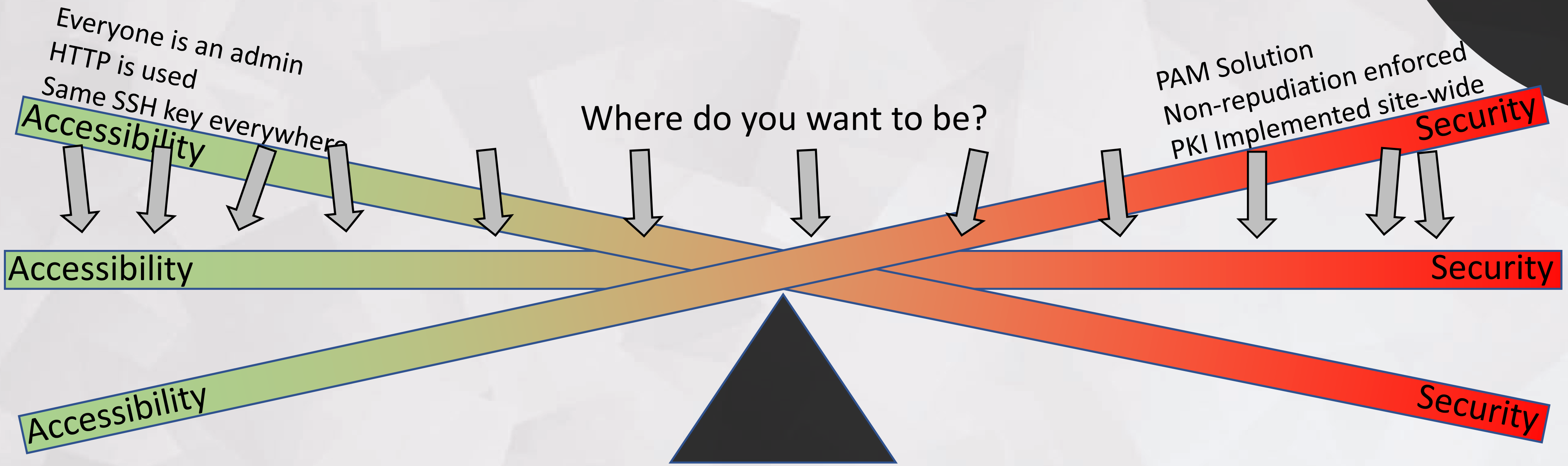
My Perspective

- Worked on WIN-T in PEO C3T
    - Desire to implement DevOps Strong
        - Understanding of DevOps, not so much
    - Paradigm shift, Major growing pains
        - Attempted mid-project change
        - Resulted in Agile WaterOps
- DISA
    - Started program with Agile in mind
        - Grew larger, testing automation became key, added to DevOps
    - Pipeline matured, became primary development environment
        - Security became paramount
            - ATO
            - STIGs
            - All
            - while
            - maintaining
            - a
            - Production
            - infrastructure

The Seesaw

The Seesaw

- All systems are a balance between ease-of-use and security
  - Mutually Exclusive
- Pick where you want to end up
  - Have reasons for each action
  - Have possible results if action isn't followed



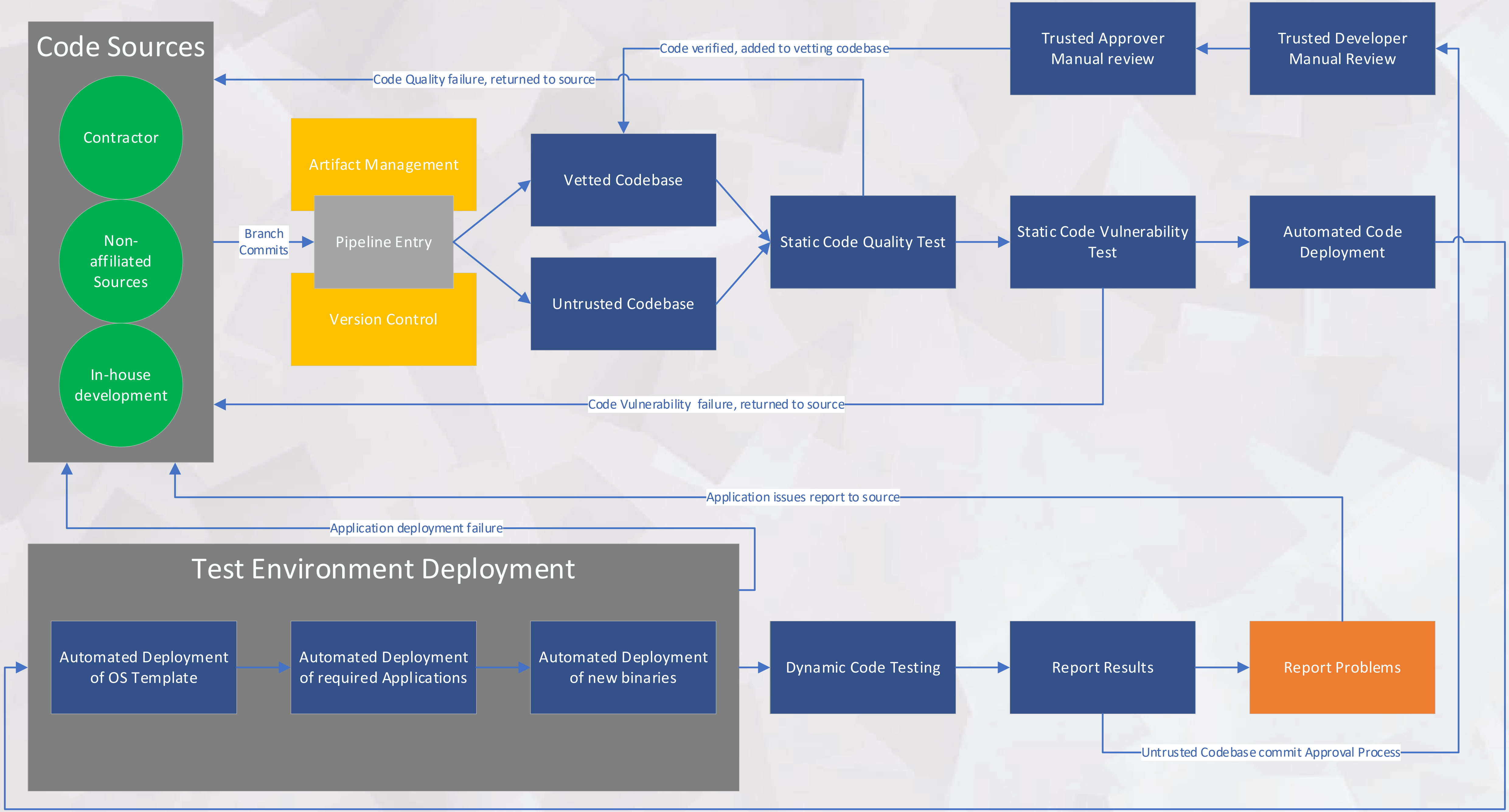
- Determining the desired security posture will dictate everything afterward
- Pivoting during Production is exceedingly painful, and not just for the admins

So What’s DevSecOps?

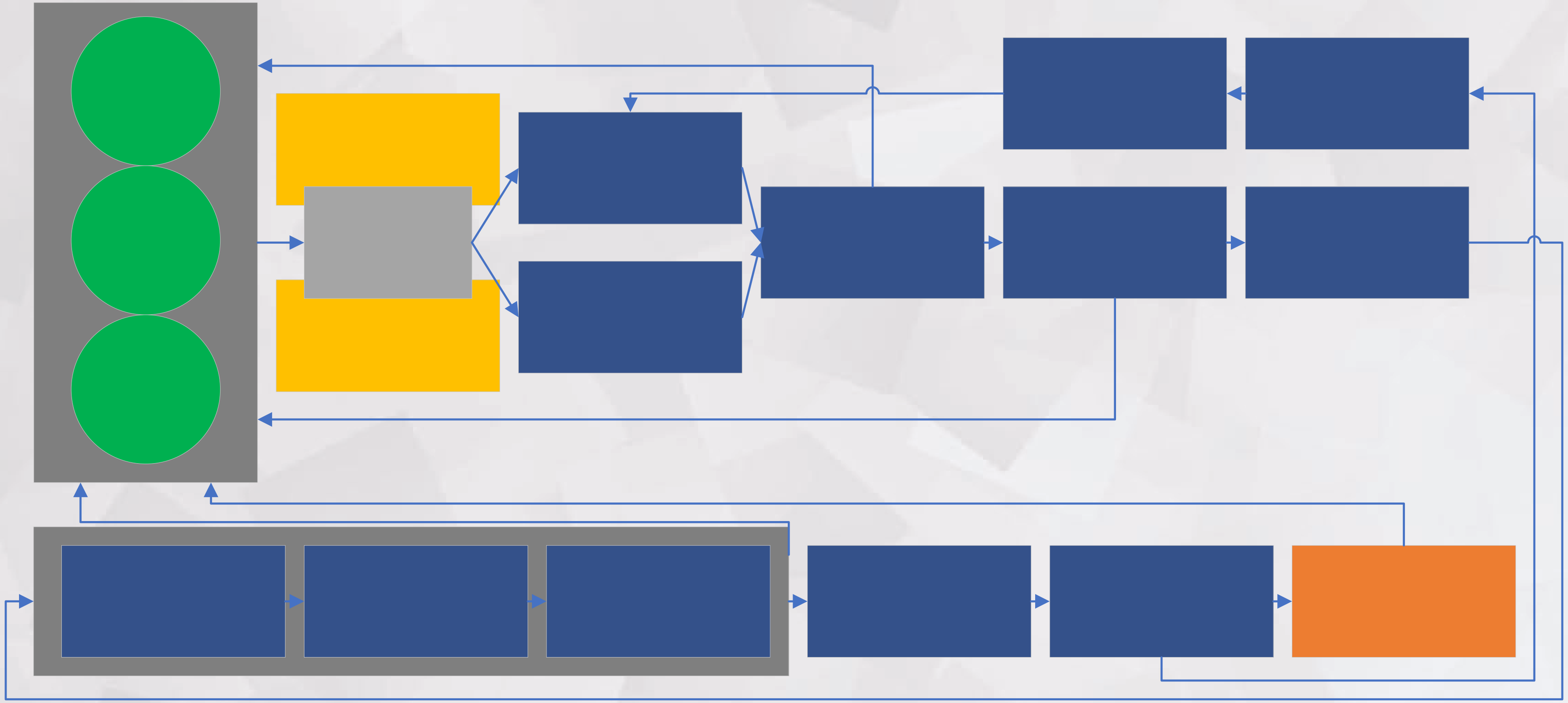
So What’s DevSecOps?

- Cattle, not pets
  - Databases are the only unique things
  - Be able to blow away and recreate any machine
- Known baseline, version-controlled
  - Base OS + Automation Engine + whatever App +STIGs
  - Dynamically build or have baseline ready
- “Pipeline”
  - Like DevOps, but add in:
    - Automated Code Checking
      - Static/Dynamic
    - Developer Isolation
    - Standardized Coding Practices
  - Not a “free flow”
  - Rather a “controlled stream”
- Surrounded by Security Controls

Defense-in-Depth in-depth


Code Sources
Contractor
Non-affiliated Sources
In-house development
Branch Commits
Artifact Management
Pipeline Entry
Version Control
Vetted Codebase
Untrusted Codebase
Static Code Quality Test
Static Code Vulnerability Test
Automated Code Deployment
Trusted Approver Manual review
Trusted Developer Manual Review
Code verified, added to vetting codebase
Code Quality failure, returned to source
Code Vulnerability failure, returned to source
Application issues report to source
Application deployment failure
Test Environment Deployment
Automated Deployment of OS Template
Automated Deployment of required Applications
Automated Deployment of new binaries
Dynamic Code Testing
Report Results
Report Problems
Untrusted Codebase commit Approval Process

Defense-in-Depth in-depth



VPN

Firewall

MFA

Bastion Hosts

OOB

PAM

Kitchen Sink

IdAM

Vulnerability Scanning

HBSS

SIEM

Backups

Monitoring

Defense-in-Depth in-depth

- Common Firewall rules
  - Geo-IP blocking
  - Controlled ingress/egress
- MFA
  - Soft Token, hard token, CAC
- VPN
  - Alternate access methods for secured computers
    - Web portal into environment
- Bastion Hosts
  - Windows and Linux
- OOB for Admin access (Double MFA?)
  - Separate compute stack?
- PAM
  - All users or just admins?
- LDAP / AD
  - Constant management

Defense-in-Depth in-depth

- Vulnerability Scanner
  - ACAS
  - Nessus
- HBSS
  - Centrally managed
- SIEM
  - Log aggregation
  - Log Correlation
- Backup
  - Veeam
  - Data-at-rest encryption
- System Monitoring
  - WhatsUp Gold
  - SolarWinds
- Update Method
  - WSUS / Artifactory
- Kitchen Sink
  - Enamel
  - Steel

Defense-in-Depth in-depth

- Any one of these could be a career
- This list is not exhaustive, but it is not all needed
  - What is the requirement?
- Use the desired end-state to steer
  - Where on the seesaw?
- Any half-implemented solution is worst than no solution
  - Possible Exploits
  - Data-at-Rest
- Every one of these expands your possible attack vectors
  - Except the sink

No one is happy, and that’s a good thing

- Everything is a balance between ease-of-use and security
  - Mutually-Exclusive
- More hoops to jump through = More pushback
  - Will seek the easiest path
    - SSH Tunnels
    - Local Environments
    - Alternative collaboration tools
- Moving from a lower to higher security model is much harder
  - “It worked fine before”
- Admins have to work with multiple authentication schemes
- System integration becomes difficult as firewalls are put in place
  - Downtime
  - Maintenance Windows
- Everything is harder for everyone
  - Including the adversary

No one is happy, and that’s a good thing

Remember: This should be status quo for industry, not an anomaly

So, how?

So, how?

- First and foremost – Get Senior Leadership backing
  - Define their role for them (likely disciplinary)
- Put a line in the sand
  - No other features until A, B, and C are fully functional
- Put the key collaboration services inside the enclave
  - Collaboration Suite
    - SharePoint
    - Alfresco
  - Issue Tracking Suite
    - TRS
    - Bugzilla
  - Chat
  - Help Desk
- Provide the carrot
  - Specialized software tools
  - Resource access
  - Working enclaves
  - Recognition

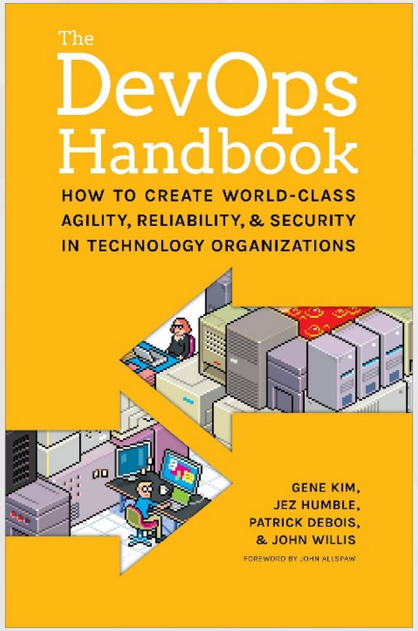
So, how?

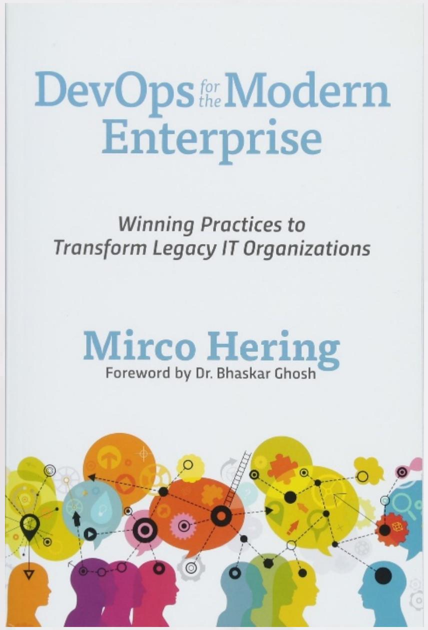
- Know your audience
  - Communicate the reason for actions, don’t just lock things down
  - Expect push-back, and deal with it for what it is – A desire to do it the easy way
- Share the success stories
  - “We observed over 3500 attempts against our ingress point to exploit XYZ, all were successfully blocked”
- Share the horror stories
  - “XYZ was just exploited in the wild, 200,000 systems impacted. All files deleted. Fortunately, it can’t touch us because of the ABC security solution we put in place”

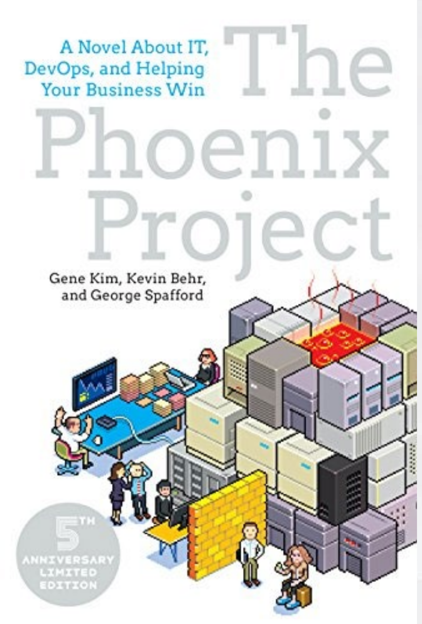
Questions?

- Suggested Reading:
  - The Phoenix Project - Gene Kim, George Spafford, and Kevin Behr
    - Novel about DevOps
    - ISBN: 978-1-94-2788-29-4
  - DevOps for the Modern Enterprise – Mirco Hering
    - Transforming legacy IT organizations
    - ISBN: 978-1-94-2788-19-5
  - The DevOps Handbook - Gene Kim, Jez Humble, and Patrick Debois
    - How to do DevOps
    - ISBN: 978-1-942788-00-3

Questions?


The DevOps Handbook
HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS
GENE KIM, JEZ HUMBLE, PATRICK DEBOIS, & JOHN WILLIS
FOREWORD BY JOHN ALLSPAW


DevOps for the Modern Enterprise
Winning Practices to Transform Legacy IT Organizations
Mirco Hering
Foreword by Dr. Bhaskar Ghosh


A Novel About IT, DevOps, and Helping Your Business Win
The Phoenix Project
Gene Kim, Kevin Behr, and George Spafford
5TH ANNIVERSARY LIMITED EDITION