DoD Enterprise DevSecOps Community of Practice

September 9, 2021
Agenda

• Welcome and Opening Remarks
  Mr. Jason Weiss, OSD, DoD CIO

• SpaceCAMP
  1st Lt. Christopher Paul, Agile Space Operations Software Manager

• F-16 In-Flight Software Update
  Capt. Justin Marsh, Chief, F-16 Advanced Programs Section (WP)

• National Background Investigations Services (NBIS)
  Mr. Jeffery Smith, Program Manager NBIS

• Open Discussion

• Closing Remarks
  Mr. Rob Vietmeyer, OSD, DoD CIO
THE PREMIER SOFTWARE FACTORY FOR SPACE

https://spacecamp.il2.dso.mil/#/home
F-16 Rapid EW Reprogram Demo and Software for DoD DSO Community of Practice

Billy Hagon & Annette Becker
9-Sep-21
Advanced Battle Management System (ABMS)

Multi-Domain Operations
powered by
Advanced Battle Management System

0 Digital Architecture Standards, Concepts
1 Sensor Integration
2 Data
3 Secure Processing (U, S/REL, S, SCI, S/SAR, TS/SAR)
4 Connectivity
5 Apps
6 Effects Integration

GLOBAL

LOCAL

Pound for Pound, Best Fighter Around!
F-16 Roadmap to ABMS

- **Phase 1 Demo (June/July 2021)**
  - F-16 Airborne MDF Update via VMF
    - 1 A/C with 1 Back Up
    - CDU, DVR, ALQ-213 OFP
      - Updated from M7.3 Tape 7 core baseline

- **Phase 2 Demo (~2022)**
  - F-16 Ground Test to Receive/Display Broadband Full-Motion Video via Starlink

- **Phase 3+ Demo (2023+)**
  - Testing of Future Conformal Antenna
Phase 1: Overview

Receive K-Series Free Text message via ARC-210 radio
  - Free Text message defined in MIL-STD-6017D in K01.1
  - Up to 1400 bits
  - Current M7.3 C37 candidate
  - Free Text message contains information to update ASQ-213 MDF

Airworthiness submitted under: HL-718-05-37 (CDU cA baseline)
  - No change to core OFPs (MMC, MFDS, Mission planning, UFC, ARC-210)
Phase 1: Workflow

1. CDU receives K01.1 message via ARC-210 (existing capability)
   a. Fielded in 2015 with SCU 8 for Pre Block – SCU 8
   b. Prioritized/Approved candidate C37 for M7.3 for Post Block – tested in Tape 5, 6, & 7
2. CDU compiles existing MDF to a new ALQ-213 binary with new threat data
3. CDU writes updated binary to appropriate DVR location
4. Pilot power cycles ALQ-213 to update MDF binary
5. New threat information will allow correlations with existing/detected 56M

Note 1: Blue symbolizes existing interfaces (no change); Orange symbolizes ABMS demo
Note 2: CDU, DVR, ALQ-213 will have unique OFP IDs for ABMS demo
Phase 1: CDU Workflow

1. From the K01.1 VMF message, threat update received
2. CDU amends existing MDF file with threat update
3. CDU executes Compile batch file to generate new binary file
4. CDU executes Build batch files to generate new binary file
5. CDU executes Deploy batch files to move new binary file to DVR folder
6. Pilot is notified of successful compile and power cycles ALQ-213 to load new file
Phase 1: CDU App Development
Phase 1: CDU Pipeline

1. Code committed or scheduled task triggered
2. Code is compiled and built into binaries
3. If attempted code commit doesn’t successfully compile the commit is rejected
4. Unit tests are run against binaries
5. If attempted code commit doesn’t successfully pass tests the commit is rejected
6. Code repository is updated with committed code update
7. Static code analysis is completed and results are sent to the database
8. Code metrics are calculated and stored for analysis
9. OFP is sent to storage location to be used as needed

Pound for Pound, Best Fighter Around!
Phase 1: Pilot Interface

Pound for Pound, Best Fighter Around!
Phase 1: Pilot Interface

V-TXT from CALLSIGN
New ABMS Message

New OSB

Update MDF
CDU Software Dev History

- Initially Waterfall (SDCRs, DCRs, SCRs)
- Pilots drove change to Scrum approximately 5 years ago
- User Story/Feature/Epic tracking
  - Excel Spreadsheet  MS Access DB  Team Foundation Server (Azure)
- Code repo
  - Subversion (SVN)  GIT  GIT in Azure
- Automated builds
  - Nightly Coverity (Windows Workstation)  Nightly TFS Coverity  TFS Pipelines
- Automated testing
  - None  Optional  Part of Definition of Done & peer reviewed
F-16 Software Journey

- Continuous ATO (cATO)
- DevSecOps
  - Continuous Monitoring
  - Telemetry Capture
  - Service Mesh
  - Secure Containers
- Shift Cybersecurity Left
- DevOps
- Continuous Integration
- Agile, Microservices, Test Driven Development
- Iterative with Hybrid or SOA Monolithic Architectures
- Monolithic Architecture, Manual Processes

End to end cycle time – Design to Delivery

Significant investment of time, effort and tools are required to achieve high DevSecOps maturity

Today

Difficult

Adoption Challenge
CDU Development Process

1. An idea is born - WIT Enhancements/NCCS
2. Pilots vote to set priority (as needed)
3. Story boarding (as needed)
4. User stories (US) initiated
5. US are further fleshed out during team Grooming/Planning meetings
6. Team commits to US as it become the highest priority
7. Code development
8. US testing is completed by CDU System Testers (OFP generated through Test Stand Pipeline)
9. Peer Review/TFS Pull Request managed and enforced through TFS Branch Policies
10. Product Owner confirms the Definition of Done and the acceptance criteria has been met
11. Code check in (code checked through Test Stand Pipeline)
12. System documentation updated per actual developed capability
13. Safety of Flight is completed before release is sent to Flight Test
14. Released to Flight Test (OFP generated through Release Pipeline) (every 2 weeks)
CDU Pipeline Tools

- Azure DevOps (TFS) – work item management, peer reviews, pipeline automation
  - Why: ease of set up, peer reviews, integration with other Microsoft products
- GIT – code repo
  - Why: flexible branch management
- NUnit – unit testing
  - Why: ease of testing a range of values versus MSTest; and less code rewrite versus XUnit
- Coverity – code analysis
  - 309th standard
- Appium (not integrated yet) – user interface testing
  - Why: ease of use over Test Stack White
CDU Multiple Pipelines

- Nightly
  - only pipeline that includes Coverity due to run time
- Test Stand
  - runs each time a developer pushes new code into Trunk
  - is used by developers to generate OFP for test stand
- Release
  - used to generate OFP being sent to Flight Test
CDU Nightly Pipeline

1. Most current code is pulled from Trunk
2. Additional pieces needed for successful build are gathered from other repos and network locations
3. OFP Build, Coverity analysis, and code metrics analysis are performed
4. Unit tests are run
5. Results from Coverity are loaded to the Coverity server
6. Email with code metric data sent
CDU Test Stand Pipeline

1. Developer pushes code to repo or chooses a branch to target
2. Code to be pushed is pulled from the appropriate location
3. Additional pieces needed for successful build are gathered from other repos and network locations
4. Build is attempted
5. Unit tests are run
6. If build is successful and unit tests pass Azure allows developer pushed code to be committed to code repo
7. Fully configured build for test stand testing saved to network drive
CDU Release Pipeline

1. Pipeline accepts new OFP ID
2. Most current code is pulled from targeted branch
3. Additional pieces needed for successful build are gathered from other repos and network locations
4. Appropriate flags are set for aircraft configuration
5. OFP ID is set
6. OFP is built
7. Unit tests are run
8. Build output is cleaned up to ensure that only needed files are part of the release to be sent to flight test
9. Configured OFP is saved to a network location
National Background Investigation System (NBIS) Program Overview for the DoD CIO DevSecOps Community of Practice

September 9, 2021

Mr. Jeff Smith, NBIS Executive Program Manager
Mr. Ben Cox, NBIS Chief Engineer
Agenda

• NBIS Program Overview
• NBIS AWS GovCloud Environments and Code Progression
• Cyber Excellence Cell (CEC) - Response Team
• Cyber Excellence Cell (CEC) - Proactive Team
• Future Initiatives
NBIS is designed to provide a secure, end-to-end IT architecture for the USG Personnel Vetting Enterprise. Besides replacing the Legacy IT architecture, which only assisted with background investigations, NBIS will cover every component of personnel vetting from initiation to adjudication, including the newly established continuous vetting requirement of Trusted Workforce 2.0, and will be available for all USG Departments and Agencies.

- 2015 OPM data breach resulted in NBIS requirement (legislation & EO)
- NBIS failed to meet projected deployment timelines
- DCSA re-baselined NBIS and restructured the program to move delivery left
- Mitigating impacts of the FY21 $12M mark
- NBIS met its first six post re-baselining milestones on schedule
NBIS 101 - Functionality

NBIS will provide a secure end-to-end IT architecture for USG Personnel Vetting Enterprise

- Enhance and replace functionality contained in seven disparate Legacy IT systems into a “unified platform” in support of strategic Business Process Re-engineering (BPR) objectives
  - improved Investigative Records Repository (iIRR)
  - Joint Personnel Adjudication System (JPAS)
  - Mirador
  - Defense Information System for Security (DISS)
  - Defense Central Index of Investigations (DCII)
  - OPMs Legacy IT Background Investigations System
    - eQIP/PIPS/OPIS/FWS/NFWS
  - Secure Web Fingerprint Transmission (SWFT)
- Support Continuous Vetting (CV)
- Sufficient Flexibility to Accommodate Policy and Environment Changes
NBIS Roadmap to Operational Capability

NBIS Operational Capability

Phased Deployment / NBIS Onboarding

Legacy System Sunset

Decommission JPAS

Phased DISS Transition to NBIS (Based on NBIS Deployment)

Data Migration

Decommission DISS

Decommission Mirador

Decommission Legacy IT BIES
Cyber Excellence Cell – Response Team

Security Incident Event Management (SIEM) Highlights:

- Real time event collection, analysis, and response
- Models and integrates with C5ISR and DISA CSSP capabilities
- Rapid incorporation of new indicators of concern
- Advanced Analytics – Machine Learning and Hunt
- Experienced cyber analysts from Pentagon, USCC, NSA, and Commercial

*Practice, Practice, Practice!*
Cyber Excellence Cell – Proactive Team

- Formal Red Team Event Management - Twice yearly
- Internal Blue Team
  - Discovery
  - Automated Pen Testing—CATO
  - Detection and Analysis
  - Vulnerability Reporting
  - Enumeration of Test Environments
Future Initiatives—DevSecOps

• Automation and Orchestration - Removing humans from processes
  – Automated Patching
  – Test Harness for Automated User insertion in Testing
  – Security, Orchestration, Automation, & Response (SOAR) – Orchestration of Event Actions
  – Software Defined Networking (SDN)—Automating the creation of Servers, VPCs, Environments, etc.
  – Configuration Management - AWS Config

• OS/Application Standardization through Containerization
  – Ironbank Secure Containers
  – Fargate

• Metrics
  – Deployment Frequency
  – Lead Time for Changes
  – Change Failure Rate
  – Time to Restore Services
Questions
Code Progression – Documented & Practiced Workflows

- Development
- Test Merge
- Integration Test Deployment
- PMO Test Deployment
- Operational Test
- Training
- Production Deployment
- Operating System Patching
Questions?
Closing Comments:

Mr. Rob Vietmeyer, OSD, DoD CIO, Cloud and Software Modernization
September 28 @ 1:00-3:00pm ET

Please join Mr. Nicolas Chaillan, the DAF Chief Software Officer, for his final Ask Me Anything as the Department of the Air Force Chief Software Officer.

Mr. Chaillan will take live questions and will do his best to help you or direct you to the right folks one last time.

Please join us, please share! This event is open to anyone.

Link: https://us02web.zoom.us/j/458527876
Contact Information

DevSecOps Mailbox  osd.devsecops@mail.mil
MilSuite Site  https://www.milsuite.mil/book/groups/dod-enterprise-devsecops
Air Force Site  https://software.af.mil/