From Static ATOs to Assured Cybersecurity: Understanding the Modern Software Supply Chain



How Platform One and Iron Bank are helping the DoW understand and act on software supply chain risk

PLATFORM ONE

Lee C. Lambert
Capt, AFLCMC/HNCX
Iron Bank Value Stream Lead, Platform One

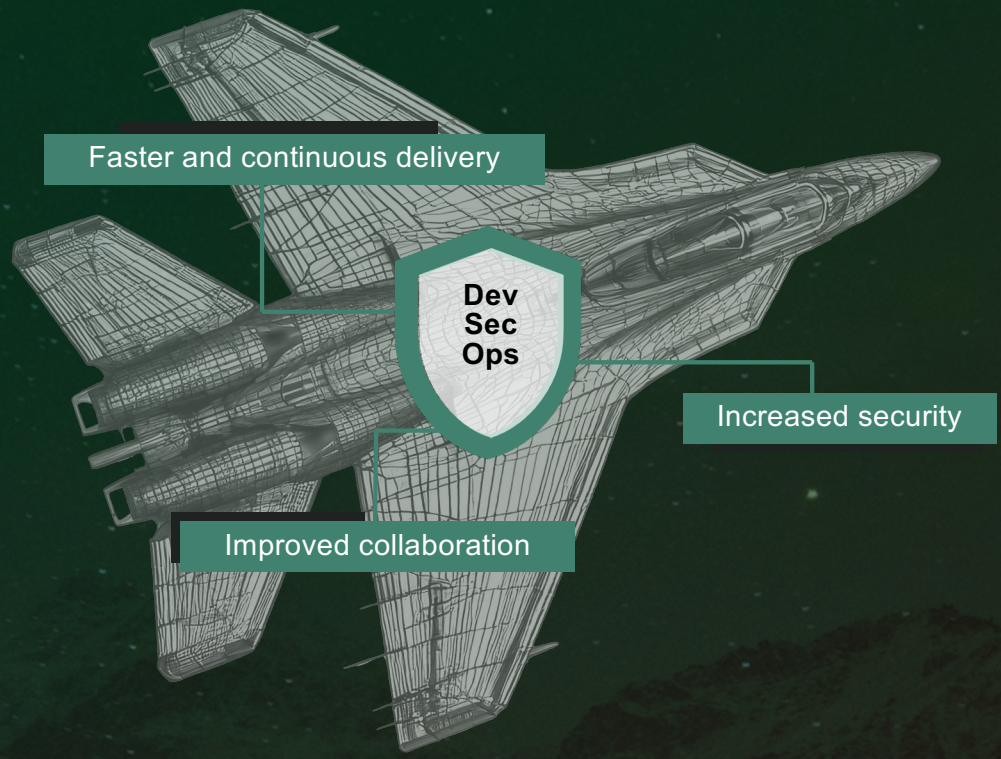
Overview

- About Platform One (P1) and software supply chain relevancy
- The Problem: ATO's, CtFs, cATOs
- The Foundation: Iron Bank & Containers, the world of open-source software
- The Solution: Data Mesh Environments (DME)

Securing Advantage With Software

All modern DoW systems, including weapon systems, are software-intensive.

Platform One (P1) is the U.S. military's software delivery engine. We provide open-source tools and enterprise solutions for teams to build, deploy, and secure better software at scale.



PLATFORM ONE

Relevancy today: the changing threat

Securing America's Defense Supply Chains and Ensuring Domestic Acquisition of Critical Materials.

The White House.
Presidential Action, July 2026
Executive Order 14415

Section 1. Policy. The United States military is the most effective and powerful fighting force on the planet. It fields the most advanced weapons systems and technologies in the world, utilizing cutting edge equipment to dominate the modern battlefield. To continue this dominance in an era of renewed great power competition, the United States must secure its supply chains against physical, cyber, and economic subversion. It is the policy of the United States that not only the finished equipment deployed by our military, but also the critical materials and components necessary to manufacture, maintain, sustain, and repair that equipment, are sourced domestically or from allied nations.

PLATFORM ONE

The Problem

ATOs, CtFs, cATOs

PLATFORM ONE

The ATO: a laborious, point-in-time paper trail

An Authority to Operate is a signed decision by an Authorizing Official to accept a system's risk — built on the NIST RMF and SP 800-53 controls.[1] It authorizes the whole system, not a single container.

1 Prepare → 2 Categorize → 3 Select → 4 Implement → 5 Assess → 6 Authorize → 7 Monitor

18
different offices required by Step 1 alone[2]

1–2 yr
typical authorization period before full re-authorization

2 yr
Iron Bank's 2026 ATO — a hard-won exception, not the norm

Meticulous documentation + continuous monitoring — for a decision that then freezes for a year or more.

PLATFORM ONE

Certificate to Field & Party Bus: fielding in ~30 days

Certificate to Field[1]

A CtF rides an existing ATO: software that falls within an already-authorized boundary can be fielded without standing up a brand-new ATO — expediting deployment while inheriting the accredited controls.

Platform One · Party Bus[3]

A managed DevSecOps platform-as-a-service that uses its ATO to field DoD applications fast.

up to 2 yr

traditional deploy

30 days

via CtF on Party Bus

PLATFORM ONE

cATO is the answer on paper — nobody in the DoW has one

Traditional ATO

- Snapshot in time
- Re-papered every 1–2 years
- Risk accepted, then frozen
- Blind to daily change

Continuous ATO (cATO)[4]

- Continuous monitoring
- Demonstrated system maturity
- Active cyber defense
- Real-time risk visibility

The reality gap: Currently unaware of any DoW organization that has a current cATO. Platform One is currently working toward one. The model exists, but the risk acceptance of the model is a dynamic shift in how the DoW has operated for years.

ATO is a snapshot in time, software change is continuous.

The traditional authorization processes assess controls and evidence at a defined point in time. It does not certify that the risk stopped changing the moment it was signed…*because risk isn't staic, it changes as fast as second-by-second*. The challenge is that software continues to evolve after that assessment:

01 Container images are rebuilt

02 Packages and dependencies change

03 New CVEs are published

04 Exploitability information evolves

05 Application code is added

06 Runtime behavior changes

07 Upstream maintainers issue patches or new releases

PLATFORM ONE

The Foundation

Containers, open source, and the ground truth of the supply chain.

PLATFORM ONE

Software depends on a vast array of shared code

Open-source software has supported commercial and government systems for decades. What's changed is the degree of dependency, reuse, automation, and connectivity - which increases the risk.

70% - 90%
of code relies on open-source software[5]

Up to 98%
of commercial codebases contain open-source components[6]

More than 1,000,000
packages represented within Iron Bank Repositories

1 software package
can affect hundreds or thousands of downstream container images

PLATFORM ONE

The unit of modern software is the container image

APPLICATION CODE

Your mission software

LIBRARIES & RUNTIMES

Open-source packages & dependencies

BASE OS

Minimal operating-system layer



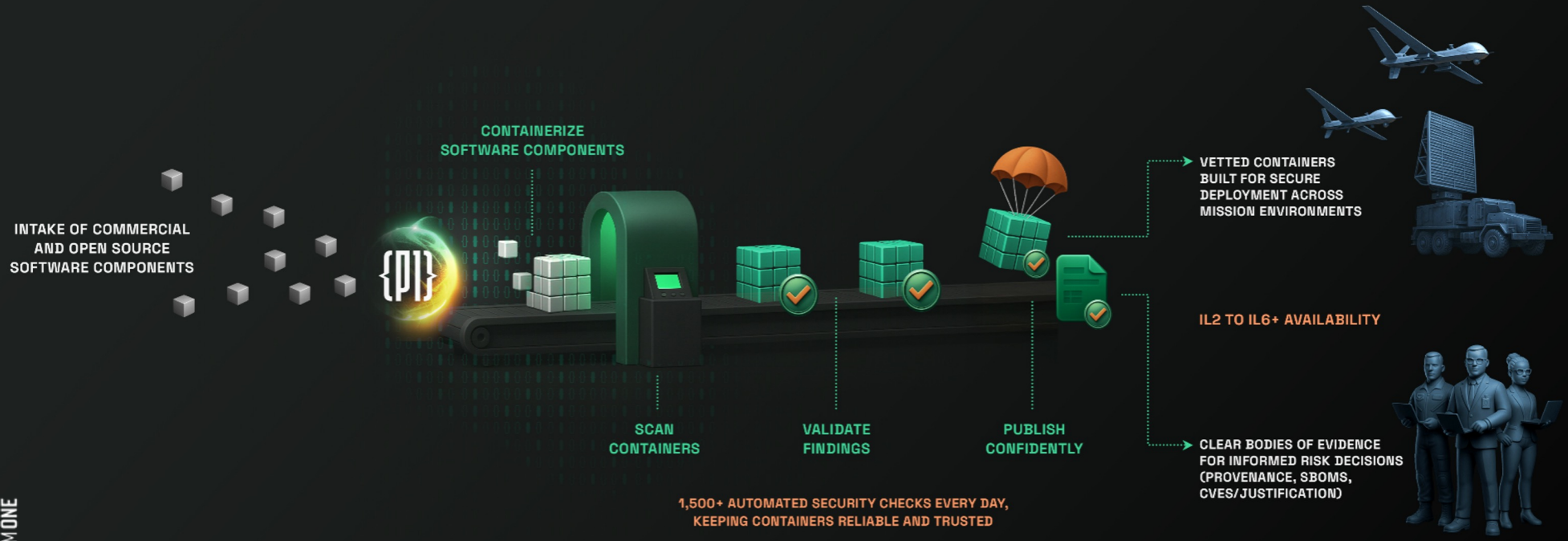
Container image[7]

A standardized and layered package of files, binaries, libraries and config — no full OS. Spins up fast, small footprint, platform-agnostic. Ideal for scalable, cloud-native microservices.

PLATFORM ONE

{IB} IRON BANK

What Iron Bank Does



1,500+ AUTOMATED SECURITY CHECKS EVERY DAY, KEEPING CONTAINERS RELIABLE AND TRUSTED

PLATFORM ONE

Why Iron Bank was built?

Iron Bank created a reusable foundation for software supply chain assurance.[3]

It provides a government-operated third-party process for vetting and distributing container images.

- Scanning through multiple enterprise vulnerability scanners
- Software Bills of Materials identifying packages and components
- CVE findings associated with identified software
- Security attestations and provenance information
- Vulnerability justifications and exploitability context
- Acceptance Baseline Criteria (ABC) & Overall Risk Assessment (ORA) scoring
- The Vulnerability Assessment Tool, or VAT, for viewing and evaluating results
- Recurring image rebuilds that incorporate available updates and patches

Instead of every program independently repeating the same work, Iron Bank creates security evidence that can be reviewed, reused, and incorporated into downstream risk decisions.

PLATFORM ONE

Why software supply chain risk is a mission issue

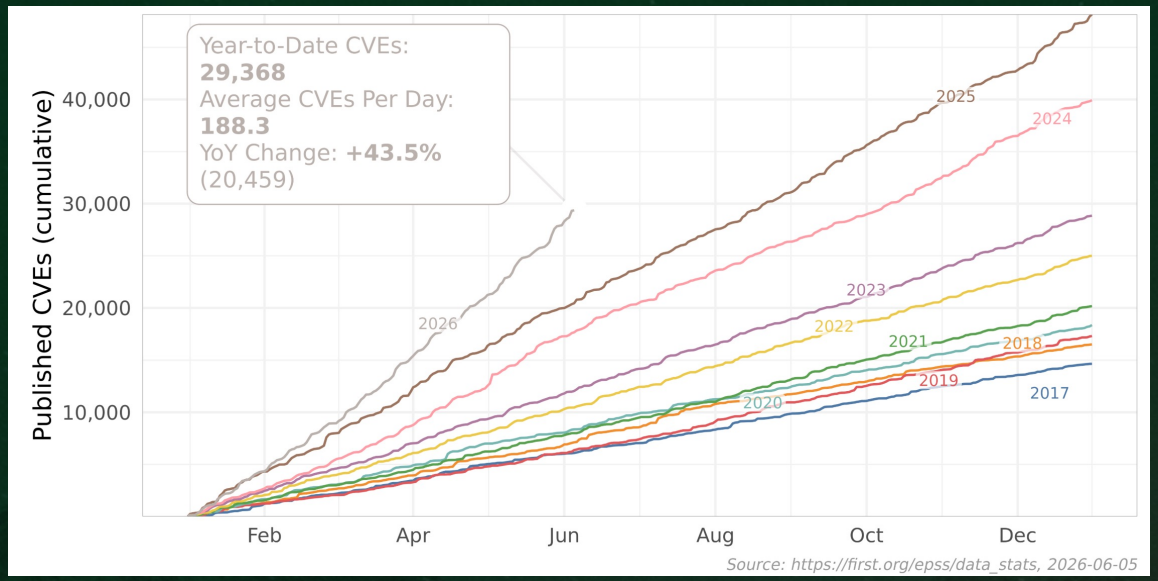
Software Components	Risks
Code: Application & Open-Source	**Origin:** Who created each part? Where does the contributor live? Who do they work for?
Dependencies: OS, Libraries, Containers	**Build Integrity:** What code bases are foundational to the software and how were they made? What vulnerabilities leave me open to attack?
Process: Build Tools & Pipelines	**Chain of Custody:** How was it delivered and has delivery methods changed or been compromised?

The software supply chain is more than a codebase or boxes to check. It is inherently human — and the human element can matter as much as the code.

The XZ attack was social, not just technical, an ATO wouldn't see that…[8]

XZ ran safely for years. The vulnerability wasn't in the code — it was in the trust model.

CVEs are exploding — and AI is pouring fuel on it



CVEs published in 2026, as of June 2 *Source: FIRST / EPSS*[9]

+40% year-over-year growth in tracked CVEs, 2025 → 2026

~188 new CVEs identified every single day

29,368 Year to date CVE's as of June 2026

LLMs (Claude's Mythos, OpenAI's Daybreak)[10] *are accelerating both discovery — and disclosure volume.*

Different scanners may produce different results → Scanners intentionally generate false positives to avoid missing any real vulnerabilities → A CVE may exist in a package but may not be exploitable in the deployed configuration → Static image scans do not show what an application does at runtime → New vulnerabilities can be disclosed after an authorization package has been completed

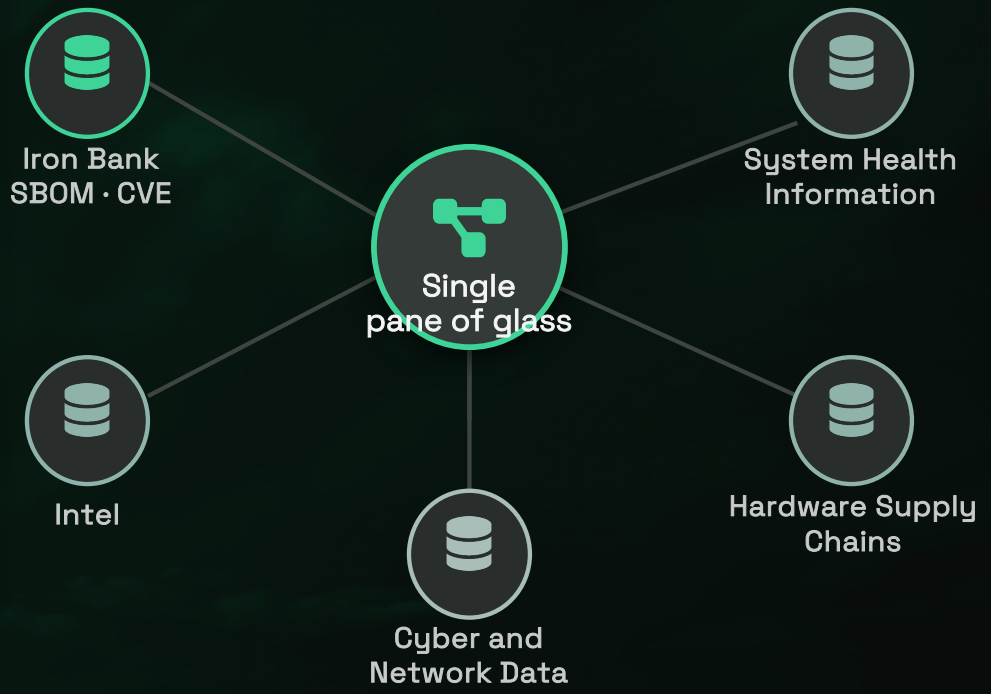
The Solution

Data Lake, Data Mesh Environment (DME)

Data Mesh Environment (DME)

> *Like the ceramic tiles in mosaics, these individual warfighting platforms are put together to make a larger picture, or in this case, a force package.*
>
> — Stew Magnuson, DARPA, on Mosaic Warfare[11]

The DAF Data Strategy names the Data Mesh Environment (DME) as the standard — with Iron Bank as the DAF's central SBOM data lake repository as a mesh node in it.[12]



IRON BANK

What's in progress/on the horizon

- Security Requirement Guide (SRG) for Container STIGs
- Federal Information Processing Standards (FIPs) Labels
- Supply-chain Levels for Software Artifacts (SLSA) Labels

- Verified Publisher
- Event-driven rebuilds
- VEX documentation
- Foreign Ownership, Control, or Influence (FOCI) understanding
- Shift-left for CVE mitigation
- Package management

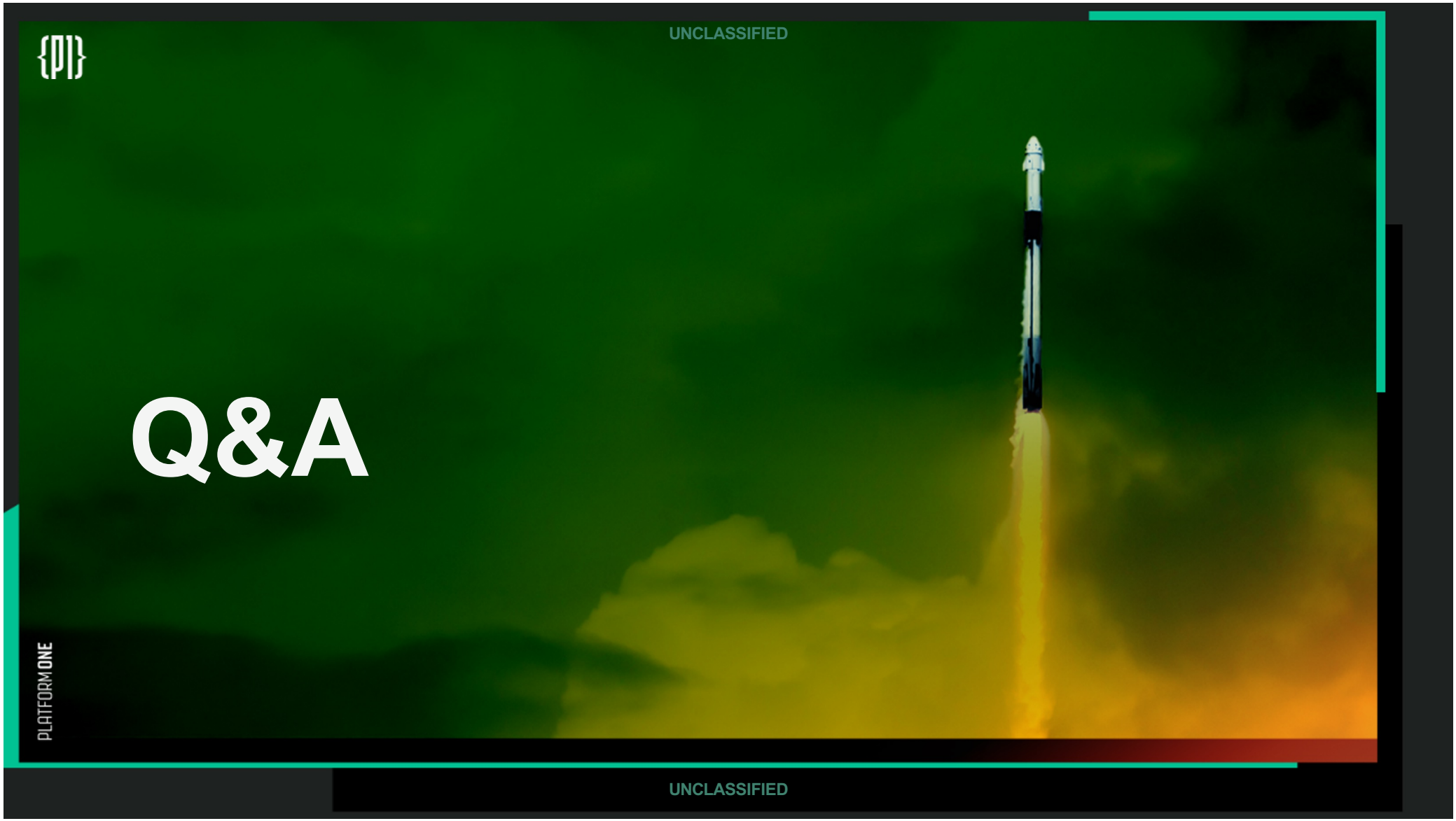
- Integration with DAF DME efforts
- IB Datalake integration with AI Model Context Protocol (MCP) for near real-time supply chain information

- Rebasing images
- Continuous authorization preparation and monitoring for P1 and consumers

PLATFORM ONE

Close

- About Platform One (P1) and software supply chain relevancy
- The Problem: ATO's, CtFs, and the infamous cATO
- The Foundation: open source supply chains are immense
- The Solution: Data Mesh Environments (DME) with domain expertise given to each node, showing a full picture for a cATO


Q&A
PLATFORM ONE

{ THANK YOU }

PLATFORM ONE
DEPARTMENT OF DEFENSE



Sources

1 C. C. Harris, “Unpacking the ATO Process,” NPS (2025); Second Front, “DoD ATO Explained” (2025); GSA, “Intro to ATOs” (2023).

2 OpenControl, “Steps of the ATO Process” (2026); Department of the Air Force, AFI 17-101 RMF (2020).

3 Platform One, “Party Bus: Managed DevSecOps PaaS” (2026).

4 Rise8, “ATO vs Continuous ATO” (2025); DoD CIO, “cATO Evaluation Criteria” (2024).

5 Linux Foundation, “A Summary of Census II” (2022).

6 Black Duck, “2026 Open Source Security & Risk Analysis Report” (2026).

7 Docker, Inc. “What Is a Container?” & “What Is an Image?” (2026).

8 Akamai, “XZ Utils Backdoor” (2024); M. Burgess, WIRED (2025).

9 FIRST, “EPSS Data & Statistics” (2024); Mell & Spring, NIST CSWP 41 (2025); C. Wiens, “Anatomy of a CVE,” Anchore (2017).

10 Anthropic, “Project Glasswing”; OpenAI, “Daybreak” (2026).

11 S. Magnuson, “DARPA Tiles Together a Vision of Mosaic Warfare” (2018).

12 Department of the Air Force, “DAF Data Strategy” (2026).



{ Backup Slides }

PLATFORM ONE
DEPARTMENT OF DEFENSE



The security model has to move with the threat

COMPLIANCE-CENTERED

Where the DoW is today

- Authority granted as a point-in-time document
- Re-authorized every 1–2 years
- Controls checked; risk accepted on paper
- A snapshot — blind to what changed yesterday

DATA-CENTERED

Where the fight is going

- Live data as the source of truth
- Continuous ATO (cATO): monitored, not re-papered
- Patches monitored real-time
- A data mesh feeding real-time risk information

FOUNDATION · SBOM

The SBOM: a nutrition label for software

What an SBOM lists

- Software versions
- Packages detected
- Licensing
- Suppliers & timestamps
- Cryptographic hashes

Formats: CycloneDX · SPDX[13]

Mandated — and scored

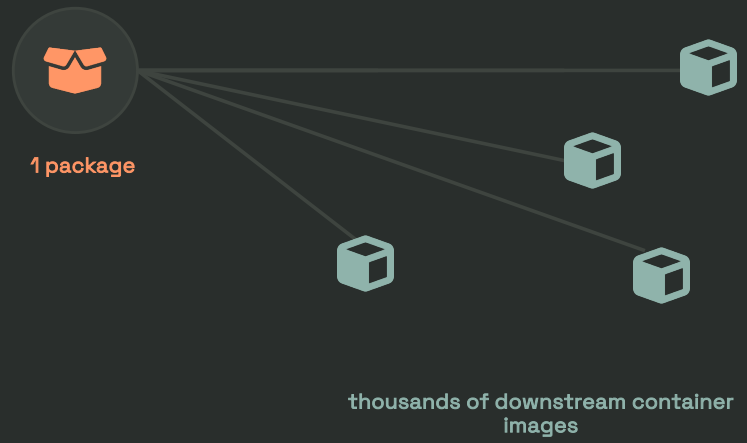
Executive Order 14028[14] mandates SBOM creation, delivery, and tracking for government software. An SBOM lets an operator instantly answer: “am I exposed to the vulnerability announced this morning?”

Iron Bank goes further

ABC (Acceptance Baseline Criteria) + **ORA** (Overall Risk Assessment) turn the SBOM into a consumable health score for downstream users.

Packages: One package can touch thousands of images

Why packages are the high-value target



RECENT SUPPLY-CHAIN EXPLOITS

- **Shai-Hulud**[15] — self-propagating compromise across the npm ecosystem
- **Namastex**[16] — malicious Namastex npm packages worm, 2026
- **axios compromise**[17] — trusted HTTP package poisoned at the source
- **Fast glob**[18] — Russian-maintained code found inside DoD systems

PLATFORM ONE

So How Safe is Open Source?
XZ Utils: a clean codebase, backdoored through a person



The attack was social, not just technical, an ATO wouldn't see that...[8]

1. **A trusted staple** XZ Utils — a compression tool in countless Linux distros — was maintained by one person for years.
2. **Manufactured pressure** Sockpuppet accounts pressured the overworked maintainer to hand off help.
3. **Earned trust** An attacker gained the maintainer's confidence and became a co-maintainer.
4. **The payload** From inside, they planted a backdoor to execute code on other people's systems.

The software supply chain is more than a codebase or boxes to check. It is inherently human — and the human element can matter as much as the code.

XZ ran safely for years. The vulnerability wasn't in the code — it was in the trust model.

PLATFORM ONE

A thicket of container standards — each a point-in-time gate

FIPS 140

Federal Information Processing Standards (FIPS). Federal crypto validation. 140-2 expires Sept 2026 → 140-3[19]

Iron Bank is currently adding "FIPS Ready" tags on images since the config can change post-pull

STIG[20]

Security Technical Implementation Guide (STIG). Hundreds of config controls. No container STIG yet — OS STIGs get force-fit.

Iron Bank + DISA/MITRE working an SRG to cut controls ~9×.

SLSA[21]

Supply-chain Levels for Software Artifacts (SLSA, pronounced "Salsa"). Build provenance & tamper-resistance, Levels 0–3. Most major suppliers ship a SLSA attestation.

Iron Bank is currently implementing SLSA Level 2.

“Minimal” is not the same as “Hardened”

MINIMAL[22]

Contains only what the workload needs

Slimmed-down surface → few or no CVEs remain. Achieved by stripping an image down, or building up from scratch. Some base images such as Alpine are commonly used for their minimal surface.

HARDENED[23]

Removes access, not just packages

Strips root access, permissions and unneeded binaries. Shrinks what an attacker can do once inside. This is the security property that matters.

They’re complementary, not synonyms. There exists a plethora of commercial companies offering a “near-zero CVE” catalog which typically indicates the images are hardened — but pick images on access model, needs, licensing, and SLA fit, not necessarily the CVE count on the marketing page.

Two ways to harden — and a developer's dilemma

Post-removal (slim down)

Take an open-source image, run it through pipelines that strip unneeded packages and permissions until only the essentials remain.

Ground-up (build back)

Start from a fully stripped image and add back only the packages and permissions the workload requires to run.

THE DEVELOPER'S DILEMMA

Hardened images strip permissions — which makes them painful to build on, pushing developers toward bloated, insecure images.

DEV image more permissions, room to build

PROD image locked down — often a one-line swap[24]

PLATFORM ONE

A clean scan is a clean foundation — not a clean system

STATIC (image) scan[25]

Checks what's stored — layers, packages, metadata — before release. This is what Iron Bank gives you: a secure understanding of the foundation.

Sees: image layers · packages · config

RUNTIME (workload) scan

Checks what the workload does after startup — process creation, file access, outbound connections, etc.

Only runtime sees: downloaded payloads · live credential abuse

The point: Iron Bank secures the base layer. Once your code sits on top, scan the runtime — because the exploit that matters most may only appear when the system is live.

REFERENCES · 2 OF 2

Back-up Slides Sources

13 OPSWAT, “SBOM Formats & Standards Explained” (2025); Anchore (syft/grype).

14 Federal Register, “Improving the Nation’s Cybersecurity,” Exec. Order 14028 (2021).

15 Unit 42, “‘Shai-Hulud’ Worm Compromises npm Ecosystem” (2025).

16 L. French, “Namastex npm Packages Compromised in ‘CanisterWorm’ Attack” (2026).

17 Huntress, “Supply Chain Compromise of axios npm Package” (2026).

18 TWiT.tv, “How Did Russian-Maintained Code End Up in Pentagon Software?” (2025).

19 NIST, “FIPS Compliance FAQs” (2018/2024); Chainguard, “FIPS 140-2 vs 140-3” (2025).

20 Puppet, Inc., “DISA STIGs.”

21 SLSA, “Supply-chain Levels for Software Artifacts” & “Build Track.”

22 Minimus, “Minimal Container Images Explained.”

23 Docker Hardened Images (2025); Chainguard; RapidFort; Red Hat; Minimus.

24 Docker, Inc., “Use a Docker Hardened Image” (2026).

25 NHI Management Group, “Static Image vs Runtime Container Security.”

