Continuous Authorization at Machine Speed

Eliminating the 12-Month Compliance Cycle

DAFITC 2026

August 24–26, 2026 · Montgomery, AL

SPEAKERS

Shiv Bhuvanapalli
Chief AI Officer, Inalab Group

Carlton Harris,
ISSO, Inalab Group

INALAB

Authorization doesn't have to be a 12-month nightmare.

The Compliance Tax

The current ATO process imposes a structural cost on every program—measured in time, money, and accuracy. These are not edge cases. They are the norm.

12-18 mo

AVERAGE TIME-TO-ATO

Per system, from initiation to authorization signature

$2M+

AVERAGE COST PER SYSTEM

Direct and indirect costs across the authorization lifecycle

THE COST OF THE STATUS QUO

100s
OF ENGINEERING HOURS
Diverted from mission work to documentation

1,000s
OF DOCUMENTS
Collected, formatted, and maintained

3-4x
REWORK & REVIEWS
Iterative cycles to prove the same facts

$500K+
ONGOING ANNUAL COST
To maintain documentation that is already out of date

The cost of the status quo is measurable — and avoidable.

THE REALITY ON THE GROUND

EVIDENCE DECAYS IMMEDIATELY

Evidence is accurate on signature day. Fiction by month 3. The system keeps changing; the documentation does not.

KNOWLEDGE LOST IN TRANSLATION

The engineers who know the system aren't writing the SSP. Knowledge gets lost in translation to templates, producing documents that describe an idealized system, not the real one.

COMPLIANCE THEATER

When the goal becomes completing paperwork rather than documenting real security posture, compliance theater replaces genuine security documentation.

Sources: GAO-22-104402 | NIST SP 800-37 Rev 2 implementation surveys | DoD CIO FY2023 RMF metrics

Why It Stays Broken

> **Authorization is treated as an EVENT, not a FUNCTION.**

The structural incentives of the current model actively resist change. Understanding why the system persists is the first step toward replacing it.

Collect Once

Evidence is gathered at a single point in time. Write narratives once. Get a signature. Move on. The process is designed for a snapshot, not a continuous state.

Wrong Incentives

The incentive structure rewards **completing paperwork**, not maintaining accuracy. Teams are measured on ATO attainment, not on the ongoing fidelity of their documentation.

Avoidance Loop

Because re-authorization is so painful, organizations actively avoid it. Nobody wants to re-authorize because it's a **12-month nightmare**—so they don't, and drift accumulates.

⚠ The result: systems operate under authorizations that no longer reflect their actual security posture. The ATO becomes a historical artifact, not a living assessment.

Fixing the incentive structure starts with fixing the workflow.

The Real Gap

The policy moved. The workflow didn't.

SWFT, cATO, continuous monitoring—the policy intent is right. The frameworks exist. The guidance is clear.
But intent and execution remain **disconnected for most organizations** today.

WHERE POLICY IS

SWFT
Streamlined authorization pathways

cATO
Continuous authorization to operate

Continuous Monitoring
Ongoing risk visibility and real-time assurance

THE GAP

Policy intent

≠

Execution reality



WHERE MOST ORGANIZATIONS ARE

- Still collect evidence **manually**
- Still write **6ill** narratives **by hand** or copy from templates
- Still wait **months for** assessor review
- Still treat ATO as **a one-time event**
- **Documentation drifts.** Risk increases.

The gap is not a policy gap. **It is an execution gap** — a failure to translate policy intent into operational workflow.

- Policy intent is clear
- Guidance and standards exist
- Execution is the barrier
- Closing the gap drives mission speed

What If You Could Automate What You Already Know

> **The problem isn't knowledge—it's capturing it continuously and automatically.**

Every program team already possesses the information required for a high-fidelity authorization package. The barrier is not awareness—it is the manual, episodic process of translating that knowledge into compliance artifacts.

System Architecture

You know your system architecture—how components connect, where boundaries exist, and what the data flows look like.

Data Location

You know where your data lives—classification levels, storage locations, and access patterns across the environment.

Deployment State

You know what's deployed and what's drifted—the delta between your authorized baseline and your current running configuration.

Threat Environment

You know your threat environment—the adversary TTPs relevant to your mission, your data, and your exposure surface.

Today: three processes you can build to make that happen. The following section walks through each automation pathway—turning institutional knowledge into continuously maintained, machine-speed authorization evidence.

You already have the knowledge. Now capture it automatically.

Authorization as a Continuous Function

Reframe: not 'get an ATO' — maintain authorization continuously

1. Continuous evidence collection
2. AI-generated narratives from real data
3. Threat-informed control prioritization

You can build each of these against your own infrastructure

Authorization doesn't have to be a 12-month nightmare.

Process 1: Automate Evidence Collection

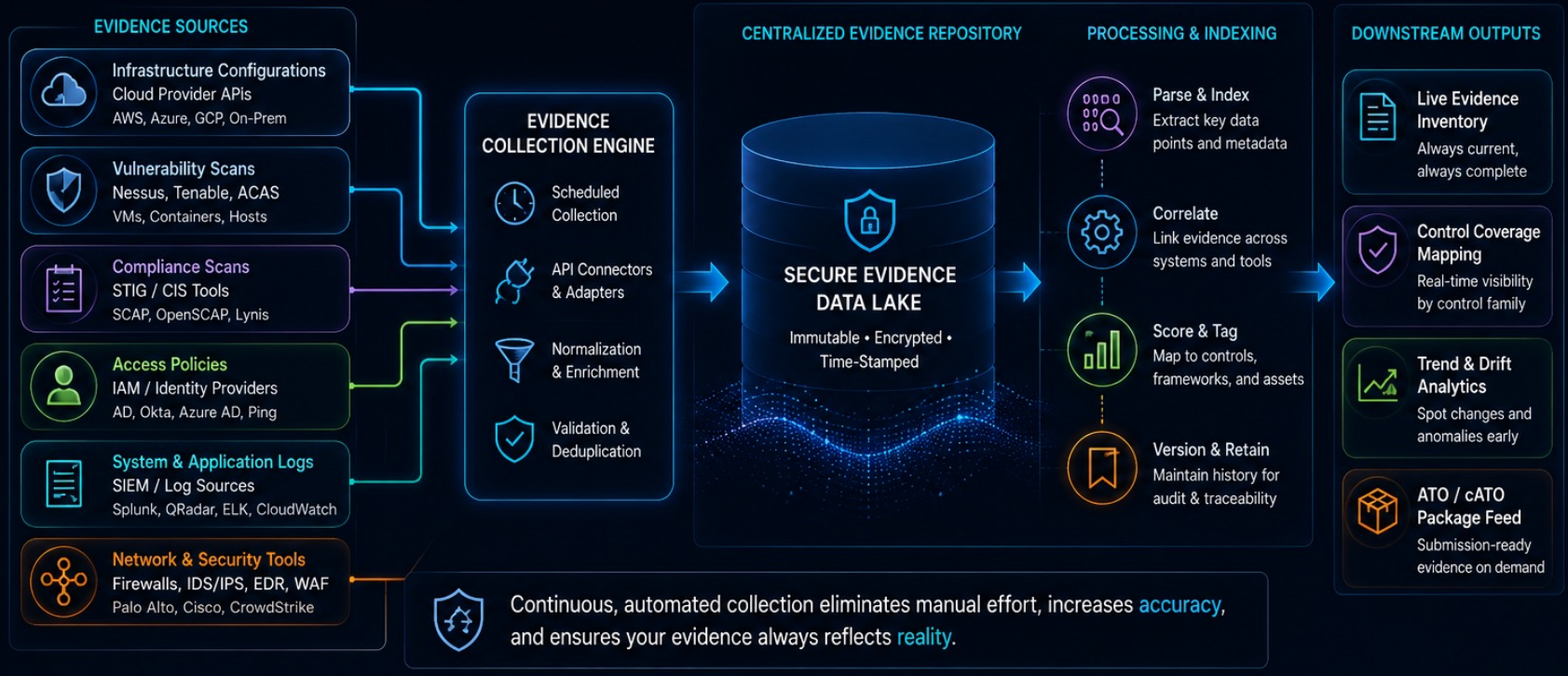
- → Infrastructure Configs → Cloud Provider APIs
- → Vulnerability Scans → Nessus / Tenable / ACAS
- → Compliance Scans → STIG / CIS Tools
- → Access Policies → IAM / Identity Systems
- → Deployment State → CI/CD Pipeline
- → Patch Levels → Config Management Tools

⚠ **The pattern: adapters that pull from existing tools on a schedule**

You can build each of these against your own infrastructure

Process 1: Automate Evidence Collection

Continuously collect, normalize, and store authorization evidence from across your infrastructure and tools — at machine speed.



Continuous, automated collection eliminates manual effort, increases accuracy, and ensures your evidence always reflects reality.

KEY BENEFITS

- **Faster** — Collect in minutes, not months
- **More Accurate** — Eliminate manual errors and gaps
- **Always Current** — Real-time evidence for real-world systems
- **Lower Cost** — Reduce labor and rework
- **Audit Ready** — Immutable, traceable, and defensible

What Continuous Evidence Gets You

- Authorization package reflects your system AS IT EXISTS RIGHT NOW
- Drift detection becomes trivial: diff last collection vs. current
- "ATO expiration" loses meaning when evidence is always fresh
- Assessors verify against real data, not 6-month-old snapshots
- Re-authorization becomes an update, not a restart

Your package stays alive. Re-auth becomes an update, not a restart.

Drift Detection in Action

Tuesday (Baseline State)

- 3 open ports (443, 8080, 22)
- 4 IAM roles
- MFA enforced on all accounts
- Baseline matches authorization

Wednesday (With Drift)

- 4 open ports (added 3389 - RDP)
- 5 IAM roles (new admin role)
- MFA disabled on 1 service account
- DRIFT DETECTED - 3 controls affected

⚠ System automatically categorizes the drift, maps affected controls, and drafts a POA&M entry. Your package never goes stale.

Drift caught in hours, not discovered at re-authorization.

The Template Problem

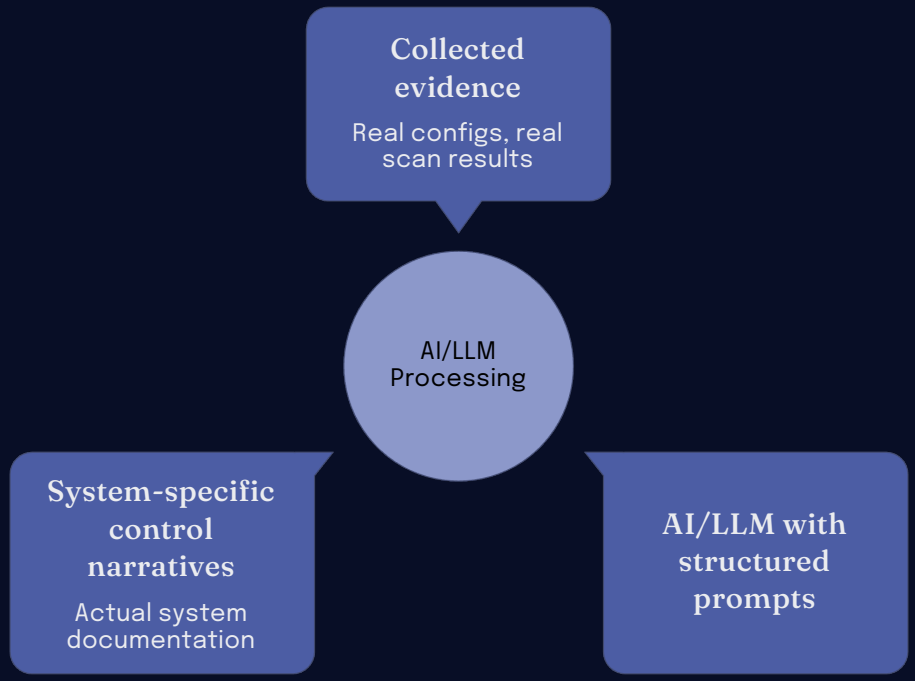
> "The organization manages information system accounts by identifying account types, establishing conditions for group membership, identifying authorized users, specifying access privileges, requiring appropriate approvals for account creation, and reviewing accounts on a periodic basis in accordance with AC-2."

- Copied from a prior authorization
- Based on NIST boilerplate language
- Could describe literally any system
- Assessors accept them because the alternative is worse

⚠ **The result: compliance theater – documents say what they're supposed to say, but don't describe what the system actually does.**

The gap isn't policy – it's execution.

Process 2: Generate Narratives From Your Data



- **Every claim maps to collected evidence**
- **Narrative describes YOUR system, not a generic template**
- **Assessors can follow the chain: narrative → evidence → actual system state**

You already have the knowledge. Now capture it automatically.

Before and After: Same Control, Different World

BEFORE (Template)

> "The organization manages information system accounts by identifying account types, establishing conditions for group membership, identifying authorized users, specifying access privileges, requiring appropriate approvals for account creation, and reviewing accounts on a periodic basis in accordance with AC-2."

AFTER (AI-Generated from Evidence)

> "System access managed through AWS IAM with role-based access control via 7 IAM roles mapped to 4 functional groups. Provisioning requires ServiceNow approval (evidence: CloudTrail logs, 23 events in 90 days). Quarterly review via IAM Access Analyzer (last scan 2026-07-15, 3 findings remediated in 48hrs). MFA required for privileged access (evidence: 100% compliance)."

The gap isn't policy – it's execution.

How to Build This

Components

LLM
Cloud-hosted or on-prem depending on classification

Structured Prompts
One per control family, referencing relevant evidence types

Evidence Context Window
Feed relevant collected data per control

Output Formatting
OSCAL for machines, Word/PDF for humans, eMASS for submission

The Workflow



Human reviews AI output, not the other way around

The Equal Treatment Problem

300+ NIST 800-53 Rev 5 controls

Current approach: treat them all with equal weight – spread resources equally thin across everything.

Web App vs. Embedded System

A web application has different risks than an air-gapped embedded system.

PII vs. Logistics

A system handling PII has different threats than a logistics tracker.

Targeted Threats

Your adversary uses specific TTPs – not all 200+ in ATT&CK.

> **When you treat everything equally, you get mediocre coverage everywhere instead of strong coverage where it matters.**

Process 3: Map Your System to the Threat



1. Characterize Your System

Architecture, exposure, data sensitivity, connectivity

2. Identify Relevant Adversary TTPs

MITRE ATT&CK filtered by sector, architecture, exposure + threat intel feeds

3. Map TTPs to NIST Controls

Which controls mitigate techniques relevant to you?

4. Prioritize

High-relevance controls get more rigor; low-relevance get baseline coverage

This funnel illustrates a targeted approach to security, moving from broad threat landscapes to a prioritized set of controls specific to your system's unique characteristics and relevant adversary tactics.

Fixing the incentive structure starts with fixing the workflow.

Automating Threat-Informed Prioritization

AI Agents That:

- Consume threat intelligence feeds (CISA, sector-specific, adversary tracking)
- Map current adversary TTPs to MITRE ATT&CK techniques
- Score technique relevance against your system architecture
- Produce a prioritized control list specific to your environment

Build This With:



MITRE ATT&CK framework

(free, open)



CTI feeds

(CISA, open source, or commercial)



Your system boundary definition



Scoring logic

Weighs your specific architecture against active threat landscape

You already have the knowledge. Now capture it automatically.

What Threat-Informed Prioritization Changes

AO makes risk decisions based on threat relevance, not equal weighting

Remediation effort targets controls that reduce real risk

POA&Ms become risk-ranked, not just a backlog

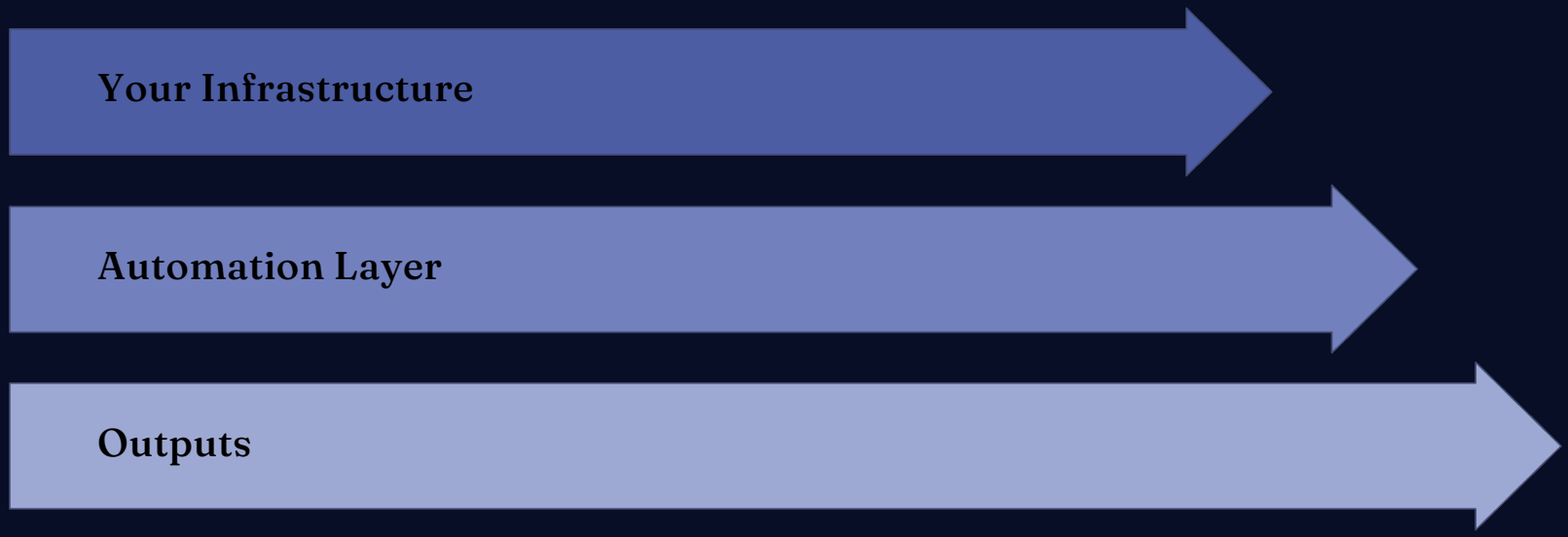
You can explain WHY a control is prioritized

"This mitigates T1190 (Exploit Public-Facing Application) actively used by [adversary group] against systems with our architecture"

Assessment becomes a conversation about risk, not a checklist exercise

Risk-informed conversations replace compliance checklists.

The Full Continuous Authorization Flow



Authorization doesn't have to be a 12-month nightmare.

What You Need to Build This

Keep it grounded and practical – this is the "you can actually do this" slide. Make it clear these are all achievable with existing tools and resources.

Access to your infrastructure APIs

(You already have this)

Evidence adapters for your specific tools

(The integration work)

An LLM layer for narrative generation

- Cloud: AWS Bedrock, Azure OpenAI (for unclassified/CUI)
- On-prem: self-hosted models for classified

MITRE ATT&CK mapping for your architecture

(One-time + periodic refresh)

Human review workflow

(Automation proposes, humans approve)

Export formatting

(OSCAL libraries exist; NIST provides reference implementations)

Three buildable processes. No vendor required.

Realistic Expectations

This is NOT 'push a button, get an ATO'

The Reality

- Initial setup: 2-4 weeks to build adapters, map environment, validate outputs
- First-pass narratives need human refinement – quality improves over time
- Runs automatically, but humans review and approve

What You Gain

Update cycle drops from months to days

Evidence is always current

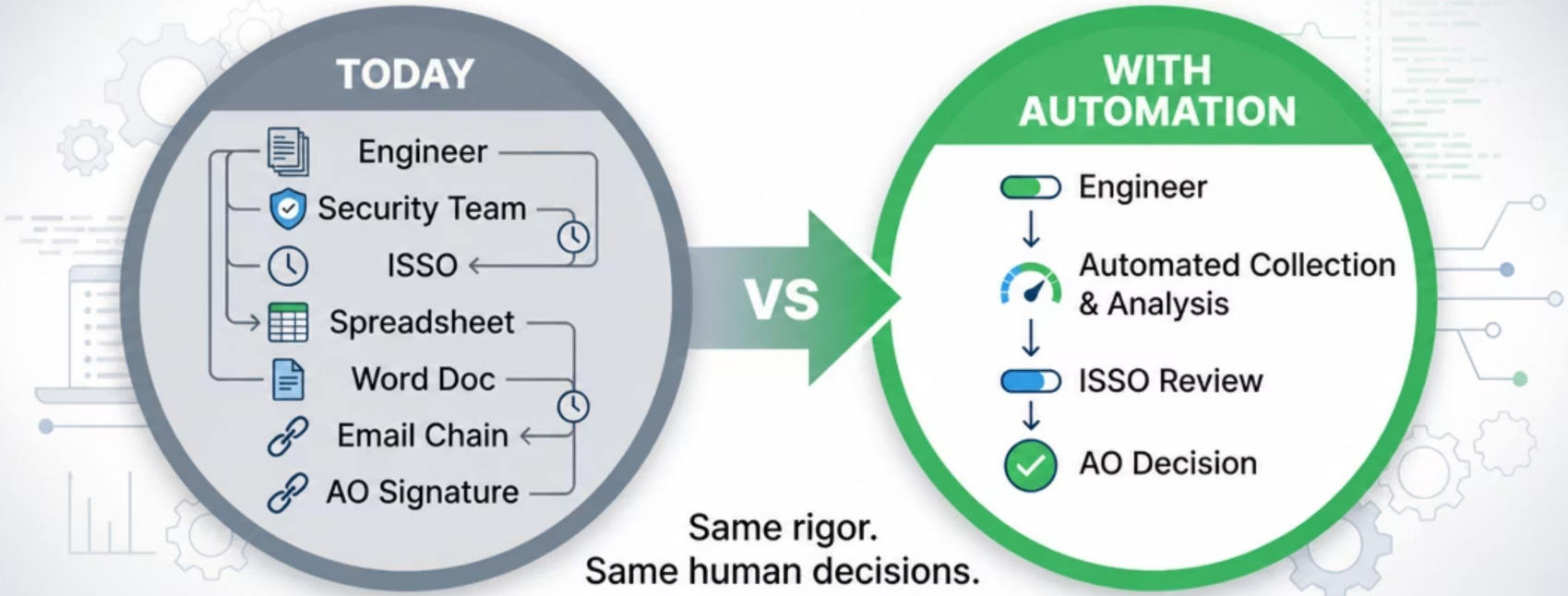
Narratives describe your actual system

Assessment becomes verification, not discovery

Start small. One system. One control family. Prove it works. Expand.

Authorization doesn't have to be a 12-month nightmare.

What Actually Changes For You



Same rigor.
Same human decisions.
Fewer manual handoffs.

The workflow shrinks. The quality improves.

What Automation Handles

- Evidence collection and scheduled refresh
- Narrative drafting from collected data
- Drift detection and alerting (something changed – here's what)
- Format conversion and package assembly (OSCAL, Word, eMASS)
- Threat mapping and control prioritization scoring
- Repetitive comparison and diff analysis

Everything repetitive goes to machines.

What Humans Still Own

These are the high-stakes, judgment-call decisions that require human expertise and authority.

- Risk acceptance decisions – the AO's job doesn't get automated
- Edge cases and context that data alone can't capture (examples: "We accepted this risk because of mission timeline", "This control is implemented differently due to operational constraints")
- Organizational risk tolerance – different missions, different thresholds
- Validating AI outputs – trust but verify
- Deciding what's "good enough" for your mission context

Judgment, context, and risk decisions stay human.

The Shift in the ISSO's Day

FROM

- Document assembler
- Chasing evidence
- Formatting packages
- Reactive compliance

TO

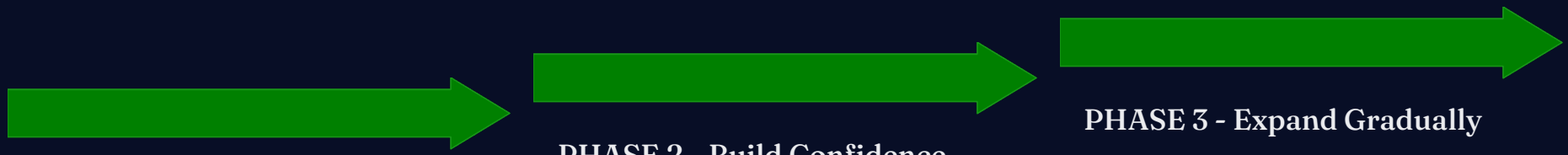
- Risk advisor
- Analyzing threats
- Advising on risk decisions
- Proactive security posture

Same people. Higher-value work. The expertise ISSOs already have finally gets used where it matters most – making security decisions, not formatting Word documents.

Same people. Higher-value work.

Building Trust: Where to Draw the Line

This outlines a measured, evidence-based approach to expanding automation, progressing from tight human control to calibrated trust.



PHASE 1 - Start Conservative

- **AI collects and drafts, human reviews everything before submission**
- **Human approves every narrative before it enters the package**

PHASE 2 - Build Confidence

- **Track accuracy over time**
- **How often does review change AI output?**
- **As confidence grows, expand automation scope**

PHASE 3 - Expand Gradually

- **Low-risk controls: AI-generated, human spot-checked**
- **High-risk controls: AI-drafted, human-reviewed in detail**
- **Risk decisions: always human**

Trust builds on data, not faith.

Three Processes, One Outcome

1. **Automate evidence collection — your package is always current**
2. **Generate narratives from real data — your documentation is always accurate**
3. **Prioritize by threat — your effort goes where risk actually lives**

You know your system better than anyone

These processes capture that knowledge continuously

Authorization becomes a living function, not a periodic event

Start with one system. One control family. Prove it. Expand.

You already have the knowledge. Now capture it automatically.

Questions?

Shiv Bhuvanapalli, Chief AI Officer, Inalab Group
sbhuvanapalli@inalabgroup.com

DAFITC 2026 · August 24–26, Montgomery, AL







