BlueFlag Security

SOLUTION BRIEF

Beyond IAM. Into the SDLC.

Why traditional Identity & Access Management stops short — and how BlueFlag Security extends governance to the people, machines, and AI agents that actually change your software.

> ***Traditional IAM controls access.*** **BlueFlag controls who can change software — and how.**
>
> EXECUTIVE SUMMARY

THE IDENTITY GAP IN MODERN SOFTWARE DELIVERY

IAM was built for access. Software is built by change.

Identity & Access Management has matured into a critical enterprise control plane — granting and revoking access to applications, infrastructure, and data. But software delivery has fundamentally changed: humans, CI/CD pipelines, bots, and AI coding agents now collectively decide what reaches production. Traditional IAM was never designed to govern that.

Where traditional IAM stops short

Identities live in silos

Entra/Okta knows the corporate identity. GitHub knows the developer. JFrog knows the publisher. No system sees the same person across all three.

Machines aren't first-class

CI/CD accounts, service accounts, and AI agents are treated as technical accounts — not as autonomous actors with risk profiles and behavior.

No behavioral context

IAM enforces who can do what — but not what they actually do. Anomalous commits, unusual deploys, and pipeline misuse fly under the radar.

Privilege paths invisible

Roles look safe in isolation. Combined across SDLC tools, they create toxic privilege paths from developer keyboard to production.

THE REALITY

IAM tells you who can log in.
BlueFlag tells you who can change software — and how.

TOP 5 DIFFERENTIATED USE CASES

Where BlueFlag goes where IAM can't.

Five outcome-driven capabilities that extend identity governance from authentication into software change.

1 Cross-Platform Identity Correlation — The Developer Identity Graph

WHAT BLUEFLAG DOES

- → Correlates identities across IAM (Entra, Okta), GitHub/GitLab, JFrog, and CI/CD
- → Resolves multiple tool-level accounts to one unified developer identity
- → Builds a behavioral graph spanning every SDLC system

WHY TRADITIONAL IAM CAN'T

- ✕ *Manages identities per system, not across SDLC tooling*
- ✕ *No behavioral context across tools*
- ✕ *Cannot answer "who, across our systems, can change this code?"*

OUTCOME **Single source of truth for "who can change software."**

2 Non-Human Identity Governance — AI & Machine Identities

WHAT BLUEFLAG DOES

- → Models CI/CD accounts, service accounts, and AI agents as first-class identities
- → Governs permissions, usage, and full lifecycle
- → Tracks behavior of pipelines, bots, and AI coding/testing agents

WHY TRADITIONAL IAM CAN'T

- ✕ *Treats these as technical accounts — not autonomous actors with risk*
- ✕ *No lifecycle, no behavior tracking, no AI-aware policy*
- ✕ *Machine identities outnumber humans 45:1 — and IAM doesn't see them*

OUTCOME **Full control of machine and AI-driven software changes.**

3 Behavioral Risk Analysis in the SDLC

WHAT BLUEFLAG DOES

- → Analyzes behavior across commits, PRs, and pipelines
- → Detects unusual deploys and pipeline privilege escalation
- → Flags suspicious repository access in real time

WHY TRADITIONAL IAM CAN'T

- ✕ *Focuses on access control, not developer behavior*
- ✕ *No awareness of "normal" SDLC patterns*
- ✕ *Cannot distinguish a routine push from a risky one*

OUTCOME **Detect risk before code reaches production.**

TOP 5 DIFFERENTIATED USE CASES (CONTINUED)

4 Toxic Combination & Privilege Path Analysis

WHAT BLUEFLAG DOES

- → Identifies dangerous combinations: developer + direct production deploy rights
- → Surfaces CI/CD account + artifact publishing + release approval overlaps
- → Flags AI agent + unrestricted repo + pipeline execution risks
- → Maps end-to-end privilege escalation paths across tools

WHY TRADITIONAL IAM CAN'T

- ✕ *Sees roles in isolation — not combined risk across SDLC systems*
- ✕ *No cross-tool privilege graph*
- ✕ *Toxic combinations only become visible after a breach*

OUTCOME **Eliminate hidden privilege escalation paths.**

5 End-to-End SDLC Traceability — Identity → Code → Deployment

WHAT BLUEFLAG DOES

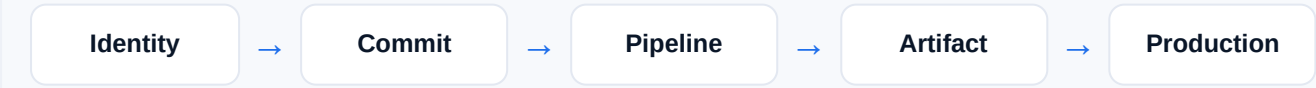
- → Links identity → commit → pipeline → artifact → production
- → Provides a full audit trail with continuous compliance evidence
- → Establishes accountability across the entire SDLC

WHY TRADITIONAL IAM CAN'T

- ✕ *Stops at authentication and authorization*
- ✕ *Has no visibility into what actually happened in software delivery*
- ✕ *Cannot produce end-to-end change attribution for auditors*

OUTCOME **Board-level visibility into software change risk.**

The full traceability chain



STRATEGIC SUMMARY

Capability-by-capability comparison.

BlueFlag isn't a replacement for IAM — it's the layer above it. Where IAM stops at "who can log in," BlueFlag begins at "who, what, and how can change software."

Capability	Traditional IAM	BlueFlag Security
Identity management	✓ Yes	Yes — Extended
Cross-SDLC identity correlation	✕ No	Yes
AI & machine identity governance	~ Limited	Yes
Behavioral risk in development	✕ No	Yes
SDLC privilege path analysis	✕ No	Yes
End-to-end SDLC traceability	✕ No	Yes

From → To

WITH IAM ALONE

Authenticated access — but blind to what humans, machines, and AI agents actually do inside software delivery.

WITH BLUEFLAG

Identity-first governance across the full SDLC — every change attributable, every privilege bounded, every workflow auditable.

THE EXECUTIVE BOTTOM LINE

One line your board will remember.

EXECUTIVE ONE-LINER

Traditional IAM controls access.

BlueFlag controls **who can change software** — and **how.**

Why this matters now

AI is changing software

AI coding agents now write, review, and deploy code. They need identity governance — not just API keys.

Machines outnumber humans

Non-human identities (CI/CD, services, agents) are 45× more numerous than humans, and grow faster.

Regulation is catching up

DORA, NIST SSDF, and ISO 27001 increasingly demand identity attribution across the full software supply chain.

Engage with BlueFlag Security

1. Identity Discovery

30-minute scoping call to map your SDLC tools, identities, and current IAM coverage gaps.

2. Risk Assessment

Free baseline assessment — surface toxic privilege combinations and unattributed machine identities.

3. 30-Day Pilot

Deploy BlueFlag governance on a single Jira/Git project. Demonstrate audit and compliance value fast.

Ready to govern who can change your software?

Start with a discovery conversation. We'll come prepared.

www.blueflagsecurity.com