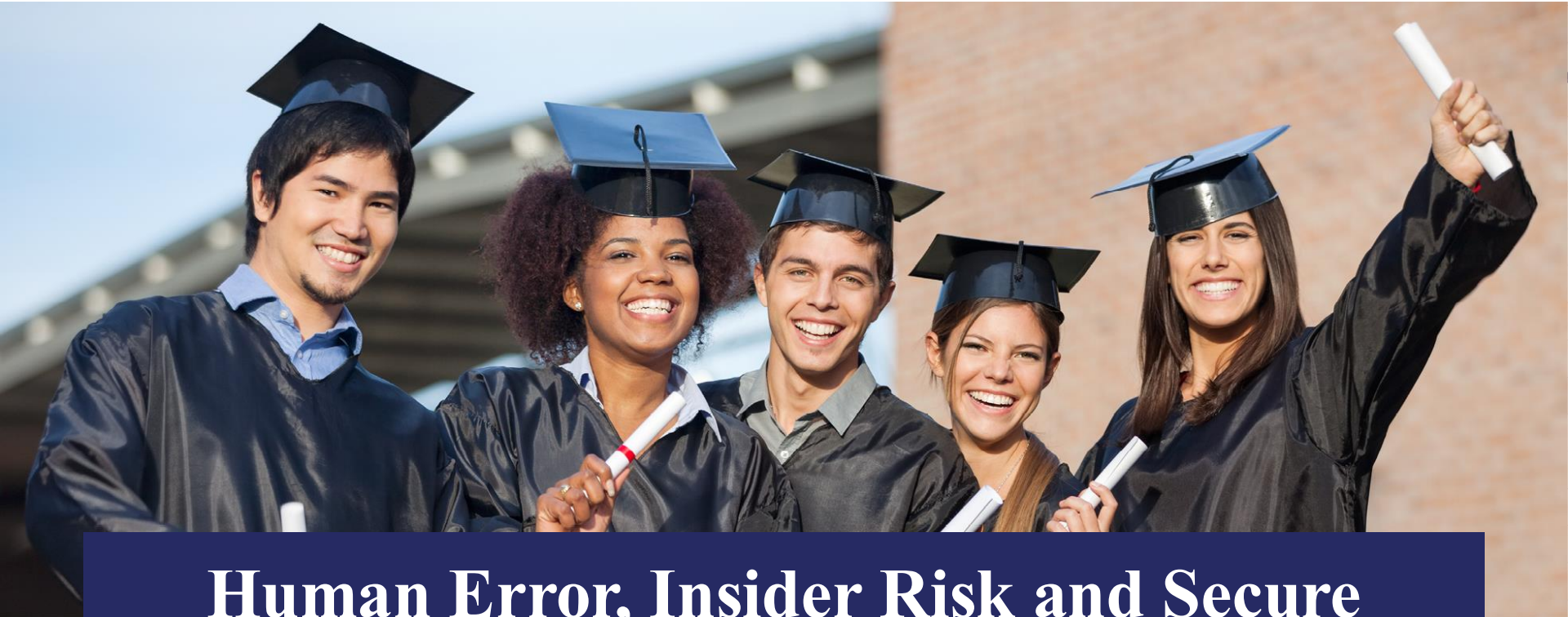




Human Error, Insider Risk and Secure Behavior Design

**Florida Association of Management Information Systems
(FAMIS)**

June 23-25, 2026



FLORIDA DEPARTMENT OF
EDUCATION
fldoe.org

www.FLDOE.org

Session Objectives



Understand

Human error through
systems thinking lenses.



Define

Insider risk in practical,
actionable operational terms.



Apply

Secure behavior design
principles to environments.



Connect

Controls to NIST best
practices and frameworks.



Strengthen

Endpoint, process and
governance safeguards.



Collaborate

Scale Controls across
diverse environments.

Blame Vs. Systems Thinking

The Blame Model

- Asks: "Who made the mistake?"
- Focuses on individual punishment and retraining, often suppressing reporting of near-misses.

The Systems Model

- Asks: "What conditions made the mistake likely?"
- Redesigns the context (tools, workload, process) to make the correct action the easiest one.

What Is Human Error?

Human error is not random or a character flaw. It is an **outcome** shaped by:

- Complexity & Ambiguity
- Fatigue & Friction
- Slips, Omissions and Mistakes
- Unsafe workarounds for slow processes

Real-world Phishing Examples Tied To Human Error Conditions

- **Fatigue:** A team member working late clicks a fake “password reset” email because they are too mentally drained to notice the suspicious sender address.
- **Complexity and ambiguity:** A vendor payment phishing email looks credible because it references a real project and uses familiar terminology, making it hard to distinguish from legitimate communication.
- **Slips and omissions:** An employee intends to verify an urgent invoice request but gets interrupted and forgets the step before approving payment.
- **Mistakes:** A new employee believes a fake Information Technology email is genuine because they do not yet know what official account notices normally look like.
- **Unsafe workarounds:** Staff become used to bypassing slow official tools, so they are more likely to trust phishing emails offering “faster” file access or approvals.

Operational Human Factors

- **Alert Fatigue:** Drowning in false positives.
- **Context Switching:** Loss of focus on security details.
- **Privilege Sprawl:** Accumulated access over time.
- **Ambiguous Ownership:** Unclear responsibility.
- **Shadow IT:** Using tools outside governance.
- **Training Overload:** Diminishing returns on annual "death by PowerPoint."

Common Security-Relevant Errors

- 🔒 Misaddressed sensitive emails
- 🔒 Weak or reused passwords
- 🔒 Approval without verification
- 🔒 Improper data handling
- 🔒 Privilege misuse through convenience
- 🔒 Delayed patching or missed alerts
- 🔒 Bypassing Virtual Private Network for speed
- 🔒 Insecure document sharing



What Is Insider Risk?

Any internal actor—employees, contractors, vendors or partners—can create risk.

Managing insider risk means managing **trust, access, visibility and response** across all internal actors.

Categories Of Insider Risk

Malicious

Intentional theft or damage.

Requires strong monitoring and segregation.

Negligent

Careless or shortcut-taking behavior. Needs reduced complexity.

Compromised

External actors using internal credentials. Highlight's identity protection.

Well-Intentioned

People trying to do their jobs under poor system design conditions.

Insider Risk Signals



Unusual Access



Large Transfers



Policy Overrides



Off-Hours Admin



Privilege Escalation



Disabled Controls

Human-Centered Response

Contain

Stop the bleeding
immediately.

Preserve

Evidence gathering for
fact-finding.

Determine

Error, compromise or
misuse?

Learn

Capture lessons for
system updates.

Secure Behavior Design

Design The Environment

Treat security as a design challenge. If the approved path is hard, users will shortcut it.

- Reduce friction for approved actions
- Increase friction for risky actions
- Use smart defaults and guardrails

Principles Of Design



Right Path = Easy

Make the most secure action
the path of least resistance.



Risk = Visible

Ensure risky choices trigger
clear, contextual prompts.



Least Privilege

Default to the minimum
access required for the task.



Recovery

Design for mistakes. Ensure
recovery is easy and safe.

Separation Of Duties

4 Core Pillars:

Authorization, Custody, Recordkeeping and Reconciliation

Why It Matters

Foundational control to prevent concentration of power. No single person should be able to **Initiate, Approve, Execute and Conceal** a high-risk action.

- Reduce fraud and error risk
- Improves oversight and trust
- Supports compliance audits

Separation Of Duties In Practice

Role	Primary Responsibilities	What This Role Cannot Do
Developer	Designs, codes, unit tests and submits application changes through the approved Software Development Life Cycle (SDLC) process.	Cannot deploy code to production, cannot assign themselves elevated application access and cannot make direct production database changes.
Deployment Administrator	Promotes approved releases to test and production environments, validates deployment completion and maintains deployment logs.	Cannot alter source code, cannot approve their own release content and cannot grant themselves business-admin privileges inside the application.
Business Owner / In-App Admin	Approves business roles, manages user entitlements within the application and authorizes access based on business need.	Cannot change application code, cannot deploy releases and cannot administer the underlying infrastructure or database.
Database Administrator (DBA)	Manages database performance, backups, recovery, security configuration and controlled schema changes.	Cannot approve business access, cannot deploy application releases independently and cannot modify business logic outside approved change procedures.

Least Privilege & Just-In-Time Access

- Grant only the access needed for the role.
- Just-in-Time (JIT): limited duration of elevated access.
- Review entitlements regularly.
- Remove dormant and excessive permissions.

Reduces blast radius of human error and insider misuse simultaneously.

Aligning With NIST & 60GG-2 Best Practices



Identify

Assets, roles, risks and dependencies.



Protect

Access control and behavior design.



Detect

Monitoring and endpoint visibility.



Respond

Containment and containment plans.



Recover

Resilience and post-incident learning.

NIST 800-53 Control Themes

- ❖ **Access Control (AC):** Identity and privilege.
- ❖ **Awareness (AT):** Training and procedures.
- ❖ **Audit (A):** Logging and accountability.
- ❖ **Configuration (CM):** Consistent baselines.
- ❖ **Incident Response (IR):** Response testing.
- ❖ **Integrity (SI):** System and information safety.

Practical Risk Controls



**Phishing-Resistant
Multi-factor
Authentication (MFA)**



**Data Loss Prevention
(DLP)**



**Privileged Access
Management (PAM)**



Approval Workflows



Session Logging



Immutable Backups

Endpoint Security

Where Behavior Meets Risk

Visibility and control at the device level are non-negotiable. Endpoints (EDR/XDR) are the primary telemetry sources for both error and abuse.

- EDR/XDR for behavioral alerts
- Device Enroll for configuration governance
- Strong endpoint ops reduce risk

Endpoint Tool Categories

EDR/XDR

Endpoint Detection &
Response

MDM/UEM

Configuration Control

Vulnerability Mgmt

Exposure Reduction

App Control

Approved Execution

Encryption

Physical Data Safety

Remote Support

Auditable Assistance

Operationalizing Your Tools

- **Standard Baselines:** Consistency across environments.
- **Asset Inventory:** You can't protect what you don't see.
- **Telemetry Tuning:** Reducing alert fatigue for IT teams.
- **Documented Playbooks:** Clear steps when a tool triggers an alert.

Playbook Example

1. **Trigger:** High-severity EDR alert for suspicious script execution
2. **Severity:** High
3. **Primary owner:** Security Operations Center Analyst
4. **Escalation owner:** Incident response manager
5. **Containment action:** Isolate device and disable risky sessions
6. **Investigation sources:** EDR, Security Information and Event Management (SIEM), identity logs, firewall, Domain Name System (DNS)
7. **Recovery action:** Reimage device and restore user access
8. **Success criteria:** Threat removed, no lateral spread, user restored safely, lessons documented

System Security Plans (SSP)

Element	Purpose
System Description	Defines boundaries and data classification.
Control Implementation	Documents HOW a system is protected.
Roles & Responsibilities	Identified WHO is accountable for each control.
Interconnections	Maps dependencies to other district systems.

SSP & Error Reduction

Clarity

Reduces ambiguity in ownership and procedures. People know their specific safeguarding responsibilities.

Consistency

Ensures safeguards are preserved during staffing changes or technology updates.

Escalation

Defines clear paths for reporting and resolving system anomalies.

Training



Role-Based

Specific training for
admins vs users.



Just-In-Time

Guidance at the point of
action.



Scenario-Based

Practical learning over
abstract theory.

Cross-Functional Operating Model

Security is not an IT silo. It requires alignment between:

- Security & IT Operations
- HR & Legal Counsel
- Audit & Compliance
- District Leadership & Stakeholders

Implementation Roadmap

Assess

Current state risks &
controls.

Prioritize

Focus on high-impact
gaps.

Standardize

Core controls across
environments.

Scale

Measure, refine and
deploy.

Key Takeaways

- Human error is a **System Issue**, not just a people issue.
- Insider risk includes both **Intentional and Unintentional** acts.
- **Secure Behavior** can be intentionally designed into the environment.
- **NIST & 60GG Alignment** ensures consistency and compliance.
- **Technical OPS** (ERD, SoD, SSPs) make these concepts operations.
- **Collaboration and Training** turns isolated controls into a functional program.



www.FLDOE.org

Thank you!

Build systems where secure behavior is the easiest behavior.