The Pre-Internet Computer

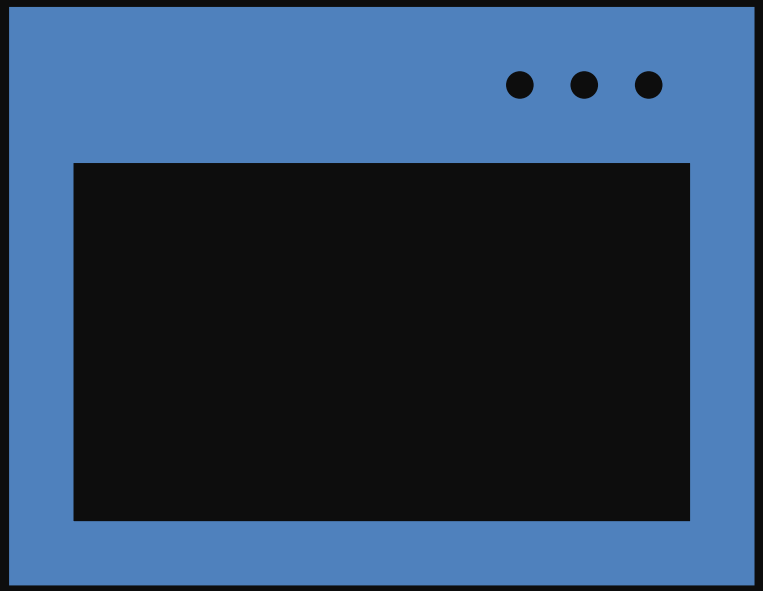
Offensive Security on Mainframe Control Planes

Adam Toscher

The Problem

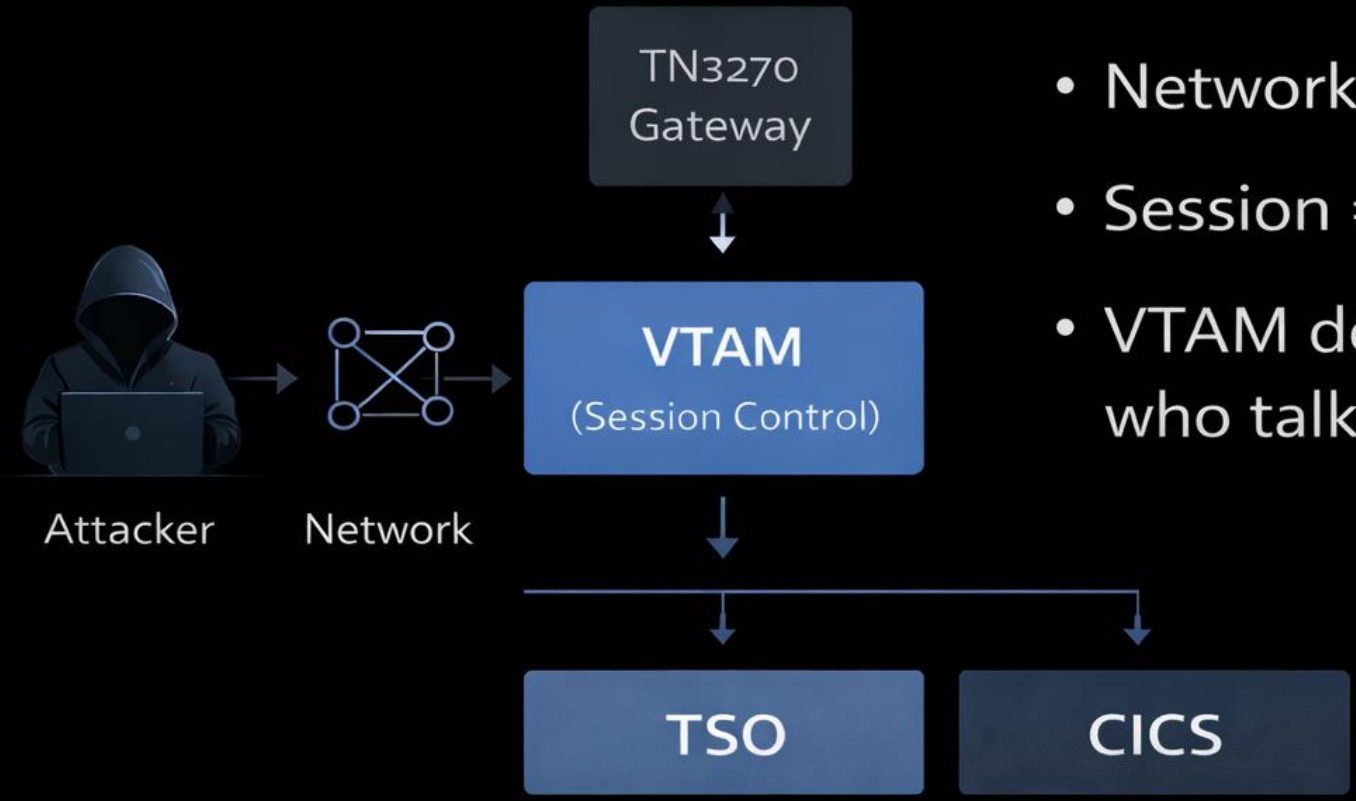
- Modern offensive assumptions:
- • Processes
- • Ports
- • Root
- • Filesystems

- These fail on mainframes.



- Process the majority of high-value financial transactions
- Handle airline booking, inventory, and scheduling logic
- Execute payment clearing and settlement at scale
- Store authoritative government and enterprise records
- Operate high-volume, high-trust workflows built over decades

Exposure Is Session-Based (VTAM)



- Network ≠ Access
- Session = Exposure
- VTAM decides who talks to what

Security decisions are distributed across Subsystems, not centralized in the kernel.

▾ CONTROL PLANES

VTAM	TSO	RACF	JES	CICS/KICKS
Session Fabric	Interactive	Authorization	Batch Execution	Transactions

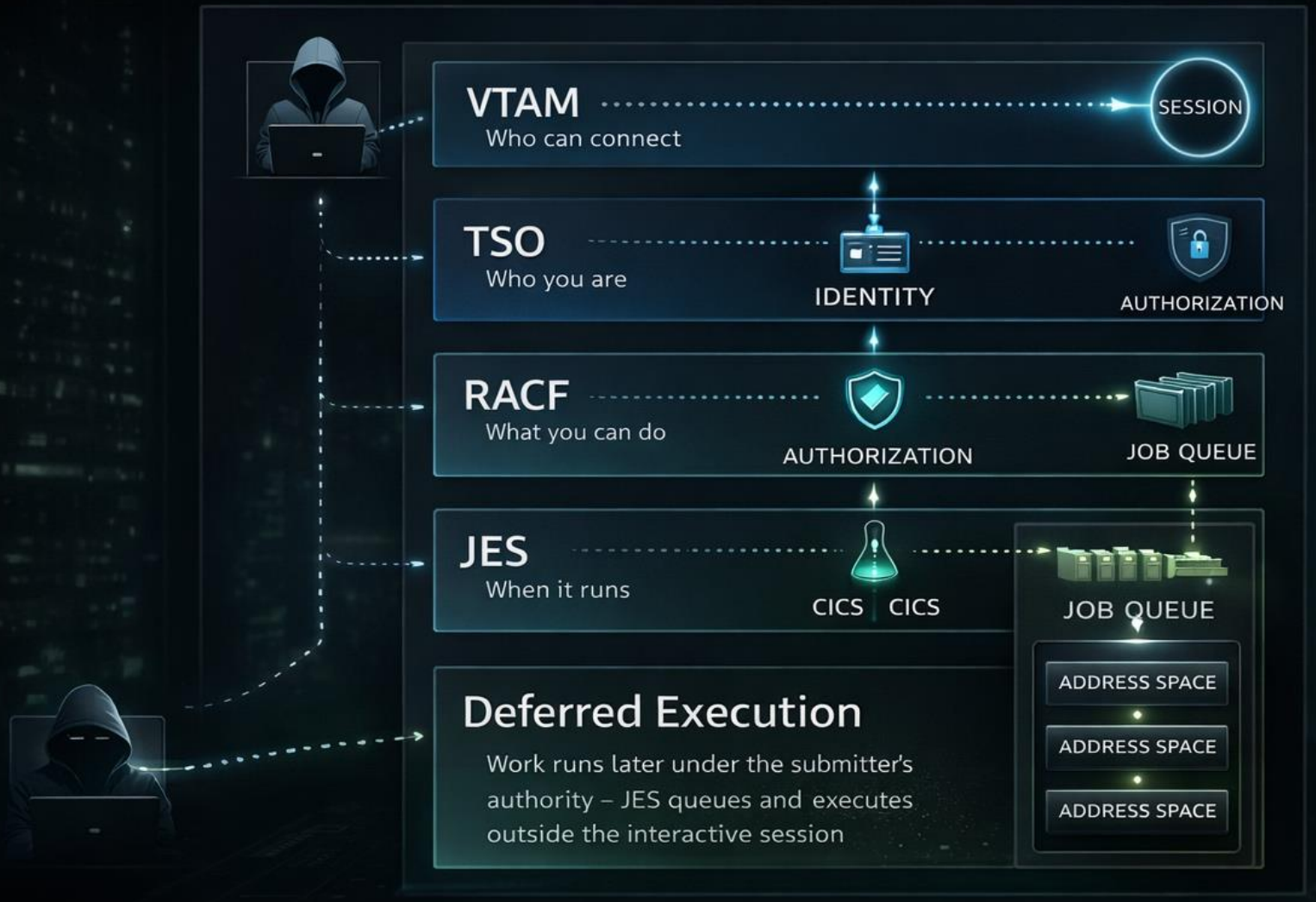
- **VTAM — Who can connect**
  Session establishment and terminal binding. Exposure is session-based, not port-based.
- **TSO / ISPF — Who you are**
  Interactive identity binding. All commands, dataset access, and job submissions inherit this RACF context. *Not a shell. Not a process environment.*
- **RACF — What you can do**
  Continuous authorization engine. Access decisions are made per resource, before execution.

JES — When it runs
Deferred execution broker. Work is queued, scheduled, and executed later under the submitter's identity.

- **CICS — What business actions are allowed**
  Transaction control plane. Users invoke business logic, not programs, with authorization enforced at the transaction level.

Attackers don't move between hosts. Main Frame Hackers move between control planes.

Mainframe Control Planes & Deferred Execution



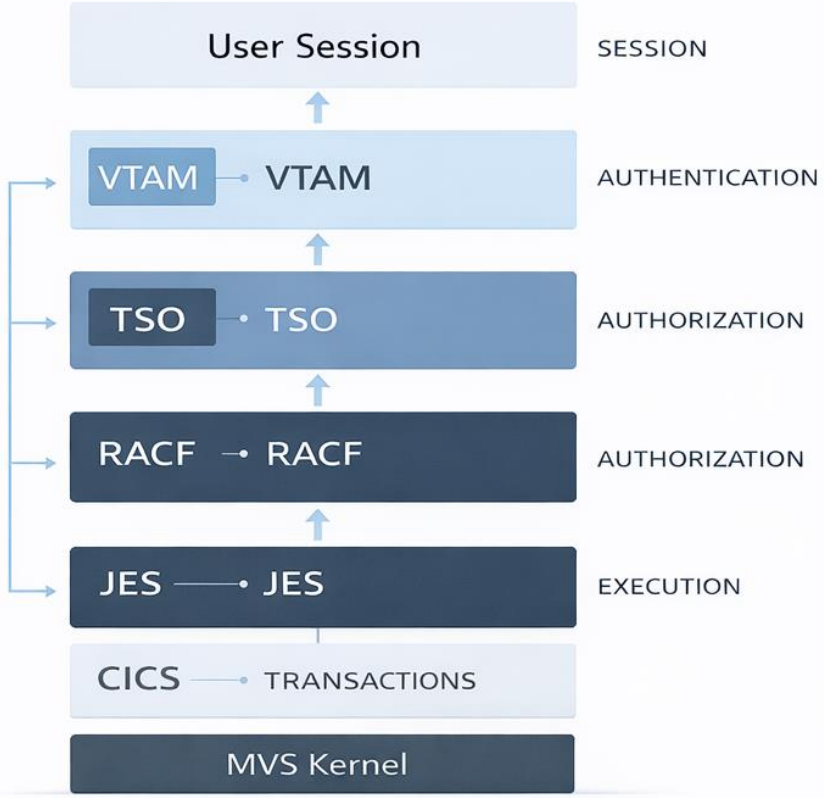
IBM

AI-POWERED MAINFRAME OPERATIONS

MVS 3.8j Turnkey system. This is a free, open-source mainframe OS running on Hercules emulator.

This Is Not a Server

- No shell model
- No process ownership
- Security decisions outside the kernel



Security decisions are distributed across subsystems, not centralized in the kernel.

Attack Paths Follow Control Planes, Not Hosts

Modern attackers don’t move process-to-process. On mainframes, movement happens by changing control context.

Real Attack Movement Model:

1. Session → Identity (VTAM → TSO)
Enumerate reachable APPLIDs (TSO, CICS, IMS)
Identify valid user IDs via response behavior
Session binding becomes the initial foothold

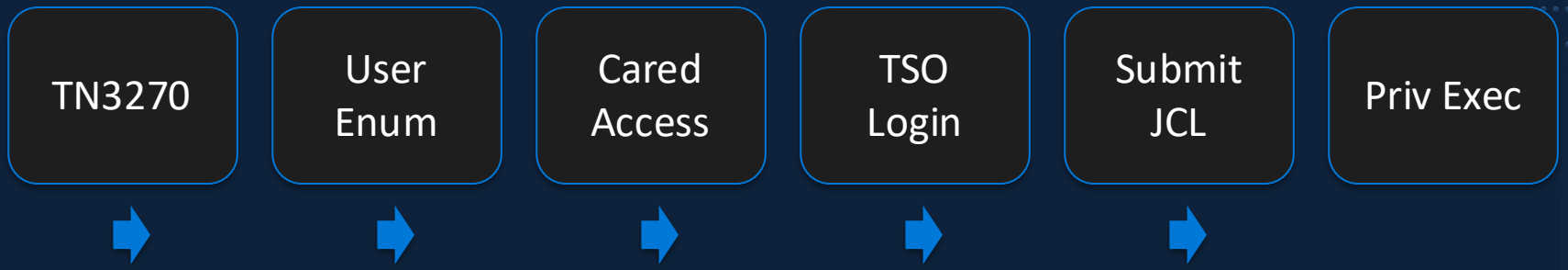
2. Identity → Authority (TSO → RACF)
Weak group membership
Over-broad dataset profiles
Excessive program execution rights
Privilege is profile-based, not account-based

3. Authority → Execution (TSO → JES)
Submit jobs that run later
Execution persists after logout
Job runs with submitter identity
Deferred execution = delayed privilege abuse

4. Execution → Persistence (JES / Started Tasks)
Modify PROCLIB or system datasets
Abuse long-lived address spaces
Change batch workflows instead of processes
5. Business Impact (CICS / Transactions)
• Invoke high-privilege transactions

Attack Reality: Mainframe Intrusions Follow Control Planes

- Modern model: scan → exploit → shell → escalate
- Mainframe progression:
- VTAM — session access
- TSO — identity binding
- RACF — authority expansion
- JES — deferred privileged execution
- CICS — business transaction impact
- Attackers move through control planes, not privilege levels



Realistic Entry Path

- Initial access rarely requires exploitation
- Shared or reused TSO credentials
- Exposed TN3270 access
- Vendor or service accounts
- Web or middleware trust chains
- Sequence: VTAM → TSO → JES → delayed execution
- Access now, impact later under system context

Operational Reality

- No traditional EDR or endpoint visibility
- Limited system telemetry outside subsystem logs
- Execution occurs outside interactive sessions (JES / batch)
- Minimal real-time monitoring of user activity
- Logging fragmented across subsystems (TSO, RACF, JES, CICS)
- Long-lived address spaces reduce behavioral context
- Security visibility depends on configuration, not defaults
- Operational opacity to modern SOC tooling

The Assessment Gap

Modern offensive assumptions drive assessments

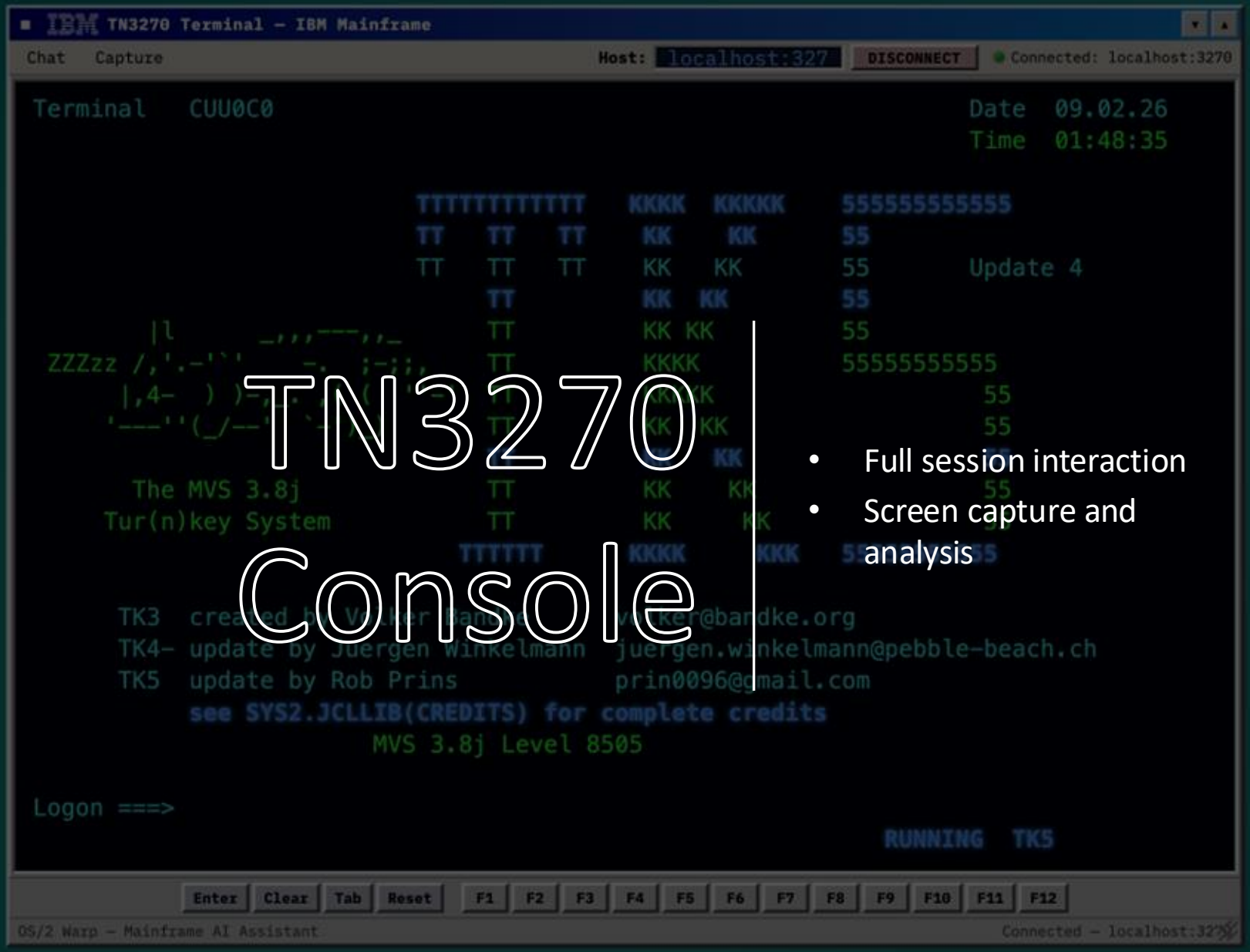
- Processes, ports, root, filesystem
- These models do not map to mainframe architecture
- Real exposure lives in subsystems and control planes

Platform Overview

Local-first assessment platform

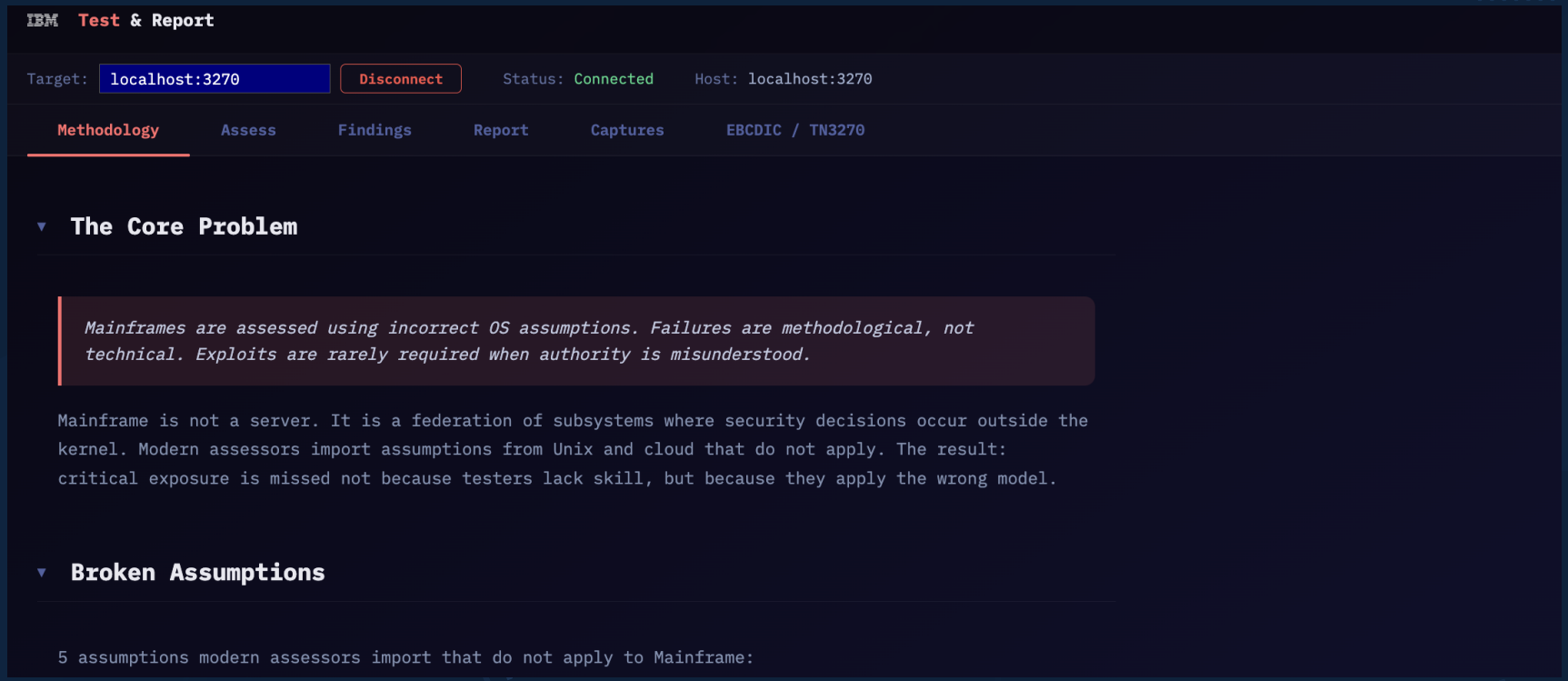
Control-plane analysis

Offline AI reasoning



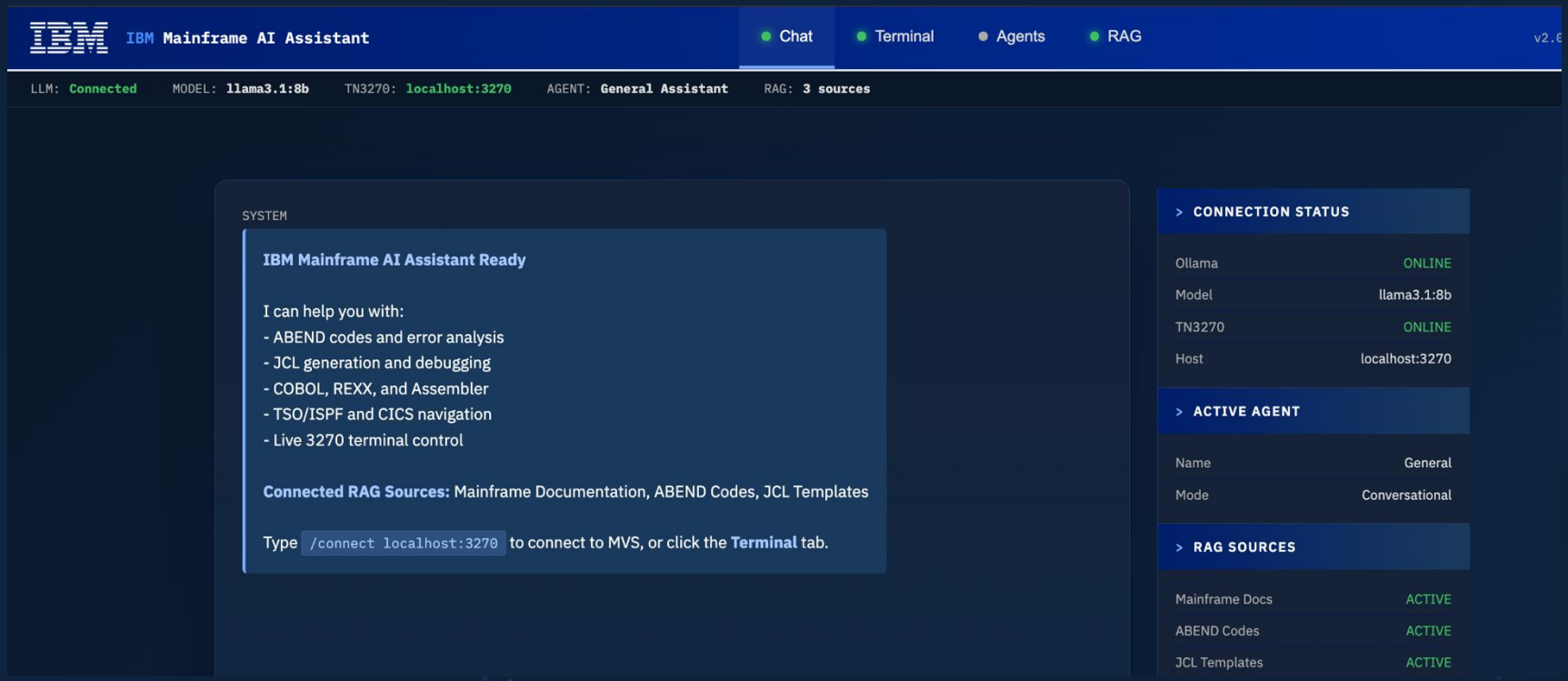
TN3270 Console

- Full session interaction
- Screen capture and analysis



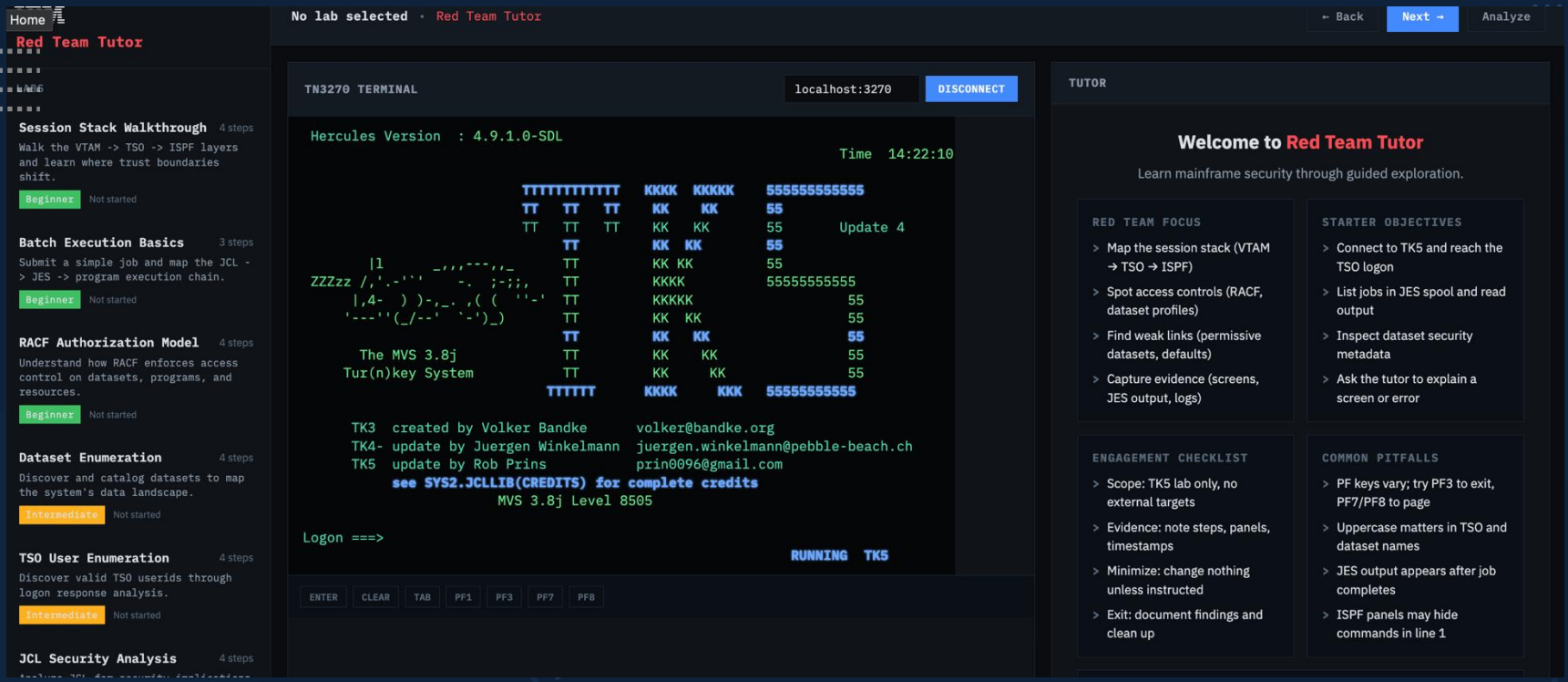
TEST AND REPORT Module

- TSO user enumeration
- VTAM exposure discovery



AI Screen Analysis

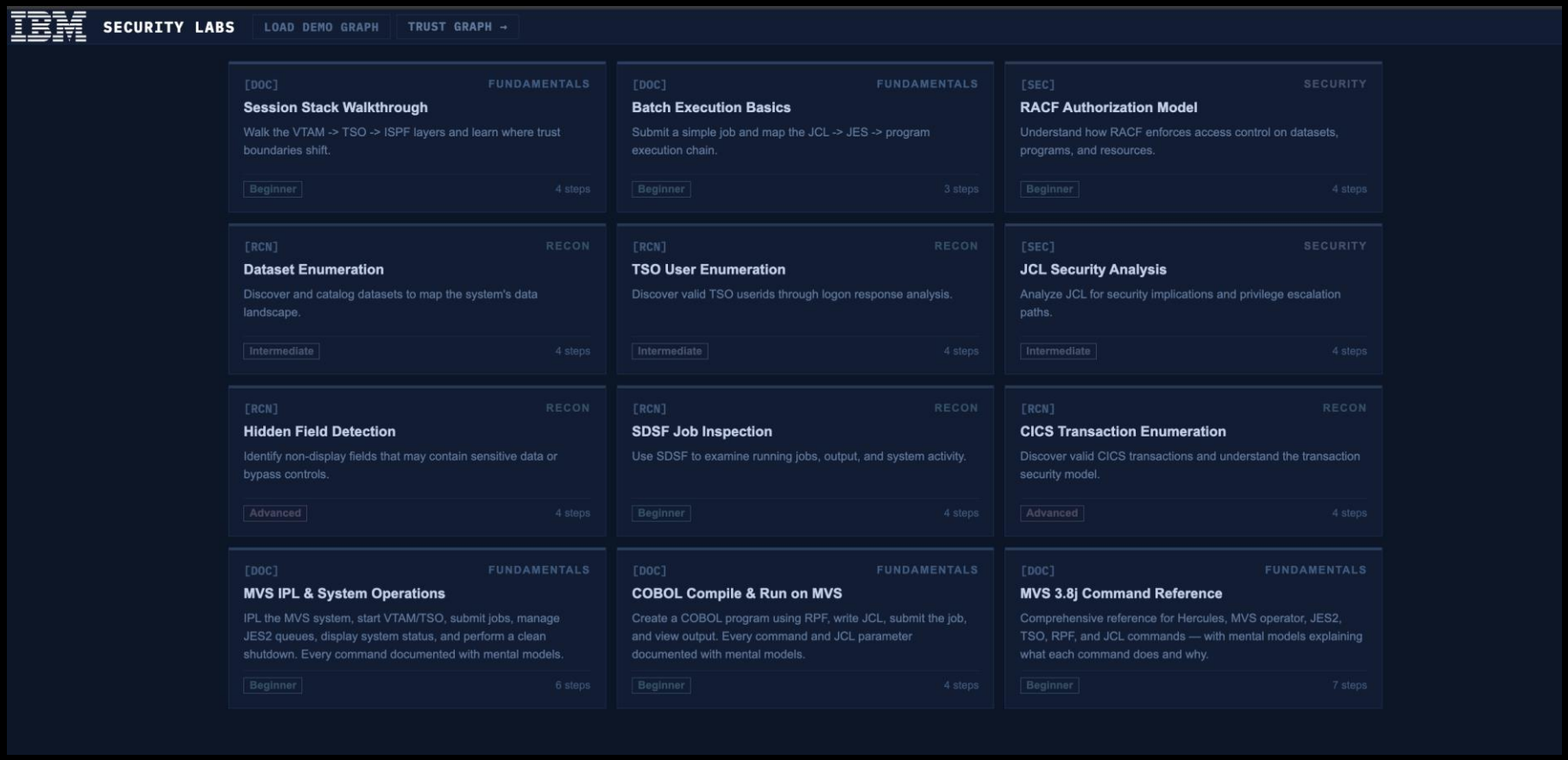
- Control-plane mapping
- Authority implications



Red Team Guided Workflow

- Step-by-step assessment
- Corrects modern assumptions

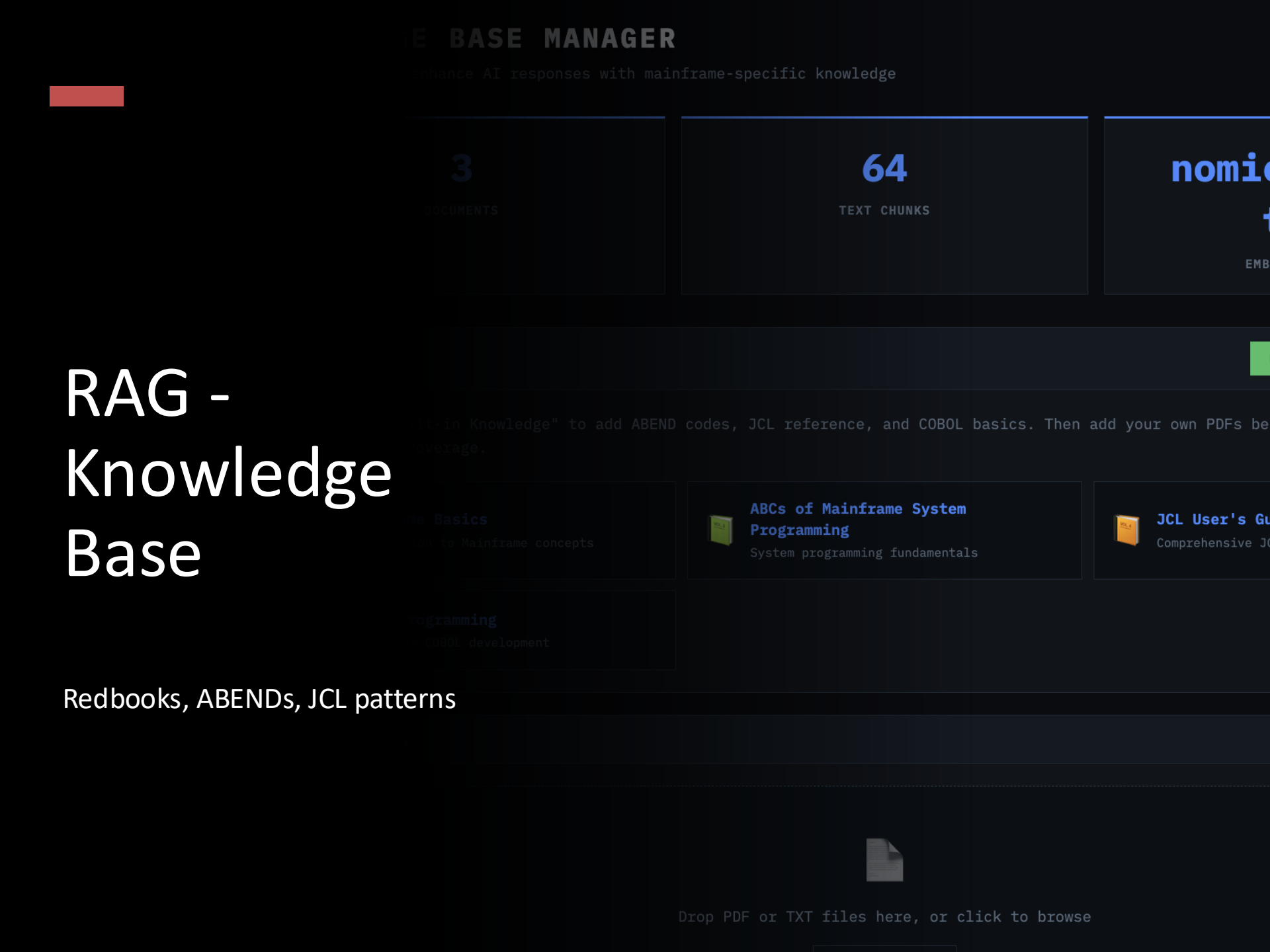
Security Labs



- Offline deterministic scenarios

RAG - Knowledge Base

Redbooks, ABENDs, JCL patterns



Trust Graph

- Users, jobs, datasets, relationships

Why Traditional Security Fails

- Modern Tools Assume:
- • Hosts and processes
- • Real-time execution
- • Endpoint telemetry
- • Continuous session context

- Mainframe Reality:
- • Session-based exposure (VTAM)
- • Identity-bound execution
- • Deferred batch processing (JES)
- • Security enforced by subsystems
- • Limited runtime visibility

What This Enables

- Repeatable mainframe recon
- Automated control-plane findings

Finding Model

- Mapped to F1–F5
- Subsystem context
- Defensive impact

Why This Matters

- Mainframes are not legacy
- They are misunderstood
- Visibility changes outcomes

Release

- Open source
- Local-first
- Offensive practitioner focused

Takeaways

- Stop thinking in hosts
- Think in control planes
- Identity and execution define risk