

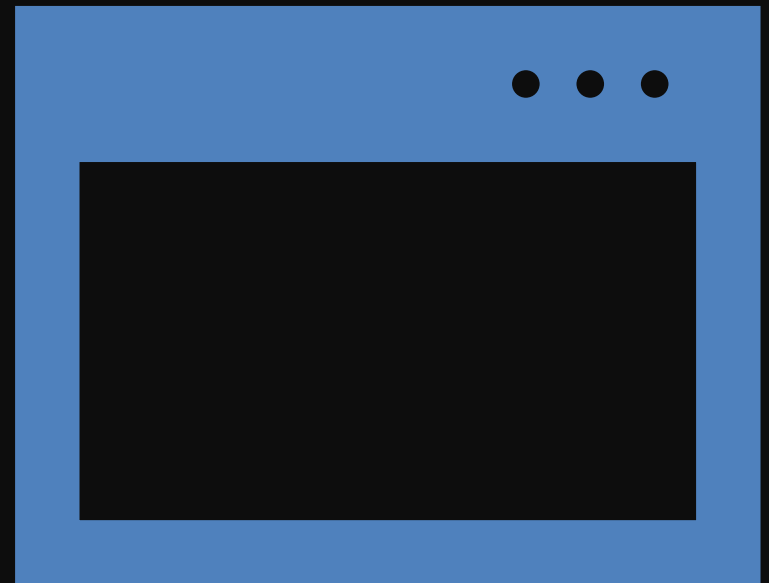
The Pre-Internet Computer

Offensive Security on
Mainframe Control Planes

Adam Toscher

The Problem

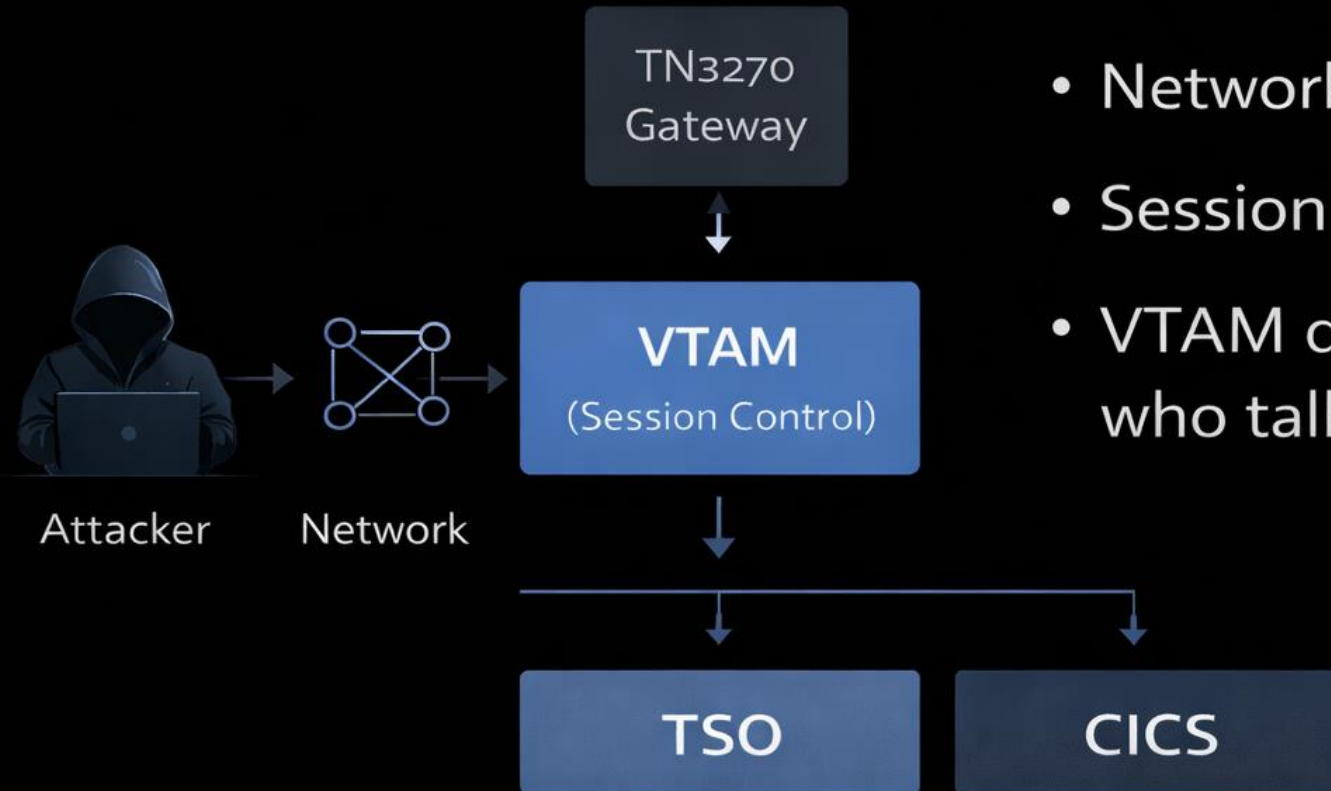
- Modern offensive assumptions:
 - Processes
 - Ports
 - Root
 - Filesystems
- These fail on mainframes.



-
- 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 \neq Access
- Session = Exposure
- VTAM decides who talks to what

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



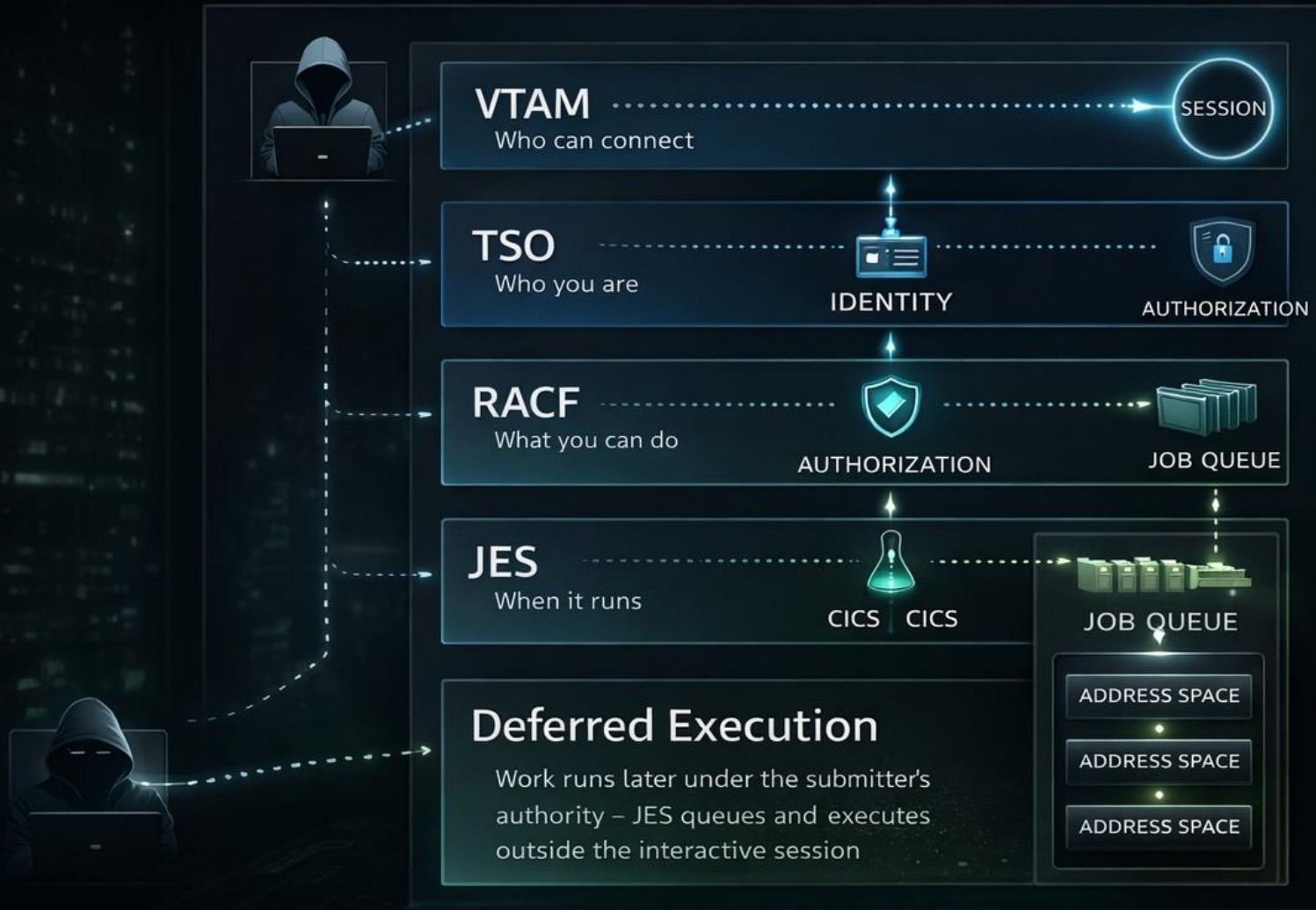
▼ CONTROL PLANES



- **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





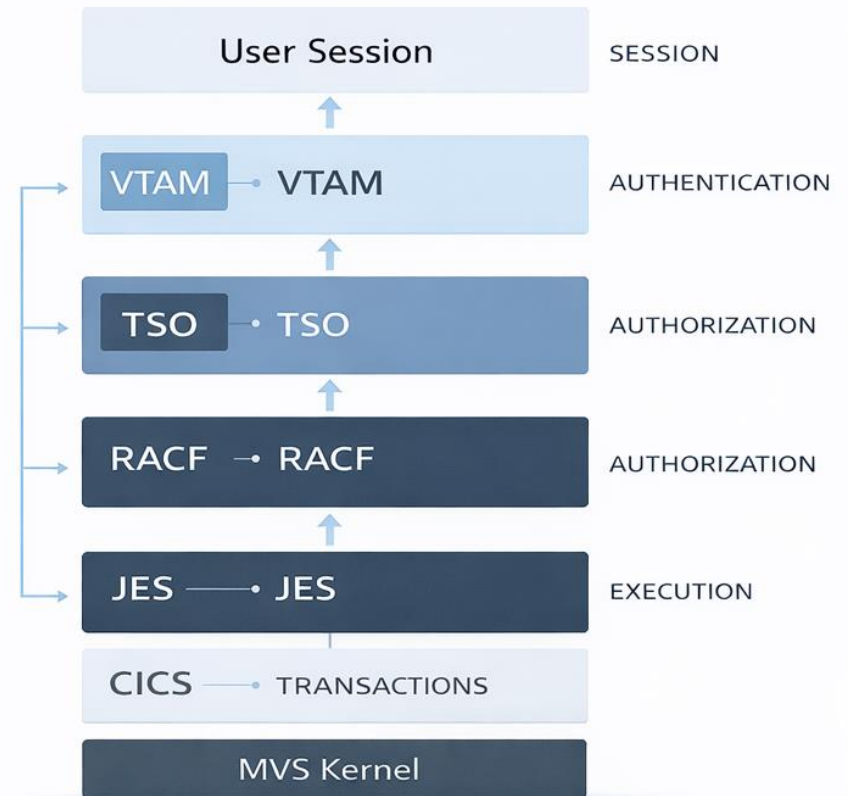
AI-POWERED MAINFRAME OPERATIONS



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

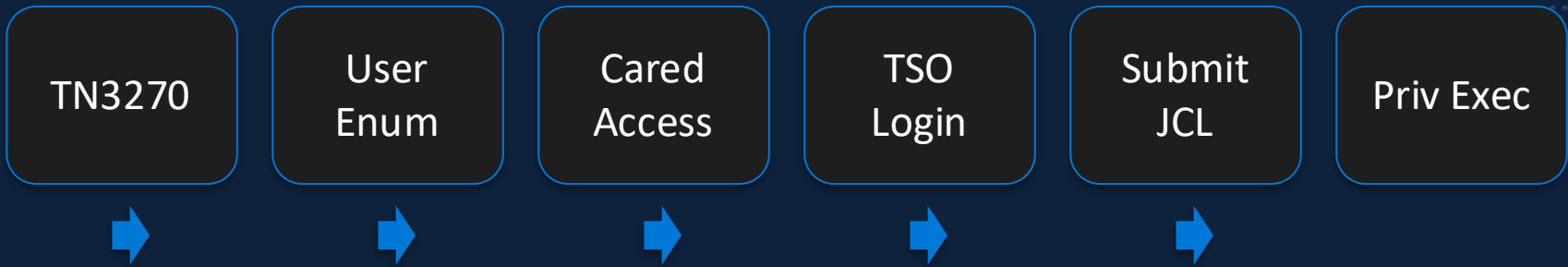
TSO — identity binding

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



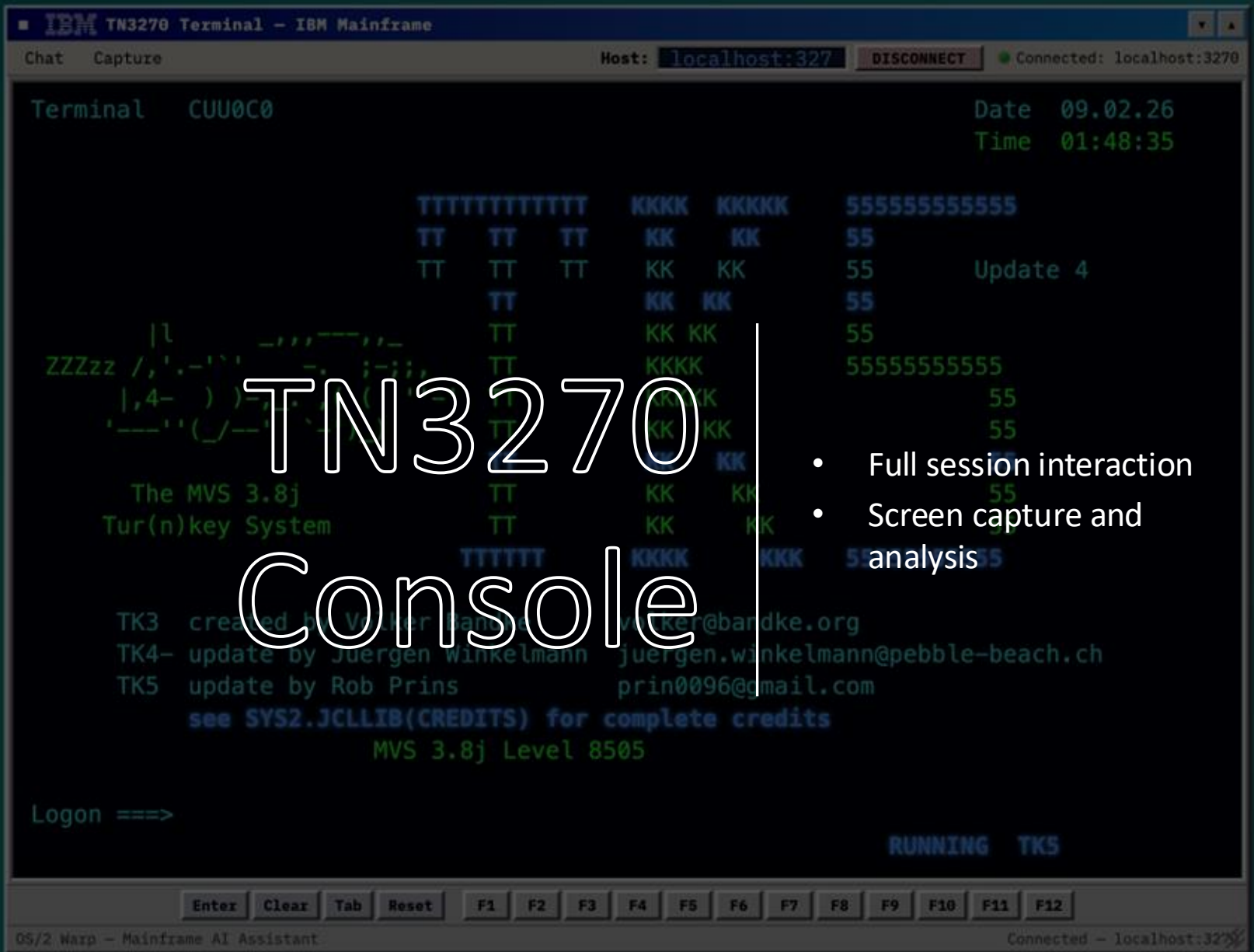
Local-first
assessment
platform



Control-plane
analysis



Offline AI
reasoning



TN3270 Console

- Full session interaction
- Screen capture and analysis

Target: localhost:3270

Disconnect

Status: Connected

Host: localhost:3270

Methodology

Assess

Findings

Report

Captures

EBCDIC / TN3270

▼ The Core Problem

Mainframes are assessed using incorrect OS assumptions. Failures are methodological, not technical. Exploits are rarely required when authority is misunderstood.

Mainframe is not a server. It is a federation of subsystems where security decisions occur outside the kernel. Modern assessors import assumptions from Unix and cloud that do not apply. The result: critical exposure is missed not because testers lack skill, but because they apply the wrong model.

▼ Broken Assumptions

5 assumptions modern assessors import that do not apply to Mainframe:

TEST AND REPORT Module

- TSO user enumeration
- VTAM exposure discovery



LLM: **Connected** MODEL: llama3.1:8b TN3270: localhost:3270 AGENT: General Assistant RAG: 3 sources

SYSTEM

IBM Mainframe AI Assistant Ready

I can help you with:

- ABEND codes and error analysis
- JCL generation and debugging
- COBOL, REXX, and Assembler
- TSO/ISPF and CICS navigation
- Live 3270 terminal control

Connected RAG Sources: Mainframe Documentation, ABEND Codes, JCL Templates

Type `/connect localhost:3270` to connect to MVS, or click the **Terminal** tab.

> CONNECTION STATUS

Ollama	ONLINE
Model	llama3.1:8b
TN3270	ONLINE
Host	localhost:3270

> ACTIVE AGENT

Name	General
Mode	Conversational

> RAG SOURCES

Mainframe Docs	ACTIVE
ABEND Codes	ACTIVE
JCL Templates	ACTIVE

AI Screen Analysis

- Control-plane mapping
- Authority implications

Security Labs

The screenshot displays the IBM Security Labs interface. At the top, there is a navigation bar with the IBM logo, the text "SECURITY LABS", and two menu items: "LOAD DEMO GRAPH" and "TRUST GRAPH →". Below the navigation bar is a grid of 12 lab cards, each with a title, a brief description, a difficulty level, and a step count.

Category	Lab Title	Description	Difficulty	Steps
[DOC] FUNDAMENTALS	Session Stack Walkthrough	Walk the VTAM -> TSO -> ISPF layers and learn where trust boundaries shift.	Beginner	4 steps
[DOC] FUNDAMENTALS	Batch Execution Basics	Submit a simple job and map the JCL -> JES -> program execution chain.	Beginner	3 steps
[SEC] SECURITY	RACF Authorization Model	Understand how RACF enforces access control on datasets, programs, and resources.	Beginner	4 steps
[RCN] RECON	Dataset Enumeration	Discover and catalog datasets to map the system's data landscape.	Intermediate	4 steps
[RCN] RECON	TSO User Enumeration	Discover valid TSO userids through logon response analysis.	Intermediate	4 steps
[SEC] SECURITY	JCL Security Analysis	Analyze JCL for security implications and privilege escalation paths.	Intermediate	4 steps
[RCN] RECON	Hidden Field Detection	Identify non-display fields that may contain sensitive data or bypass controls.	Advanced	4 steps
[RCN] RECON	SDSF Job Inspection	Use SDSF to examine running jobs, output, and system activity.	Beginner	4 steps
[RCN] RECON	CICS Transaction Enumeration	Discover valid CICS transactions and understand the transaction security model.	Advanced	4 steps
[DOC] FUNDAMENTALS	MVS IPL & System Operations	IPL the MVS system, start VTAM/TSO, submit jobs, manage JES2 queues, display system status, and perform a clean shutdown. Every command documented with mental models.	Beginner	6 steps
[DOC] FUNDAMENTALS	COBOL Compile & Run on MVS	Create a COBOL program using RPF, write JCL, submit the job, and view output. Every command and JCL parameter documented with mental models.	Beginner	4 steps
[DOC] FUNDAMENTALS	MVS 3.8j Command Reference	Comprehensive reference for Hercules, MVS operator, JES2, TSO, RPF, and JCL commands — with mental models explaining what each command does and why.	Beginner	7 steps

- Offline deterministic scenarios

KNOWLEDGE BASE MANAGER

Enhance AI responses with mainframe-specific knowledge

3

DOCUMENTS

64

TEXT CHUNKS

nomio

EMBEDDINGS

RAG - Knowledge Base

"Mainframe Knowledge" to add ABEND codes, JCL reference, and COBOL basics. Then add your own PDFs for additional coverage.

COBOL Basics

Introduction to Mainframe concepts

COBOL Programming

COBOL development



ABCs of Mainframe System
Programming

System programming fundamentals



JCL User's Guide

Comprehensive JCL reference

Redbooks, ABENDs, JCL patterns



Drop PDF or TXT files here, or click to browse

EntryPoint	2
Panel	3
CICSRegion	1
Transaction	2
Job	2
Proc	0
Program	2
Dataset	3
Loadlib	1
ReturnCode	1

QUERIES

- Paths to Job Submit
- Library Load Chain
- Shared Datasets
- Boundary Crossings
- ABEND Chains
- Dataset Conflicts
- Loadlib Hotspots
- Dataset Fan-out
- Job-Program Chains
- Orphan Datasets

```

graph TD
    TSO_LOGON --> PAY1
    TSO_LOGON --> TEST_DATA_FI
    PAY1 --> DEMOJOB
    TEST_DATA_FI --> DEMOJOB
    DEMOJOB --> ISPF_PRIMARY
    DEMOJOB --> CICS_PROD
    DEMOJOB --> ISPF_EDIT
    DEMOJOB --> PAYROLL
    ISPF_PRIMARY --> CICS_PROD
    ISPF_PRIMARY --> 0000
    CICS_PROD --> CEMT
    0000 --> CEMT
    PAYROLL --> PAYROLL_1
    IEBGENER --> SDSF
    SDSF --> DEMOJOB
    SYS1_LINKLIB --> DEMOJOB
    PROD_DATA_FI --> DEMOJOB
  
```

Select a node to view details

INGEST DATA

JCL ▼

Paste JCL or SYSOUT here...

INGEST TO GRAPH

FINDING DRAFT

Finding title

GENERATE FROM QUERY

No finding generated yet.

QUERY RESULTS

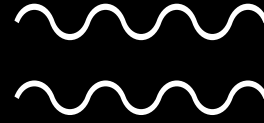
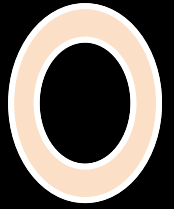
Run a query to see results

Trust Graph

- Users, jobs, datasets, relationships

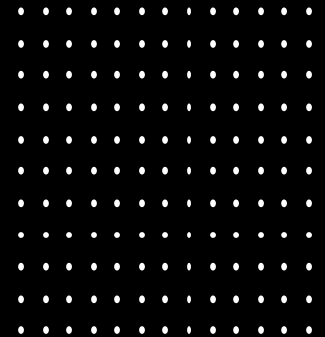
Why Traditional Security Fails

- 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



- Repeatable mainframe recon
- Automated control-plane findings

What This
Enables



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
-