The Secret Life of SAF
What Happens When You Press Enter

Kevin Segreti, Technical Consultant & Instructor
Kevin.Segreti@Broadcom.com

Agenda

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

01. What is SAF?

Architectural boundary between z/OS applications and the installed External Security Manager (ESM).

02. RACROUTE Calls

Deep dive into core macro verbs: `VERIFY`, `AUTH`, `FASTAUTH`, and `EXTRACT`.

03. Return Codes

Decoding program evaluation of RC=0, RC=4, and RC=8 return codes.

04. Passive Engine

Why RACF, ACF2, and Top Secret rely 100% on application program code for enforcement.

05. App & Code Demo

Complete walkthrough of a real z/OS CICS application and Assembler security code logic.

WHAT IS SAF?

What is SAF?

SAF stands for **System Authorization Facility**.

It is an integrated z/OS component that acts as a standardized routing interface between application programs and installed External Security Managers (ESMs) like RACF, ACF2, or Top Secret. Rather than making authorization decisions itself, SAF receives incoming security requests via RACROUTE macros and forwards them to the active ESM for profile evaluation.

Once the ESM determines permission, SAF passes the return codes back to the calling application, providing a single, consistent security gateway across the entire operating system.

Understanding SAF & The Security Router

SHARE
EDUCATE • NETWORK • INFLUENCE

THE FUNDAMENTAL TRUTH ABOUT ESMs:

The External Security Manager (RACF, ACF2, Top Secret) is purely reactive. It never monitors memory or proactively blocks code execution. It only answers questions when explicitly invoked by application code!

1. SAF Router (ICFGSAF)

- Central z/OS routing interface between applications and the ESM.
- Standardizes security interface calls across all z/OS software.

2. ESM Decision Engine

- RACF, ACF2, or Top Secret act strictly as a decision matrix.
- Evaluates database profiles and calculates return codes.

3. Application Enforcement

- Security enforcement happens 100% inside application code.
- If a program ignores RC=8, access is granted anyway!

The Big Picture: When You Press Enter

Step 1

User Action: Users actions generates a security request via CICS, TSO, Batch, or REST API.

Step 2

Application Program: Program formats parameters and issues a RACROUTE macro.

Step 3

SAF Router: z/OS Router passes request to active ESM (RACF/ACF2/Top Secret).

Step 4

ESM Decision: ESM inspects profiles and returns R15 / Reason Codes.

Step 5

Enforcement: Application inspects R15 and branches to allow or deny.

Key Architectural Insight: The security check is a request-and-response query loop. The ESM returns a mathematical decision (RC=0, RC=4, RC=8). Control returns to the calling program, which holds 100% responsibility to enforce that decision.

RACROUTE CALLS

Introduction to RACROUTE Macros

Applications invoke SAF using the standardized RACROUTE macro instruction. The macro builds a Parameter List (PLIST) in work area memory (WORKA) and transfers control to SAF.

REQUEST=VERIFY (Authentication)

Identifies and authenticates users (Password, Passphrase, PassTicket, MFA). Constructs the Accessor Environment Control Element (ACEE) in memory.

REQUEST=AUTH (Standard Authorization)

Standard SAF resource check via SVC. Checks ESM profiles, creates SMF audit records, and evaluates entity permission levels.

REQUEST=FASTAUTH (High-Speed Check)

In-memory vector table check bypassing SVC overhead. High-performance lookup preferred by CICS, IMS, and Db2.

REQUEST=EXTRACT (Profile Retrieval)

Reads or replaces specific fields inside security profiles (e.g., user segment attributes) without performing an authorization check.

RACROUTE REQUEST=VERIFY: Authentication

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

1. Credential Check

Validates user identity and password/phrase against ESM database profiles.

2. ACEE Creation

Constructs the Accessor Environment Control Element (ACEE) containing user ID, group, and privilege attributes.

3. Return to Program

Returns a primary return code (0 for success, 8 for failure) and a detailed reason code. On successful authentication with ENVIR=CREATE, it returns the memory address of the newly created **ACEE.**

CRITICAL ARCHITECTURAL DISTINCTION:

REQUEST=VERIFY strictly answers *"WHO ARE YOU?"* and creates the user security anchor (ACEE). It does **NOT** grant access to specific datasets, transactions, or resources by itself!

AUTH vs FASTAUTH

When to Use REQUEST=AUTH

- **Low-to-moderate volume checks:** Batch job initialization, dataset opens, utility execution, or one-off resource checks.
- **Immediate auditing is required:** You need guaranteed, synchronous SMF Type 80 audit record creation before program execution continues.
- **Profiles are dynamic/rarely used:** The resource class is not kept permanently in-storage (RACLISTed).

When to Use REQUEST=FASTAUTH

- **High-throughput subsystems:** Used extensively by **CICS**, **Db2**, **IMS**, and **WebSphere/zOS Connect** where thousands of authorization checks occur per second.
- **Looping authorization checks:** Checking access across hundreds of transaction options or table columns in milliseconds.
- **Execution in constrained z/OS states:** Code running in SRB mode, cross-memory mode, or under z/OS kernel locks where issuing an SVC would cause a system abend (e.g., `A03` / `0C4`).

RETURN CODES

REQUEST=AUTH/FASTAUTH Return Codes

SHARE
EDUCATE ▸ NETWORK ▸ INFLUENCE

RC = 0 (ALLOW)

ESM explicitly found a matching profile granting the requested access level (READ, UPDATE, CONTROL, ALTER).

RC = 8 (DENY)

ESM explicitly evaluated permissions and denied access, or user has insufficient authority.

RC = 4 (NO DECISION)

Resource is UNPROTECTED, class is uninitialized, or ESM has no defined rule for this entity.

CRITICAL Security Warning (RC=4):

When SAF returns `RC=4`, the ESM makes NO decision. The calling program holds 100% of the responsibility to decide if `RC=4` means **ALLOW** or **DENY**. Default fail-safe handling differs between application products!

RACROUTE REQUEST=EXTRACT Data & Audit Realities

SHARE
EDUCATE • NETWORK • INFLUENCE

Profile Data Read

Extracts segment & field metadata from ESM database (e.g., TSO, OMVS) into caller work area buffers.

The Audit Void

Unlike `AUTH` calls, `EXTRACT` do es NOT generate ESM violation records or SMF logs when fields are missing.

Common Uses

Extract is used when a secure value is needed to be retrieved from the ESM. Examples of subsystems that use extract calls are TSO, OMVS, CICS, and DB2.

AUDIT LESSON: RC=0 on an EXTRACT call only proves macro completion. The program holds full responsibility to inspect the buffer, detect missing fields, and supply fallback defaults!

PASSIVE ENGINE

The Decision Engine & The Documentation Imperative

SHARE
EDUCATE • NETWORK • INFLUENCE

THE COMMON MYTH

"The ESM Administrator knows every resource check controlling an application, and what RACF/ACF2/Top Secret resource controls it."

THE ARCHITECTURAL REALITY

The ESM is a blind decision engine. Only source code knows what resource names are queried and how return codes are evaluated!

The Documentation Mandate for Developers

- ESM admins cannot guess un-documented SAF resource names embedded inside application binaries.
- Application developers MUST document all `RACROUTE` calls, classes, entity names, and `RC=0/4/8` handling strategies.

Logging Mechanics & LOG= Parameter Suppression

SHARE
EDUCATE • NETWORK • INFLUENCE

LOG=NONE

Suppresses all auditing (successes and violations). Zero SMF Type 80 records generated in ESM logs.

Valid Pre-Auth Use Case

Used to dynamically build application UI menus based on authority, silently hiding unauthorized options.

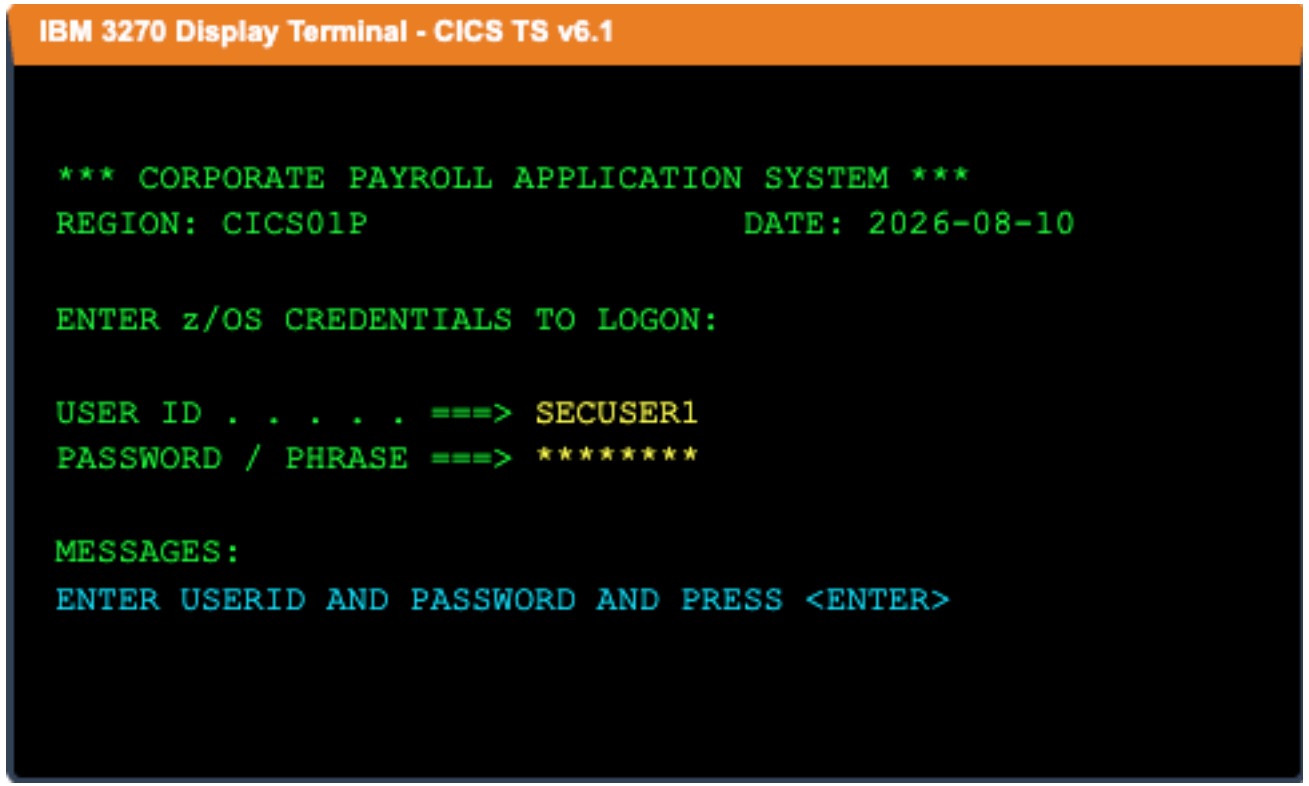
The Audit Blindspot

Suppressed calls leave ZERO audit trail. A live diagnostic trace (GTF / SAF Trace) is the ONLY way to catch these calls.

DEVELOPER WARNING: Because LOG=NONE calls are completely invisible in standard security reports, thorough application documentation is mandatory, so security auditors understand program behavior!

APPLICATION CODE AND DEMO

Phase 1: 3270 Sign-On Screen (PAYR)



Workflow Annotations

1. **User Action:** User types User ID and Password into BMS map fields and presses Enter.
2. **Program Action:** CICS program extracts credentials and prepares `RACROUTE REQUEST=VERIFY`.
3. **SAF Operation:** Authenticates user against ESM database and builds in-memory ACEE.

Phase 1 Logic: RACROUTE REQUEST=VERIFY Code

```
* PHASE 1: AUTHENTICATE USER & CREATE ACEE
         RACROUTE REQUEST=VERIFY,          X
                  USERID=USERADDR,         X
                  PASSCHK=YES,             X
                  PASSWD=PASSADDR,         X
                  ENVIR=CREATE,            X
                  WORKA=SAFWORKA
         LTR   R15,R15
         BNZ   VERIFY_FAILED
         ST    R1,USERACEE
```

Key Parameter Breakdown

- PASSCHK=YES: Validates password/phrase against ESM user profile.
- ENVIR=CREATE: Instructs ESM to build in-memory ACEE anchor.
- `Register` R15: If R15=0, ACEE address returns in R1. If R15!=0, sign-on is rejected.

Phase 2: 3270 Dynamic Menu Screen (PAYR)

IBM 3270 Display Terminal - CICS TS v6.1

```
*** PAYROLL MAIN MENU - DYNAMICALLY BUILT ***
USER: SECUSER1 | DEPT: HR01

SELECT AN OPTION:
1. VIEW PERSONAL RECORDS
2. GENERATE PAYROLL REPORT
3. UPDATE HOURLY WAGE RATES
[4. SALARY ADJUSTMENTS - HIDDEN VIA PRE-AUTH]

ENTER OPTION NUMBER ===> 2
```

Workflow Annotations

1. **Profile Data Extract:** RACROUTE REQUEST=EXTRACT reads Dept ID (HR01).
2. **Silent Pre-Auth:** RACROUTE REQUEST=AUTH,LOG=NONE tests option 4 silently.
3. **Tailored Display:** Option 4 is hidden from terminal screen without audit violation logs.

Phase 2 Logic: EXTRACT & LOG=NONE Code

```
* Step 2A: Extract User Department Field
         RACROUTE REQUEST=EXTRACT,          X
                  TYPE=EXTRACT,FIELDS=DEPT
* Step 2B: Silent Pre-Auth Menu Loop
MENU_LOOP RACROUTE REQUEST=AUTH,             X
                  CLASS='TCICSTRN',          X
                  ENTITY=TRAN_NAME,          X
                  LOG=NONE
         LTR    R15,R15
         BNZ    SKIP_MENU_ITEM
```

Key Parameter Breakdown

- REQUEST=EXTRACT: Queries user security segment to retrieve Department ID.
- LOG=NONE: Evaluates Option 4 permissions without writing SMF violation logs.
- Branching: If R15!=0, program branches to SKIP_MENU_ITEM.

Phase 3: 3270 Access Denied Screen (PAYR0004)

Workflow Annotations

1. **Forced Submission:** User attempts to execute Option 4 by directly typing option 4.
2. **Standard AUTH Call:** Program issues RACROUTE REQUEST=AUTH with default auditing.
3. **Application Enforcement:** SAF returns RC=8, ESM logs SMF Type 80, program blocks execution!

Phase 3 Logic: REQUEST=AUTH Enforcement Code

```
* Step 3: Standard Authorization Call
         RACROUTE REQUEST=AUTH,              X
                  CLASS='TCICSTRN',          X
                  ENTITY=TRAN_NAME,          X
                  ATTR=READ,WORKA=SAFWORKA
         C        R15,=F'0'
         BE       EXECUTE_OPTION
         C        R15,=F'4'
         BE       CHECK_APP_POLICY
DENIED   BAL      R14,BUILD_ERR_MAP
```

Key Parameter Breakdown

- `Audited Request`: Full RACROUTE call generates SMF Type 80 record on RC=8.
- `Return Code Inspection`: Program evaluates R15 for RC=0, RC=4, or RC=8.
- `Program Enforcement`: Program traps non-zero R15 and branches to `BUILD_ERR_MAP`.

Summary & Key Takeaways

- **1. ESM is a Passive Engine:** RACF, ACF2, and Top Secret answer queries—they do not automatically block execution in application memory.
- **2. Application Owns Enforcement:** Enforcement occurs in application code upon inspecting R15. If a program ignores RC=8, access is granted!
- **3. Beware of LOG=NONE Blindspots:** Suppressed calls generate zero SMF/ESM log records and can only be caught via live GTF/SAF traces.
- **4. Documentation is Imperative:** Developers MUST document every RACROUTE call, class, entity name, and return code handling strategy.

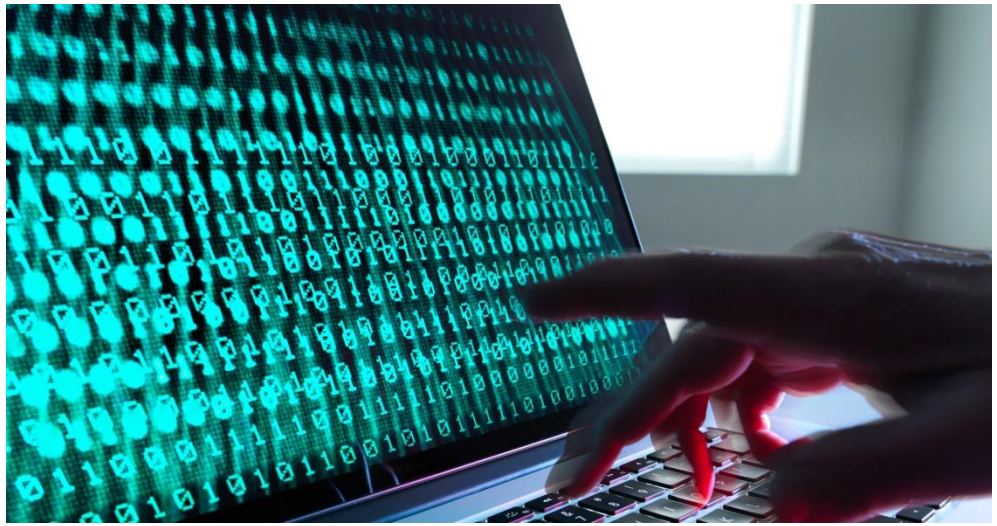
SHARE
EDUCATE • NETWORK • INFLUENCE

Thank You!

Hands-On Workshop: Ethical Mainframe Hacking (BYOD)

Join us for a **hands-on** opportunity to learn about ethical hacking and z/OS!

- Explore a vulnerable web application with a database backend,
- Learn the process of identification, fuzzing, iterating potential exploits
- Ultimately hacking into the mainframe's database



Location: **Room 409**

- Monday Aug 17 2:30-3:30pm
- Wednesday Aug 19 9:15-10:15am

Limited space — come early!

Upcoming Mainframe Conferences!

SHARE
EDUCATE • NETWORK • INFLUENCE

VSC 2026

Mainframe Technical Exchange™

GS

Vanguard Security & Compliance

- Skillful instruction on cybersecurity threats, protection strategies, system configuration, and compliance in a proven format structured to provide the highest quality learning experience.
- Series of focused workshops and hands-on labs, attendees will have the opportunity to earn CPE credits.
- Sept 14-17, 2026 in Dallas TX

Broadcom Mainframe Technical Exchange

- Network with peers and Mainframe technical experts
- Participate in technical how-to sessions and hands-on workshops, hear product updates, provide feedback
- No registration fee! Virtual MTE
- Oct. 20-22, 2026 in North American MTE Plano, TX

• GS UK | forge26

- The four-day in-person conference
- 'Forge' features keynote and plenary sessions, specialist technical streams, workshops, customer case studies, hands-on discussions, and community-led presentations covering a wide range of enterprise technology topics.
- Nov 2-5, 2026 in Whittlebury Park Towcester, UK

Technical Education Direct from the Experts

SHARE
EDUCATE • NETWORK • INFLUENCE

On-Demand Virtual

Virtual MTE
Access Sessions On-Demand through September 30, 2026

In-Person

North American MTE

Plano, TX
Oct. 20-22, 2026

Join a Mainframe Technical Exchange

- Network with peers and Mainframe technical experts
- Participate in technical how-to sessions and hands-on workshops, hear product updates, provide feedback
- No registration fee!

"The event covered all the relevant topics, and the content was amazing. The hands-on experience was also a good way to get actual experience with the tools."

Senthil Mathur, Manager, Accenture