HOW TO BREAK INTO z/OS SYSTEMS

Stuart Henderson
the Henderson Group
Bethesda, MD
(301) 229-7187
www.stuhenderson.com

AGENDA

- **Introduction and Architecture**
- **Getting To First Base**
- **Getting to Second Base**
- **Getting to Third Base**
- **Coming Home**
- **Summary and Call to Action**

INTRODUCTION AND ARCHITECTURE

- **Background**
- **Two Types of Protection**
- **Four Steps to Scoring**
- **This applies to: MVS and z/OS**

Background

- **MVS (z/OS) systems provide a solid security architecture. But they can be hacked if we don't maintain the architecture or if we don't use the security software properly**
- **Today we'll show you how the architecture provides a solid wall, and how we often open backdoors in the wall.**
- **We'll then start to fill in the details of how these backdoors can be abused, and how to protect against such abuse**

Two Types of Potential Hacker

  - **Charlie Oldtimer, current or former employee who knows the structures and the naming conventions**

  - **Nosy Outsider, just browsing, without any insider knowledge**

- **Most computer crimes are committed by Charlie, but our Internet connections are exposing our mainframes to Nosy.**

Two Types of Protection

- **Trusted Computing Base**
  - **Hardware Controls**
  - **Operating System Uses Them for Two Purposes**

- **Security Software (RACF, ACF2, TopSecret)**
  - **Q1: Who is this user?**
  - **Q2: Can This User Do X?**

Trusted Computing Base: Hardware Controls in MVS and Two Purposes

- **Hardware Controls (Used to Restrict)**
  - Supervisor State (instructions you can execute)
  - Protect Keys (memory you can update)
  - Address Spaces (memory you can access)

Two Purposes

- MVS Uses The Hardware Controls to Prevent Users from Interfering with:
  - Other Users
  - MVS Itself

Why Do You Suppose That

There are so few MVS-specific viruses?

Backdoors to the Hardware Controls

- Almost always the result of installing purchased software
- Occasionally the result of system programmer modifications
- Always better control with a QA program, and a “standard to compare against”

Backdoors to Get Supervisor State

- User SVCs (Supervisor Calls) to request MVS to execute a privileged function
- APF(Authorized Program Facility) Authorization (uses MODESET)
- TSO/APF Authorization
- I/O Appendages
- Functional Subsystems (e.g. JES, DB2, etc.)
- Exits

Backdoors to Get Protect Key Zero

- APF Authorization (uses MODESET)
- Program Properties Table (which also lets you specify that when the program opens a dataset, then Open is NOT to call RACF or TopSecret)

Backdoors to Cross Address Spaces

- **CSA (Common System Area)**
- **SRBs (Service Request Blocks, to schedule a program to execute in another address space**
- **Cross memory services,**
- **Data spaces and Hiperbatch,**

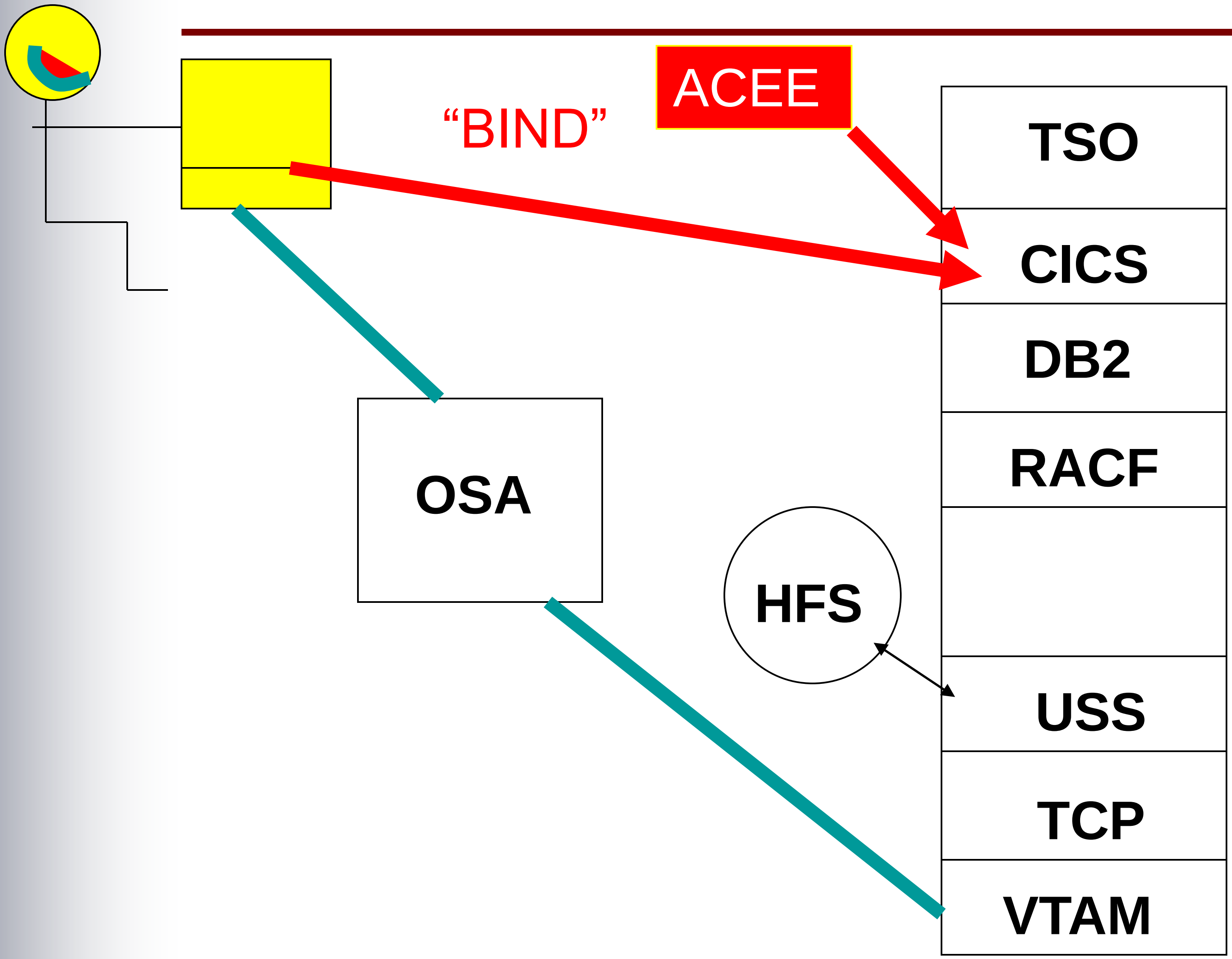
Of These Backdoors:

- Crossing Address Spaces Offers the Least Opportunity
- User SVCs and APF Authorization Are the Most Commonly Abused
- Most are defined, and controlled, in the dataset SYS1.PARMLIB

Security Software: Two Big Questions

- Q1: Who is this User?
  - Something he knows (a password or encryption key)
  - Something he holds (a key or smart card)
  - Something he is (biometrics)
- Q2: Can this User Do X?
  - Compare UserId to Dataset or Resource Rules

How Your Terminal Connects to the Mainframe



Security Software Works Effectively Only When

- It always gets control ("Always-Call")
- It fails requests which have no matching rule ("Protectall")
- It is well administered

The Biggest Sources of Problems:

- Lazy Protection (for example with STCs)
- Tape Datasets
- Passwords
- Internet Connections (Culture Differences)
- Sniffer Programs
- APPN (Advanced Peer to Peer Networking)
- Shared Devices (Disk, Other)

Four Steps to Scoring

- ***First Base (Physical or Network Access)***

- **Second Base (Bind/Logon or Initiator)**

- **Third Base (Data and Resources)**

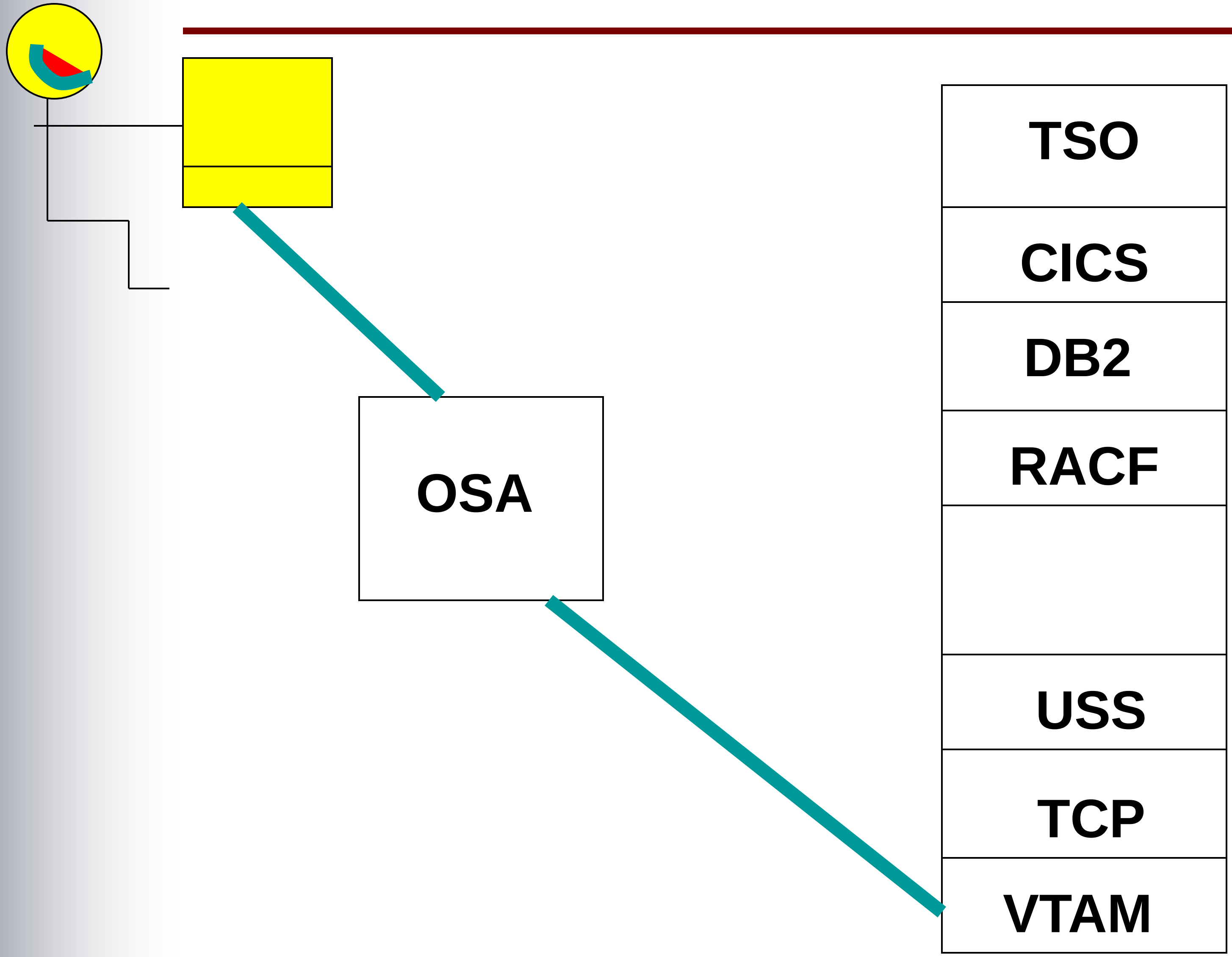
- **Coming Home (OS Privileges)**

1) GETTING TO FIRST BASE

- Physical Access

- To a Terminal or Console
  - To a Dial-up Port
  - To an IP Address
  - To APPN Network (Spoofing)
  - To a POE (Port of Entry)
  - To an Ethernet Cable
  - To a Switched Device

How Your Terminal Connects to the Mainframe



Ports of Entry (POEs) [1 of 3]

- TSO
- Internal Reader
- Started Task Initiator
- Logical Card Reader (obsolete)

Ports of Entry (POEs) [2 of 3]

- RJE
- NJE
- APPC
- USS (OMVS)
- IP

Ports of Entry (POEs) [3 of 3]

- CICS
- IMS
- DDF (in DB2, the Distributed Data Facility)
- MQSeries
- Other Applids

Three Ways to Get to the “Starting Position”

1. Walk Into an Office and Turn on a Terminal (Relies on Building Security)

2. Dial-In (Use a War Dialer to Learn the Phone Number)

3. TN3270 (Telnet for 3270 fans)

Methods of Attack for First Base

- Dial-in with a war dialer program
- Work there
- Walk in there
- Plug into a LAN
- the Internet (especially with TN3270)
- Get Access thru NJE or RJE, especially if dial-in
- Write a program which opens a TCP/IP port, call it from outside the company
- Use APPN to Spoof a CP and APPLID
- Shared Network Devices
- iPhone and other apps

Examples of First Base Attacks

- 1A) Dial-In Port
- 1B) IP Addresses
- 1C) NJE RJE

Attack Method Example 1A

- Download a war dialer program from the Internet
- Install it on your PC
- Use it to dial every phone number in an area code and centrex (the first three digits of the phone number after the area code)

Attack Method Example 1A (continued)

- When a computer answers, have the war dialer send control characters over the line. The response will tell you what type of computer answered

- Use your PC to dial the numbers where an MVS system answers. You are now on first base.

Attack Method Example 1B

- Many companies are finding it irresistible to connect their mainframes to the Internet

- You can log onto the Internet and use standard DNS (Domain Name System) commands to learn the IP addresses which correspond to a given company. You can use DNS commands to learn the IP addresses and names of all the servers that a company has connected to the Internet.

Attack Method Example 1B (continued)

- Download a port scanner from the Internet to your PC, use it to “PING” every port at each server to learn which TCP ports are active. Then try to attack each of them, one at a time.

Attack Method Example 1C

- **Get physical access to an RJE or NJE machine which is connected to the target computer**

- **Use it to send batch jobs, printouts, operator commands, or punched decks to execute, print, or punch there**

- **Can you see how three of these four could cause mischief?**

PROTECTION AGAINST FIRST BASE ATTACKS

- Call-back Boxes
- TERMINAL Protection with RACF for Dial-in Ports and for FTP IP addresses
- Firewalls
- Explicit Control of Ports and Applids with TCP/IP control file
- No modems allowed on PCs connected to a LAN (These bypass the firewall protection.)

Additional Protection With TCP/IP

- **SSL (Secure Sockets Layer, being supplanted by TLS [Transport Layer Security])**
- **Kerberos**
- **Firewalls and Policy Agent (PAGENT)**
- **The SERVAUTH Resource Class**

(More info available in articles at website:

www.stuhenderson.com)

Additional Protection With TCP/IP

- **Intrusion Detection Software**
  - **Built-In for Free**
  - **Detects Patterns of Incoming Messages Which Identify Likely Attacks**
  - **Policy Agent Software on Mainframe**

Additional Protection With APPN

- Learn Your Adjacent Networks, and their Adjacent Networks
- VTAM Options to Prevent Spoofing
- SAF Calls for VTAMAPPL and APPCLU
- See Related Presentation on Securing APPN

Four Steps to Scoring

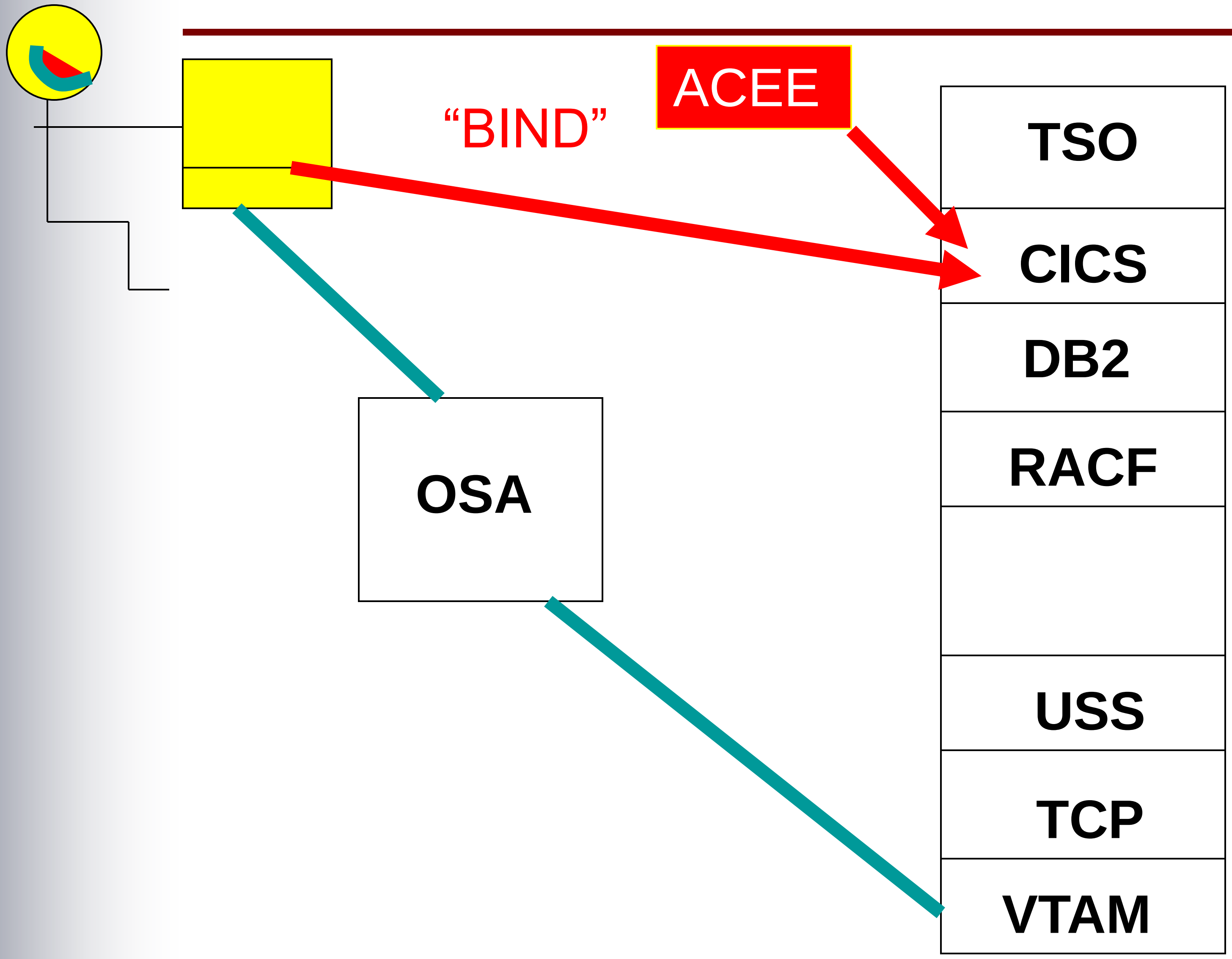
- First Base (Physical or Network Access)
- ***Second Base (Bind/Logon or Initiator)***
- Third Base (Data and Resources)
- Coming Home (OS Privileges)

2) GETTING TO SECOND BASE

(Assume that you have gotten to first.)

- **Getting a Bind and Logon**
- **Getting an Initiator (Batch Jobs and Started Tasks)**
- **Getting identified to the security software (ACEE) or bypassing it**

How Your Terminal Connects to the Mainframe



Second Base Attack Methods

- Applids which don’t call RACF (hard-coded lists of userids and passwords)
- POEs which don’t call RACF
- SYS1.UADS
- Learning passwords
- Spoofing an applid
- VTAM sessions with less than optimum security settings for identifying other LUs and CPs
- CICS Region Default Userids
- Modifying JCL for Started Tasks

Learning passwords

- Sniffing passwords on a LAN
- Spoofing by pretending to be an applid
- Guessing passwords
- Password cracker program
- Calling the Help Desk
- Watching them go by on the Internet

EXAMPLES OF SECOND BASE ATTACKS

- **2A) Bind to applid which doesn't call RACF**

- **2B) Log on thru UADS without RACF**

- **2C) Spoof an applid to learn passwords**

- **2D) Started task Userid**

Attack Method Example 2A

- Browse SYS1.VTAMLST for names of applids (or guess them)

- Find an applid which does not call the security software to check out userids and passwords

Attack Method Example 2A (continued)

- Get to first base, then logon to the applid, trying vendor supplied default userids and passwords

- If you can’t guess the userids and passwords, browse the software library to find the module where they are hard-coded. Read them there.

Attack Method Example 2B

- Browse SYS1.UADS. It often has userids which are not defined to RACF. Its passwords are not encrypted. Default security access is "anyone can read"
- Use of its userids and passwords to logon to TSO without going through RACF (or ACF2 or TopSecret).

Attack Method Example 2C

- **Log onto TSO, browse SYS1.VTAMLST to find an applid which is not in use and which has the authority to seize control of a terminal**

- **Write and execute a program which opens the applid (lying to VTAM, saying “I am applid so and so”)**

Attack Method Example 2C (continued)

- Have the program seize control of someone’s terminal (the RACF admin’s would be a good target).
- Have the program paint a fake logon screen to collect userids and passwords
- Have the program issue “Line error. Please try again.” or pass the terminal to the real applid

Attack Method Example 2D

- **Browse system control blocks in memory to find the RACF started task table.**

- **Browse the started task table to find the userids for started tasks.**

Attack Method Example 2D (continued)

- Learn the default group for each such userid, which is often the password.
- Submit a batch job, specifying USER= and PASSWORD= for the started task userid. If the batch job works, then you can do anything that that userid can.

PROTECTION AGAINST SECOND BASE ATTACKS

- **Make every applid call RACF to check userid and password**
- **Protected Userids**
- **RACF Resource Classes: JESINPUT, NODES, VTAMAPPL, TSOPROC, PROPCNTL, SURROGAT**
- **APPL Resource Class (TSO, USS, FTP, CICS, IMS, other)**

PROTECTION AGAINST SECOND BASE ATTACKS

- Default UACC of NONE (not READ)
- SETR JES(BATCHALLRACF)
- Encryption to ID user (SSL and Kerberos), Especially Kerberos with Active Directory on Windows LANs
- Verifying VTAM session security options

Four Steps to Scoring

- First Base (Physical or Network Access)
- Second Base (Bind/Logon or Initiator)
- ***Third Base (Data and Resources)***
- Coming Home (OS Privileges)

Methods of Attack for Third Base

- Assume that you have access to the system, and a successful logon or initiation

- Now you want to be able to access data to which you are not authorized

3) GETTING TO THIRD BASE

- Access to Data
  - MVS Disk Data
  - MVS Tape Data
  - Print Files
  - OMVS Files

- Access to Resources

Access by Shared Hardware

- Imagine Two CPUs with a Common, Shared DASD Device
- It Has a Sensitive Dataset, Perhaps the Accounts Payable File of Checks to Be Written
- Each CPU has a Different RACF Database, with Different Rules for This Dataset
- The Auditors Only Look at One of the RACF Databases.
- What Other Shared Devices Could Cause Security Problems?

Access to Disk Data

- Utilities like DFDSS, IOF, FDR (but not AMASPZAP, note to auditors please)
- Residual data (Tape and Disk)
- Programs Marked in Program Properties Table Not to Call RACF or TopSecret When Opening a Dataset
- DB2

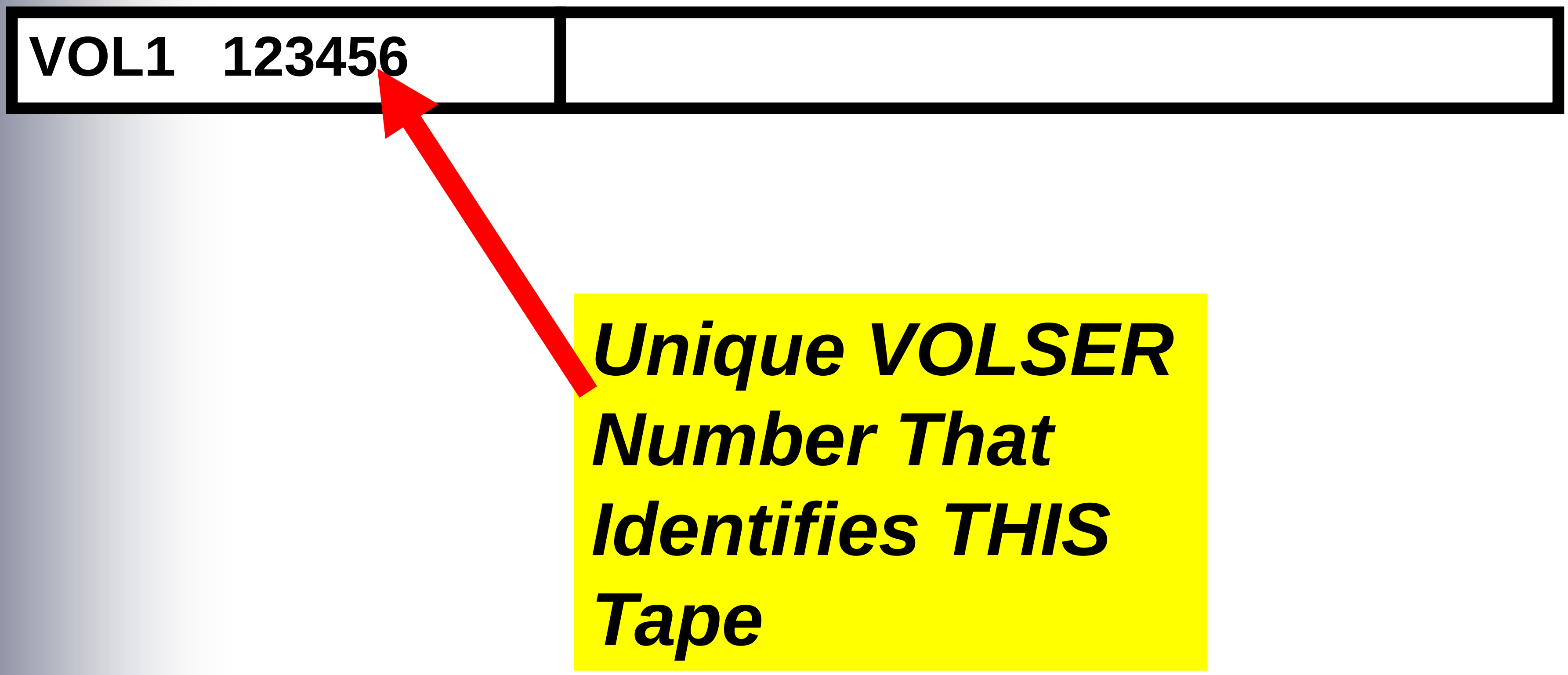
Access to Disk Data (continued)

- With RACF, Started Task marked TRUSTED or PRIVILEGED in RACF
- Clumsy RACF GLOBAL Rules (for example, SYS1.*/READ)
- OPERATIONS attribute in RACF

Access to Tape Datasets

- The “17-character dsname” weakness
- The “2 files on a tape” weakness
- Residual data (after the retention period is up)
- Many disk attacks work for tape too.
- Bypass Label Processing

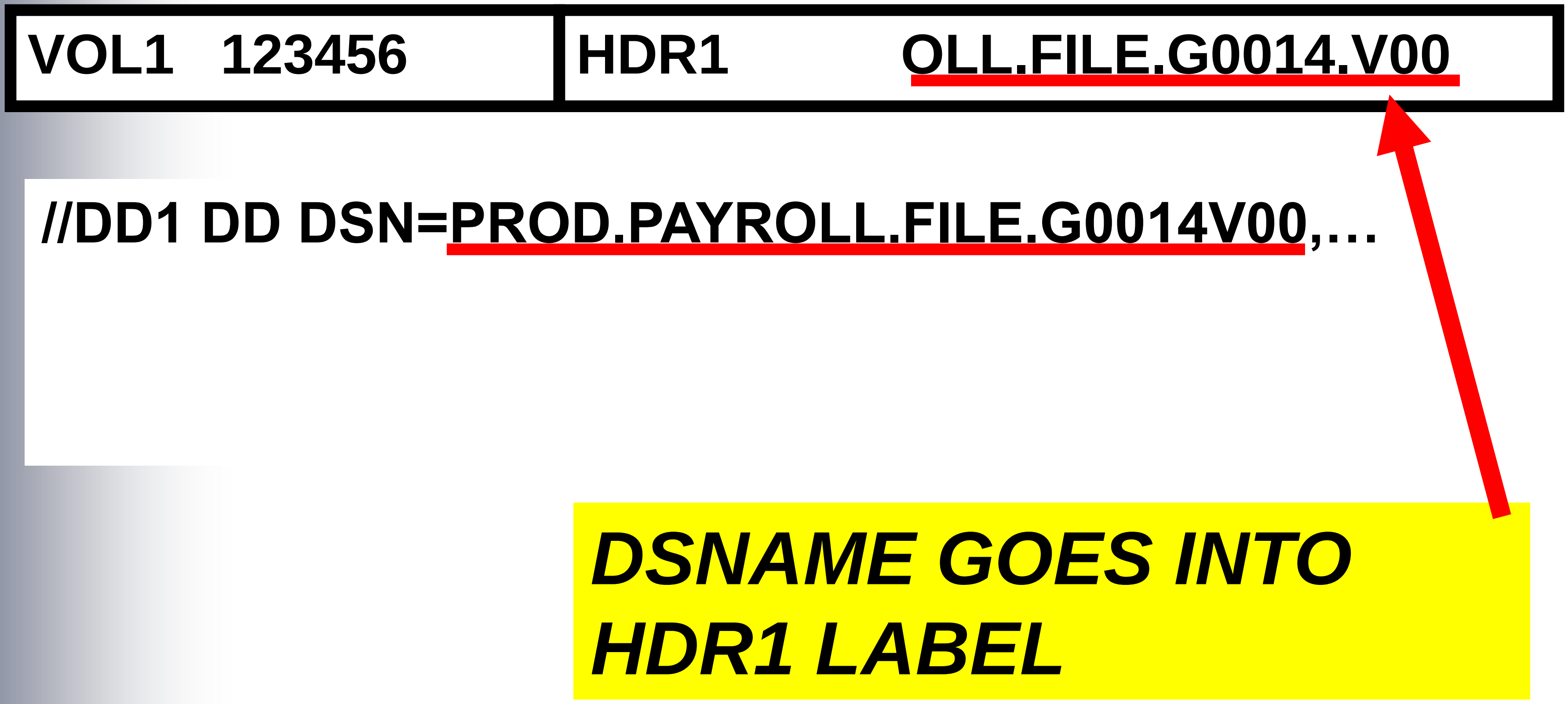
The First Two Records on a Tape Are Called Labels:



VOL1 123456

Unique VOLSER Number That Identifies THIS Tape

The First Two Records on a Tape Are Called Labels:



VOL1 123456

HDR1 OLL.FILE.G0014.V00

//DD1 DD DSN=PROD.PAYROLL.FILE.G0014V00,…

DSNAME GOES INTO HDR1 LABEL

OPEN Makes Two Checks for SL Tapes

VOL1 123456	**HDR1 OLL.FILE.G0014.V00**

```
//DD1 DD DSN=PROD.PAYROLL.FILE.G0014V00,…
```

1. **Compare DSNAME in Label Against DD Card**
2. **Call SAF with RACHECK**

But If the Rogue STU Wants to Read It:

VOL1 123456	HDR1 OLL.FILE.G0014.V00

```
//DD1 DD DSN=PROD.PAYROLL.FILE.G0014V00,…


//DD2 DD DSN=STU.XX.OLL.FILE.G0014V00,…
```

1. Compare DSNAME in Label Against DD Card
2. Call SAF with RACHECK

But If the Rogue STU Wants to Read It:

VOL1 123456	HDR1 OLL.FILE.G0014.V00

```
//DD1 DD DSN=PROD.PAYROLL.FILE.G0014V00,…


//DD2 DD DSN=STU.XX.OLL.FILE.G0014V00,…
```

1. Compare DSNAME in Label Against DD Card
2. Call SAF with RACHECK

EXAMPLE ATTACKS FOR THIRD BASE

- 3A) Residual data on disk
- 3B) Residual data on tape
- 3C) 17 character tape label
- 3D) BLP

Attack Method Example 3A

- Learn what disk packs sensitive datasets are allocated on, and when they are erased

- Just after they are erased, use ISPF to allocate large datasets on those disk packs.

Attack Method Example 3A (continued)

- Write a program to read the datasets you’ve allocated, skipping over any initial end-of-file mark

- Browse the datasets, doing a FIND for a field which you know is in the sensitive data

Attack Method Example 3B

- Find the volser of a tape which has sensitive production data, and whose retention period will shortly expire
- Wait until the retention expires, then run a program with a DD card calling for that specific tape, to write a new dataset on the tape.
- Have the program read the residual data.

Attack Method Example 3C

- Find a production dataset with a long dsname

- Run an IEBGENER with a SYSUT1 DD card specifying a dsname that has your userid as the High Level Qualifer

- Make sure that the dsname on the DD card has the rightmost 17 characters the same as on the tape standard label

Attack Method Example 3C (continued)

- When you run the IEBGENER, the DD card will match the dsname in the tape label (all 17 characters of it)
- When you run the IEBGENER, the call to RACF will allow the access, since for the full 44 character dsname, your userid will be the HLQ

Attack Method Example 3D

- Get permission to put LABEL=(2,BLP) on your DD cards
- If the security software doesn’t control it, you just need permission from JES
- If the security software does control it, stamp your foot and insist that you need it to do your job
- Read any tape dataset you feel like

Access to Print Datasets

- Software such as SDSF
- Browse the spool file directly if its RACF Dataset Rule has a UACC of READ (or the relevant GLOBAL rule)

Access to USS (OMVS) Files

- (More details in a future session)

- Use of ACLs (Access Control Lists) with USS

PROTECTION AGAINST THIRD BASE ATTACKS

- **PROTECTALL, TAPEDSN, JESSPOOL**

- **Erase on Scratch (Note super-fast DASD enhancement) See more info at www.stuhenderson.com/RUGNEW75.pdf**

- **UACC(NONE) as default, including GLOBAL rules**

PROTECTION AGAINST THIRD BASE ATTACKS

- **Security software control over BLP**
- **Tape management software and pooling**
- **Encryption (hardware and software, DB2)**
- **Tape Drive Encryption**
- **Only Send Out Brand New Tapes**
- **Restrict granting of OPERATIONS**
- **DB2 V8 use of SECLABEL and MLS**
- **Parmlib Member DEVSUPxx Fields Named TAPEAUTHxx**

Make RACF Userids Be Restricted

- ALU userid RESTRICTED
- Doesn’t Permit Access By:
  - UACC
  - ID(*)
  - GLOBAL rules
  - USS ACLs (Access Control Lists)
- Does Permit By:
  - WARNING
  - UNIX File Security Bits (Note Exception with UNIXPRIV Rule RESTRICTED.FILESYS.ACCESS)

Four Steps to Scoring

- First Base (Physical or Network Access)
- Second Base (Bind/Logon or Initiator)
- Third Base (Data and Resources)
- ***Coming Home (OS Privileges)***

4) COMING HOME

- Getting Operating System Privileges

- Best hunting is in user SVCs and APF authorized libraries. Look for short user SVCs that provide “authorization”.

- If you have supervisor state, you can get protect key zero, and vice versa.

EXAMPLE ATTACKS FOR HOME PLATE

- **4A) User SVCs**
- **4B) APF Authorization**

Attack Method Example 4A

- Logon to TSO (second base).
- Use the TEST (or REXX or whatever) command in TSO to browse control blocks, starting with address 16 (10 in hex), which has the address of the CVT or Communications Vector Table. This is the "mother of all control blocks" in MVS.

Attack Method Example 4A (continued)

- The CVT has, at displacement C8, the address of the SCVT, or Secondary CVT.

- Use the TEST command (or a REXX EXEC or whatever tool you like) to examine this. The SCVT plus displacement 84 (hex) will give you the address of the SVC table.

- Skip the first 200 entries in the SVC table. (They are covered by IBM’s integrity statement. Each entry is 8 bytes long.)

Attack Method Example 4A (continued)

- A REXX Exec might look like this:

```
CVTPTR  =    C2D(STORAGE(10,4))
          /*Set CVTPTR to address of CVT*/

SCVT       =
    C2D(STORAGE(D2X(200+CVTPTR) ,4))
         /*Set SCVT to address of sec.CVT*/
```

Attack Method Example 4A (continued)

- The first 4 bytes of each entry is the address of the program which is the SVC.
- (If you can’t read it in memory, learn the name and browse it in its library).
- Browse the program at that address. The first 100 bytes will likely give you the copyright notice, including the program name and vendor.

Attack Method Example 4A (continued)

- Use the TEST command to display the program, translating it from machine language to assembler. Continue until you find an SVC which will give you supervisor state.

- Write a program which you will execute as a batch job, and which executes that SVC.

Attack Method Example 4B

- Use the same approach as in 4A to browse system control blocks to get the address of the APF authorized library table. This is the same source used by DSMON.

- Find an APF authorized library which you can update. Your best bet is usually the library for the most recently installed software.

Attack Method Example 4B (continued)

- Write a program which issues MODESET to request operating system privileges (Supervisor State).
- Assemble and link the program. Move it into the APF library, marking it APF authorized in the directory of the PDS.
- Execute the program.

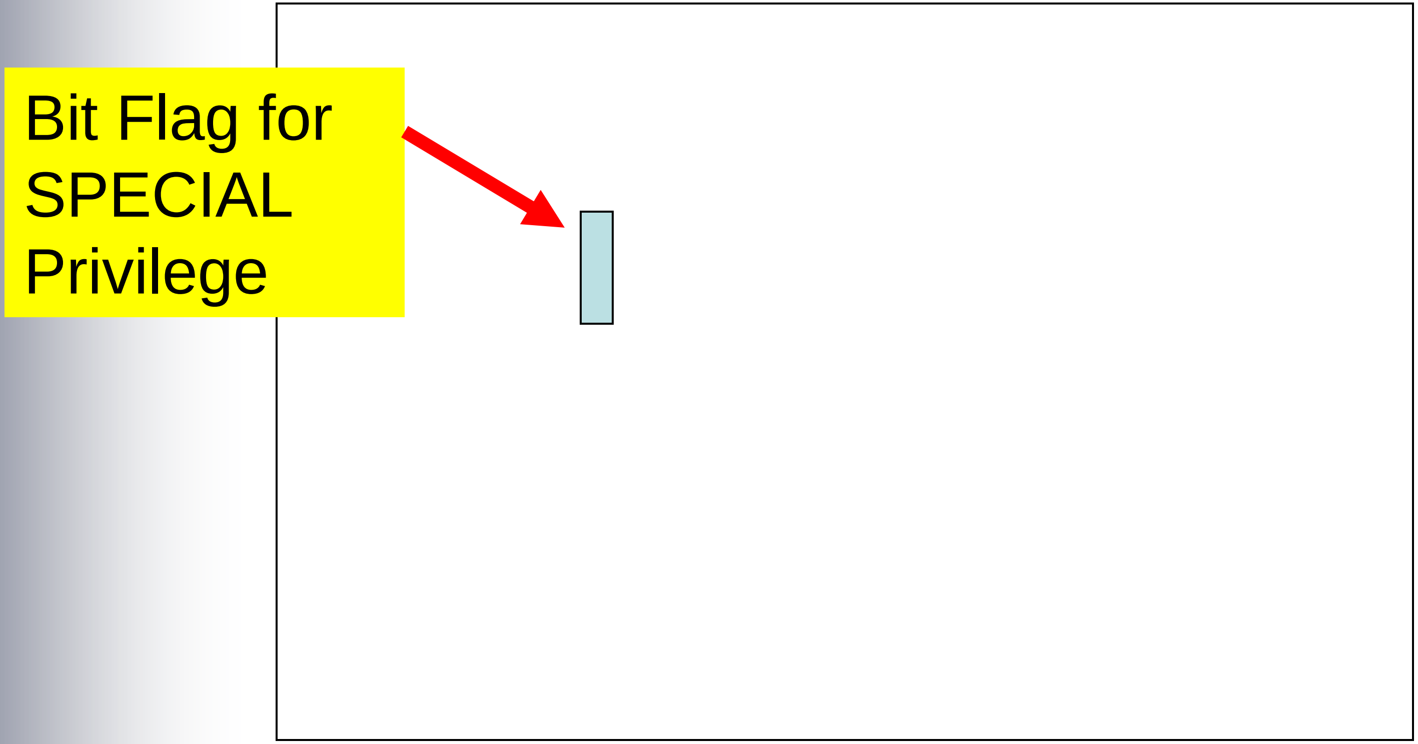
How RACF, ACF2, TSS Rely On TCB

How RACF, ACF2, TSS Rely On TCB

An ACEE (RACF Flavor)

How RACF, ACF2, TSS Rely On TCB

An ACEE (RACF Flavor)



PROTECTION AGAINST HOME PLATE ATTACKS

- **Change control and RACF control over system libraries, especially SYS1.PARMLIB.**

- **Importance of comparison to a standard**

- **PROGRAM control with security software**

PROTECTION AGAINST HOME PLATE ATTACKS

- Holding Vendors Accountable by Requesting Integrity Statements Comparable to IBM's for MVS and CA Technologies' for Their Products

- Logic in User SVCs to protect them against abuse

- See IBM manual listed at end for chapter "Protecting the System"

- Have the pitcher cover home when the catcher loses control of the ball

WHAT WE LET PEOPLE DEMAND

- **The application programmer**
- **The bank teller**
- **The system programmer**

If we truly need to update any system library any time, what does that say about the quality control in our shop?

5) SUMMARY AND CALL TO ACTION

- You can see how z/OS provides a reliable security architecture.
- You can also see ways that you can have holes in that architecture.
- Expanding the approach outlined here will show you any holes in your shop, and how to fix them.

For Further Information:

- See article on SERVAUTH and back issues of the RACF User News and Mainframe Audit News at www.stuhenderson.com
- IBM manual “MVS Programming: Authorized Services Guide”, especially the chapter “Protecting the System”
- Information on VTAM session security options and exposures at www.net-q.com

For Further Information:

- CA Technologies Integrity Statement for All of Their Software Products (http://arcserve.com/~/media/Files/TechnicalDocuments/common-integrity-statement.pdf)