Is Your z/OS System Secure?

Ray Overby
Key Resources, Inc.

March 2, 2011
Session 8511

What Are We Worried About?

- System Integrity Vulnerabilities exist in z/OS sites
- A Vulnerability can be exploited
- Exploiting a Vulnerability would:
  - Allow a user to bypass installation controls
  - Allow access to sensitive data
  - Allow modification to sensitive data
  - Allow disruption of system services
- This would be a violation of PCI, HIPAA, SOX and/or the site’s mission

What is a Vulnerability?

- A weakness in the system
- Which would allow an attacker to circumvent installation controls
- It could be caused by:
  - Hardware configuration
  - System configuration parameters
  - Security System configuration or controls
  - Lack of System Integrity caused by poor coding logic in Authorized Programs and Interfaces

What is an Exploit?

- An Exploit is a method of taking advantage of a vulnerability
- Exploits are based upon one or more vulnerabilities

What Can You Do With An Exploit?

- You can bypass installation controls
- And gain unauthorized access to data
  - Without proper permissions
  - Without proper logging (SMF)
- Exploits can be created in your environment
- Exploits can be imported from outside sources

Exploit Requirements

- May require certain software to be installed
  - Certain ISV products
  - Certain shareware programs (www.cbttape.org)
  - Certain installation developed Exits, etc.
- May require certain software maintenance levels
  - Release of z/OS or ISV products
  - Certain PTFs not installed
- May require certain features implemented or active

Vulnerabilities can be …

- Exploited by knowledgeable insiders with a high level of technical expertise
- Exploited by Script Kiddies with a lower level of technical expertise
- Can cause Compliance Violations
- Can cause loss or modification of confidential information without generating SMF or other log records

Vulnerabilities Can Be Caused By:

- Poor Hardware configuration
- Poor Operating System configuration parameters
- Poor Security System configuration or controls
- Poor coding logic in Authorized Programs and Interfaces

Poor Hardware Configuration

- Shared DASD between Test and Production Systems
- Non-shared security database
- Access to production data from test system using test system authorities
- Not only data shared, but shared system (APF) and application program libraries

Poor Operating System configuration

- IPL parameters
- STC parameters
- May let security default to don't care
- May not protect by default

Poor Security System Controls

- Failure to secure APF Libraries
- List of APF Libraries is defined in SYS1.PARMLIB
- Libraries can be added and removed via Operator Commands
- Programs in these libraries can be marked as authorized .. e.g. AC(1)
- Authorized programs can modify z/OS control blocks so as to obtain access to data without SMF records being generated

Poor Coding Logic In Authorized Programs and Interfaces

- Allows PSW Key 8 problem state programs to obtain control in an authorized state
- Once authorized, it can then
  - Dynamically modify security credentials
  - Turn off SMF or other logging
- Similar to being able to add a program to an APF Authorized Library in previous example

System Integrity vs System Security

- SHARE Security Project formed in 1972
- To develop security requirements for future IBM Operating Systems
- Problem that could not be overcome was that any security rules could be bypassed if the defined Operating System Interfaces could be circumvented
- The Security Project conclusion was:

There can be no System Security without Operating System Integrity

Three Security Systems for z/OS

- RACF developed by IBM and introduced in 1976
- ACF2 developed by SKK and introduced in 1978 (now owned by CA)
- Top Secret developed by CGA Allen and introduced in 1981 (now owned by CA)

ALL DEPEND ON PROPER z/OS CONFIGURATION, SECURITY SYSTEM CONTROLS AND SYSTEM INTEGRITY

IBM’s Commitment to z/OS Integrity

IBM’s commitment includes design and development practices intended to prevent unauthorized application programs, subsystems, and users from bypassing z/OS security – that is, to prevent them from gaining access, circumventing, disabling, altering, or obtaining control of key z/OS system processes and resources unless allowed by the installation.

Specifically, z/OS “System Integrity” is defined as the inability of any program not authorized by a mechanism under the installation’s control to circumvent or disable store or fetch protection, access a resource protected by the z/OS Security Server (RACF®), or obtain control in an authorized state; that is, in supervisor state, with a protection key less than eight (8), or Authorized Program Facility (APF) authorized.

In the event that an IBM System Integrity problem is reported, IBM will always take action to resolve it.

Every Installation's Responsibility

z/OS V1 R12 MVS Authorized Assembler Services Guide:

To ensure that system integrity is effective and to avoid compromising any of the integrity controls provided in the system, the installation must assume responsibility for the following:

- That its own modifications and additions to the system do not introduce any integrity exposures. That is, all installation-written authorized code (for example, an installation SVC) must perform the same or an equivalent type of validity checking and control that the system uses to maintain its integrity.

* It is the responsibility of the installation to verify that any authorized programs added to the system control program will not introduce any integrity exposures.

What Does It Mean?

- If you have a secure hardware configuration
- And if you have secure system configuration parameters
- And if you have secure installation security controls

But if you don’t have system integrity

You are NOT secure

System Integrity Vulnerabilities

- Are independent of the Security System
  - ACF2, RACF or Top Secret
- Must be remediated by the Code Owner
- You are dependent on the Code Owner to address them!

Who is the Code Owner?

- IBM in the case of the z/OS Operating System
- Vendors for Program Products
    - Could be IBM or other Independent Software Vendor
- Installation Staff for locally developed Operating System SVCs, System Exits, APF Authorized Programs, etc.
- ? for code obtained from other places

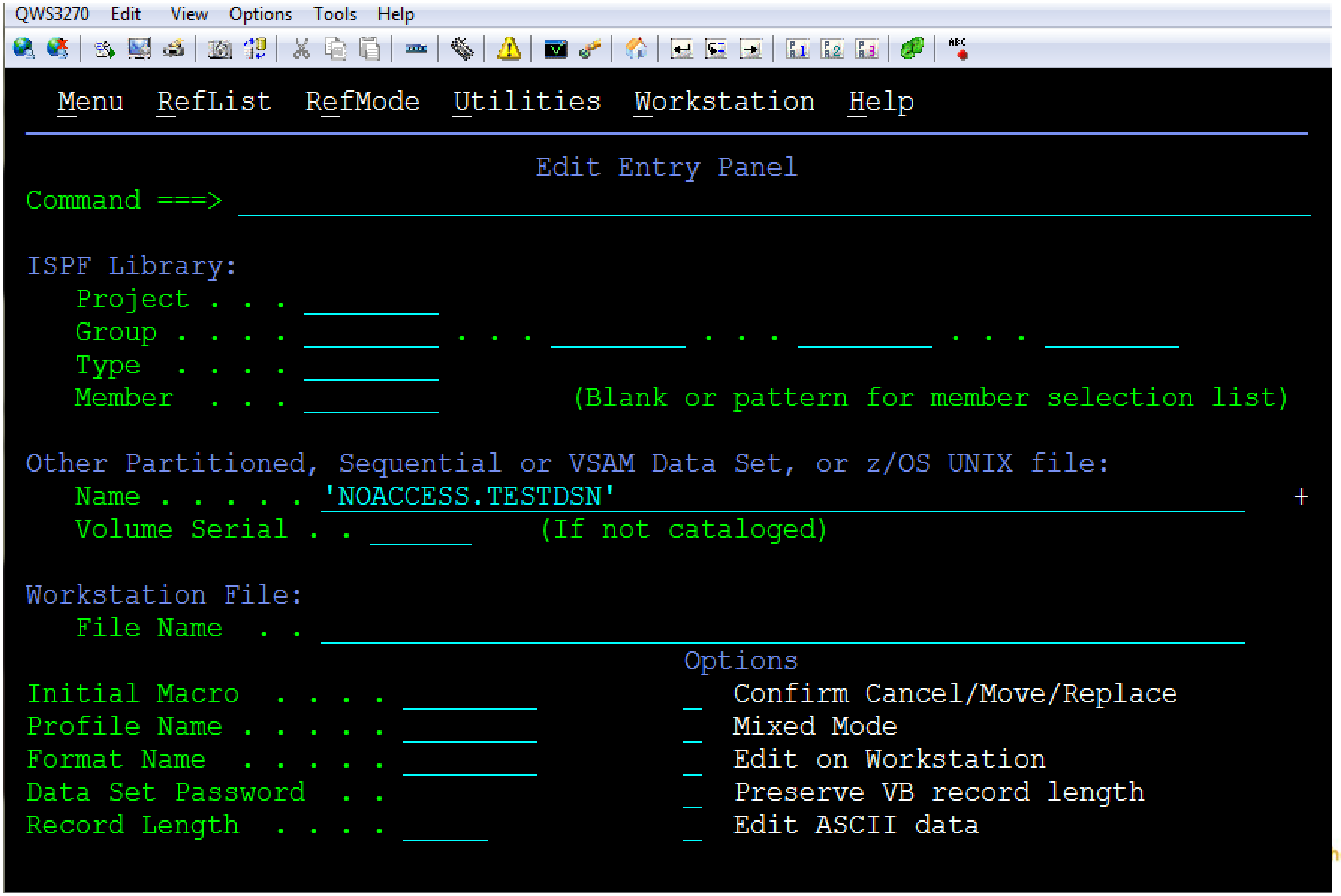
System Integrity Based Exploit

- The following is an exploit based upon a system integrity vulnerability
- z/OS R1.11 ADCD System
- No extra-ordinary security authority is required
- Security System is RACF but with minor changes it would have worked with ACF2 or Top Secret

System Integrity Based Exploit

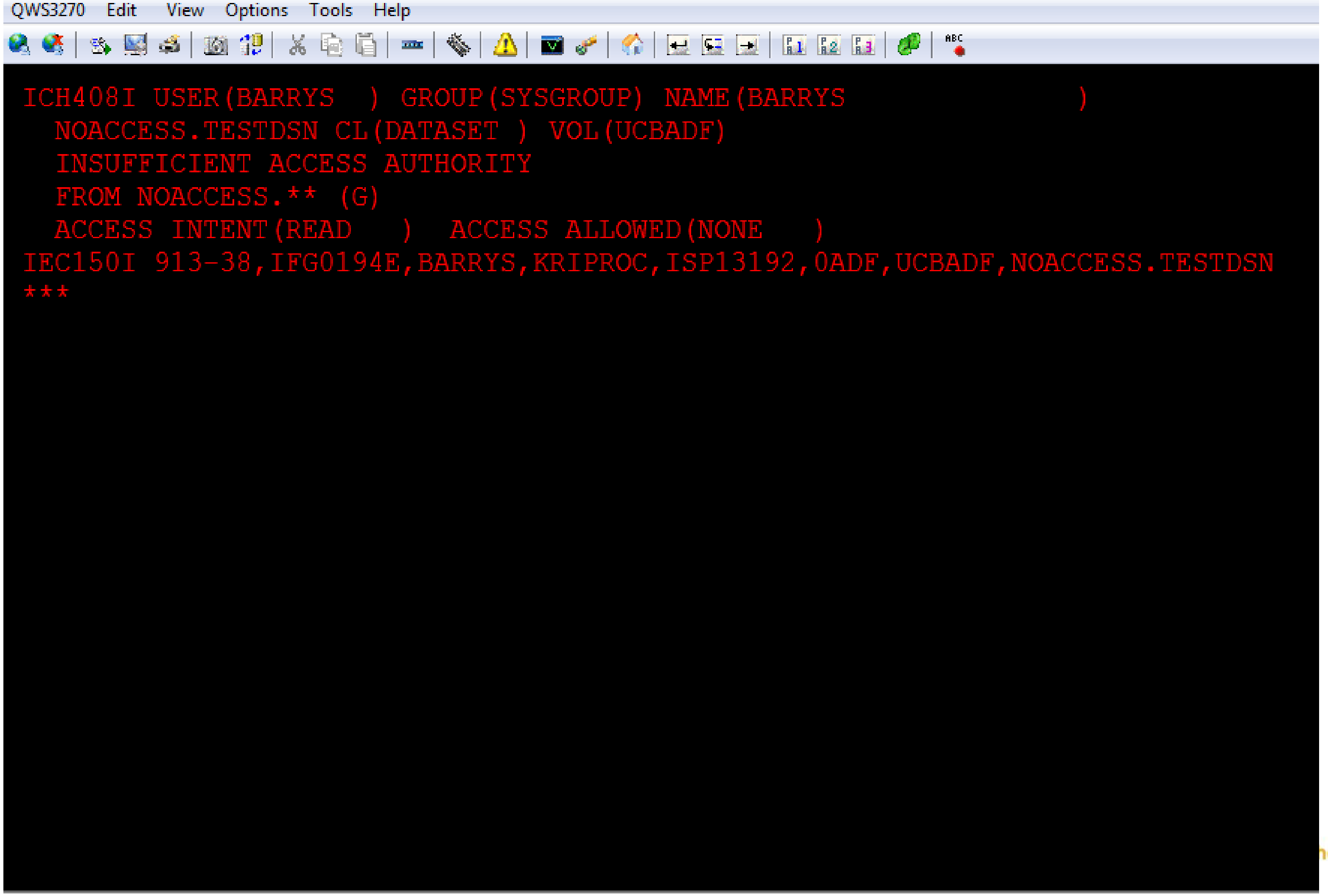
- TSO access is used (but not required)
- Need the ability to assemble and link a program OR could have file transferred a load module
- Need authority to create a load module library
  - Normal library – not APF authorized
- Need to be able to issue the TSO CALL command

Access a Dataset



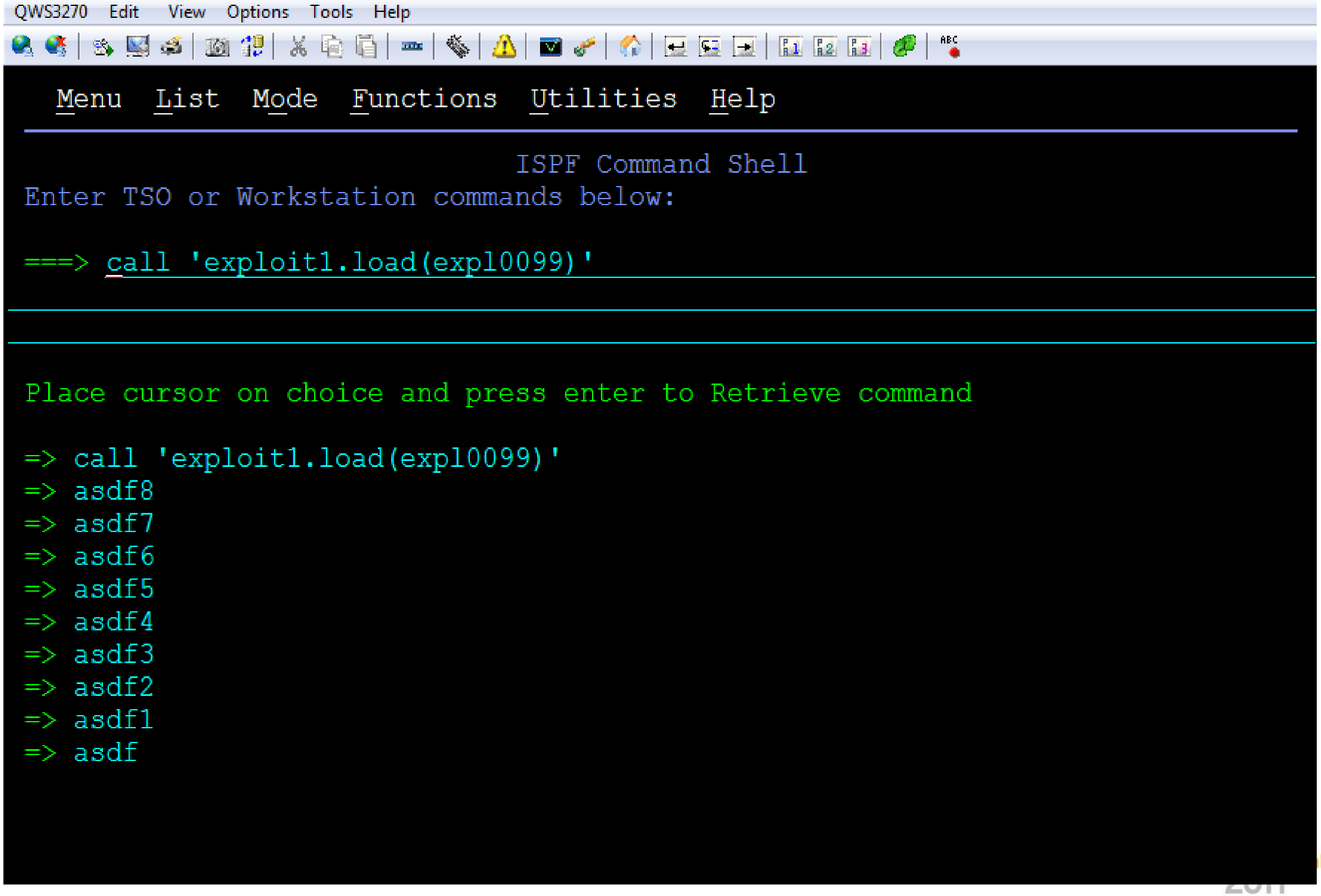
```
   Menu  RefList  RefMode  Utilities  Workstation  Help
 ------------------------------------------------------------------------------
                              Edit Entry Panel
 Command ===> ________________________________________________________________

 ISPF Library:
    Project . . . ________
    Group . . . . ________  . . . ________  . . . ________  . . . ________
    Type  . . . . ________
    Member  . . . ________          (Blank or pattern for member selection list)

 Other Partitioned, Sequential or VSAM Data Set, or z/OS UNIX file:
    Name . . . . . 'NOACCESS.TESTDSN'                                      +
    Volume Serial . . ______      (If not cataloged)

 Workstation File:
    File Name  . . ________________________________________________________
                                         Options
 Initial Macro  . . . . ________          _  Confirm Cancel/Move/Replace
 Profile Name . . . . . ________          _  Mixed Mode
 Format Name  . . . . . ________          _  Edit on Workstation
 Data Set Password  . .                   _  Preserve VB record length
 Record Length  . . . . _____             _  Edit ASCII data
```

Denied by RACF – 913 ABEND!!



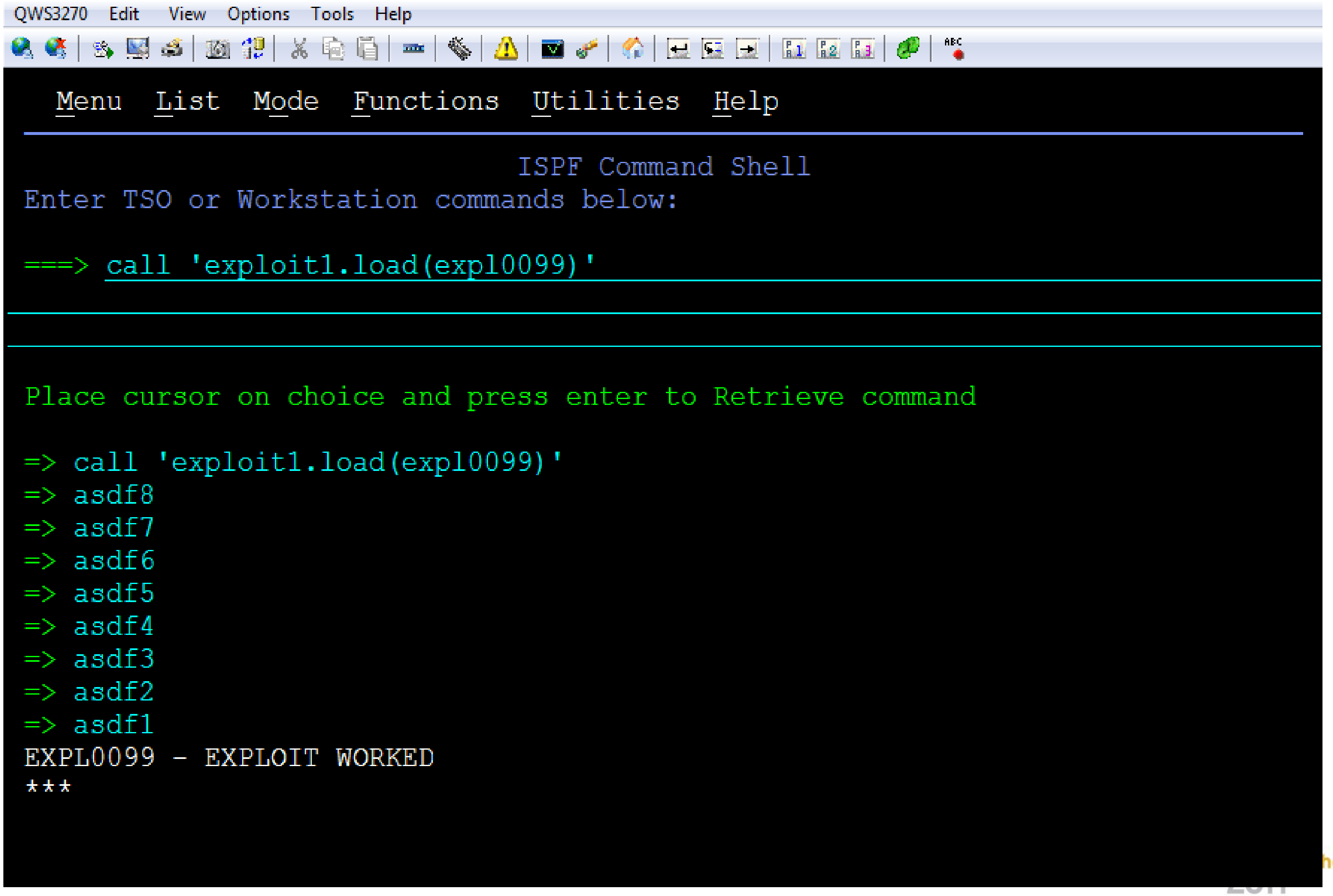
```
ICH408I USER(BARRYS  ) GROUP(SYSGROUP) NAME(BARRYS              )
  NOACCESS.TESTDSN CL(DATASET ) VOL(UCBADF)
  INSUFFICIENT ACCESS AUTHORITY
  FROM NOACCESS.** (G)
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-38,IFG0194E,BARRYS,KRIPROC,ISP13192,0ADF,UCBADF,NOACCESS.TESTDSN
***
```

Run an Exploit


QWS3270 Edit View Options Tools Help
Menu List Mode Functions Utilities Help

```
                              ISPF Command Shell
Enter TSO or Workstation commands below:

===> call 'exploit1.load(expl0099)'



Place cursor on choice and press enter to Retrieve command

=> call 'exploit1.load(expl0099)'
=> asdf8
=> asdf7
=> asdf6
=> asdf5
=> asdf4
=> asdf3
=> asdf2
=> asdf1
=> asdf
```

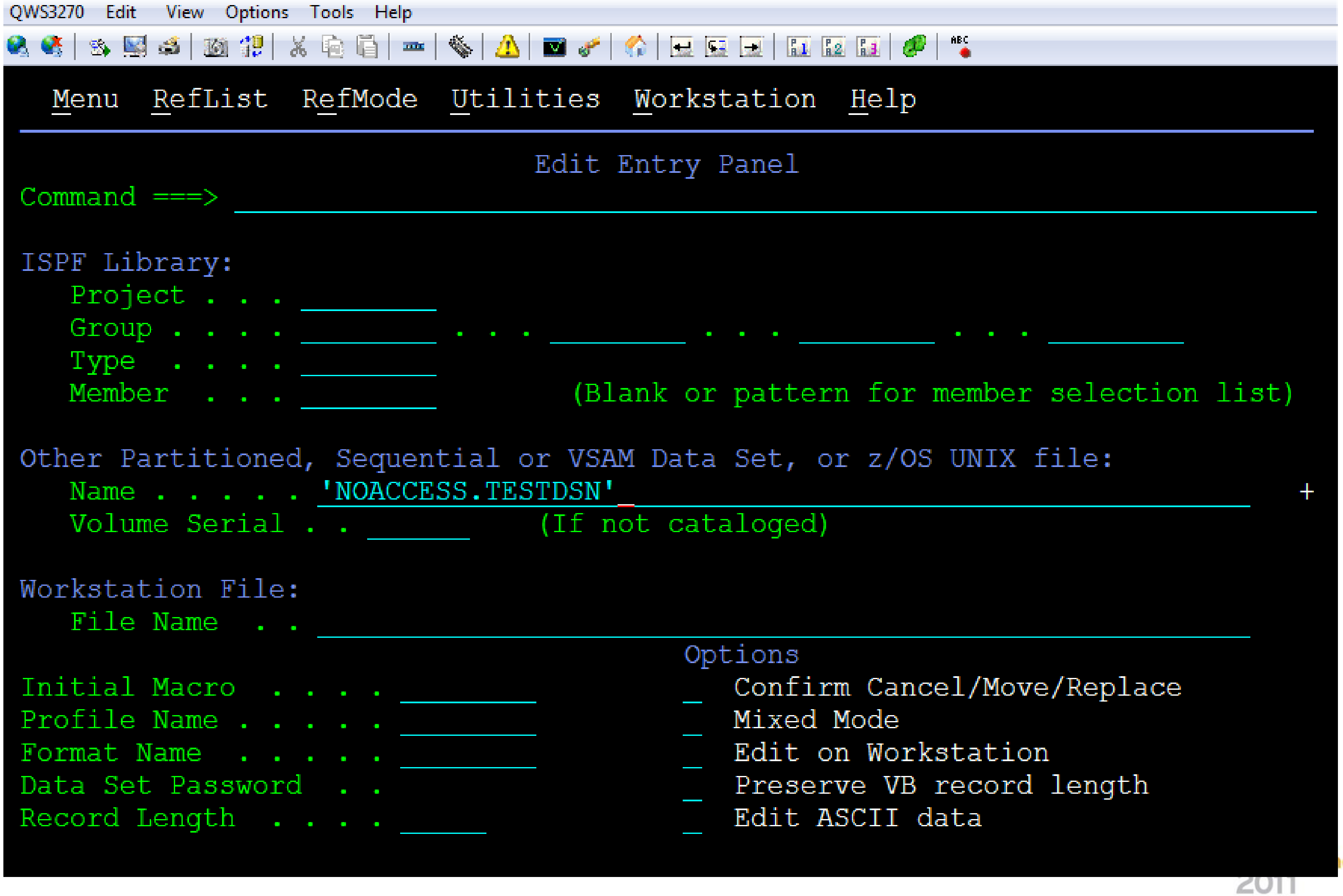
Now in RACF PRIVILEGED!!


QWS3270 Edit View Options Tools Help

Menu List Mode Functions Utilities Help

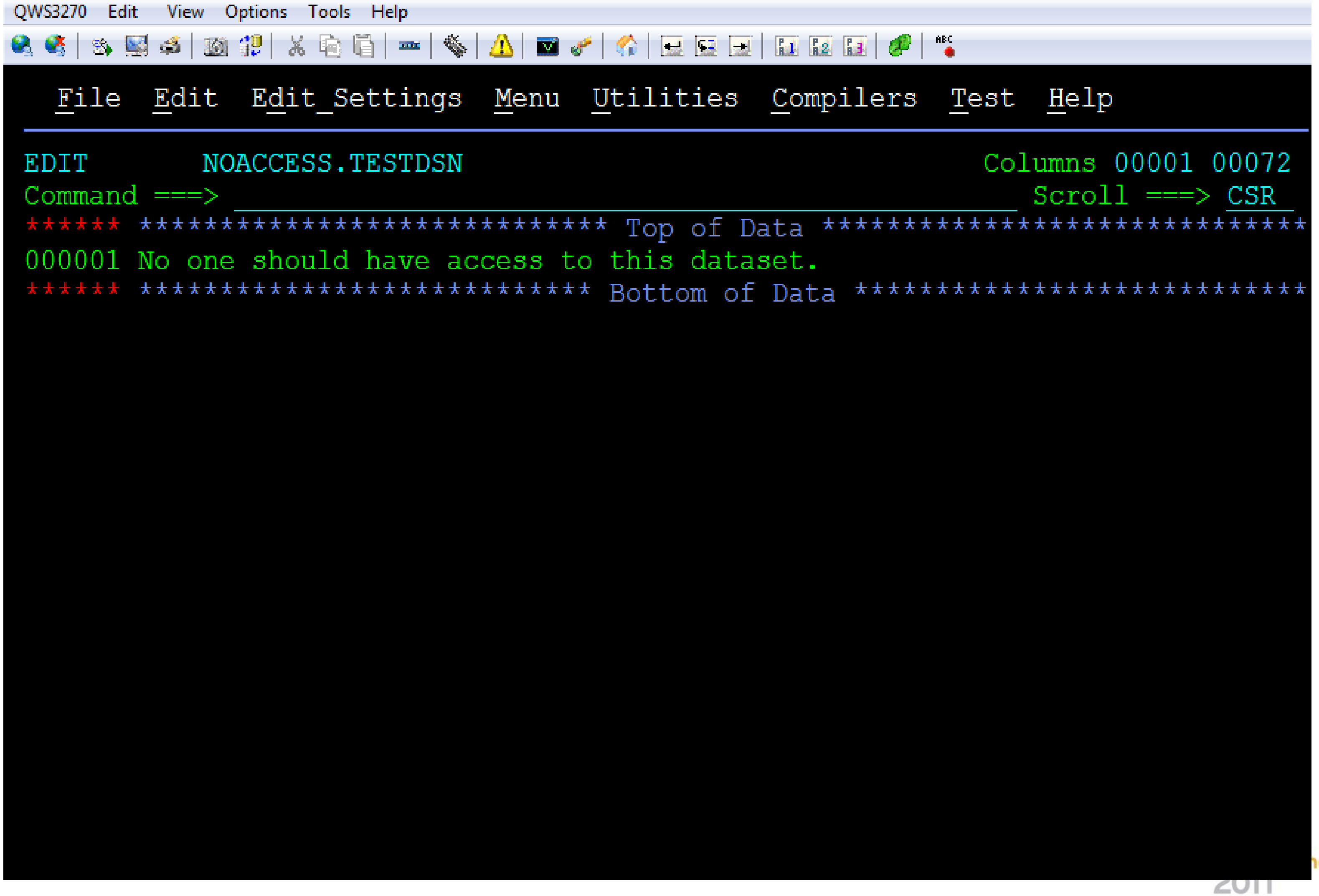
```
                        ISPF Command Shell
Enter TSO or Workstation commands below:

===> call 'exploit1.load(expl0099)'



Place cursor on choice and press enter to Retrieve command

=> call 'exploit1.load(expl0099)'
=> asdf8
=> asdf7
=> asdf6
=> asdf5
=> asdf4
=> asdf3
=> asdf2
=> asdf1
EXPL0099 - EXPLOIT WORKED
***
```

Access the Dataset Again



```
  Menu  RefList  RefMode  Utilities  Workstation  Help
 ------------------------------------------------------------------------------
                             Edit Entry Panel
 Command ===> ________________________________________________________________

 ISPF Library:
    Project . . . ________
    Group . . . . ________  . . . ________  . . . ________  . . . ________
    Type  . . . . ________
    Member  . . . ________          (Blank or pattern for member selection list)

 Other Partitioned, Sequential or VSAM Data Set, or z/OS UNIX file:
    Name . . . . . . 'NOACCESS.TESTDSN'_________________________________________ +
    Volume Serial . . ______      (If not cataloged)

 Workstation File:
    File Name  . . ________________________________________________________
                                        Options
 Initial Macro  . . . . ________        _  Confirm Cancel/Move/Replace
 Profile Name . . . . . ________        _  Mixed Mode
 Format Name  . . . . . ________        _  Edit on Workstation
 Data Set Password  . .                 _  Preserve VB record length
 Record Length  . . . . _____           _  Edit ASCII data
```

Now Have Access!!



```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
 -------------------------------------------------------------------------------
 EDIT       NOACCESS.TESTDSN                                Columns 00001 00072
 Command ===> ________________________________________________ Scroll ===> CSR
 ****** ***************************** Top of Data ******************************
 000001 No one should have access to this dataset.
 ****** **************************** Bottom of Data ****************************
```

What If The Dataset Contained

- Personal Credit or Financial Information?
- Personal Medical Information?
- A company’s internal strategy plans, design documents, etc.?
- Classified Government Information?
- Your Encryption Keys?

System Integrity Vulnerability Example

- Is similar to a poor security system controls issue
- Write a program to dynamically modify your security credentials
- A System Integrity Vulnerability does NOT require an APF authorized library – any load module library will do!
- You can remediate the poor Operating System or Security System controls issue
- But you cannot remediate the system integrity vulnerability

Does Your z/OS System Have These System Integrity Vulnerabilities?

- YES!!
- 67 of them have been reported to IBM
- All have APARs and PTFs now
- ISVs have them also
- Installation written code could have them as well
- More are likely to be found

But, you say:

- These attacks would not be from insiders
- Insiders are a trusted bunch of people
- My Insiders don’t have the technical expertise
- Well …

Example (now fixed by Vendor)

- KRI located a Potential Vulnerability
- The Exploit created to take advantage of the vulnerability was an **11 line REXX Program**
- Which gave them RACF Privileged authority! Total access to all datasets without any security logging records!
- It doesn't take a technical genius to enter an 11 line REXX Exec!

Exploits can be imported from the outside

- The exploit shown in the presentation
- If the load module was published on the internet
- It could be uploaded to z/OS
- And then executed
- How much technical expertise is required for this?
- How many people in your organization could do this?

USA Today – Jan 6, 2011

- White House asks agencies to review data security
- Office of Management and Budget (OMB) Memorandum
- *In an attempt to tighten control of classified information, the Obama administration has issued government-wide guidelines urging officials to be wary of "insider threats" and suggesting how supervisors can evaluate employee "trustworthiness."*

OMB Memorandum – Jan 3, 2011

- What steps has your agency taken to implement the latest version of the NIST SP-800 series guidance on Information Assurance, Risk Management, and Continuous Monitoring?
- NIST 800-53 – Control CA-2 Security Assessments
  - The organization includes, as part of a security control assessment, malicious user testing and penetration testing

2010 Verizon Data Breach Report

- 49% of data breaches were caused by insiders
- 48% were attributed to users who abused their privileges
- 79% of victims, subject to the PCI-DSS standard, had not achieved compliance prior to the breach
- PCI Standards include penetration testing and vulnerability scans!

How Do I Protect My System?

- Use the Compliance Regulations to your advantage
- They call for:
  - Penetration Testing
  - Vulnerability Scans
  - Malicious User Testing
- We always thought these were for networks and non-mainframe servers because mainframes were inherently secure
- But, although mainframes are more secure, they still have vulnerabilities

PCI Requirement 11.3 Guidance

Before applications, network devices, and systems are released into production, they should be hardened and secured using security best practices (per Requirement 2.2).

Vulnerability scans and penetration tests will expose any remaining vulnerabilities that could later be found and exploited by an attacker.

NIST 800-53 – Control CA-2
Security Assessments

- The organization includes, as part of a security control assessment, malicious user testing and penetration testing

ISO/IEC 27001

15.2.2 Technical compliance checking

Information systems should be regularly checked for compliance with security implementation

Standards

Compliance checking also covers, for example, penetration testing and vulnerability assessments, which might be carried out by independent experts specifically contracted for this purpose

Securing a z/OS Mainframe

- Secure Mainframes are assumed because of CA-ACF2, IBM-RACF and CA-Top-Secret
- It is assumed that the mainframe configuration parameters and Security System are properly configured
- The MOST complete guide to securing a mainframe is the DISA STIG

  http://iase.disa.mil/stigs/checklist/
- DISA STIGs do not test for system integrity vulnerabilities
- Integrity exposures are a serious compromise to your security controls

Two Completely Different Issues:

- System Configuration and Security System Controls Exposures
  - Addressed by the Defense Information Systems Agency (DISA) Security Technical Implementation Guidelines (STIGs)
- System Integrity Issues
  - Addressed by the Vulnerability Scans and Penetration Tests

And you need to address both of them!

System Integrity Vulnerability Scans and Penetration Testing

- Helps safeguard your organization
- Prevents financial loss through fraud
- Stops hackers, spies and disgruntled employees
- Provides Due Diligence
- Provides compliance to industry standards
  - PCI Requirement 11.3
  - Government Standards – NIST 800.53
  - International Standards – ISO 27001

Thank you!

Ray Overby

Key Resources, Inc.

Ray.Overby@kr-inc.com

www.vatsecurity.com

(312) KRI-0007