17:08 Murder on the USS Table

by Soldier of Fortran
concerning an adventure with Bigendian Smalls

The following is a dramatization of how I learned to write assembler, deal with mainframe forums, and make kick-ass VTAM USS Tables. Names have been fabricated, and I won't let the truth get in the way of a good story, but the information is real.

It was about eleven o'clock in the evening, early summer, with the new moon leaving an inky darkness on the streets. The kids were in bed dreaming of sweet things while I was nursing a cheap bourbon at the kitchen table. Dressed in an old t-shirt reminding me of better days, and cheap polyester pants, I was getting ready to call it a night when I saw trouble. Trouble has a name, Bigendian Smalls. A tall, blonde, drink of water who knows more about mainframe hacking than anyone else on the planet, with a penchant for cargo shorts. I could never say no to cargo shorts.

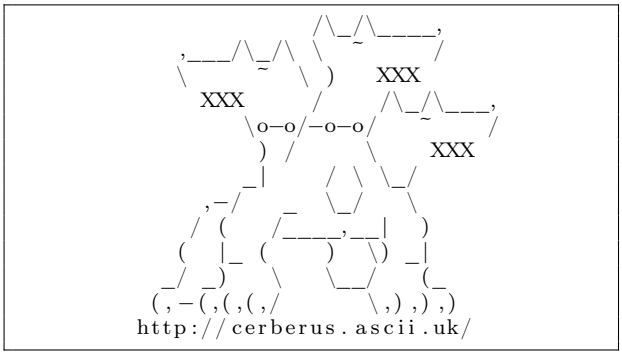
The notification pinged my phone before it made it to Chrome. I knew, right then and there I wasn't calling it a night. Biggie needed something, and he needed it sooner rather than later. One thing you should know about me, I'm no sucker, but when a friend is in need I jump at the chance to lend a hand.

Before opening the message, I poured myself another glass. The sound of the cheap, room temperature bourbon cracking the ice broke the silence in my small kitchen, like an e-sport pro cracking her knuckles before a match. I opened the message:

"Hey, I need your help. Can you make a mainframe logon screen for Kerberos? But can you add that stupid Windows 10 upgrade popup when someone hits enter?"

"Yeah," I replied. I'm not known for much. I don't have money. I'm as cheap as a Garfield joke in the Sunday papers. But I can do one thing well: Mainframe EBCDIC Art.

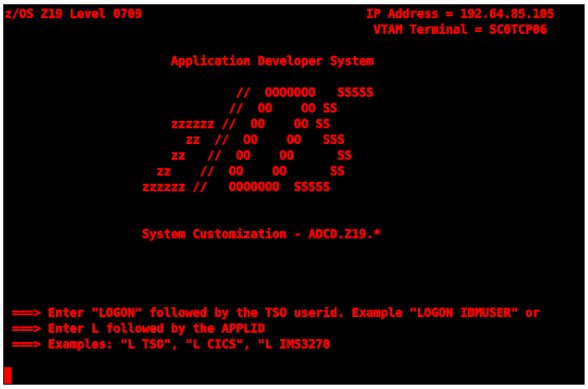
I knew It was going to be a play on Cerberus, the three-headed dog. Finding that ASCII was the easy part. ASCII art has been around since the creation of the keyboard. People need to make art, regardless of the tool. Finding ASCII art was going to be simple. Google, DuckDuckGo, or in desperate times and lots of good scotch, Bing, will supply the base that I need to create my master piece. The first response for a search for "Cerberus" and "ASCII" yielded my three-headed muse.



The rest, however would require a friend's previous work, as well as a deep understanding of the TN3270 protocol and mainframe assembler.

– — — – — — — — – — —

When I got in to this game six years ago it was because I was tired of looking at the red "Z."



That red was rough, as though accessing this mainframe was going to lead me right to Satan himself. (Little did I know I'd actually be begging to get by Cerberus.)

The world of mainframes, it's a different world. A seedier world. One not well-travelled by the young, and often frequented by the harsh winds of corporate rule. Nothing on the mainframe comes easy or free. If you want to make art, you'll need more than just a keyboard.

I started innocently enough, naively searching simple terms like "change mainframe logon screen." I stumbled around search results, and into chatrooms like a newborn giraffe learning to walk. You know the type, a conversation where everyone is trying to

prove who's the smartest in the room. While ultimately useless, those initial searches taught me three things: I needed to understand the TN3270 protocol, z/OS High Level Assembler (HLASM), and what the hell a VTAM and the USS Table were.

– — — – — — — – — — –

I always knew I would have to learn TN3270. It's the core of mainframe–user interaction. That green screen you see in movies when they say someone "just hacked a mainframe." I just never thought it would be to make art for my friends. TN3270 is based on Telnet. Or put another way, Telnet is to TN3270 as a bike is to an expensive motorcycle. They sort of start out the same but after you make the wheels and frame they're about as different as every two-bit shoe shine.

Looking at the way mainframes and their clients talk to one another is easy enough to understand, at first. Take a look at Figure 18.

For anyone who understood telnet like I did, this handshake was easy enough to understand.

```
IAC: Telnet Command
DO/WILL: Do this! I will!
SB: sub command
```

But that's where it ended. Once the client was done negotiating the telnet options, the rest of the data looked garbled if you weren't trained to spot it.

You see, mainframes came from looms. Looms spoke in punchcards which eventually moved to computers speaking EBCDIC. So, mainframes kept the language alive, like a small Quebec town trying to keep French alive. That TN3270 data was now going to be driven by an exclusively EBCDIC character set. All the rest of the options negotiated, and commands sent, would be in this strange, ancient language. Lucky for me, my friend Tommy knows all about TN3270 and EBCDIC.[25] And Tommy owed me a favor.

– — — – — — — – — — –

Just past a Chinese restaurant's dumpster was the entrance to Tommy's place. You'd never know it even existed unless you went down the alleyway to relieve yourself. As I approached the dark green door, I couldn't help but notice the pungent smell of decaying cabbage and dreams, steam billowing out of a vent smelled vaguely of pork dumplings. I knocked three times. The door opened suddenly and I was ushered in. I felt Tommy slam the door shut and heard no fewer than three cheap chain-locks set in to place.

Tommy's place was stark white, like a website from the early 90s. No art, no flashing neon, just plain white with some printouts stuck on the white walls and the quiet hum of an unseen computer. The kind of place that makes you want to slowly wander around an Ikea. Tommy liked to keep things clean and simple and this place reflected that.

Tommy, in his white lab coat, was a just a regular man. As regular and boring as a vodka with lime and soda, if vodka, with lime and soda, wore large rimmed glasses. But he knew his way around TN3270, and that's what I needed right now.

"So, I hear you need some help with TN3270?" Tommy asked. He already knew why I was there.

"Yeah, I can't figure this garbage out and I need help writing my own," I replied.

Tommy sighed and began explaining what I needed to know. He walked over to one of three whiteboards in the room.

"The key thing you need to know is that after you negotiate TN3270 there are seven control characters. But if all you want to do it make art, you only need to know these four:

```
SF  – "\x1D" – aka Start Field
SBA – "\x11" – aka Set Buffer Attribute
IC  – "\x13" – aka Insert Cursor
SFE – "\x29" – aka Start Field Extended
```

"Unlike telnet, TN3270 is a basically 1920 character string, for the original 24×80 size. The terminal knows you're starting 'cuz the first byte you send is a command (i.e. `\x05`) followed by a Write Control Character (WCC). For you, sir artist, you'll want to send 'Erase/Write/Alternate.' or `\xF5\x7A`. This gives you a blank canvas to work with by clearing the screen and resetting the terminal.

"The remaining makeup of the screen is up to you. You use SBA to tell the terminal where you want your cursor to be, then use the 'Start Field'/'Start Field Extended' commands to tell the terminal what kind of field it is going to be, also known as an attribute. Start field is used to lock and unlock the screen, but for your art it doesn't matter.

"One thing you'll need to watch out for, anytime you use SF/SFE, is that it takes up one byte on the

[25] `http://www.tommysprinkle.com/mvs/P3270/ctlchars.htm`

```
TN3270(KINGPIN,23): << IAC DO TN3270
TN3270(KINGPIN,23): >> IAC WILL TN3270
TN3270(KINGPIN,23): Entering TN3270 Mode:
TN3270(KINGPIN,23):     Creating Empty IBM-3278-2 Buffer
TN3270(KINGPIN,23):     Created buffers of length: 1920
TN3270(KINGPIN,23): Current State: 'TN3270E mode'
TN3270(KINGPIN,23): << IAC SB TN3270 TN3270E_SEND TN3270E_DEVICE_TYPE SE
TN3270(KINGPIN,23): >> IAC SB TN3270 TN3270E_DEVICE_TYPE TN3270E_REQUEST IBM-3278-2-E IAC SE
TN3270(KINGPIN,23): << IAC SB TN3270 TN3270E_DEVICE_TYPE TN3270E_IS I B M - 3 2 7 8 - 2 - E
                    TN3270E_CONNECT S M O G L U 0 2 SE
TN3270(KINGPIN,23): Confirmed Terminal Type: IBM-3278-2-E
TN3270(KINGPIN,23): LU Name: SMOGLU02
TN3270(KINGPIN,23): >> IAC SB TN3270 TN3270E_FUNCTIONS TN3270E_REQUEST IAC SE
TN3270(KINGPIN,23): << IAC SB TN3270 TN3270E_FUNCTIONS TN3270E_IS SE
TN3270(KINGPIN,23): >> IAC SB TN3270 TN3270E_FUNCTIONS TN3270E_REQUEST IAC SE
TN3270(KINGPIN,23): Processing TN3270 Data
```

Figure 18. TN3270 Packet Trace

screen. Setting the buffer location does not. Once you're done with your art, you'll need to place the cursor somewhere, using IC."

Starting to understand, I headed to the white board and wrote Figure 19 in black marker.

"Yes! That's it!" exclaimed Tommy. "With what you have now, you could make a monochrome masterpiece! Keep in mind that the SF eats up one space. So basically you could fill out the rest of the screen's 1,919 characters, remembering that the line wraps at every 80 characters. But let's talk about SF and SFE."

"In your, frankly simple, example," Tommy continued, "you'd never get any color. To do that, we need to talk about the Start Field Extended (`\x29`) command. That command is made up of the SFE byte itself, followed by a byte for the number of attributes, and then the attributes themselves.

"There's two attributes we care about: SF (`\xC0`), and the most important one, which I'll get to in a minute. SF is what we use like above to control the screen. If we wanted to protect the screen from being edited we could set it to `\xF8`.

"Now, you'll want to listen closely because this attribute is arguably the most important to you. The color attribute (`\x42`) lets you set a color. Your choices are `\xF1` through `\xF7`."

```
F1 Blue
F2 Red
F3 Pink
F4 Green
F5 Turquoise
F6 Yellow
F7 White
```

```
\x05 WCC SBA 0 0 SF 0 Here Lies Trouble IC
\x05 \x7A \x11 \x00 \x00 \x1D \x00 Here Lies Trouble \x13
```

Figure 19. Placing the cursor after drawing.

```
1 \x05  WCC SBA    0     0    SF    0  Here Lies Trouble  SFE  1 COLOR WHITE Double  IC
  \x05 \x7A \x11 \x00 \x00 \x1D \x00 Here Lies Trouble \x29 \x01 \x42  \xF7  Double \x13
```

Tommy grabs the black marker from my hand and begins adding to my simple example.

"So, with a bit of this code, we can add a color statement to your commands. Remember to move the cursor to the end though.

"There's one last thing you should know, but it's a little advanced. You can set the location using SBA followed by a row/column value. Right now, you've set the buffer to 0/0. But using this special table," Tommy pointed to a printout he had laminated and stuck to his wall,[26] "we can point the buffer anywhere we—"

Just then the door burst open, the sounds of those cheap locks breaking and hitting the floor echoed through the room. A dark figure stood in the doorway holding some type of automatic gun, which I couldn't place. Tommy quickly took cover behind a desk and I followed suit. I heard a voice yell out "How dare you teach him the way! He might not have the access he needs! Did you ask if he's allowed to make the kind of changes you're teaching? He should've spoken to his system programmer and read the manuals!"

Tommy, visibly shaken, shouted, "Rico! I'm sorry! I owed someone a favor and…"

Rico opened fire. Little pieces of shattered whiteboard hitting me in the face. He wasn't aiming for us, but had destroyed our notes on the white board. I looked over and saw Tommy cowering under his desk, I had figured 'Tommy' was a nickname for a favorite firearm, guess I was wrong.

"You've given out free TN3270 help for the last time Tommy!" Rico shouts, and I heard the familiar sound of a gun being reloaded. I took a quick peek from my hiding place and noticed that Rico hadn't even bothered to take cover, still standing in the doorway. Not wanting my epitaph to read, "Here lies a coward who died learning TN3270 behind a Chinese restaurant," I pulled out my Colt detective special and opened fire. My aim had always been atrocious, but I fired blindly in the direction of the door, heard a yelp, and then silence.

Tommy popped his head above the desk, "He's gone, looks like he ran off, you better get out of here in case he and his goons return."

I took this as my cue and headed towards the door. I noticed part of the frame had splintered, and in the center of those splinters was my slug. looks like I just missed Rico.

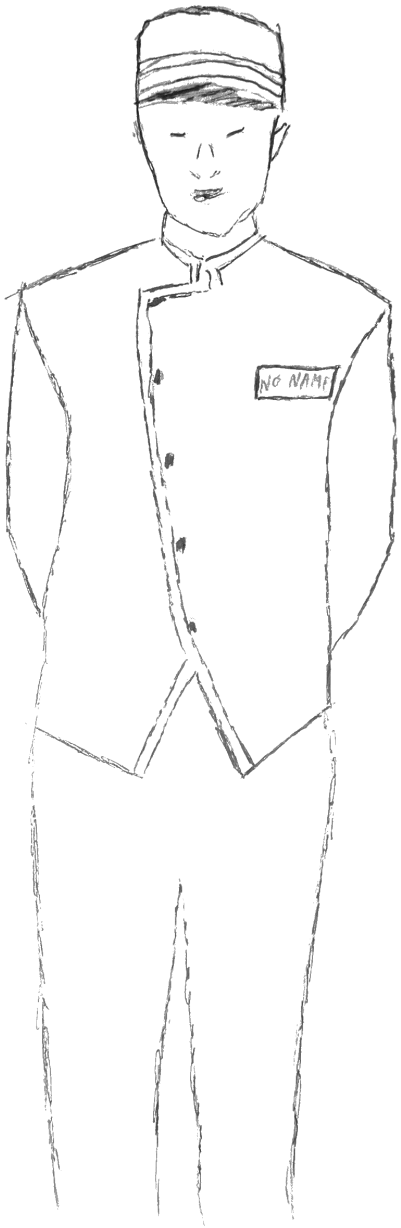
Tommy grabbed my arm as I'm about to leave, "You still need to learn some assembler and VTAM, go talk to Dave at The Empress, he can help you out. But never come back here again, you're too much trouble."

– — — – — — — — – — –

The Empress. On the books it was a hotel. Off the books it's where you went when you wanted help forgetting about the outside world. The lobby looked and smelled like a cheap computer case that hadn't been cleaned out for years. Half the lights in the chandelier didn't work, and it cast odd shadows on the furniture, giving the impression someone was there, watching you. It was the kind of place European tourists booked because Travelocity got them a great deal, but the price would immediately change once they arrived. No one came to the Empress for its good looks. Not-quite-top-40 music emanated from the barroom.

I walked to the front desk, where a young man with a name tag that said "No Name" looked me up and down. "Can I help you?" Millennial sarcasm dripped off of every syllable. "I need to speak to Dave," I replied. The clerk's eyes widened a little, he quickly looked around and whispered "follow me."

[26] http://www.tommysprinkle.com/mvs/P3270/bufaddr.htm



The clerk walked me past the kitchen, through the back hallways, in to the laundry room. He ushered me in, then abruptly left. A sole person was folding linens in front of an industrial washing machine, a freshly lit cigarette hung loosely from his lips. The fluorescent light turned his skin a pale shade of blue. "Dave?" I called out.[27] Dave put the bed sheet down and walked over. 'Who wants to know?" he asked.

"Tommy sent me," I replied.

Dave takes a long pull on his coffin nail, "Shit," he says exhaling a large puff, "you tell Tommy that we're square after this. I assume you're here to learn HLASM? Can I ask why?"

"I'm trying to make some my mainframe look beter." I replied.

Dave wasn't a tall man, but his stature, deep voice, and frame more than made up for it. The type of man you could trust to knock you out in one punch. His white hotel uniform was stained with what I hoped wasn't blood.

He sighed and said "this way."

Dave led me to a small room off the laundry area with some books on the wall, lit by a single, bare bulb in the ceiling fixture. A black chalkboard stood in one corner, an old terminal on a standing desk, all the rage these days, at in the other. The walls were bare concrete. "I assume you already know JCL?" queried Dave.

"Yes" I replied with a failed attempt at sarcasm, "of course I know JCL."[28]

"Good, this will be easy then." He took another pull of his smoke and began writing on the blackboard, "There're four executables available to you to compile an HLASM program on the mainframe. They are:

```
ASMAC    - Assembles only
ASMACL   - Assembles and link edits
ASMACLG - Assembles, links and runs
ASMACG   - Assembles, uses a loader to run
```

Dave walked over to the terminal and pulled up a file on the screen. "You need to pass it some options, like this," he said, pointing to a line on the screen:

```
//BUILD    EXEC ASMACL
//C.SYSLIB   DD DSN=SYS1.SISTMAC1,DISP=SHR
//             DD DSN=SYS1.MACLIB,DISP=SHR
//C.SYSIN    DD *
```

"Anything you type on the next line, after the * must be in HLASM and will be compiled by ASMACL. Don't worry about finding it, ASMACL is given to us by Big Blue." Dave's calloused fingers flew over the keyboard and a moment later I was staring at a blank file with the JCL job card and

[27] http://csc.columbusstate.edu/woolbright/WOOLBRIG.htm

[28] PoC||GTFO 12:6, a JCL Adventure with Network Job Entries

compiler stuff filled out. "First, there're some rules with HLASM you should know. Each line can either be an instruction, continuation, or comment. Comments start with '*'. A Continuation line means that in the previous line there's a character (any character, doesn't matter which) in column 72, and the continued line itself must start on column 16."

"You with me so far?"

I nodded.

"Good. Now, If it's not a comment or a continuation, the line can be broken down like so:

"The first 10 characters can be empty or be a name/label. Following that you have your instruction, a space, then your operands for that instruction. Anything after the operands is a comment until the 71st column. Here's a dirty example." (Figure 20.)

"Every line can have a name. In HLASM you can create basic variables with an & in front of them. But not every line needs a name. Take a look at these three lines:

```
&BLUE    SETC 'X''290142F1'
         DC &BLUE Make it blue!
         DC C'Big Blue' Simple text
```

"Line one sets a symbol/label to &BLUE. If Tommy did his job right you should be able to recognize what it is supposed to do. The next line is DC, Declare Constant. Notice &BLUE has an X. That means it's in hex. When we want to send text, we can use 'C' for `CHAR`. If we wanted we could've written the above like this." I watched as his fingers danced across the keyboard.

```
         DC X'290142F1'
         DC C'Big Blue'
```

"But you'll likely be switching colors, so setting them all to variables makes your life easier. One caveat with using variables in HLASM: The assembler will replace any value you have with the variable, take a look at this:

```
&KINGPIN   SETC 'BOSS'
&BOSSBEGN  SETC 'B'.'&KINGPIN'
&BOSSEND   SETC 'E'.'&KINGPIN'
&BOSSBEGN  EQU *
* SOME CODE
&BOSSEND   EQU *
```

"Lets break this down so you can see what the compiler would do:

```
&KINGPIN   = 'BOSS'
&BOSSBEGN  = BBOSS
&BOSSEND   = EBOSS

BBOSS      EQU *
* SOME CODE
EBOSS      EQU *
```

"This understanding will come in handy when you're making a USS Table." I still didn't know what a USS Table was, but I let him go on. "If you have stuff you're going to do over and over again, it would be easier to make a function, or in HLASM a macro, to handle the various request types. Macros are easy. On a single line you declare '`MACRO`' in column 10. The next line you give the macro a name, and it's operands. You end a macro with the word '`MEND`' in column 10 on a single line. For example:"

```
         MACRO
&NAME    SCREEN &MSG=.,&TEXT=.
         DC &MSG
         DC &TEXT
         MEND
*
         SCREEN MSG=03,TEXT='Big Blue'
```

I thought I was starting to get it, so I decided to ask a question. "How would we do an IF statement?" I asked.

```
1---------10---------20---------30---------40---------50---------60---------70---------80
SYMBOL   DC X'DEADBEEF' A comment
* Another comment
         DC C'Hello World' I'm a single line
         DC C'HELLO                                                              X
               WORLD' I'm a continuation
```

Figure 20. Dave's Example

Dave smiles, but only a little, and walks back over to the blackboard and scribbles out the following:

```
&MSG      SETC C'04'
AIF ('&MSG' NE '02').SKIP
          DC C'Not Equal to 2'
.SKIP     ANOP
          DC C'End of Line'
```

"In HLASM you can use the AIF instruction. It's kind of like an IF. Here we have some code that will print 'Not Equal to 2' and 'End of Line.' If we set &MSG to '02' it would jump ahead to `.SKIP`, what Big Blue would call a label.

"I see you staring at that ANOP. I know what you're thinking, and the answer is yes. It's exactly like a NOP in x86. Except it's not an opcode, but a HLASM assembler instruction."

Dave headed back to the terminal and quickly scrolled to the bottom. "There's one last thing, since we're using ASMACL you need to tell the compiler where to put the compiled files. Take a look at this."

```
//L.SYSLMOD DD DISP=SHR,DSN=USER.VTAMLIB
//L.SYSIN   DD *
  NAME USSCORP(R)
```

Dave tapped on the glowing screen. "This line right here. This tells the compiler to make a file USSCORP in the folder USER.VTAMLIB." I knew he meant Member and Partitioned Dataset but I figured Dave was dumbing things down for me and didn't want to interrupt. "That's where your new USS Table goes," he continued.

I jumped as someone softly knocked on the door, guess I was still a little jumpy from my encounter at Tommy's. I saw through the round window in the door that the clerk had returned. Dave headed over and opened the door. I couldn't quite make out what they were saying to each other. Dave looked at his watch and turned to me, "Look, this has been swell, but you gotta get outta here. If my boss finds out I taught you this there'll be hell to pay and I'm not looking to sleep with the fishes tonight—or any night. Sorry we're cutting this short, normally I'd be teaching you about the 16 registers and program entrance and exit, but we don't have time for that. And besides, you don't need it to be a VTAM artist, but if you want to learn, read this." And he shoved a rather large slide deck in to my chest, at least 400 pages thick.[29]

No Name told me to follow him yet again. As we left the laundry room I saw Dave stuffing soiled linens in to one of those washers; this time there's no wondering if it was blood or not. No Name ushered me down a different hallway than the one we came in. He walked quickly, with purpose. I struggle to keep up.

We ended up at a door labeled 'Emergency Exit.' No Name opened the door and I headed through. Before I could turn around to say thanks, the big metal door slammed closed. I found myself in another dead-end alleyway. The air was cool now, the wind moist, betraying a rain fall that was yet to start.

I began heading towards the road when a shadowy figure stepped into the alley. I couldn't make out what he looked like, the neon signs behind him made a perfect silhouette. But I could already tell by his stance I was in trouble.

"So," the figure called out, "the boss tells me you're trying to change the USS Table eh?" I figured this must be one of Rico's goons.

"I don't mean nothing by it," I replied, "I'm just trying to make my mainframe nicer."

"Rico has a message for you 'if you're trying to change the mainframe you should be talking to the people who run your mainframe, I've had enough of this business.' "

The gunshot echoed through the alleyway, the round hitting me square in the chest like a gamer punching his monitor in a rage quit. I landed on flat my back, smacked my head on the cold concrete, and sent pages of assembler lessons flying through the air. The wind knocked out of me, I felt the blackness take hold as I lay on the sidewalk. I could barely make out the figure standing over me, whispering "when you get to the pearly gates, tell 'em the EF Boys sent ya."

[29] `unzip pocorgtfo17.pdf Asm-1.PPTx`

You know those dreams you have. The kind where you're in a water park, floating along a lazy river, or down a waterslide. I was having one of those. It was nice. Until I realized why I was dreaming of getting wet. I woke face up, in an alleyway, the rain pounding me mercilessly. My trench coat was drenched by the downpour. I stood up, slowly, still dizzy from getting knocked out.

How had I survived? I looked around and saw papers strewn about the alley. Something shiny, just next to where I took my forced nap, caught my eye. It was a neat pile of papers, held together by a dimple on the top sheet. I took a closer look and picked up the pages.

Well I'll be damned, the 400+ pages of assembler material took the bullet for me. Almost square in the middle was the bullet meant to end my journey. I eternally grateful that Dave had given me those pages. Now, determined more than ever to finish what I started, I headed towards the street. I had two of the three pieces to the puzzle, but I needed dry clothes and my office was closer than going home.

– — — – — — — – — – –

Nestled above a tech start-up on its last legs was a door that read 'Soldier of FORTRAN: Mainframe Hacker Extraordinaire.' Inside was a desk, a chair, an LCD monitor and a PC older than the startup. A window, a quarter of the Venetian blinds torn free, looked out over the street. I didn't bother turning on the lights. The orange light that bled in from the lamppost on the street was enough. I pulled out my phone, put it on the desk, and started changing in to my dry clothes. The clothes were for when I hoped I would start biking to work which, as with all new year's resolutions, were yesterday's dream.

Now dry, I decided to power on my PC and take some notes. I wrote down what I knew about TN3270 thanks to Tommy and HLASM courtesy of Dave. I was still missing a big piece. Where could I learn about this USS Table. My searches all led to the same place: The Mailing-List. A terrible bar on the other side of town I had no desire to visit. The Mailing-List, or 'Dash L' as some people called it, was filled with some of the meanest, least helpful individuals on this Big Blue planet. I was likely to get chased out of the place before I was even done asking my question, let alone receiving an answer.

Don't get me wrong, sometimes Dash L had some great conversations, I know because I often lurk there for information I can use. But I had never worked up the courage to ask a question there, lest I be banned for life. But, with nothing else to go on I grabbed my coat and umbrella and headed for the door.

Just then, my phone rang. I didn't recognize the name-Nigel, or the number. I decided to answer the phone. "Who's this, how'd you get my private number?" No reply. I went to hang up the phone when I heard, "try searching for USSTAB and MSG10." My phone vibrated, letting me know the call was over. I ran to the window and peered out in to the rainy night. The street was empty except for a man with an umbrella putting his phone away. I ran down the stairs and caught a glimpse of the man as he got into his Tesla and sped off.

Back at my desk, I searched for USSTAB and MSG10 and one name kept coming back: Big John. I knew Big John, of course. Anyone who did mainframe hacking knew him. He now played the ivories over at a fancy new club, the Duchess. My dusty work clothes would have to be fancy enough.

– — — – — — — – — – –

You wouldn't know the Duchess was much, just by looking at it. A single purple bulb above a bright red vinyl entrance. The lamp shade cast a triangle of light over the door. The only giveaway that this was a happening place was the sound of 80s Synth rolling down the streets. Not the cheap elevator synth you get while waiting for your coffee, this was real synth: soulful and painful. The kind that made you doubt yourself and your life choices.

I walked to the door and knocked. A slit opened up, "Can we help you?" a woman's voice asked. I couldn't wait for this new speakeasy revival trend to die. "Yes," I replied, "I'm here to see Big John."

"You have a reservation?" she asked.

"Nope, just here to see Big John."

"Honey, you outta luck. We got a whole room of people here to see Big John, and they got reservations!"

"How much sweetener to see him play tonight?" I ask.

A second slot near my dad gut opened up, and a drawer popped out, almost like the door was happy to see me. I placed the only fifty I had in the tray. The drawer and slit closed and the door opened.

A young woman took my coat and brought me to a table. I took my seat and casually looked around. The room was dimly lit, with most of the light coming from the stage. Smoke hung in the air like a summer haze waiting for a good thunderstorm. A

waitress asked, “Drink sir?” I ordered a dirty martini and enjoyed the rest of the show. It’d been a shit day, I needed a break.

Once the show was done and the band started to pack up, I walked up to Big John. “Apparently you’re a man who can help me with USSTAB and some TN3270 animations.” I say. He finished putting away his keytar in its carrying case. “I could be, what’s in it for me?” My wallet was empty so I figured a play on his emotional side might work, “You’d get a chance to piss off Rico and the EF Gang.”

Big John looked at me and smiled. “Anything to piss of that hothead, follow me.” I grabbed my coat from the front and followed him.

Big John was the type of guy who lived up to the name. He was massive. Use to play professional football before he got injured and went back to his original loves: hacking and piano. Long dark hair and an even longer and darker beard made him look menacing. But if you ever knew Big John, you’d know he was just a big ‘ol softy.

John led me to another alleyway behind the Duchess. What was it with this city and alleyways? It looked like the rain had let up, but it had left a cold, damp feeling in the air. Parked in the alley was a van, with a wizard riding a corvette painted on the side. Big John opened the back, set his keytar down and motioned for me to get in the van.

Inside was a nicer office space than I have. Expensive, custom mechanical keyboards lined one wall. Large 4k monitors hung on moveable arms. An Aeron chair was bolted to the floor. Somewhere, invisible to me, was a computer powerful enough to drive this setup.

“So, I take it you’ve been to both Tommy and Dave already?” he asked over the clicking of his mechanical keyboard as he logged on.

“Yes,” I reply. “I think I understand enough to get started making my own logon screens. I can control the flow and color of a TN3270 session, and I know how to use HLASM to do so. But Dave kept referring to things like MSGs and a USS Table which makes no sense to me.”

Big John chuckled and sat down, lighting what looked like a hand-rolled cigarette but smelled like a skunk. “Don’t worry about Dave,” he said, taking a few puffs, “he’s an ex-EF Boy, he’s still trying to get use to sharing information that people can understand. Sometimes he’s still a little cryptic. Let’s get started.”

“When you connect to a mainframe, nine times outta ten its going to be VTAM,” Big John explains.

“VTAM is like the first screen of an infocom game. It lets you know where you are, but from there it’s up to you where you go, you get me?” he asks between puffs.

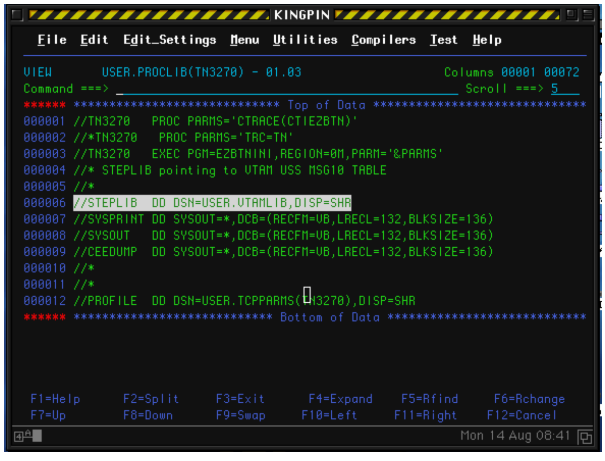
I did, and I didn’t. All I wanted to do was make pretty mainframes.

“First thing you gotta know about VTAM is that it uses what it calls Unformatted System Services tables. Or USS tables for short. This file is normally specified in your TN3270 configuration file.” Big John swiveled his chair and launched his TN3270 client, connected, and opened a file labeled ‘USER.TCPPARMS(TN3270)’ He pointed to a specific line:

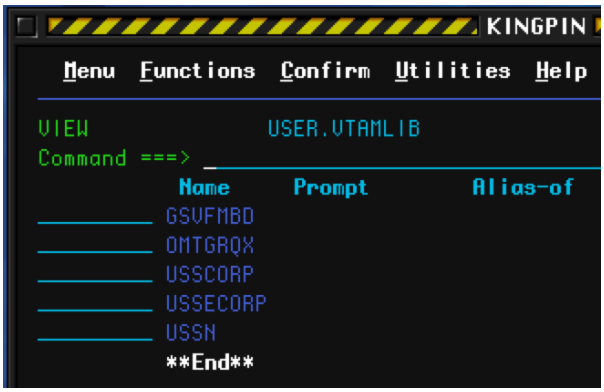
```
1 USSTCP  USSECORP
```

“This line right here tells TCP to tell VTAM to use the file ’USSECORP’ when a client connects.” he said, closing the file. He then opened ’USER.PROCLIB(TN3270)’ and pointed at a different line:

```
1 //STEPLIB     DD  DSN=USER.VTAMLIB,DISP=SHR
```



“And that right there is where we’re gonna find USSECORP,” again he closed the current file and opened another folder: ‘USER.VTAMLIB’. And sure enough, glowing a deep blue, in the back of this van was USSECORP:



"So now you know where to send your compiled HLASM, your 'L.SYSLMOD'. Just overwrite that file and you'll be good to go. Oh wait!" John laughed, "I haven't explained how you can use the USS Table to make it less boring. Right, well it's easy—ish.

"The USS Table is basically a set of macros you call to tell VTAM what to do on each message or command it receives. Let's take a look at this example." He pointed to the other screen.

```
1  USSN   TITLE  'GROOVY SCREEN'
           USSTAB FORMAT=DYNAMIC
3          USSMSG MSG=10,BUFFER=(BUF010,SCAN)
   BUF010 DS   0H
5      DC   AL2(END010-BUF010)
       DC   X'F57A'
7          DC     X'2902C0F842F1'
           DC     C'Hello Flynn'
9          DC     10C' '
           DC     X'13' Insert Cursor
11 END010 EQU *
   END    USSEND
13     END
```

"We start the USS Table with the Macro 'USSTAB' passing it the argument FORMAT. Just always set it to DYNAMIC. This is saying, from here on out we're in USSTAB. The next line"

```
1  USSMSG MSG=10,BUFFER=(BUF010,SCAN)
```

"This calls the USSMSG macro, which you can read in SYS1.SISTMAC1(USSMSG). You can pass it a bunch of variables, but for you, just pass it the MSG= and BUFFER= variables. MSG=10 in our case is the default 'hey you just connected' message. BUFFER takes two arguments. SCAN will look through and replace any instance of keywords with the actual variable. Some examples would be @@@@DATE and @@@@TIME. Which would replace those items with the actual date/time. BUF010 is a pointer. It points to a data structure. The first thing BUFFER expects is the length of the buffer. Since we might add/remove more to our screen we can use just get the total size by subtracting the location of END010 by BEGIN010. Everything else inside there is what will be sent to VTAM to send to your TN3270 emulator. You keepin' up my man?"

"Yeah," I replied. "I think I got it. That line `X'2902C0F842F1'` is a TN3270 command setting the text blue (`\x42 \xF1`) and that other line, two down, with `10C`, just means to repeat that space ten times before we insert the cursor."

John smirked, "well look at you, the artist. When you're done setting USS Tab stuff you just end it with USSEND. Keep in mind, there're fourteen MSGs, not that you'll need to deal with them if you don't want to."

Big John got up and settled into the driver's seat, "Where ya headin?" he asked. I guess he was done teaching me what I needed to learn. "Fifth and Gibson," I replied. Back to my office. I was eager to get started on my own screen now that I knew what I was doing. I buckled in next to Big John and got to the office, thankfully no sight of Rico or his EF Boys.

– – – – — – – — – – – –

Back at my desk I created two things. First, I made a quick and dirty python script so I could rapidly prototype TN3270 command ideas I had (included). Second I decided to code up a macro to handle all the MSG types:

First we needed that sweet, sweet JCL header:

```
1  //COOLSCRN JOB 'build tso screen','IBMUSER',
         NOTIFY=&SYSUID,
   //     MSGCLASS=H, MSGLEVEL=(1,1)
3  //BUILD    EXEC ASMACL
   //C.SYSLIB   DD DSN=SYS1.SISTMAC1,DISP=SHR
5  //           DD DSN=SYS1.MACLIB,DISP=SHR
   //C.SYSIN    DD *
```

Next, I needed a way to handle all the messages. I whipped up a quick macro, with all the colors I might need.

```
         MACRO
&NAME    SCREEN &MSG=.,&TEXT=.
         AIF    ('&MSG' EQ '.' OR '&TEXT' EQ
      '.').END
         LCLC   &BFNAME,&BFSTART,&BFEND
&BLUE    SETC   'X''290142F1'''
&RED     SETC   'X''290142F2'''
&PINK    SETC   'X''290142F3'''
&GREEN   SETC   'X''290142F4'''
&TURQ    SETC   'X''290142F5'''
&YELLOW  SETC   'X''290142F6'''
&WHITE   SETC   'X''290142F7'''
&BFNAME  SETC   'BUF'.'&MSG'
&BFBEGIN SETC   '&BFNAME'.'B'
&BFEND   SETC   '&BFNAME'.'E'
.BEGIN   DS     0F
&BFNAME  DC     AL2(&BFEND-&BFBEGIN)
&BFBEGIN EQU    *
         DC     X'05F7'
         DC     X'110000'
* Fancy art goes here
         DC     X'13'
&BFEND   EQU    *
.END     MEND
```

I needed to address each of the messages, so I did that here. STDTRANS I copied from Big Blue themselves.

```
USSTAB   USSTAB TABLE=STDTRANS,FORMAT=DYNAMIC
         USSMSG MSG=00,BUFFER=(BUF00,SCAN)
         USSMSG MSG=01,BUFFER=(BUF01,SCAN)
         USSMSG MSG=02,BUFFER=(BUF02,SCAN)
         USSMSG MSG=03,BUFFER=(BUF03,SCAN)
         USSMSG MSG=04,BUFFER=(BUF04,SCAN)
         USSMSG MSG=05,BUFFER=(BUF05,SCAN)
         USSMSG MSG=06,BUFFER=(BUF06,SCAN)
         USSMSG MSG=08,BUFFER=(BUF08,SCAN)
         USSMSG MSG=10,BUFFER=(BUF10,SCAN)
         USSMSG MSG=11,BUFFER=(BUF11,SCAN)
         USSMSG MSG=12,BUFFER=(BUF12,SCAN)
         USSMSG MSG=14,BUFFER=(BUF14,SCAN)
STDTRANS DC     X'000102030440060708090A0B0C0D0E0F'
         DC     X'101112131415161718191A1B1C1D1E1F'
         DC     X'202122232425262728292A2B2C2D2E2F'
         DC     X'303132333435363738393A3B3C3D3E3F'
         DC     X'404142434445464748494A4B4C4D4E4F'
         DC     X'505152535455565758595A5B5C5D5E5F'
         DC     X'606062636465666768696A6B6C6D6E6F'
         DC     X'707172737475767778797A7B7C7D7E7F'
         DC     X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'
         DC     X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
         DC     X'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'
         DC     X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
         DC     X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'
         DC     X'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'
         DC     X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'
         DC     X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
END      USSEND
```

After that I call the macro for every msg type and end the HLASM.

```
         SCREEN MSG=00,TEXT='Launchin your program, see'
         SCREEN MSG=01,TEXT='I doubt you meant to do that'
         SCREEN MSG=02,TEXT='No, seriously'
         SCREEN MSG=03,TEXT='Parameter is unrecognized!'
         SCREEN MSG=04,TEXT='Parameter with value is invalid
          '
         SCREEN MSG=05,TEXT='The key you pressed is inactive
          '
         SCREEN MSG=06,TEXT='There is not such session.'
         SCREEN MSG=08,TEXT='Command failed as storage
          shortage.'
         SCREEN MSG=10,TEXT='  '
         SCREEN MSG=11,TEXT='Your session has ended'
         SCREEN MSG=12,TEXT='Required parameter is missing'
         SCREEN MSG=14,TEXT='There is an undefined USS
          message'
END
```

Finally, I added the JCL footer.

```
/*
//L.SYSLMOD DD DSN=USER.VTAMLIB,DISP=SHR
//L.SYSIN   DD *
  NAME USSN(R)
//*
```

Happy with the code I'd just written I made myself a screen I'd be happy to see each and every day:

I shut down my computer, ordered an Uber, and headed out of the office.

A car pulled up as I looked up from my phone. This wasn't my Uber, this was a Tesla, a black Tesla. The back door opened. Rico sat in the back, his one eye covered with a patch, gave him the look of a pirate, as did the gun he had pointed at my face. "Get in," he said, motioning with the large revolver. Having no other option, I shrugged and got in the back of this Tesla-and wondered how much a no-show was gonna cost me on Uber. The Tesla sped off, and slammed me in to the back of my seat.

After a few moments of silence, "Just who the fuck do you think you are?" Rico asked.

"Hey, Rico, all I wanted to do was make a nice logon screen for my mainframe." I quipped. This visibly upset Rico. The driver quietly snickered in the

front seat, then said "This guy thinks he's a sysprog now?"

"Shut up Oren!" Rico turned to me, "It works like this: we control the information. We decide who knows what. You're wastin' everyone's time over some aesthetic changes. The very fact that you phrase it as 'logon screen' means you're not ready to know this information!"

I stammered a response, "Look, I don't get what the big deal is, if you don't want to help who cares?" and I showed him a screenshot of my mainframe.

This was not a good idea. Rico's face turned bright red. "BULLSHIT! You've wasted plenty of people's time! Tommy, Dave, John. You should've gone back and read the manuals, like I had to. All 14,000 pages. Instead, you want a short cut. A hand out. Well, sonny, nothing comes easy. There is no possible way your system didn't come with customization rules, documentation and changes. That just not how it's done!"

I realized at this point Rico had never heard about the fact that you can emulate your own mainframe at home.[30] Oren, turned his head to look at me, "Yeah, there ain't no way you get to run your own system and do what you want all willy-nilly."

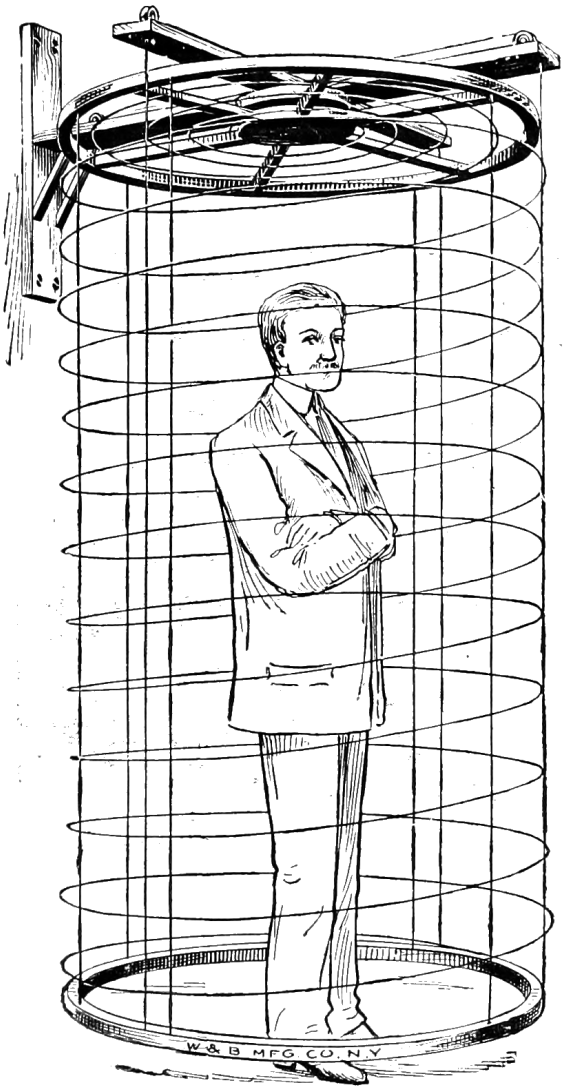
I noticed the red light before Oren and Rico, and got ready to put a dumb plan in to action. Oren slammed on the brakes and sent Rico flying in to the seat in front of him. Why don't bad guys ever wear their seatbelts? While Rico was slightly stunned, I lunged and wrestled the gun free from his hands. At the same time, I grabbed my own pea shooter and pointed one each at Oren and Rico.

"Enough of this shit," I yelled, "you're too late anyway, I've already built and replaced my USS Table." I made sure to use the correct terminology now. "I already shot and missed you once today Rico, I won't miss a second time. Now let me out of this car!"

"Ok, ok. Cool it." said Oren as he slowed the car. Rico just sat and stewed.

I stepped out of the car. "This isn't the last you've heard from us!" Rico yelled, and the black Tesla sped off in to the night.

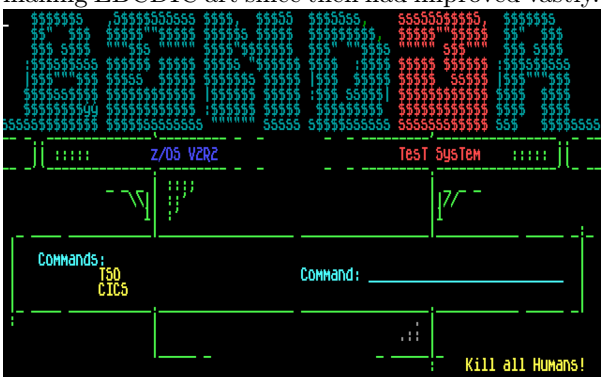
He was right, of course. It wouldn't be the last time I clashed with the EF gang and lived to tell about it.



[30] https://www.ibm.com/us-en/marketplace/z-systems-development-test-environment

– — — – — — — — – — –

I couldn't believe that was six years ago. Bigendian knew to reach out to me because I had done some nice screens for him in the past. My skills at making EBCDIC art since then had improved vastly.



Thanks to another meeting years later with Big John, I learned you can add lines and graphics to make shapes using the rarely documented SFE GE SHAPE (\x08) command. At this point, I had the three-headed beast as a rough idea in my head what I wanted the screen to look like. But, I needed a way to animate the Windows 10 update nag screen.

Like a small dog running in to a screen door, it hit me. I could use the MSGs and an AIF to display the nag screen!

You see, when you first connect, that's a MSG10 screen. If you hit enter, to the user it appears as though the screen just refreshed. But what's really happening is VTAM loads a MSG02 screen. Because you entered an invalid command (nothing). I could use an AIF statement to only show the Windows 10 nag screen if an invalid command was entered.

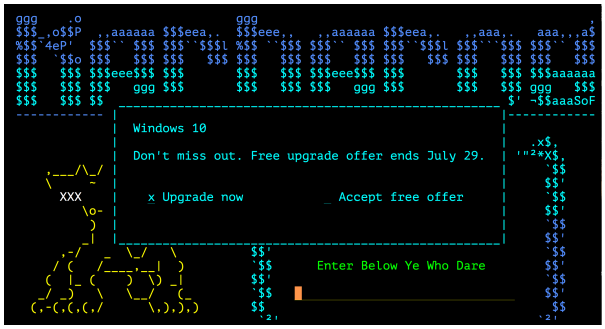
Above, where I declared the colors, I could also declare some shapes:

```
1 &UPRIGHT     SETC   'X''08D5'''
  &DOWNRIGHT   SETC   'X''08D4'''
3 &UPLEFT      SETC   'X''08C5'''
  &DOWNLEFT    SETC   'X''08C4'''
5 &HBAR        SETC   'X''08A2'''
  &VBAR        SETC   'X''0885'''
```

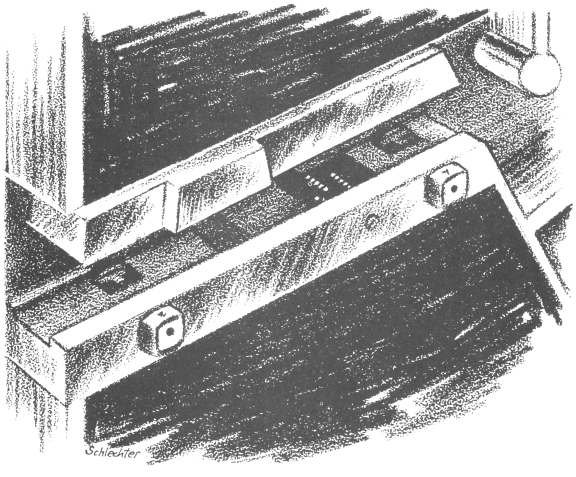
And, with the help of Tommy's table, the one that gave me the coordinates for screen positions, and Big John's graphics, I could overlay the nag box on the screen. But only if the MSG is type 02. See Figure 21.

With that final piece of the puzzle I gave Bigendian Smalls a short demo.

Then I hit <enter>, and it all came together.



"Wow, that's really awesome." he replied over ICQ. It sure was.



```
          AIF ('&MSG' NE '02').SKIP
*  TOP BAR
          DC    X'11C76D'                SBA, 1050     ROW 10 COL 13
          DC    &COLOR&BG&TURQ
          DC    &UPLEFT
          DC    52&HBAR
          DC    &UPRIGHT
* BOX WALLS
          DC    X'11C87D'                SBA, ROW 11 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'11C9F3'                SBA, ROW 11 COL 66
          DC    &VBAR
          DC    X'114A4D'                SBA, ROW 11 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'114BC3'                SBA, ROW 11 COL 66
          DC    &VBAR
          DC    X'114B5D'                SBA, ROW 11 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'114CD3'                SBA, ROW 11 COL 66
          DC    &VBAR
          DC    X'114C6D'                SBA, ROW 11 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'114DE3'                SBA, ROW 11 COL 66
          DC    &VBAR
          DC    X'114D7D'                SBA, ROW 11 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'1103B3'                SBA, ROW 11 COL 66
          DC    &VBAR
          DC    X'114F4D'                SBA, ROW 12 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'110403'                SBA, ROW 12 COL 66
          DC    &VBAR
          DC    X'11505D'                SBA, ROW 13 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'110453'                SBA, ROW 13 COL 66
          DC    &VBAR
          DC    X'11D16D'                SBA, ROW 14 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'1104A3'                SBA, ROW 14 COL 66
          DC    &VBAR
          DC    X'11D27D'                SBA, ROW 15 COL 13
          DC    &COLOR&BG&TURQ
          DC    &VBAR
          DC    52C' '
          DC    X'1104F3'                SBA, ROW 15 COL 66
          DC    X'0885'
*  BOTTOM BAR
          DC    X'11050D'                SBA, ROW 16 COL 13
          DC    &COLOR&BG&TURQ
          DC    &DOWNLEFT
          DC    52&HBAR
          DC    &DOWNRIGHT
* INSIDE BOX
          DC    X'114A50'                SBA, ROW 11 COL 16
          DC    &COLOR&BG&TURQ
          DC C'Windows 10'
          DC    X'114CF1'                SBA, ROW 13 COL 16
          DC C'Don''t miss out. Free upgrade offer ends July 29.'
* ACCEPT LINE
          DC    X'1150E3'                SBA, ROW 15 COL 18
          DC C'x Upgrade now              Accept free offer'
* UNDERLINES
          DC    X'1150E2'                SBA, ROW 15 COL 18
          DC    X'290341F442F5C0C8'      SFE, UNPROTECTED/UNDL/TURQ
          DC    C'x'
          DC    &COLOR&BG&TURQ
          DC    X'11507A'                SBA, ROW 15 COL 42
          DC    X'290341F442F5C0C8'      SFE, UNPROTECTED/UNDL/TURQ
          DC    X'40'
          DC    &COLOR&BG&TURQ
.SKIP     ANOP
```

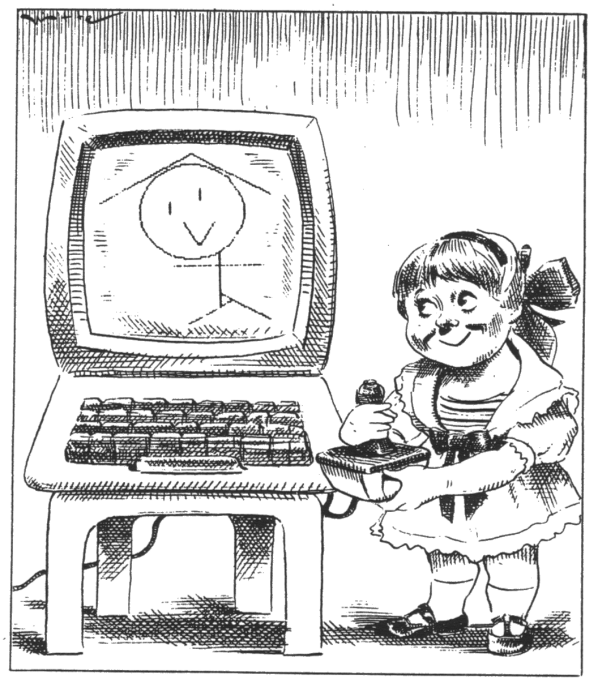
Figure 21. Upgrade Offer

17:09 Protecting ELF Files by Infecting Them

by Leandro "acidx" Pereira

Writing viruses is a sure way to learn not only the intricacies of linkers and loaders, but also techniques to covertly add additional code to an existing executable. Using such clever techniques to wreck havoc is not very neighborly, so here's a way to have some fun, by injecting additional code to tighten the security of an ELF executable.

Since there's no need for us to hide the payload, the injection technique used here is pretty rudimentary. We find some empty space in a text segment, divert the entry point to that space, run a bit of code, then execute the program as usual. Our payload will not delete files, scan the network for vulnerabilities, self-replicate, or anything nefarious; rather, it will use `seccomp-bpf` to limit the system calls a process can invoke.



[31] `man 2 bpf`

Caveats

By design, `seccomp-bpf` is unable to read memory; this means that string arguments, such as in the `open()` syscall, cannot be verified. It would otherwise be a race condition, as memory could be modified after the filter had approved the system call dispatch, thwarting the mechanism.

It's not always easy to determine which system calls a program will invoke. One could run it under `strace(1)`, but that would require a rather high test coverage to be accurate. It's also likely that the standard library might change the set of system calls, even as the program's local code is unchanged. Grouping system calls by functionality sets might be a practical way to build the white list.

Which system calls a process invokes might change depending on program state. For instance, during initialization, it is acceptable for a program to open and read files; it might not be so after the initialization is complete.

Also, `seccomp-bpf` filters are limited in size. This makes it more difficult to provide fine-grained filters, although eBPF maps[31] could be used to shrink this PoC so slightly better filters could be created.

Scripting like a kid

Filters for `seccomp-bpf` are installed using the `prctl(2)` system call. In order for the filter to be effective, two calls are necessary. The first call will forbid changes to the filter during execution, while the second will actually install it.

The first call is simple enough, as it only has numeric arguments. The second call, which contains the BPF program itself, is slightly trickier. It's not possible to know, beforehand, where the BPF program will land in memory. This is not such a big issue, though; the common trick is to read the stack, knowing that the `call` instruction on x86 will store the return address on the stack. If the BPF program is right after the call instruction, it's easy to obtain its address from the stack.