2025/12/2025 — REVERSE ENGINEERING NOTES

FIRST ATTEMPT WITH THE NOTEPAD I HAVENT USED IN EONS
↳ ALSO BARELY USED IT WHEN I FIRST BOUGHT IT
↳ IT WILL DEFINETLY TAKE SOME GETTING USED TO BUT I THINK THAT I SETITUP WELL...

HOW DIFFRENT IS MY MOUSE WRITING VS HOW MY WRITING IS W/ THE PEN.

↳ MUCH FASTER & ALOT MORE LEGIBLE BUT STILL NEEDS WORK
↳ LOOKS LIKE A ~~DUMMY~~ WROTE IT
↳ GETTING BETTER?

ALSO HURTS MY WRIST LESS AS THERE IS LESS FORCE NEEDED TO WRITE.

STACK: USED FOR FUNCTION CALLS AND THEIR LOCAL VARIABLES
↳ KEY PROPERTIES:
↳ AUTOMATIC
↳ L.I.F.O
↳ LAST IN FIRST OUT
↳ FAST
↳ LIFETIME ADDED TO SCOPE

- TYPICALLY USED TO STORE
  ↳ FUNCTION RETURN ADDRESSES
  ↳ SAVED REGISTERS
  ↳ FUNCTION ARGUEMENTS
  ↳ LOCAL VARIABLES

HEAP: A BIG POOL OF MEMORY USED FOR DYNAMIC ALLOCATIONS, WHERE YOU DECIDE AT RUNTIME HOW MUCH TO ALLOCATE & WHEN TO FREE IT
↳ KEY PROPERTIES:
↳ MANUAL (MALLOC, NEW, FREE, DELETE)
↳ FLEXIBLE SIZE & LIFETIME
↳ OBJECTS CAN
↳ BE LARGE
↳ OUTLIVE THE FUNCTION THAT CREATED THEM
↳ BE SHARED ACROSS FUNCTIONS
↳ SLOWER THAN THE STACK

TYPICALLY USED TO STORE:
↳ OBJECTS CREATED WITH
↳ DATA STRUCTURES WHOSE SIZE IS ONLY KNOWN AT RUNTIME
↳ LONG LIVED PROGRAM STATE(S)

THERE IS A PUSH INSTRUCTION FOR CRACKME.234008
↳ FEW INSTRUCTIONS LATER THIS IS LOADED INTO REGISTER EAX TO BE USED IN A CMP AGAINST 0xA
↳ DECIMAL VALUE: 10
↳ IM PRETTY SURE THAT IS A NEWLINE CHARACTER
↳ YUP. IM RIGHT → /N
↳ 10 = 0xA = NL LINEFEED, NEW LINE
↳ AFTER THE CMP INSTRUCTION THERE IS A JE JUMP
↳ JUMPS IF EAX = 0xA
↳ NEW LINE CHARACTER
↳ PROBABLY CHECKING FOR EMPTY INPUT?
↳ AFTER CMP WE HAVE ANOTHER CMP INSTRUCTION

↳ CMP EAX, 0x9
↳ NOT REALLY SURE WHAT THIS IS DOING YET
↳ DECIMAL VALUE = 9
↳ ASCII FOR HORIZONTAL TAB
↳ \t → /6

↳ AFTER PASSING THIS SECOND CMP WE HIT A JUMP
↳ WHY ARE MY H'S & Y'S SO [redacted]...
↳ HHHHHHHHHHHHHHHHHH
↳ YYYY YYYYYYYY
↳ JUMPS TO ADDRESS 0x7310099
↳ WHICH IS THE AUTHENTICATION FAILED BRANCH LOGIC
↳ PUSHES "AUTHENTICATION FAILED", CALLS PRINTF, THEN JUMPS TO EXITPROCESS THUNK.

DOES THIS MEAN THE FIRST CHARACTER NEEDS TO BE A TAB?
↳ LETS TRY IT OUT
↳ AUTHENTICATION STILL FAILED
↳ LETS STEP THROW IT

MY NICEST LINE YET

↳ SAME RESULT... → 1 IF SUCCESS, 0 OTHERWISE
→ GETS STORED IN EAX (x86 32-BIT)

- BOOL READCONSOLEA (
  - HANDLE — H CONSOLE INPUT
  - LPVOID — LPBUFFER → POINTER WHERE THE ACTUAL CHARACTERS ARE WRITTEN
  - DWORD — NNUMBER OF CHARS TO READ
  - LPDWORD — LPNUMBEROFCHARSREAD
  - LPVOID — PINPUTCONTROL → USUALLY NULL

)

→ POINTER TO A DWORD WHERE THE FUNCTION STORES HOW MANY CHARS WERE READ

- EAX SEEMS TO CHANGE BASED ON THE INPUT AT 'CMP EAX, 0XA'
  - GETS LOADED FROM CRACKME.7314008
    - THIS ADDRESS GOT PUSHED A FEW INSTRUCTIONS ABOVE
- 7314008
  - GLOBAL VARIABLE (DWORD) IN THE DATA SECTION

RECALL BEFORE THE READCONSOLEA CALL THERE ARE MULTIPLE PUSH INSTRUCTIONS

↳ PUSH	0	→ LPRESERVED
↳ PUSH	CRACKME.7314008	→ NUMBER OF CHARS READ (LPDWORD)
↳ PUSH	0x100	→ NUMBER OF CHARS TO READ (DWORD)
↳ PUSH	CRACKME.7310099	→ LPBUFFER
↳ PUSH	DS:[731000]	→ HCONSOLEINPUT (LPVOID)

^ DWORD PTR

→ THESE INSTRUCTIONS ARE LOADING THE REQUIRED ARGUMENTS FOR THE METHOD READCONSOLEA

- SO...
  - WITH THAT, WE NOW KNOW THAT THE FIRST CMP INSTRUCTION IS NOT CHECKING FOR NEW LINE / END OF LINE OR INPUT / EMPTY INPUT
  - IT IS ACTUALLY CHECKING THE INPUTS LENGTH.
    - ↳ CMP EAX, 0XA
      - ↳ ENSURE LENGTH IS 10 CHARACTERS LONG
      - ↳ IF IT IS 10 CHARACTERS LONG THEN JUMP TO COMPARISON LOGIC

↳ IF THE INPUT IS NOT 10 CHARACTERS LONG THEN CHECK IF THE INPUT IS 9 CHARACTERS LONG
↳ CMP EAX, 0x9
↳ IF NOT 9 OR 10 CHARACTERS LONG THEN JUMP TO AUTHENTICATON FAILED LOGIC
↳ IF IT IS, THEN PROCEED TO FURTHER VALIDATION STEPS

ENTERING A STRING THAT IS 10 CHARACTERS LONG: HELLOWORLD (LOWERCASE)
↳ 734008 = 0xC = 12 ??? WHY & HOW
↳ WHAT IS 734008 FOR 8 CHARACTERS LONG: HELLOWRLD (LOWERCASE)
↳ THIS RESULTED IN 734008 = 0xA = 10
↳ THIS IS BECAUSE, IN LINE MODE, READCONSOLEA INCLUDES THE CR+LF FROM YOUR PRESSING ENTER
↳ AFTER HITTING ENTER WHAT THE BUFFER ACTUALLY HOLDS:
↳ USERINPUT + CR LF
↳ 0xD & 0xA
↳ ARDS TWO BYTES
TO THE LENGTH

- ANYTHING THAT LOOKS LIKE MODULE NAME.ADDRESS IS A SYMBOL INSIDE THAT MODULE'S IMAGE
↳ AGAIN, MEANS IT'S NOT ON THE STACK.
↳ NOT ON HEAP EITHER → NO HEAPALLOC, NO POINTER STORED SOMEWHERE
- DWORD = 32 BIT UNSIGNED INTEGER
- IN READCONSOLEA THE SECOND PARAMETER LPVOID -LPBUFFER IS A POINTER TO A BUFFER IN MEMORY WHERE THE USER'S INPUT WILL BE STORED
↳ PUSH CRACKME.732DD9 → THIS IS THE ADDRESS WHERE THE USER'S INPUT WILL BE STORED
- LP = LONG POINTER
- WORD = 16-BIT UNSIGNED
- BYTE = 8-BIT UNSIGNED
- QWORD = 64-BIT BIT
↳ LESS COMMON IN WINAPI TYPEDEFS
- SO... THE CMP EAX, 0xA IS ACTUALLY CHECKING IF THE INPUT IS 8 CHARACTERS LONG WHILST 0x9 IS CHECKING IF THE INPUT IS 7 CHARACTERS LONG
↳ FIRST CLUE CONFIRMED: FLAG MUST BE 8 OR 7 CHARACTERS LONG
↳ GOING BACK & CHANGING MY INPUT BY SHORTENING IT BY 2 CHARACTERS
↳ "HELLOWOR" (LOWERCASE)
↳ CLUE CONFIRMED → JUMP TAKEN TO FURTHER VALIDATION STEPS
↳ CALL CRACKME.7310D8
↳ I ASSUME THIS IS THE COMPARISON FUNCTION AS THE IMMEDIATE FOLLOWING INSTRUCTION IS A CMP EAX, EAX
↳ WHICH MEANS ITS CHECKING IF THE RETURN VALUE IS NOT ZERO
↳ SO... I NEED TO SET A BREAKPOINT ON THAT CALL & STEP INTO IT
- CMP AL, BYTE PTR DS:[EDI]
↳ EDI IS A POINTER, [EDI] IS THE VALUE THAT IT POINTS TO
- IN THE VALIDATION FUNCTION WE ARE MOVING POINTERS INTO REGISTERS EDI & ESI
↳ MOV ESI, CRACKME.732DD9 → GLOBAL VARIABLE HOLDING USER INPUT
↳ MOV EDI, CRACKME.732DB → GLOBAL VARIABLE HOLDING CHARACTER SET
↳ THE MOV AL & CMP AL INSTRUCTIONS THAT FOLLOW ↳ ARE WITHIN A LOOP
↳ FIRST ITERATION → COMPARE USER INPUT [FIRST CHARACTER] AGAINST FIRST CHARACTER OF CHARACTER SET (@)

↳ THE INCREMENTS FOR EDI & ESI INDICATE THAT THE POINTERS ARE MOVING FORWARD BY 1 BYTE EACH LOOP (8-BITS) , 1 CHARACTER.

↳ BY THIS INTUITION I WILL JUST GRAB THE FIRST 8 CHARACTERS & INPUT THEM AS THE FLAG TO CHECK IF IM RIGHT.

YUP - I WAS RIGHT & FOUND THE FLAG: @CBGD GF1

↳ BUT I WAS WRONG ABOUT EITHER 7 OR 8.

↳ EITHER 7 OR 8 CHARACTER INPUT WILL BE PASSED TO THE VALIDATION FUNCTION BUT IT WILL ALWAYS VALIDATE FOR 8 CHARACTERS. SO EVEN THOUGH A 7 CHARACTER INPUT WILL PASS THE MAIN FUNCTION CHECKS & PROCEED TO THE VALIDATION FUNCTION. THE FLAG / INPUT NEEDS TO BE 8 CHARACTERS.