recruitment CTF

2024-25

CHALLENGE NAME: [be-in-your-limits]

DEV: [Pushkar deore]

CATEGORY: [Rev Engg]

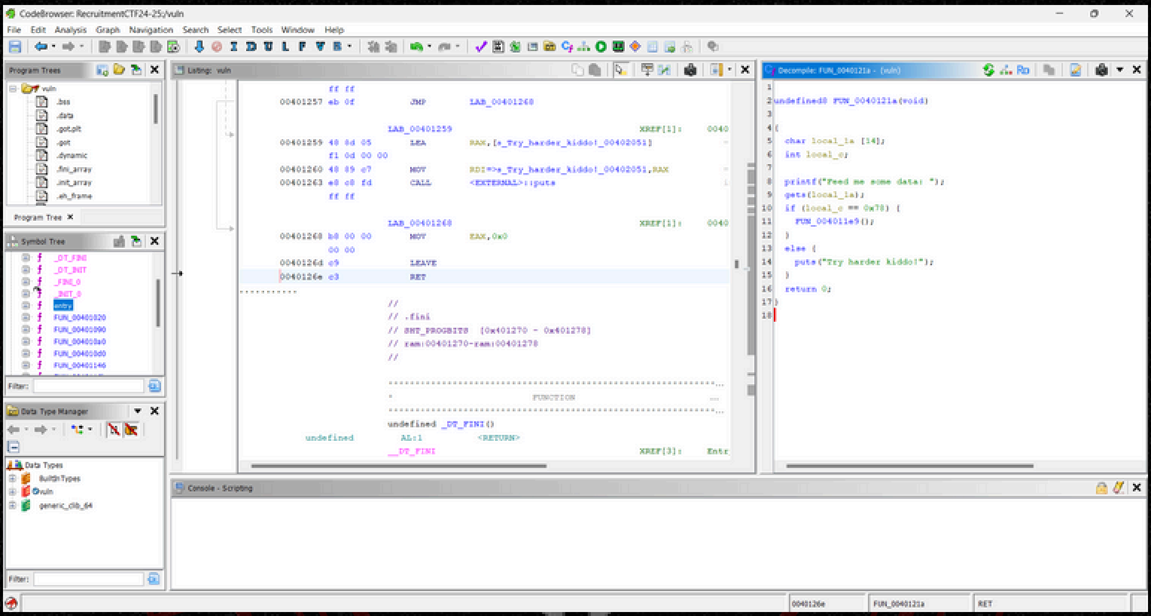
LEVEL: [EASY]

Challenge Description:

Keep them under check.

WRITEUP:

1) We are given an ELF file called vuln. If we run it, it asks for some data which might give us the flag. To craft a suitable payload, first let us open it up in Ghidra.



```
                             ff ff
        00401257 eb 0f           JMP        LAB_00401268

                             LAB_00401259                          XREF[1]:     0040
        00401259 48 8d 05        LEA        RAX,[s_Try_harder_kiddo!_00402051]
                 f1 0d 00 00
        00401260 48 89 c7        MOV        RDI=>s_Try_harder_kiddo!_00402051,RAX
        00401263 e8 c8 fd        CALL       <EXTERNAL>::puts
                 ff ff

                             LAB_00401268                          XREF[1]:     0040
        00401268 b8 00 00        MOV        EAX,0x0
                 00 00
        0040126d c9              LEAVE
        0040126e c3              RET

                             //
                             // .fini
                             // SHT_PROGBITS  [0x401270 - 0x401278]
                             // ram:00401270-ram:00401278
                             //

                             **************************************************************
                             *                          FUNCTION                          *
                             **************************************************************
                             undefined _DT_FINI()
             undefined         AL:1           <RETURN>
                             __DT_FINI                             XREF[3]:     Entr
```

```

undefined8 FUN_0040121a(void)

{
  char local_1a [14];
  int local_c;

  printf("Feed me some data: ");
  gets(local_1a);
  if (local_c == 0x78) {
    FUN_004011e9();
  }
  else {
    puts("Try harder kiddo!");
  }
  return 0;
}
```

2) Here, we can see an array of length 14 and an char variable. In order to proceed further, the char variable must be set to 0x78, decimal equivalent of 120, which is ASCII equivalent of ‘x’.

3) But the problem is that we are only able to provide input for the array and not the char. Speaking of input, we observe that the input is taken using the ‘gets()’ function, which is vulnerable to buffer overflow.

4) From the above mentioned information, we should try to do a buffer overflow, such that we are also able to satisfy the char value condition.

5) Therefore an appropriate input should be “aaaaaaaaaaaaaax”. Here first 14 ‘a’s will fill up the array and the 15th character ‘x’ will be stored in char.

6) This is due to the fact that gets() function doesn’t check the size of input and simply stores it. This vulnerability is known as buffer overflow.

7) This will satisfy the condition and hence print the flag.