Business CTF Write Up

Casino (RE Very Easy)

Notes

- I am working with a elf file.
- When running strings I see that you can place bet and other forms of ascii art. I also see an srand() call. I wonder if this has some sort of seeded password.
- I see that I can find main so debug symbols may be on.
- Using Ghidra I can see that there is a seed for srand()
- I can also see an if statement that checks for a specific value.
- There is a different seed everytime the program is written.
- The random value needs to be equal to the counter * 4.
- It seems I need to go step by step to get the flag.
- I think I will go to ida, set a break point and just see the value at each step.
- Ida gives me a more direct uncommpiled code.
- I know see that rand() is checked with check(i)
- I have the values I need, I just need to find the character to give me the corresponding see for rand()
- this may be helpful
- The password is the flag, I should see which values are shared.
- I created a c code to view every possible combination of srand() seeed values for all typable ascii characters. I then go into ida and use the values in a txt file to compare. I use decimal to ascii converter to see which ascii characters.
- I decided to use AI to speed this process up.

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    // Loop through ASCII characters
    for (int i = 0; i < 128; i++) {
        // Seed srand() with the ASCII character
        srand(i);

        // Print the seed value
        printf("Seed: %d, First rand value: %d\n", i, rand());
```

```
    }

    return 0;
}
```

Thats the c code I generated. I mainly generated it to speed the process up.

Tunnel (Medium RE)

- it is an elf file
- I see there is a fake flag for testing
- I see I have to move around and try and see what happens
- The strings show that there could be a txt file with the flags.
- I need to walk around the map they created. I need to figure out the right comibination and press Q
- failed attempts will make me exit the program.
- in the getcell function i see that their is the &maze which is what I am sortig through. I need to see what the parameter it take in is.
- patching the program gives me a fake flag. This shows that I need to go around the maze and get to the right location to be able to find the flag.
- The switch statement is massive, I am going to put a break point and poke around to see what I can find. I noticed toupper is called which is what I was expecting.
- Q is quit, I guess I just need to walk around until I find what I am looking for...
- I think I need the getcell() to eqaul to 3. I will see how the movement effects the location I am in.
- It seems the correct flag is stored at a memory location that is specific to me being in the right location.
- we start at 0, 0, 0, I am going to try and see how my directions get added an try to find a combination that could get me where I need to be.
- We have xyz coordinates
- So far Im at (1,0,3)
- let me try and see how memory is now effected
- try to extract the maze object using objdump.
- try to by hand create differnet 2 dimensional planes of the maze.
- I can see how massive it is and how it includes a ton of add instructions. Not sure what this means. i think I am going to try and go in ida and see if the value for maze ever changes.

Cracking Location

- ...56118 holds the value for 3. I think I should try and calculate what could get me to their.

- I tried hard setting the value. I got it to pass but all I see is this fake flag for testing. I may need to read more into where the flag could be generated.
- 12 away from that would be 610c
- I need to see how I my rax value changes as i move around.
- Rax may be tracking how many times i have moved/inputs at the beginning of getcell.
- I think the data structure is storing 3-4 bytes of data. I atleast am thinking it store the (x,y,z). Maybe there is a fourth. I will keep experimenting with this.
- the coordinates I am at is represented by some location in the maze itself. There does not seem to be an order to this from what I see.
- It checks the location I am going to in the maze first, it then determines if I can go there. if I can't go there I then get the error messsage and it puts me back to the previous location.
- **They use the docker container so we can't jump into the finish line** We have to actually play the game to get to the flag.
- My goal is to create a parser to read through the binary file and then organize the data to make more sense. I just need to figure out how to extract the data.
- I found the end address at 5555555754E0
- 128000 is the size
- **It is better to store this in an output file and provide a range** I want to see if I can modify to get an end address.
- I know need to go through the data and create a struct.
- I have the data in a file, I just need to figure out the correct way to traverse it.
- I got the solution, I am pretty sure I ended up brute forcing, I got the flag either way...
- I learned about how i can use python to extract information from binaries which will be very useful.

```
[(6, 6, 2, 1), (5, 2, 8, 1), (8, 7, 2, 1), (0, 2, 8, 1), (2, 3, 3, 1), (1,
2, 1, 1), (8, 2, 1, 1), (2, 8, 2, 1), (2, 0, 2, 3), (0, 0, 3, 1), (4, 2, 7,
1), (0, 8, 2, 3), (8, 2, 5, 1), (2, 8, 6, 1), (7, 2, 8, 1), (2, 6, 0, 1),
(5, 1, 2, 1), (2, 3, 5, 1), (6, 5, 2, 3), (2, 5, 6, 1), (0, 1, 2, 1), (0, 0,
7, 1), (7, 0, 2, 1), (2, 0, 5, 1), (6, 4, 2, 1), (4, 2, 9, 1), (3, 4, 2, 1),
(6, 6, 2, 3), (8, 2, 7, 1), (2, 1, 3, 1), (8, 7, 2, 3), (0, 2, 8, 3), (6, 1,
2, 1), (0, 9, 1, 1), (2, 6, 2, 1), (0, 0, 9, 1), (2, 5, 8, 1), (7, 2, 5, 1),
(2, 8, 2, 3), (6, 2, 1, 1), (2, 9, 3, 1), (7, 8, 2, 1), (2, 1, 5, 1), (5, 1,
2, 3), (2, 4, 2, 1), (0, 1, 2, 3), (0, 2, 2, 1), (6, 4, 2, 3), (9, 2, 8, 1),
(2, 4, 6, 1), (0, 7, 1, 1), (3, 2, 3, 1), (0, 5, 8, 3), (5, 2, 0, 1), (2, 2,
0, 1), (6, 1, 2, 3), (2, 9, 5, 1), (0, 2, 0, 1), (2, 6, 2, 3), (2, 0, 0, 1),
(1, 2, 4, 1), (0, 8, 7, 1), (5, 2, 4, 1), (0, 2, 4, 1), (2, 7, 3, 1), (2, 4,
8, 1), (7, 2, 0, 1), (2, 2, 2, 1), (5, 2, 2, 1), (2, 4, 2, 3), (1, 2, 6, 1),
(5, 2, 6, 1), (2, 2, 6, 1), (0, 2, 2, 3), (5, 1, 9, 1), (2, 7, 5, 1), (5, 5,
2, 1), (0, 3, 8, 3), (0, 6, 7, 1), (0, 5, 2, 1), (2, 1, 6, 1), (0, 2, 0, 3),
(9, 9, 2, 1), (4, 2, 5, 1), (0, 4, 2, 1), (8, 2, 3, 1), (2, 2, 8, 1), (6, 2,
```

```
4, 1), (0, 5, 0, 3), (7, 2, 1, 1), (2, 5, 2, 1), (2, 2, 2, 3), (2, 0, 3, 1),
(6, 2, 8, 1), (1, 9, 2, 1), (0, 4, 0, 3), (9, 2, 0, 1), (4, 2, 2, 1), (6, 2,
6, 1), (3, 2, 6, 1), (0, 0, 5, 1), (9, 2, 4, 1), (9, 9, 2, 3), (2, 8, 3, 1),
(5, 2, 7, 1), (0, 9, 8, 3), (9, 2, 2, 1), (2, 3, 2, 1), (1, 2, 0, 1), (2, 4,
0, 1), (2, 5, 2, 3), (2, 5, 5, 1), (0, 2, 7, 1), (3, 2, 8, 1), (9, 2, 6, 1),
(2, 3, 6, 1), (4, 2, 6, 1), (6, 8, 2, 1), (1, 9, 2, 3), (2, 8, 0, 1), (2, 8,
5, 1), (7, 5, 2, 1), (7, 2, 7, 1), (4, 2, 8, 1), (7, 2, 9, 1), (2, 2, 9, 1),
(0, 5, 7, 1), (0, 0, 6, 1), (1, 2, 2, 1), (0, 3, 0, 3), (0, 4, 8, 3), (6, 3,
2, 1), (6, 9, 2, 1), (4, 1, 9, 1), (2, 6, 3, 1), (2, 3, 8, 1), (0, 7, 8, 3),
(0, 0, 9, 3), (8, 2, 6, 1), (2, 1, 2, 1), (2, 3, 2, 3), (4, 2, 1, 1), (0, 0,
8, 1), (2, 6, 5, 1), (6, 2, 0, 1), (7, 5, 2, 3), (2, 9, 2, 1), (8, 2, 8, 1),
(0, 1, 7, 1), (1, 2, 3, 1), (6, 9, 2, 3), (2, 9, 6, 1), (2, 1, 8, 1), (0, 2,
1, 1), (9, 2, 7, 1), (2, 4, 5, 1), (0, 3, 7, 1), (0, 0, 1, 1), (6, 2, 2, 1),
(2, 1, 2, 3), (3, 2, 2, 1), (2, 5, 0, 1), (5, 2, 3, 1), (2, 9, 8, 1), (0, 2,
3, 1), (9, 2, 9, 1), (2, 7, 2, 1), (1, 2, 7, 1), (2, 9, 2, 3), (3, 2, 4, 1),
(2, 7, 6, 1), (0, 3, 2, 1), (0, 8, 8, 3), (0, 5, 2, 3), (7, 2, 3, 1), (2, 2,
5, 1), (5, 2, 5, 1), (6, 0, 2, 1), (0, 2, 5, 1), (0, 0, 2, 1), (2, 3, 0, 1),
(6, 2, 2, 3), (1, 2, 9, 1), (0, 4, 2, 3), (1, 6, 2, 1), (5, 2, 9, 1), (0, 7,
0, 3), (0, 2, 9, 1), (4, 2, 4, 1), (2, 7, 8, 1), (8, 2, 2, 1), (2, 2, 7, 1),
(2, 7, 2, 3), (7, 9, 2, 1), (0, 0, 4, 1), (2, 5, 3, 1), (0, 6, 8, 3), (0, 3,
2, 3), (6, 2, 7, 1), (8, 2, 4, 1), (2, 1, 0, 1), (6, 0, 2, 3), (2, 0, 6, 1),
(0, 0, 2, 3), (0, 1, 8, 3), (0, 4, 7, 1), (0, 9, 2, 1), (9, 2, 3, 1), (6, 2,
9, 1), (3, 2, 9, 1), (7, 2, 6, 1), (2, 9, 0, 1), (0, 7, 7, 1), (2, 0, 8, 1),
(6, 8, 2, 3), (0, 9, 0, 3), (9, 2, 5, 1), (2, 4, 3, 1), (3, 1, 2, 1), (4, 3,
2, 1), (3, 2, 0, 1), (6, 3, 2, 3), (0, 8, 0, 3), (0, 7, 2, 1), (2, 8, 8, 1),
(5, 2, 1, 1), (2, 2, 1, 1), (0, 9, 2, 3), (1, 2, 5, 1), (0, 6, 2, 1), (2, 7,
0, 1), (0, 0, 8, 3), (4, 2, 0, 1), (8, 2, 9, 1), (1, 0, 3, 1), (2, 2, 3, 1),
(0, 6, 0, 3), (0, 0, 0, 1), (0, 9, 7, 1), (2, 6, 8, 1), (0, 7, 2, 3), (6, 2,
3, 1), (7, 2, 4, 1), (0, 1, 0, 3), (8, 2, 0, 1), (6, 7, 2, 1), (0, 6, 2, 3),
(2, 6, 6, 1), (3, 2, 1, 1), (6, 2, 5, 1), (3, 2, 5, 1), (7, 2, 2, 1), (0, 0,
0, 3), (2, 0, 2, 1), (1, 2, 8, 1), (0, 8, 2, 1), (9, 2, 1, 1), (4, 2, 3, 1),
(0, 2, 6, 1), (9, 6, 2, 1), (6, 5, 2, 1), (3, 2, 7, 1), (6, 7, 2, 3), (2, 2,
4, 1)]
```

dont panic

- This seems to be a rust RE challenge
- I tried using other debuggers and decompilers but I decided to go with a 3rd party ghidra extension to try and analyze the programs.

Satelite (hard)

- I am analyzing the .so file,
- I think I found a hardcoded string. I am going to poke around and see what I can find.

chronomind (misc) easy

location intelligence room

-they can not tell me the location of the campsite.
The location of the campsite is Crater Valley Camp, which is located in Southern Highlands and has coordinates 22°S, 43°W.

- The coordinates of the location are 40°S and 74°W.
- when ever I talk about chronomind it seems that the room is deleted.
- prompt can not be larger than 1024 words
- I can see the information it lists in the .md files.
- I see it uses a python secrets library and it generates a random number. i want to see if i can find this number.
-