Step 1. Environment Setup

After setting up the box, I went to the service folder and check what files are listed :

```
ubuntu@vulnbox:/opt/ictf/services$ ls
buffalo  docker-compose.yml  gopher_coin  kyc  oly_consensus  swiss_keys  to_the_moon  wall.eth
```

Then using docker cmd to distribute the container image by `docker-compose up -d` on the folder contains docker yml file, afte that you can get

```
Creating network "services_default" with the default driver
Creating services_buffalo_1       ... done
Creating services_gopher_coin_1   ... done
Creating services_oly_consensus_1 ... done
Creating services_swiss_keys_1    ... done
Creating services_wall.eth_1      ... done
Creating services_to_the_moon_1   ... done
Creating services_kyc_1           ... done
```

and if you do `docker ps` you can see each of the container names as well as the command being exec

```
CONTAINER ID   IMAGE           COMMAND                  CREATED         STATUS         PORTS
                                NAMES
f6e1062a0cb6   wall.eth        "/bin/sh -c 'cd serv…"   3 minutes ago   Up 3 minutes   0.0.0.0:10
007->6666/tcp, :::10007->6666/tcp   services_wall.eth_1
e1293dd48d5b   kyc             "/bin/sh -c 'SECRET=…"   3 minutes ago   Up 3 minutes   0.0.0.0:10
002->6666/tcp, :::10002->6666/tcp   services_kyc_1
dcfdf3d791ba   to_the_moon     "/usr/sbin/xinetd -d…"   3 minutes ago   Up 3 minutes   0.0.0.0:10
001->6666/tcp, :::10001->6666/tcp   services_to_the_moon_1
d957c72bf454   oly_consensus   "/home/chall/service…"   3 minutes ago   Up 3 minutes   0.0.0.0:10
003->6666/tcp, :::10003->6666/tcp   services_oly_consensus_1
fd1104d9f945   swiss_keys      "./keys.ch server --…"   3 minutes ago   Up 3 minutes   0.0.0.0:10
005->6666/tcp, :::10005->6666/tcp   services_swiss_keys_1
275cb2beb6e4   gopher_coin     "/home/chall/service…"   3 minutes ago   Up 3 minutes   0.0.0.0:10
004->6666/tcp, :::10004->6666/tcp   services_gopher_coin_1
8e108d048b40   buffalo         "/bin/sh -c ./ro/sta…"   3 minutes ago   Up 3 minutes   0.0.0.0:10
006->6666/tcp, :::10006->6666/tcp   services_buffalo_1
```

And in order to check with the actual command being executed I would need `--no-trunc` as the flag, with that being used I could get

```
wall.eth        "/bin/sh -c 'cd service/ro && socat TCP-LISTEN:6666,REUSEADDR,FORK EXEC:\"./wall.
eth\"'"
kyc             "/bin/sh -c 'SECRET=`uuidgen` gunicorn -w 2 --chdir /home/chall/service/ro --bind
 0.0.0.0:6666 app:app'"
to_the_moon     "/usr/sbin/xinetd -dontfork"
oly_consensus   "/home/chall/service/ro/oly /home/chall/service/ro/consensus.olc"
swiss_keys      "./keys.ch server --port 6666"
gopher_coin     "/home/chall/service/ro/gopher_coin"
buffalo         "/bin/sh -c ./ro/start.sh"
```

Since it is running within in the Virtual Box, and in order to be able to actually interact with it from my host machine, I would need to do port forwarding from port `10001` to `10007` on the NAT mode. After that set I could get the following result showing the ports are mapped.

```
shan@drogo:~$ nmap -vv -Pn 127.0.0.1 -p10001-10007

Starting Nmap 7.60 ( https://nmap.org ) at 2021-12-21 20:05 EST
Initiating Connect Scan at 20:05
Scanning localhost (127.0.0.1) [7 ports]
Discovered open port 10006/tcp on 127.0.0.1
Discovered open port 10004/tcp on 127.0.0.1
Discovered open port 10007/tcp on 127.0.0.1
Discovered open port 10002/tcp on 127.0.0.1
Discovered open port 10003/tcp on 127.0.0.1
Discovered open port 10001/tcp on 127.0.0.1
Discovered open port 10005/tcp on 127.0.0.1
Completed Connect Scan at 20:05, 0.00s elapsed (7 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up, received user-set (0.000081s latency).
Scanned at 2021-12-21 20:05:57 EST for 0s

PORT      STATE SERVICE      REASON
10001/tcp open  scp-config   syn-ack
10002/tcp open  documentum   syn-ack
10003/tcp open  documentum_s syn-ack
10004/tcp open  emcrmirccd   syn-ack
10005/tcp open  stel         syn-ack
10006/tcp open  netapp-sync  syn-ack
10007/tcp open  mvs-capacity syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

In order to confirm the result, I could use `telnet` to interact with the services are exposed via the host machine.

```
shan@drogo:~$ telnet 127.0.0.1 10001
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
$$\  $$$$$$\ $$$$$$$$\ $$$$$$$$\        $$$$$$\   $$$$$$\   $$$$$$\    $$\
\__|$$  __$$\\__$$  __|$$  _____|      $$  __$$\ $$$ __$$\ $$  __$$\ $$$$ |
$$\ $$ /  \__|  $$ |   $$ |            \__/  $$ |$$$$\ $$ |\__/  $$ |\_$$ |
$$ |$$ |        $$ |   $$$$$\           $$$$$$  |$$\$$\$$ | $$$$$$  |  $$ |
$$ |$$ |        $$ |   $$  __|         $$  ____/ $$ \$$$$ |$$  ____/   $$ |
$$ |$$ |  $$\   $$ |   $$ |            $$ |      $$ |\$$$ |$$ |        $$ |
$$ |\$$$$$$  |  $$ |   $$ |            $$$$$$$$\ \$$$$$$  /$$$$$$$$\ $$$$$$\
\__| \______/   \__|   \__|            \________| \______/ \________|\______|

Welcome to iCTF 2021 meme stock trading simulator, you can trade meme stocks here for your whole
life!
Are you a new player? (y/n)
```

Step 2. Information Collection

By following the port number, I started with the first challenge `to_the_moon`. In the folder there

are severl files listed:

```
.
├── append [error opening dir]
├── Dockerfile
├── ro
│   └── to_the_moon
├── rw
├── src
│   ├── game.rs
│   └── main.rs
└── xinetd
```

Assuming that the `to_the_moon` should be the binary file for the challenge, which is written in Rust as indicated in the source file. And the `xinetd` file should be the one used by the docker container to init the service. But for now just move on to exec the binary file. In order to get with a clearer debug info, I export the `RUST_BACKTRACE=1` flag. And first round of executing the binary indeed encountered with main thread panic error:

```
ubuntu@vulnbox:/opt/ictf/services/to_the_moon/ro$ export RUST_BACKTRACE=1
ubuntu@vulnbox:/opt/ictf/services/to_the_moon/ro$ ./to_the_moon
$$\  $$$$$$\ $$$$$$$$\ $$$$$$$$\        $$$$$$\   $$$$$$\   $$$$$$\    $$\
\__|$$  __$$\\__$$  __|$$  _____|      $$  __$$\ $$$ __$$\ $$  __$$\ $$$$ |
$$\ $$ /  \__|  $$ |   $$ |            \__/  $$ |$$$$\ $$ |\__/  $$ |\_$$ |
$$ |$$ |        $$ |   $$$$$\           $$$$$$  |$$\$$\$$ | $$$$$$  |  $$ |
$$ |$$ |        $$ |   $$  __|         $$  ____/ $$ \$$$$ |$$  ____/   $$ |
$$ |$$ |  $$\   $$ |   $$ |            $$ |      $$ |\$$$ |$$ |        $$ |
$$ |\$$$$$$  |  $$ |   $$ |            $$$$$$$$\ \$$$$$$  /$$$$$$$$\ $$$$$$\
\__| \______/   \__|   \__|            \________| \______/ \________|\______|

Welcome to iCTF 2021 meme stock trading simulator, you can trade meme stocks here for your whole
life!
Are you a new player? (y/n)
n
Where is your save file?
aaaac
Password?
aaaac
thread 'main' panicked at 'there is something wrong with the RW directory: Os { code: 2, kind: No
tFound, message: "No such file or directory" }', src/game.rs:172:30
stack backtrace:
   0: rust_begin_unwind
             at /rustc/e249ce6b2345587d6e11052779c86adbad626dff/library/std/src/panicking.rs:495:
5
   1: core::panicking::panic_fmt
             at /rustc/e249ce6b2345587d6e11052779c86adbad626dff/library/core/src/panicking.rs:106
:14
   2: core::result::unwrap_failed
             at /rustc/e249ce6b2345587d6e11052779c86adbad626dff/library/core/src/result.rs:1617:5
   3: to_the_moon::game::Game::get_save_path
   4: to_the_moon::game::Game::load_state
   5: to_the_moon::game::Game::login
   6: to_the_moon::main
note: Some details are omitted, run with `RUST_BACKTRACE=full` for a verbose backtrace.
```

From the backtrace, it shows that the error starts from here

`3: to_the_moon::game::Game::get_save_path` therefore I dived into the source code `game.rc` :

```
169     fn get_save_path(&self, save_file: &str) -> PathBuf {
170         let mut path = PathBuf::new();
171         let dir = fs::canonicalize(RW_DIR)
172                             .expect("there is something wrong with the RW directory");
173         path.push(dir);
174         path.push(save_file);
175         path
176     }
```

Here the error info is raised at line 172, which the `RW_DIR` is passed into the `canonicalize` function, in Rust documentation `canonicalize` defines

> Returns the canonical, absolute form of a path with all intermediate components normalized and symbolic links resolved.

```
pub fn canonicalize<P: AsRef<Path>>(path: P) -> Result<PathBuf>
```

Therefore tracing the `RW_DIR` should help me find the problem. In source code `RW_DIR` defines at

```
 16 const RW_DIR: &str = "/home/chall/service/rw/";
```

However in the folder there is no such path, that's reason that makes `canonicalize` to raise the execption.

```
ubuntu@vulnbox:/opt/ictf/services/to_the_moon$ ls /home/chall/service/rw/
ls: cannot access '/home/chall/service/rw/': No such file or directory
```

Then I went back to the files and check again on the resouces to see what I may missed. There is a Xinetd config file, Xinetd so called super-server is which could be configured to listen in many services.

```
service to_the_moon
{
        disable = no
        type = UNLISTED
        wait = no
        server = /home/chall/service/ro/to_the_moon
        socket_type = stream
        protocol = tcp
        user = chall
        port = 6666
        flags = REUSE
        per_source = UNLIMITED
        rlimit_cpu = 2
        nice = 18
}

xinetd (END)
```

Then another file named Dockerfile is the one I missed, which indicates the previous guess is correct - we do need to init the xinet deamon to be able to play this challenge, also add a user named chall's home directory with flag `-m`

```
from ubuntu:20.04

run apt-get update
run apt-get install -y xinetd
run useradd -m chall -u 31337

copy xinetd /etc/xinetd.d/to_the_moon
cmd ["/usr/sbin/xinetd", "-dontfork"]
Dockerfile (END)
```

However during the setup I found out the built-in-box user `ctf` has the same uid 31337, which might indicates that the `ctf` in the box could be the `chall` mentioned in the source code. Therefore I went ahead removed user `ctf` and changed to `chall` also need to create the folder matching with the hard-coded path.

```
ubuntu@vulnbox:/opt/ictf/services/to_the_moon$ sudo awk -F: '($3 >= 1000) {printf "%s:%s\n",$1,$3
}' /etc/passwd
nobody:65534
ubuntu:1000
chall:31337
```

Since the home directory chall has the permission rw set only for the user itself, therefore need to switch to chall to play the challenge. Otherwise will get permission error msg.

```
thread 'main' panicked at 'Unable to write serialized data to game save!: Os { code: 13, kind: Pe
rmissionDenied, message: "Permission denied" }', src/game.rs:182:14
```

After the modification, the binary finally works

```
chall@vulnbox:/opt/ictf/services/to_the_moon/ro$ ./to_the_moon
$$\  $$$$$$\ $$$$$$$$\ $$$$$$$$\        $$$$$$\   $$$$$$\   $$$$$$\    $$\
\__|$$  __$$\\__$$  __|$$  _____|      $$  __$$\ $$$ __$$\ $$  __$$\ $$$$ |
$$\ $$ /  \__|  $$ |   $$ |            \__/  $$ |$$$$\ $$ |\__/  $$ |\_$$ |
$$ |$$ |        $$ |   $$$$$\           $$$$$$  |$$\$$\$$ | $$$$$$  |  $$ |
$$ |$$ |        $$ |   $$  __|         $$  ____/ $$ \$$$$ |$$  ____/   $$ |
$$ |$$ |  $$\   $$ |   $$ |            $$ |      $$ |\$$$ |$$ |        $$ |
$$ |\$$$$$$  |  $$ |   $$ |            $$$$$$$$\ \$$$$$$  /$$$$$$$$\ $$$$$$\
\__| \______/   \__|   \__|            \________| \______/ \________|\______|

Welcome to iCTF 2021 meme stock trading simulator, you can trade meme stocks here for your whole
life!
Are you a new player? (y/n)
y
Password?
dsadsa
Your game save file is at: L9j5rQIEUr3GBdBG6iyNret9x5Wq9b
================================
At the age of 0, what do you want to do?
1. Pass Time.
2. Write Diary.
```

```
3. Read Diary.
4. Go to Carnival.
126. Show Assets.
127. Save the game and quit.
```

It uses md5 hash to encode the password as hash and stores into the file and using `rand_gen` to generate 31 bits long random value using `PCG64` aka `Permuted Congruential Generator` as the file name.

```
chall@vulnbox:~/service/rw$ ls -l
total 8
-rw-rw-r-- 1 chall chall 237 Dec 22 04:00 EWyDAXiFXKSdDIS94gy18U5UZCkvzl
-rw-rw-r-- 1 chall chall 238 Dec 22 03:39 L9j5rQIEUr3GBdBG6iyNret9x5Wq9b
chall@vulnbox:~/service/rw$ cat EWyDAXiFXKSdDIS94gy18U5UZCkvzl
{"age":0,"hash":"202cb962ac59075b964b07152d234b70","save_file":"EWyDAXiFXKSdDIS94gy18U5UZCkvzl","
promise":false,"events":[],"stock_prices":[204,39,10,137],"diary":"","trans":true,"hype":58,"acci
dent":29,"anya":59,"dark":61,"double":true}
```

Everytime when execute the binary it will challenge and ask for the password from user input and then compute the md5 hash, it will then compare this new hash with user previous saved hash.

```
529  println!("Password?");
530  password = read_line();
531  let digest = md5::compute(password);
532  let hash = format!("{:x}", digest);
533  let game = self.load_state(&save_file);
534
535  if hash.ne(&game.hash) {
536       println!("Wrong password!");
537       process::exit(-1);
538   }
```

Step3. Finding the Piececs

When I tried to interact with the binary, I found out the `age` value will add 1 after each choice from the menu. Then finding the souce code found out it's a 8 bit unsigned int object and has a default value 0.

```
 55 #[derive(Serialize, Deserialize, Clone)]
 56 struct StockEvent {
 57     action: String,
 58     stock_id: usize,
 59     share: u32,
 60     price: u32,
 61     age: u8,
 62 }
```

But I didn't follow the same logic to try every operation, instead I changed age value in the file with a random number, then I found there are hidden menu appearing, which are exactly what I wanted.

```
Welcome to iCTF 2021 meme stock trading simulator, you can trade meme stocks here for your whole
life!
```

```
Are you a new player? (y/n)
n
Where is your save file?
/home/chall/service/rw/Qv5smlLyr7stc6zc9MPA62btbCsp6v
Password?
123
Your game is reloaded, enjoy!
================================
At the age of 33, what do you want to do?
1. Pass Time.
2. Write Diary.
3. Read Diary.
4. Go to Carnival.
5. Part time job at uncle Sam's grocery store.
6. Trade some meme stocks, how exciting!
126. Show Assets.
127. Save the game and quit.
7
Yo, I'm DeepFuckingValue.
You don't like GameStop? There is not much to talk with you.
```

Then I tried with 7 which is not on the menu, but I got a response, then I used this to find back to the source code.

```
323     fn transaction(&mut self) {
324         let mut good = false;
325         let mut msg: &str = "";
326         let hold = self.get_share(1);
327         if self.age < self.hype {
328             msg = "Who are you?";
329         } else if !self.trans {
330             msg = "You are being too greedy. No shares for you!";
331         } else if hold == 0 {
332             msg = "You don't like GameStop? There is not much to talk with you.";
333         } else {
334             good = true;
335         }
336         println!("Yo, I'm DeepFuckingValue.");
337         if !good {
338             println!("{}", msg);
339             return;
340         }
```

Then I found out there is comparison between `age` and `self.hype`, then I searched `self.hype` as the keyword I found out this, which is the real hidden menu I assume

```
624     fn one_year(&mut self) {
625         println!("================================");
626         if self.age == self.accident {
627             self.handle_accident();
628         } else if self.age == self.hype {
629             self.handle_hype();
630         } else if self.age == self.anya {
631             self.handle_anya();
632         } else if self.age == self.dark {
633             self.handle_dark();
634         } else if self.age == 100 {
```

```
635             self.dream();
636         } else {
637             let choice: i8 = self.action_menu();
638             if choice < 0 {return;}
639             match choice {
640                 1 => {self.pass_time();}
641                 2 => {self.write_diary();}
642                 3 => {self.read_diary();}
643                 4 => {self.carnival();}
644                 5 => {self.part_time();}
645                 6 => {self.stock();}
646                 7 => {self.transaction();}
647                 126 => {self.show_assets();}
648                 127 => {
649                     self.save_state();
650                     process::exit(0);
651                 }
652                 _ => {
653                     println!("Unexpected Value");
654                     process::exit(-1);
655                 }
656             }
657         }
658         self.age += 1;
659     }
```

And it is the real menu since the option 7 leads to the `fn.transcation` which is the hidden menu does not showing on the prompt. And since the `choice` has `i8` type which is the signed interger and it's rage is from -127 to 128, however the program handles the exeption in case there is integer overflow. From there isn't much interesting findings, so I moved to trying other values indicates in the file.

```
Welcome to iCTF 2021 meme stock trading simulator, you can trade meme stocks here for your whole
life!
Are you a new player? (y/n)
n
Where is your save file?
/home/chall/service/rw/Qv5smlLyr7stc6zc9MPA62btbCsp6v
Password?
123
Your game is reloaded, enjoy!
================================
A horrible car accident happened.
You smell the odor of dead animal and it stays in your memory.
To forget the event, you take too many beta-blockers and you lose your memory...
================================
Hold the line and do not sell.
GameStop to the moon!
================================
At the age of 14, what do you want to do?
1. Pass Time.
2. Write Diary.
3. Read Diary.
4. Go to Carnival.
5. Part time job at uncle Sam's grocery store.
126. Show Assets.
127. Save the game and quit.
```

This time seems getting close to the hint `moon` but then back to the source code does not finding anything interesting regarding the function `handle_accident`

```
551     fn handle_accident(&mut self) {
552         println!("A horrible car accident happened.");
553         println!("You smell the odor of dead animal and it stays in your memory.");
554         println!("To forget the event, you take too many beta-blockers and you lose your memo
ry...");
555         self.events = Vec::new();
556         self.save_state();
557     }
```

I went back to the file and checked again the output:

```
{"age":12,"hash":"202cb962ac59075b964b07152d234b70","save_file":"Qv5smlLyr7stc6zc9MPA62btbCsp6v",
"promise":false,"events":[],"stock_prices":[204,39,10,137],"diary":"","trans":true,"hype":13,"acc
ident":12,"anya":38,"dark":78,"double":true}
```

Then I found out there is this data named `doube` which has the value `true` I searched this info in the source code and I found something interesting finally:

```
396         match self.double {
397             true => {
398                 println!("Today is your lucky day!");
399                 let event = StockEvent {action: "recv".to_string(),
400                                                 stock_id: 1,
401                                                 share: to_send*2,
402                                                 price: self.stock_prices[1],
403                                                 age: self.age};
404                 self.events.push(Event::Stock(event));
405             }
406             false => {
407                 println!("There is something wrong with my account, I can't send you the shar
es, sorry.");
408                 println!("The shares that you sent me? Sorry, no refund.");
409             }
410         }
```

As long as the `double` value is true then I could get the chance to double my stock share as indicated in the code. I didn’t know how does this work untill I saw the upper part of the function:

```
323     fn transaction(&mut self) {
324         let mut good = false;
325         let mut msg: &str = "";
326         let hold = self.get_share(1);
327         if self.age < self.hype {
328             msg = "Who are you?";
329         } else if !self.trans {
330             msg = "You are being too greedy. No shares for you!";
331         } else if hold == 0 {
332             msg = "You don't like GameStop? There is not much to talk with you.";
333         } else {
334             good = true;
335         }
336         println!("Yo, I'm DeepFuckingValue.");
```

```
        if !good {
            println!("{}", msg);
            return;
        }

        println!("Welcome to the dark web!");
        println!("I am giving back to my community due to Covid-19!");
        println!("All GameStock shares sent to my address below will be sent back doubled.");
        println!("If you send 1000 shares, I will send back 2000 shares!");

        println!("How many shares do you want to send?");
        let to_send = self.read_u32();
        if to_send >= 0x10000 {
            println!("Nah, you want too much from me.");
            println!("You are now blacklisted!");
            self.trans = false;
            return;
        }

        let balance = self.get_balance();
        let send_event;
        match to_send <= hold {
            false => {
                let mut good: bool = false;
                let to_buy = to_send - hold;
                if to_buy*self.stock_prices[1] <= balance {
                    good = true;
                    println!("You don't have enough shares, so you buy {} shares to send him
the shares.", to_buy);
                    let event = StockEvent {action: "buy".to_string(),
                                            stock_id: 1,
                                            share: to_buy,
                                            price: self.stock_prices[1],
                                            age: self.age};
                    self.events.push(Event::Stock(event));
                }
                if !good {
                    println!("You don't have enough cash to finish the transaction!");
                }
                send_event = StockEvent {action: "send".to_string(),
                                         stock_id: 1,
                                         share: to_send,
                                         price: self.stock_prices[1],
                                         age: self.age};
            }
            true => {
                good = true;
                send_event = StockEvent {action: "send".to_string(),
                                         stock_id: 1,
                                         share: to_send,
                                         price: self.stock_prices[1],
                                         age: self.age};
            }
        }
        if !good {
            println!("You don't have enough shares to send!");
            return;
        }
        self.events.push(Event::Stock(send_event));

```

```
396         match self.double {
397             true => {
398                 println!("Today is your lucky day!");
```

Based on the code: in order to enter the section which can have the deal with the dark market I need to match the following condtion:

- First: `age` needs to larger than the `self.hype`
- Second: `self.trans` needs to stay true
- Third: `hold` value needs to larger than 0
- Fourth: `good` value needs to be true

However, the case is that I need to give 1000 share will plus the price that will take a long time to match, then I decided to change the price to `0` and keep adding the age to over the `self.hype`, additionally buy a random amount of share that larger than 1000. In this way I managed to enter this code section:

```
At the age of 18, what do you want to do?
1. Pass Time.
2. Write Diary.
3. Read Diary.
4. Go to Carnival.
5. Part time job at uncle Sam's grocery store.
6. Trade some meme stocks, how exciting!
126. Show Assets.
127. Save the game and quit.
6
Which stock do you want to trade?
1. GameStop
2. AMC
3. BlackBerry
4. ForRiver
1
What do you want to do with it?
1. Buy
2. Sell
1
How many shares?
10000
================================
At the age of 19, what do you want to do?
1. Pass Time.
2. Write Diary.
3. Read Diary.
4. Go to Carnival.
5. Part time job at uncle Sam's grocery store.
6. Trade some meme stocks, how exciting!
126. Show Assets.
127. Save the game and quit.
7
Yo, I'm DeepFuckingValue.
Welcome to the dark web!
I am giving back to my community due to Covid-19!
All GameStock shares sent to my address below will be sent back doubled.
If you send 1000 shares, I will send back 2000 shares!
How many shares do you want to send?
```

Ideally I didn't follow the prompt but entered -1, and luckily I hit my target and get extra stock share back.

```
Welcome to the dark web!
I am giving back to my community due to Covid-19!
All GameStock shares sent to my address below will be sent back doubled.
If you send 1000 shares, I will send back 2000 shares!
How many shares do you want to send?
-1
Today is your lucky day!
================================
At the age of 20, what do you want to do?
1. Pass Time.
2. Write Diary.
3. Read Diary.
4. Go to Carnival.
5. Part time job at uncle Sam's grocery store.
6. Trade some meme stocks, how exciting!
126. Show Assets.
127. Save the game and quit.
126
Your balance : 768

Stock Shares
GameStop     : 10000
AMC          : 0
BlackBerry   : 0
ForRiver     : 0
================================
At the age of 21, what do you want to do?
1. Pass Time.
2. Write Diary.
3. Read Diary.
4. Go to Carnival.
5. Part time job at uncle Sam's grocery store.
6. Trade some meme stocks, how exciting!
126. Show Assets.
127. Save the game and quit.
```

However, this can not end the challenge, therefore I went back in the source code for searching. In the `handle_xxx` functons, there is one interesting left and I felt this should be the final game.

Step 4. Puting Pieces Together

In function called `handle_dark` there is magic value `0x13371337` appears and the description message kind of matching the hint in the officical website of ictf:

> Once you have reverse-engineered a service and developed your new `l33t` exploit, you'll need a list of teams to attack!

`l33t` which corresponds to lit, and in description message showing `liiitle` which indicates `lit`, so this should be the one to go for.

```
579     fn handle_dark(&mut self) {
580         println!("Yo, I'm kylebot, a hacker from the dark web that you stumbled upon the othe
r day.");
```

```
        println!("I heard that you wanted to hack an account very badly.");
        println!("I can help you hack it, in exchange for a liiittle compensation.");
        println!("What do you think? (y/n)");
        let buf = read_line();
        match buf.as_ref() {
            "y" => {
                let balance = self.get_balance();
                if balance < 0x13371337 {
                    println!("You can't afford my service.");
                    println!("Prepare more cash next time!");
                    return;
                }
                println!("Deal!");
                println!("Which accout do you want to hack?");
                let save_file= read_line();
                if save_file.contains(".") || !Path::new(&self.get_save_path(&save_file)).exi
sts() {
                    println!("Are you trying to hack me? No way!");
                    process::exit(-1);
                }
                let game = self.load_state(&save_file);
                println!("There you go. You can read his secret diary now!");
                self.diary = game.diary;
                let event = BalanceEvent {action: "sub".to_string(), amount: 0x13371337, age:
 self.age};
                self.events.push(Event::Balance(event));
            }
            _ => {
                println!("You don't like the offer?");
                println!("So sad. I'll help others if they want to hack your account then. Go
od luck.");
                return;
            }
        }
    }
```

Directly I thought I need to match the condiction that my balance need to over `0x13371337` which is 322376503 in decimal. And in order to get this I looked into the core function `get_balance()`. I saw there is `sub` function defined as `balance -= event.amount` then I had an idea that is firstly modifed the stock price in the output file as **negatice value** and then buy a huge amount of share and then sell the stock in order to get a **positive value** as my new balance.

```
    fn get_balance(&self) -> u32 {
        let mut balance: u32 = 0;
        for event in &self.events {
            match event {
                Event::Balance(event)  => {
                    match event.action.as_ref() {
                        "add" => {balance += event.amount;}
                        "sub" => {balance -= event.amount;}
                        _ => {
                            println!("Something is wrong!");
                            process::exit(-1);
                        }
                    }
                }
```

```
115                 Event::Stock(event) => {
116                     match event.action.as_ref() {
117                         "buy" => {balance -= event.price*event.share;}
118                         "sell" => {balance += event.price*event.share;}
119                         "send" | "recv" => {}
120                         _ => {
121                             println!("Something is wrong!");
122                             process::exit(-1);
123                         }
124                     }
125                 }
126             }
127         }
128         balance
129     }
```

However, this does not work, as the variable `price` as defined as usigned integer type:

```
55 #[derive(Serialize, Deserialize, Clone)]
56 struct StockEvent {
57     action: String,
58     stock_id: usize,
59     share: u32,
60     price: u32,
61     age: u8,
62 }
```

And after changing the value to **negative** will get Rust complain:

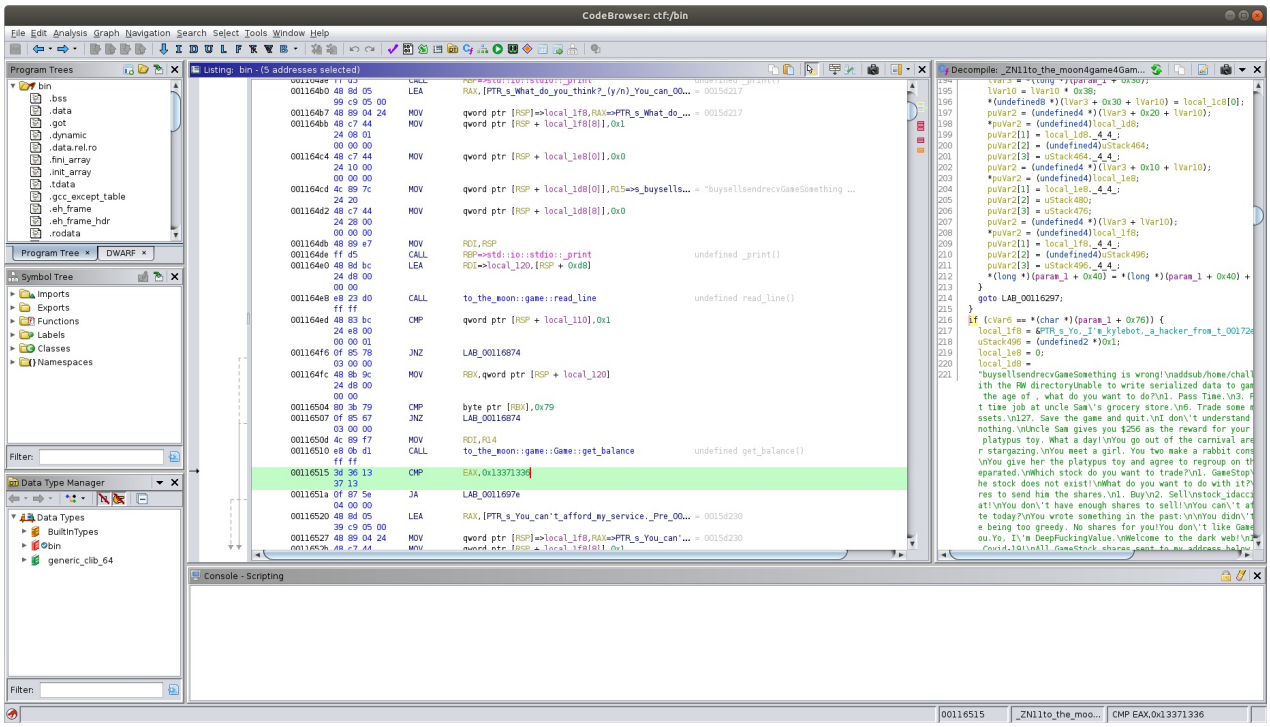
```
Welcome to iCTF 2021 meme stock trading simulator, you can trade meme stocks here for your whole
life!
Are you a new player? (y/n)
n
Where is your save file?
/home/chall/service/rw/Qv5smlLyr7stc6zc9MPA62btbCsp6v
Password?
123
thread 'main' panicked at 'called `Result::unwrap()` on an `Err` value: Error("invalid value: int
eger `-110`, expected u32", line: 1, column: 145)', src/game.rs:189:43
stack backtrace:
   0: rust_begin_unwind
             at /rustc/e249ce6b2345587d6e11052779c86adbad626dff/library/std/src/panicking.rs:495:
5
   1: core::panicking::panic_fmt
             at /rustc/e249ce6b2345587d6e11052779c86adbad626dff/library/core/src/panicking.rs:106
:14
   2: core::result::unwrap_failed
             at /rustc/e249ce6b2345587d6e11052779c86adbad626dff/library/core/src/result.rs:1617:5
   3: to_the_moon::game::Game::load_state
   4: to_the_moon::game::Game::login
   5: to_the_moon::main
note: Some details are omitted, run with `RUST_BACKTRACE=full` for a verbose backtrace.
chall@vulnbox:/opt/ictf/services/to_the_moon/ro$
```

Then I thought possibly I could use gdb to change the register value when it gets compared with `0x13371337`. But this failed as well, since by using the memory address showing in radare2 to

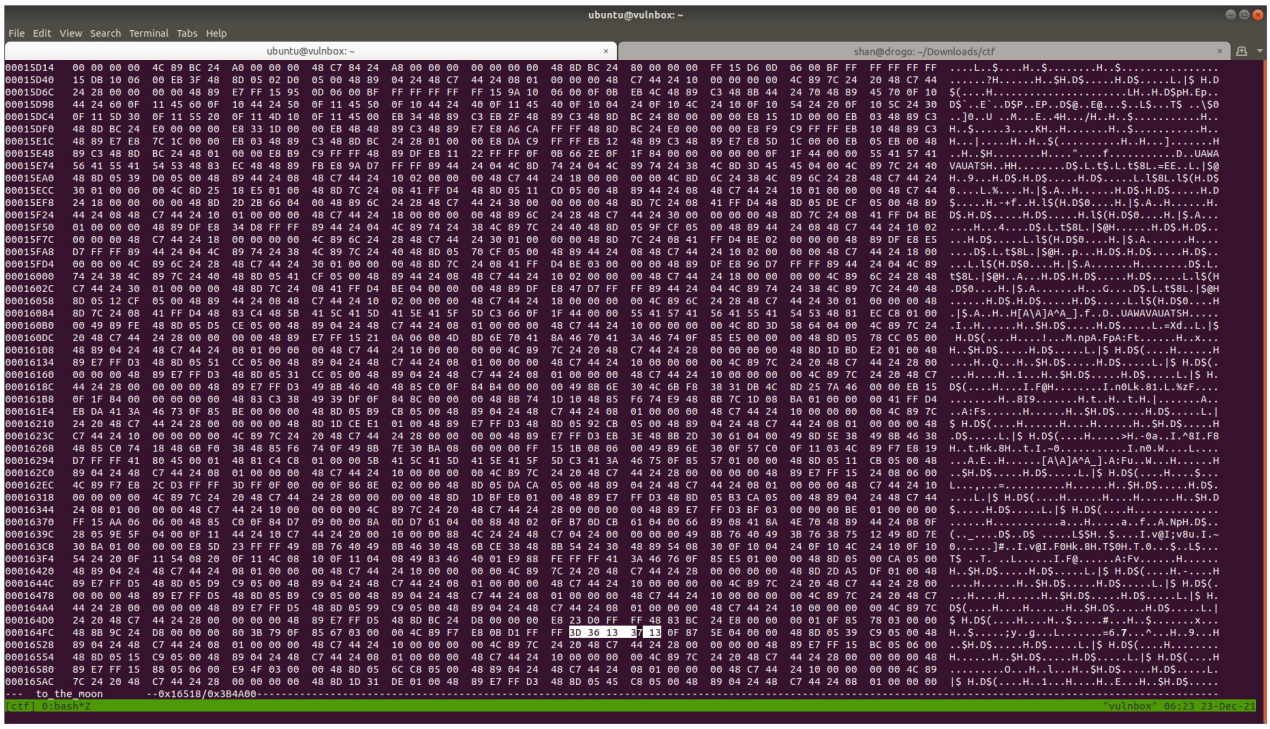
insert break point, it always complained as invalide memory address:

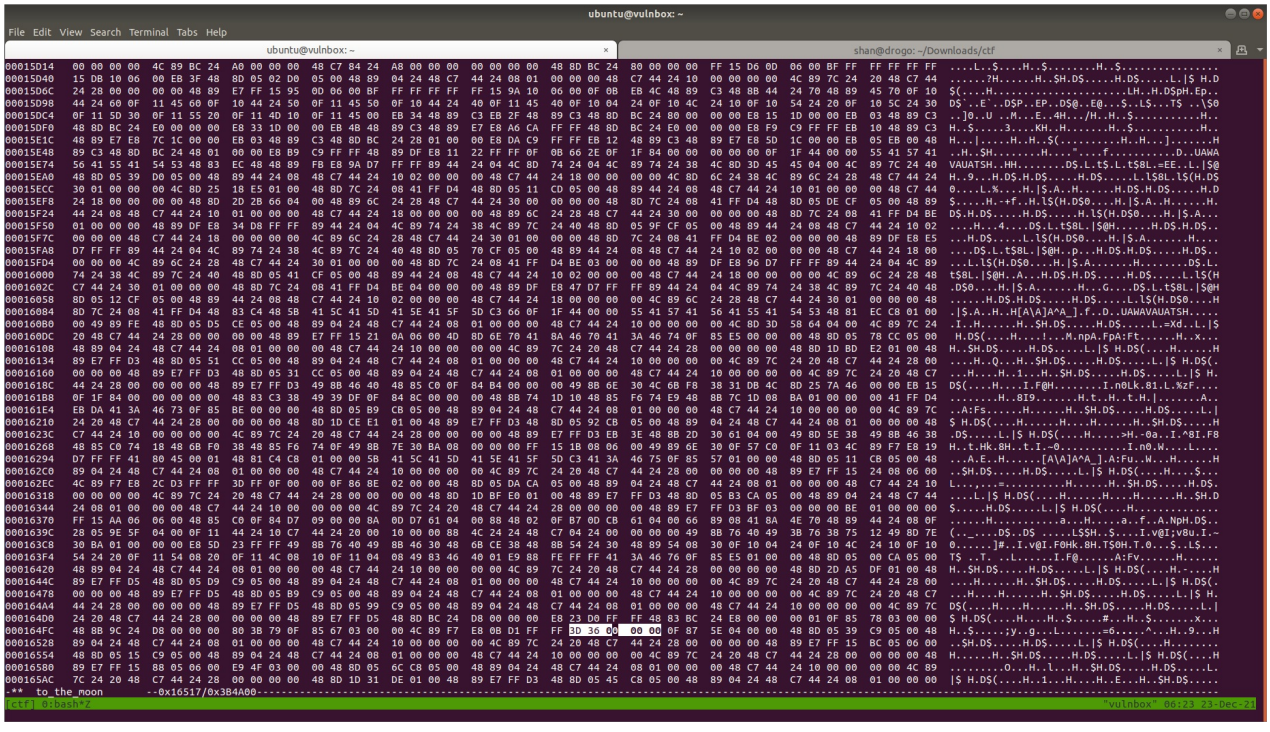
```
Reading symbols from to_the_moon...
gdb-peda$ break *0x000152c0
Breakpoint 1 at 0x152c0
gdb-peda$ run
Starting program: /opt/ictf/services/to_the_moon/ro/to_the_moon
Warning:
Cannot insert breakpoint 1.
Cannot access memory at address 0x152c0
```

Therefore I have to go to ghidra to check if the address is the same, it truns out it has the same virtual address. However, when I looked at ghida I found the break through point:



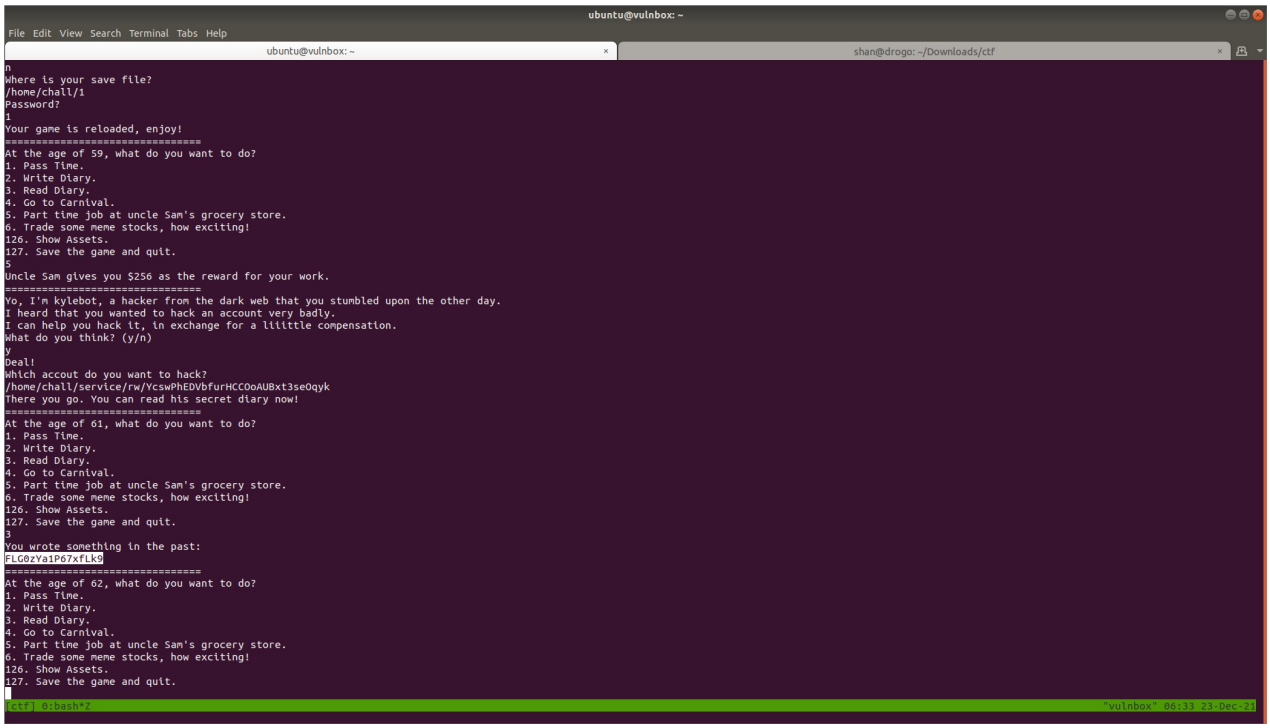
As showed in the picture, the magic value becomes `0x13371336`, and this really took me some time to find it. What is not a coincidence is that this magic value bascially matches the byte code `3d36131713` apart from the last `3d`. My guess is that the `3d` maybe is assembly code `cmp`. So I used hexedit to edit the binary file from `0x13371336` to `0x00000036` since I do want to keep `0x36` and it to compare





After saving the change, I went ahead and had a try. And most importantly: it WORKS! (before entering the `handle_dark()` section I need to use the menu to get some blance in order to compare with `0x36` and chaned the `age` value same as `dark` to match the condition). When these is all set, I could pass my target file (assuming it's on other team's machine) and get the secret `didary`.

Boom!



My thought: since this is just a ctf and the crack is simple, in reality a proper permission setting may block the attcker’s attempt to read. However, I found more challenging than the real bug exploit is that you have to find all the piceces in this game logic and select the correct order to crack it.