The Otter Scrolls (justCTF teaser 2024)

> by FluxFingers

Challenge

> Behold the ancient Spellbook, a tome of arcane wisdom where spells cast by mystical Otters weave the threads of fate. Embark on this enchanted journey where the secrets of the blockchain will be revealed. Brave adventurers, your quest awaits; may your courage be as boundless as the magic that guides you.

Challenge created by embe221ed & Darkstar49 from OtterSec

`nc tos.nc.jctf.pro 31337`

https://s3.cdn.justctf.team/a3cc5591-ad0a-47e5-bce0-78ff9bb7d2f3/tos_docker.tar.gz

Given is a docker compose file that spawns a server for the Sui smart contract platform. Additionally a template for the solution is given.

MOVE!

The given smart contract platform uses the Move programming language. It is very similar to Rust but with some additions.

In the challenge we are given a table (called dictionary in some other languages) of vectors:

```
        let mut all_words = table::new(ctx);

        let fire = vector[
            string::utf8(b"Blast"),
            string::utf8(b"Inferno"),
            string::utf8(b"Pyre"),
            string::utf8(b"Fenix"),
            string::utf8(b"Ember")
        ];

        let wind = vector[
            string::utf8(b"Zephyr"),
            string::utf8(b"Swirl"),
            string::utf8(b"Breeze"),
            string::utf8(b"Gust"),
            string::utf8(b"Sigil")
        ];

        let water = vector[
            string::utf8(b"Aquarius"),
            string::utf8(b"Mistwalker"),
            string::utf8(b"Waves"),
            string::utf8(b"Call"),
            string::utf8(b"Storm")
        ];

        let earth = vector[
            string::utf8(b"Tremor"),
            string::utf8(b"Stoneheart"),
            string::utf8(b"Grip"),
            string::utf8(b"Granite"),
            string::utf8(b"Mudslide")
        ];

        let power = vector[
            string::utf8(b"Alakazam"),
            string::utf8(b"Hocus"),
            string::utf8(b"Pocus"),
            string::utf8(b"Wazzup"),
            string::utf8(b"Wrath")
        ];

        table::add(&mut all_words, 0, fire);
        table::add(&mut all_words, 1, wind);
        table::add(&mut all_words, 2, water);
        table::add(&mut all_words, 3, earth);
        table::add(&mut all_words, 4, power);
```

We can then call `cast_spell` which takes a vector of integers. These integers are then used as indices into each of the vectors. The following conditions must be fulfilled:

```
        if (fire_word == string::utf8(b"Inferno")) {
            if (wind_word == string::utf8(b"Zephyr")) {
                if (water_word == string::utf8(b"Call")) {
                    if (earth_word == string::utf8(b"Granite")) {
                        if (power_word == string::utf8(b"Wazzup")) {
                            book.casted = true;
                        }
                    }
                }
            }
        }
```

Choosing the correct indices is simple and then just call the function:

```
    public fun solve(
        _spellbook: &mut theotterscrolls::Spellbook,
        _ctx: &mut TxContext
    ) {
        theotterscrolls::cast_spell(vector[1, 0, 3, 3, 3], _spellbook);
    }
```

The solution can then be build in the same docker container (`embe221ed/otter_template:latest`) as the server. And running it yields the correct flag `justCTF{Th4t_sp3ll_looks_d4ngerous...keep_y0ur_distance}`