


Bonjour

☰ Event	SEETF 2022
☰ Tags	Web
👤 Author	 KH Lai

Challenge Description



Challenge Walkthrough

The challenge gives us a instance to connect to. There are 4 options as shown below.


```
[1] - Create an account which will be used to deploy the challenge contract
[2] - Deploy the challenge contract using your generated account
[3] - Get your flag once you meet the requirement
[4] - Show the contract source code
```


The steps to complete the challenge and get the flag are as follow:


- Setup a MetaMask account. [\[Guide\]](#).
- Generate an account which will give us a deployer address and a token (will be used to verify whether the requirements have been fulfilled or not).
- Send 0.002 Ether to the deployer's address.
- Deploy the challenge contract which will give us the challenge contract's address and a transaction hash.
- Fulfill the challenge contract's requirements.
- Obtain the flag.

After setting up a MetaMask account and connecting to the SEETF Ethereum network, send 0.002 Ether to the deployer's address.

[← Edit](#)


 Account 1



 0x0D7...3480

New address detected! Click here to add to your address book.

SENDING ETH

 0.002

[EDIT](#)

Estimated gas fee ⓘ	0.000021 0.000021 ETH
	Max fee: 0.000021 ETH

Total	0.002021 0.002021 ETH
Amount + gas fee	Max amount: 0.002021 ETH

Reject

Confirm

After the transaction has been completed. We can deploy the challenge contract. It will ask for the token in order to verify whether the 0.002 Ether has been sent or not. The it will give us the challenge contract address and a transaction hash. Now we will have to fulfill the requirements in order to get the flag. Lets take a look at the challenge contract source code.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Bonjour {
    string public welcomeMessage;
    constructor() {
        welcomeMessage = "Bonjour";
    }

    function setWelcomeMessage(string memory _welcomeMessage) public {
        welcomeMessage = _welcomeMessage;
    }

    function isSolved() public view returns (bool) {
        return keccak256(abi.encodePacked("Welcome to SEETF")) == keccak256(abi.encodePacked(welcomeMessage));
    }
}
```

In the function `isSolved()`, it returns a boolean value. We have to make it returns true in order for us to get the flag. For it to return true, the encoded value of "Welcome to SEETF" have to be equals to the encoded value of the welcomeMessage which is "Bonjour". Both values goes through keccak256 hash algorithm. Now that we know the requirements, lets try to make the function return true.

We will be using the Remix IDE. First, we have to make sure that the environment that Remix is running on is **Injected Web3**. This option allows us to connect to the MetaMask browser extension. Once connected, Remix will automatically assign your Metamask account address.

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

Injected Web3 *Custom (1337) network*

ACCOUNT *+*

0xF5B...76b75 (29.975324 ETH) *Copy Edit*

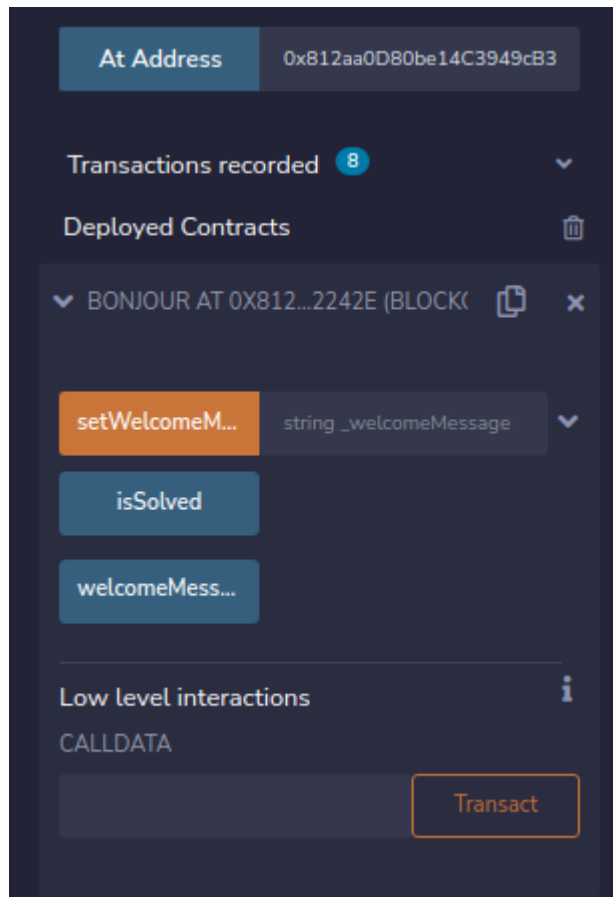
GAS LIMIT

3000000

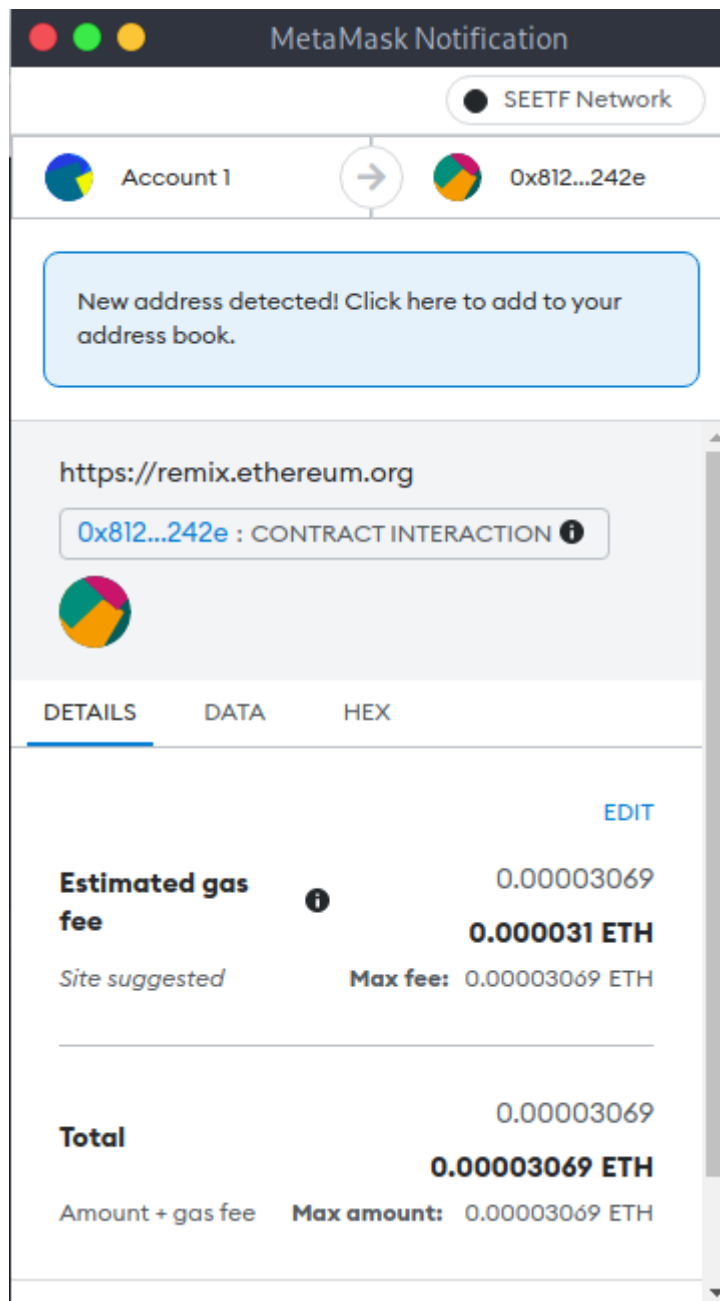
VALUE

0 Wei

Then we will load the challenge contract address. Once loaded, we can see the `setWelcomeMessage()` function and `isSolved()` function. To make it return true, we just have to set the welcome message to `"Welcome to SEETF"` and transact the data.



Deploying the data requires a gas fee. After the transaction is done. We can connect back into the challenge, insert the token and obtain the flag.



Flag

```
[+] flag: SEE{W3lc0mE_t0_SEETF_a71cda2f322e7834169418a9d1a036a0}
```