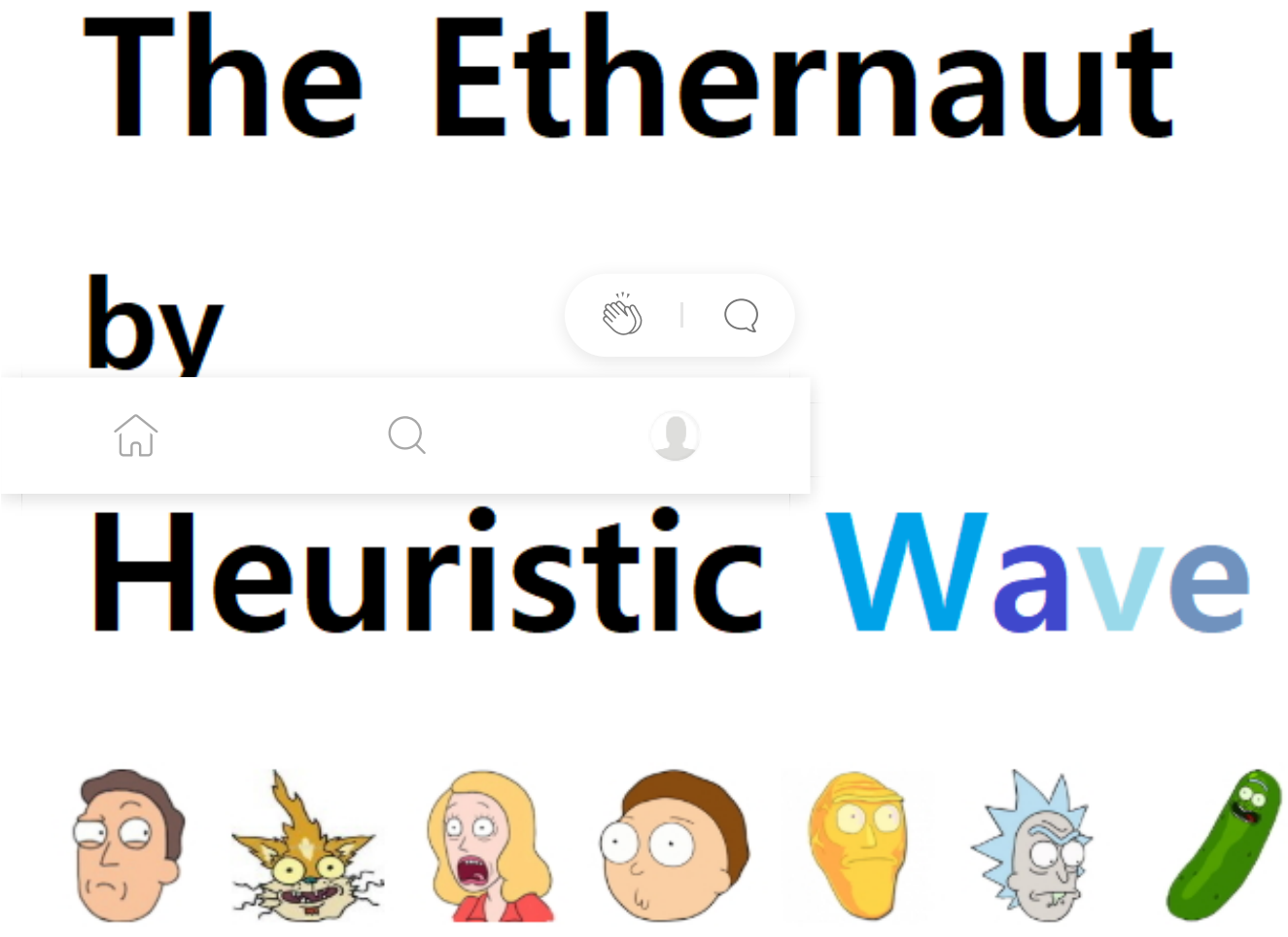
Heuristic Wave Follow
Nov 10, 2018 · 5 min read

Ethernaut Delegation Problem — 이더넛 6 단계 문제 해설

문제 해설에 들어가기 전, 이번 포스팅은 이더넛 내에서 콘솔창과 상호작용을 할 줄 알고 기본적인 리믹스 및 메타마스크 사용법이 숙지되어 있다는 가정 하에 해설을 진행합니다.



Delegation Problem

주어진 컨트랙트의 오너가 되어보는 문제다.

low level의 함수 delegatecall은 다른 컨트랙트의 함수를 호출하는 메소드다.
다른 컨트랙트의 함수를 사용하여 delegatecall이 쓰여진 내부 data를 수정할 수 있다.

어떤 역할을 하는지 궁금하다면 아래링크가 도움이 될 것이다.

http://ihpark92.tistory.com/54?category=747041

문제에 들어가기전 우리는 `Fallback methods`와 `Method ids`라는 힌트 키워드를 숙지하고 주어진 소스코드를 분석해보자!

```
pragma solidity ^0.4.18;

contract Delegate {

  address public owner;

  function Delegate(address _owner) public {
    owner = _owner;
  }

  function pwn() public {
    owner = msg.sender;
  }
}

contract Delegation {

  address public owner;
  Delegate delegate;

  function Delegation(address _delegateAddress) public {
    delegate = Delegate(_delegateAddress);
    owner = msg.sender;
  }

  function() public {
    if(delegate.delegatecall(msg.data)) {
      this;
    }
```

```
    }
}
```

주어진 코드를 살펴보면 오너 권한을 바꾸는 `pwn()` 함수가 눈에 들어온다. 또 Delegation 컨트랙트에서 `msg.data`를 주목하자. 여기까지만 읽고 감이 온다면, 그 감이 거의 맞을 것이다.

msg.data에는 함수의 시그니처를 넣을 수 있는데, 그 시그니처는 sha3해시 값의 첫 4byte를 의미한다. 즉, 우리는 코드를 만나기전에 힌트로 주어진 Delegation컨트랙트의 fallback함수를 호출하여 msg.data에 `Method ids`라는 힌트였던 pwn() 함수의 시그니처를 넣어 부른다면 Delegation 컨트랙트의 owner를 바꿀 수 있을 것이다.

이제 본격적으로 문제를 풀어보자!

리믹스에 주어진 소스코드를 복사한 후, 버전에 맞는 컴파일러를 지정하고 컴파일을 하자.

문제에서 주어진 Instance address를 리믹스에서 At Address에 넣어서 불러오면 Delegation컨트랙트가 생성된다. Delegate 컨트랙트가 아니라 Delegation이라는 점을 주목하자! 즉, 우리는 Delegation컨트랙트의 오너 권한을 탈취해야 한다.

이 후, 리믹스의 Compile 탭에서 Details 버튼을 눌러 아래와 같은 화면을 찾으면 pwn()함수의 FunctionHashes를 찾을 수 있다. 우리는 `dd365b8b`를 메타마스크 전송 시에 사용할 것이니 기억해 두자.

메타마스에서 대상에는 fallback 함수를 호출할 delegation컨트랙트의 CA주소가 들어가고 아까 알아둔 pwn()의functionhashes를 메타마스크의 `Hex Data` 부분에 0x를 붙여서 충분한 가스 수수료를 주고 트랜잭션을 발생시키자. 이후, Delegation의 owner가 Delegate의 pwn함수를 이용하게 한 나의 주소로 바뀌었다는 것을 확인하고 제출하면

다음단계로 통과하게 된다.

이번단계에서 delegatecall의 역할을 확실하게 알았다면 당신의 솔리디티 활용능력은 향상하게 될 것이다. delegatecall은 본래 이더리움 네트워크상에 존재하는 다른 컨트랙트의 함수를 재사용을 할 수 있다. 즉, 다른 언어에서 라이브러리를 활용하는 기능과 매우 유사하다.

다음 7단계 Force에서 만나요~