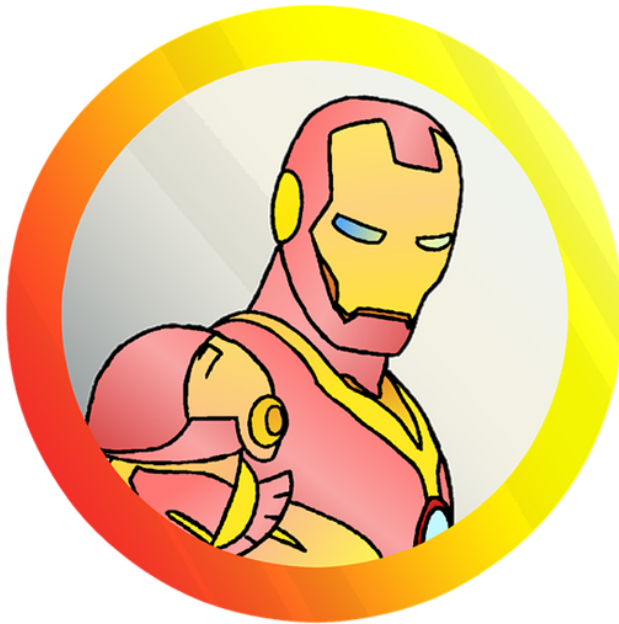


[Ethereum] Ethernaut 풀이 - 2.Fallout

modolee (47) ▾ (@modolee)in #kr-dev (/trending/kr-dev) • 4년 전



steemit
modolee

안녕하세요. 개발자 모도리입니다.

Ethernaut 문제 풀이 시리즈를 계속 진행해 보려고 합니다.

이번에는 Level 2 문제를 풀어보겠습니다. 지난 게시물은 아래를 확인해 주세요.

- Ethernaut 소개 (<https://steemit.com/kr-dev/@modolee/ethereum-ethernaut>)
- Ethernaut 풀이 - 0.Hello Ethenaut (<https://steemit.com/kr-dev/@modolee/ethereum-ethernaut-0-hello-ethernaut>)
- Ethernaut 풀이 - 1.Fallback (<https://steemit.com/kr->

2. Fallout

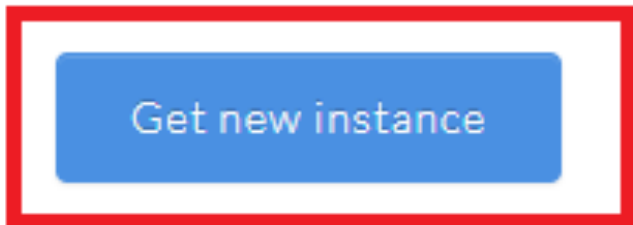
임무 확인 & 새로운 인스턴스 생성

Level 1 보다 쉬워 보입니다. 그냥 ownership만 가져오면 됩니다.

Claim ownership of the contract below to complete this level.

Things that might help

- Solidity Remix IDE



Get new instance 버튼을 눌러서 인스턴스를 생성합니다.



^^.js:59

Type help() for a listing of custom web3 addons

^^.js:116

^^.js:124

Annoying 'Slow network detected' message? Try Dev Tools settings -> User messages only or disable `chrome://flags/#enable-webfonts-intervention-v2`

=> Level address

^^.js:38

0x220beee334f1c1f8078352d88bcc4e6165b792f6

=> Player address

^^.js:38

0xf4690d756fa9dbf696da5c321b5260a6c48fbc6f

=> Ethernaut address

^^.js:38

0xc833a73d33071725143d7cf7dfd4f4bba6b5ced2

⓪ Requesting new instance from level...

^^.js:48

< < <<PLEASE WAIT>> > >

🔍 Sent transaction 🔍 <https://ropsten.etherscan.io/tx/0x5eb8b24...>

^^.js:134

🔍 Mined transaction 🔍 <https://ropsten.etherscan.io/tx/0x5eb8b24...>

^^.js:134

^^.js:31

▶ {tx: "0x5eb8b2468666093fbfaa4f3a84028e60d043d1a72ef711170ada9871ce66c5bc", receipt: {...}, Logs: Array(1)}

=> Instance address

^^.js:38

0x76e9b2a3c57451a9f2b7d9a80bfbdafaeb68b9e

Solidity 코드 분석

```

pragma solidity ^0.4.18;

import 'zeppelin-solidity/contracts/ownership/Ownable.sol';

contract Fallout is Ownable {

    mapping (address => uint) allocations;

    /* constructor */
    function Fallout() public payable {
        owner = msg.sender;
        allocations[owner] = msg.value;
    }

    function allocate() public payable {
        allocations[msg.sender] += msg.value;
    }

    function sendAllocation(address allocator) public {
        require(allocations[allocator] > 0);
        allocator.transfer(allocations[allocator]);
    }

    function collectAllocations() public onlyOwner {
        msg.sender.transfer(this.balance);
    }

    function allocatorBalance(address allocator) public view returns (uint) {
        return allocations[allocator];
    }
}

```

- ownership을 가져오려면 owner를 변경하는 부분을 찾아야 합니다.
- 생성자 부분에만 owner를 설정하는 부분이 있고, 나머지 코드에는 어디에도 owner를 설정하는 부분이 없습니다.
- 생성자는 컨트랙트를 최초 배포하는 Level 컨트랙트가 부를텐데, 어떻게 해야 되는걸까요?? (이미 이상한 부분은 찾으신 분들도 계시겠지만... 잠시만 기다려 주세요 ㅋㅋ)
- 현재 owner를 확인해 보면 역시나 Level 컨트랙트가 owner로 설정되어 있습니다.

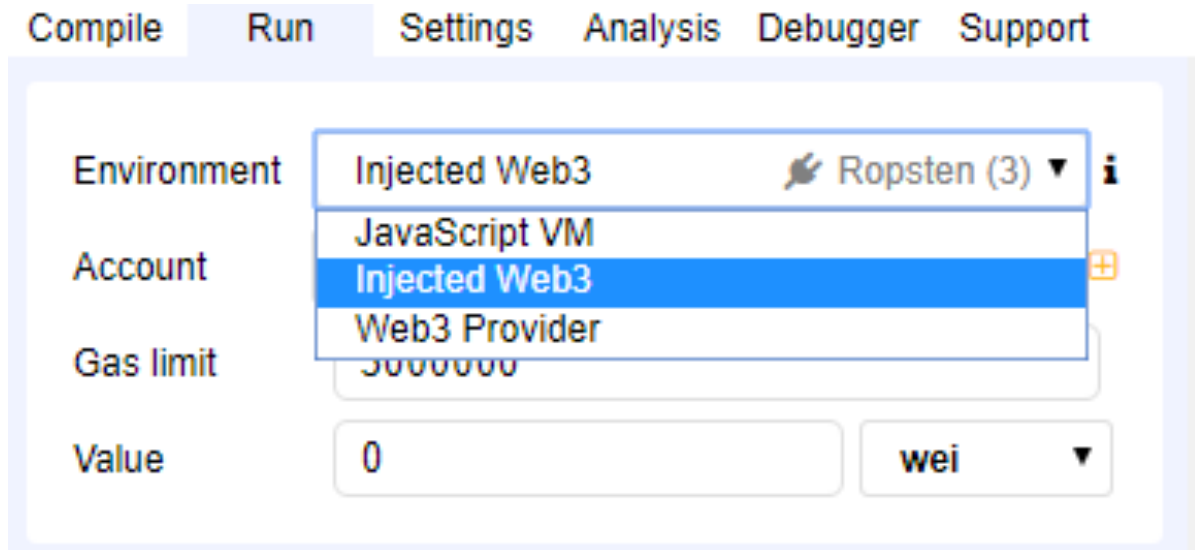
```

await contract.owner()
"0x220beee334f1c1f8078352d88bcc4e6165b792f6"
level
"0x220beee334f1c1f8078352d88bcc4e6165b792f6"

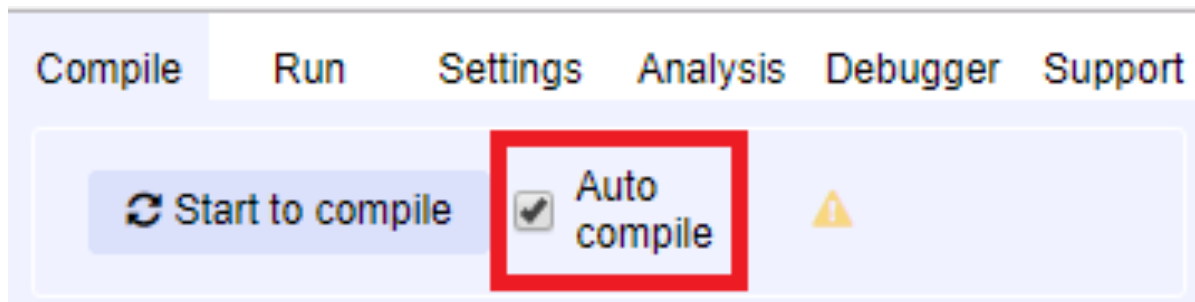
```

Remix 이용하기

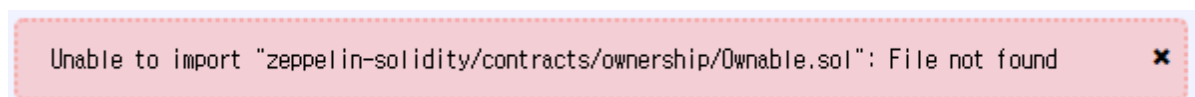
- 임무 확인 시 Solidity Remix IDE를 쓰면 도움이 될 거라는 문구를 봤습니다. 그러면 Remix에 소스 코드를 붙여 넣고 확인해 보겠습니다.
- 그 전에 Remix가 익숙하지 않으신 분들은 이 글 [\[링크\]](#)을 읽어 보시면 쉽게 사용하실 수 있습니다.
- 우리는 MetaMask를 사용하고 있으므로 Run 탭의 Environment를 Injected Web3를 선택합니다.



- 그리고 Remix 우측 상단의 + 버튼을 눌러서 새로운 파일을 만들고 Level 2 소스 코드를 전체 복사해서 붙여 넣습니다.
- Compile 탭으로 가서 **Auto compile** 옵션을 선택합니다.



- 그러면 **Unable to import "zeppelin-solidity/contracts/ownership/Ownable.sol": File not found** 라는 에러 메시지가 나옵니다.



- 해당 메시지가 나오는 이유는 방금 우리가 붙여 넣은 소스는 truffle framework를 이용해서 작성 된 코드로 로컬에 Ownable.sol 파일을 가지고 있으면서 컴파일 시 사용하던 코드입니다. 그런데 Remix에선 해당 파일

을 불러올 수 없으니 직접 코드를 붙여 넣어줘야 합니다. 해당 코드는 [openzeppelin](#)에서 가지고 오겠습니다.

- import 코드를 지워주고

```
pragma solidity ^0.4.18;
import 'zeppelin-solidity/contracts/ownership/Ownable.sol';
contract Fallout is Ownable {
    mapping (address => uint) allocations;

    /* constructor */
    function Fallout() public payable {
        owner = msg.sender;
        allocations[owner] = msg.value;
    }

    function allocate() public payable {
        allocations[msg.sender] += msg.value;
    }

    function sendAllocation(address allocator) public {
        require(allocations[allocator] > 0);
        allocator.transfer(allocations[allocator]);
    }

    function collectAllocations() public onlyOwner {
        msg.sender.transfer(this.balance);
    }

    function allocatorBalance(address allocator) public view returns (uint) {
        return allocations[allocator];
    }
}
```

- 그 자리에 pragma를 제외한 나머지 ownable 코드를 붙여 넣습니다.

```

pragma solidity ^0.4.18;

/**
 * @title Ownable
 * @dev The Ownable contract has an owner address, and provides basic authorizat
 * functions, this simplifies the implementation of "user permissions".
 */
contract Ownable {
    address public owner;

    event OwnershipRenounced(address indexed previousOwner);
    event OwnershipTransferred(
        address indexed previousOwner,
        address indexed newOwner
    );

    /**
     * @dev The Ownable constructor sets the original `owner` of the contract to t
     * account.
     */
    constructor() public {
        owner = msg.sender;
    }

    /**
     * @dev Throws if called by any account other than the owner.
     */
    modifier onlyOwner() {
        require(msg.sender == owner);
        _;
    }

    /**
     * @dev Allows the current owner to relinquish control of the contract.
     * @notice Renouncing to ownership will leave the contract without an owner.
     * It will not be possible to call the functions with the `onlyOwner`
     * modifier anymore.
     */
    function renounceOwnership() public onlyOwner {
        emit OwnershipRenounced(owner);
        owner = address(0);
    }
}

```

- 아직 warning 메시지가 남아 있습니다.

browser/Level2.sol:85:25: Warning: Using contract member "balance" inherited from the address*
 msg.sender.transfer(this.balance);
 ^-----^

- this.balance 코드에서 this를 address 타입으로 명시적 형변환을 해줘야 합니다.

```

function collectAllocations() public onlyOwner {
    msg.sender.transfer(address(this).balance);
}

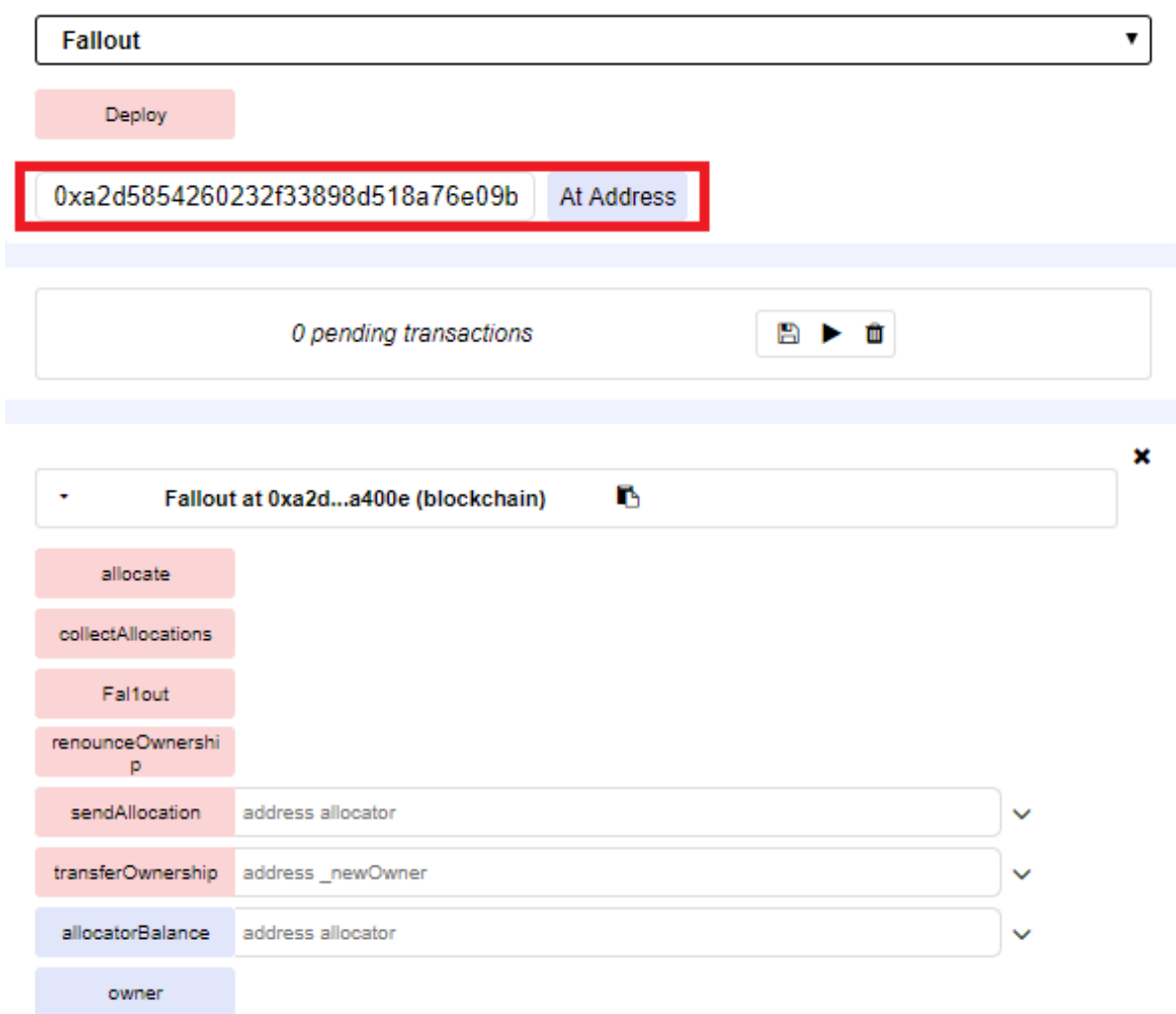
```

- 이제 컴파일이 무사히 됐습니다. Run탭으로 넘가면 컨트랙트를 Deploy할 수 있는 환경이 구축되었습니다.
- 하지만 우리는 직접 컨트랙트를 배포하는 것이 아니라 Level 컨트랙트가 배포해 놓은 컨트랙트를 불러서 사용해야 됩니다. (그래야 해당 컨트랙트의 상태가 변경되어 임무를 완수할 수 있으니깐요)
- 콘솔에서 Instance address를 확인합니다.


=> Instance address
0xa2d5854260232f33898d518a76e09ba23cda400e

^^.js:38

- 해당 주소를 복사해서 Remix의 Load contract from Address 창에 붙여 넣고 At Address 버튼을 눌러주면 해당 컨트랙트를 불러옵니다.




- 뭔가 이상한 함수가 하나 있습니다. (Fallout 인줄 알았는데, Fallout 이었네요... 낚였습니다.)

▼ **Fallout at 0xa2d...a400e (blockchain)** 

- allocate
- collectAllocations
- Fallout**
- renounceOwnership
- sendAllocation address allocator ▼
- transferOwnership address _newOwner ▼
- allocatorBalance address allocator ▼
- owner

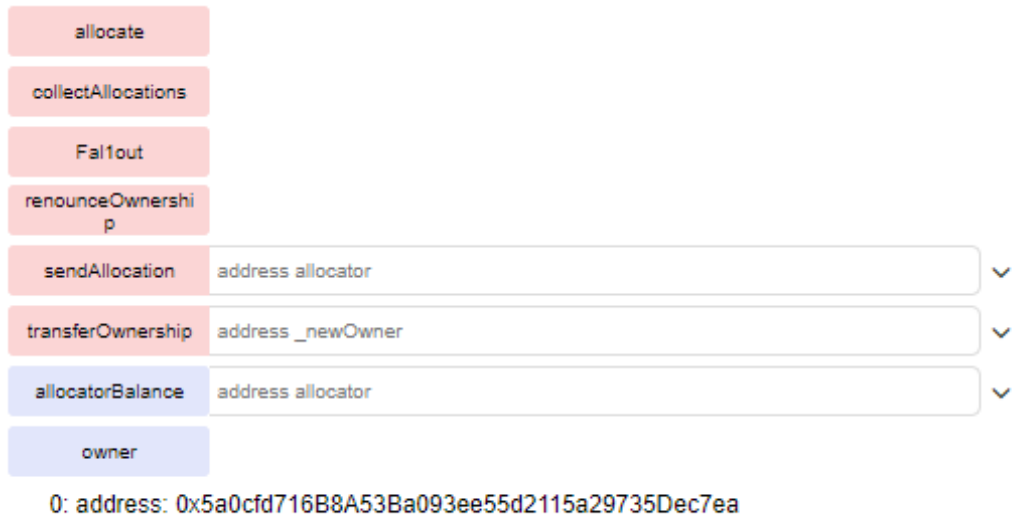
- 정상적인 생성자였다면 함수 목록에 표시되지 않습니다. Level 1 컨트랙트를 살펴 보시죠. (Fallback이라는 이름의 함수는 없습니다. (fallback)은 다른 함수 입니다.)

▼ **Fallback at 0xbf4...2a829 (blockchain)** 

- (fallback)
- contribute
- renounceOwnership
- transferOwnership address _newOwner
- withdraw
- contributions address
- getContribution
- owner

문제 풀이

- 트릭을 발견했으니, 문제는 굉장히 간단해 졌습니다. 그냥 Fal1out을 호출하면 owner가 될 수 있습니다.
- 호출하고 owner를 확인해 보겠습니다.
- Remix에서 확인

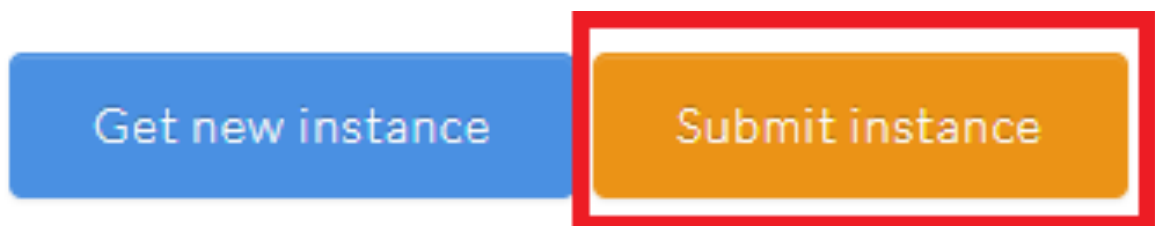


- Console에서 확인

```
> await contract.owner()
< "0x5a0cfd716b8a53ba093ee55d2115a29735dec7ea"
> player
< "0x5a0cfd716b8a53ba093ee55d2115a29735dec7ea"
```

답안 제출

- Submit instance 버튼을 누릅니다.



- 임무를 완수했습니다.



말하고자 하는 취약점

- Fallout은 Fal1out으로 잘못 쓰는 사소한 실수로 인해 컨트랙트의 ownership을 아무나 획득할 수 있게 되었습니다.
- 이런 경우가 종종 발생하여 Solidity 0.4.23 버전부터는 컨트랙트와 동일한 이름으로 생성자를 만드는 대신 constructor라는 별도의 이름으로 선언하게 업데이트 되었습니다.

- 취약점을 개선한 버전이 지속적으로 나오는 만큼, 컨트랙트 작성 시 최대한 최신 버전에 맞춰서 컨트랙트를 작성하는게 좋지 않을까 생각합니다.
- 그리고 역시 auditing이 필요하다~ 가 결론일 것 같네요.^^

오늘은 이것으로 마치겠습니다. 감사합니다.

[#kr \(/trending/kr\)](#)

[#ethereum \(/trending/ethereum\)](#)

[#solidity \(/trending/solidity\)](#)

[#ethernaut \(/trending/ethernaut\)](#)

🕒 4년 전 in [#kr-dev \(/trending/kr-dev\)](#) by

[modolee \(47\)](#) ▾ [\(/@modolee\)](#)

📈 📉 \$0.12 ▾ 8 보팅 ▾

🔗 [댓글 달기](#) | 💬 1

[\(/kr-dev/@modolee/ethereum-ethernaut-2-fallout\)](#)



정렬 순서: [Trending](#) ▾

[bible.com \(-7\)](#) ▾ [\(/@bible.com\)\(1\)](#) 4년 전 [\(/kr-dev/@bible.com/re-modolee-ethereum-ethernaut-2-fallout-20180620t013533140z#@bible.com/re-modolee-ethereum-ethernaut-2-fallout-20180620t013533140z\)](#) [-]

Get a free Bible for your phone, tablet, and computer. [bible.com](#) 🌐

📈 📉 \$0.00 [댓글 달기](#)