SOLIDITY SECURITY

And best practices

PRIVATE AND RANDOMNESS

- Everything publicly visible
  - Also local variables
  - And state variables marked as private
- Random numbers depend on miners
  - Be careful

RE-ENTRENCY

- Checks-Effects-Interactions pattern

```
pragma solidity ^0.4.11;


contract Fund {
    /// Mapping of ether shares of the contract.
    mapping(address => uint) shares;
    /// Withdraw your share.
    function withdraw() {
        var share = shares[msg.sender];
        shares[msg.sender] = 0;
        msg.sender.transfer(share);
    }
}
```

GAS LIMIT AND LOOPS

- Be careful with loops
- Block gas limit
  - Can only consume a certain amount of gas
- Does not apply to constant functions
  - Attention when called from other functions
- Be explicit in the contract docs

SENDING AND RECEIVING ETHER

- Receiving ether cannot be 100% prevented
  - Set the contract address as receiver when mining
  - Selfdestruct(contractaddress);
- Receiving ether without a specific function calls the fallback-function
  - If none available, throws an exception
  - Gas stipend only (2300 gas)
- Use add.transfer(…)

CALLSTACK DEPTH

- Maximum call-stack: 1024
- .send() returns false if callstack is depleted.
  - Same with other low-level functions
  - Call, callcode, deletegatecall

MSG.SENDER VS TX.ORIGIN

- Never use tx.origin for authorization

- A contract can be tricked into thinking you are authorized
  - Via a third party

MSG.SENDER VS. TX.ORIGIN

```
pragma solidity ^0.4.11;


// THIS CONTRACT CONTAINS A BUG - DO NOT USE
contract TxUserWallet {
    address owner;

    function TxUserWallet() {
        owner = msg.sender;
    }

    function transferTo(address dest, uint amount) {
        require(tx.origin == owner);
        dest.transfer(amount);
    }
}
```

MSG.SENDER VS. TX.ORIGIN

```
pragma solidity ^0.4.11;

interface TxUserWallet {

    function transferTo(address dest, uint amount);
}


contract TxAttackWallet {

    address owner;

    function TxAttackWallet() {
        owner = msg.sender;
    }

    function() {
        TxUserWallet(msg.sender).transferTo(owner, msg.sender.balance);
    }
}
```

KEY TAKE AWAYS

- Restrict the amount of ether or tokens „just to be safe“
- Keep it small and modular
- Check the docs frequently
  - They change frequently
- Use the Checks-Effects-Interactions Pattern
- Include a Fail-Safe Mode
  - Has any Ether leaked?
- Formal Verification

THANKS

- Questions?
  - Head over to the Q&A
  - And help each other if possible
- Feedback?
  - Always greatly appreciated
- Other questions?
  - Send us a message