卡亭科技
CHAITIN

EVM Opcode JOP

Jump Oriented Programming in Smart Contract Honeypot

Xiaohang (Sean) Yu at Chaitin Tech

Who Am I – Xiaohang (Sean) Yu

- Security Researcher at Chaitin Tech in Beijing, China.
- Focus on blockchain security
- Find 0days in blockchain projects, smart contracts, exchanges, wallets…
- JHU MSSI graduate
- OSCP & Pentester & CTFer
- First-time speaker at DEFCON

What is this about

- A (maybe) new type of smart contract honeypot
    - runtime code replacement
    - JOP (ROP on smart contract)
- A new smart contract CTF challenge type
- A good study case for Ethereum smart contract RE
- Etherscan verification process

Background

- Realworld CTF 2018
    - Online: Jul 2018, Final: Dec 2018
- Want to exam whether ROP is possible in smart contract

Outlines

- Puzzle Description
- Warming up
  - first glance
  - explore a little bit
- Reverse Engineering
  - byte-by-byte explanation
  - step-by-step stack/memory/storage transition
  - JOP and other honeypot techniques

卡亭科技 CHAITIN

Acoraida Monica Puzzle Description

The Story

"I exhausted all the words, couldn't describe her beauty", so many handsome charming men say that.

That girl is Acoraida Monica. Her cute comes from her intelligence. She only admires who are smarter than her.

So she designed this Q&A game and delegate someone to publish it on an Ethereum chain.

Anyone who solves her puzzle could win 1,000,000 ETH and ... maybe her heart. :P

卡亭科技 CHAITIN

Acoraida Monica Puzzle Description

Description

1. Web3 http provider to access the private chain (geth):
A geth RPC URL (https://192.168.10.100:8545)

2. Target contract address:
0xa9e63a4487f06aa51c2f815b4807b80b64363594

3. Player account:
0x19baa751d1092c906ac84ea4681fa91e269e6cb9

4. Query the source code of a specific contract address:
http://192.168.10.100:8081/<THE CONTRACT ADDRESS>

卡亭科技 CHAITIN

Acoraida Monica Puzzle Description

Goal

Drain the target contract, transfer ALL the ETH to your account, specifically:

1. The balance of *0x19baa7...6cb9* should become at least 1e24 wei (=1,000,000 eth)
2. The balance of *0xa9e63a...3594* should become 0.
3. If these two conditions achieved, the admin account will send the flag inside a transaction, which is from *0x492705c00090cb7c1cbb5ec3ab0b09f310dec399* to *0x19baa7...6cb9*.

卡奇科技 CHAITIN

Acoraida Monica Puzzle Description

Files provided

genesis.json

player-privkey.txt

AcoraidaMonicaGame.sol (the target contract)

Acoraida Monica Puzzle Description

Player: 200 ETH

POA, testing consensus

```
{
    ...
    "alloc": {
        ...,
        "0x19baa751d1092c906ac84ea4681fa91e269e6cb9": {
            "balance": "200000000000000000000"
        }
    },
    "clique":{
        "period": 1,
        "epoch": 30000
    }
}
```

genesis.json

Acoraida Monica Puzzle Description

```solidity
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    uint256 public version = 4;
    string public description = "Acoraida Monica admires smart guys, she'd like to pay
10000ETH to the one who could answer her question. Would it be you?";
    string public constant sampleQuestion = "Who is Acoraida Monica?";
    string public constant sampleAnswer = "$*!&#^[` a@.3;Ta&*T` R`<`~5Z`^5V You beat
me! :D";
    Logger public constant logger=Logger(0x5e351bd4247f0526359fb22078ba725a192872f3);
    address questioner;
    string public question;
    bytes32 private answerHash;

    constructor(bytes a) { ... }
    modifier onlyHuman{ ... }
    function Start(string _question, string _answer) public payable{ ... }
    function NewRound(string _question, bytes32 _answerHash) public payable{ ... }
    function TheAnswerIs(string _answer) onlyHuman public payable{ ... }
    function () payable {}
}
```

```solidity
contract Logger{
    event WeHaveAWinner(address);
    event NewQuestion(string);
    event NewAnswerHs(bytes32);
    function AcoraidaMonicaWantsToKeepALogOfTheWinner(address winner) public {
        emit WeHaveAWinner(winner);
    }
    function AcoraidaMonicaWantsToKnowTheNewQuestion(string _question) public{
        emit NewQuestion(_question);
    }
    function AcoraidaMonicaWantsToKnowTheNewAnswerHash(bytes32 _answerHash) public {
        emit NewAnswerHs(_answerHash);
    }
}
```

AcoraidaMonicaGame.sol

Acoraida Monica Puzzle Description

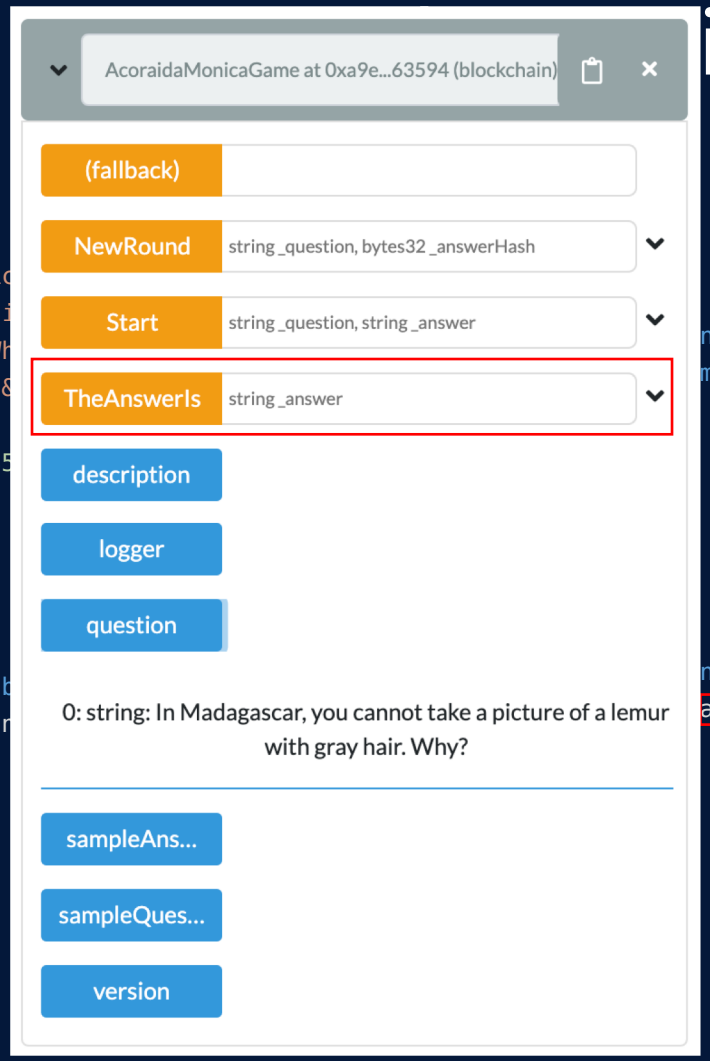
```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    uint256 public version = 4;
    string public description = "Acoraida Monica admires smart guys, she'd like to pay
10000ETH to the one who could answer her question. Would it be you?";
    string public constant sampleQuestion = "Who is Acoraida Monica?";
    string public constant sampleAnswer = "$*!&#^[` a@.3;Ta&*T` R`<`~5Z`^5V You beat
me! :D";
    Logger public constant logger=Logger(0x5e351bd4247f0526359fb22078ba725a192872f3);
    address questioner;
    string public question;
    bytes32 private answerHash;

    constructor(bytes a) { ... }
    modifier onlyHuman{ ... } function () payable {}
    function Start(string _question, string _answer) public payable{
        if(answerHash==0){
            answerHash = keccak256(_answer);
            question = _question;
            questioner = msg.sender;
        }
    }
```

```
    function NewRound(string _question, bytes32 _answerHash) public payable{
        if(msg.sender == questioner && msg.value >= 0.5 ether){
            require(_answerHash != keccak256(sampleAnswer));
            question = _question;
            answerHash = _answerHash;
            logger.AcoraidaMonicaWantsToKnowTheNewQuestion(_question);
            logger.AcoraidaMonicaWantsToKnowTheNewAnswerHash(_answerHash);
        }
    }
    function TheAnswerIs(string _answer) onlyHuman public payable{
        if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
            questioner = msg.sender;
            msg.sender.transfer(address(this).balance);
            logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
        }
    }
}
```

AcoraidaMonicaGame.sol

ica Puzzle Description

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    uint256 public version = 4;
    string public description = "Acoraida Monic
10000ETH to the one who could answer her questi
    string public constant sampleQuestion = "Wh
    string public constant sampleAnswer = "$*!&
me! :D";
    Logger public constant logger=Logger(0x5e35
    address questioner;
    string public question;
    bytes32 private answerHash;

    constructor(bytes a) { ... }
    modifier onlyHuman{ ... } function () payab
    function Start(string _question, string _ar
        if(answerHash==0){
            answerHash = keccak256(_answer);
            question = _question;
            questioner = msg.sender;
        }
    }
```




```
n NewRound(string _question, bytes32 _answerHash) public payable{
msg.sender == questioner && msg.value >= 0.5 ether){
 require(_answerHash != keccak256(sampleAnswer));
 question = _question;
 answerHash = _answerHash;
 logger.AcoraidaMonicaWantsToKnowTheNewQuestion(_question);
 logger.AcoraidaMonicaWantsToKnowTheNewAnswerHash(_answerHash);


n TheAnswerIs(string _answer) onlyHuman public payable{
answerHash == keccak256(_answer) && msg.value >= 1 ether){
 questioner = msg.sender;
 msg.sender.transfer(address(this).balance);
 logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
```

AcoraidaMonicaGame.sol

Acoraida Monica Puzzle Description

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    uint256 public version = 4;
    string public description = "Acoraida Monica admires smart guys, she'd like to pay
10000ETH to the one who could answer her question. Would it be you?";
    string public constant sampleQuestion = "Who is Acoraida Monica?";
    string public constant sampleAnswer = "$*!&#^[` a@.3;Ta&*T` R`<`~5Z`^5V You beat
me! :D";
    Logger public constant logger=Logger(0x5e351bd4247f0526359fb22078ba725a192872f3);
    address questioner;
    string public question;
    bytes32 private answerHash;

    constructor(bytes a) { ... }
    modifier onlyHuman{ ... } function () payable {}
    function Start(string _question, string _answer) public payable{
        if(answerHash==0){
            answerHash = keccak256(_answer);
            question = _question;
            questioner = msg.sender;
        }
    }

    function NewRound(string _question, bytes32 _answerHash) public payable{
        if(msg.sender == questioner && msg.value >= 0.5 ether){
            require(_answerHash != keccak256(sampleAnswer));
            question = _question;
            answerHash = _answerHash;
            logger.AcoraidaMonicaWantsToKnowTheNewQuestion(_question);
            logger.AcoraidaMonicaWantsToKnowTheNewAnswerHash(_answerHash);
        }
    }
    function TheAnswerIs(string _answer) onlyHuman public payable{
        if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
            questioner = msg.sender;
            msg.sender.transfer(address(this).balance);
            logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
        }
    }
}
```

AcoraidaMonicaGame.sol

Acoraida Monica Warming Up

```
> getTransactionsByAccount("0xa9e63a4487f06aa51c2f815b4807b80b64363594",0,100)
Searching for transactions to/from account "0xa9e63a4487f06aa51c2f815b4807b80b64363594" within blocks 0 and 100
Searching block 0
 tx hash         : 0x541b1ce0ba6f4a7711362268f2f69de6f6a5135d0191e2deff0172c5bede40df
  nonce          : 5
  blockHash      : 0x208995ec51dffdb0a865d76748a0c2eb06faba778da82bcb8194e1b391c6128a
  blockNumber    : 23
  transactionIndex: 0
  from           : 0x492705c00090cb7c1cbb5ec3ab0b09f310dec399
  to             : 0xa9e63a4487f06aa51c2f815b4807b80b64363594
  value          : 1e+24
  time           : 1564904313 Sun, 04 Aug 2019 07:38:33 GMT
  gasPrice       : 20000000000
  gas            : 90000
  input          :
0xc76de3e900000000000000000000000000000000000000000000000000000000000000400000000000000000000000000000000000000000000000000000000
0000000c000000000000000000000000000000000000000000000000000000000000048496e204d6164616761736361722c20796f752063616e6e6f742074616
b6520612070696374757265206f662061206c656d75722077697468206772617920686169722e205768793f000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000027596f75206e65656420612063616d65726120696e7374656164206f662067
72617920686169722e00000000000000000000000000000000000000000000000000
```

```
$ cat get_abi.py
from _pysha3 import keccak_256
import sys
def getabi(func):
    sha3_hash = keccak_256(func.encode('utf-8')).hexdigest()
    method_id = "0x"+sha3_hash[:8]
    return method_id
print(getabi('Start(string,string)'))
$ python get_abi.py
0xc76de3e9
```

Acoraida Monica Warming Up

```
> getTransactionsByAccount("0xa9e63a4487f06aa51c2f815b4807b80b64363594",0,100)
Searching for transactions to/from account "0xa9e63a4487f06aa51c2f815b4807b80b64363594" within blocks 0 and 100
Searching block 0
  tx hash          : 0x541b1ce0ba6f4a7711362268f2f69de6f6a5135d0191e2deff0172c5bede40df
   nonce           : 5
   blockHash       : 0x208995ec51dffdb0a865d76748a0c2eb06faba778da82bcb8194e1b391c6128a
   blockNumber     : 23
   transactionIndex: 0
   from            : 0x492705c00090cb7c1cbb5ec3ab0b09f310dec399
   to              : 0xa9e63a4487f06aa51c2f815b4807b80b64363594
   value           : 1e+24
   time            : 1564904313 Sun, 04 Aug 2019 07:38:33 GMT
   gasPrice        : 20000000000
   gas             : 90000
   input           :
0xc76de3e9000000000000000000000000000000000000000000000000000000000000004000000000000000000000000000000000000000000000000000000000000000
0000000c000000000000000000000000000000000000000000000000000000000000000048496e204d6164616761736361722c20796f752063616e6e6f742074616
b65206120706963747265206f662061206c656d7572207769746820677261792068616972e205768793f000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000027596f75206e65656420612063616d65726120696e7374656164206f662067
72617920686169722e000000000000000000000000000000000000000000000000000000000000
```

Start(string,string)

length

"In Madagascar, you cannot take a picture of a lemur with gray hair. Why?"

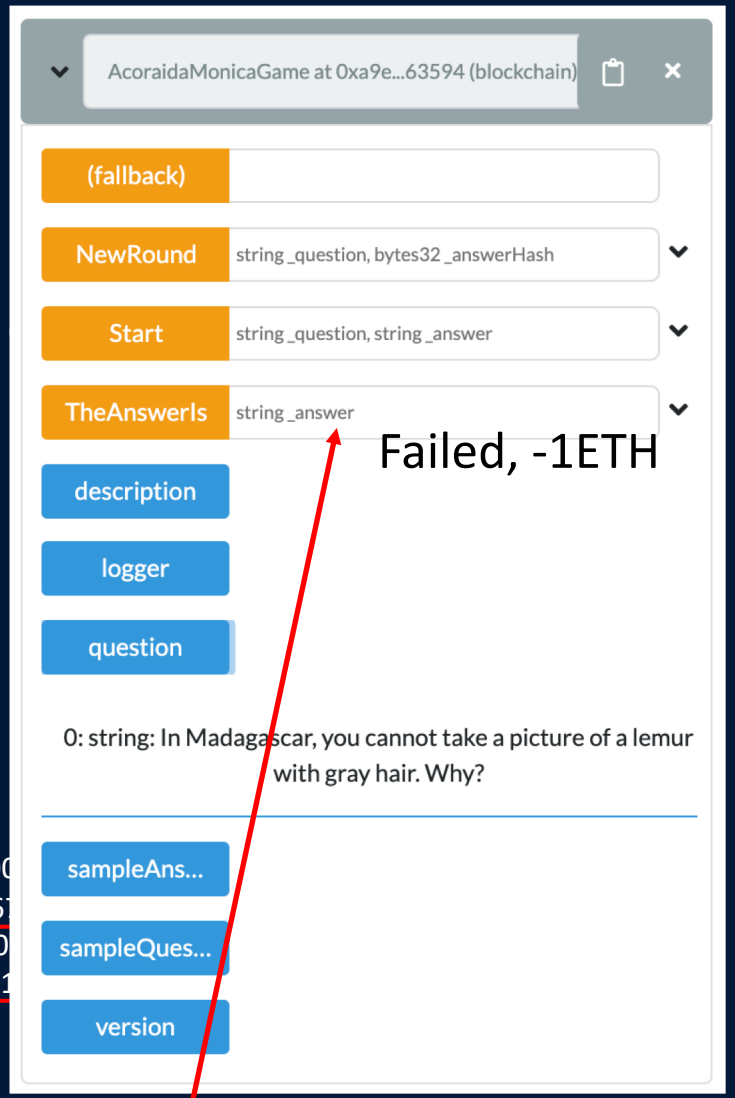
"You need a camera instead of gray hair."

Acoraida Monica

```
> getTransactionsByAccount("0xa9e63a4487f06aa51c2f815b4807b80b64363594",0,100)
Searching for transactions to/from account "0xa9e63a4487f06aa51c2f815b4807b80b64363594" within blocks
Searching block 0
  tx hash          : 0x541b1ce0ba6f4a7711362268f2f69de6f6a5135d0191e2deff0172c5bede40df
   nonce           : 5
   blockHash       : 0x208995ec51dffdb0a865d76748a0c2eb06faba778da82bcb8194e1b391c6128a
   blockNumber     : 23
   transactionIndex: 0
   from            : 0x492705c00090cb7c1cbb5ec3ab0b09f310dec399
   to              : 0xa9e63a4487f06aa51c2f815b4807b80b64363594
   value           : 1e+24
   time            : 1564904313 Sun, 04 Aug 2019 07:38:33 GMT
   gasPrice        : 20000000000
   gas             : 90000
   input           :
0xc76de3e9000000000000000000000000000000000000000000000000000000000000004000000000000000
0000000c000000000000000000000000000000000000000000000000000000000000000048496e204d6164616
b652061207069637475726520 6f662061206c656d7572207769746820677261792068 6169722e205768793f00
0000000000000000000000000000000000000000000000000000000000027596f75206e6565642061
726179206861697 22e000000000000000000000000000000000000000000000000000000000000
```

length

Start(string,string)



“In Madagascar, you cannot take a picture of a lemur with gray hair. Why?”

“You need a camera instead of gray hair.”

长亭科技 CHAITIN

Acoraida Monica Warming Up

tx hash : 0x1bc243319346823a5212baf4e1301021c3dca7e2b4a6022eef5543364f174092
nonce : 4
blockHash : 0x2e2f7cf7ec34c5eb368211459f3bc1ed5c0aa62c32b34fb1078351ad61d6434c
blockNumber : 19
transactionIndex: 1
from : 0x492705c00090cb7c1cbb5ec3ab0b09f310dec399
to : null
value : 0
time : 1564904309 Sun, 04 Aug 2019 07:38:29 GMT
gasPrice : 18000000000
gas : 3000000
input :

0x608060405234801561001057600080fd5b506040516101be3803806101be83398101604081815282516020808501518386015117fc76de3e9000000000000000000000000
0000000000000000000000000000008652600486019485529086018051604487015280519396909591019360016 0a060020a0387169363c76de3e993879387939283926024830192606401919087019080838360005b838110156100b2578181015183820152602001610 09a565b505050509050908101906 01f1680156100df5780820380516001836020036101000a031916815260200191505b508381038252845181528451602091820191860190808383600 05b83811015610112578181015183820152602001610 0fa565b5050505090509081019060 1f16801561013f5780820380516001836020036101000a031916815260200191505b5094505050505060006040518083038160008780 3b158015610 16057600080fd5b505af115801561017457 3d6000803e3d6000fd5b505050505050506035806101896000396000f3006080604052600080fd00a165627a7a723058203e4de9
ab30a2fe5e13562c8ce9a8dbc53b04e0ebee143230356f0bebf608c938002900000000000000000000000000a9e63a4487f06aa51c2f815b4807b80b6436359400000000000000000
00000000000000000000000000000000000000000000000600000000000000000000000000000000000000000000000000000000000000e0000000000000000000000000000
000000000000000000000000000000000000000048496e204d616461676173636172 2c20796f752063616e6e6f742074616b6520612070696374757265206f662061206c656d75
72207769746820677261792068616972 2e2057687 93f0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000017200000000000000000000000000000000000000000000000000000000000000


```
interface b{
    function Start(string _question, string _answer) public payable;
}
contract a{
    constructor(address t, string q, string r) public{
        b(t).Start(q,r);
    }
}
```

"r" (the real answer)

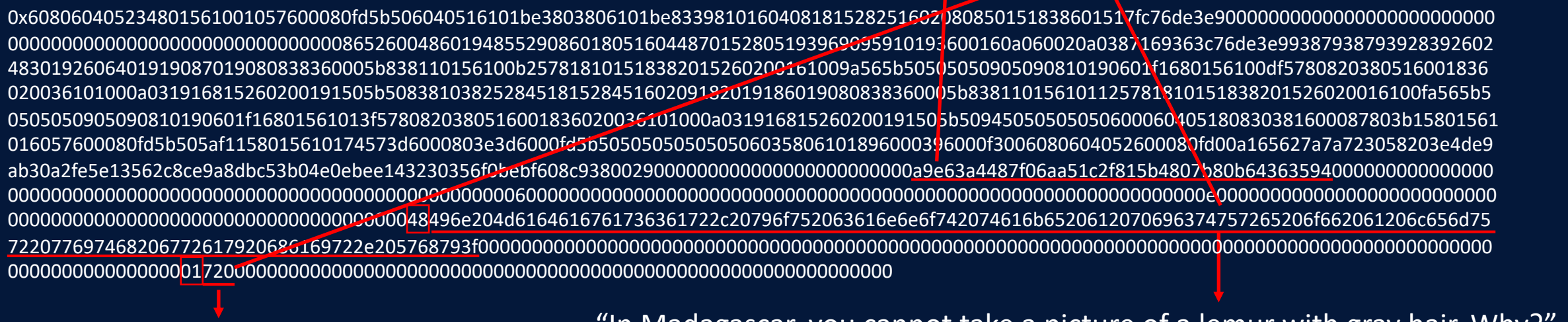
"In Madagascar, you cannot take a picture of a lemur with gray hair. Why?"

Acoraida Monica Warming Up

```
function Start(string _question, string _answer) public payable{
    if(answerHash==0){
        answerHash = keccak256(_answer);
        question = _question;
        questioner = msg.sender;
    }
}
```

```
interface b{
    function Start(string _question, string _answer) public payable;
}
contract a{
    constructor(address t, string q, string r) public{
        b(t).Start(q,r);
    }
}
```

Internal tx to hide the actual 'Start' function call



"r" (the real answer)

"In Madagascar, you cannot take a picture of a lemur with gray hair. Why?"

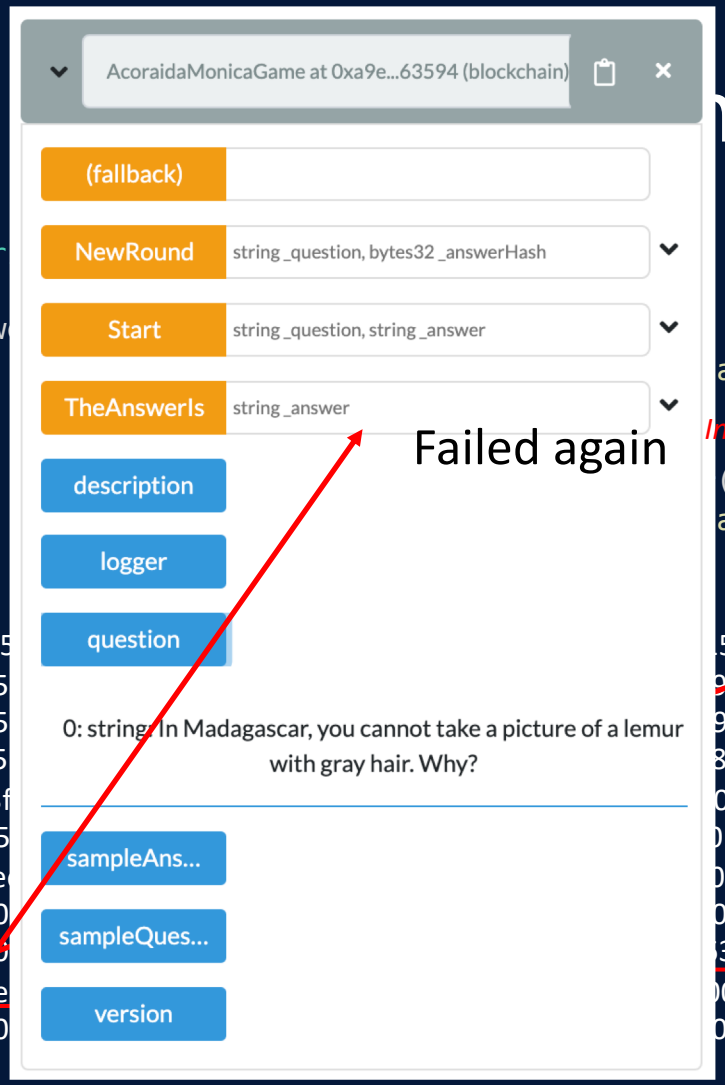
nica Warming Up

```
function Start(string _question, str
    if(answerHash==0){
        answerHash = keccak256(_answ
        question = _question;
        questioner = msg.sender;
    }
}
```

```
art(string _question, string _answer) public payable;
```

Internal tx to hide the actual 'Start' function call

```
(address t, string q, string r) public{
art(q,r);
```



Failed again

0x6080604052348015610010576000 80fd5
0000000000000000000000000000000865
4830192606401919087019080838360005
020036101000a031916815260200191505
0505050905090810190601f16801561013f
016057600080fd5b505af11580156101745
ab30a2fe5e13562c8ce9a8dbc53b04e0ebe
0000000000000000000000000000000000
0000000000000000000000000000000000
722077697468206772617920686169722e
0000000000000000017200000000000000

528251602080850151838601517fc76de3e900000000000000000000000000
0995910193600160a060020a0387169363c76de3e99387938793928392602
9a565b505050509050908101906 01f1680156100df5780820380516001836
80838360005b838110156101125781810151838201526020016100fa565b5
020019150 5b5094505050505060006040518083038160008 7803b15801561
0189600039 6000f3006080604052600080fd00a165627a7a723058203e4de9
0000000a9e63a4487f06aa51c2f815b4807b80b643635940000000000000000
3616e6e6f742074616b652061207069637475726 5206f662061206c656d75
00000

"r" (the real answer)

"In Madagascar, you cannot take a picture of a lemur with gray hair. Why?"

Warming Up

Query the source code server (provided in the description)

```
contract AcoraidaMonicaGame{
    ...
    Logger public constant logger=Logger(0x5e351bd4247f0526359fb22078ba725a192872f3);
    ...
    function TheAnswerIs(string _answer) onlyHuman public payable{
        if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
            questioner = msg.sender;
            msg.sender.transfer(address(this).balance);
            logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
        }
    }
contract LoggerAgent{
    bytes32 private constant ownerSlot = keccak256("Acoraida Monica is cute :P");
    bytes32 private constant implSlot = keccak256("So is her logger :D");
    constructor() public{...}
    modifier onlyOwner{...}
    function getAddress(bytes32 _slot) internal view returns (address value) {...}
    function setAddress(bytes32 _slot, address _address) internal {...}
    function owner() public view returns (address){...}
    function implementation() public view returns (address){
        return getAddress(implSlot);
    }
    function setOwner(address _owner) onlyOwner public{...}
    function upgrade(address _impl) onlyOwner public {
        setAddress(implSlot, _impl);
    }
    function _delegateforward(address _impl) internal {...}
    function () payable public{
        _delegateforward(implementation());
    }
}
```

```
contract Logger{
    event WeHaveAWinner(address);
    event NewQuestion(string);
    event NewAnswerHs(bytes32);
    function AcoraidaMonicaWantsToKeepALogOfTheWinner(address winner) public {
        emit WeHaveAWinner(winner);
    }
    function AcoraidaMonicaWantsToKnowTheNewQuestion(string _question) public{
        emit NewQuestion(_question);
    }
    function AcoraidaMonicaWantsToKnowTheNewAnswerHash(bytes32 _answerHash) public {
        emit NewAnswerHs(_answerHash);
    }
}
```

卡亭科技
CHAITIN

Warming Up

Look still good, but ...

Warming Up

```
contract AcoraidaMonicaGame{
    ...
    Logger public constant logger=Logger(0x5e351bd4247f0526359fb22078ba725a192872f3);
    ...
    function TheAnswerIs(string _answer) onlyHuman public payable{
        if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
            questioner = msg.sender;
            msg.sender.transfer(address(this).balance);
            logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
        }
    }
contract LoggerAgent{
    bytes32 private constant ownerSlot = keccak256("Acoraida Monica is cute :P");
    bytes32 private constant implSlot = keccak256("So is her logger :D");
    constructor() public{...}
    modifier onlyOwner{...}
    function getAddress(bytes32 _slot) internal view returns (address value) {...}
    function setAddress(bytes32 _slot, address _address) internal {...}
    function owner() public view returns (address){...}
    function implementation() public view returns (address){
        return getAddress(implSlot);
    }
    function setOwner(address _owner) onlyOwner public{...}
    function upgrade(address _impl) onlyOwner public {
        setAddress(implSlot, _impl);
    }
    function _delegateforward(address _impl) internal {...}
    function () payable public{
        _delegateforward(implementation());
    }
}
```

```
contract Logger{
    event WeHaveAWinner(address);
    event NewQuestion(string);
    event NewAnswerHs(bytes32);
    function AcoraidaMonicaWantsToKeepALogOfTheWinner(address winner) public {
        emit WeHaveAWinner(winner);
    }
    function AcoraidaMonicaWantsToKnowTheNewQuestion(string _question) public{
        emit NewQuestion(_question);
    }
    function AcoraidaMonicaWantsToKnowTheNewAnswerHash(bytes32 _answerHash) public {
        emit NewAnswerHs(_answerHash);
    }
}
```

0x0900f010

Function signature collision!

Warming Up

```
contract AcoraidaMonicaGame{
    ...
    Logger public constant logger=Logger(0x5e351bd4247f0526359fb22078ba725a192872f3);
    ...
    function TheAnswerIs(string _answer) onlyHuman public payable{
        if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
            questioner = msg.sender;
            msg.sender.transfer(address(this).balance);
            logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
        }
    }
```

reverted

```
contract LoggerAgent{
    bytes32 private constant ownerSlot = keccak256("Acoraida Monica is cute :P");
    bytes32 private constant implSlot = keccak256("So is her logger :D");
    constructor() public{...}
    modifier onlyOwner{...}
    function getAddress(bytes32 _slot) internal view returns (address value) {...}
    function setAddress(bytes32 _slot, address _address) internal {...}
    function owner() public view returns (address){...}
    function implementation() public view returns (address){
        return getAddress(implSlot);
    }
    function setOwner(address _owner) onlyOwner public{...}
    function upgrade(address _impl) onlyOwner public {
        setAddress(implSlot, _impl);
    }
    function _delegateforward(address _impl) internal {...}
    function () payable public{
        _delegateforward(implementation());
    }
}
```

```
contract Logger{
    event WeHaveAWinner(address);
    event NewQuestion(string);
    event NewAnswerHs(bytes32);
    function AcoraidaMonicaWantsToKeepALogOfTheWinner(address winner) public {
        emit WeHaveAWinner(winner);
    }
    function AcoraidaMonicaWantsToKnowTheNewQuestion(string _question) public{
        emit NewQuestion(_question);
    }
    function AcoraidaMonicaWantsToKnowTheNewAnswerHash(bytes32 _answerHash) public {
        emit NewAnswerHs(_answerHash);
    }
}
```

卡亭科技
CHAITIN

Warming Up

Looks like a dead end …

RE

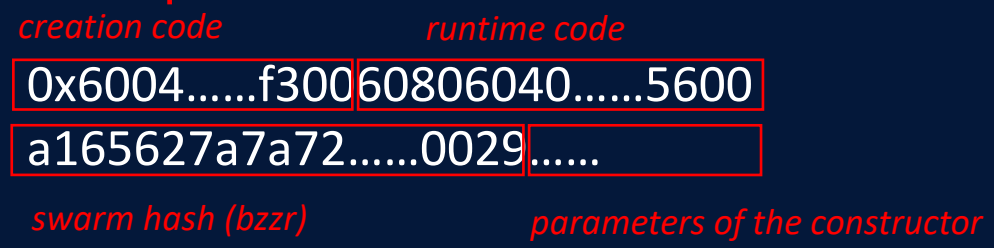
Looks like a dead end …

RE

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    ...
    constructor(bytes a) {
        assembly{
        pc
        0xe1
        add
        jump
        }
    }
   ...
}
```

Creation tx input data (Common)

exec all instructions contained in the constructor
return(runtime code)



长亭科技
CHAITIN

RE

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    ...
    constructor(bytes a) {
        assembly{
        pc
        0xe1
        add
        jump
        }
    }
   ...
}
```

Creation tx input data (AcoraidaMonicaGame)

0x6004600055610120604052607960808190527f41636f7261696461204d6f6e6963612061646d6972657320736d617274206
77560a09081527f79732c207368652764206c696b6520746f20706179203130303030455448207460c0527f6f20746865206f6
e652077686f20636f756c6420616e7377657220686572207160e0527f75657374696f6e2e20576f756c6420697420626520796
f753f0000000000000000610100526100b191600191906100db565b503480156100be57600080fd5b5060405161017738038061
0177833981016040528051015860e101565b828054600181600116156101000203166002900490600052602060002090601f
016020900481019282601f1061011c57805160ff1916838001178555610149565b82800160010185558215610149579182015
b8281111561014957825182559160200191906001019061012e565b50610155929150610159565b5090565b61017391905b
80821115610155576000815560010161015f565b905600000000000000000000000000000000000000000000000000000000
000000002000000000000000000000000000000000000000000000000000000000000009d55b633ccfd60b61262a556109c4
610171f36080604052600436106100985763ffffffff7c0100000000000000000000000000000000000000000000000000000000
0600035041663281ba0a8811461009a5780633fad9ae01461012457806346a3ec671461013957806354fd4d50146101855780
637284e416146101ac578063aa332032146101c1578063c76de3e91461020f578063f24ccbfe14610299578063f7ff68a314610
2d7575b005b3480156100a657600080fd5b506100af6102ec565b60408051602080825283518183015283519192839290830
19185019080838360005b838110156100e95781810151838201526020016100d1565b50505050905090810190601f1680156
101165780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b3480156101305
... ...
828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f1061093e578051
60ff191683800117855561096b565b8280016001018555821561096b579182015b8281111561096b57825182559160200191
9060010190610950565b5061097792915061097b565b5090565b61099591905b80821115610977576000815560010161098
1565b905600a165627a7a723058205bf450560abe05048d3aecd129bb6420ca5acbef2f19c0ef9dca3f158d49e23f0029000000
0000000000000000

RE

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    ...
    constructor(bytes a) {
        assembly{
        pc
        0xe1
        add
        jump
        }
    }
   ...
}
```

creation code

Creation tx input data (AcoraidaMonicaGame)

0x60046000556101206040526079608081905 27f41636f7261696461204d6f6e6963612061646d6972657320736d617274206 77560a09081527f79732c207368652764206c696b6520746f2070617920313030303045544820746 0c0527f6f20746865206f6 e652077686f20636f756c6420616e737765722068657220716 0e0527f75657374696f6e2e20576f756c642069742042652079 6f753f0000000000000061010052610 0b191600191906100db565b503480156100be57600080fd5b5060405161017738038061 0177833981016040528051015860e101565b828054600181600116156101000203166002900490600052602060002090601f 016020900481019282601f1061011c57805160ff191683800117855561014956 5b828001600101855582156101495791820 15b828111156101495782518255916020019190600101906101 2e565b506101559291506101595 65b5090565b61017391905b 8082111561015557600081556001016101 5f565b905600

0x60046000556101206040526079608081905 27f41636f7261696461204d6f6e6963612061646d6972657320736d617274206 77560a09081527f79732c207368652764206c696b6520746f2070617920313030303045544820746 0c0527f6f20746865206f6 e652077686f20636f756c6420616e737765722068657220716 0e0527f75657374696f6e2e20576f756c642069742042652079 6f753f0000000000000061010052610 0b191600191906100db565b503480156100be57600080fd5b5060405161017738038061 0177833981016040528051015860e101565b828054600181600116156101000203166002900490600052602060002090601f 016020900481019282601f1061011c57805160ff191683800117855561014956 5b828001600101855582156101495791820 15b828111156101495782518255916020019190600101906101 2e565b506101559291506101595 65b5090565b61017391905b 8082111561015557600081556001016101 5f565b905600

RE

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    ...
    constructor(bytes a) {
        assembly{
        pc
        0xe1
        add
        jump
        }
    }
   ...
}
```

creation code

parameter (bytes a)

Creation tx input data (AcoraidaMonicaGame)

0000000000000000000000000000000000000000000000000000000000000000
0000000020000000000000000000000000000000000000000000000000000009d55b633ccfd60b61262a556109c4
610171f360806040526004361061009857 63ffffffff7c0100000000000000000000000000000000000000000000000000000000
0600035041663281ba0a8811461009a5780633fad9ae0146101245780634a3ec67146101395780635fd4d50146101855780
637284e416146101ac578063aa332032146101c1578063c76de3e91461020f578063f24ccbfe14610299578063f7ff68a314610
2d7575b005b3480156100a657600080fd5b506100af6102ec565b60408051602080825283518183015283519192839290830
191850190808383600005b838110156100e95781810151838201526020016100d1565b50505050905090810190601f1680156
101165780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b3480156101305
… …
828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f1061093e578051
60ff191683800117855561096b565b828001600101855582156109 6b579182015b828111156109 6b5782518255916020019
906001019061095056 5b506109779291506109 7b565b5090565b61099591905b8082111561097757600081556001016098
1565b905600a165627a7a723058205bf450560abe05048d3aecd129bb6420ca5acbef2f19c0ef9dca3f158d49e23f0029000000
0000000000000000

RE

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    ...
    constructor(bytes a) {
        assembly{
        pc
        0xe1
        add
        jump
        }
    }
   ...
}
```

creation code

runtime code

swarm hash (bzzr)

parameter (bytes a)

Creation tx input data (AcoraidaMonicaGame)

0x60046000556101206040526079608081905 27f41636f7261696461204d6f6e6963612061646d6972657320736d617274206
77560a09081527f79732c207368652764206c696b6520746f2070617920313030303045544820746 0c0527f6f20746865206f6
e652077686f20636f756c6420616e737765722068657220716 0e0527f75657374696f6e2e20576f756c6420697420626520796
f753f000000000000006101005261 00b191600191906100db565b503480156100be57600080fd5b506040516101773803806 1
0177833981016040528051015860e101565b828054600181600116156101000203166002900490600052602060002090601f
016020900481019282601f1061011c57805160ff191683800117855561014956 5b828001600101855582156101495791820 15
b828111156101495782518255916020019190600101906 1012e565b506101559291506101595 65b5090565b61017391905b
8082111561015557600081556001016101 5f565b905600000000000000000000000000000000000000000000000000000000
0000000020000000000000000000000000000000000000000000000000000000000000009d55b633ccfd60b61262a556109c4
610171f360806040526004361061009857 63ffffffff7c010000000000000000000000000000000000000000000000000000000
0600035041663281ba0a8811461009a5780633fad9ae0146101245780634 6a3ec67146101395780635 4fd4d50146101855780
637284e416146101ac578063aa332032146101c1578063c76de3e91461020f578063f24ccbfe1461029578063f7ff68a314610
2d7575b005b3480156100a657600080fd5b506100af6102ec565b604080516020808252835181830152835191928392908 30
191850190808383600 05b838110156100e95781810151838201526020016100d1565b505050509050908 10190601f1680156
101165780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b3480156101305
… …
828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f1061093e578051
60ff191683800117855561096b565b828001600101855582156 1096b579182015b828111156 1096b5782518255916020019 1
90600101906109505 65b506109779291506 1097b565b5090565b61099591905b808211156109775760008155600101610 98
1565b905600a165627a7a723058205bf450560abe05048d3aecd129bb6420ca5acbef2f19c0ef9dca3f158d49e23f0029000000
0000000000000000

RE

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    ...
    constructor(bytes a) {
        assembly{
            pc
            0xe1
            add
            jump
        }
    }
    ...
}
```

```
0214 PC
0215 PUSH1 e1
0217 ADD
0218 JUMP
```

Creation tx input data (AcoraidaMonicaGame)

```
0439 JUMPDEST
0440 PUSH4 3ccfd60b
0445 PUSH2 262a
0448 SSTORE
0449 PUSH2 09c4
0452 PUSH2 0171
0455 RETURN
```


0x6004600055610120604052607960808190527f41636f7261696461204d6f6e ... 7274206
77560a09081527f79732c207368652764206c696b6520746f207061792031303 ... 65206f6
e652077686f20636f756c6420616e7377657220686572207160e0527f7565737 ... 6520796
f753f0000000000000000610100526100b191600191906100db565b5034801561 ... 8038061
0177833981016040528051015860e101565b828054600181600116156101000 ... 2090601f
016020900481019282601f1061011c57805160ff191683800117855561014956 ... 9182015
b8281111561014957825182559160200191906001019061012e565b5061015 ... 391905b
80821115610155576000815560010161015f565b905600000000000000000 ... 0000000
00000000200000000000000000000000000000000000000000000000000000009d55b633ccfd60b61262a556109c4
610171f36080604052600436106100985763ffffffff7c010000000000000000000000000000000000000000000000000000000
0600035041663281ba0a8811461009a5780633fad9ae01461012457806346a3ec671461013957806354fd4d50146101855780
... 610299578063f7ff68a314610
2d7575b005b3480156100a657600080fd5b506100af6102ec565b60408051602080825283518183015283519192839290830
19185019080838360005b838110156100e95781810151838201526020016100d1565b50505050905090810190601f1680156
101165780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b3480156101305
... ...
828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f1061093e578051
60ff1916838001178555561096b565b82800160010185555821561096b579182015b8281111561096b57825182559160200191
9060010190610950565b5061097792915061097b565b5090565b61099591905b80821115610977576000815560010161098
1565b905600a165627a7a723058205bf450560abe05048d3aecd129bb6420ca5acbef2f19c0ef9dca3f158d49e23f0029000000
0000000000000000

RE

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    ...
    constructor(bytes a) {
        assembly{
        pc
        0xe1
        add
        jump
        }
    }
   ...
}
```

```
0214 PC
0215 PUSH1 e1
0217 ADD
0218 JUMP
```

Creation tx input data (AcoraidaMonicaGame)

```
0439 JUMPDEST
0440 PUSH4 3ccfd60b
0445 PUSH2 262a
0448 SSTORE
0449 PUSH2 09c4
0452 PUSH2 0171
0455 RETURN
```

The real runtime code:

60806040526004361061009857 63ffffffff7c0100000000000000000000000000000000000000000000000000000000 0600035041663281ba0a8811461009a5780633fad9ae0146101245780634a3ec6714610139578063546354fd4d50146101855780637284e416146101ac578063aa332032146101c1578063c76de3e91461020f578063f24ccbfe1461029578063f7ff68a3146102d7575b005b3480156100a657600080fd5b506100af6102ec565b6040805160208082528351818301528351919283929083019185019080838360005b838110156100e957818101518382015260200161 00d1565b50505050905090810190601f16801561011657808203805160018360200361010 00a031916815260200191505b509250505060405180910390f35b3480156101305 ... ...
828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f1061093e57805160ff1916838001178555610960b565b828001600101855582156096b579182015b828111156096b57825182559160200191906001019061095565b506109779291506109 7b565b5090565b61099591905b8082111561097757600081556001016109815565b905600a165627a7a723058205bf450560abe05048d3aecd129bb6420ca5acbef2f19c0ef9dca3f158d49e23f0029000000000000000000000000

RE

We've replaced the runtime code, what can we do now?

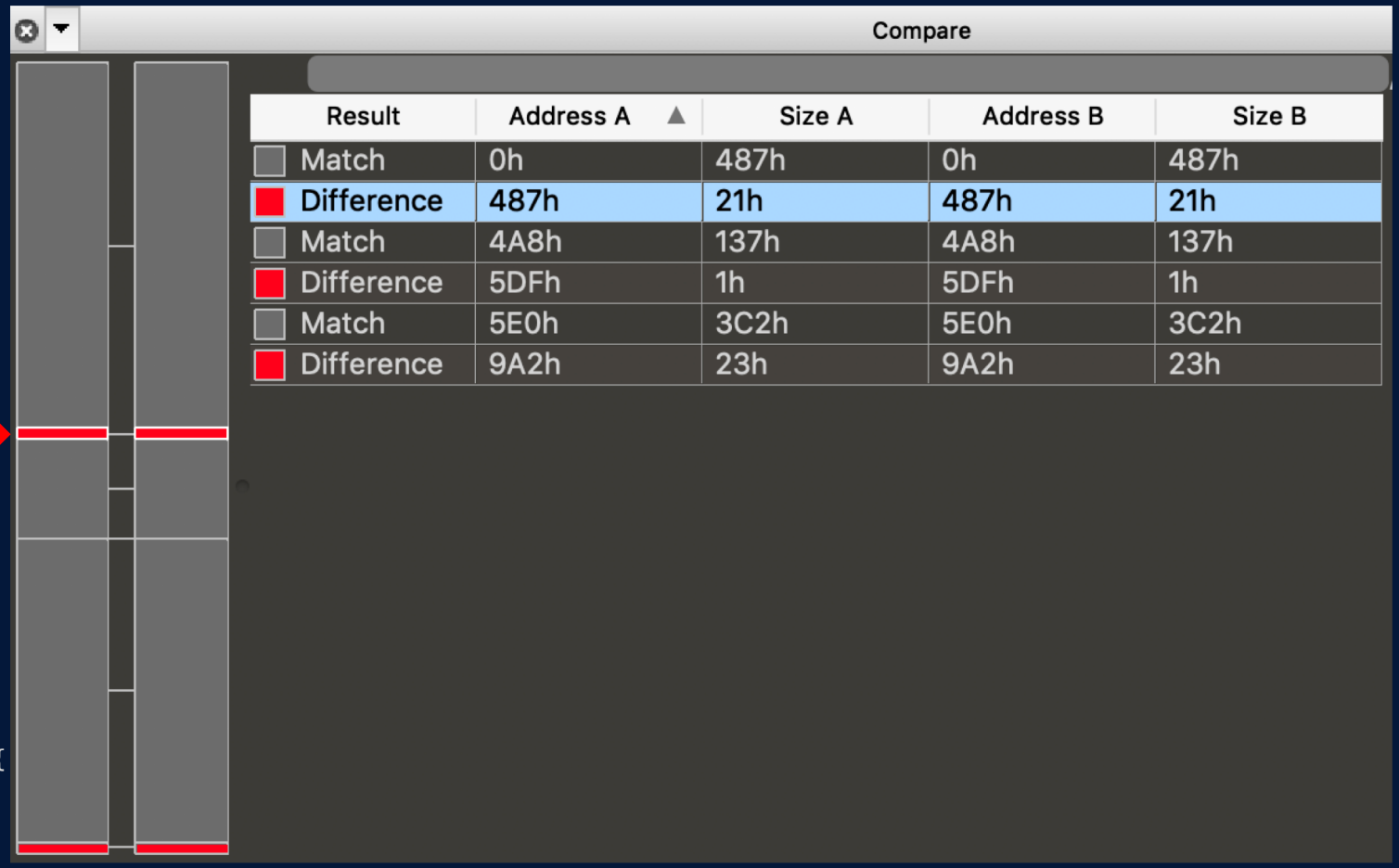
卡亭科技
CHAITIN

RE

?

60806040526004361061009857 63ffffffff7c01000000000000000000000000000000000000000000000000000000000000000060003504166328 1ba0a8811461009a5780633fad9ae0146101245780634 6a3ec67146101395780635 4fd4d50146101855780637 284e416146101ac578063aa332032146101c1578063c76de3e91461020f578063f24ccbfe14610299578063f7ff68a3146102d7575b005b3480156100a657600080fd5b506100af6102ec565b60408051602080825283518183015283519192839290830191850190808383 60005b838110156100e95781810151838201526020016100d1565b50505050905090810190601f16801561011657808203805160018360200361010 00a031916815260200191505b509250505060405180910390f35b3480156101305 … …
828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f1061093e57805160ff191683800117855561096b565b828001600101855582156109 6b579182015b828111156109 6b5782518255916020019190600101906109505 65b50610977929150610 97b565b5090565b61099591905b8082111561097757600081556001016109815 65b905600a165627a7a723058205bf450560abe05048d3aecd129bb6420ca5acbef2f19c0ef9dca3f158d49e23f0029000000000000000000000000

RE




```
constructor(bytes a) {
    /*assembly{
    pc
    0xe1
    add
    jump
    }*/
}
```

compiled from source code *vs.* deployed on the chain

```
constructor(bytes a) {
    assembly{
    pc
    0xe1
    add
    jump
    }
}
```

```
constructor(bytes a) {
    assembly{
    pc
    0xe1
    add
    jump
    }
}
```

AMReal.bin

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0460h:	15	61	05	36	57	60	02	80	54	73	FF	FF	FF	FF	FF	FF
0470h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	19	16
0480h:	33	90	81	17	90	91	55	61	04	B1	36	60	A0	14	15	81
0490h:	57	61	40	2E	61	04	9D	33	61	FF	FF	16	56	5B	51	01
04A0h:	56	15	80	15	61	04	B1	56	3D	60	00	80	3E	3D	60	00

RE

?

```
constructor(bytes a) {
    /*assembly{
    pc
    0xe1
    add
    jump
    }*/
}
```

AMSrc.bin

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0460h:	15	61	05	36	57	60	02	80	54	73	FF	FF	FF	FF	FF	FF
0470h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	19	16
0480h:	33	90	81	17	90	91	55	60	40	51	30	31	80	15	61	08
0490h:	FC	02	91	60	00	81	81	81	85	88	88	F1	93	50	50	50
04A0h:	50	15	80	15	61	04	B1	57	3D	60	00	80	3E	3D	60	00

compiled from source code *vs.* deployed on the chain

RE

AMReal.bin

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0460h:	15	61	05	36	57	60	02	80	54	73	FF	FF	FF	FF	FF	FF
0470h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	19	16
0480h:	33	90	81	17	90	91	55	61	04	B1	36	60	A0	14	15	81
0490h:	57	61	40	2E	61	04	9D	33	61	FF	FF	16	56	5B	51	01
04A0h:	56	15	80	15	61	04	B1	56	3D	60	00	80	3E	3D	60	00

```
PUSH2 loc_4B1
CALLDATASIZE
PUSH1 0xa0
EQ
ISZERO
DUP2
JUMPI
```

```
loc_4B1:
JUMPDEST
POP
PUSH1 0x40
DUP1
MLOAD
PUSH32 0x900f010000000000000000000000000000000000000000
DUP2
MSTORE
CALLER
PUSH1 0x4
DUP3
ADD
MSTORE
SWAP1
MLOAD
PUSH20 0x5e351bd4247f0526359fb22078ba725a192872f3
SWAP2
PUSH4 0x900f010
SWAP2
PUSH1 0x24
DUP1
DUP4
ADD
SWAP3
```

```
PUSH2 0x402e
PUSH2 loc_49D
CALLER
PUSH2 0xffff
AND
JUMP
```

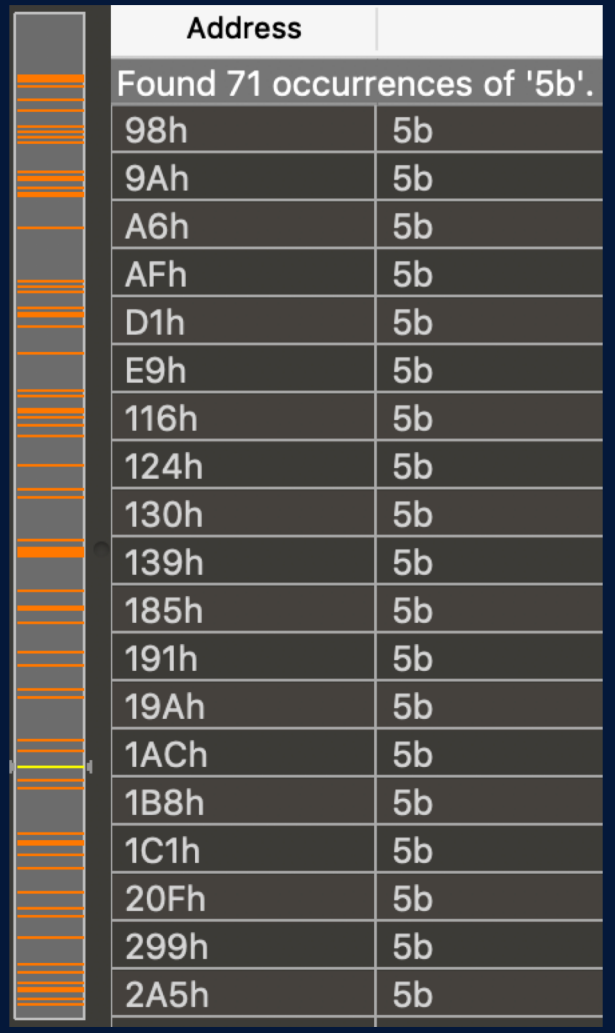
```
function TheAnswerIs(string _answer) onlyHuman public payable{
    //require(msg.sender != questioner);
    if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
        questioner = msg.sender;
        msg.sender.transfer(address(this).balance);
        logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
    }
}
```

```
push 0x04b1
if (calldatasize == 0xa0){
    push 0x402e
    push 0x049d
    jump to msg.sender & 0xffff
}
```

Arbitrary Jump!

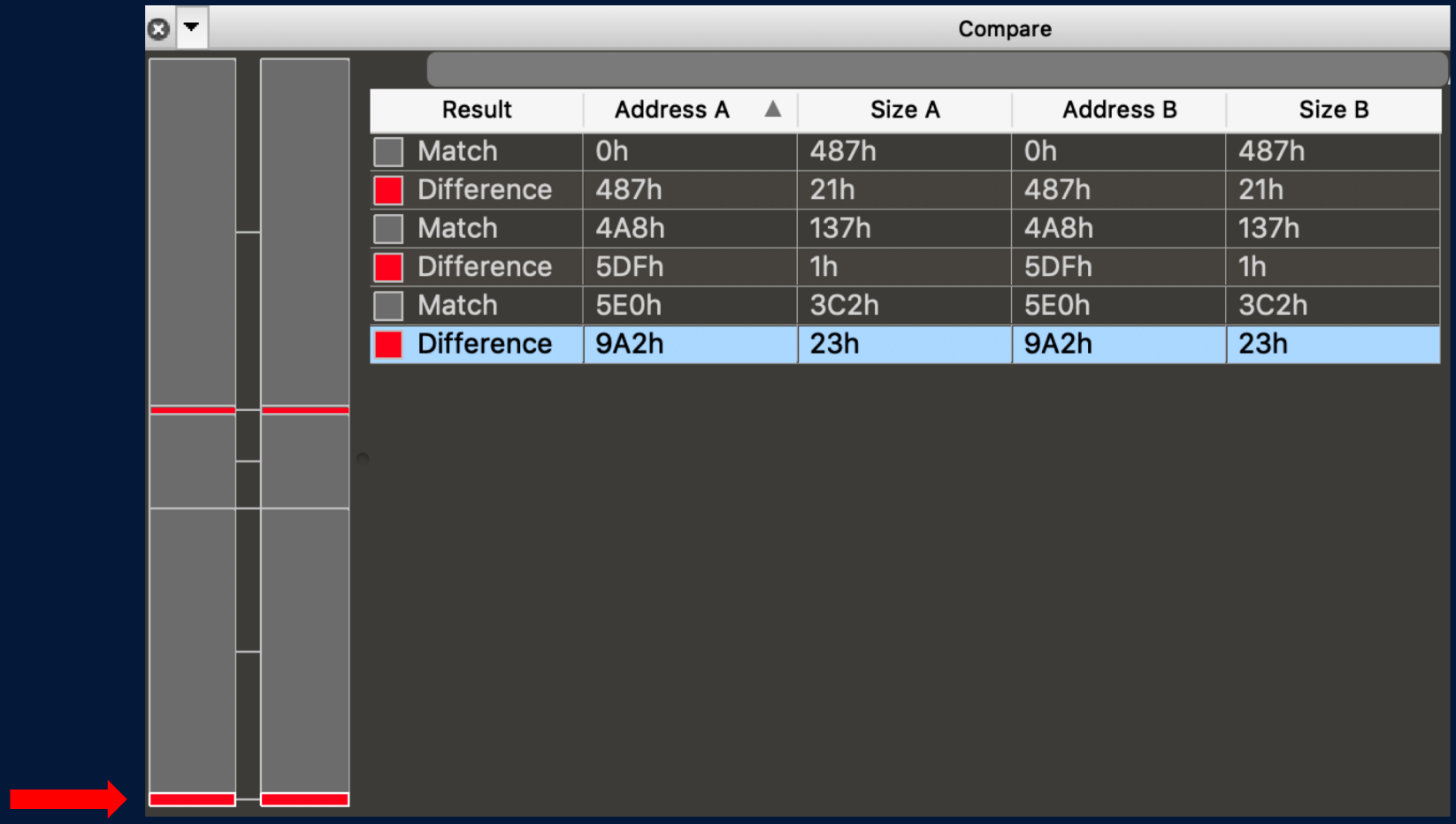
RE

EVM can only jump to 0x5b(jumpdest) instruction.



Address	
Found 71 occurrences of '5b'.	
98h	5b
9Ah	5b
A6h	5b
AFh	5b
D1h	5b
E9h	5b
116h	5b
124h	5b
130h	5b
139h	5b
185h	5b
191h	5b
19Ah	5b
1ACh	5b
1B8h	5b
1C1h	5b
20Fh	5b
299h	5b
2A5h	5b

RE



compiled from source code *vs.* deployed on the chain

RE

```
constructor(bytes a) {
    assembly{
    pc
    0xe1
    add
    jump
    }
}
```

AMReal.bin

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0990h:	01	61	09	81	56	5B	90	56	00	A1	65	62	7A	7A	72	30
09A0h:	58	20	5B	F4	50	56	0A	BE	05	04	8D	3A	EC	D1	29	BB
09B0h:	64	20	CA	5A	CB	EF	2F	19	C0	EF	9D	CA	3F	15	8D	49
09C0h:	E2	3F	00	29	0A											

```
constructor(bytes a) {
    /*assembly{
    pc
    0xe1
    add
    jump
    }*/
}
```

AMSrc.bin

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0990h:	01	61	09	81	56	5B	90	56	00	A1	65	62	7A	7A	72	30
09A0h:	58	20	26	E1	EA	24	AC	62	DC	75	D6	0D	69	68	95	60
09B0h:	69	78	8F	D0	50	BC	A6	44	BA	F2	CD	DB	FD	29	A0	0A
09C0h:	55	93	00	29	0A											

compiled from source code *vs.* deployed on the chain

RE

```
constructor(bytes a) {
    assembly{
    pc
    0xe1
    add
    jump
    }
}
```

AMReal.bin ×

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0990h:	01	61	09	81	56	5B	90	56	00	A1	65	62	7A	7A	72	30
09A0h:	58	20	5B	F4	50	56	0A	BE	05	04	8D	3A	EC	D1	29	BB
09B0h:	64	20	CA	5A	CB	EF	2F	19	C0	EF	9D	CA	3F	15	8D	49
09C0h:	E2	3F	00	29	0A											

Encoding of the Metadata Hash in the Bytecode

Because we might support other ways to retrieve the metadata file in the future, the mapping `{"bzzr0": <Swarm hash>}` is stored CBOR-encoded. Since the beginning of that encoding is not easy to find, its length is added in a two-byte big-endian encoding. The current version of the Solidity compiler thus adds the following to the end of the deployed bytecode:

```
0xa1 0x65 'b' 'z' 'z' 'r' '0' 0x58 0x20 <32 bytes swarm hash> 0x00 0x29
```

RE

```
constructor(bytes a) {
    assembly{
    pc
    0xe1
    add
    jump
    }
}
```

AMReal.bin

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0990h:	01	61	09	81	56	5B	90	56	00	A1	65	62	7A	7A	72	30
09A0h:	58	20	5B	F4	50	56	0A	BE	05	04	8D	3A	EC	D1	29	BB
09B0h:	64	20	CA	5A	CB	EF	2F	19	C0	EF	9D	CA	3F	15	8D	49
09C0h:	E2	3F	00	29	0A											

```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```

```
constructor(bytes a) {
    /*assembly{
    pc
    0xe1
    add
    jump
    }*/
}
```

AMSrc.bin

Edit As: Hex Run Script Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0990h:	01	61	09	81	56	5B	90	56	00	A1	65	62	7A	7A	72	30
09A0h:	58	20	26	E1	EA	24	AC	62	DC	75	D6	0D	69	68	95	60
09B0h:	69	78	8F	D0	50	BC	A6	44	BA	F2	CD	DB	FD	29	A0	0A
09C0h:	55	93	00	29	0A											

compiled from source code *vs.* deployed on the chain

长亭科技 CHAITIN

RE

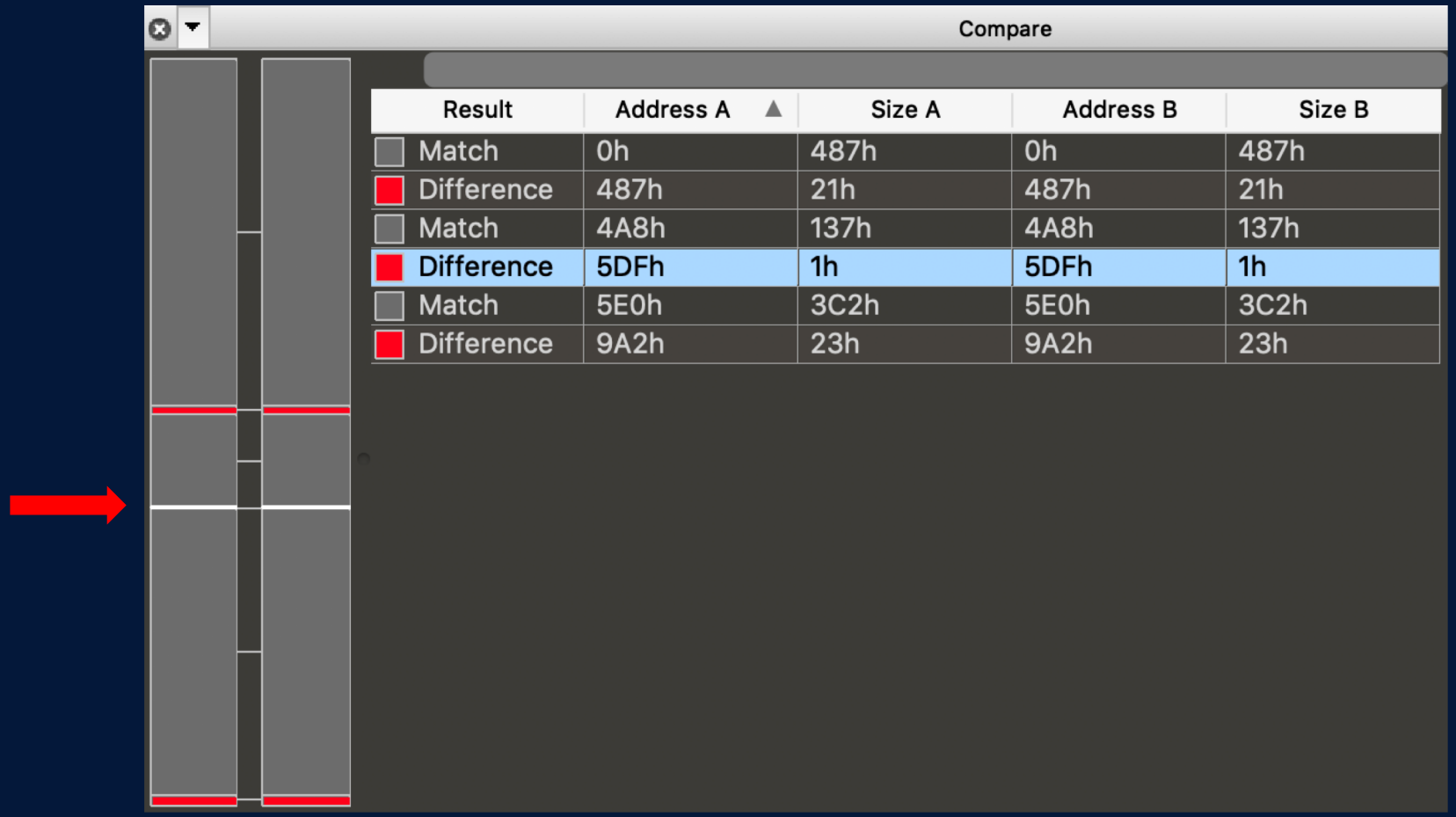
```
1169 PUSH2 402e
1172 PUSH2 049d
1175 CALLER
1176 PUSH2 ffff
1179 AND
1180 JUMP
```

```
????
???? JUMP
```


F4	DELEGATECALL	gas | addr | argsOffset | argsLength | retOffset | retLength	success	success, memory[retOffset:retOffset+retLength] = address(addr).delegatecall.gas(gas)(memory[argsOffset:argsOffset+argsLength])

```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```

RE



compiled from source code *vs.* deployed on the chain

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

AMReal.bin

Edit As: Hex | Run Script | Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
05D0h:	60	40	80	51	60	60	81	01	82	52	60	30	80	82	52	56	`@€Q``..,R`0€,RV
05E0h:	24	2A	21	26	23	5E	5B	60	20	61	40	2E	33	3B	54	61	$*!&#^[` a@.3;Ta
05F0h:	26	2A	54	60	20	52	60	3C	60	7E	35	5A	60	5E	35	56	&*T` R`<`~5Z`^5V
0600h:	60	20	83	01	90	81	52	7F	20	59	6F	75	20	62	65	61	` ƒ...R. You bea
0610h:	74	20	6D	65	21	20	3A	44	00	00	00	00	00	00	00	00	t me! :D........

AMSrc.bin

Edit As: Hex | Run Script | Run Template

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
05D0h:	60	40	80	51	60	60	81	01	82	52	60	30	80	82	52	7F	`@€Q``..,R`0€,R.
05E0h:	24	2A	21	26	23	5E	5B	60	20	61	40	2E	33	3B	54	61	$*!&#^[` a@.3;Ta
05F0h:	26	2A	54	60	20	52	60	3C	60	7E	35	5A	60	5E	35	56	&*T` R`<`~5Z`^5V
0600h:	60	20	83	01	90	81	52	7F	20	59	6F	75	20	62	65	61	` ƒ...R. You bea
0610h:	74	20	6D	65	21	20	3A	44	00	00	00	00	00	00	00	00	t me! :D........

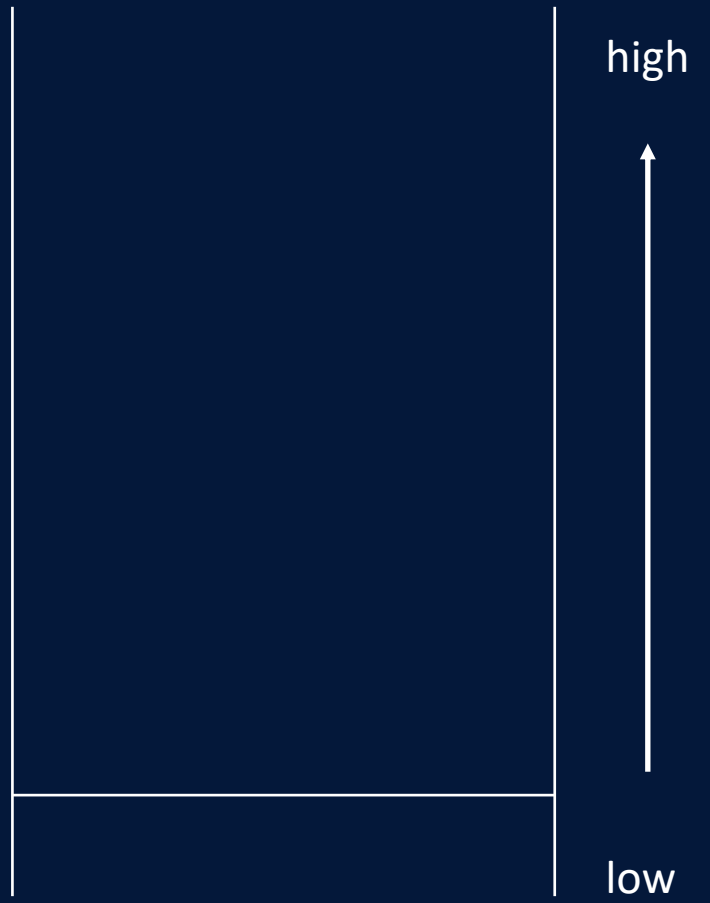
1 byte diff

```
PUSH32 242a2126235e5b602061402e333b5461262a54602052603c607e355a605e3556
```

compiled from source code *vs.* deployed on the chain

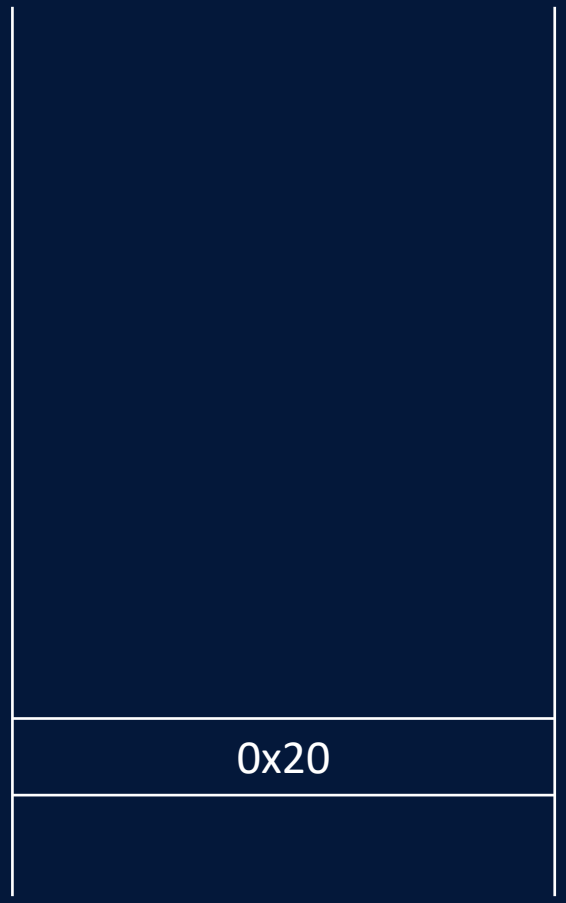
RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```



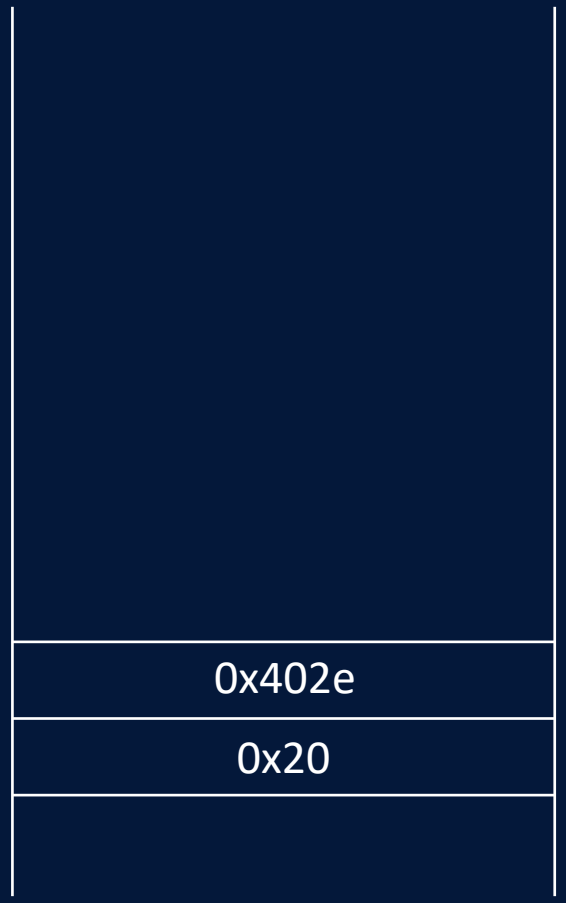
RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```



RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

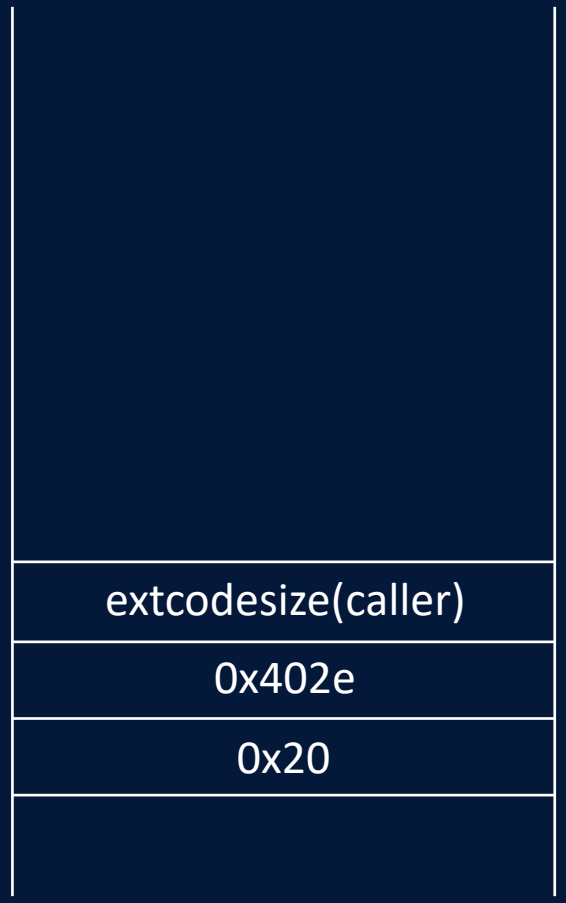


RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```



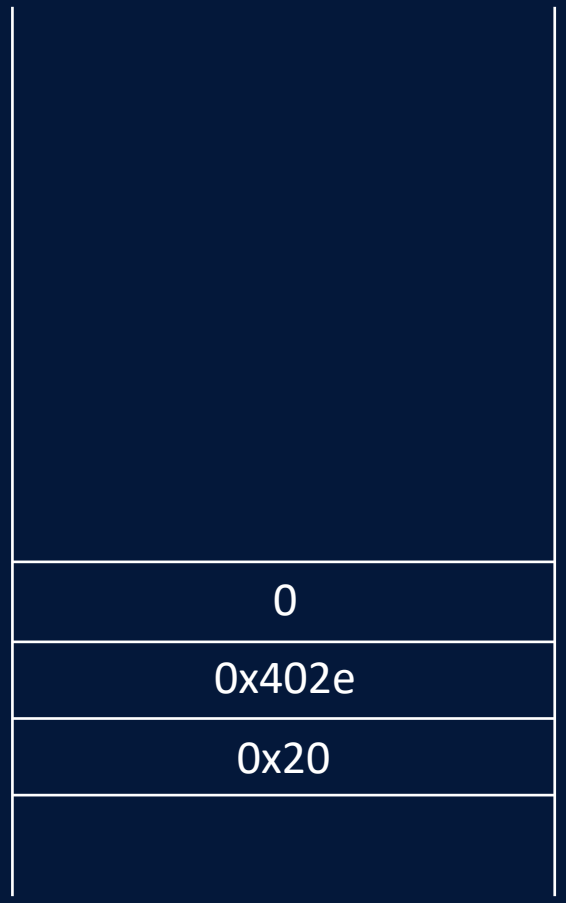

```
function TheAnswerIs(string _answer) onlyHuman public payable{
    //require(msg.sender != questioner);
    if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
        questioner = msg.sender;
        logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
    }
}
```

```
push 0x04b1
if (calldatasize == 0xa0){
    push 0x402e
    push 0x049d
    jump to msg.sender & 0xffff
}
```

```
modifier onlyHuman{
    uint size;
    address addr = msg.sender;
    assembly { size := extcodesize(addr) }
    require(size==0);
    _;
}
```

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```




```
function TheAnswerIs(string _answer) onlyHuman public payable{
    //require(msg.sender != questioner);
    if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
        questioner = msg.sender;
        logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
    }
}
```

```
push 0x04b1
if (calldatasize == 0xa0){
    push 0x402e
    push 0x049d
    jump to msg.sender & 0xffff
}
```

```
modifier onlyHuman{
    uint size;
    address addr = msg.sender;
    assembly { size := extcodesize(addr) }
    require(size==0);
    _;
}
```

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

RE

```
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    ...
    constructor(bytes a) {
        assembly{
        pc
        0xe1
        add
        jump
        }
    }
   ...
}
```

```
0213 ADD
0214 PC
0215 PUSH1 e1
0217 ADD
0218 JUMP
```

Creation tx input data (AcoraidaMonicaGame)

```
0439 JUMPDEST
0440 PUSH4 3ccfd60b
0445 PUSH2 262a
0448 SSTORE
0449 PUSH2 09c4
0452 PUSH2 0171
0455 RETURN
```

0x60046000556101206040526079608081905 27f41636f726169646120 d6f6e 72742 06
77560a09081527f79732c207368652764206c696b6520746f2070617920313 3 652 6f6
e652077686f20636f756c6420616e737765722068657220 7160e0527 756573 652079 6
f753f0000000000000061010052610 b191600191906100db565b50 48015 61 8038061
01778339810160405280510158 0e10156 5b828054600181600116156101000 2090601f
0160209004810192826 1f1061011c57805160ff191683800117855561014956 9182015
b828111156101495782518255916020019190600101906101 e565b50 1015 391905b
8082111561015557600081556001016101 f565b905600 0000000000000000 0050000
0000000020000000000000000000000000000000000000000000000000000000 9d5 b633ccfd60b61262a 556109c4
610171f360806040526004361061009857 ffffffff7c0100000000000000000000000000000000000000000000000000000000
060003504166328 1ba0a8811461009a5780633fad9ae0146101245780634 3ec671461013957806354fd4d50146101855780
610299578063f7ff68a314610
2d7575b005b3480156100a657600080fd5b506100af6102ec565b60408051602080825283518183015283519192839290830
191850190808383600 5b838110156100e9578181015183820152602001 6100d1565b50505050905090810190601f1680156
101165780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b3480156101305
… …
828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f1061093e578051
60ff191683800117855561096b565b828001600101855582156 096b579182015b828111156 096b5782518255916020019
906001019061095056 5b5061097792915061097b565b5090565b61099591905b8082111561097757600081556001016098
1565b905600a165627a7a723058205bf450560abe05048d3aecd129bb6420ca5acbef2f19c0ef9dca3f158d49e23f0029000000
0000000000000000

SSTORE:
storage[key] = value

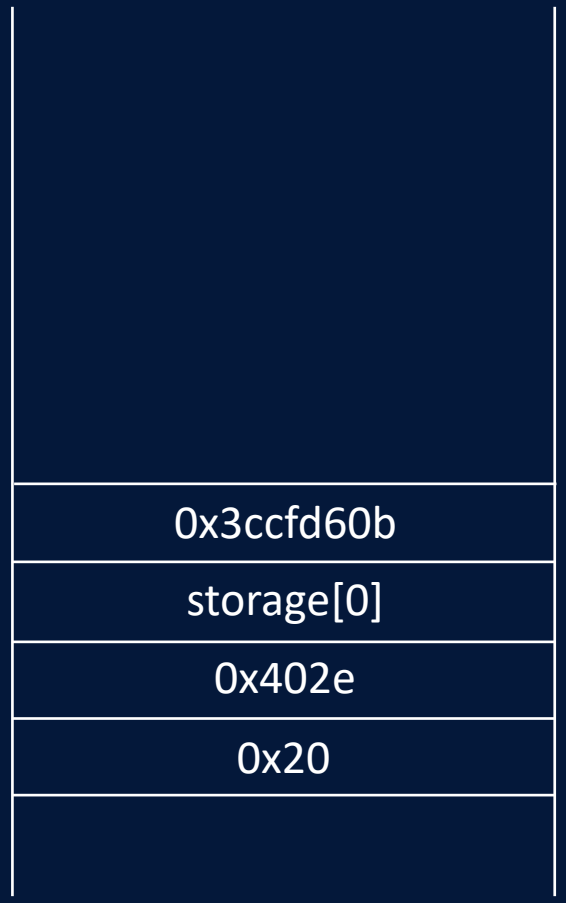
storage[0x262a] = 0x3ccfd60b
```
```

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```



ID	Text Signature	Bytes Signature
614	withdraw()	0x3ccfd60b

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

mem[0x20:0x40]=0x3ccfd60b

ID	Text Signature	Bytes Signature
614	withdraw()	0x3ccfd60b

RE

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

CALLDATALOAD(i):
stack <- calldata[i:i+32]

calldataload(0x5e)
gasRemaining
calldataload(0x7e)
0x3c
storage[0]
0x402e
0x20

mem[0x20:0x40]=0x3ccfd60b

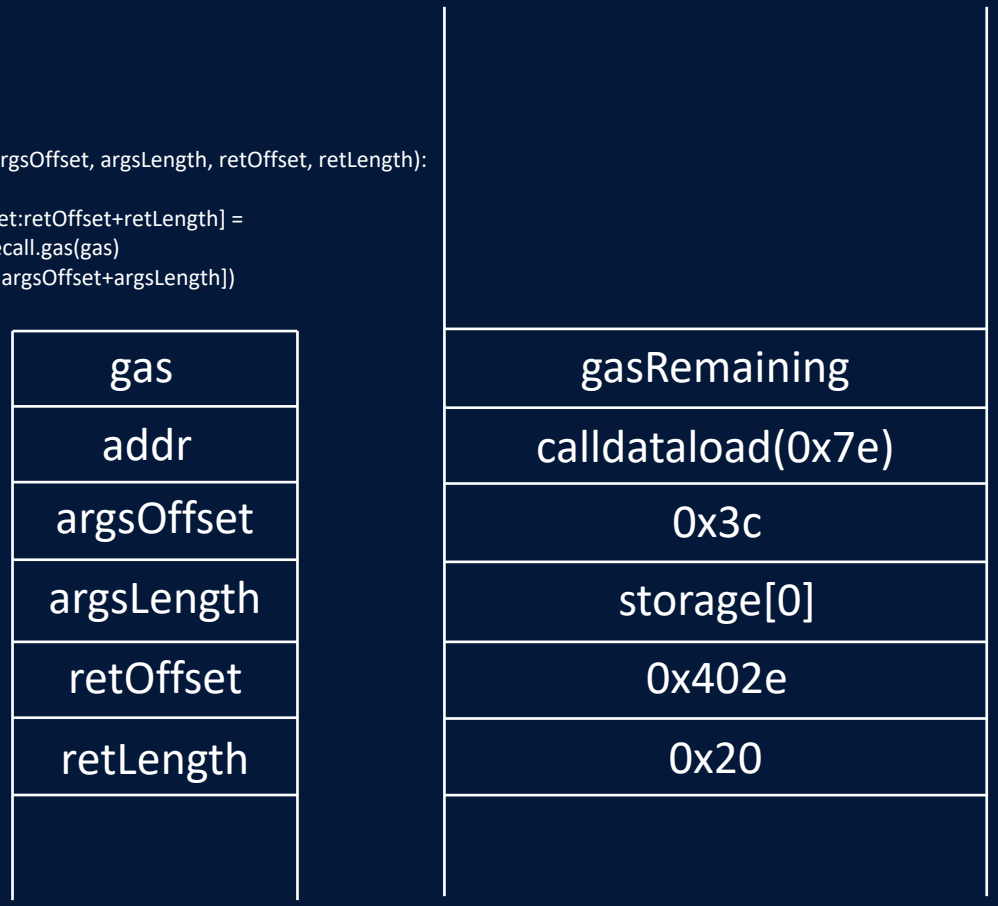
calldata[0x5e:0x7e]= 2466 = 0x09a2

```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```

RE

delegatecall(gas, address, argsOffset, argsLength, retOffset, retLength):

success, memory[retOffset:retOffset+retLength] =
address(addr).delegatecall.gas(gas)
(memory[argsOffset:argsOffset+argsLength])



mem[0x20:0x40]=0x3ccfd60b

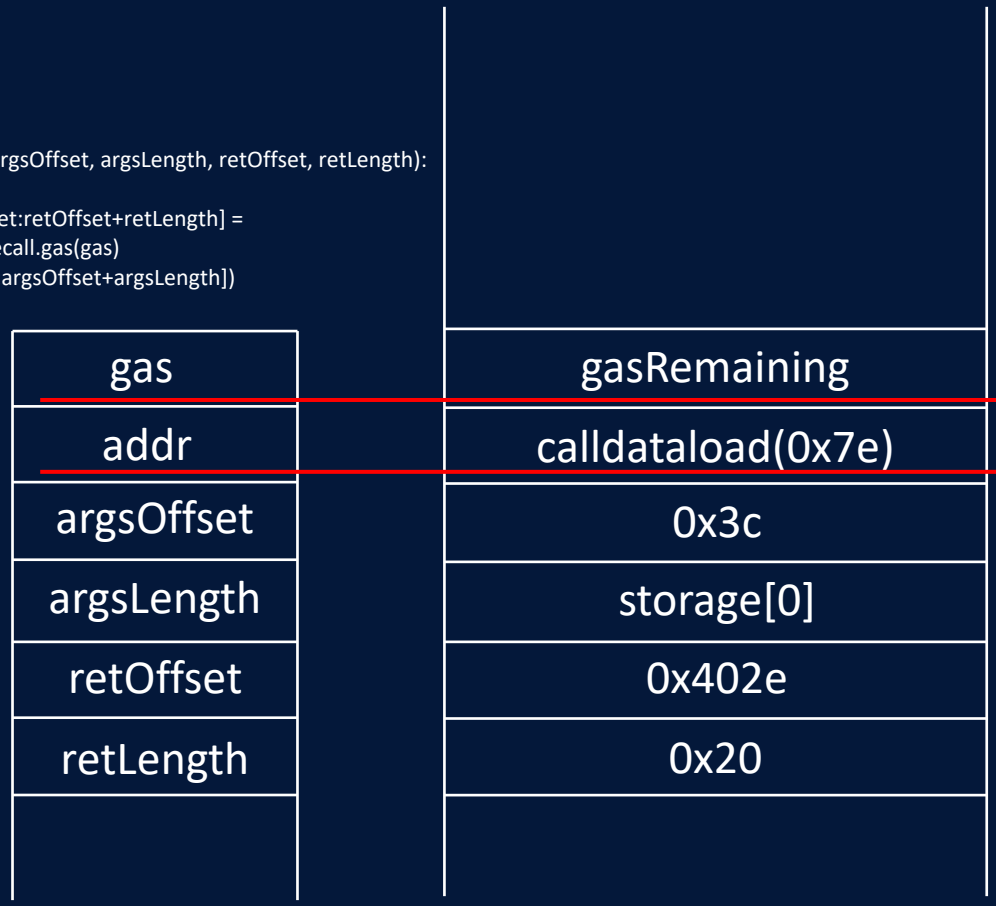
calldata[0x5e:0x7e]=0x09a2

```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```

RE

delegatecall(gas, address, argsOffset, argsLength, retOffset, retLength):

success, memory[retOffset:retOffset+retLength] =
address(addr).delegatecall.gas(gas)
(memory[argsOffset:argsOffset+argsLength])



mem[0x20:0x40]=0x3ccfd60b

calldata[0x5e:0x7e]=0x09a2

gas should be enough

calldata[0x7e:0x9e]=addr

RE

delegatecall(gas, address, argsOffset, argsLength, retOffset, retLength):

success, memory[retOffset:retOffset+retLength] =
address(addr).delegatecall.gas(gas)
(memory[argsOffset:argsOffset+argsLength])

gas
addr
argsOffset
argsLength
retOffset
retLength

gasRemaining
calldataload(0x7e)
0x3c
storage[0]
0x402e
0x20

```
0020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030h: 00 00 00 00 00 00 00 00 00 00 00 00 3C CF D6 0B
```

mem[0x20:0x40]=0x3c cfd60b

calldata[0x5e:0x7e]=0x 09a2

gas should be enough

calldata[0x7e:0x9e]=addr

RE

delegatecall(gas, address, argsOffset, argsLength, retOffset, retLength):

success, memory[retOffset:retOffset+retLength] =
address(addr).delegatecall.gas(gas)
(memory[argsOffset:argsOffset+argsLength])

gas
addr
argsOffset
argsLength
retOffset
retLength

gasRemaining
calldataload(0x7e)
0x3c
storage[0]
0x402e
0x20

```
0020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030h: 00 00 00 00 00 00 00 00 00 00 00 00 3C CF D6 0B
```

mem[0x20:0x40]=0x3c cfd60b

calldata[0x5e:0x7e]=0x 09a2

gas should be enough

calldata[0x7e:0x9e]=addr

Acoraida Monica Puzzle Description

storage[0]=0x04

```solidity
pragma solidity =0.4.25;

contract AcoraidaMonicaGame{
    uint256 public version = 4;
    string public description = "Acoraida Monica admires smart guys, she'd like to pay 10000ETH to the one who could answer her question. Would it be you?";
    string public constant sampleQuestion = "Who is Acoraida Monica?";
    string public constant sampleAnswer = "$*!&#^[` a@.3;Ta&*T` R`<`~5Z`^5V You beat me! :D";
    Logger public constant logger=Logger(0x5e351bd4247f0526359fb22078ba725a192872f3);
    address questioner;
    string public question;
    bytes32 private answerHash;

    constructor(bytes a) { ... }
    modifier onlyHuman{ ... }
    function Start(string _question, string _answer) public payable{ ... }
    function NewRound(string _question, bytes32 _answerHash) public payable{ ... }
    function TheAnswerIs(string _answer) onlyHuman public payable{ ... }
    function () payable {}
}
```

```solidity
contract Logger{
    event WeHaveAWinner(address);
    event NewQuestion(string);
    event NewAnswerHs(bytes32);
    function AcoraidaMonicaWantsToKeepALogOfTheWinner(address winner) public {
        emit WeHaveAWinner(winner);
    }
    function AcoraidaMonicaWantsToKnowTheNewQuestion(string _question) public{
        emit NewQuestion(_question);
    }
    function AcoraidaMonicaWantsToKnowTheNewAnswerHash(bytes32 _answerHash) public {
        emit NewAnswerHs(_answerHash);
    }
}
```

AcoraidaMonicaGame.sol

RE

delegatecall(gas, address, argsOffset, argsLength, retOffset, retLength):

success, memory[retOffset:retOffset+retLength] =
address(addr).delegatecall.gas(gas)
(memory[argsOffset:argsOffset+argsLength])

gas	gasRemaining
addr	calldataload(0x7e)
argsOffset	0x3c
argsLength	0x4
retOffset	0x402e
retLength	0x20

```
0020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030h: 00 00 00 00 00 00 00 00 00 00 00 00 3C CF D6 0B
```

mem[0x20:0x40]=0x3c cfd60b

calldata[0x5e:0x7e]=0x 09a2

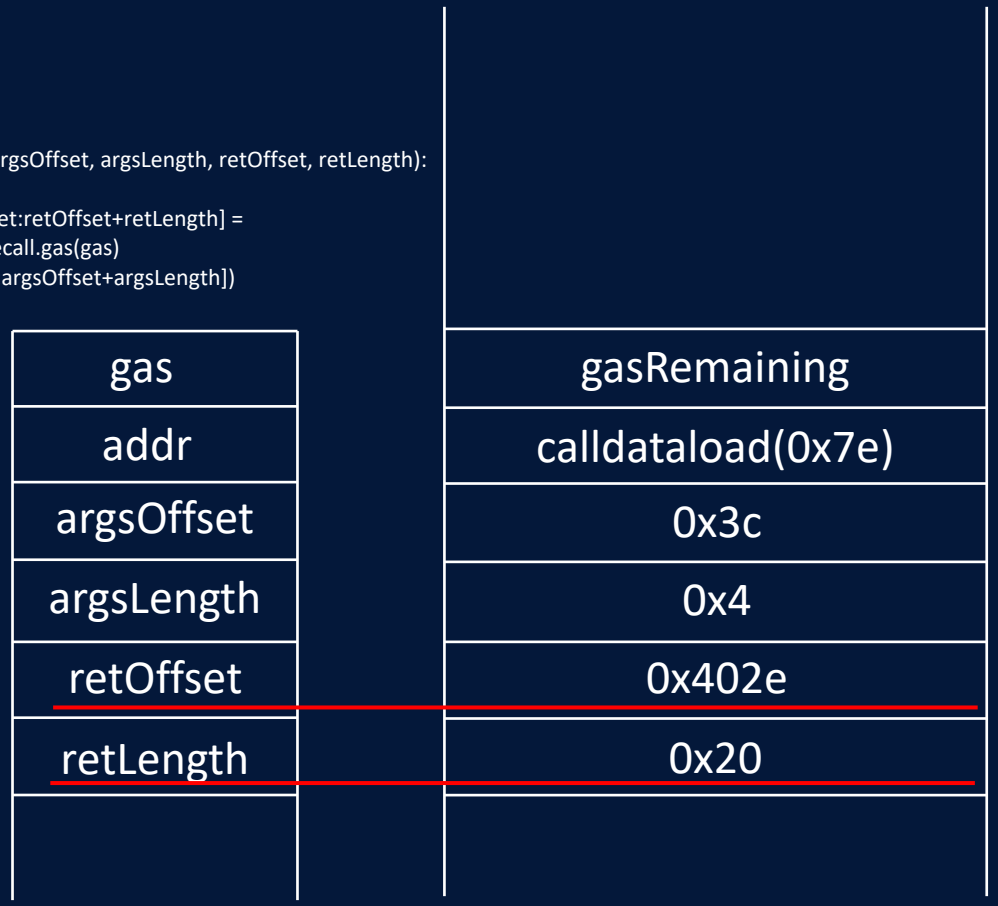
gas should be enough

calldata[0x7e:0x9e]=addr

RE

delegatecall(gas, address, argsOffset, argsLength, retOffset, retLength):

success, memory[retOffset:retOffset+retLength] =
address(addr).delegatecall.gas(gas)
(memory[argsOffset:argsOffset+argsLength])



mem[0x402e:0x404e]=returnValue

calldata[0x5e:0x7e]=0x09a2

gas should be enough

calldata[0x7e:0x9e]=addr

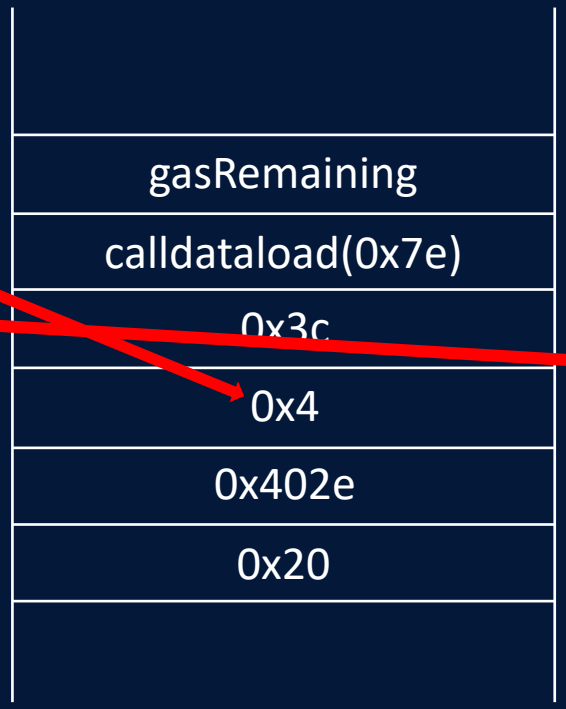
arg: 0x3ccfd60b

RE

Why not simply push data to stack?

```
1503 JUMP
1504 INVALID
1505 INVALID
1506 INVALID
1507 INVALID
1508 INVALID
1509 INVALID
1510 JUMPDEST
1511 PUSH1 20
1513 PUSH2 402e
1516 CALLER
1517 EXTCODESIZE
1518 SLOAD
1519 PUSH2 262a
1522 SLOAD
1523 PUSH1 20
1525 MSTORE
1526 PUSH1 3c
1528 PUSH1 7e
1530 CALLDATALOAD
1531 GAS
1532 PUSH1 5e
1534 CALLDATALOAD
1535 JUMP
```

```
string public constant sampleAnswer = "$*!&#^[` a@.3;Ta&*T`
R`<`~5Z`^5V You beat me! :D";
```



arg: 0x3ccfd60b

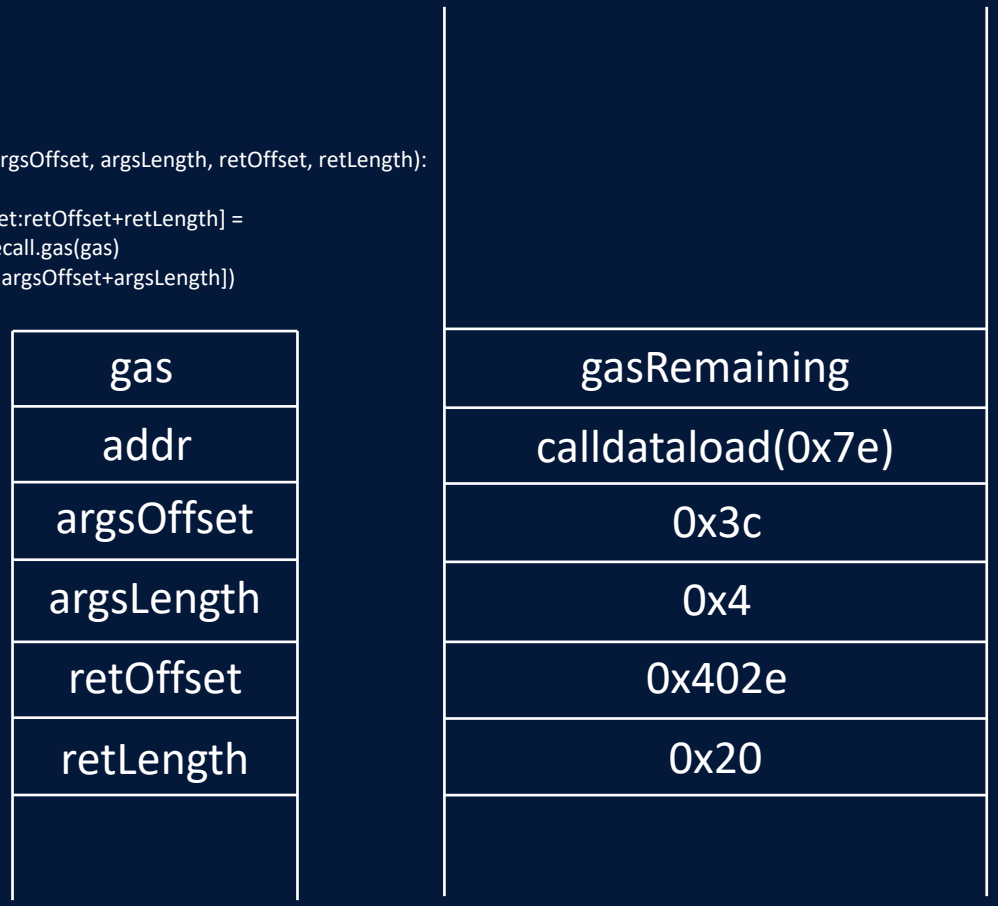
push 0x4 -> 6004

push4 0x3ccfd60b -> 633ccfd60b

RE

delegatecall(gas, address, argsOffset, argsLength, retOffset, retLength):

success, memory[retOffset:retOffset+retLength] =
address(addr).delegatecall.gas(gas)
(memory[argsOffset:argsOffset+argsLength])

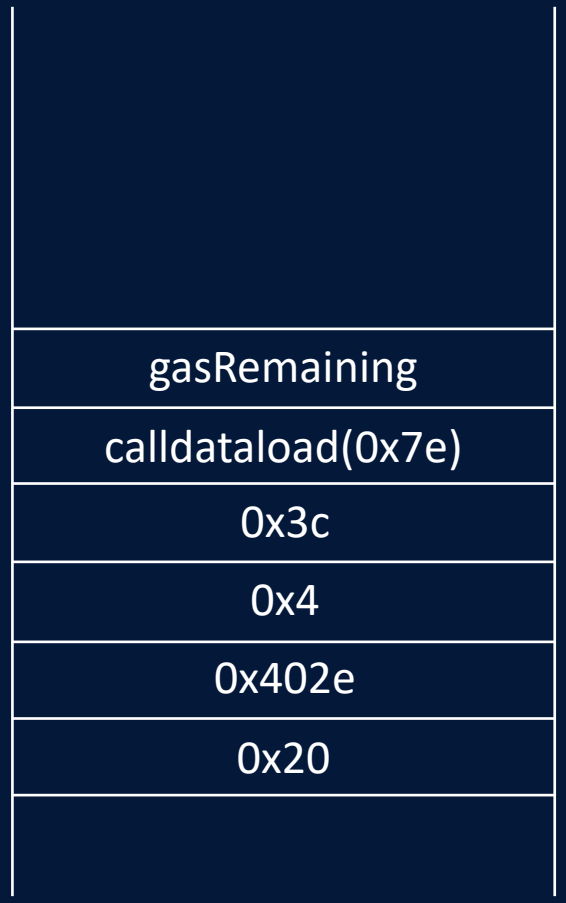


Failed again!

```
function withdraw() payable public returns(uint256){
    msg.sender.transfer(address(this).balance);
}
```

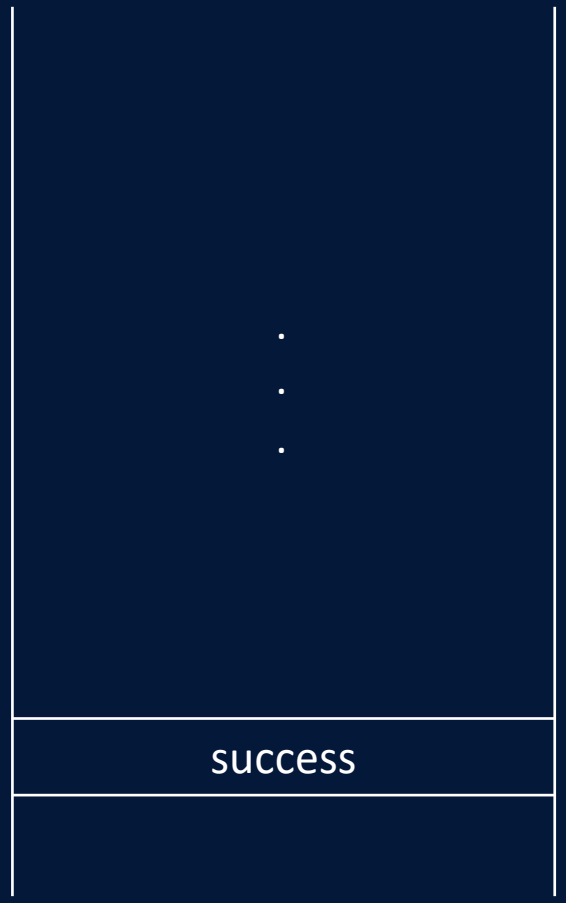
RE

```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```



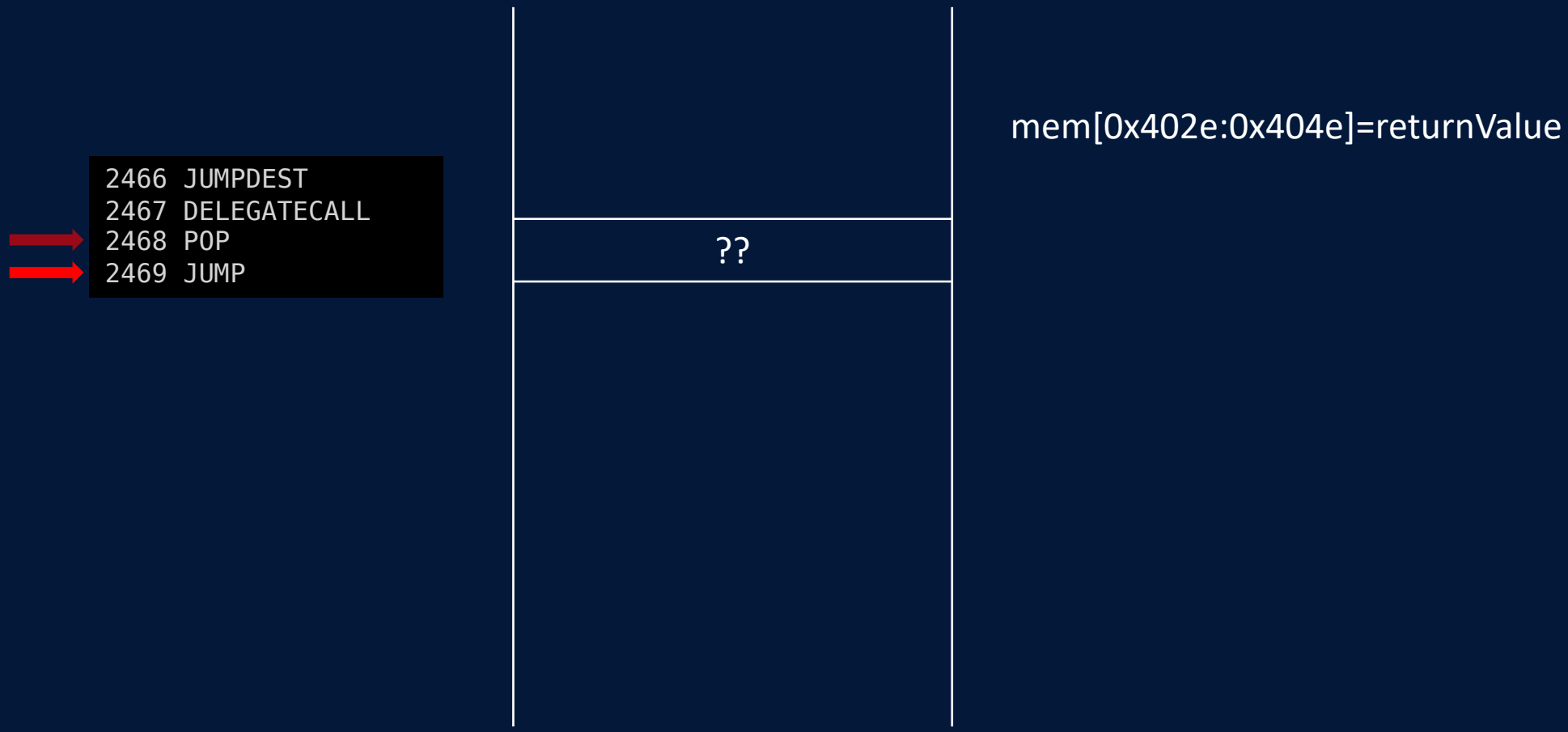
RE

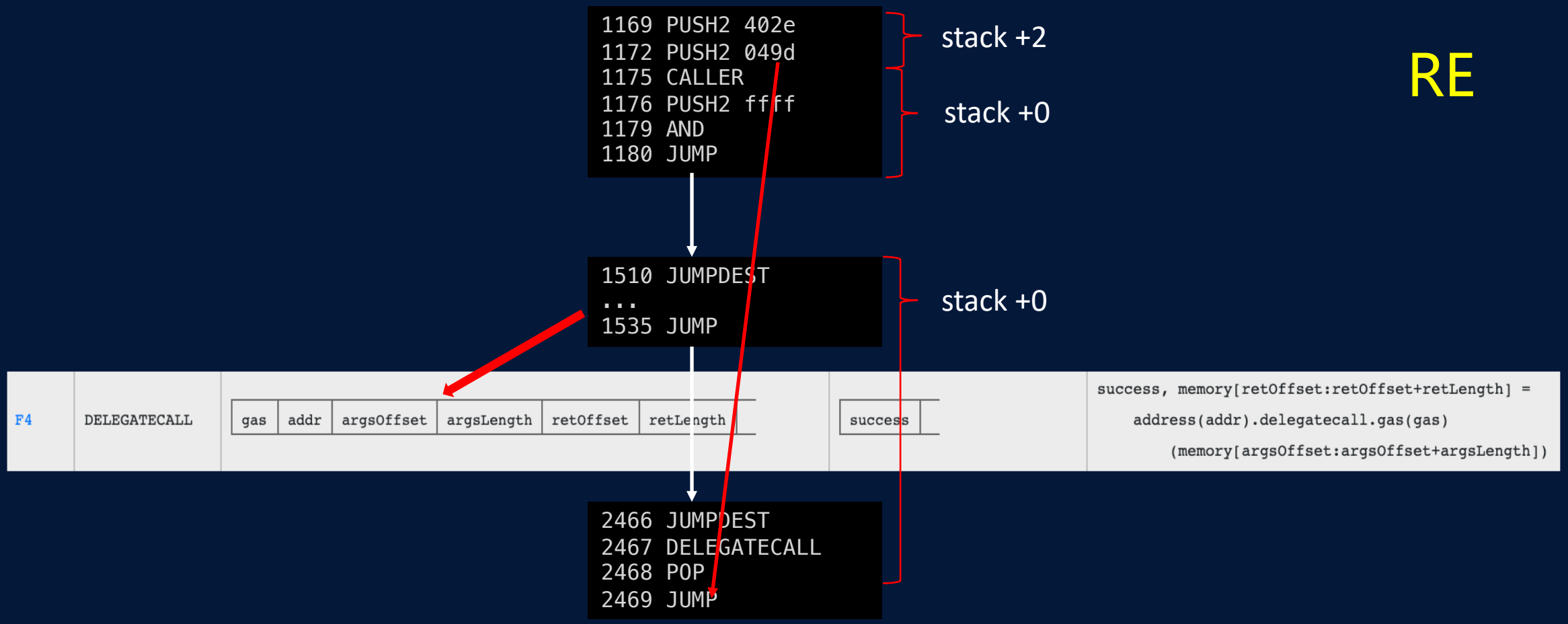
```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```



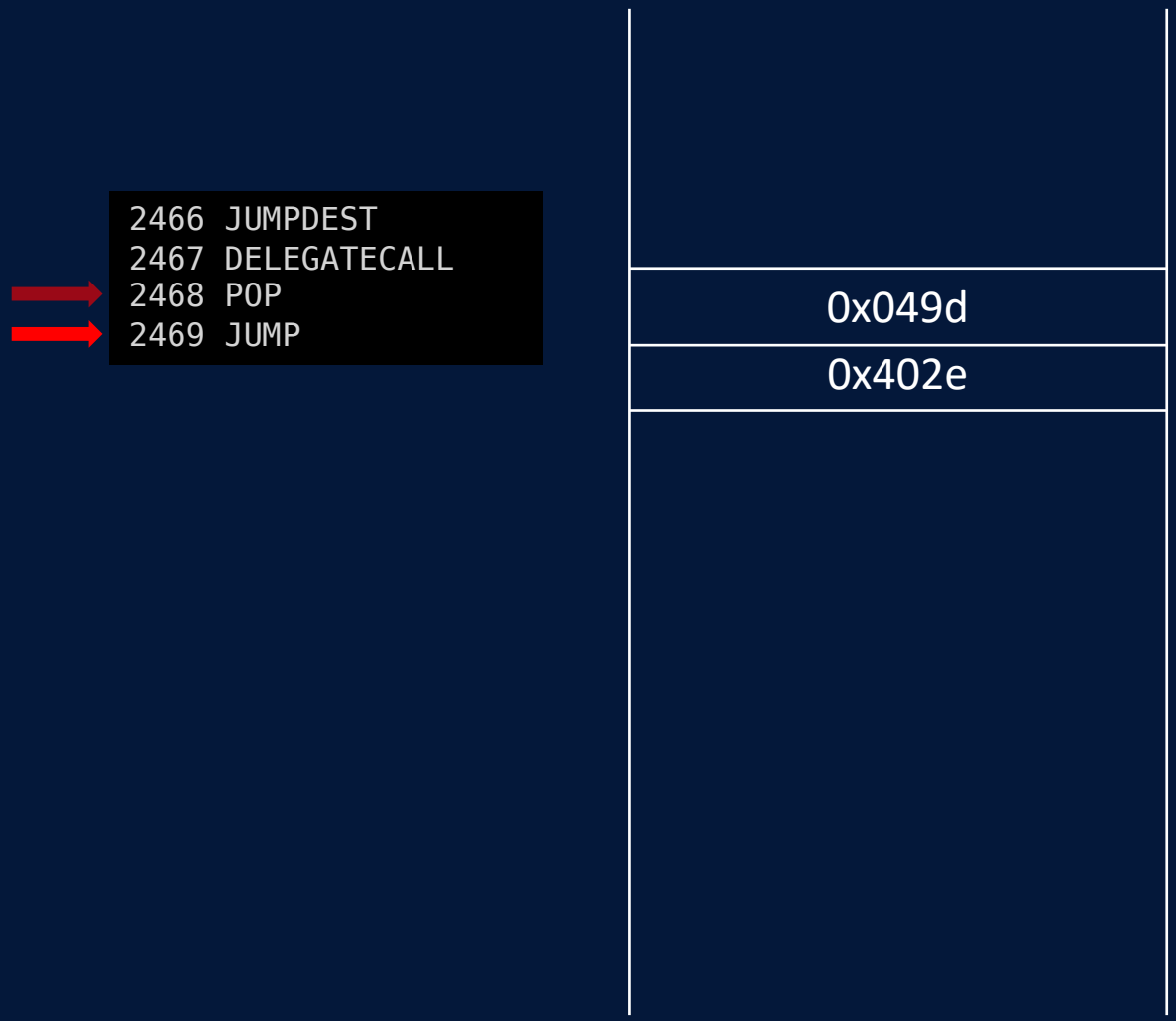
mem[0x402e:0x404e]=returnValue

RE


2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
??
mem[0x402e:0x404e]=returnValue


1169 PUSH2 402e
1172 PUSH2 049d
1175 CALLER
1176 PUSH2 ffff
1179 AND
1180 JUMP
stack +2
stack +0
1510 JUMPDEST
...
1535 JUMP
stack +0
F4
DELEGATECALL
gas
addr
argsOffset
argsLength
retOffset
retLength
success
success, memory[retOffset:retOffset+retLength] =
address(addr).delegatecall.gas(gas)
(memory[argsOffset:argsOffset+argsLength])
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
RE

RE

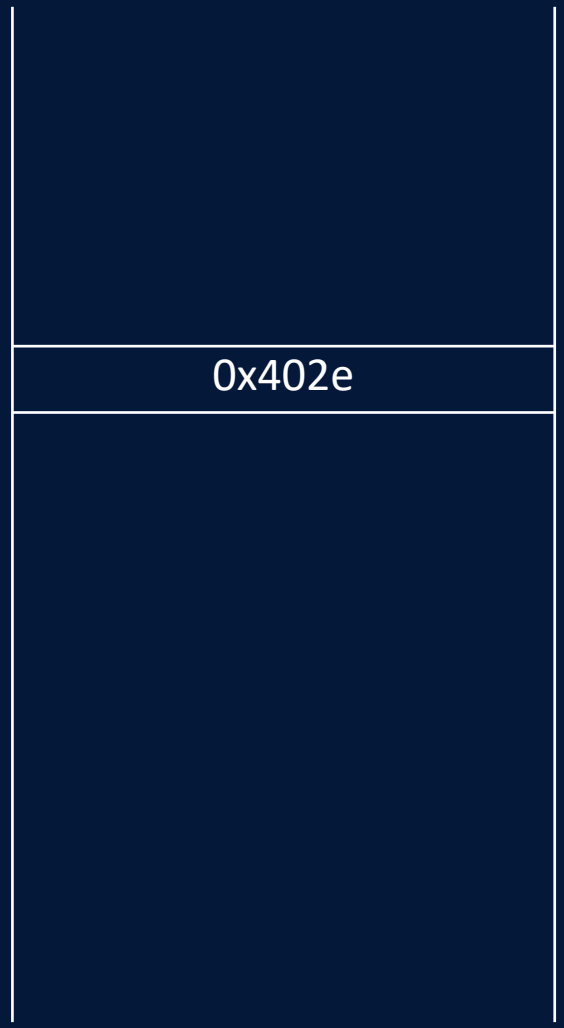

2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
0x049d
0x402e
mem[0x402e:0x404e]=returnValue

RE

```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```

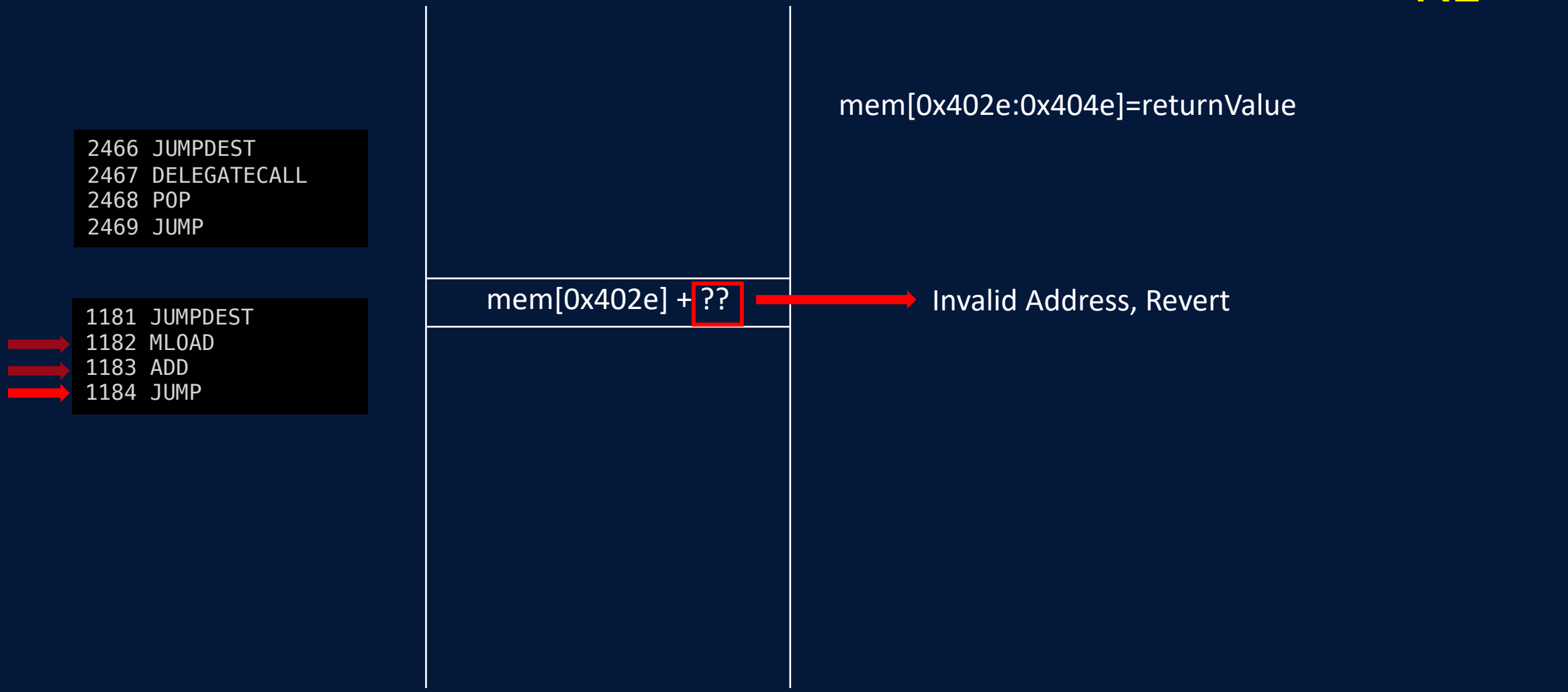
(1181==0x049d)

```
1181 JUMPDEST
1182 MLOAD
1183 ADD
1184 JUMP
```



mem[0x402e:0x404e]=returnValue

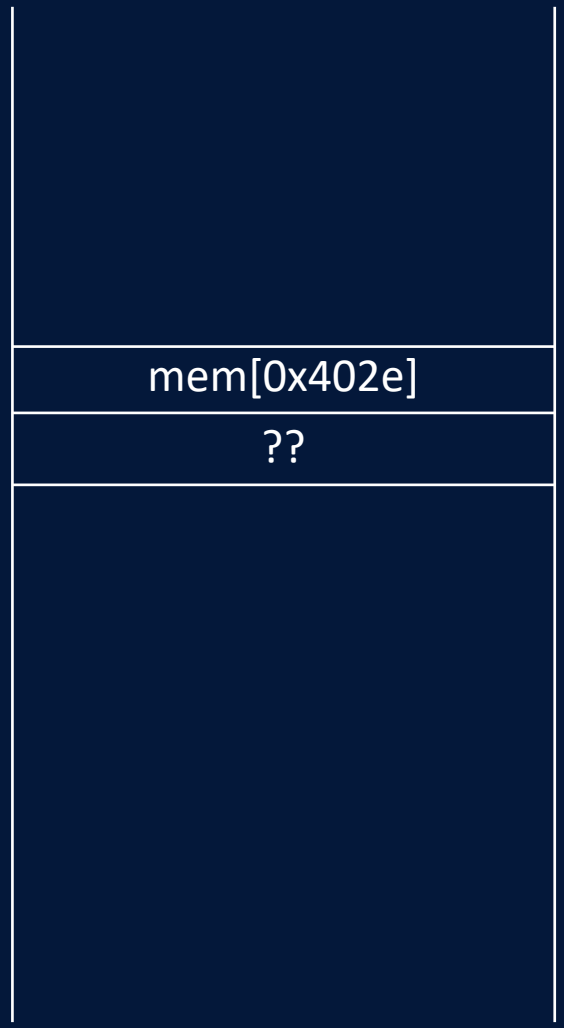
RE


2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
1181 JUMPDEST
1182 MLOAD
1183 ADD
1184 JUMP
mem[0x402e] + ??
mem[0x402e:0x404e]=returnValue
Invalid Address, Revert

RE

```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```

```
1181 JUMPDEST
1182 MLOAD
1183 ADD
1184 JUMP
```




```
function TheAnswerIs(string _answer) onlyHuman public payable{
    //require(msg.sender != questioner);
    if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
        questioner = msg.sender;

        logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
    }
}
```

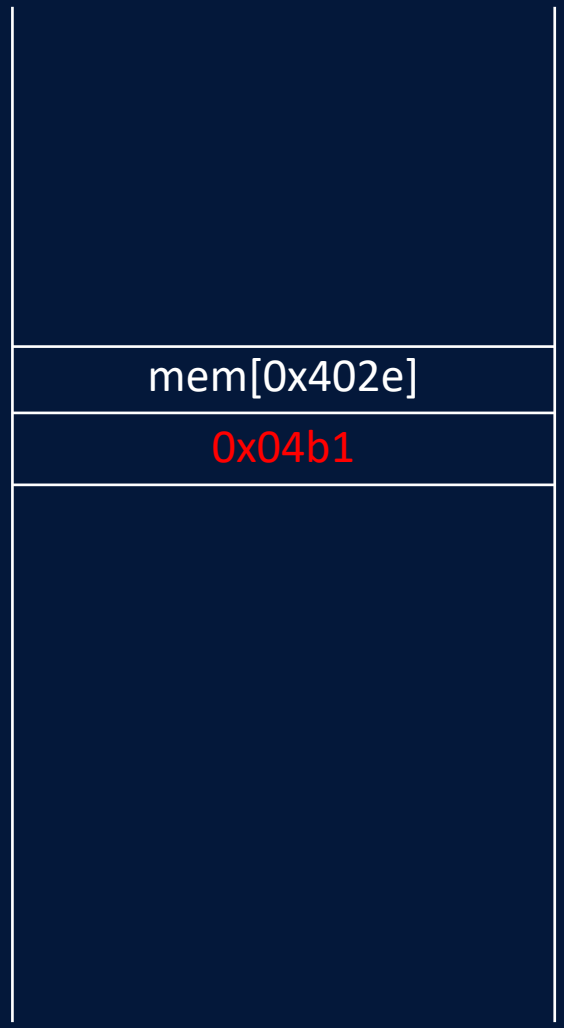
```
push 0x04b1
if (calldatasize == 0xa0){
    push 0x402e
    push 0x049d
    jump to msg.sender & 0xffff
}
```

长亭科技 CHAITIN

RE

```
2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
```

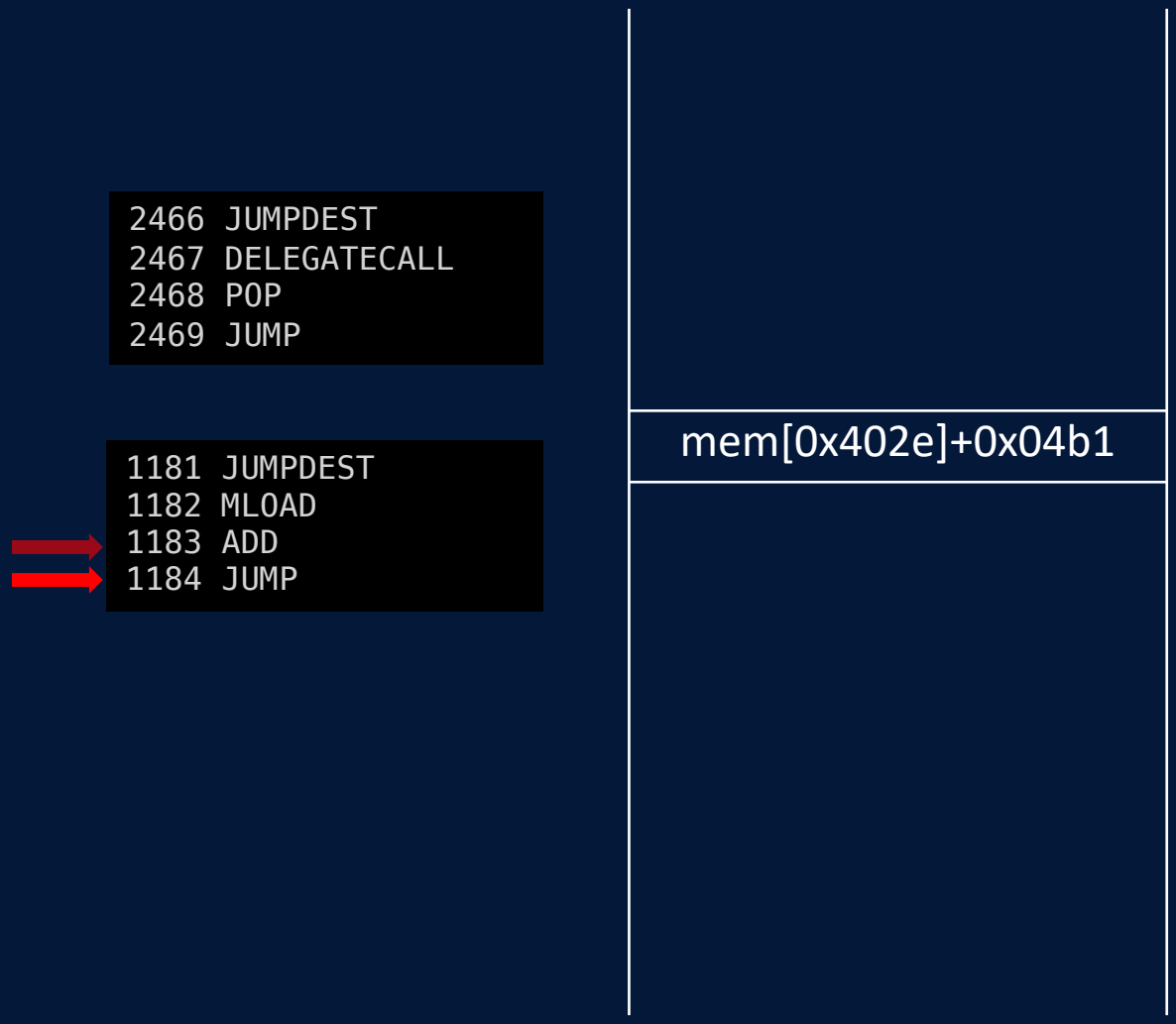
```
1181 JUMPDEST
1182 MLOAD
1183 ADD
1184 JUMP
```



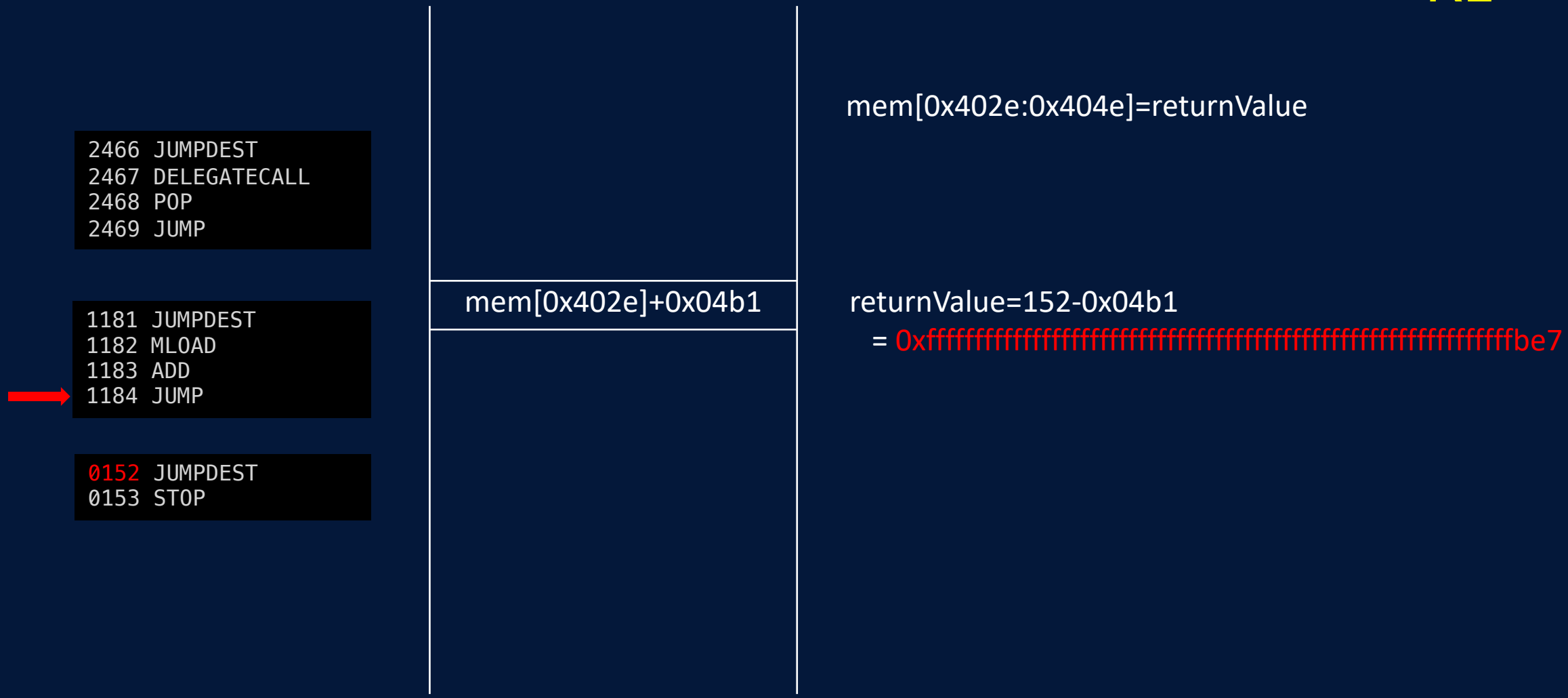

```
function TheAnswerIs(string _answer) onlyHuman public payable{
    //require(msg.sender != questioner);
    if(answerHash == keccak256(_answer) && msg.value >= 1 ether){
        questioner = msg.sender;

        logger.AcoraidaMonicaWantsToKeepALogOfTheWinner(msg.sender);
    }
}
```

```
push 0x04b1
if (calldatasize == 0xa0){
    push 0x402e
    push 0x049d
    jump to msg.sender & 0xffff
}
```

RE


2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
1181 JUMPDEST
1182 MLOAD
1183 ADD
1184 JUMP
mem[0x402e]+0x04b1
mem[0x402e:0x404e]=returnValue

RE


2466 JUMPDEST
2467 DELEGATECALL
2468 POP
2469 JUMP
1181 JUMPDEST
1182 MLOAD
1183 ADD
1184 JUMP
0152 JUMPDEST
0153 STOP
mem[0x402e]+0x04b1
mem[0x402e:0x404e]=returnValue
returnValue=152-0x04b1
= 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffbe7

Final POC

1. call TheAnswerIs(string _answer)	calldata[:4]==0x46a3ec67
2. onlyHuman(extcodesize==0)	call from constructor or non-contract address
3. _answer== "r"	calldata[0x4:0x64]==encode("r")
4. send enough ETH	msg.value >= 1 ether
5. calldatasize == 0xa0	len(calldata)==0xa0
6. first gadget jump	msg.sender&0xffff==0x05e6
7. second gadget jump	calldata[0x5e:0x7e]==0x09a2
8. set delegatecall address	calldata[0x7e:0x9e]==addr
9. calling 0x3ccfd60b	implement withdraw() function
10. gracefully stop, the last gadget jump	return 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffbe7
11. gas should be enough	

encode("r"):
00000000000000000000000000000000000000000000000
00000000000000020000000000000000000000000000000
00000000000000000000000000000000017200000000000
00000000000000000000000000000000000000000000000
0000

Final POC

```
contract poc{
    function withdraw() payable public returns(uint256){
        msg.sender.transfer(address(this).balance);
        return 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffbe7;
    }
}
```

Question: "In Madagascar, you cannot take a picture of a lemur with gray hair. Why?"

True answer "r"

calldata[0x5e:0x7e]==0x09a2

```
data='0x46a3ec67';
data+='0000000000000000000000000000000000000000000000000000000000000020';
data+='0000000000000000000000000000000000000000000000000000000000000001';
data+='7200000000000000000000000000000000000000000000000000000000000000';
data+='00000000000000000000000000000000000000000000000000000000000009a2';
data+='000000000000000000000692a70d2e424a56d2c6c27aa97d1a86395877b3a';
data+='0000';
```

It’s designed as a hint.

Turns out it became a distraction.

Fake answer "You need a camera instead of gray hair."

```
data='0x46a3ec67';
data+='0000000000000000000000000000000000000000000000000000000000000020';
data+='0000000000000000000000000000000000000000000000000000000000000027';
data+='596f75206e65656420612063616d65726120696e737465616420f66207a61';
data+='7920686169722e00000000000000000000000000000000000009a2';
data+='000000000000000000000692a70d2e424a56d2c6c27aa97d1a86395877b3a';
data+='0000';
```

Wrap it up

In this challenge:

4 gadget jumps (including delegatecall)

- *Controlled by sender address, calldata and return data*
- *Gadgets are hidden in string, normal opcode, swarm hash*

5 honeypot techniques:

- *Hidden transaction (internal tx)*
- *Function signature collision (Acorida Monica)*
- *Caller address reference*
- *Runtime code replacement*
- *Jump Oriented Programming (JOP)*

Wrap it up

Special thanks to peter50216(217)
- the only one solved this challenge during the 2-day Realworld CTF, which makes this challenge complete.

Special thanks to liveoverflow
- who made a great video serial to analyze this challenge.

Wrap it up

Etherscan Publish&Verification Process

Wrap it up

Realworld CTF 2019 is Coming in September

Thanks!

Q&A

卡亭科技
CHAITIN

EVM Opcode JOP

Jump Oriented Programming in Smart Contract Honeypot

Xiaohang (Sean) Yu at Chaitin Tech

gmail, twitter: xhyumiracle