How Smart is your Smart Contracts??

By

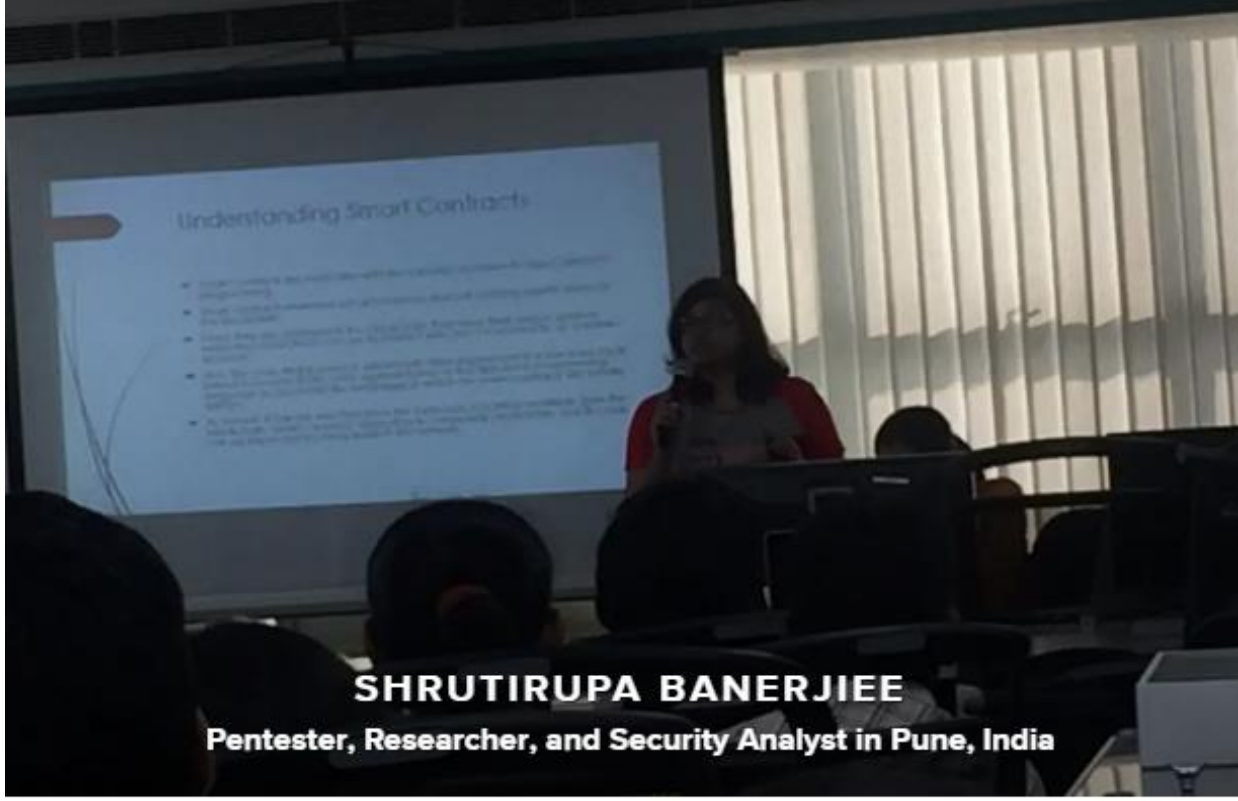
Shrutirupa Banerjiee

$ Whoami??

- WAF Research @ Qualys
- InfosecGirls Pune Chapter Lead
- Blockchain and Security Enthusiast
- Independent Researcher

$locate me

- https://about.me/shrutirupa



</> View my repo

Howdy, I'm Shrutirupa. I'm a security enthusiast living in Pune, India. I am a fan of maths, programming and security. I'm also interested in doing research independently in my areas of interest. You can view my repo with a click on the button above.

Qualys.

Everything visible. Everything secure.

Unparalleled visibility, end-to-end security and compliance for all your global IT assets

The power of the Qualys Cloud Platform

- 2-second visibility across all your assets
- Continuous assessment of your global security & compliance posture
- Identify zero-day vulnerabilities and compromised assets
- Consolidate all your security and compliance stacks
- Secure your digital transformation
- Drastically reduce your spend

Let’s Begin

$ cat Agenda

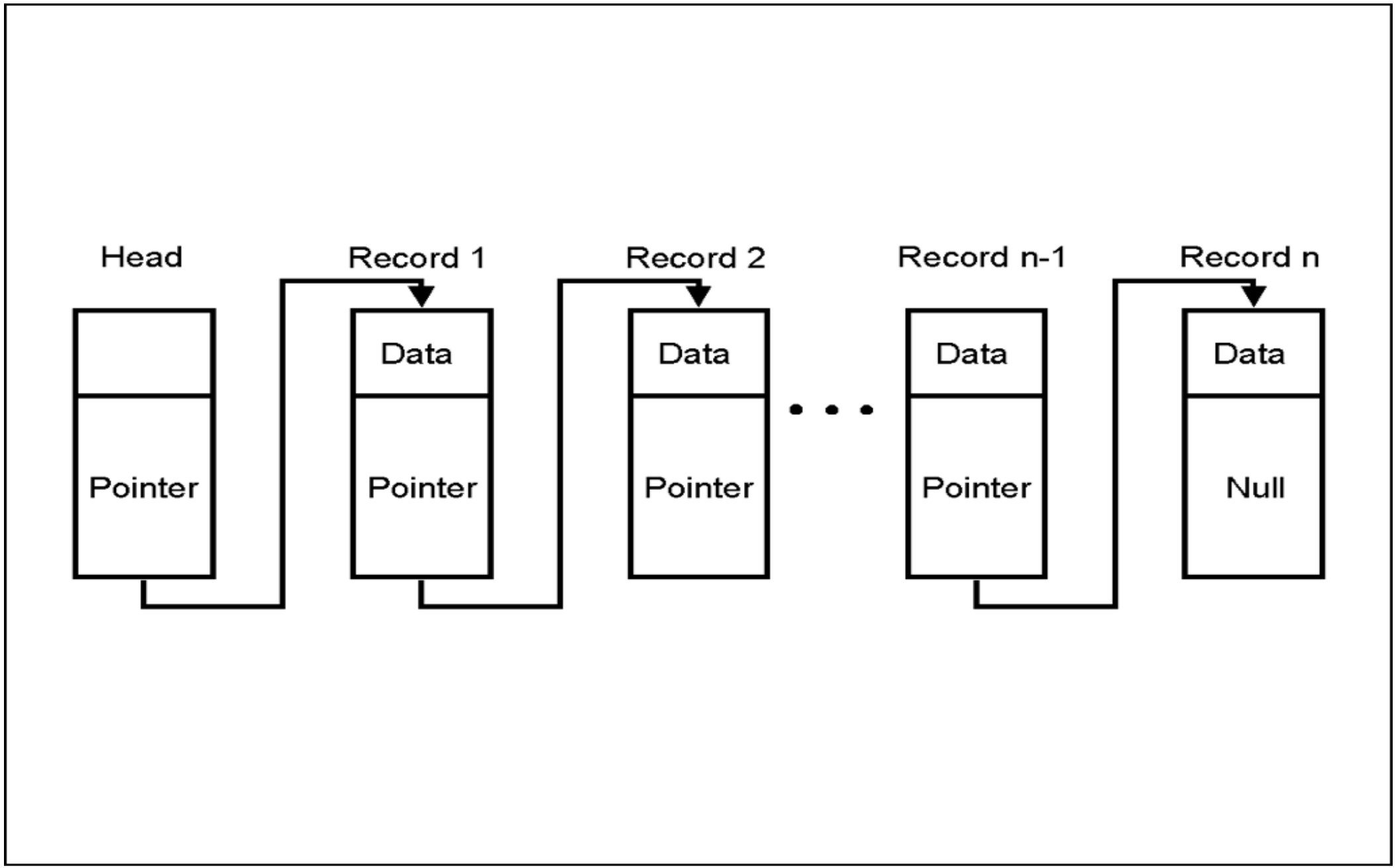
- Blockchain
- Smart Contracts
- Security - Why, How and Where??
- Is your contract smart enough?

Terminologies

- Blockchain
- Ethereum
- Smart Contracts
- Solidity
- Ether
- Gas
- EVM
- Bytecode
- Accounts

$nano Blockchain

- Chain of Blocks
- Blocks are linked together with the help of hashes
- The hash of each block is linked to it’s previous hash


Head
Record 1
Record 2
Record n-1
Record n
Data
Data
Data
Data
Pointer
Pointer
Pointer
Pointer
Null

ethereum

$ man Smart Contract

Piece of code

Written in Solidity (Ethereum based)

SMART CONTRACT

PARTIES SMART CONTRACT EXECUTION

Let us deep dive a bit more

Solidity

High Level Programming Language to write Smart Contracts

Similar to Object Oriented Programming Language

Instead of the keyword “class”, “contract” is used

```
contract Sample {

    int256 num;

    function sample(int256 num1) public  {
        num = num1;
    }


}
```

$man accounts

- Public-private address/key pairs
- Helps you to send/receive ethers

There are two types of accounts:

- *Externally owned – controlled by private keys*
- *Contract – controlled by code*

$man EVM

- Turing complete (partially)
- Read and executes bytecodes

Note: contract code is compiled into bytecode

$man gas

- Execution fee
- Fuel to keep on running the cryptocurrency

Why do we use Smart Contracts?

Creating some kind of logics and conditions

Hence, Dapps(Decentralized Applications)

Why do we need security in Smart Contracts??

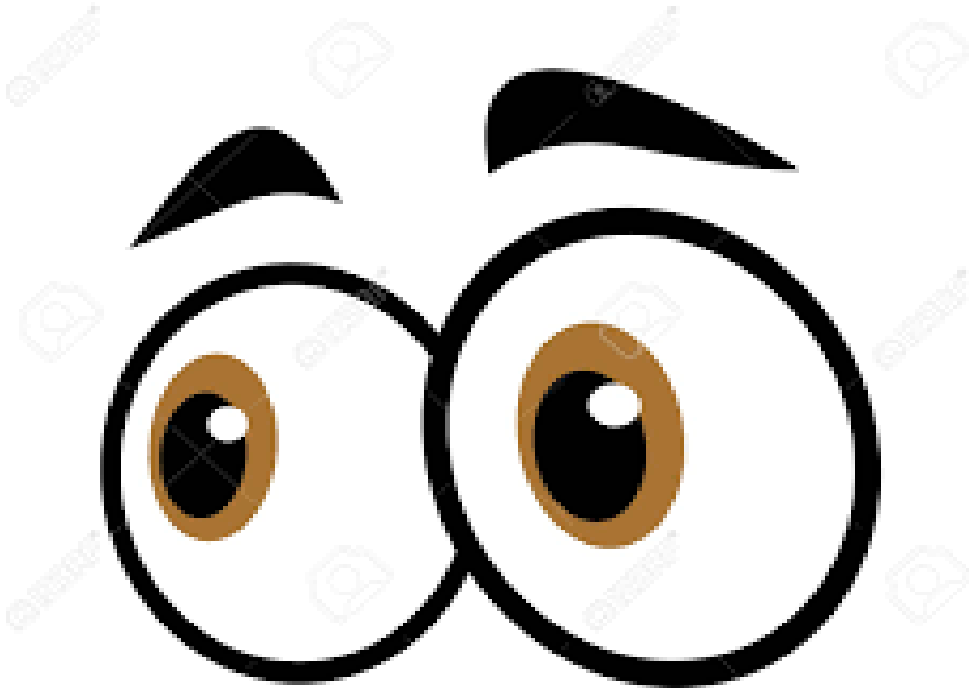
The dapps still are applications which can be vulnerable

Once deployed, can not be reversed

Some famous hacks already present

Areas to focus on!!

Function visibility!!



How many ways can we define visibility?

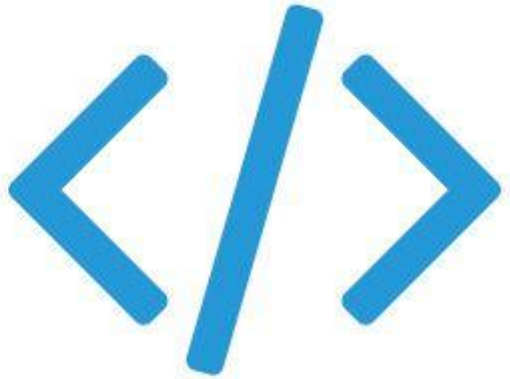
- Public
- Private
- Internal
- External

If not explicitly mentioned, functions are public by default and can be used by external users

Note: private and internal prevents other contracts from accessing and modifying the state variable but is still visible outside the blockchain

So do not write any secret over there!!

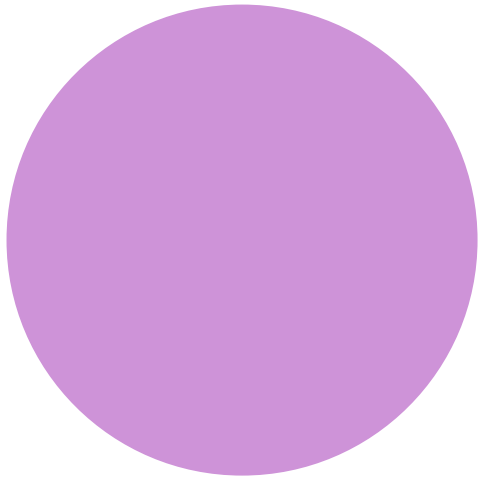
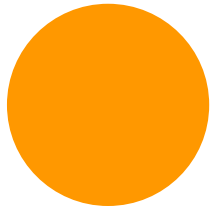
Safemath Library!!



Why do we need safemath library

Integer data-types do not have built-in protection for integer overflow and underflow errors

Unsafe external calls!!

Is it possible to take control over the flow of the contract function repeatedly??

reentrancy

Are you still using the call function?

If yes, have you defined the gas limit at least???

Is the address being validated properly?

Have u checked if it's shorter than the required length?

Is your solidity compiler version below than 0.4.22??

Did you crosscheck the constructor name that you defined?

What else should you look at for your code?

tx.origin

block.timestamp

access modifiers

malicious miners and many more...

Nothing is 100 % Secure

Once in a blockchain, Always in the blockchain

$ find / -name Questions.ask 2 > /dev/freak_crypt

References:

- https://hackernoon.com/hackpedia-16-solidity-hacks-vulnerabilities-their-fixes-and-real-world-examples-f3210eba5148
- http://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/
- https://github.com/crytic/not-so-smart-contracts
- https://blog.openzeppelin.com/on-the-parity-wallet-multisig-hack-405a8c12e8f7/
- https://www.coindesk.com/information/ethereum-smart-contracts-work
- https://yos.io/2018/10/20/smart-contract-vulnerabilities-and-how-to-mitigate-them/
- https://github.com/smartdec/classification

Bsides singapore

- Thanks for the opportunity


Thank
You