Immunefi

How to Earn Millions in Web3 Bug Bounties

Arbaz Hussain & Nemveer

Bsides Ahmedabad, October 12th 2024

Introduction

- Smart contract triagers at @immunefi
- Interested in investigating hacks on blockchain.
- Previously:
  - Nemveer : Independent web3 BH
  - Arbaz : web2 appsec engg
- Create educational content.

ArbazKiraak

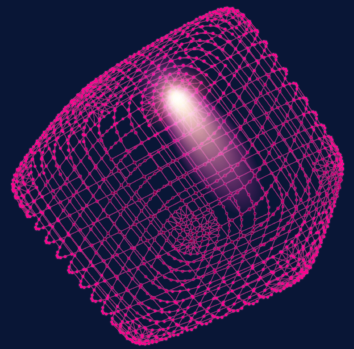
nem_veer

@immunefi-team/bugfix-reviews-pocs

https://immunefi.medium.com

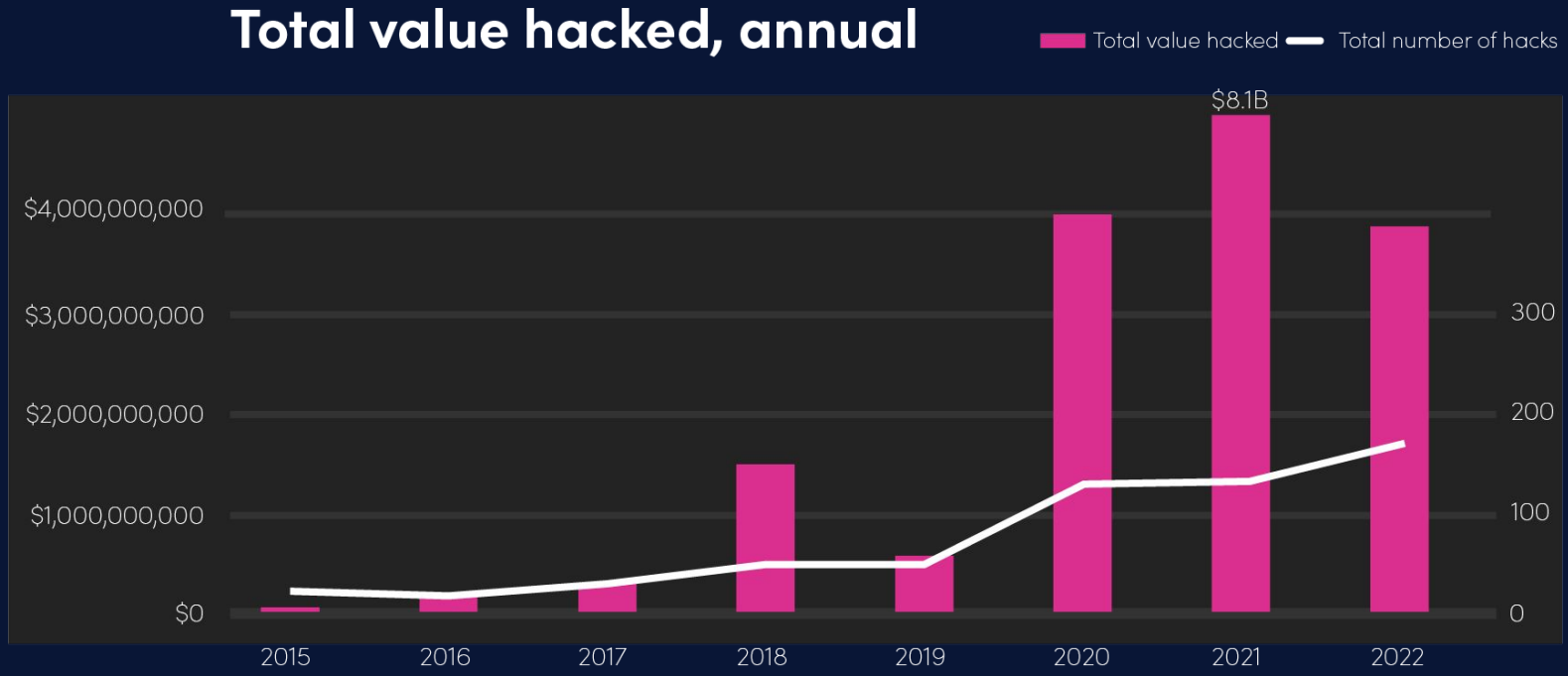
Agenda

- Why are bug bounties in web3 so crucial?
- Key differences between Web2 and Web3
  - Architecture Overview
  - Bug bounties Comparison
- How the top Security Researchers use the Immunefi Platform to make millions
- Anatomy of high impact bug reports on Immunefi
- How to get started with bug bounties in web3
  - Roadmap to bug hunting in web3
  - Resources for you to get started



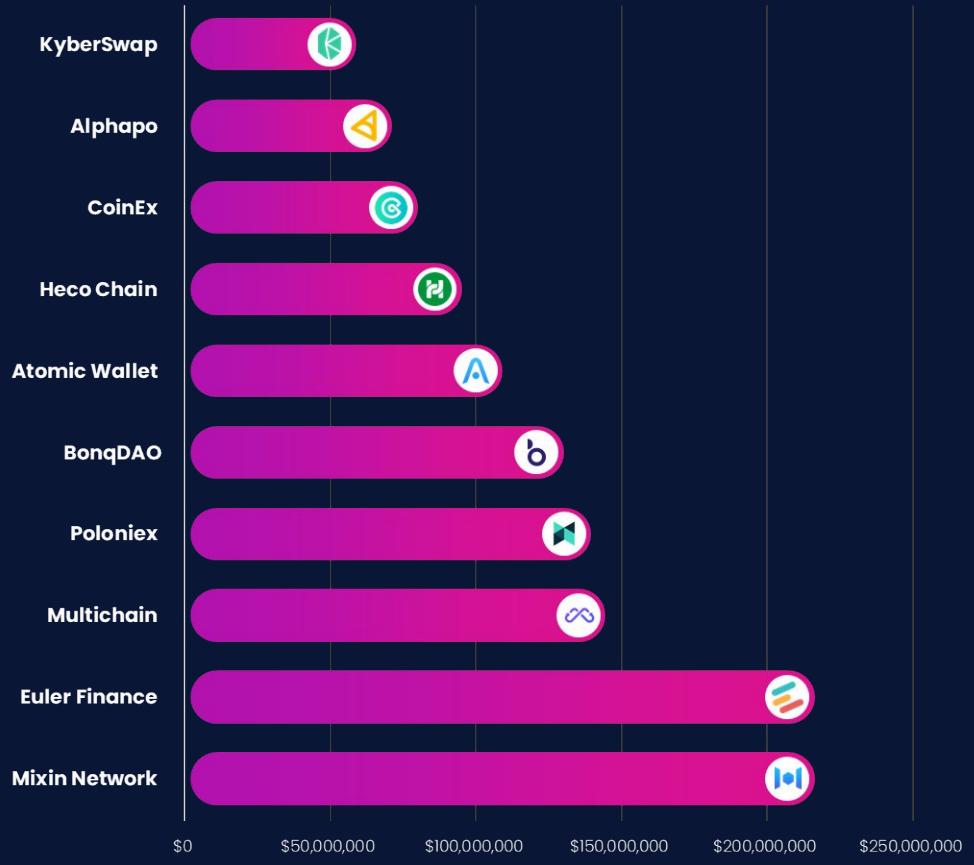
Why Bug Bounties are Crucial in Web3

Hacks are way more common in web3...



Source: Immunefi Research

$1.8 Bn losses in 2023 in Blockchain



Source: Immunefi Research

Some of the biggest hacks in web3

Nomad bridge hack ~ $200M

Technology

Crypto Bridge Nomad Drained of Nearly $200M in Exploit

The exploit calls the security of cross-chain token bridges into question once again.

By Sam Kessler, Brandy Betz Aug 2, 2022 at 9:05 a.m. Updated May 11, 2023 at 9:26 p.m.

- A cross-chain bridge enabling the transfer of tokens and data between blockchains
- A recent update to one of Nomad's smart contracts made it easy for users to spoof transactions.
- This meant users were able to withdraw money from the Nomad bridge that didn't actually belong to them.

Some of the biggest hacks in web3

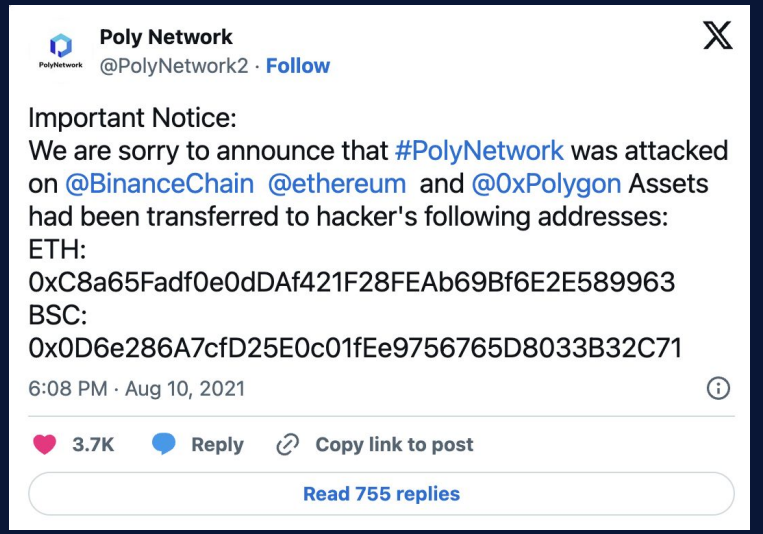
Polynetwork hack ~ $600M

Markets

Cross-Chain DeFi Site Poly Network Hacked; Hundreds of Millions Potentially Lost

DeFi platform Poly Network was attacked on Tuesday, with the alleged hacker draining roughly $600 million in crypto.

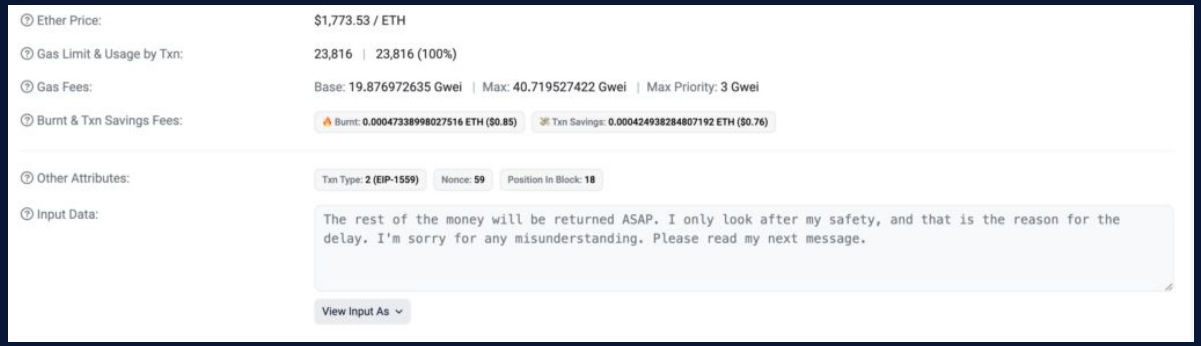
By Eliza Gkritsi, Muyao Shen Aug 10, 2021 at 7:26 p.m. Updated Sep 14, 2021 at 7:07 p.m.



- Amongst the largest hacks in crypto history
- Hacker managed to steal $611m taking $273m in Ethereum, $253m in Binance Smart Chain and $85m in Polygon.
- Attacker created malicious parameters to bypass bridge's validation process and withdraw tokens from the bridge to their own address.

Some of the biggest hacks in web3

Eular hack ~ $197M



- One of the most sophisticated exploit
- Attack was made possible by a vulnerability in the EToken smart contract of Eular, lacking some critical validations.
- At the end, hacker returned stolen funds, marking one of the largest DeFi recoveries

- Not enough security awareness
- Continuous adoption
- Rapidly evolving landscape of Web3 leads to the swift emergence of new vulnerabilities and attack vectors
- Security is an always-on requirement post security audit

A Bug Bounty program is the last line of defence for projects.

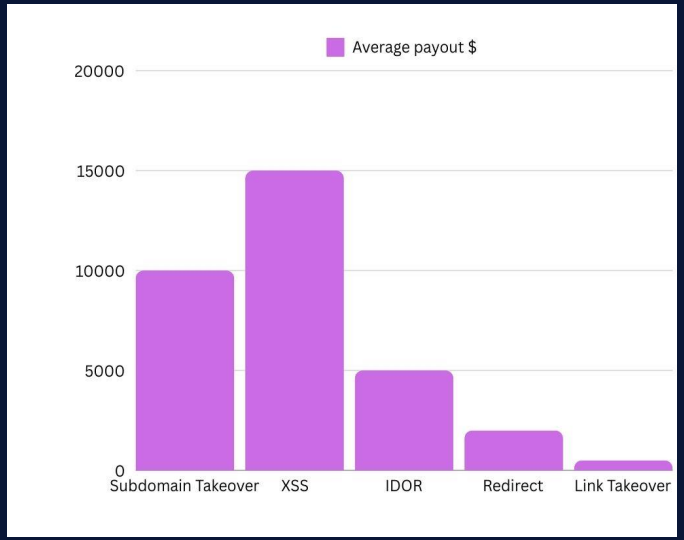
Difference in Web2 & Web3 Bug Bounties

It's quite simple... Let us show you how

Top 5 Most reported Web2 Attack Vectors on Immunefi!

Average bounties for each attack vector

- Subdomain Takeovers : **$10k**
- Cross Site Scripting (XSS) : **$15k**
- Insecure Direct Object References (IDOR) : **$5k**
- Open Redirect : **$3k**
- Outgoing Broken Link Takeovers : **$1k**



https://immunefi.com/immunefi-vulnerability-severity-classification-system-v2-2/

Aspect	WEB2	WEB3
What's at stake	Personal data (e.g PII)	Tokens or Assets
Impact severity	Higher severity with greater impact of user data	Immediate financial loss due to instant access to assets
Bounty Amount	Generally Lower	Higher, due to direct value at risk
Hacker Traceability	Easier to trace hackers in centralized systems	Challenging due to decentralised system
Vulnerability Emergence	Slower due to established security practices	Rapid leading to new attack vectors introducing vulnerabilities

Impact outcome of XSS

- Stored Cross SIte Scripting (XSS)
- Substituting the contract addresses.
- Modifying transaction arguments or parameters.

Severity stands **critical** considering the digital assets at risk.

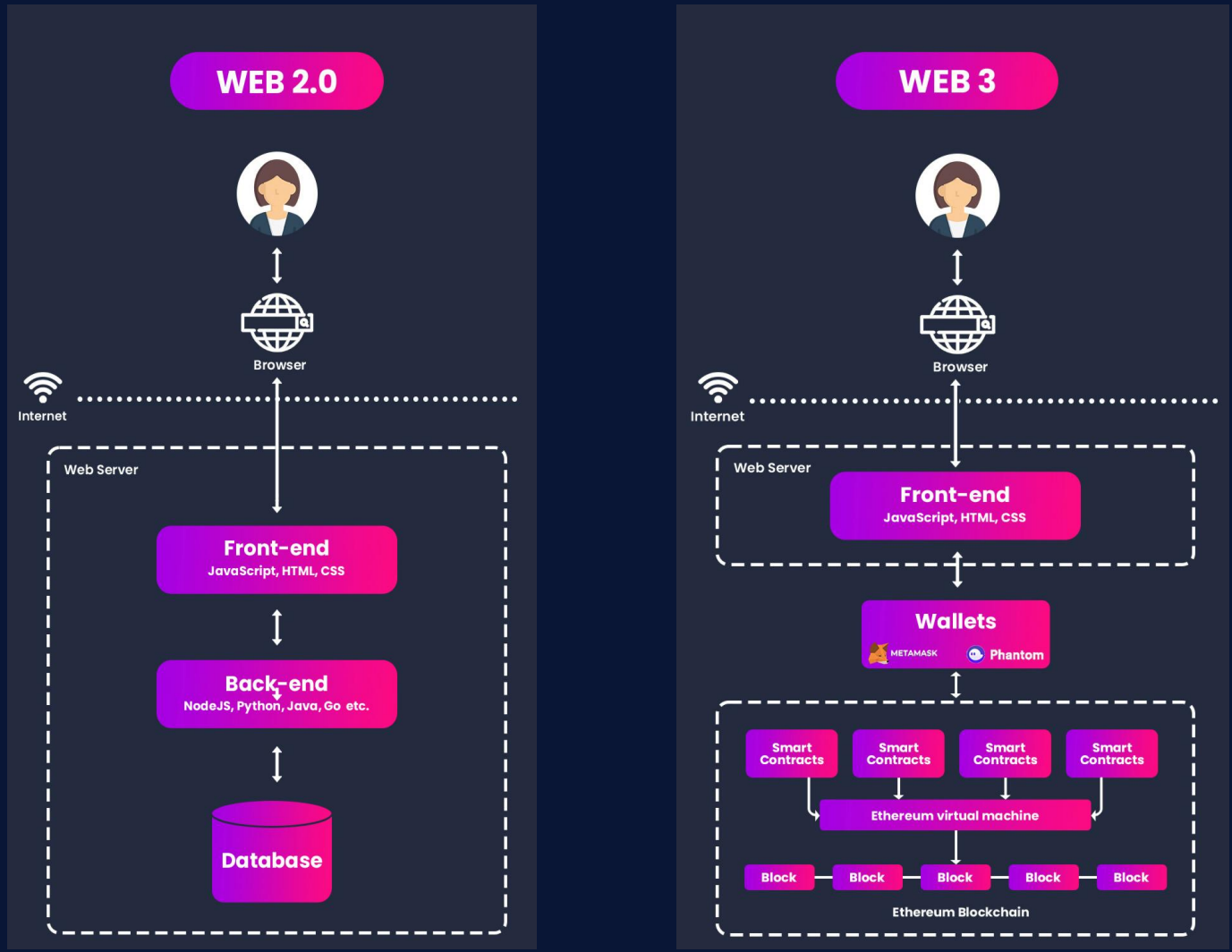
BadgerDAO Reveals Details of How It Was Hacked for $120M

The DeFi platform said an application platform that runs on its cloud network was the vector for the attack.

By Nelson Wang Dec 11, 2021 at 4:55 a.m. Updated Dec 12, 2021 at 10:09 p.m.

source: coindesk

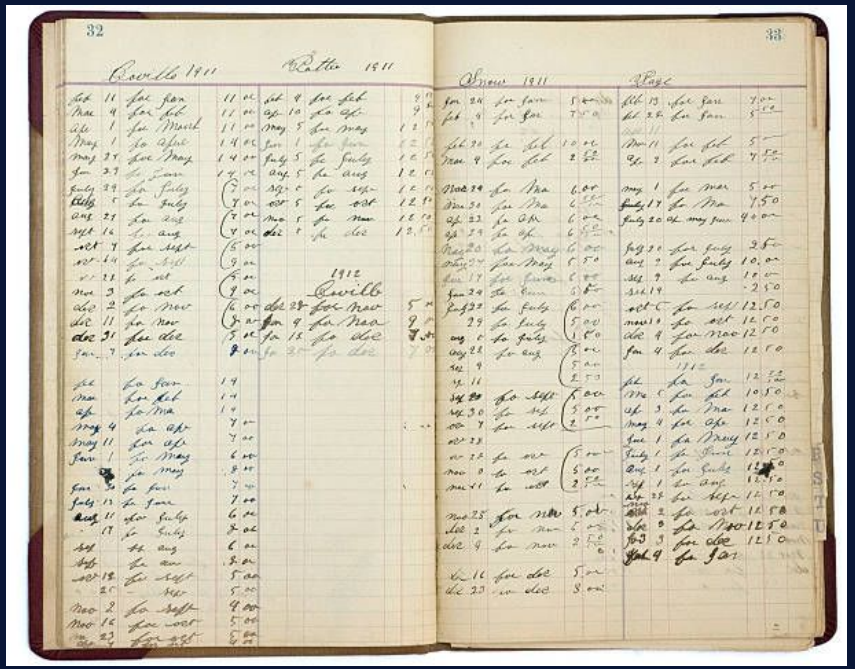
Web2 vs Web3 Architecture


WEB 2.0
Browser
Internet
Web Server
Front-end
JavaScript, HTML, CSS
Back-end
NodeJS, Python, Java, Go etc.
Database
WEB 3
Browser
Internet
Web Server
Front-end
JavaScript, HTML, CSS
Wallets
METAMASK
Phantom
Smart Contracts
Smart Contracts
Smart Contracts
Smart Contracts
Ethereum virtual machine
Block
Block
Block
Block
Block
Ethereum Blockchain

Evolution of Blockchain

Early money as ledger

Proto-cuneiform tablet, Uruk, ca 3100–2900 B.C.



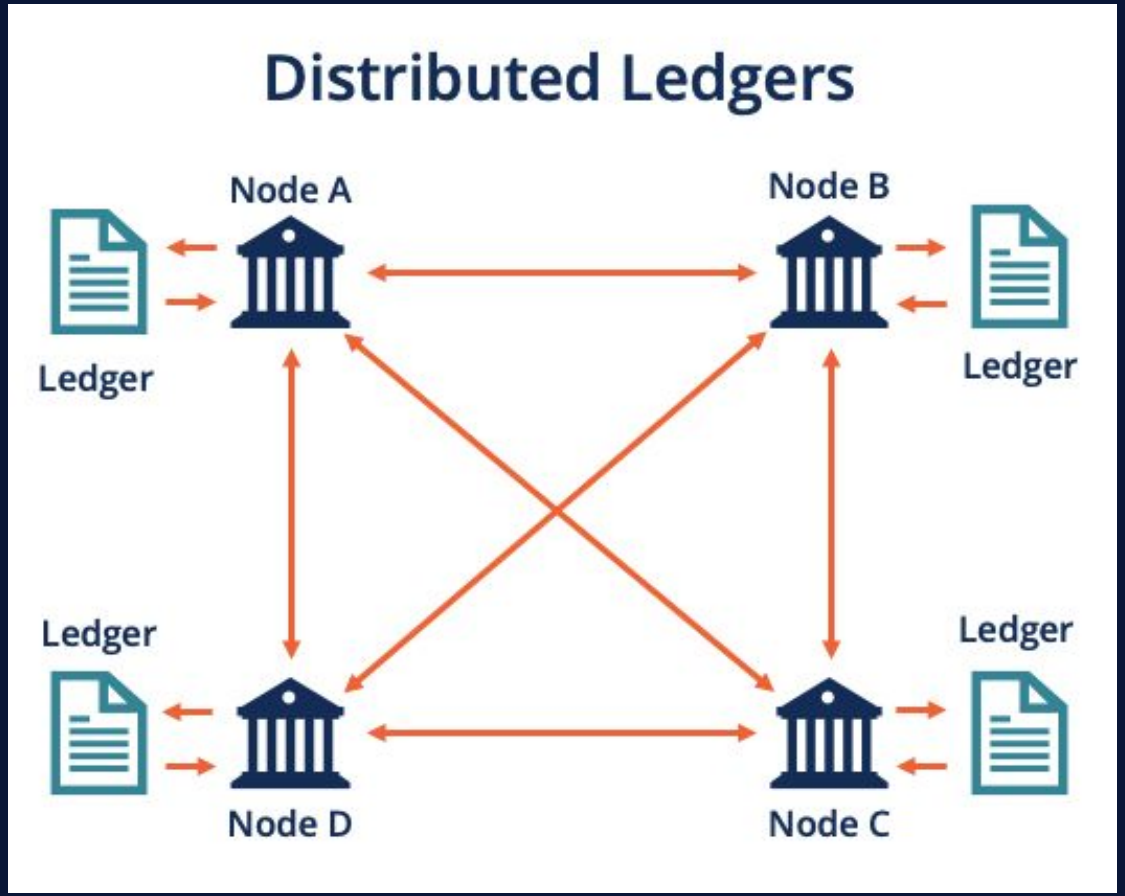
Old ledger from 1911 stock photo

Alice pays Bob $20
Bob pays Casey $40
Casey pays Daniel $30
Daniel pays Alice $10

Alice
Bob
Casey
Daniel

Common Ledger

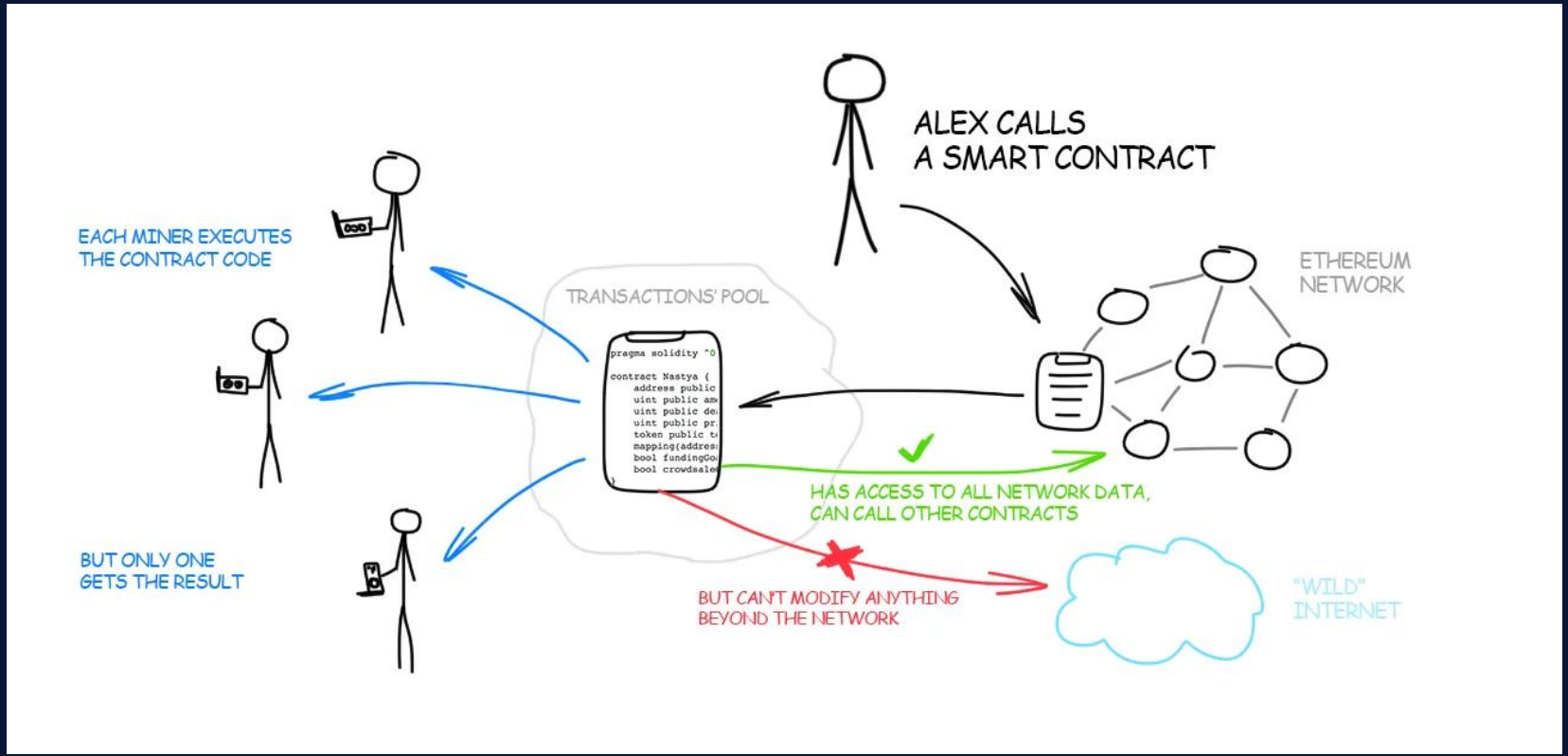
Everyone has a copy of the same ledger



Problems are soluble

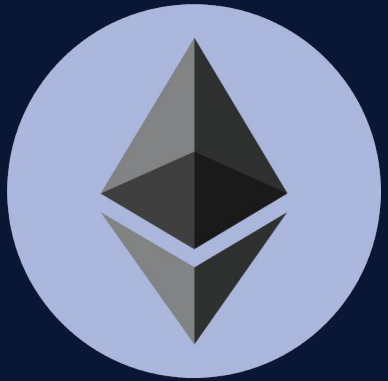
- Proof of work
- Cryptographic Hash Functions & Digital Signatures
- Merkle trees
- Blockchain
- Native currency

Ethereum – programmable currency

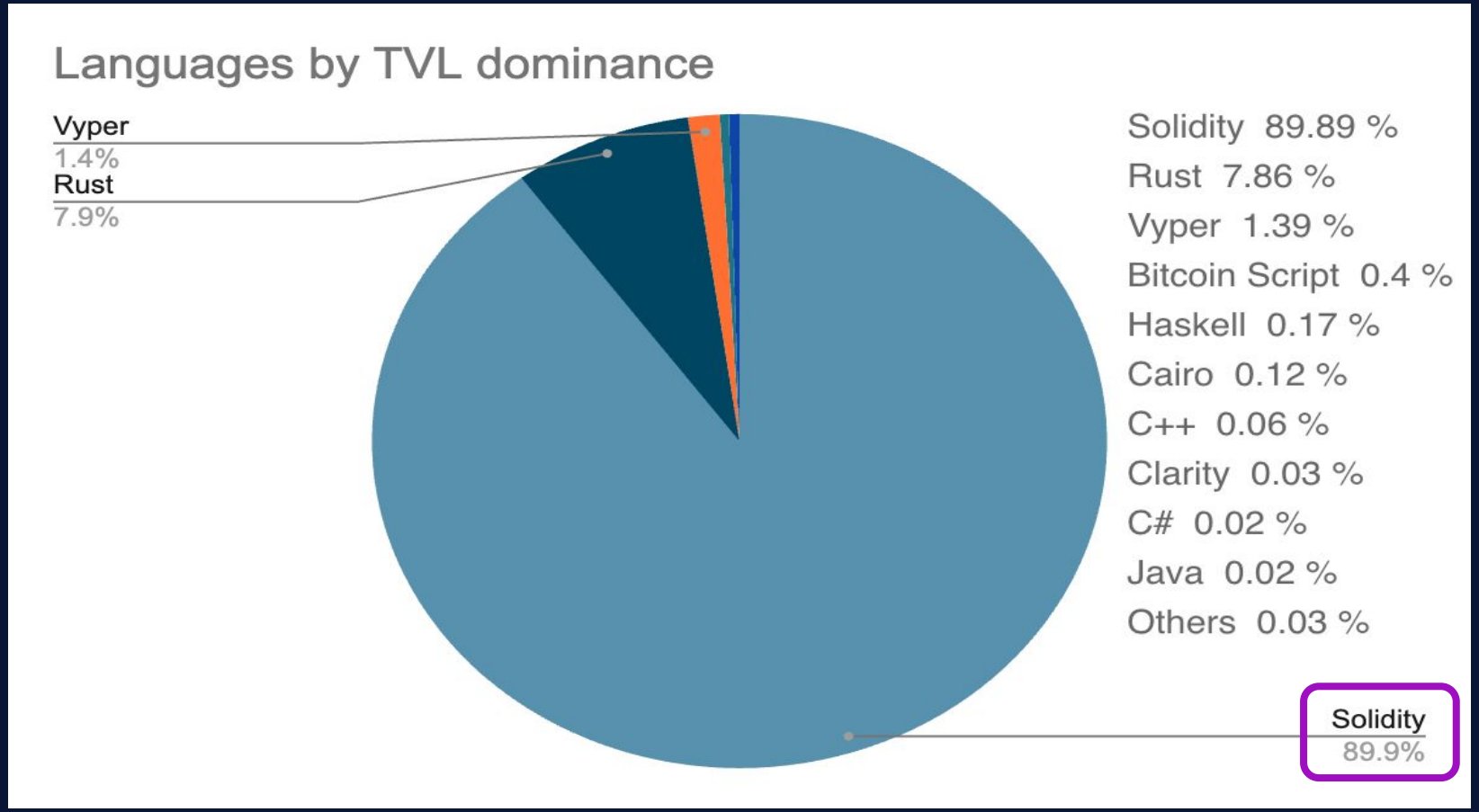


⛓️💰 Web3

- Multi-sigs +$100B
- Autonomous Market Makers
- Decentralized borrowing and lending, flash loans
- **Decentralized Finance**



Languages used for smart contracts



Differences in Web2 & Web3 Languages

Language	Blockchain	Similar to...
Solidity	Arbitrum, BNB, Ethereum, Polygon, Optimism	Javascript
Rust	Solana, Polkadot, Cosmos	C, C++
Vyper	Same as Solidity	Python

Solidity

```
contract SSimpleStorage {
    uint256 storedNumber;

    function storeNumber(uint256 newNumber) external {
        storedNumber = newNumber;
    }

    function readNumber() external view returns (uint256) {
        return storedNumber;
    }
}
```

Rust

```
struct NumberStorage(Option<u32>);

impl NumberStorage {
    fn new() -> Self { Self(None) }
    fn store_number(&mut self, value: u32) { self.0 = Some(value); }
    fn read_number(&self) -> Option<u32> { self.0 }
}

fn main() {
    let mut storage = NumberStorage::new();
    storage.store_number(42);
    println!("Stored number: {:?}", storage.read_number());
```

Vyper

```
storedNumer: uint256

@external
def storeNumber(newNumber: uint256):
    self.storedNumer = newNumber

@external
@view
def readNumber() -> uint256:
    return self.storedNumer
```

How Immunefi bug bounty platform works?

Bug bounties and Immunefi

- **Code is Law, Code is Money**
  - In DeFi and blockchain, exploited bugs translate directly into financial loss.
  - Bug bounties serve as a frontline defense in web3.

- **Immunefi: Leading the Charge**
  - Hosts the largest bug bounty program in DeFi and blockchain.
  - Attracts top security talent with substantial rewards.
  - Aligns incentives to convert potential blackhats into whitehats.

Immunefi

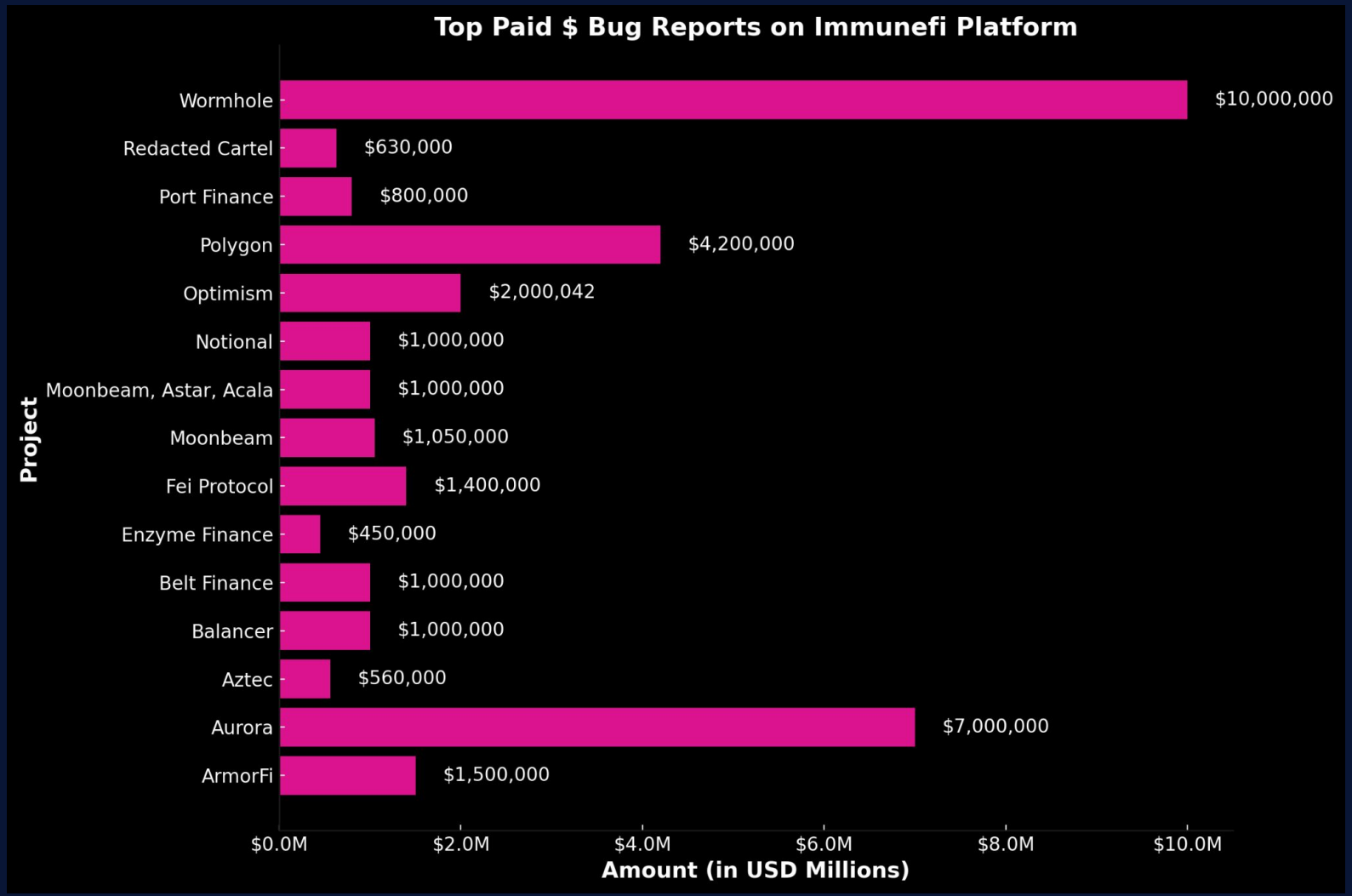
Leading Blockchain security with largest and most proficient community of elite security researchers safeguarding over 300+ leading Blockchain projects

45k	1000+	$100M	+52%	$25B	$60B
Largest Blockchain Security Community	SRs reported bugs to avert mainnet theft of funds	Bounties Paid	Report Hit Rate* on Impact Focused Bugs	Loss Prevented	Users Funds Protected

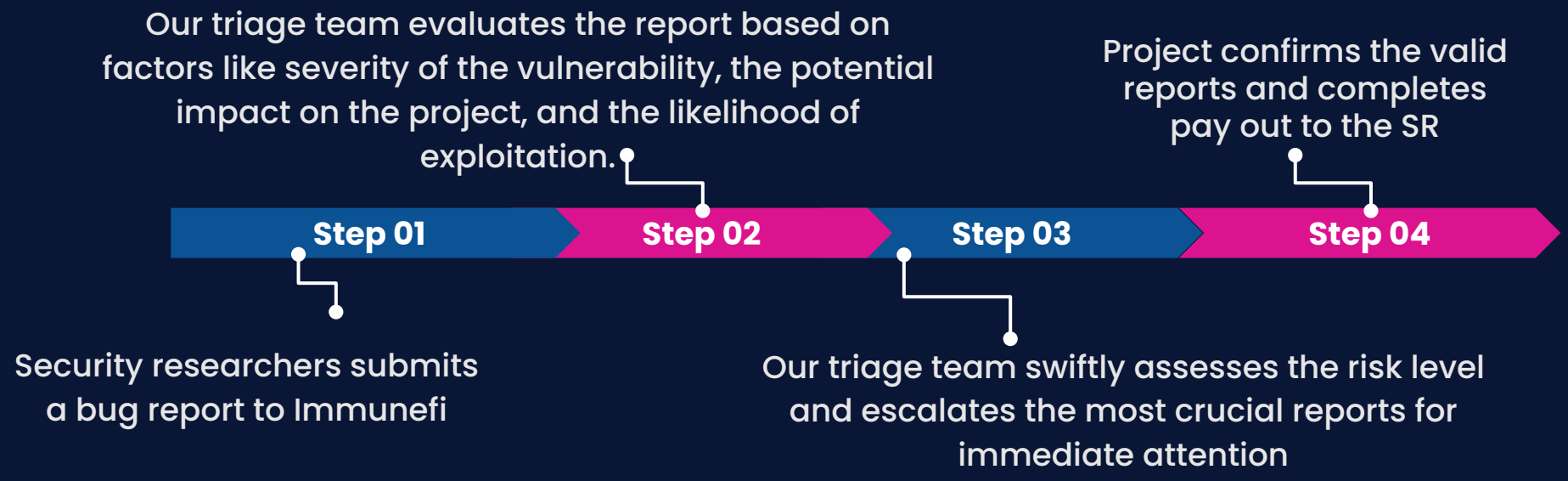
TRUSTED BY

polygon · GMX · MAKER · OPTIMISM · AAVE

ApeCoin · Chainlink · GNOSIS · zkSync lite

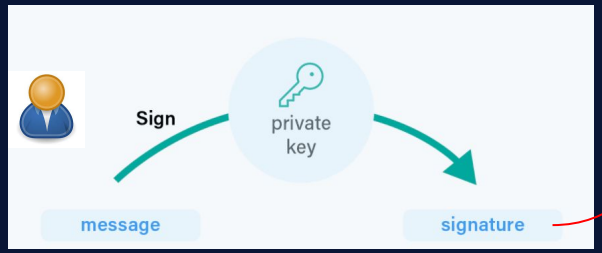

Top Paid $ Bug Reports on Immunefi Platform
Project
Wormhole
Redacted Cartel
Port Finance
Polygon
Optimism
Notional
Moonbeam, Astar, Acala
Moonbeam
Fei Protocol
Enzyme Finance
Belt Finance
Balancer
Aztec
Aurora
ArmorFi
$10,000,000
$630,000
$800,000
$4,200,000
$2,000,042
$1,000,000
$1,000,000
$1,050,000
$1,400,000
$450,000
$1,000,000
$1,000,000
$560,000
$7,000,000
$1,500,000
$0.0M
$2.0M
$4.0M
$6.0M
$8.0M
$10.0M
Amount (in USD Millions)

How bug hunting on Immunifi Works?



Review of the Top Paid Bug reports on Immunefi

Polygon – $2M Bounty



Ether on Ethereum = Matic on Polygon

ecrecover = A built-in function which returns address of signer from hashed data and signature

```
function transferWithSig(
        bytes calldata sig,
        uint256 amount,
        bytes32 data,
        uint256 expiration,
        address to
    ) external returns (address from) {
        require(amount > 0);
        require(
            expiration == 0 || block.number <= expiration,
            "Signature is expired"
        );

        bytes32 dataHash = hashEIP712MessageWithAddress(
            hashTokenTransferOrder(msg.sender, amount, data, expiration),
            address(this)
        );

        require(disabledHashes[dataHash] == false, "Sig deactivated");
        disabledHashes[dataHash] = true;

        from = ecrecovery(dataHash, sig);
        _transferFrom(from, address(uint160(to)), amount);
    }
```

Signature passed by operator

Polygon – $2M Bounty

```
function ecrecovery(bytes32 hash, bytes memory sig)
        public
        pure
        returns (address result)
    {
        bytes32 r;
        bytes32 s;
        uint8 v;
        if (sig.length != 65) {
            return address(0x0);
        }
```

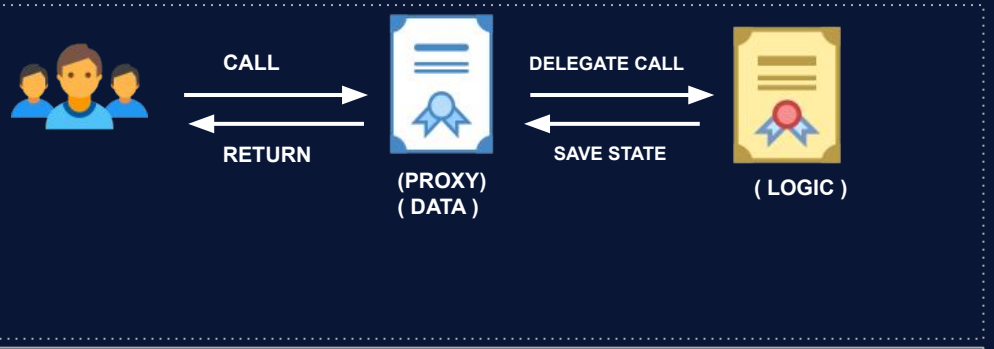
If bytes in signatures are less than 65, it will return address(0) as the signer

```
function _transfer(address sender, address recipient, uint256 amount)
        internal
    {
        require(recipient != address(this), "can't send to MRC20");
        address(uint160(recipient)).transfer(amount);
        emit Transfer(sender, recipient, amount);
    }
```

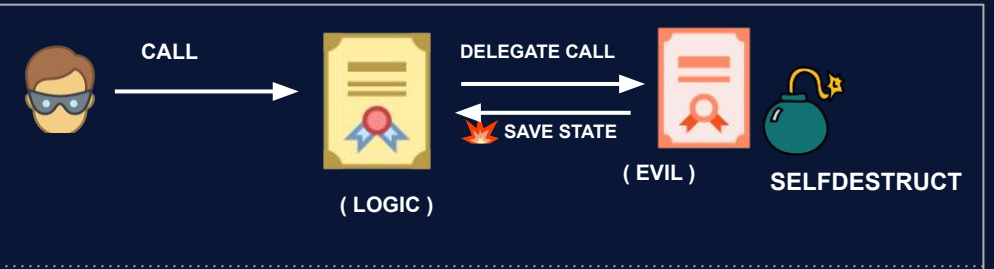
It doesn't check that the sender has enough balance to transfer.

It allows an exploit where anybody can craft such a signature with less than 65 bytes and mint an arbitrary number of tokens

Wormhole - $10M Bounty

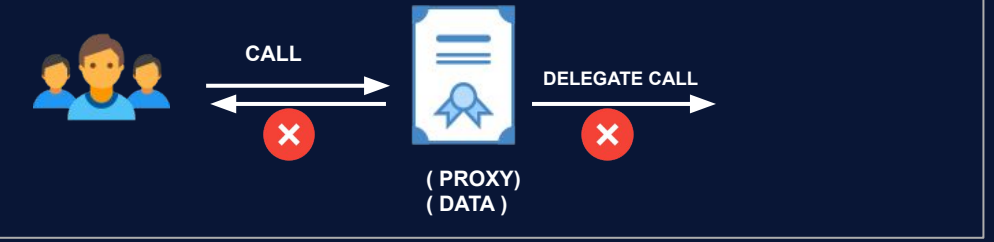


Normal Workflow.



1. Malicious user deploys Evil contract containing SELFDESTRUCT opcode.

2. Delegate(call) to Evil contract.

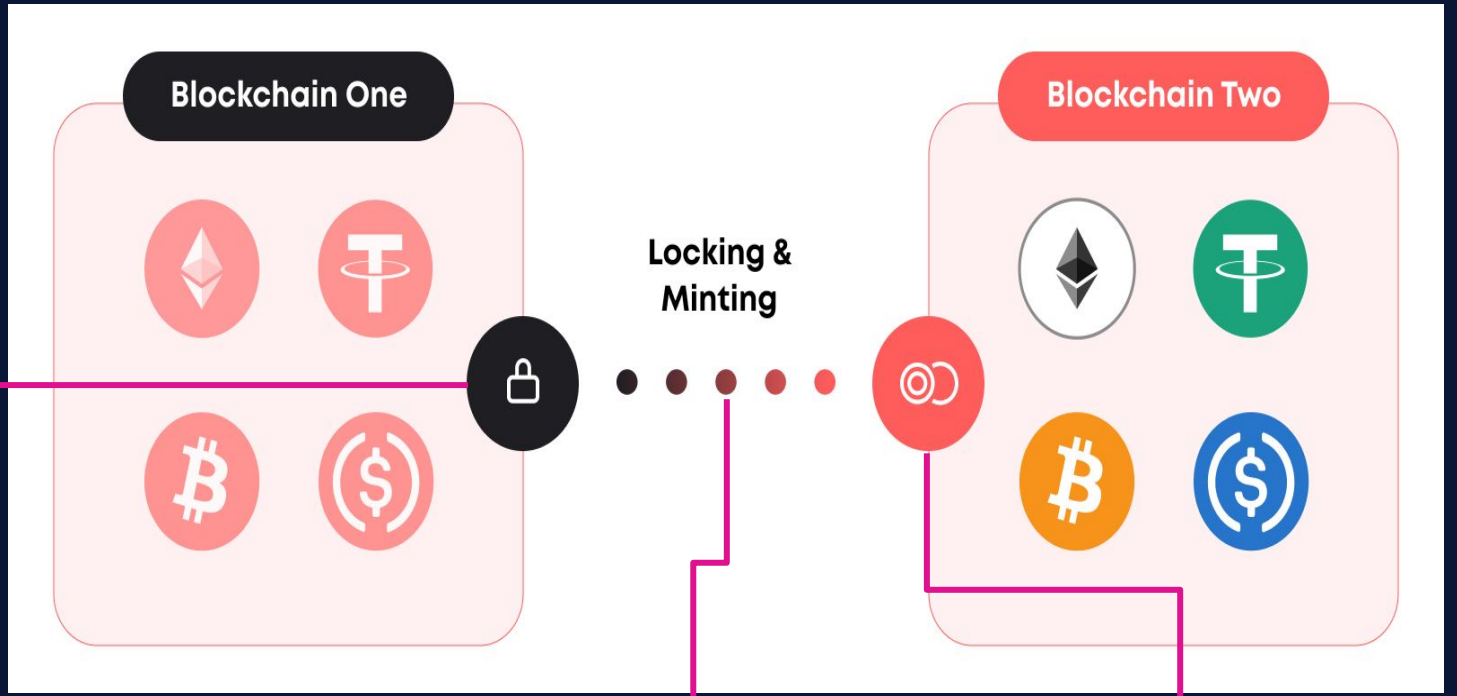


Its storage and code are erased from the blockchain **making funds locked**.

Proxy contract is bricked.

Aurora Finance : $6M Bounty

Bridges in blockchains to communicate with each other

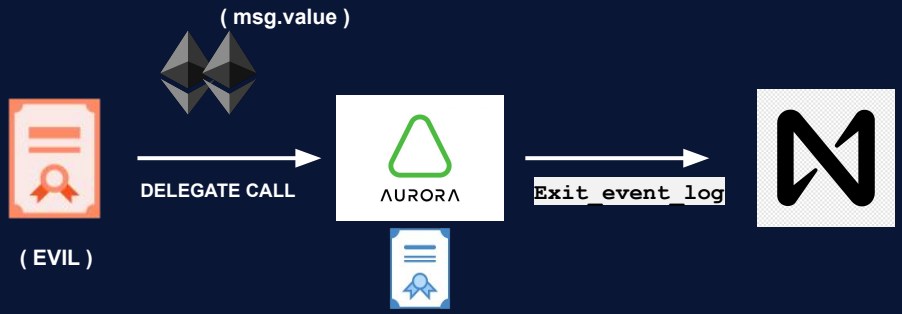


Sends Tokens to special contract to lock

Emits deposit event

Release on others

Aurora Finance : $6M Bounty



1. Attacker creates a malicious contract.

2. It triggers a delegateCall to the bridge contract, simulating an ETH transfer.

3. No actual ETH is sent.

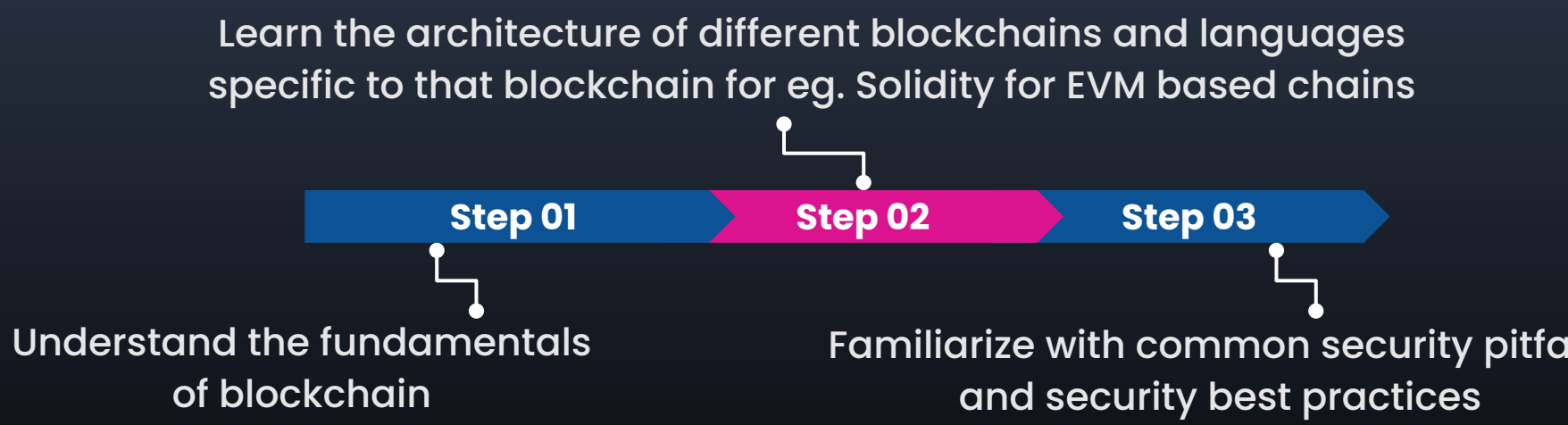
4. Bridge contract runs as if ETH was received (via msg.value).

5. Bridge emits an event, and the attacker receives ETH on the NEAR blockchain.

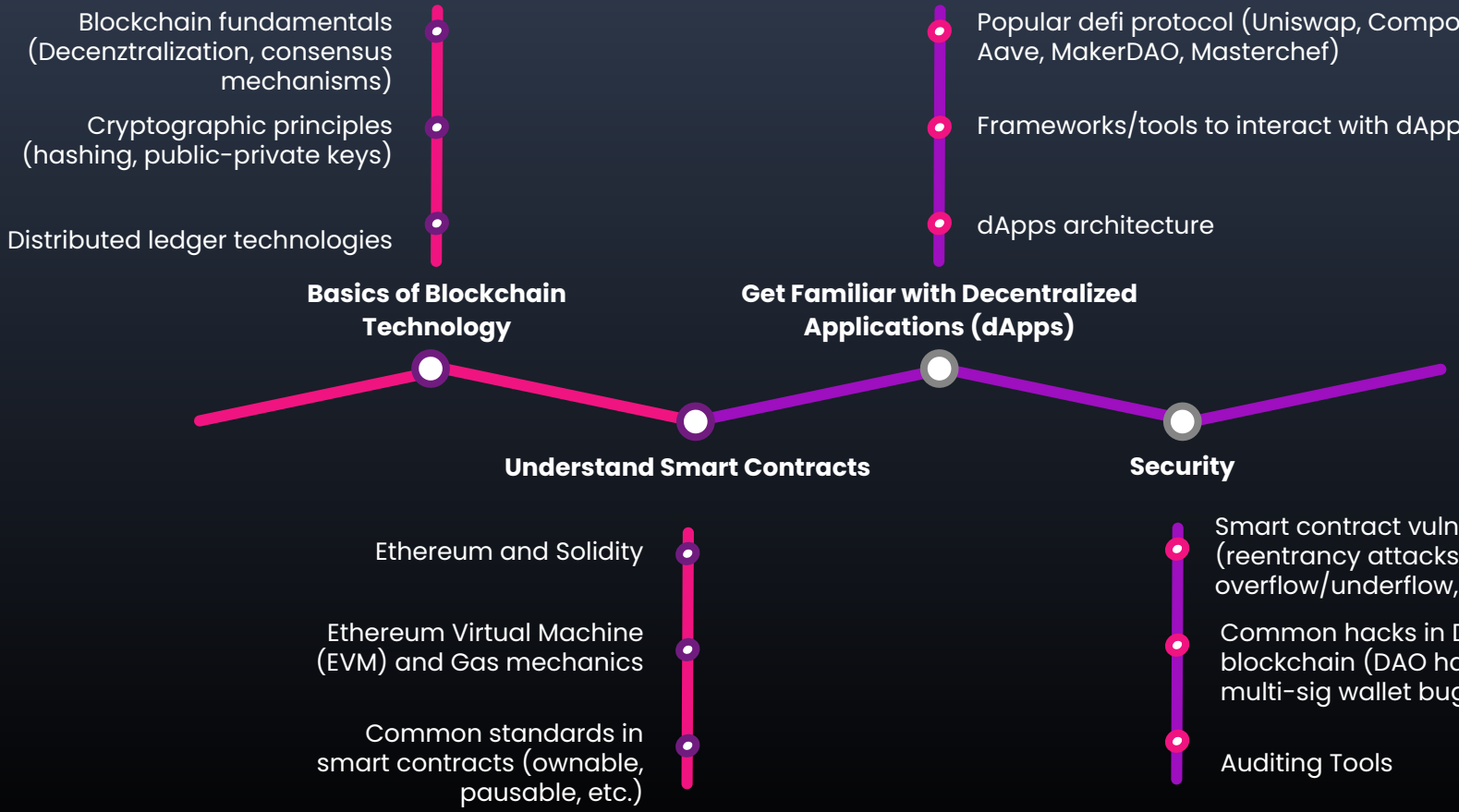
```rust
impl ExitToNear {
    /// Exit to NEAR precompile address
    pub const ADDRESS: Address =
        super::make_address(0xe9217bc7,
0x0b7ed1f598ddd3199e80b093fa71124f);

    pub fn new(current_account_id: AccountId) -> Self {
        Self { current_account_id }
    }
}

impl Precompile for ExitToNear {
        .......
        let exit_event_log = Log {
            address: Self::ADDRESS.raw(),
            topics: exit_event_log.topics,
            data: exit_event_log.data,
        };
        Ok(PrecompileOutput {
            logs: vec![promise_log, exit_event_log],
            ..Default::default()
        }
        .into())
}
```

Who wants to be a Web3 Hacker?

Roadmap to be a Web3 Hacker?



Roadmap to be a Web3 Hacker?



Useful links to get you started

- https://medium.com/immunefi/hacking-the-blockchain-an-ultimate-guide-4f34b33c6e8b
- https://www.youtube.com/@PatrickAlphaC/playlists
- https://solidity-by-example.org
- https://github.com/ethereumbook/ethereumbook
- https://github.com/OffcierCia/DeFi-Developer-Road-Map
- https://www.damnvulnerabledefi.xyz
- https://cmichel.io/how-to-become-a-smart-contract-auditor/
- https://ethernaut.openzeppelin.com/
- https://github.com/immunefi-team/community-challenges
- https://github.com/x676f64/secureum-mind_map
- https://cryptozombies.io/

Scan the QR →
To sign up and we will send you resources to get you started with web3 bug hunting

Win Cool Immunefi Merch

Visit our Booth
We have some exciting contests lined up for you