Blockchain Security

By Harsh Tandel at The Hacker’s Meetup,Surat

$Who am i

- Certified Blockchain Security Examiner
- Web3 Security Researcher
- Smart Contract Auditor
- Integrity Global Top 1000
- P1 Warrior Bugcrowd
- CVE 2022-334

Harsh Tandel

H4r5h_T4nd37

Blockchain Basics

What is Blockchain?

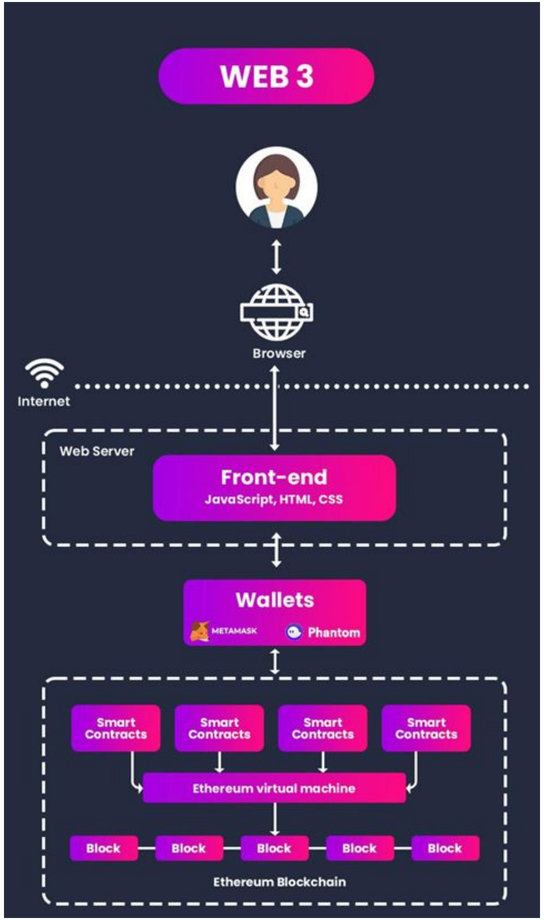
- **Definition**: A decentralized, digital ledger that records transactions across multiple computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the consensus of the network.
- **Key Characteristics**: Transparency, immutability, security, and decentralization.

Key Components

- **Blocks**: Contain a list of transactions, a timestamp, and a reference to the previous block (hash).
- **Hash**: A unique identifier for a block, generated from the block's contents.
- **Consensus Mechanism**: Protocol used to achieve agreement among nodes (e.g., Proof of Work, Proof of Stake).

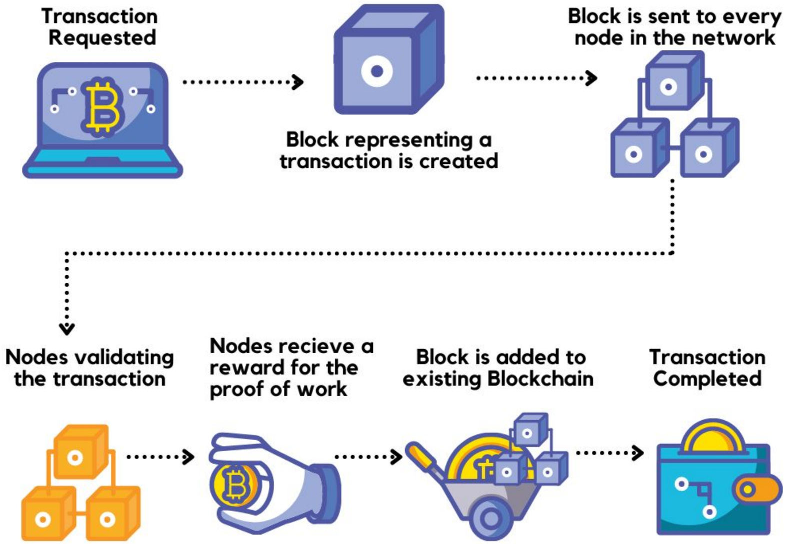
Types of Blockchain

- **Public**: Open to anyone to join and participate (e.g., Bitcoin, Ethereum).
- **Private**: Restricted to specific participants (e.g., Hyperledger, Corda).
- **Consortium**: Controlled by a group of organizations.



How Blockchain Works

1. **Transactions**: Data is recorded as transactions.
2. **Blocks**: Transactions are grouped together in blocks.
3. **Chains**: Blocks are linked together in a chronological chain.
4. **Nodes**: Computers on the network (nodes) validate and relay transactions.



Cryptographic Algorithms in Blockchain

- **SHA-256**: Commonly used in Bitcoin for hashing blocks.
- **ECDSA (Elliptic Curve Digital Signature Algorithm)**: Used for digital signatures in Bitcoin and Ethereum.

Use Cases

- **Cryptocurrencies**: Bitcoin, Ethereum, and other digital currencies.
- **Supply Chain Management**: Tracking goods from origin to consumer.
- **Healthcare**: Securely sharing patient records.
- **Finance**: Cross-border payments, smart contracts.
- **Voting Systems**: Ensuring transparent and tamper-proof elections.

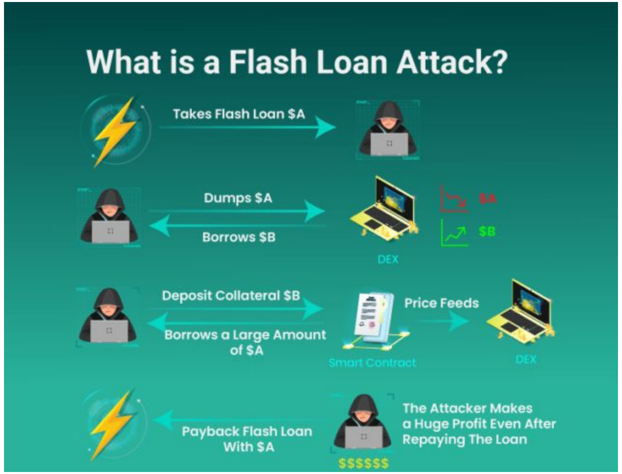
Common Security Vulnerabilities in Blockchain

1. 51% Attack

- **Description**: An attack where a single entity gains control of more than 50% of the network's mining power.
- **Impact**: Enables attackers to double-spend, halt transactions, and reverse transactions.
- **Example**: Bitcoin Gold attack in 2018 resulted in double-spending and over $18 million in losses.
- **Mitigation**: Improved consensus mechanisms, increased decentralization, and higher network difficulty.

2. Flash Loans

- **Description**: Instant, uncollateralized loans that must be repaid within the same transaction.
- **Impact**: Can be used to manipulate prices, exploit vulnerabilities in DeFi protocols.
- **Example**: bZx protocol attack in 2020, where flash loans were used to manipulate oracle prices, leading to significant losses.
- **Mitigation**: Enhanced contract security, improved price oracles, and monitoring for suspicious activity.



3. Double Spending

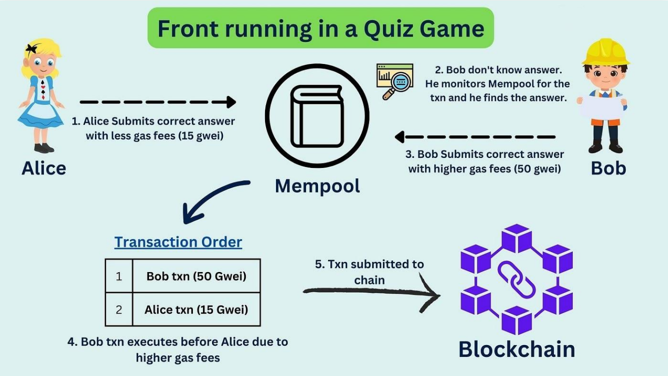
- **Description**: The act of spending the same cryptocurrency more than once.
- **Impact**: Undermines the trust and integrity of the blockchain.
- **Example**: Common in 51% attacks, where transaction history can be rewritten.
- **Mitigation**: Robust consensus algorithms and sufficient network hash rate to prevent 51% control.

4. Signature Replay Attack

- **Description**: Reusing the same transaction data on different blockchains or networks.
- **Impact**: Unauthorized transactions or double spending across different chains.
- **Example**: Ethereum and Ethereum Classic split led to potential replay attacks.
- **Mitigation**: Implement replay protection mechanisms and use unique transaction identifiers.

5. Front Running Attack

- **Description**: Exploiting knowledge of pending transactions to insert one's own transactions first by paying higher fees.
- **Impact**: Unfair advantage and financial losses for original transaction initiators.
- **Example**: Common in decentralized exchanges where order information is publicly visible.
- **Mitigation**: Implement transaction ordering rules and privacy-preserving transaction techniques.



6. Access Control Issue

- **Description**: Improper enforcement of permissions and roles within smart contracts.
- **Impact**: Unauthorized access to critical functions and potential exploitation.
- **Example**: Insecure function exposure in smart contracts allowing unauthorized transfers.
- **Mitigation**: Rigorous access control mechanisms and thorough contract audits.

7. Oracle Manipulation Attack

- **Description**: Exploiting the data provided by external oracles to manipulate smart contract outcomes.
- **Impact**: Incorrect execution of smart contracts based on false data.
- **Example**: bZx protocol exploit where attackers manipulated oracle prices to their advantage.
- **Mitigation**: Use multiple oracles for redundancy and implement decentralized oracle solutions.

8.**Sybil Attack**

- **Description**: An attacker creates multiple fake identities to gain control over the network.
- **Impact**: Disproportionate influence in consensus mechanisms and potential network disruption.
- **Examples**: An attacker creating numerous nodes to overwhelm a decentralized network’s voting system.
- **Mitigation**:Implement robust identity verification and reputation systems to limit the creation of fake identities.

Security Measures in Blockchain

1. Cryptographic Hashing

- **Description**: Cryptographic hashing is fundamental to blockchain security. Hash functions like SHA-256 convert input data into a fixed-size string of characters, which appears random.
- **Purpose**: Ensures data integrity by making it computationally infeasible to alter data without changing the hash. Even a small change in the input produces a significantly different hash.

2. Digital Signatures

- **Description**: Digital signatures use asymmetric cryptography, typically involving a pair of keys (private and public keys).
- **Purpose**: They verify the authenticity and integrity of a message or transaction. Only the holder of the private key can create a signature that others can verify using the corresponding public key.

3. Consensus Mechanisms

- **Proof of Work (PoW)**: Requires miners to solve complex mathematical puzzles to add a block. Ensures network security through computational effort.
- **Proof of Stake (PoS)**: Validators are chosen based on the number of coins they hold and are willing to "stake" as collateral.
- **Delegated Proof of Stake (DPoS)**: Stakeholders elect delegates to validate transactions and secure the network.
- **Practical Byzantine Fault Tolerance (PBFT)**: Ensures consensus despite the presence of malicious nodes. Used in permissioned blockchains.

4. Data Encryption

- **Description**: Encrypting data stored on the blockchain to protect sensitive information.
- **Purpose**: Enhances confidentiality and ensures that even if data is accessed, it cannot be read without the decryption key.

5. Smart Contract Security

- **Description**: Smart contracts are self-executing contracts with the terms of the agreement directly written into code.
- **Purpose**: Ensures automated and trustworthy execution of transactions. However, they must be carefully audited and tested to avoid vulnerabilities like the DAO attack.

6.Multi-Signature (Multi-Sig) Wallets

- **Description**: Multi-signature wallets require multiple private keys to authorize a transaction.
- **Purpose**: Enhances security by requiring multiple approvals for transactions, reducing the risk of single-point failures or fraud.

7. Zero-Knowledge Proofs

- **Description**: Cryptographic techniques that allow one party to prove to another that they know a value without revealing the value itself.
- **Purpose**: Enhances privacy and security by allowing transactions to be validated without revealing details to the network.

8. Regular Audits and Penetration Testing

- **Description**: Regular security audits and penetration testing of the blockchain network and smart contracts.
- **Purpose**: Identifies and mitigates vulnerabilities proactively, ensuring the robustness of the blockchain infrastructure.

Crypto Exchange and Crypto Wallet Security

Cryptocurrency Custody

- **Definition**: Custody involves holding and managing cryptocurrency assets on behalf of clients.
- **Types**:
    - **Self-Custody**: Individuals manage their own private keys and wallets.
    - **Third-Party Custody**: Custodial services (exchanges, financial institutions) hold and manage assets.
- **Importance**: Ensures the safety and security of assets, especially for institutional investors.

Custody Solutions

1. **Cold Storage**:
    - **Description**: Storing cryptocurrencies offline to protect them from online hacks.
    - **Examples**: Hardware wallets, paper wallets.
    - **Security**: Highly secure against online threats; vulnerable to physical theft and loss.
2. **Hot Storage**:
    - **Description**: Storing cryptocurrencies online for quick access and transactions.
    - **Examples**: Online wallets, exchange wallets.
    - **Security**: Convenient but more susceptible to online attacks.

Wallet Security

- **Definition**: Measures and practices to protect cryptocurrency wallets from unauthorized access and theft.
- **Types of Wallets**:
    - **Hardware Wallets**: Physical devices that store private keys offline.
    - **Software Wallets**: Applications or software programs for managing cryptocurrencies.
    - **Paper Wallets**: Physical printouts or handwritten notes of private and public keys.

Crypto Exchange and Crypto Wallet Pentest Methodology

1. KYC Verification testing is a must for most crypto-exchanges and ICOs.
2. Input/Output Testing Tools
3. Testing of the purchase and sale of cryptocurrency (concerns only exchanges)
4. Testing the registration process
5. Testing the Authentication Process
6. Testing of frameworks and technologies used in the development of the exchange
7. OWASP Testing
8. API testing


THANK
YOU