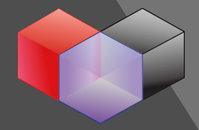
WEB3 Basic & Exploit Tech

블록체인의 기본부터 공격 테크닉까지

February 16th, 2025



POC SECURITY

Agenda

- WEB3란 무엇인가?
- Bitcoin & Ethereum Overview
- Smart Contract Exploit Tech
- WEB3 해커의 진로
- 마무리
- Q & A

Whoami

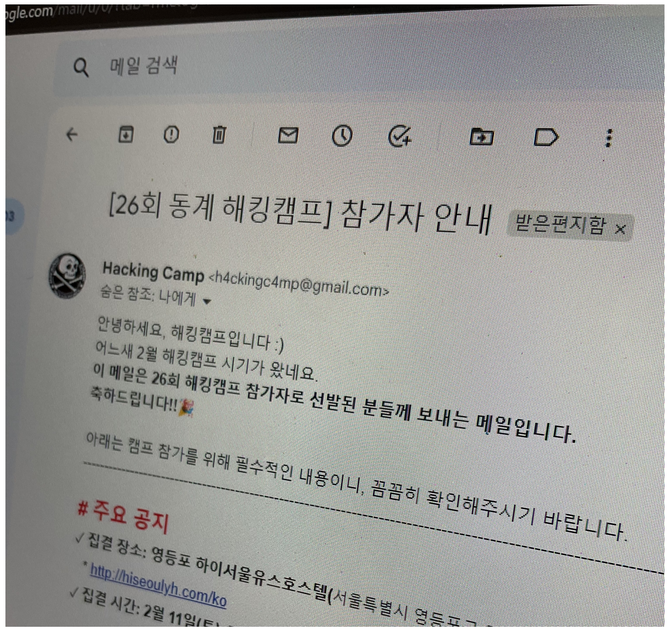
p6rkdoye0n
박도연

Security Researcher

- Zellic WEB3 Security Researcher
  - EVM, ZK, Solana 등의 보안 감사
  - Zero-day Research
- KOREA University Cyber Security
  - 평범한 대학생 (2학년 - 24학번)
  - 수상실적으로 입시 도전
- TeamH4C / Whitehat team

Hacking Camp?

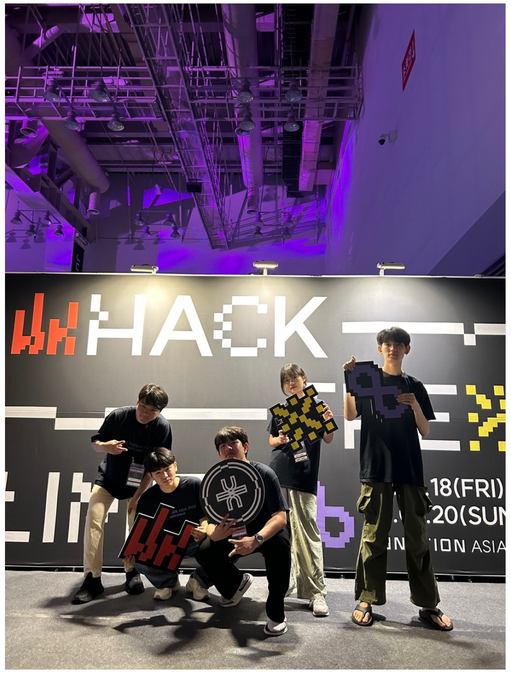
발표자에게 해킹캠프란



POC SECURITY

Hacking Camp?

발표자에게 해킹캠프란



POC SECURITY

Hacking Camp?

해킹캠프에서 참가자들이 얻어갔으면 하는 것들

POC SECURITY

WEB3란 무엇인가?

해킹의 여러 분야

- 시스템 해킹
- 웹 해킹
- 리버싱
- 암호학
- 포렌식
- AI 해킹
- 클라우드 해킹
- 임베디드 해킹

WEB3란 무엇인가?

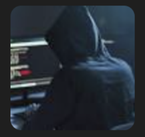
WEB3(암호화폐) 해킹의 유행

지디넷코리아

FBI "北해커, 일본서 4천500억 규모 비트코인 탈취"

일본에서 발생한 대규모 비트코인 탈취 사건이 북한 해커조직의 소행으로 밝혀졌다. 관계 기업의 데이터를 탈취한 후 이를 악용하는 치밀함이 엿보여...

1 month ago



POC SECURITY

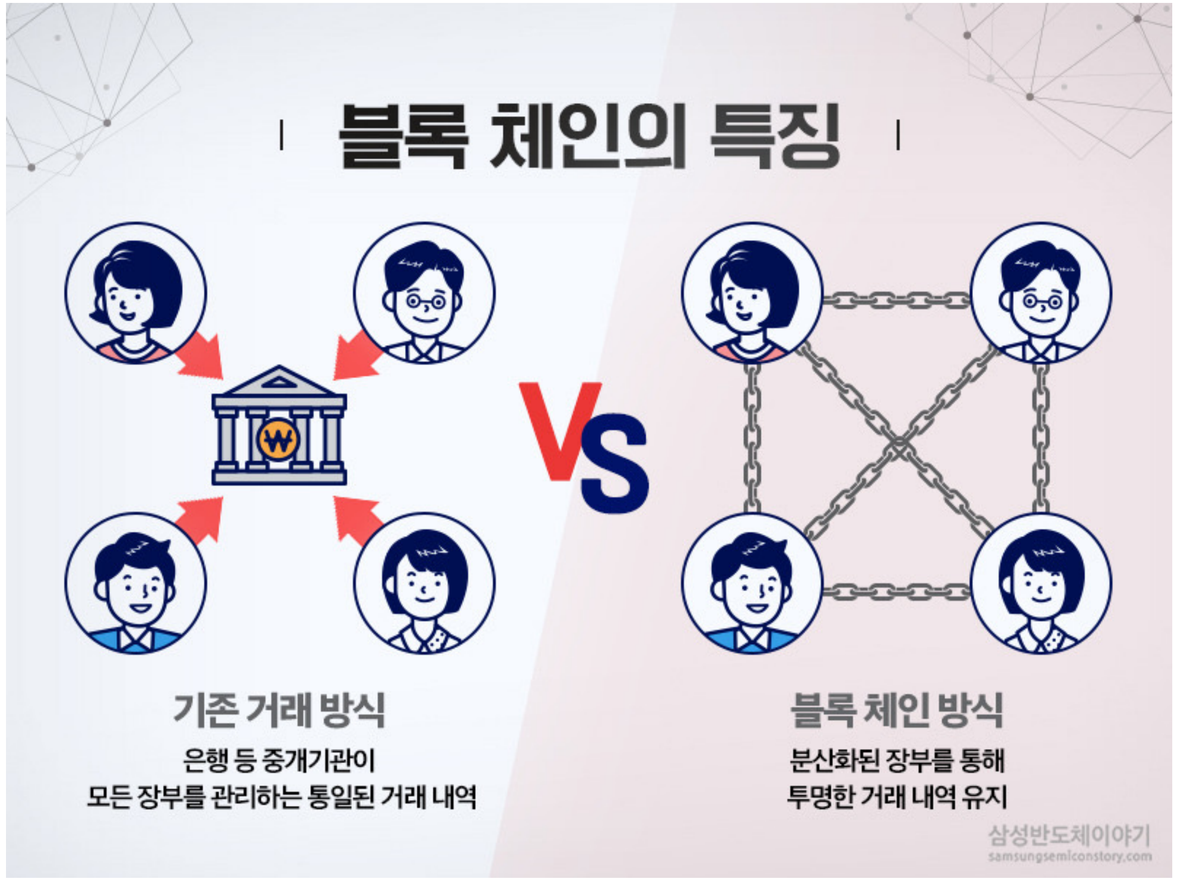
WEB3란 무엇인가?

용어정리

- **블록체인**: 블록에 데이터를 담아 체인 형태로 연결한 뒤, 수많은 컴퓨터에 이를 동시에 복제, 저장하는 분산형 데이터 저장기술

- **WEB3**: 블록체인을 웹과 결합한 기술

- **암호화폐**: 블록체인 기술을 기반으로 분산 환경에서 운영되는 디지털 자산



POC SECURITY

WEB3란 무엇인가?

WEB3 해킹 방법

- **Smart Contract 해킹**
  - ERC20, ERC721(NFT) 등
  - DEX, DeFi, Oracle, Swap 등
- **P2P Node 해킹**
  - 블록체인 (DLT) 합의 알고리즘 등
- **Crypto-Phishing**
  - 사람을 속여 악의적인 요청을 보내도록 유도
  - Chrome extension Malware를 이용한 공격이 유행
- **Private-key 탈취**
  - WEB2 취약점으로 인한 개인키 탈취
  - Crypto-Phishing으로 블록체인 지갑의 개인키를 탈취

POC SECURITY

WEB3란 무엇인가?

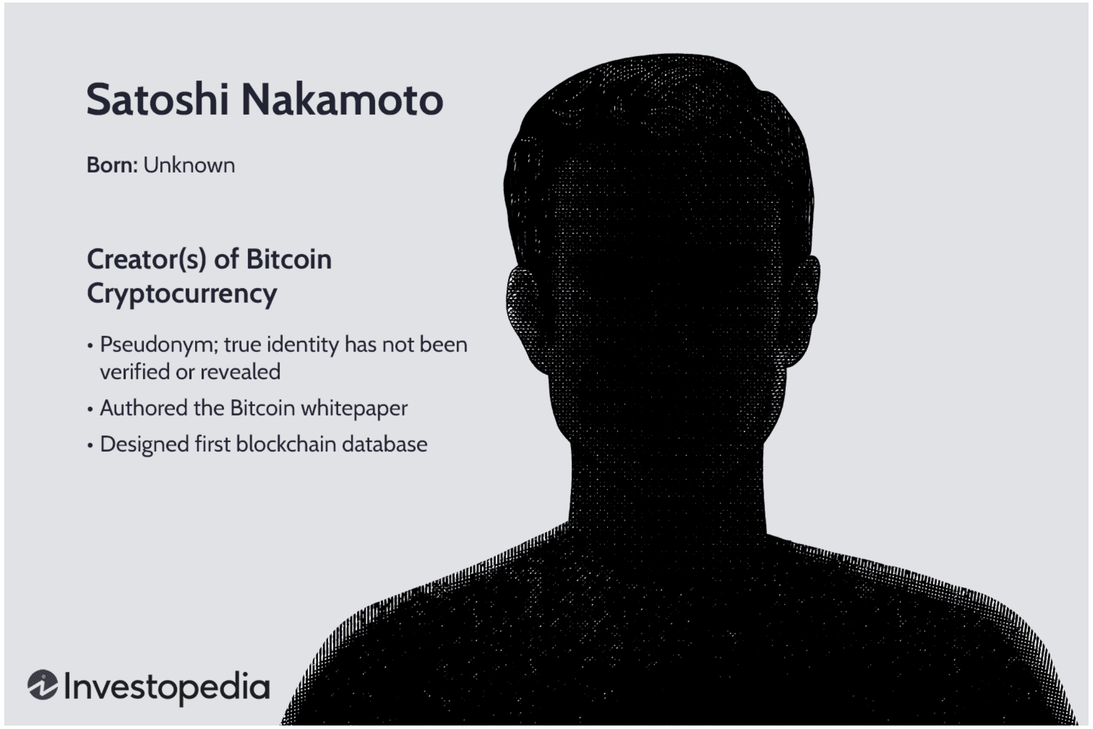
WEB3 해킹 방법

- **Smart Contract 해킹**
  - ERC20, ERC721(NFT) 등
  - DEX, DeFi, Oracle, Swap 등
- **P2P Node 해킹**
  - 블록체인 (DLT) 합의 알고리즘 등
- **Crypto-Phishing**
  - 사람을 속여 악의적인 요청을 보내도록 유도
  - Chrome extension Malware를 이용한 공격이 유행
- **Private-key 탈취**
  - WEB2 취약점으로 인한 개인키 탈취
  - Crypto-Phishing으로 블록체인 지갑의 개인키를 탈취

POC SECURITY

Bitcoin & Ethereum Overview

Bitcoin (Background)



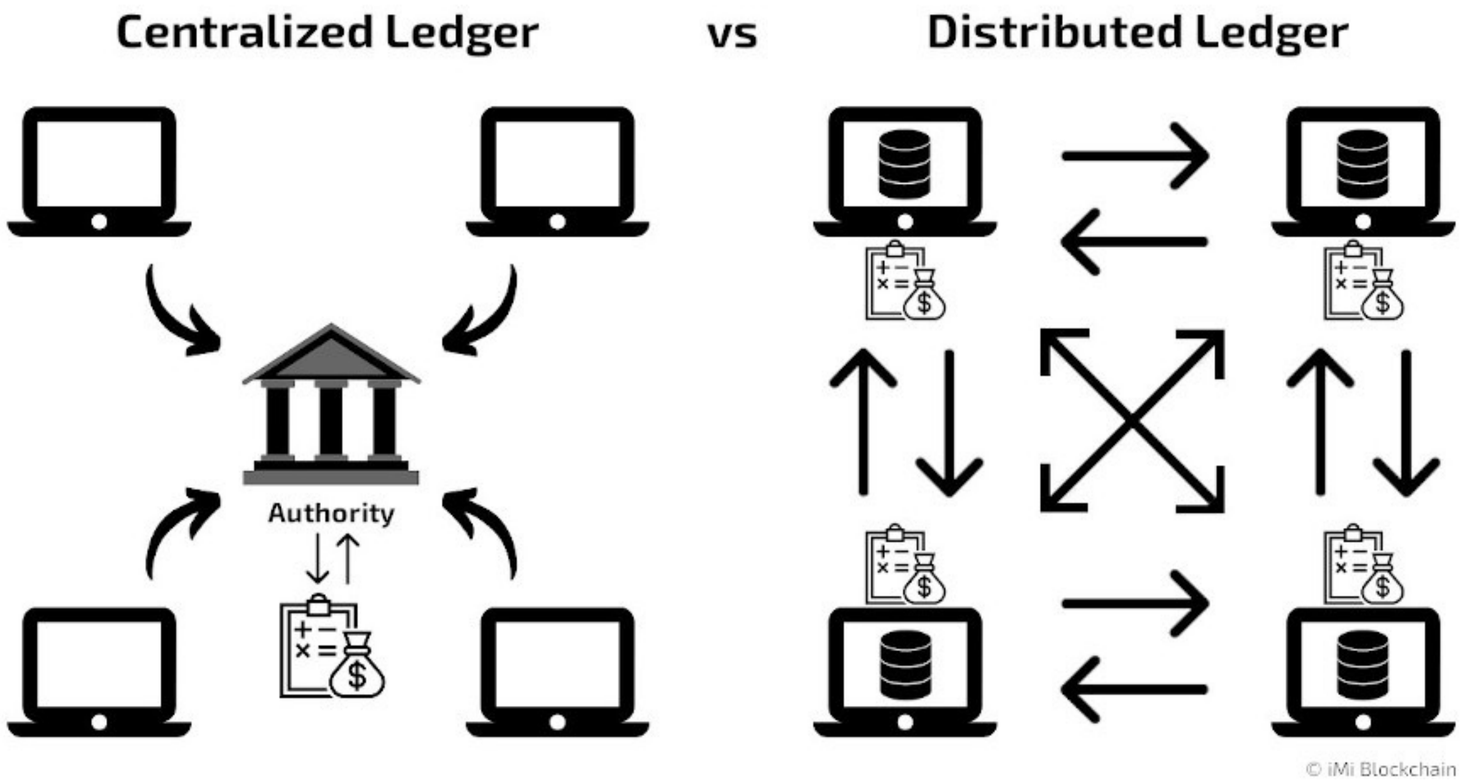
Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network.

Bitcoin & Ethereum Overview

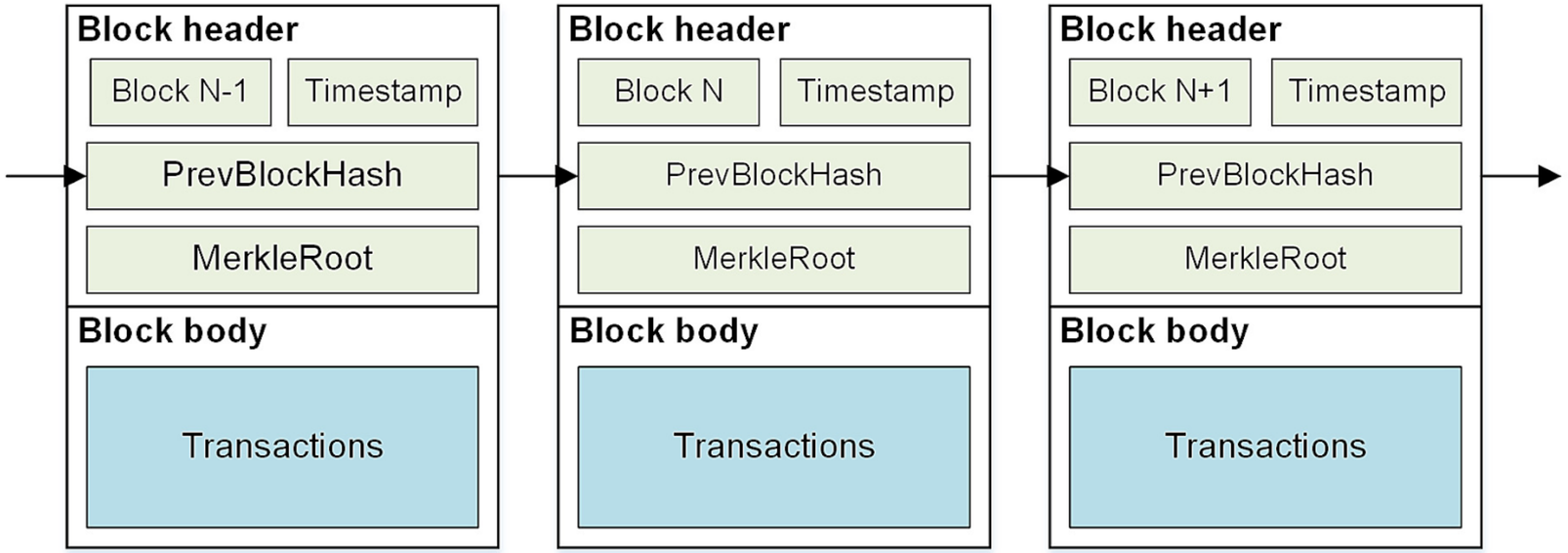
Bitcoin (Distributed Ledger)



기존의 화폐에서 신뢰기관(은행)이 하던 역할을 분산원장 기술로 대체해 탈중앙화를 이룸

Bitcoin & Ethereum Overview

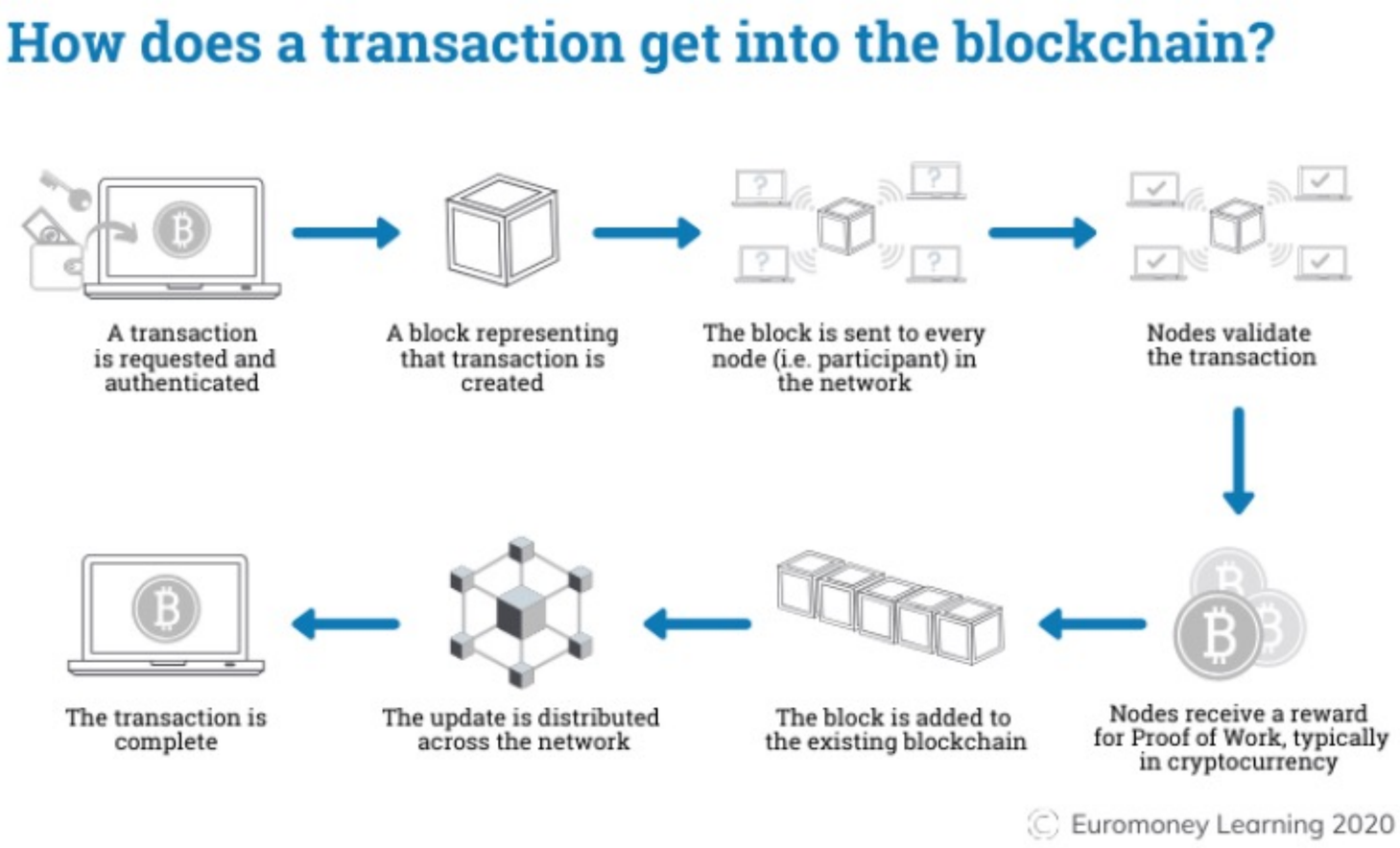
Bitcoin (Blockchain)



블록체인의 블록 구조 (블록들이 하나의 체인으로 연결되어 있음)

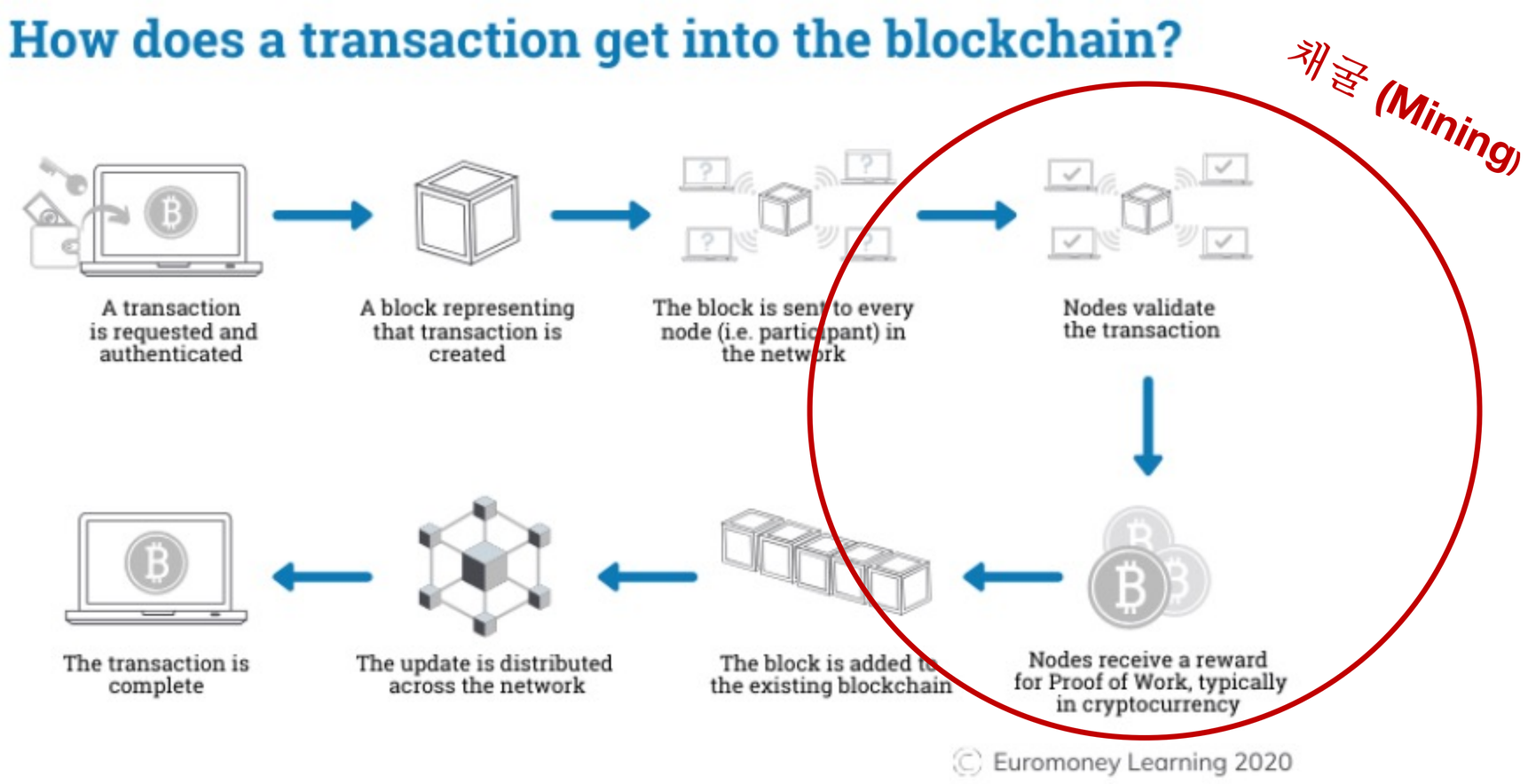
Bitcoin & Ethereum Overview

Bitcoin (Transaction)



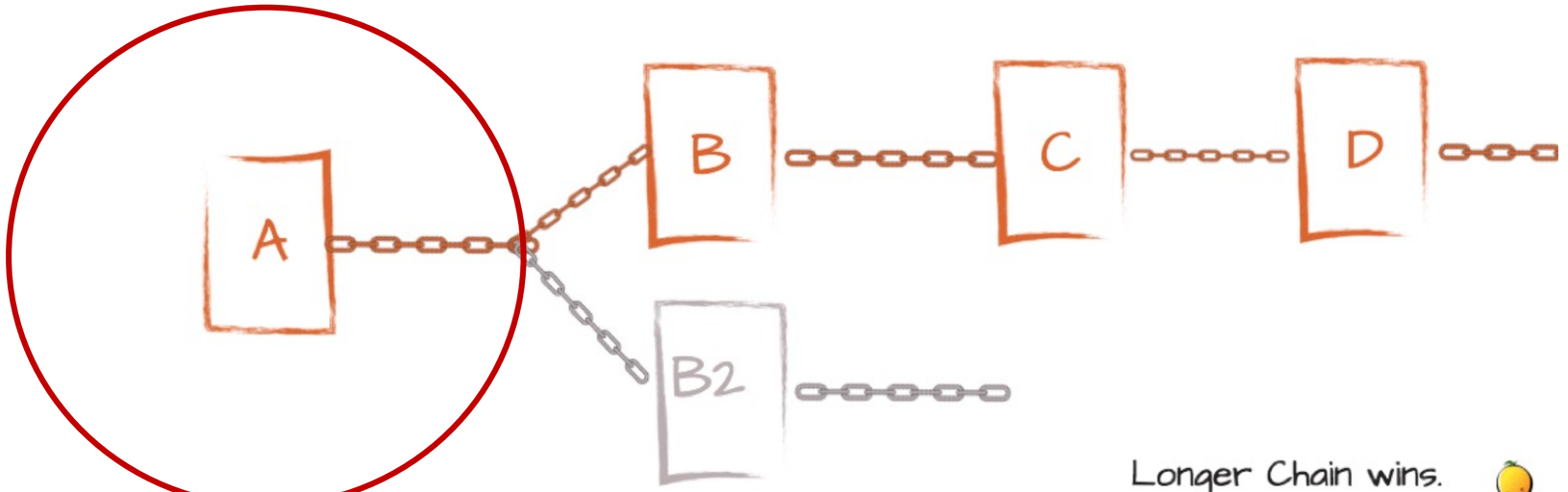
Bitcoin & Ethereum Overview

Bitcoin (Transaction)



Bitcoin & Ethereum Overview

Bitcoin (The Longest Chain Rule)

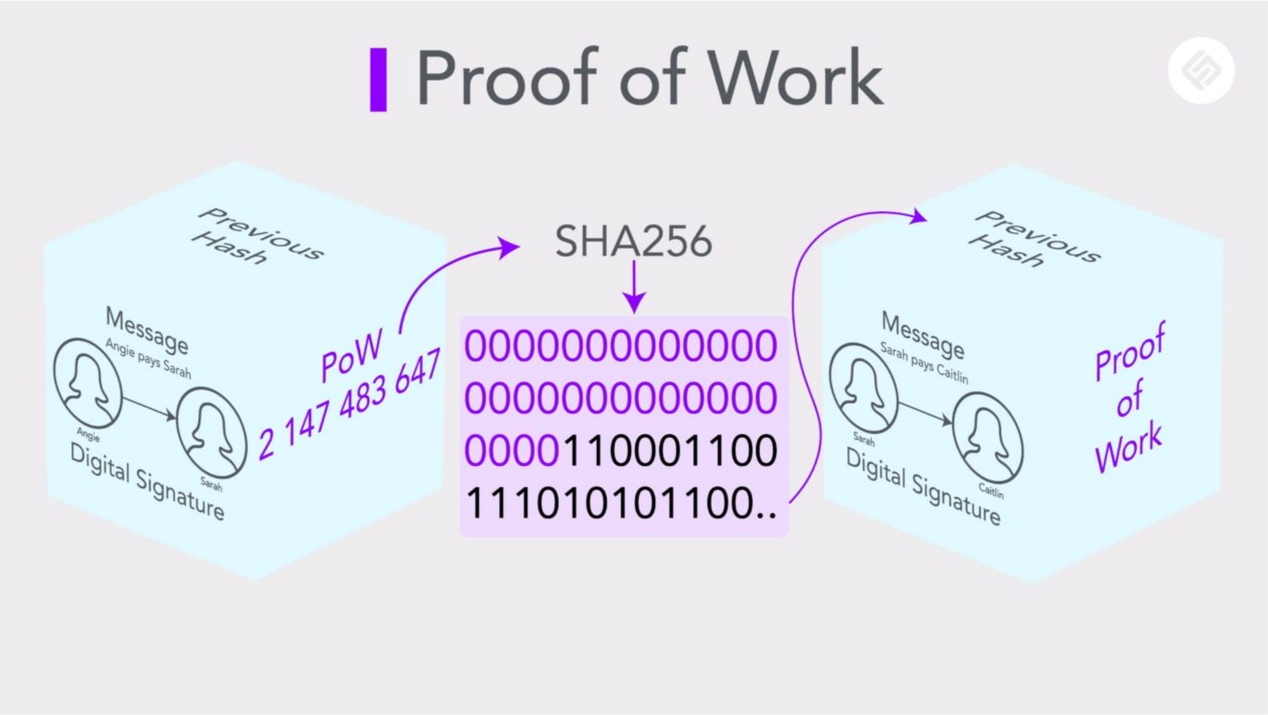


이미 컨펌 된 체인의 끝

유효한 블록은 합의가 절반(50%)이상 모인 체인의 블록

Bitcoin & Ethereum Overview

Bitcoin (Proof Of Work)



The Longest Chain Rule을 이용한 공격인 Sybil Attack 방지를 위해 나온 비트코인의 해결책 및 검증 방법

POC SECURITY

Bitcoin & Ethereum Overview

Ethereum (Background)

- WOW(월드오브워크래프트) 게임을 하던 한 청소년이 자신이 힘들게 얻은 무기가 게임사에 의해 성능이 낮아지는 패치를 당하게 되자 중앙화된 시스템에 문제의식을 갖게됨

- 프로그래머였던 아버지의 추천으로 비트코인을 연구 (고교 시절 비트코인 매거진 운영)

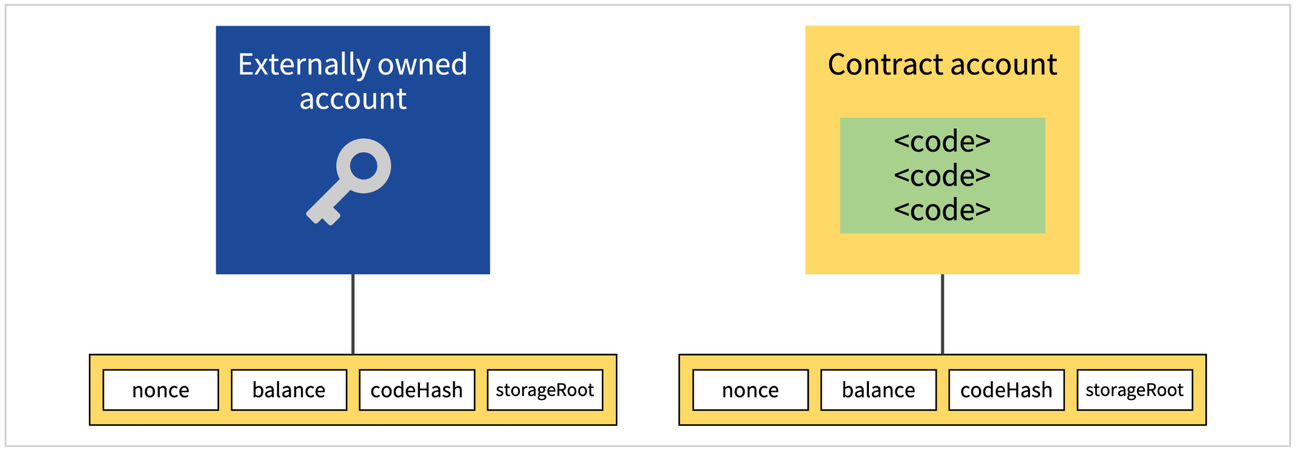
- 비트코인(블록체인)에 거래 기록만을 저장하는 것은 아깝다고 생각

- 비트코인 커뮤니티에 거래 기록과 함께 소프트웨어 코드도 저장하자는 아이디어 제안

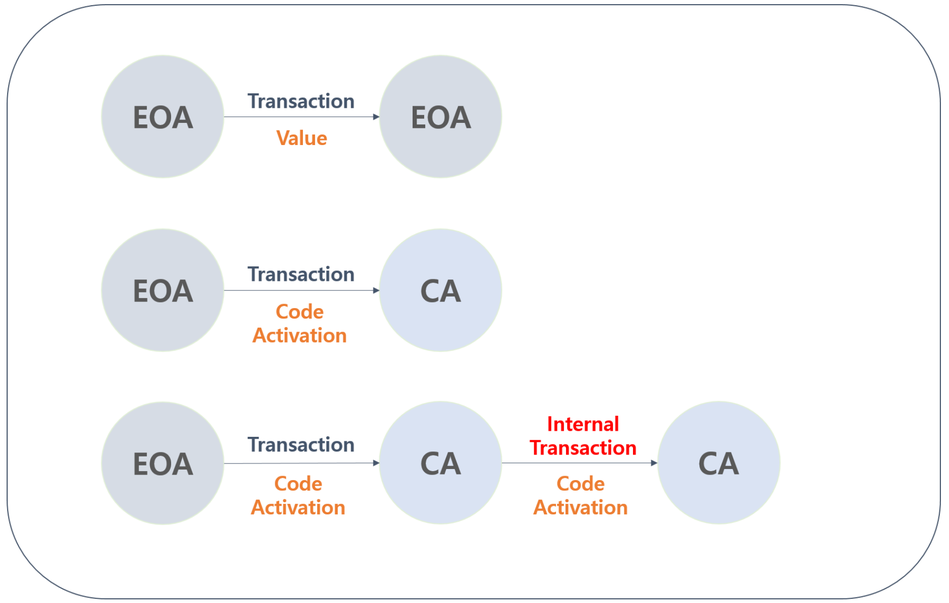
POC SECURITY

Bitcoin & Ethereum Overview

Ethereum (Account)



External Owned Accounts(EOA)
사용자 소유의 계정
→ Private Key에 의해 통제되는 계정 정보

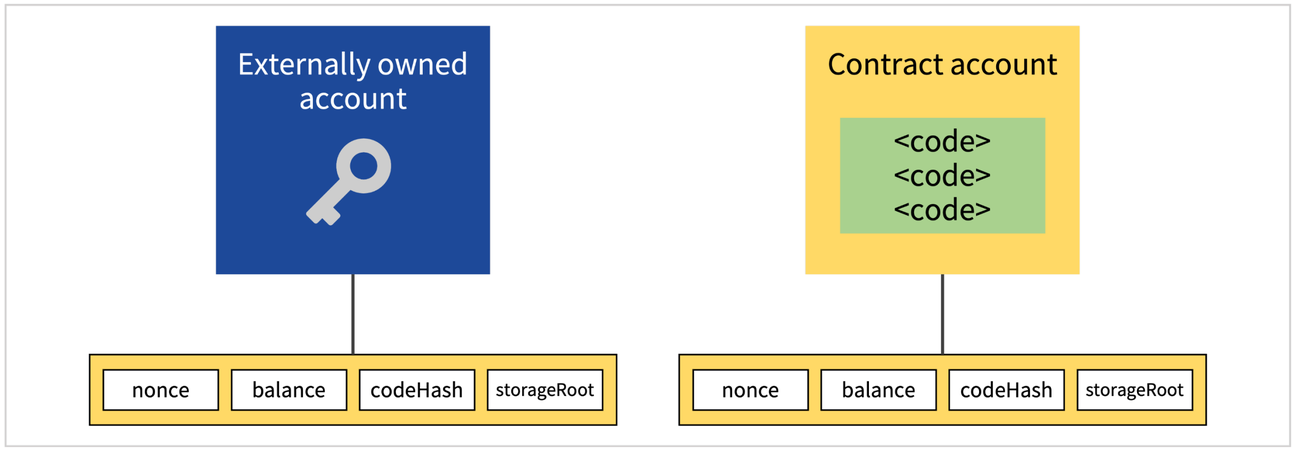


Contract Accounts (CA)
컨트랙트 코드에 의해 통제되는 계정 정보

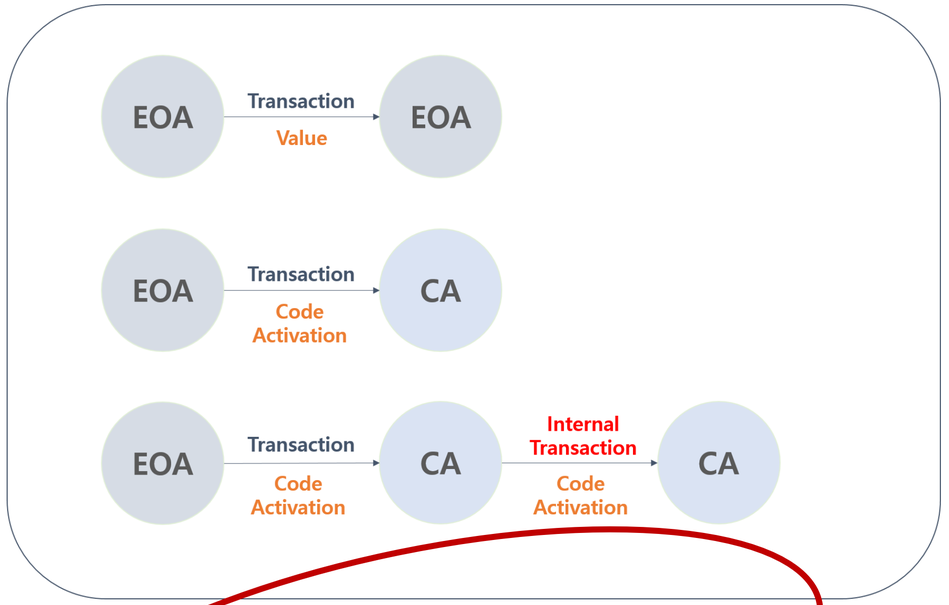
POC SECURITY

Bitcoin & Ethereum Overview

Ethereum (Smart Contract)



External Owned Accounts(EOA)
사용자 소유의 계정
→ Private Key에 의해 통제되는 계정 정보



Contract Accounts (CA)
컨트랙트 코드에 의해 통제되는 계정 정보

= Smart Contract

POC SECURITY

Bitcoin & Ethereum Overview

Ethereum (Smart Contract)

- 암호화폐 (ERC20, ERC721)
- DeFi
- Staking System
- Swap
- DEX
- DAO (Governance)

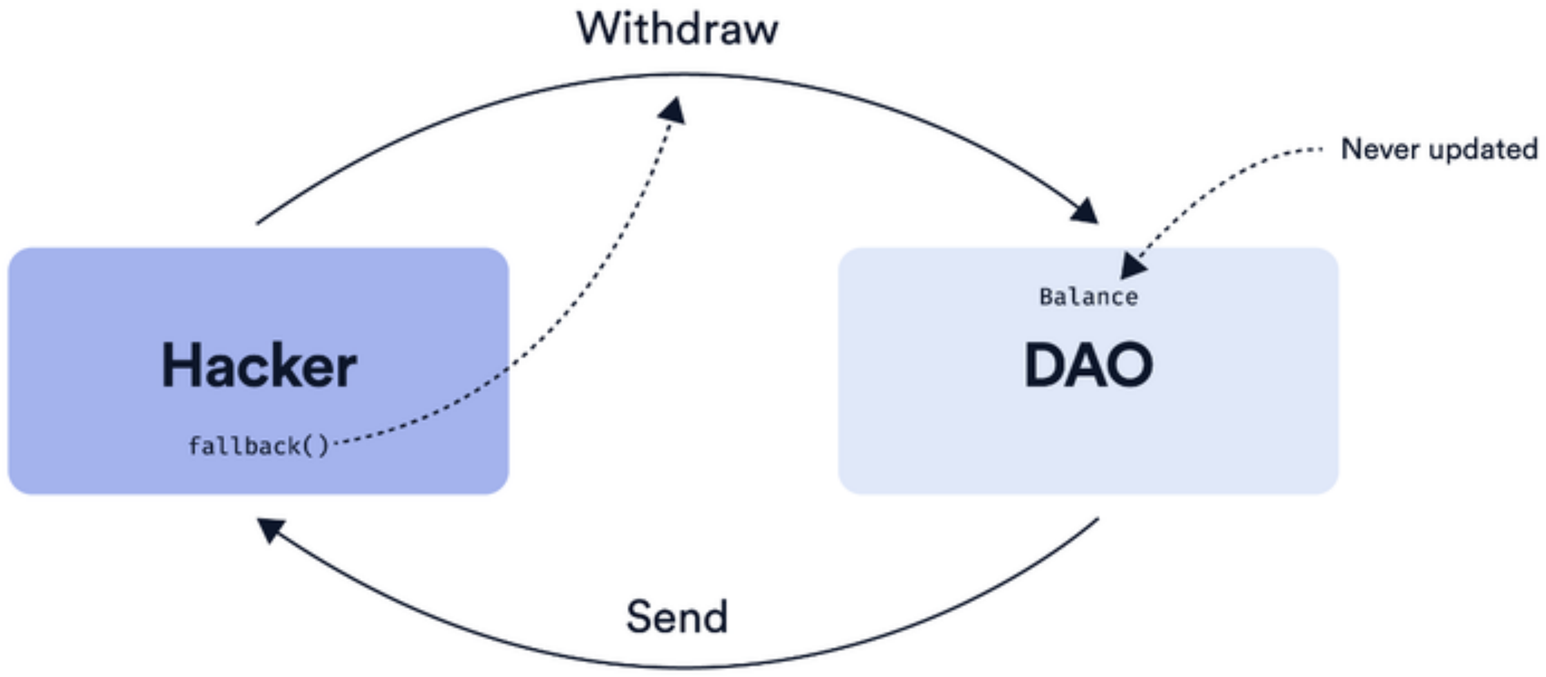
Smart Contract Exploit Tech

다양한 버그 클래스

- Improper Input Validation
- Incorrect Calculation
- Oracle/Price Manipulation
- Weak Access Control
- Replay Attacks/Signature Malleability
- Rounding Error
- Reentrancy
- Frontrunning
- Uninitialized Proxy
- Governance Attacks

Smart Contract Exploit Tech

버그 클래스 (Reentrancy Attack)



POC SECURITY

Smart Contract Exploit Tech

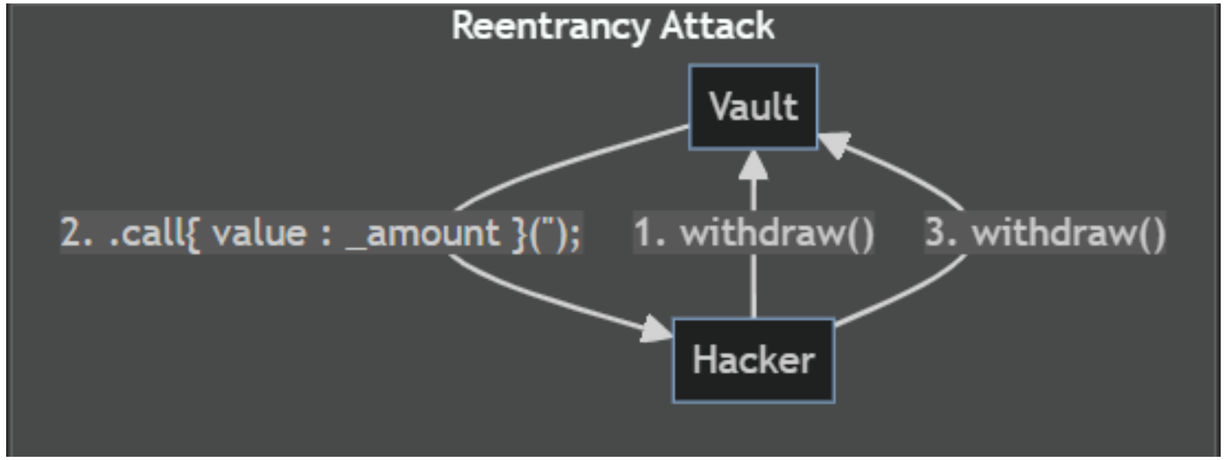
버그 클래스 (Reentrancy Attack)


```
1   //SPDX-License-Identifier :MIT
2   pragma solidity ^0.6.0;
3
4   contract Vault {
5   mapping(address => uint256) public balances;
6
7   function deposit() public payable {
8           balances[msg.sender] += msg.value;
9       }
10
11  function withdraw(uint256 _amount) public {
12          require(balances[msg.sender] >= _amount);
13
14          (bool res, ) = payable(msg.sender).call{ value : _amount }("");
15          require(res);
16          balances[msg.sender] -= _amount;
17      }
18  }
```

Smart Contract Exploit Tech

버그 클래스 (Reentrancy Attack)

```
1   //SPDX-License-Identifier :MIT
2   pragma solidity ^0.6.0;
3
4   contract Vault {
5   mapping(address => uint256) public balances;
6
7   function deposit() public payable {
8           balances[msg.sender] += msg.value;
9       }
10
11  function withdraw(uint256 _amount) public {
12          require(balances[msg.sender] >= _amount);
13
14          (bool res, ) = payable(msg.sender).call{ value : _amount }("");
15          require(res);
16          balances[msg.sender] -= _amount;
17      }
18  }
```



POC SECURITY

Smart Contract Exploit Tech

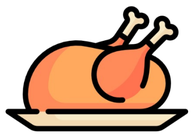
버그 클래스 (Reentrancy Attack)

POC SECURITY

Smart Contract Exploit Tech

버그 클래스 (Reentrancy Attack Mitigation - 퀴즈)

```
1   //SPDX-License-Identifier :MIT
2   pragma solidity ^0.6.0;
3
4   contract Vault {
5   mapping(address => uint256) public balances;
6
7   function deposit() public payable {
8           balances[msg.sender] += msg.value;
9       }
10
11  function withdraw(uint256 _amount) public {
12          require(balances[msg.sender] >= _amount);
13
14          (bool res, ) = payable(msg.sender).call{ value : _amount }("");
15          require(res);
16          balances[msg.sender] -= _amount;
17      }
18  }
```

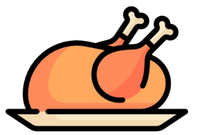


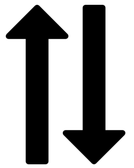
POC SECURITY

Smart Contract Exploit Tech

버그 클래스 (Reentrancy Attack Mitigation - 퀴즈)

```
1   //SPDX-License-Identifier :MIT
2   pragma solidity ^0.6.0;
3
4   contract Vault {
5   mapping(address => uint256) public balances;
6
7   function deposit() public payable {
8           balances[msg.sender] += msg.value;
9       }
10
11  function withdraw(uint256 _amount) public {
12          require(balances[msg.sender] >= _amount);
13
14          (bool res, ) = payable(msg.sender).call{ value : _amount }("");
15          require(res);
16          balances[msg.sender] -= _amount;
17      }
18  }
```



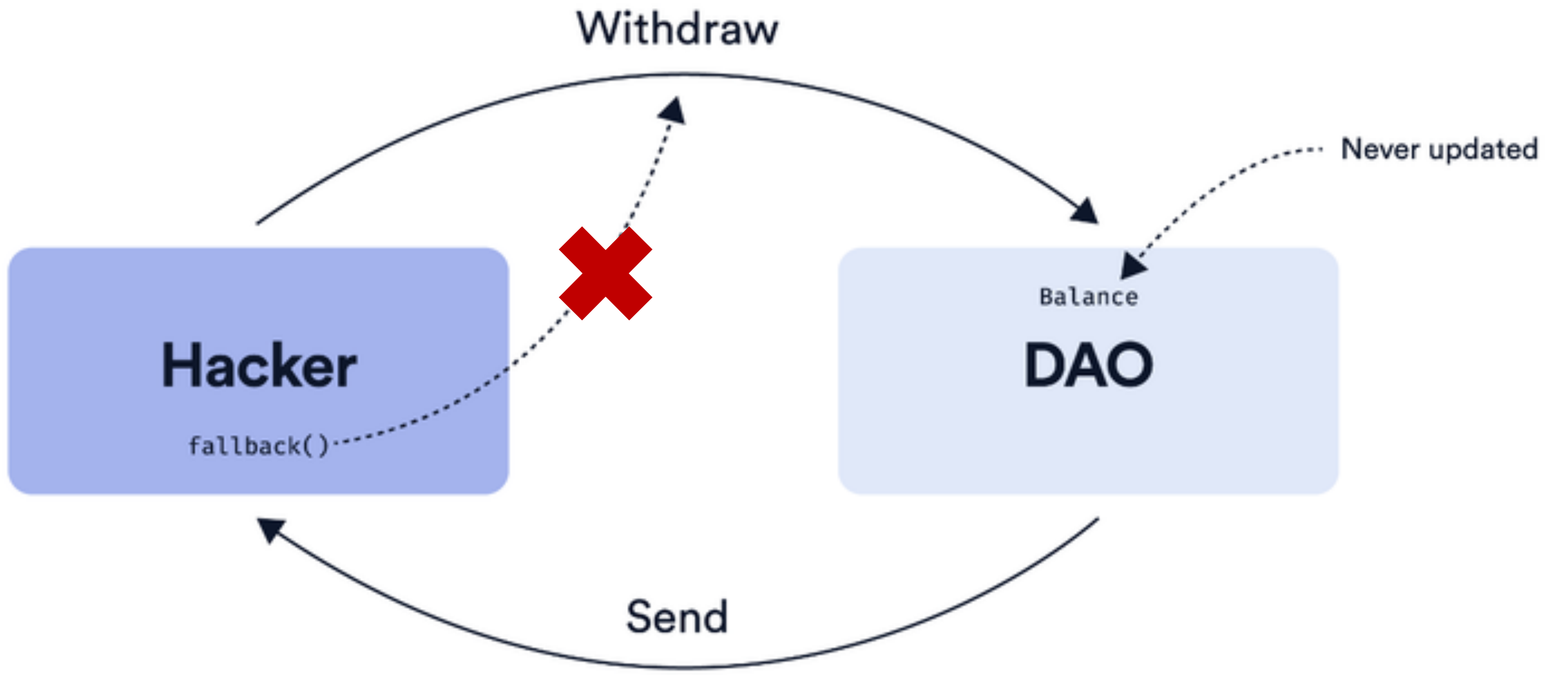


POC SECURITY

Smart Contract Exploit Tech

버그 클래스 (Reentrancy Attack Mitigation)

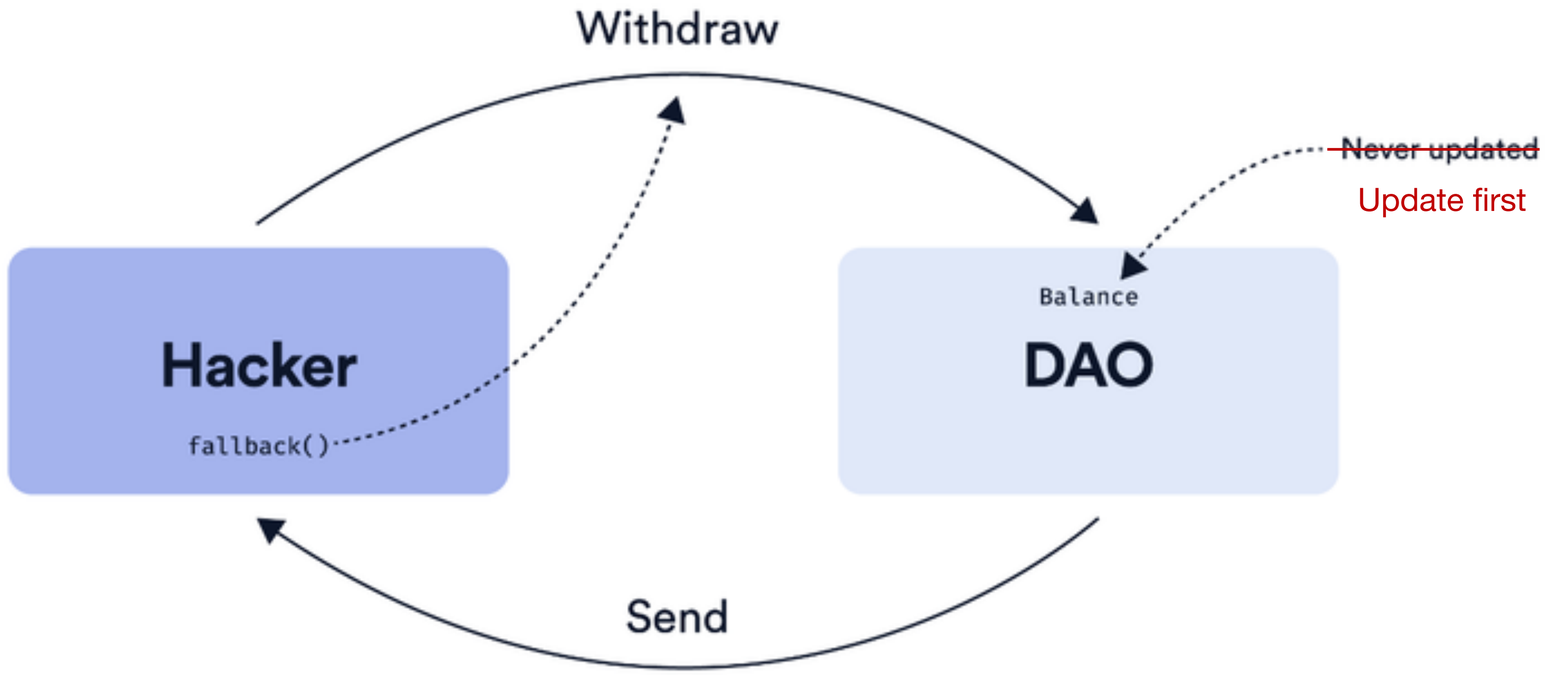
1. Reentrancy Guard 사용



Smart Contract Exploit Tech

버그 클래스 (Reentrancy Attack Mitigation)

2. Checks Effects Interactions Secure Coding



POC SECURITY

Smart Contract Exploit Tech

추천 리서치

WEB3 해커의 진로

- **해외 WEB3 보안 회사들**
  - Zellic, Ottersec, Certik, Spearbit 등등
- **국내 WEB3 보안 회사들**
  - Theori (Chainlight), Sooho, Hexlant 등
- **Bug Bounty Platform**
  - immunefi.com
  - code4rena.com
  - cantina.xyz
  - hackerone.com
  - patchday.io

마무리

- 새로운 개념들이 등장을 많이 하기 때문에
  개념들 생태계에 대한 확실한 이해가 필요하다
- WEB3 보안은 여전히 블루오션이다
- WEB3 보안 연구원의 조건이 굉장히 좋다
- 생각보다 재밌는 개념들이 많이 등장한다
- 발표자에게는 굉장히 고마운 분야이다

POC SECURITY

감사합니다.

QnA

February 16th, 2025

POC SECURITY