Web2를 넘어
Web3 보안을 탐험하는 이유

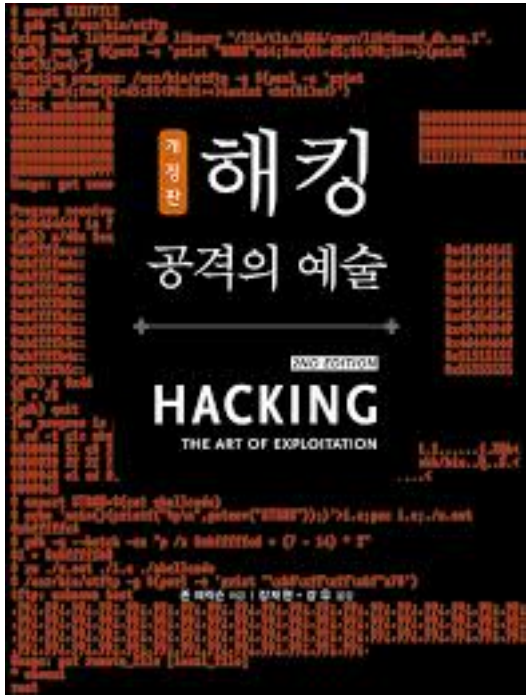
고려대학교 조원민

▪ 발표자 소개

- 고려대학교 스마트보안학과
- CyKor 동아리
- Upside Academy 2기

자세한 내용… wiimdy.kr/about

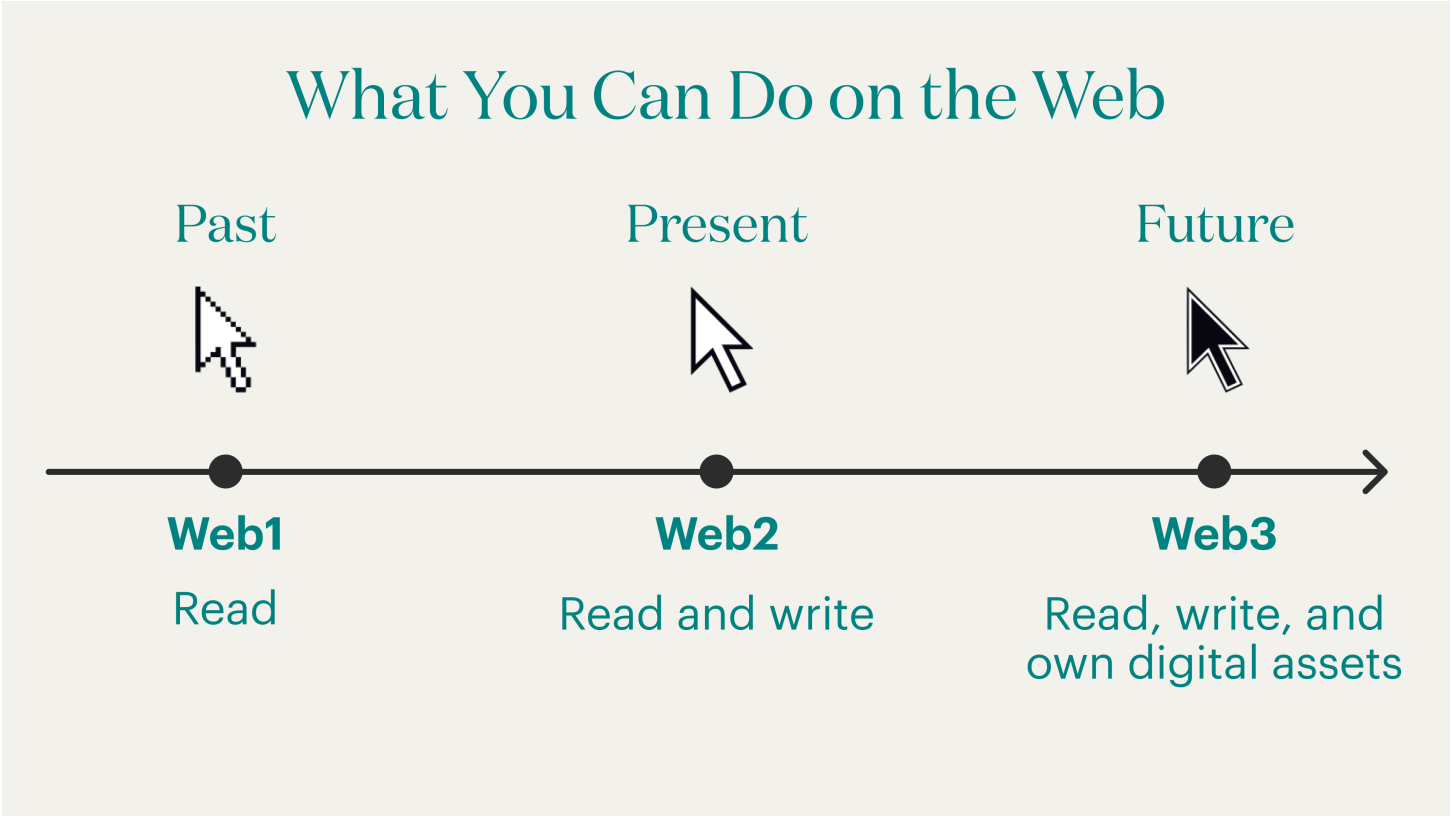
- 우리는 왜 해킹이라는 길을 가게 되었는가?



01

Web3 너가 뭔데?

- **Web3 너가 뭔데?**


What You Can Do on the Web
Past
Present
Future
Web1
Read
Web2
Read and write
Web3
Read, write, and own digital assets

Web1 & Web2

Web1: 읽기만 가능한 인터넷!

Static HTML로 구성되어 있어 사용자는 읽기만 가능

Ex) Yahoo, Google...

Web2: 사용자가 콘텐츠를 직접 생성!

현재 대부분의 홈페이지

모든 데이터와 상호작용은 특정 기업(Google, Meta 등)이

소유하고 통제하는 **중앙화된 플랫폼**을 통해 이루어짐.

Ex) 페이스북, 유튜브, 위키피디아...

Web1: 읽기만 가능한 인터넷!

World Wide Web

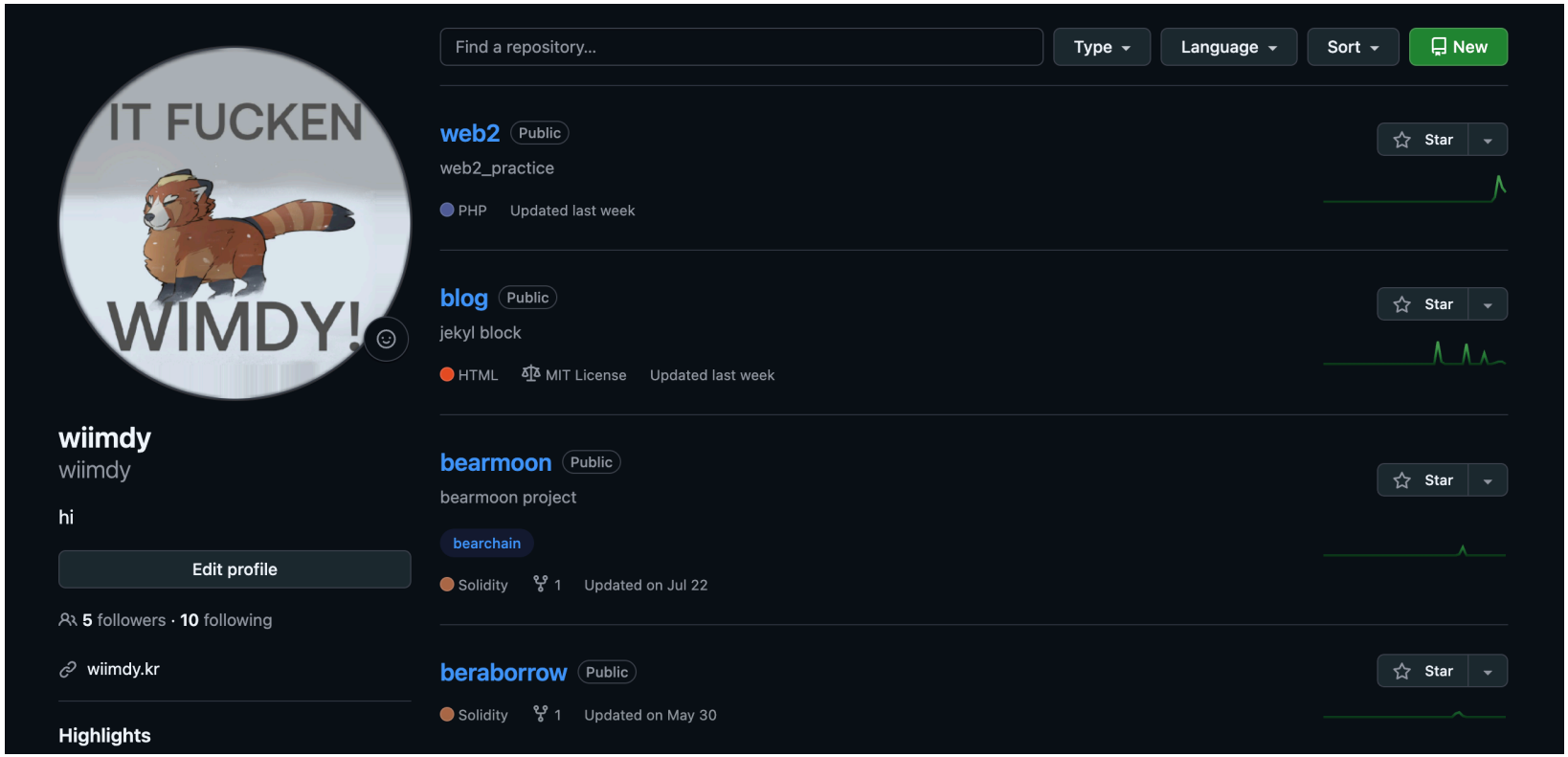
The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

https://info.cern.ch/hypertext/WWW/TheProject.html

Web2: 사용자가 콘텐츠를 직접 생성!



https://github.com/wiimdy?tab=repositories

- # Web3!

Web3: 사용자는 자신의 데이터와 디지털 자산을 누구의 허락도 없이 완벽하게 통제하고 소유

블록체인 기술을 기반으로, 특정 기업이 아닌 사용자가 직접
자신의 데이터와 디지털 자산을 소유하고 관리하는 새로운 인터넷입니다.

Web3의 특징: 탈중앙성, 퍼블릭 블록체인, 투명성..

■ Web3 세줄 요약

1. 탈중앙성
- 중앙 서버나 관리 주체 없이, 네트워크 참여자들이 공동으로 시스템을 운영하고 검증.

2. 투명성과 불변성
- 모든 소스 코드와 거래 기록이 블록체인에 영구적으로 공개.

3. 사용자 소유권과 직접적인 가치 이동
- 사용자는 자신의 디지털 자산을 지갑에 직접 소유하며, 은행 같은 중개자 없이 가치를 주고받음.

퀴즈 1.

- Q1: Web3 서비스에서 사용자의 데이터(예: 거래 기록)는 기본적으로 어디에 저장될까요?

1. 새 서비스를 운영하는 회사의 중앙 데이터베이스
2. 여러 컴퓨터에 분산되어 모두가 공유하는 블록체인
3. 사용자의 개인 컴퓨터 하드 드라이브
4. 클라우드 서버 (예: AWS, Google Cloud)

퀴즈 1.

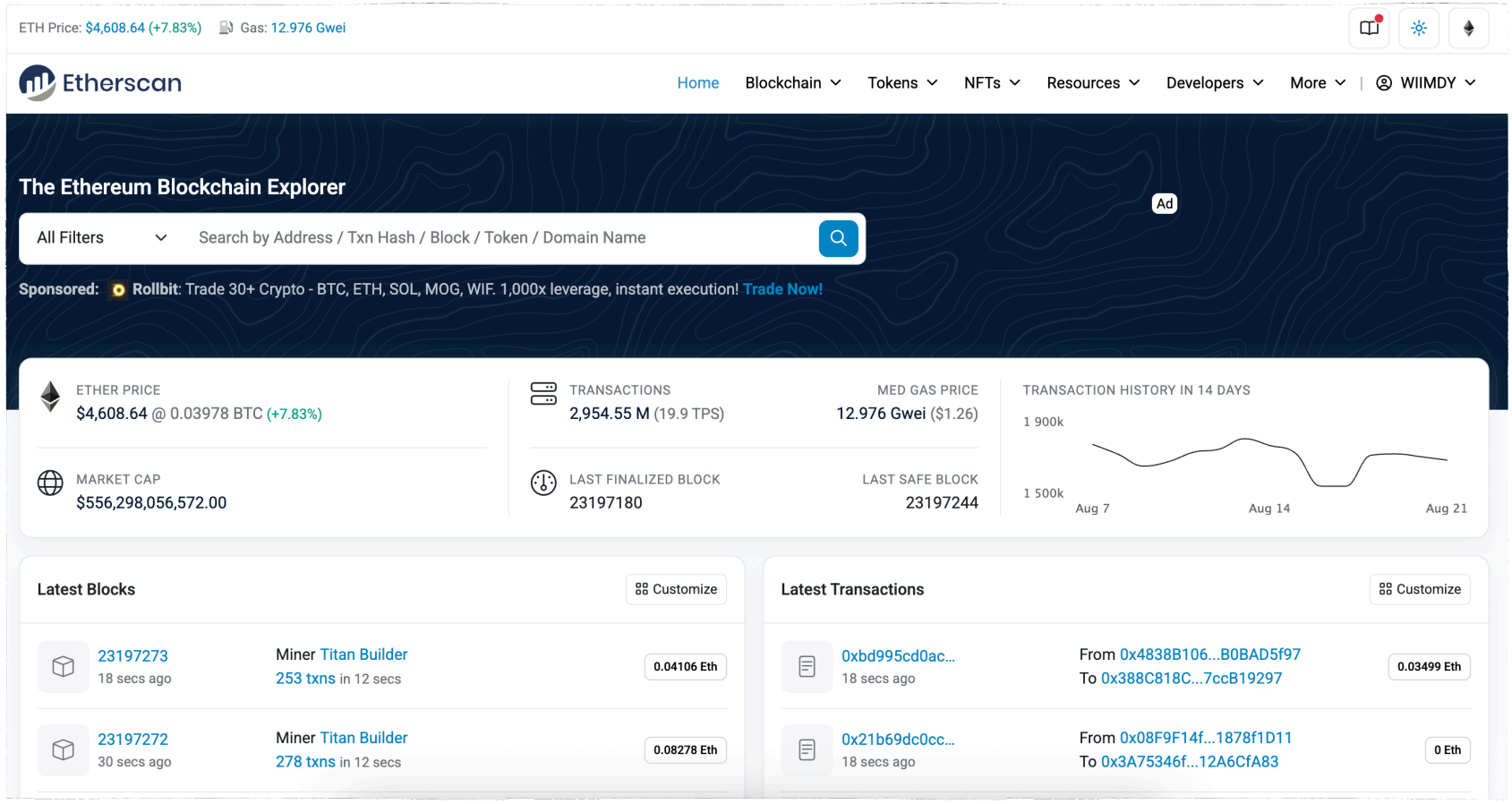
- Q1: Web3 서비스에서 사용자의 데이터(예: 거래 기록)는 기본적으로 어디에 저장될까요?

1. 새 서비스를 운영하는 회사의 중앙 데이터베이스
2. 여러 컴퓨터에 분산되어 모두가 공유하는 블록체인
3. 사용자의 개인 컴퓨터 하드 드라이브
4. 클라우드 서버 (예: AWS, Google Cloud)

02

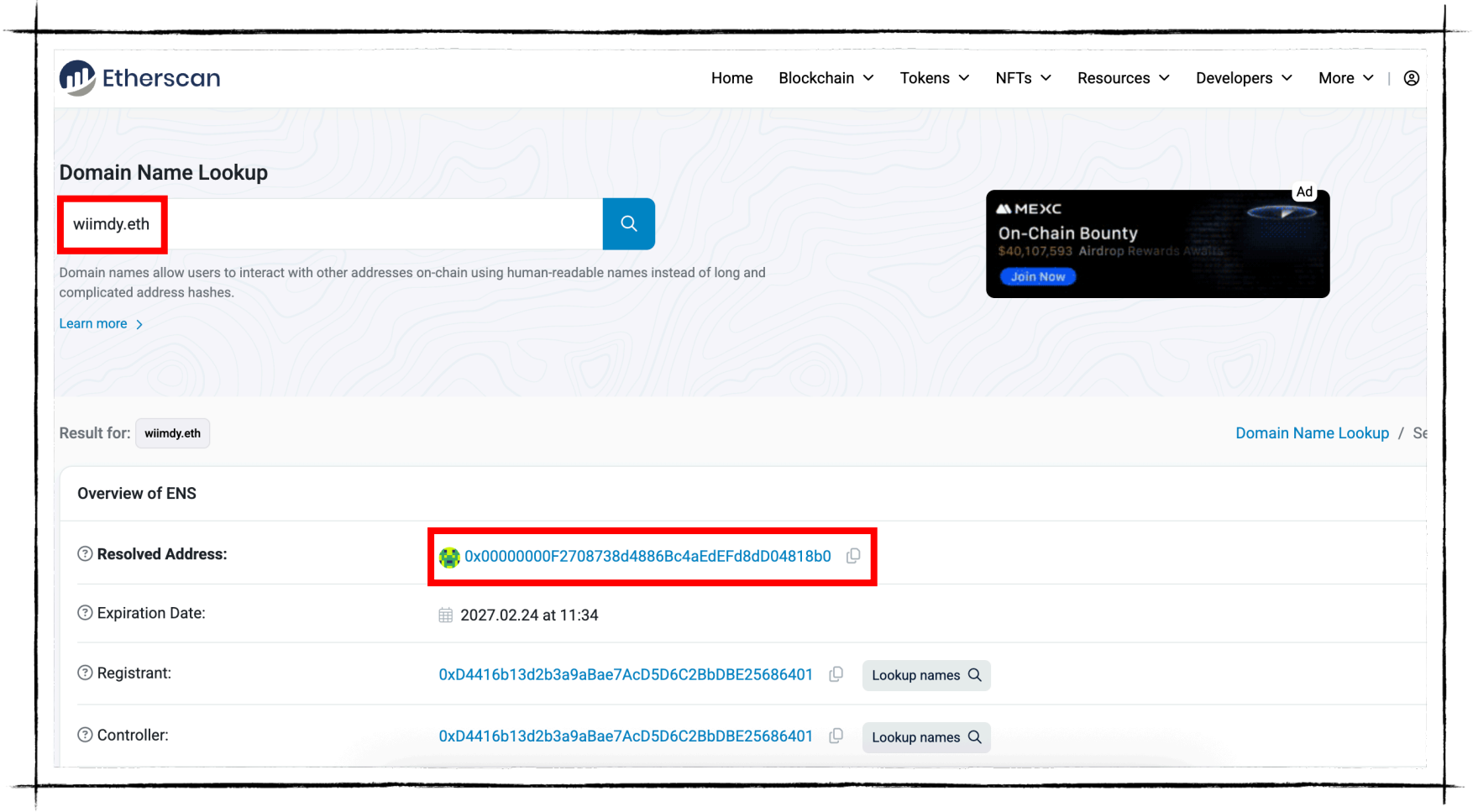
Web3 보안 맛보기

이더리움의 스캔 사이트



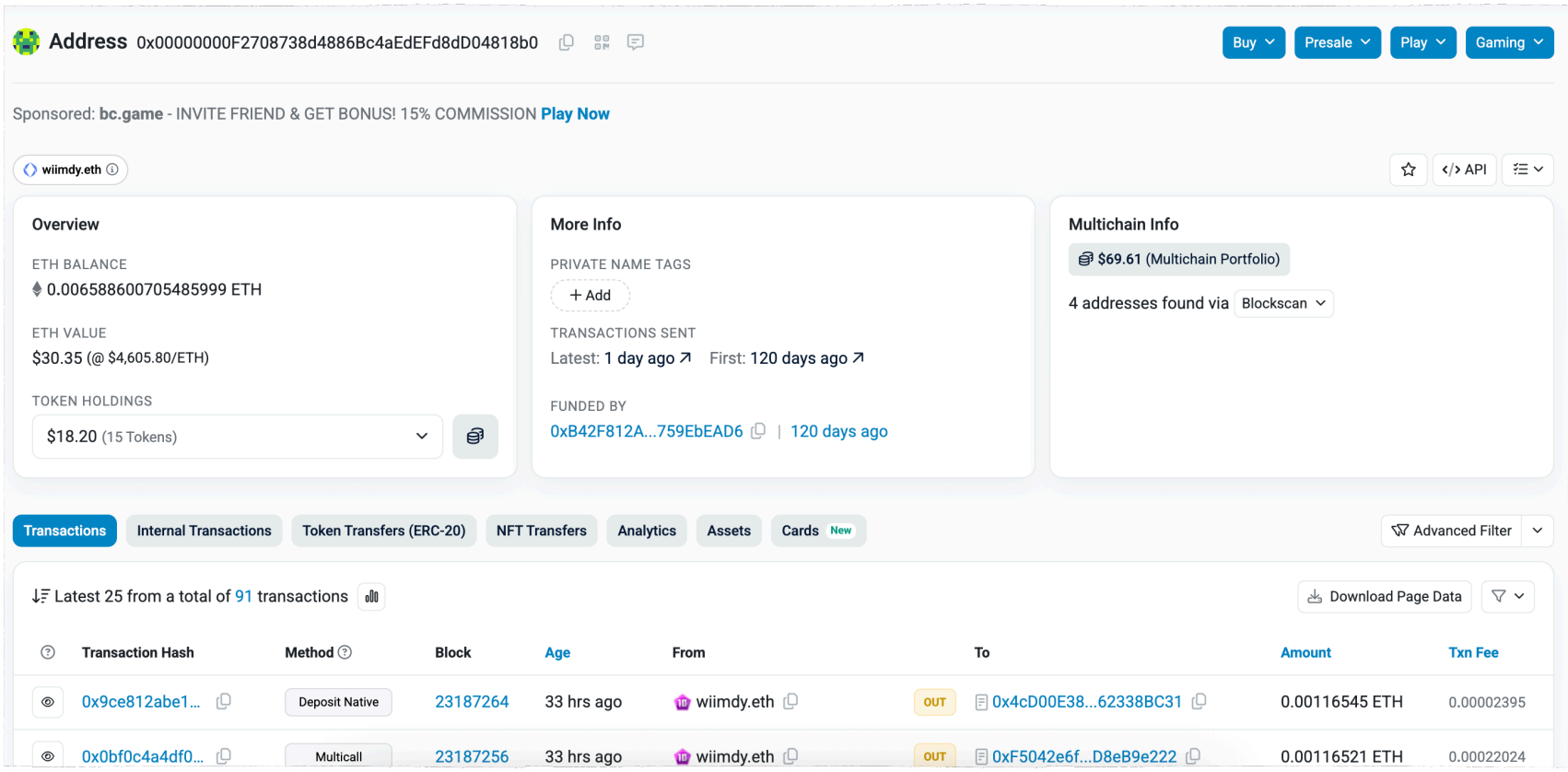
https://etherscan.io/

ENS와 지갑 주소



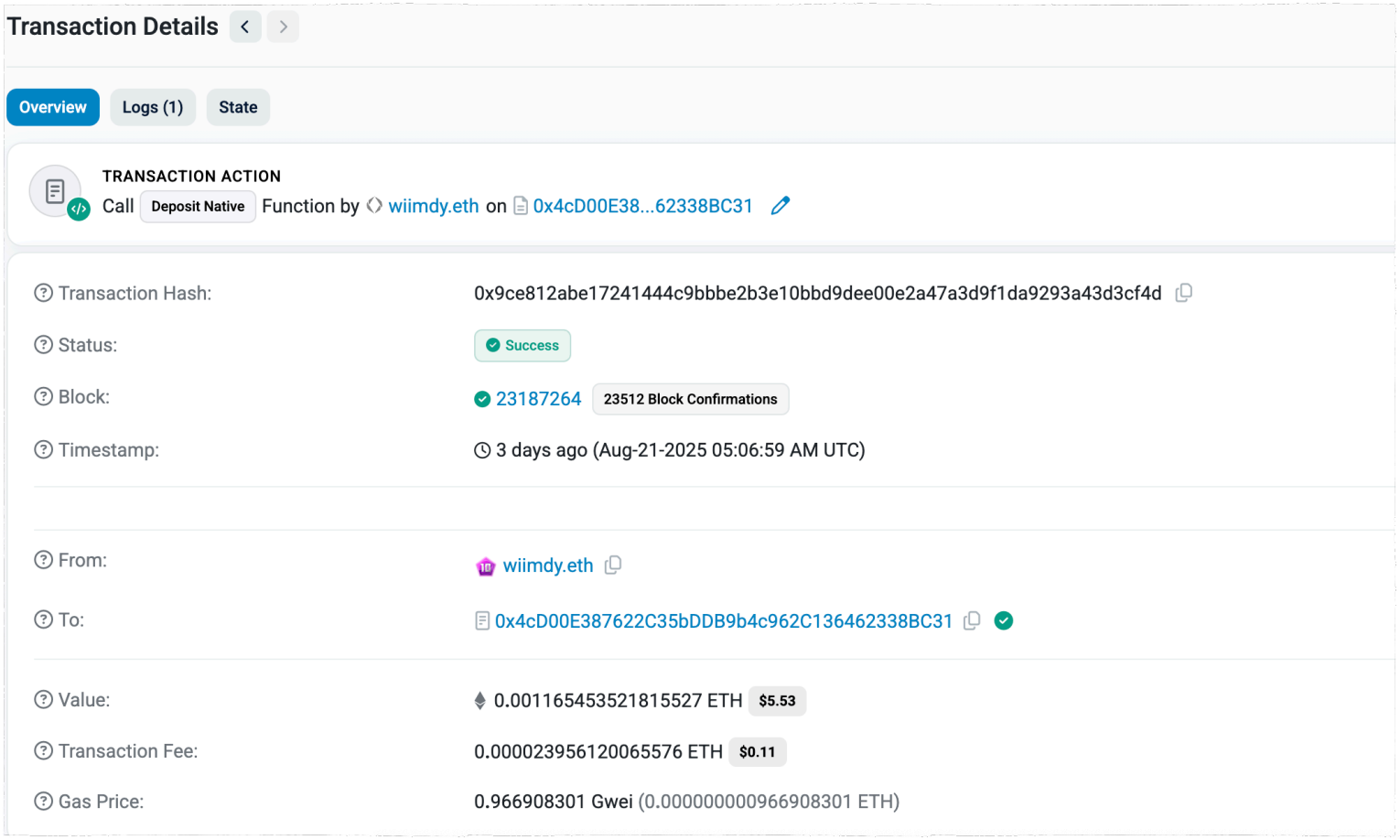
https://etherscan.io/name-lookup-search?id=wiimdy.eth

주소 조회하기



https://etherscan.io/address/0x00000000F2708738d4886Bc4aEdEFd8dD04818b0

■ 주소 거래내역 분석



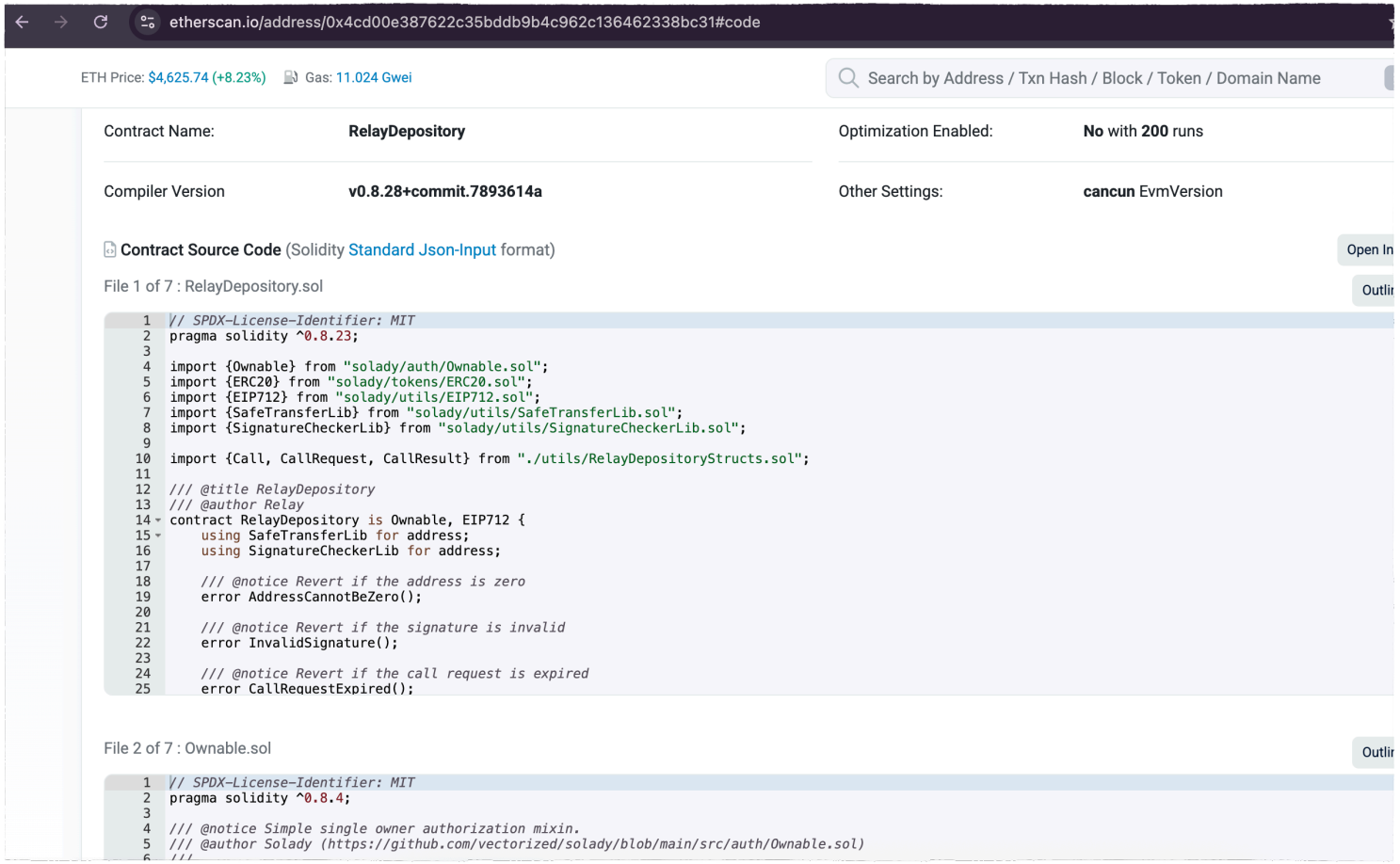
https://etherscan.io/tx/

■ 주소 거래내역 분석

Ether Price:	$4,224.96 / ETH
Gas Limit & Usage by Txn:	32,650 | 24,776 (75.88%)
Gas Fees:	Base: 0.234888175 Gwei | Max: 0.966908301 Gwei | Max Priority: 0.734584134 Gwei
Burnt & Txn Savings Fees:	🔥 Burnt: 0.0000058195894238 ETH ($0.03) 🏷 Txn Savings: 0 ETH ($0.00)
Other Attributes:	Txn Type: 2 (EIP-1559) Nonce: 81 Position In Block: 73
Input Data:	Function: depositNative(address depositor,bytes32 id) *** MethodID: 0x49290c1c [0]: 000000000000000000000000f2708738d4886bc4aedefd8dd04818b0 [1]: 5e2d158d1101bf629497edf26a0c83251927aac1afec6a581a0c4837841d9e08
	View Input As ⌄ Decode Input Data View In Decoder Advanced Filter Write Contract
More Details:	— Click to show less

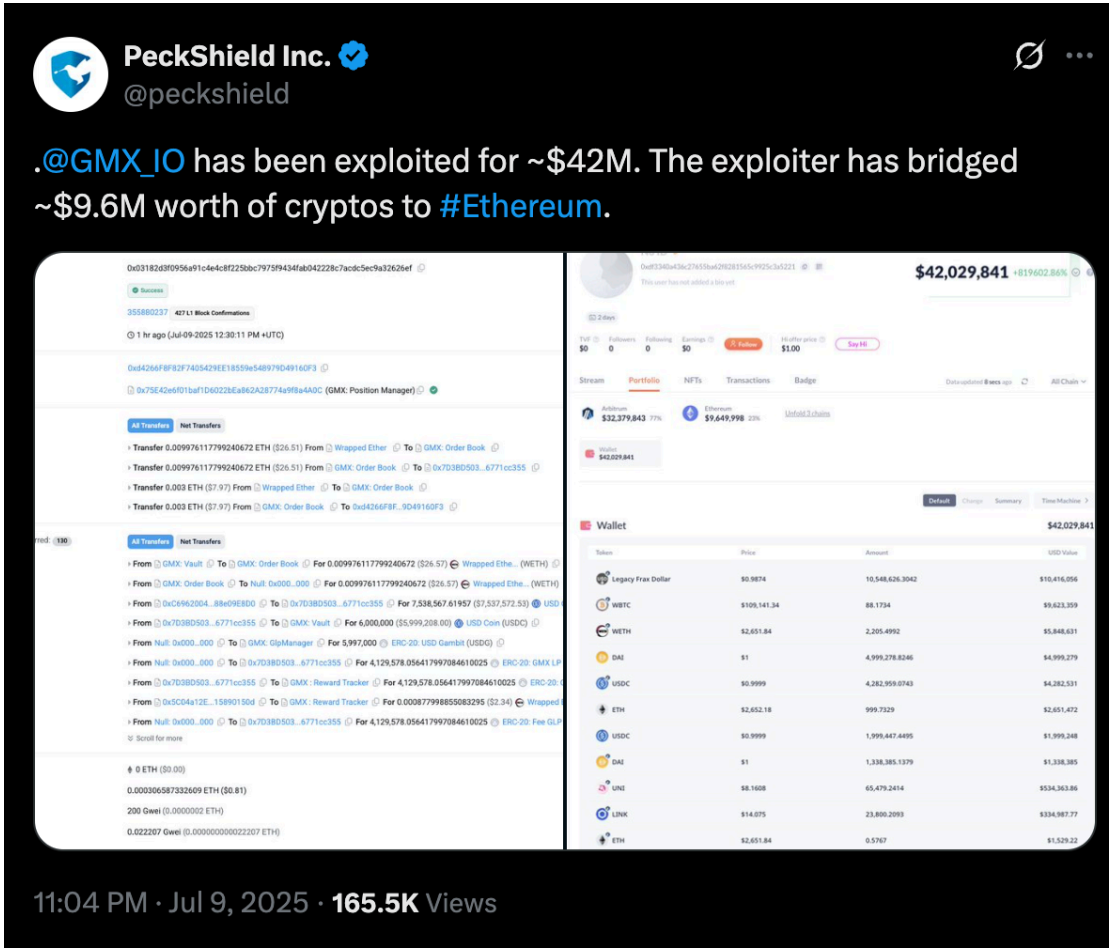
https://etherscan.io/tx/

Web3의 활동 내역 분석



https://etherscan.io/address/0x4cd00e387622c35bddb9b4c962c136462338bc31#code

실제 해킹 사건 분석



https://x.com/peckshield/status/1942947860645134450

지엠엑스🫐
@GMX_IO

Arbitrum의 GMX V1 GLP 풀에서 취약점 공격이 발생했습니다. 약 4천만 달러 상당의 토큰이 GLP 풀에서 알 수 없는 지갑으로 이체되었습니다.

GMX는 보안을 최우선으로 생각했으며, GMX 스마트 컨트랙트는 최고 보안 전문가들의 수많은 감사를 거쳤습니다. 따라서 이번 조사에서 모든 핵심 참여자는 조작이 어떻게 발생했는지, 그리고 어떤 취약점이 이를 가능하게 했는지 조사하고 있습니다. 보안 파트너들도 적극적으로 참여하여 발생한 사건을 철저히 파악하고 관련 위험을 최대한 신속하게 최소화하고 있습니다. Arbitrum의 주요 목표는 복구와 문제의 근본 원인 파악입니다.

조치 사항:

추가 공격 벡터를 차단하고 사용자를 추가적인 부정적 영향으로부터 보호하기 위해 Arbitrum과 Avalanche 모두에서 GMX V1 거래, GLP 발행 및 상환을 비활성화했습니다.

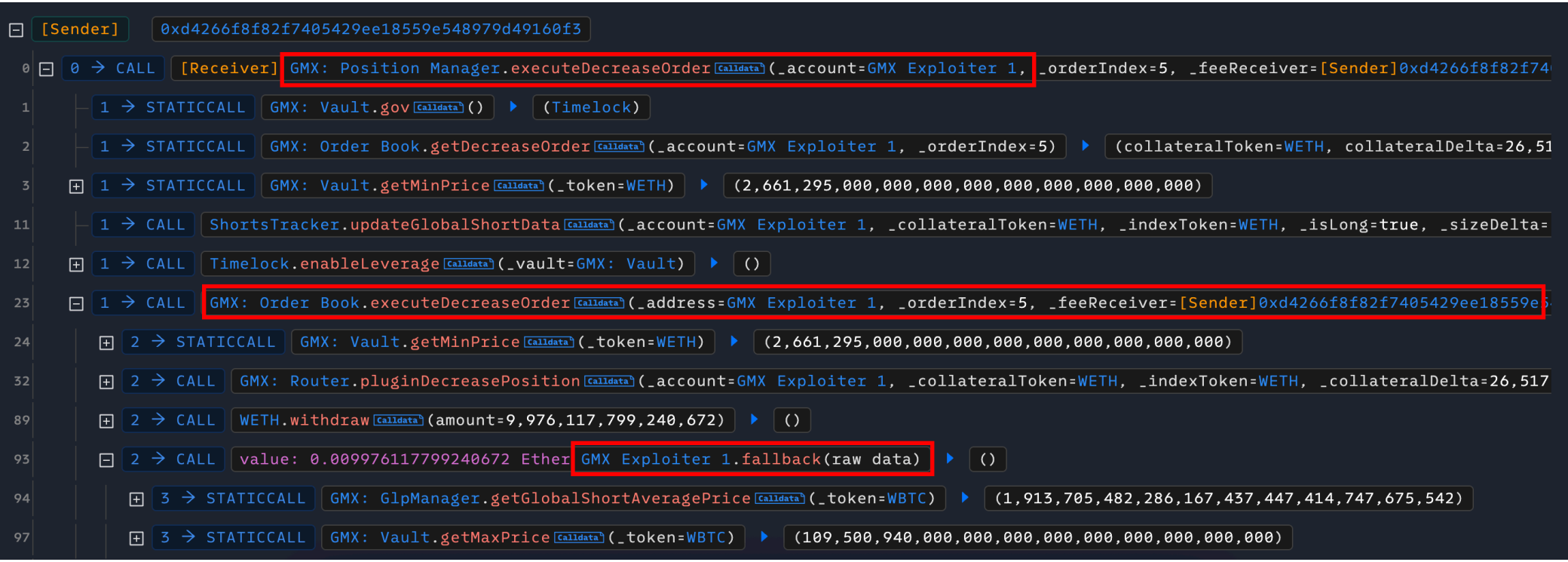
취약점 범위: 이 취약점 공격은 GMX V2, 해당 마켓, 유동성 풀, 그리고 GMX 토큰 자체에는 영향을 미치지 않습니다.

이용 가능한 정보에 따르면 해당 취약점은 GMX V1 및 GLP 풀에 국한됩니다. 더욱 완전하고 검증된 정보가 확보되는 대로 자세한 사고 보고서를 발표하겠습니다.

오후 11시 35분 · 2025년 7월 9일 · **325.7K** 조회수

https://x.com/GMX_IO/status/1942955807756165574

▪ 공격 로그 분석



https://app.blocksec.com/explorer/tx/arbitrum/0x03182d3f0956a91c4e4c8f225bbc7975f9434fab042228c7acdc5ec9a32626ef

▪ 실제 취약점 코드

master gmx-contracts / contracts / core / OrderBook.sol

Code Blame 984 lines (881 loc) · 31.6 KB

```
865             order.indexToken,
866             order.collateralDelta,
867             order.sizeDelta,
868             order.isLong,
869             address(this)
870         );
871
872         // transfer released collateral to user
873         if (order.collateralToken == weth) {
874             _transferOutETH(amountOut, payable(order.account));
875         } else {
876             IERC20(order.collateralToken).safeTransfer(order.account, amountOut);
877         }
878
```

https://github.com/gmx-io/gmx-contracts/blob/master/contracts/core/OrderBook.sol#L874

Root cause 분석

1. 공격자가 이 함수 실행

2. receiver를 인자로 넘김

3. receiver(exploiter) 코드를 실행

```
function executeDecreaseOrder(
    // transfer released collateral to user
    if (order.collateralToken == weth) {
        _transferOutETH(amountOut, payable(order.account));
    } else {
        IERC20(order.collateralToken).safeTransfer(order.account, amountOut);
    }

    // pay executor
    _transferOutETH(order.executionFee, _feeReceiver);
}

function _transferOutETH(
    uint256 _amountOut,
    address payable _receiver
) {
    IWETH(weth).withdraw(_amountOut);
    sendValue(_receiver, _amountOut);
}
function sendValue(address payable recipient, uint256 amount) {
    require(address(this).balance >= amount, "Address: insufficient balance");

    // solhint-disable-next-line avoid-low-level-calls, avoid-call-value
    (bool success, ) = recipient.call{value: amount}("");
    require(
        success,
        "Address: unable to send value, recipient may have reverted"
    );
```

Root cause 분석

exploiter를 call 하면 어떻게 되는가?

=> exploiter가 원하는 함수 실행 가능

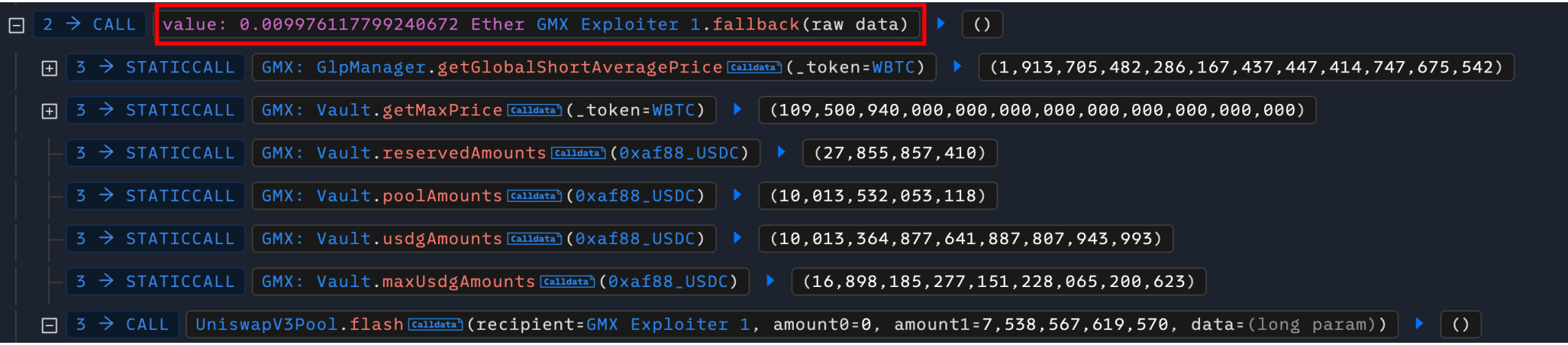
=> 이게 왜 문제?

=> 실행 흐름이 exploiter에게 넘어가서 거래소의 가격을 바꿀수 있음

(레버리지 숏 포지션을 열며..)

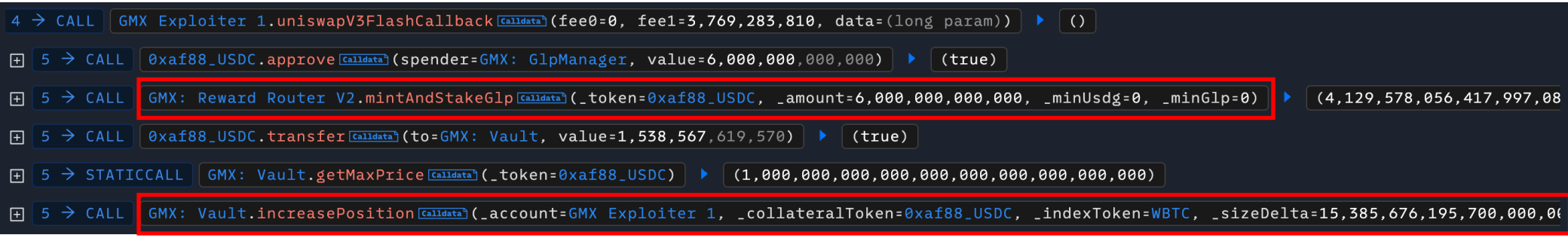
Root cause 분석

공격자에게 실행 흐름이 넘어 갔을 때



Root cause 분석

어느정도 GLP에 돈을 맡기고 레버리지 숏을 한다.

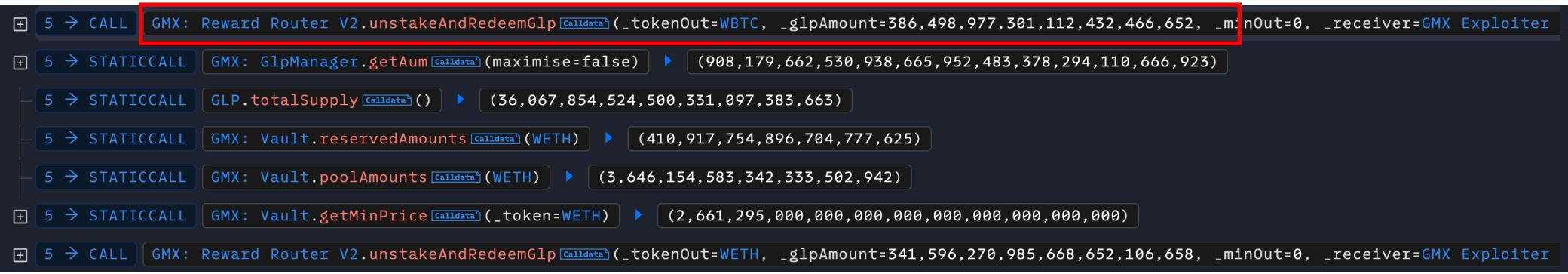


레버리지: 거래소에 돈을 빌려서 투자를 하는 기술

숏: 하락에 배팅하기 (가격이 떨어질 것이라 예상)

Root cause 분석

GLP의 가격이 급격히 올라 다시 GLP를 판다.

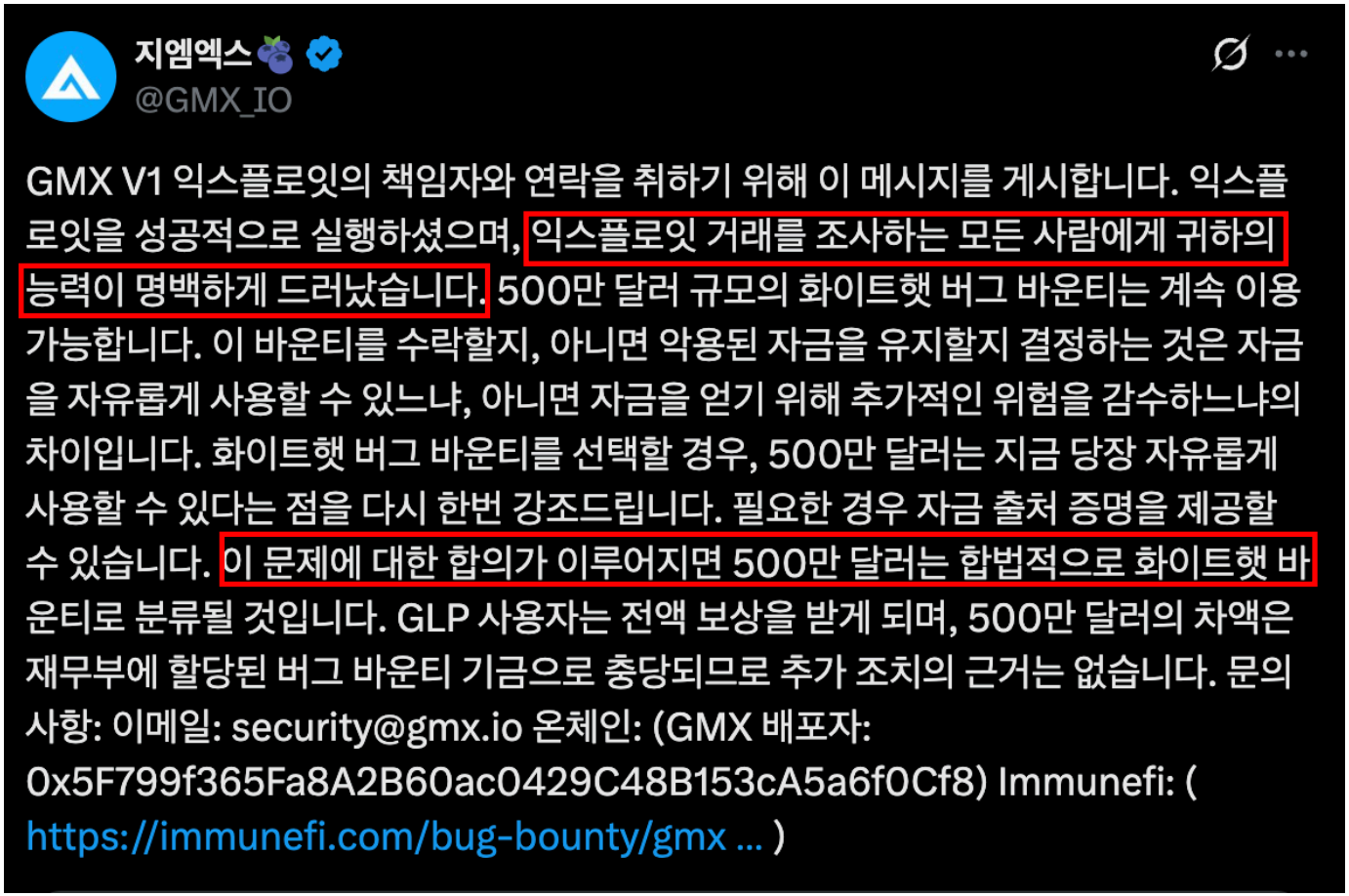


왜 GLP의 가격이 급격히 올랐는가?

=> GLP 가격 = (user_GLP / total_GLP_supply) * AUM (이놈이 급격히 상승)

AUM = (금고에 있는 모든 코인 총합) + (**숏 포지션의 손실) - (숏 포지션의 이익)

그래서 어떻게 되었는가?

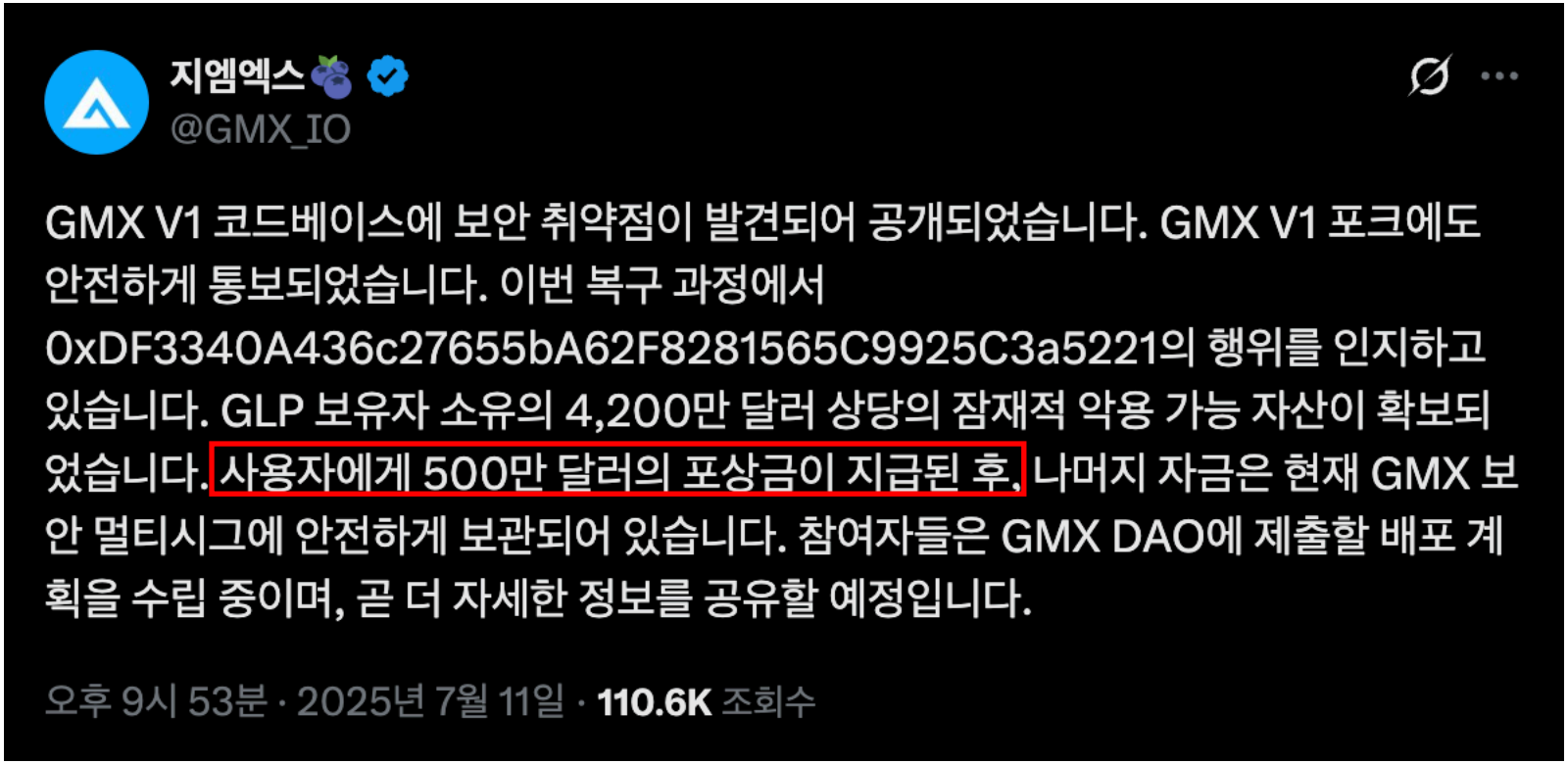


지엠엑스 🫐
@GMX_IO

GMX V1 익스플로잇의 책임자와 연락을 취하기 위해 이 메시지를 게시합니다. 익스플로잇을 성공적으로 실행하셨으며, 익스플로잇 거래를 조사하는 모든 사람에게 귀하의 능력이 명백하게 드러났습니다. 500만 달러 규모의 화이트햇 버그 바운티는 계속 이용 가능합니다. 이 바운티를 수락할지, 아니면 악용된 자금을 유지할지 결정하는 것은 자금을 자유롭게 사용할 수 있느냐, 아니면 자금을 얻기 위해 추가적인 위험을 감수하느냐의 차이입니다. 화이트햇 버그 바운티를 선택할 경우, 500만 달러는 지금 당장 자유롭게 사용할 수 있다는 점을 다시 한번 강조드립니다. 필요한 경우 자금 출처 증명을 제공할 수 있습니다. 이 문제에 대한 합의가 이루어지면 500만 달러는 합법적으로 화이트햇 바운티로 분류될 것입니다. GLP 사용자는 전액 보상을 받게 되며, 500만 달러의 차액은 재무부에 할당된 버그 바운티 기금으로 충당되므로 추가 조치의 근거는 없습니다. 문의 사항: 이메일: security@gmx.io 온체인: (GMX 배포자: 0x5F799f365Fa8A2B60ac0429C48B153cA5a6f0Cf8) Immunefi: (https://immunefi.com/bug-bounty/gmx ...)

https://x.com/GMX_IO/status/1943342382503567586

▪ 해피엔딩...



지엠엑스 🫐
@GMX_IO

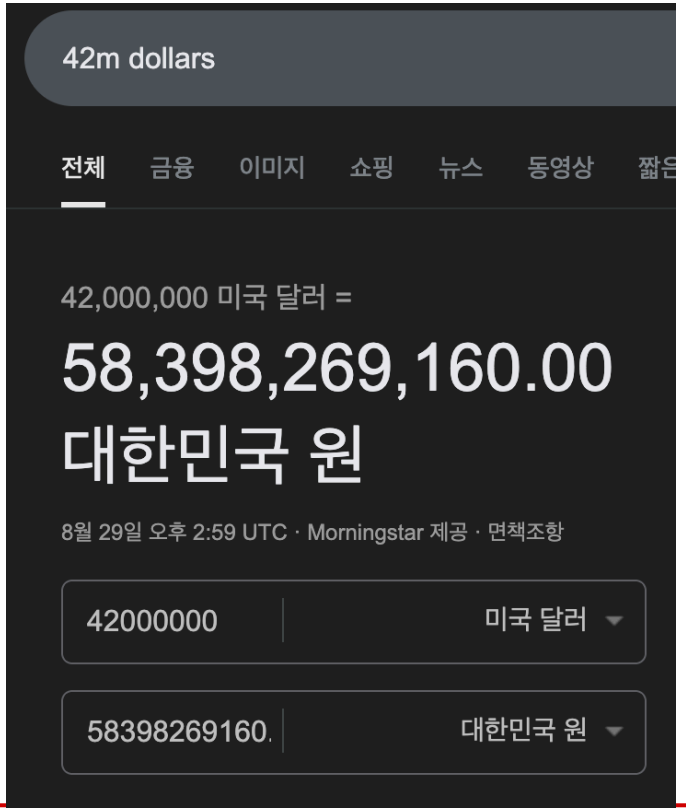
GMX V1 코드베이스에 보안 취약점이 발견되어 공개되었습니다. GMX V1 포크에도 안전하게 통보되었습니다. 이번 복구 과정에서 0xDF3340A436c27655bA62F8281565C9925C3a5221의 행위를 인지하고 있습니다. GLP 보유자 소유의 4,200만 달러 상당의 잠재적 악용 가능 자산이 확보되었습니다. 사용자에게 500만 달러의 포상금이 지급된 후, 나머지 자금은 현재 GMX 보안 멀티시그에 안전하게 보관되어 있습니다. 참여자들은 GMX DAO에 제출할 배포 계획을 수립 중이며, 곧 더 자세한 정보를 공유할 예정입니다.

오후 9시 53분 · 2025년 7월 11일 · **110.6K** 조회수

https://x.com/GMX_IO/status/1943654914749534380

왜 Web3가 재밌는가?

오픈된 코드, 오픈된 로그 => 해킹 사건들에 대해 모든 정보가 공개되어 있음
분석이 가능하고 취약점 공부가 재밌음



https://x.com/BlockSecTeam/article/1942990338731229379

▪ 결론

영어 실력과 코드를 이해할 수 있는 solidity 실력을 갖춘다면
모든 해킹 사건들을 분석할 수 있다.

=> 모든 해킹 사건들을 그때 환경으로 세팅으로 해서 exploit 코드를 작성할 수 있다!

티오리의 Web3 팀인 chainlight에서 작성한 해킹 사건 모음집

https://drive.google.com/file/d/1zt30caMFy64RehSFCA4P-jP-31IST9LD/view

03

Web3 보안, 어떻게 시작할까?

어떻게 공부할 수 있는가? - 드림핵

Smart Contract Security

★ 10.0 (1)

Tier 2 Easy Skill Path Blockchain

본 Path에서는 이더리움과 Solidity 기반 스마트 컨트랙트의 구조와 보안 취약점을 학습하고 실습을 통해 실제 보안 사고로 이어진 핵심 취약점들을 직접 분석합니다. 스마트 컨트랙트에서 자주 발생하는 보안 문제를 중심으로 Reentrancy, Integer Overflow, Storage 조작 등 실전 공격 기법과 방어 전략을 학습합니다. 본 Path를 모두 완료하면 스마트 컨트랙트 감사(Audit)에 필요한 기초 지식을 함양하여 Web3 분야의 버그 바...

학습목표 Unit 구성 수강 후기 1

이런 이유로 이 Path를 추천해요

블록체인에서 스마트 컨트랙트는 단 한 줄의 코드에만 실수가 있어도 수백억 원 규모의 자산 손실로 이어질 수 있을 만큼, 보안이 무엇보다 중요한 분야입니다. 특히 재진입성(Reentrancy), Fallback 함수 등으로 인해 발생하는 취약점은 실제 해킹 사례에서도 자주 등장하고 있습니다. 본 Path에서는 이더리움 네트워크와 EVM의 구조를 이해하는 것부터 시작해서 Solidity 기반 컨트랙트에서 발생하는 보안 취약점을 실습용 컨트랙트 환경에서 실제로 실습해볼 수 있도록 구성되어 있습니다. 블록체인 보안 연구원, 스마트 컨트랙트 감사자(Auditor)와 같은 커리어에 필요한 역량을 쌓고자 하시는 분들을 위한 Path입니다.

이런 내용을 배워요

- 이더리움 및 EVM (Ethereum Virtual Machine) 구조 이해
- Solidity 문법과 스마트 컨트랙트 개발 기초

권장 소비자 가격 1250 코인

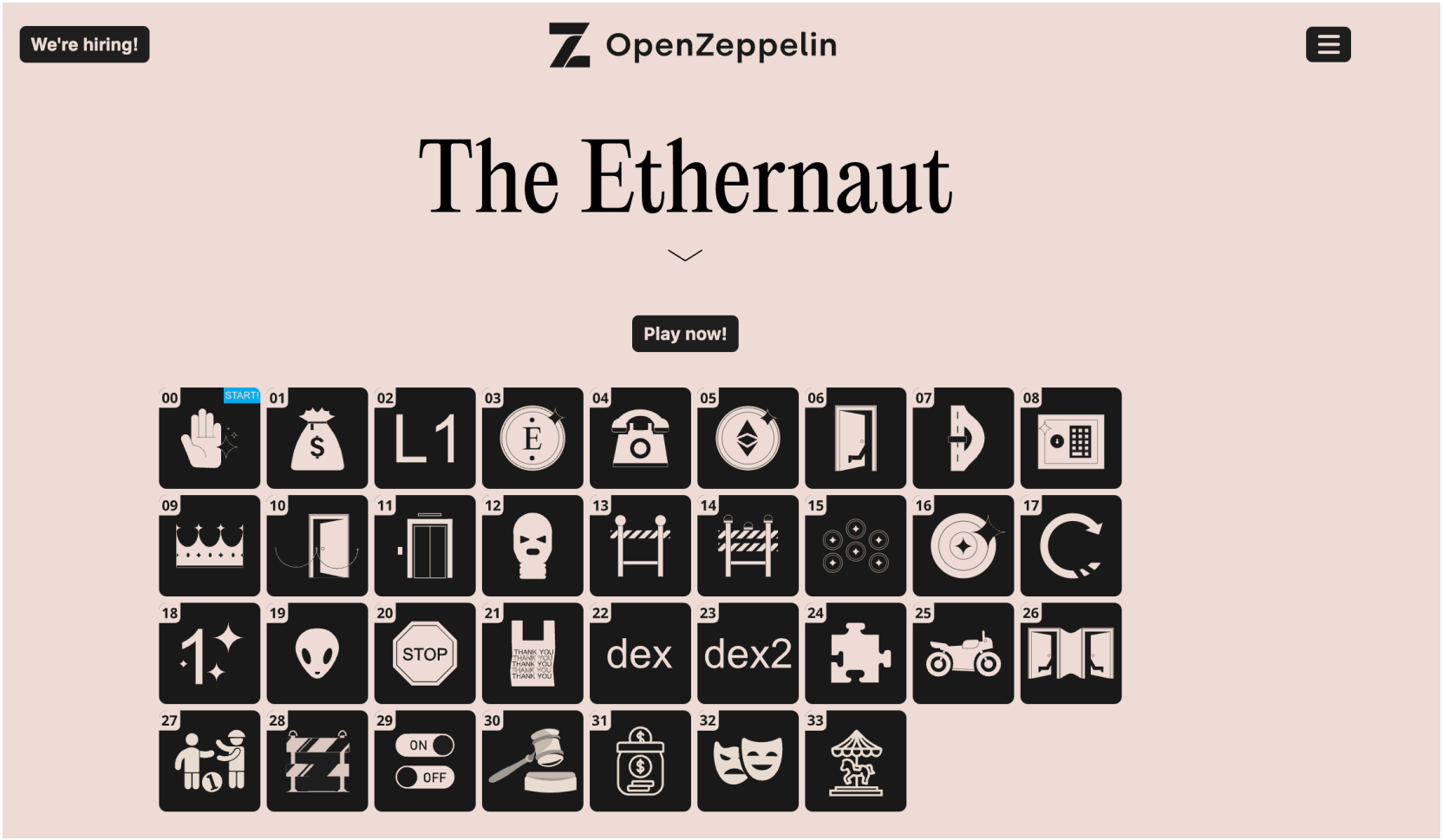
판매가 1250 코인

Pro 구독하면 무료

구독하고 무제한 수강하기

1250 코인으로 구매하기

어떻게 공부할 수 있는가? - Ethernaut



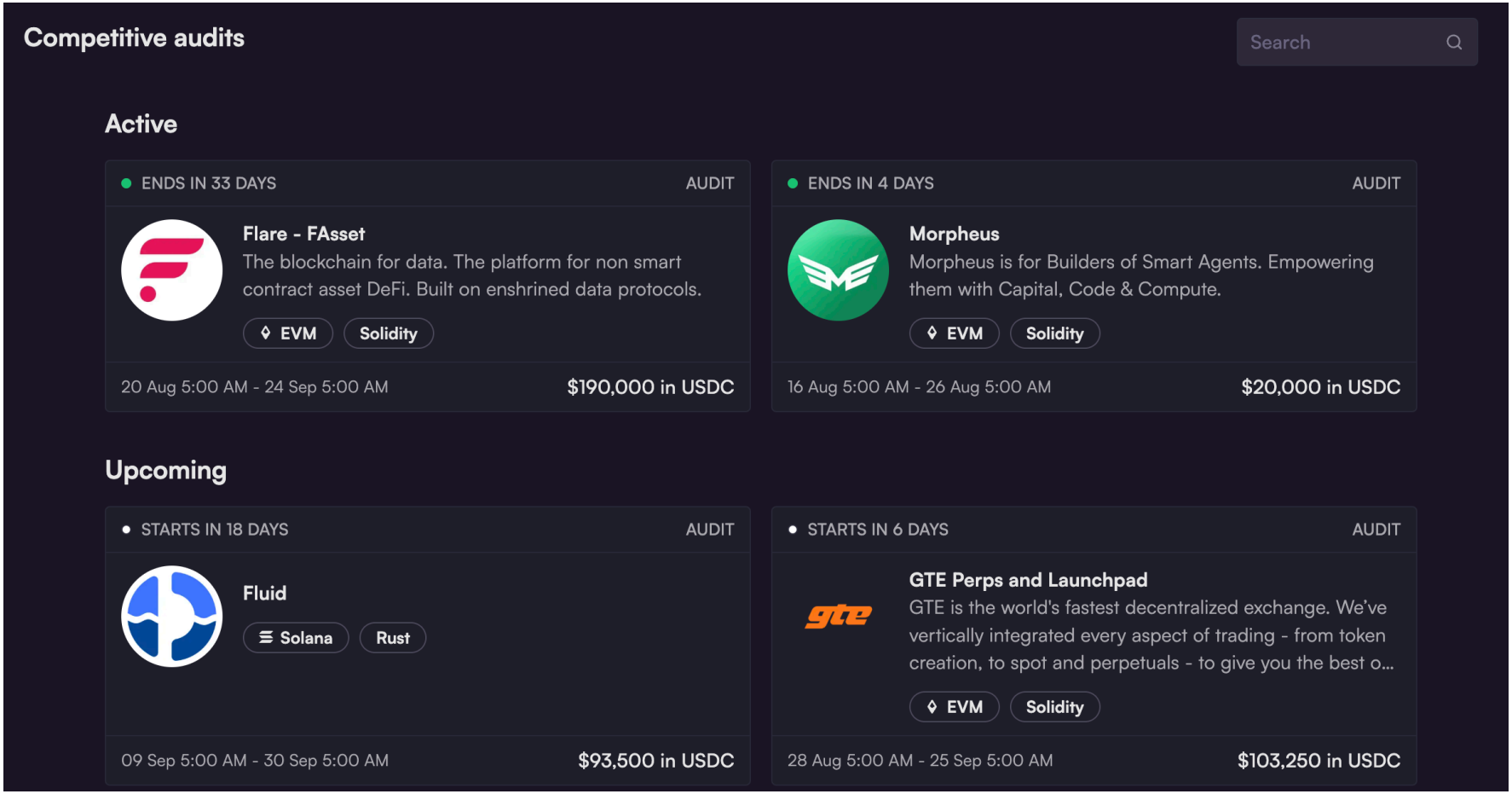
어떻게 공부할 수 있는가? - Ethernaut

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.12;

import "openzeppelin-contracts-06/math/SafeMath.sol";

contract Reentrance {
    using SafeMath for uint256;

    mapping(address => uint256) public balances;

    function donate(address _to) public payable {
        balances[_to] = balances[_to].add(msg.value);
    }

    function balanceOf(address _who) public view returns (uint256 balance) {
        return balances[_who];
    }

    function withdraw(uint256 _amount) public {
        if (balances[msg.sender] >= _amount) {
            (bool result,) = msg.sender.call{value: _amount}("");
            if (result) {
                _amount;
            }
            balances[msg.sender] -= _amount;
        }
    }

    receive() external payable {}
}
```

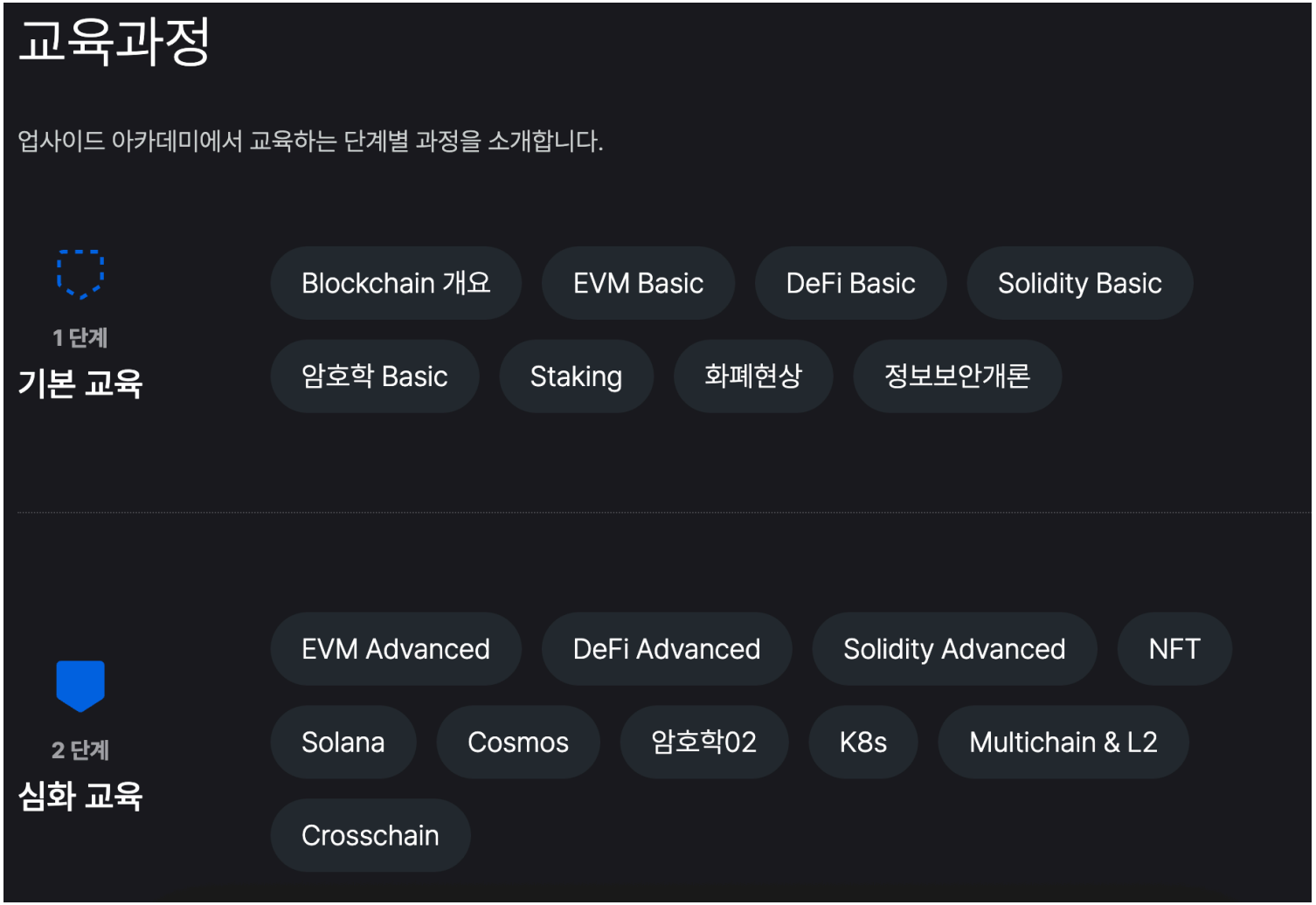
- 어떻게 공부할 수 있는가? - Ethernaut

```
function withdraw(uint256 _amount) public {
    if (balances[msg.sender] >= _amount) {
        (bool result,) = msg.sender.call{value: _amount}("");
        if (result) {
            _amount;
        }
        balances[msg.sender] -= _amount;
    }
}
```

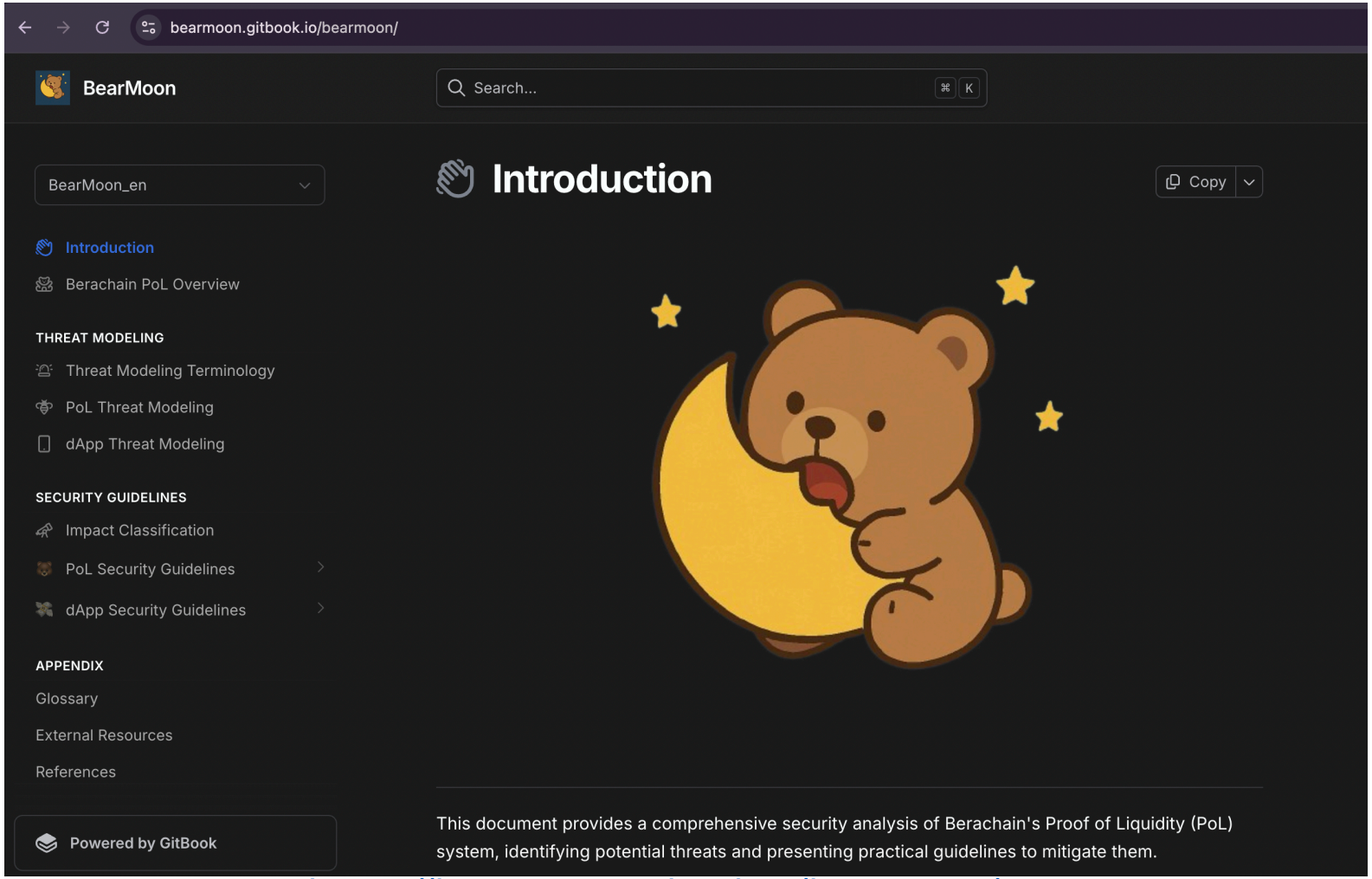
어떻게 공부할 수 있는가? - Competitive audit



어떻게 공부할 수 있는가? - Upside Academy



어떻게 공부할 수 있는가? - Upside Academy

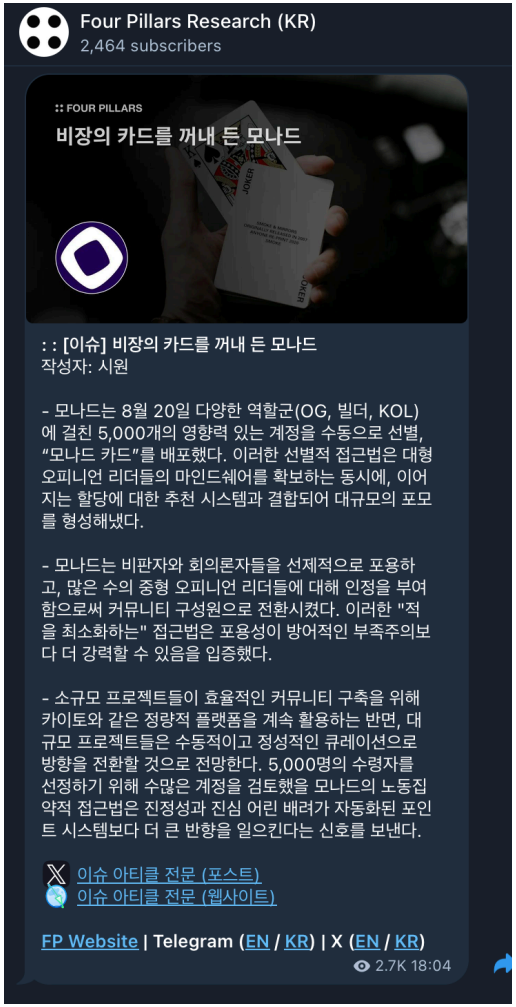


https://bearmoon.gitbook.io/bearmoon/

▪ 어떻게 공부할 수 있는가? - Upside Academy

https://upside.center/apply

▪ 어떻게 공부할 수 있는가? - 번외



■ 퀴즈 2.

- Q2: 이더리움에서 지갑 주소 및 거래 기록을 검색할 수 있는 사이트는?

1. 이더로그 (Etherlog)
2. 이더스캔 (Etherscan)
3. 이더서치 (Ethersearch)
4. 이더익스 (Etherex)

▪ 퀴즈 2.

- Q2: 이더리움에서 지갑 주소 및 거래 기록을 검색할 수 있는 사이트는?

1. 이더로그 (Etherlog)
2. 이더스캔 (Etherscan)
3. 이더서치 (Ethersearch)
4. 이더익스 (Etherex)

04

정리

Do not fear!


IT FUCKEN
WIMDY!
이 Web3 맛을
몰라서
저딴 소리
하는 거야.
헤이!
슈라이!
슈라이!
한 번
먹어보고
그런 말해!
No!
IT FUCKEN
WIMDY!

Do not fear!

Web3 솔직히 한번 해보고 싶은데 용어 같은게 너무 어렵다..

뭐 어떻게 배워야 하는지도, 해야하는지 모르겠다..

과거의 나

Do not fear!

하지만! 우리는 해커다!

Web3 한번 맛보면 다시 못 나간다.

재밌다.. 멋있다..

현재의 나

Q&A

cwm912@korea.ac.kr