CYBERSEC 2024
臺灣資安大會
5/14 Tue – 5/16 Thu
臺北南港展覽二館
Generative Future

Web3 Security Forum

Root Cause Analysis of Common DeFi Security Incidents

Speakers: York Chang, Alice Hsu

Who am I

About York Chang

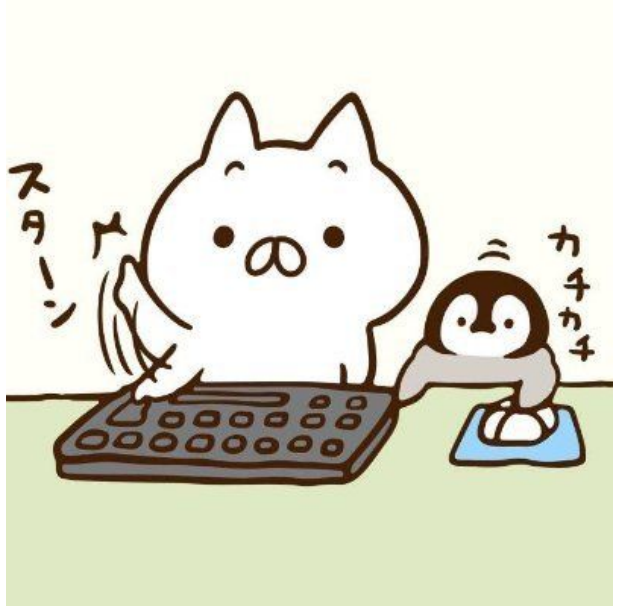
@YorkChang01

@yorkisinhere

- York is a Security Researcher, focusing on threat hunting, intelligence research and smart contract security.
- He is also a Blockchain Developer.
- He is one of the white hats at DeFiHackLabs.
- Certifications：CHFI, CEH, etc.
- Speaking Engagements: COSCUP, MOPCON, etc.

Who am I

About Alice Hsu



@AliceHsu_kou

@Alice_Hsu

- Alice is a Security Researcher.
- Her main research focuses on blockchain security.
- She has presented on the security issues of decentralized finance and smart contract detection technology at conferences such as HITCON PEACE 2022 CyberSec Salon and CyberSec 2022, etc.
- She is one of the white hats at DeFiHackLabs.

Agenda

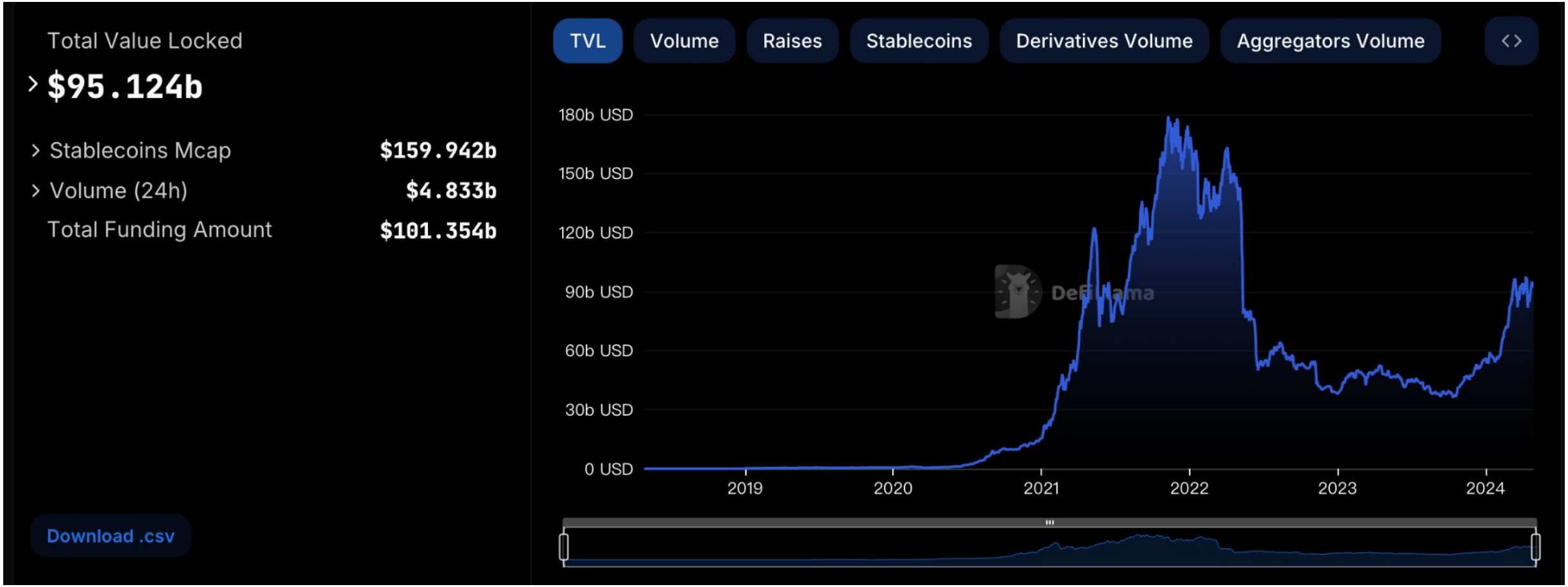
- Background
- Common DeFi Security Incidents Root Cause
  1. Business Logic
  2. Input Validation
  3. Price Manipulation
- Conclusion

What is DeFi ?

What is DeFi

- Decentralized Finance(DeFi)
- Refers to the emerging blockchain-based ecosystem of permissionless and transparent financial services.
- Including lending, derivatives, decentralized exchanges, asset management, stablecoin, insurance, borrowing, etc.
- Promoting the flourishing of blockchain technology.

DeFi Total Value Locked (TVL)



DefiLama available online: https://defillama.com/ (accessed on 2024.4.28)

DeFi Total Value Hacked (USD)

Generative Future

Total Value Hacked (USD)

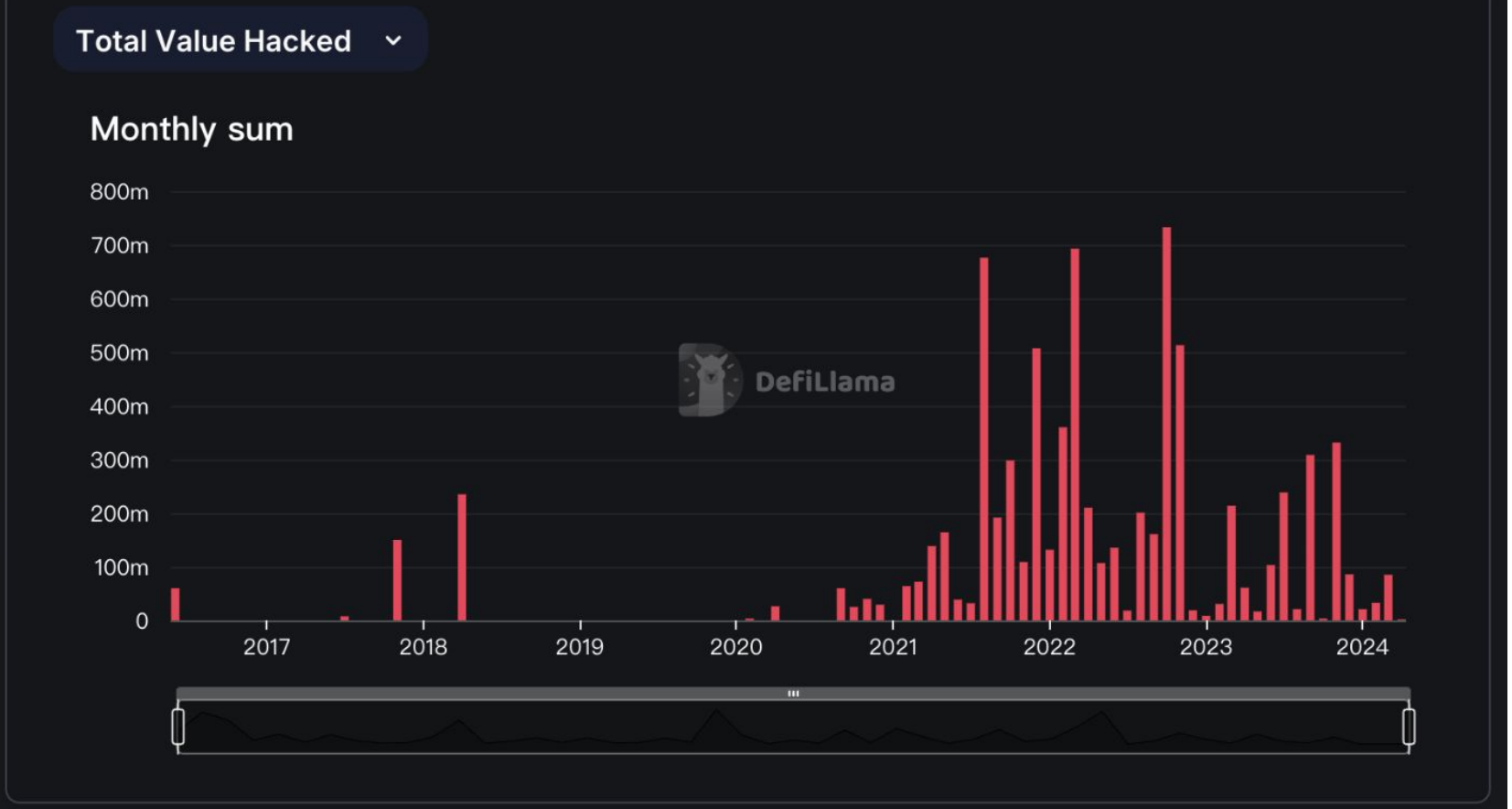
$7.77b

Total Value Hacked in DeFi (USD)

$5.85b

Total Value Hacked in Bridges (USD)

$2.83b



DefiLama available online: https://defillama.com/hacks (accessed on 2024.4.28)

So, What happens to DeFi ?

DeFi Security Risks

- Web2 related vulnerabilities
- Social engineering
- Rug pull
- Smart contract logic error

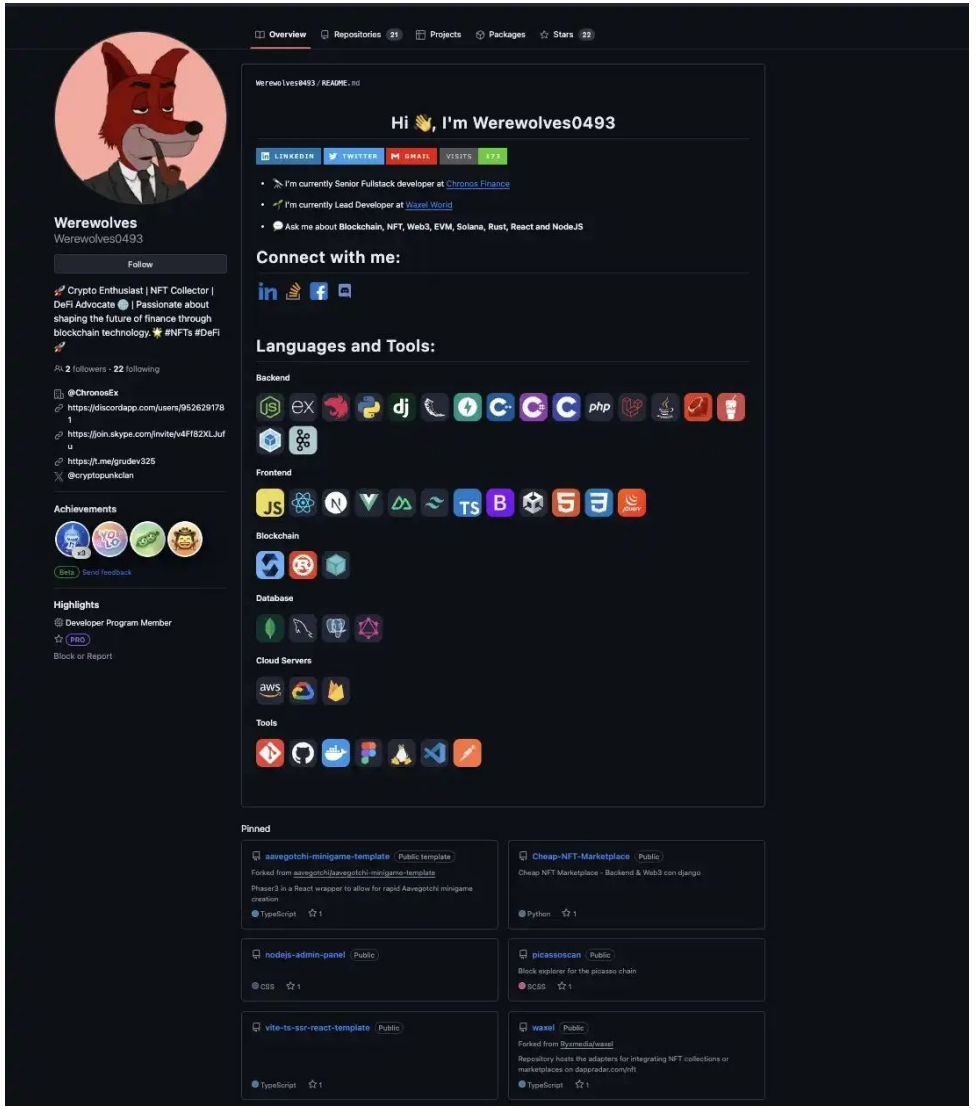
Web2 related vulnerabilities

- Private Key Compromise
  - 2024.02 PlayDapp (~$32M)
  - 2024.03 Mozaic Finance (~ $2.4M)

- 2023.07 Vyper compiler vulnerabilities (Curve finance) (~$70M)
  - Alchemix
  - JPEG’d
  - Metronome

Social engineering

- Fake website with mint / claim button
- Phishing email
- Insider threats
  - 2024.03 Munchables (~$62.8M)
  - LAZARUS GROUP
  - Rogue developer



Rug pull

- Deposited but permanently locked
- 2024.03 BitForex (~$57M)
- 2024.04 Glori Finance (~ $1.4M)

Smart contract logic error

Common DeFi Security Incidents Root Cause

(2024.1~4)

1. Business Logic
2. Input Validation
3. Price Manipulation

https://github.com/SunWeb3Sec/DeFiHackLabs

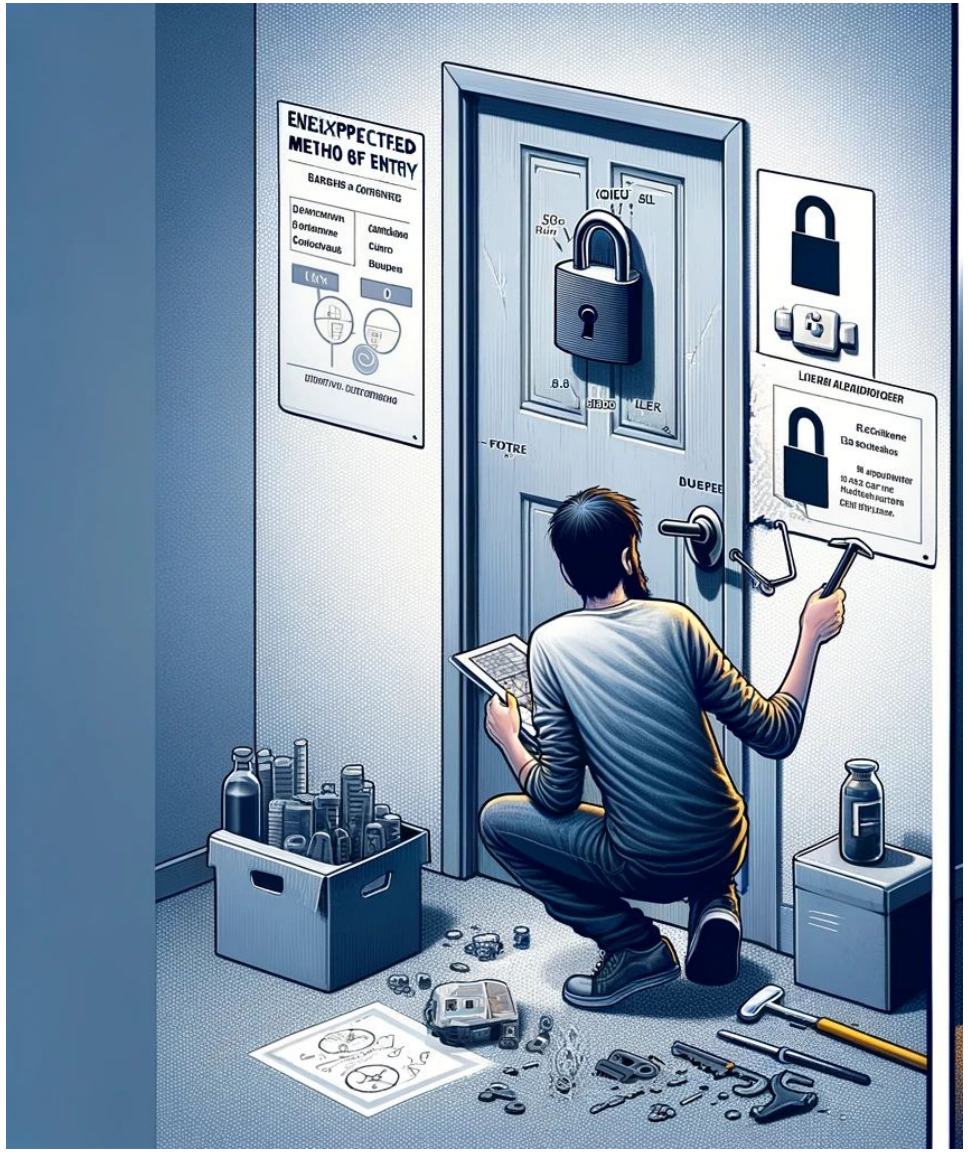
Business Logic

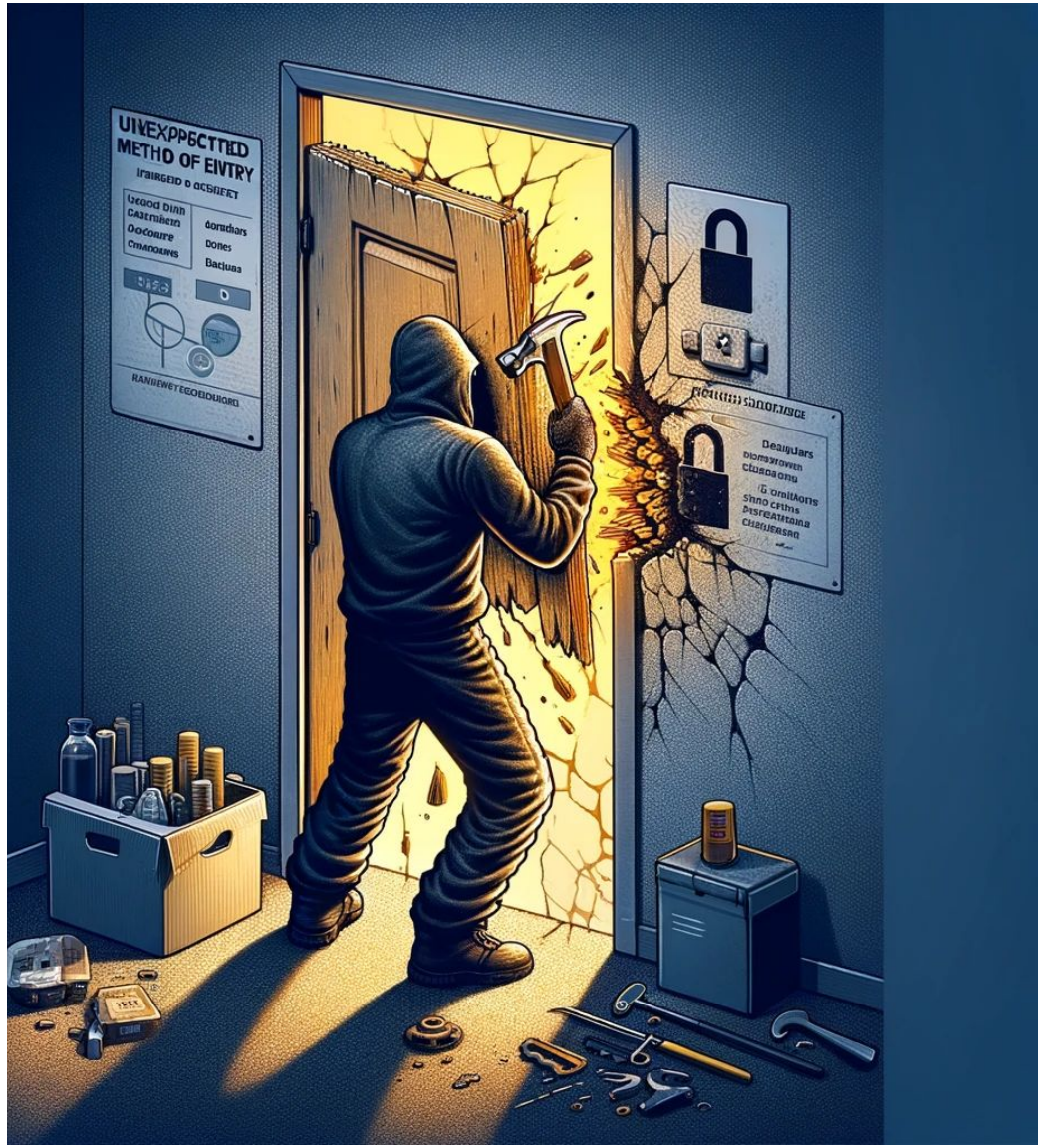
Business Logic

- A vulnerability or weakness in software or systems caused by a misunderstanding or incorrect implementation of business rules during the design or execution process.

What's different between a developer and an attacker ?

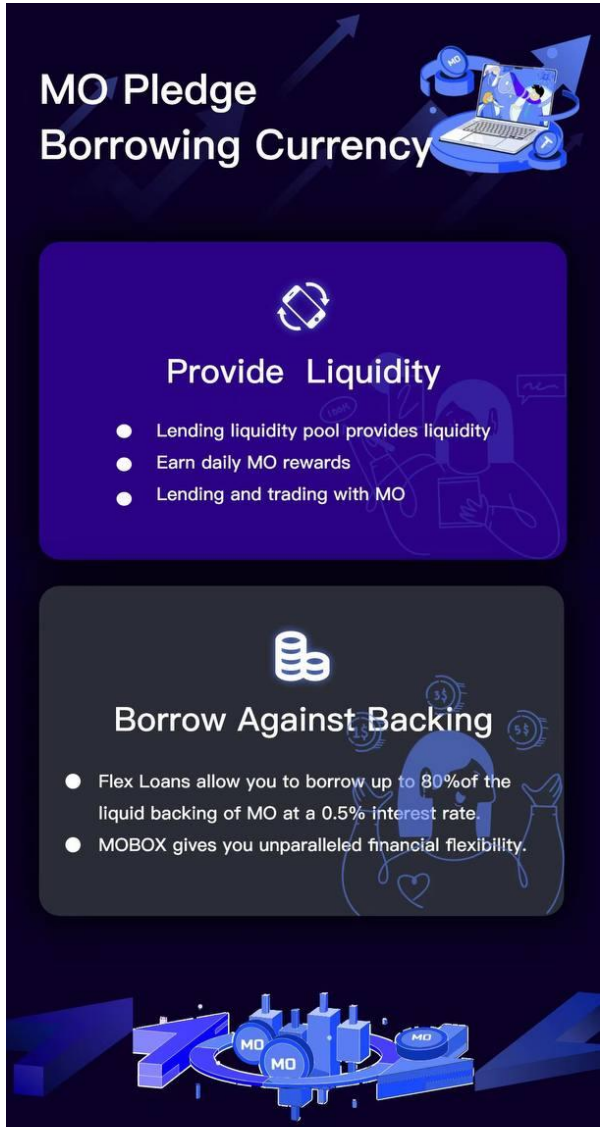
Generative Future





MOBOX Lending Protocol (MO Token)

- MOBOX is a web3 gaming platform where players can purchase NFTs or tokens to play games.
- In March this year, they designed a lending protocol to provide users with more diverse sources of funding.



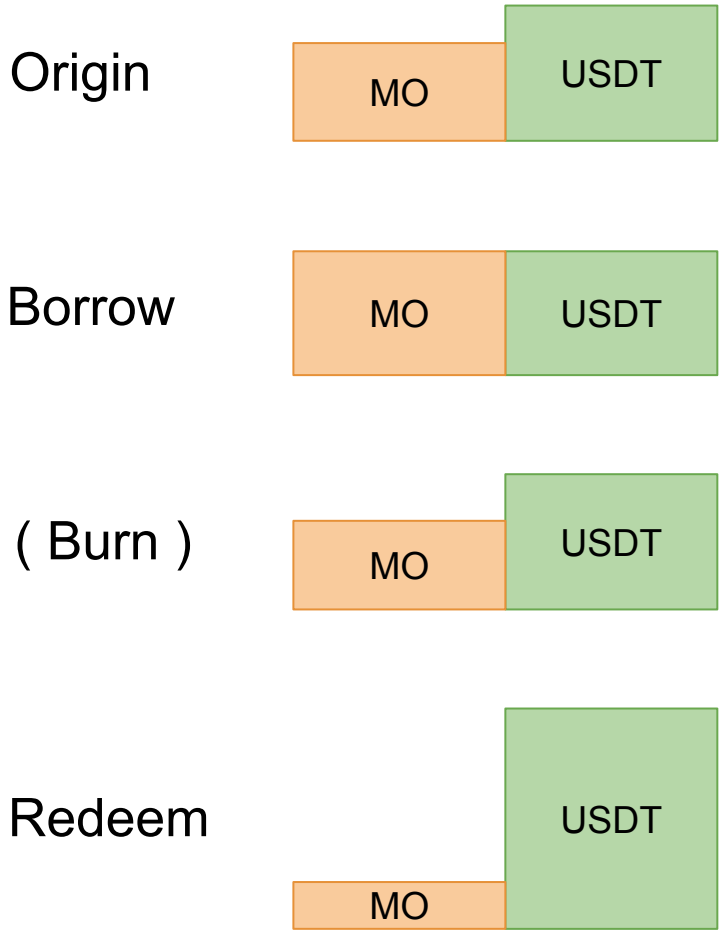
Optimism Daily: https://twitter.com/OptimismDailyTK/status/1721389261373788235

2024.03 MO Token (losses: ~$750k)

- A defective borrow() function.
- Repeated borrow and redeem actions causing price imbalance.

Vulnerable Contract


```
293     function borrow(uint256 amount, uint256 duration) public {
294         if (borrowRates[duration] == 0) revert InvalidDuration();
295         if (amount < borrowMinAmount) revert InvalidAmount();
296
297         if (IToken(borrowToken).whitelist(msg.sender) == false) {
298             IToken(borrowToken).setWhitelist(msg.sender, true);
299         }
300         IApproveProxy(approveProxy).claim(borrowToken, msg.sender, address(this), amount);
301
302         uint256 total = (amount * price() * (BASE - borrowOverCollateral)) / BASE / 1e4;
303         IERC20(supplyToken).safeTransfer(msg.sender, total);
304
305         IUniswapV2Pair(pair).setRouter(address(this));
306         IUniswapV2Pair(pair).claim(borrowToken, BURN, (amount * burnRate) / BASE);
307         IUniswapV2Pair(pair).claim(borrowToken, address(this), (amount * (BASE - burnRate)) / BASE);
308         IUniswapV2Pair(pair).sync();
309         IUniswapV2Pair(pair).setRouter(router);
310
311         address referrer = IRelationship(relationship).referrers(msg.sender);
312         if (IPoolV2(poolV2).getOrder(referrer).running == true) {
313             uint256 referralReward = (amount * inviteRewardRate) / BASE;
314             IERC20(borrowToken).safeTransfer(referrer, referralReward);
315             totalReferralRewardOf[referrer] += referralReward;
316             emit ReferralRewardClaimed(referrer, referralReward);
317         }
```



Mitigation

- Consideration of potential business logic issues.
- Research issues related to similar projects, such as lending, etc.
- Audited by trusted organizations or companies.

Input Validation

Input Validation

- Missing or improper input validation can be exploited to manipulate contract logic, inject malicious data, or cause unexpected behavior.

Input Validation

- address(0), whitelist addresses
- Edge cases in numerical calculations
- Malicious calldata

SocketGateway

- Socket is an interoperability protocol designed to allow blockchains to coordinate with one another.

2024.01 SocketGateway (losses: ~$3.3M)

Generative Future

```
function performAction(
    address fromToken,
    address toToken,
    uint256 amount,
    address receiverAddress,
    bytes32 metadata,
    bytes calldata swapExtraData
) external payable override returns (uint256) {
    uint256 _initialBalanceTokenOut;
    uint256 _finalBalanceTokenOut;

    // Swap Native to Wrapped Token
    if (fromToken == NATIVE_TOKEN_ADDRESS) {
        _initialBalanceTokenOut = ERC20(toToken).balanceOf(socketGateway);
        (bool success, ) = toToken.call{value: amount}(swapExtraData);

        if (!success) {
            revert SwapFailed();
        }

        _finalBalanceTokenOut = ERC20(toToken).balanceOf(socketGateway);

        require(
            (_finalBalanceTokenOut - _initialBalanceTokenOut) == amount,
            "Invalid wrapper contract"
        );

        // Send weth to user
        ERC20(toToken).transfer(receiverAddress, amount);
```

```
    } else {
        _initialBalanceTokenOut = address(socketGateway).balance;

        // Swap Wrapped Token To Native Token
        ERC20(fromToken).safeTransferFrom(
            msg.sender,
            socketGateway,
            amount
        );

        (bool success, ) = fromToken.call(swapExtraData);

        if (!success) {
            revert SwapFailed();
        }

        _finalBalanceTokenOut = address(socketGateway).balance;

        require(
            (_finalBalanceTokenOut - _initialBalanceTokenOut) == amount,
            "Invalid wrapper contract"
        );
```

```
abi.encodeWithSelector(
        IERC20.transferFrom.selector,
        user,
        address(this),
        IERC20(token).balanceOf(user));
```

Audit ReNFT (2024.01)

- ReNFT is a Generalised collateral-free and permissionless rentals built on top of Gnosis Safe and Seaport.
- In Guard.sol, they use _checkTransaction() to prevent transactions that involve transferring an ERC721 or ERC1155 in any way.

Lack of consideration for all scenarios.

Generative Future

```
function _checkTransaction(address from, address to, bytes memory data) private view
    bytes4 selector;

    // Load in the function selector.
    assembly {
        selector := mload(add(data, 0x20))
    }

    if (selector == e721_safe_transfer
        // Load the token ID from call
        uint256 tokenId = uint256(
            _loadValueFromCalldata(dat
        );

        // Check if the selector is al
        _revertSelectorOnActiveRental(
    } else if (selector == e721_safe_t
        // Load the token ID from call
        uint256 tokenId = uint256(
            _loadValueFromCalldata(dat
        );

        // Check if the selector is al
        _revertSelectorOnActiveRental(
    } else if (selector == e721_transf
        // Load the token ID from call
        uint256 tokenId = uint256(
            _loadValueFromCalldata(dat
```

```
        _revertSelectorOnActiveRental(selector, from,
    } else if (selector == e721_approve_selector) {
        // Load the token ID from calldata.
        uint256 tokenId = uint256(
            _loadValueFromCalldata(data, e721_approve
        );

        // Check if the selector is allowed.
        _revertSelectorOnActiveRental(selector, from,
    } else if (selector == e1155_safe_transfer_from_s
        // Load the token ID from calldata.
        uint256 tokenId = uint256(
            _loadValueFromCalldata(data, e1155_safe_t
        );

        // Check if the selector is allowed.
        _revertSelectorOnActiveRental(selector, from,
    } else if (selector == gnosis_safe_enable_module_
        // Load the extension address from calldata.
        address extension = address(
            uint160(
                uint256(
                    _loadValueFromCalldata(data, gnos
                )
            )
        );

        // Check if the extension is whitelisted.
        _revertNonWhitelistedExtension(extension);
```

```
        _revertNonWhitelistedExtension(extension);
    } else if (selector == gnosis_safe_disable_module_selector) {
        // Load the extension address from calldata.
        address extension = address(
            uint160(
                uint256(
                    _loadValueFromCalldata(data, gnosis_safe_disable_module_offset)
                )
            )
        );

        // Check if the extension is whitelisted.
        _revertNonWhitelistedExtension(extension);
    } else {
        // Revert if the `setApprovalForAll` selector is specified. This selector i
        // shared between ERC721 and ERC1155 tokens.
        if (selector == shared_set_approval_for_all_selector) {
            revert Errors.GuardPolicy_UnauthorizedSelector(
                shared_set_approval_for_all_selector
            );
        }

        // Revert if the `safeBatchTransferFrom` selector is specified. There's no
        // cheap way to check if individual items in the batch are rented out.
        // Each token ID would require a call to the storage contract to check
        // its rental status.
        if (selector == e1155_safe_batch_transfer_from_selector) {
            revert Errors.GuardPolicy_UnauthorizedSelector(
                e1155_safe_batch_transfer_from_selector
            );
```

What's missing?

- ERC721 safeTransferFrom
- ERC721 transferFrom
- ERC721 approve
- ERC1155 safeTransferFrom
- ERC1155 safeBatchTransferFrom
- setApprovalForAll
- gnosisSafe disableModule
- gnosisSafe enableModule
- gnosisSafe setGuard

NFTs can be burned by the renter

- ✗ ERC721Burnable burn
- ✗ ERC1155Burnable burn
- ✗ ERC1155Burnable burnBatch

Mitigation

- Consider various usage scenarios.
- Consider the environments in which the project will be deployed or applied.

Price Manipulation

Price Manipulation

- Price Manipulation may lead to incorrect pricing for swaps, improper reward calculation, the ability to borrow more assets than a collateralization ratio allows, or other general issues with financial transactions.

WOOFi

- WOOFi includes an sPMM, which is designed to imitate how orderbooks look (price, spread, depth, etc.) on a centralized exchange. This sPMM has the ability to change the price provided by an oracle to protect against slippage and help to balance the pool.

2024.03 WOOFi (losses: ~$8M)

Attacker borrowed ~10.6 M USDC using a Uniswap flashloan.

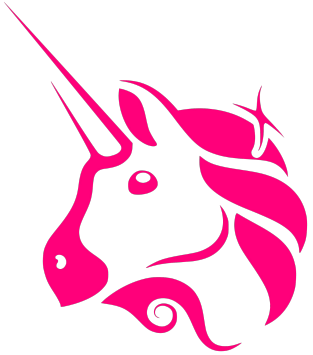
Exchanged 7 M USDC for 7.7 M WOO tokens.

Dumping 7.7 M WOO tokens.

Reacquire 10 M WOO tokens at a low cost.

Repay the flashloan.

About 2.3M in profits



Mitigation

- The token price calculation considers edge cases.
- Process for retrieving prices from oracles.

Takeaway

How can we develop projects more securely?

- Testing
- Auditing
- Continuous monitoring
- Incident Response

DEFIHACKLABS

Let's make web3 more secure!

Thank You !

@YorkChang01

@AliceHsu_kou

DefiHackLabs Line

DefiHackLabs Website