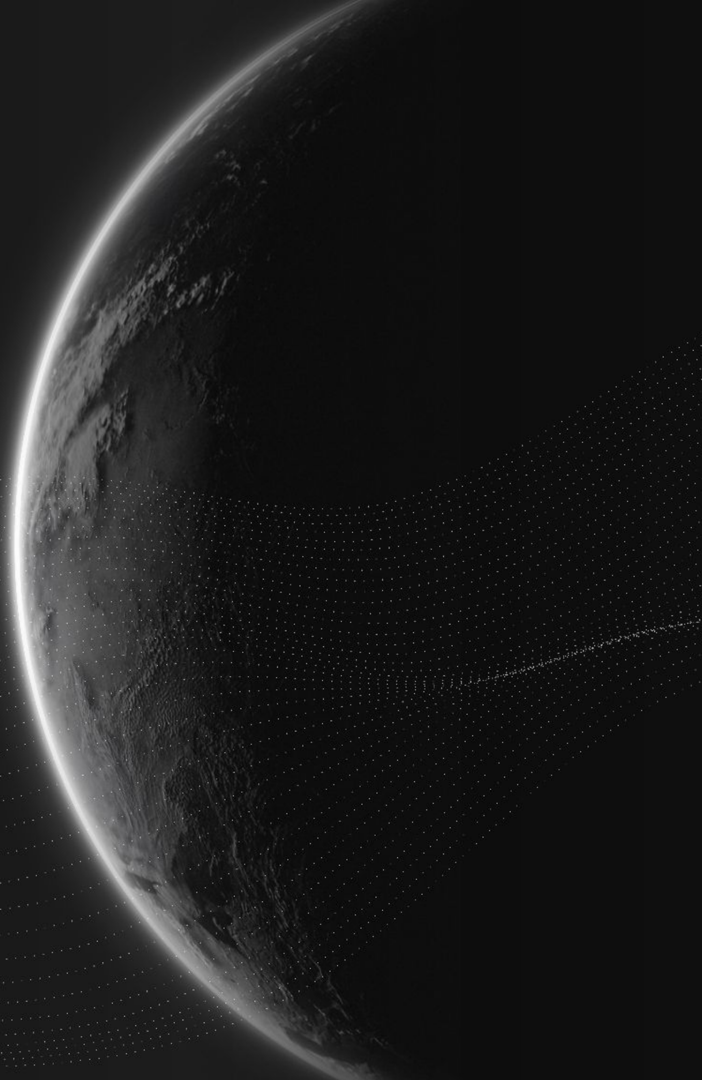
CERTIK

DeFi Security Risks and Defense

Peiyu Wang



Whoami

- **Peiyu Wang (@wisp_fly)**
- **Security Engineer at CertiK**
    - Web3 application penetration testing
        - Crypto exchange, Dapp, Wallet
    - Smart contract audit
    - Security research
    - Incident response
- **DeFi apes**
    - First learned about blockchain and DeFi when I joined CertiK five years ago.
    - Frequently use different DeFi protocols.

CERTIK

Agenda

- **DeFi Security Risks and Defenses in 7 Categories**
  - **Definition**
  - **Case Study**
  - **Best Practice and Defense**
    - DeFi user
    - DeFi project, developer
    - Security firm
- **Q & A**

1. Phishing
2. Rugpull/Exit Scam
3. MEV
4. Operational risk
5. Cloud and Infrastructure
6. Web2 attacks
7. Web3 contract exploit

Three important roles in DeFi

- **DeFi user**
  - A regular user who actively participates in DeFi with limited technical knowledge.
  - Needs to understand how to protect themselves in this "dark forest."
- **DeFi project, developer**
  - A team that develops, operates, and maintains a DeFi protocol, ensuring safe and sustainable long-term operation of the project.
  - Do their best to protect the project from attackers.
  - Responsible for providing sufficient guidance and protection to users to prevent misuse and accidental loss.
- **Security firm**
  - A group of white-hats or a company that provides security consultancy services.
  - Educates DeFi users to help them improve their self-protection skills.
  - Assists DeFi project teams in securing their projects and defending against hackers through services such as auditing.
  - Make the Web3 space safer for everyone

Phishing

CERTIK

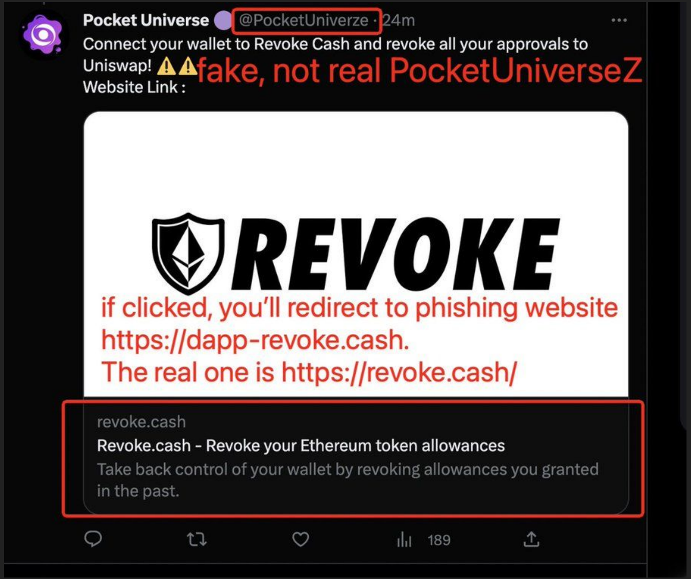
Web3 Phishing

- **Types of malicious activity that trick victims into performing actions that lead to token loss or wallet compromise.**
  - Direct token transfer or "approval"
  - Signature that can be used to obtain token transfer permission (e.g., "Permit")
  - Stealing of private key and seed phrase
- **Commonly used channel to execute phishing attacks**
  - Twitter Phishing Link
  - Phishing Website
  - Discord Scam Bots
  - Telegram Impersonation
  - Email Phishing Campaigns

CERTIK

Case study: Twitter Phishing Link

- It's common practice to remove token approval from an exploited smart contract.
- The exploited project often publishes a tweet warning users to revoke permissions.
- Attackers take advantage of the "post-incident" timing to deliver phishing attacks.



Pocket Universe @PocketUniverze · 24m

Connect your wallet to Revoke Cash and revoke all your approvals to Uniswap! ⚠⚠ fake, not real PocketUniverseZ

Website Link :

REVOKE

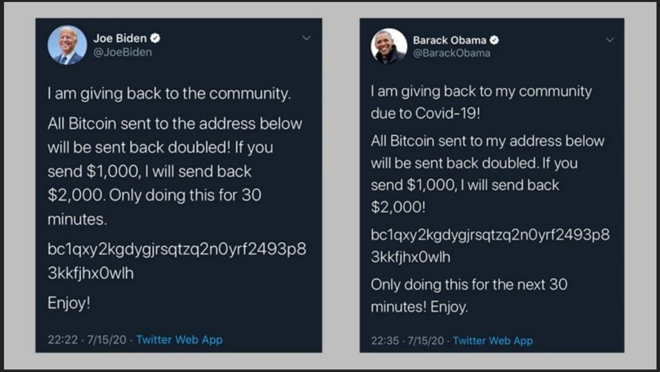
if clicked, you'll redirect to phishing website https://dapp-revoke.cash.
The real one is https://revoke.cash/

revoke.cash
Revoke.cash - Revoke your Ethereum token allowances
Take back control of your wallet by revoking allowances you granted in the past.

189

Case study: Twitter Phishing Link

- On July 15, 2020, a significant security breach occurred on Twitter, where hackers compromised 130 high-profile accounts,
- The attackers used these accounts to post fraudulent messages promoting a Bitcoin scam
- Enticing followers to send cryptocurrency to a specified address with the false promise of doubling their money.



Joe Biden @JoeBiden

I am giving back to the community.

All Bitcoin sent to the address below will be sent back doubled! If you send $1,000, I will send back $2,000. Only doing this for 30 minutes.

bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh

Enjoy!

22:22 · 7/15/20 · Twitter Web App

Barack Obama @BarackObama

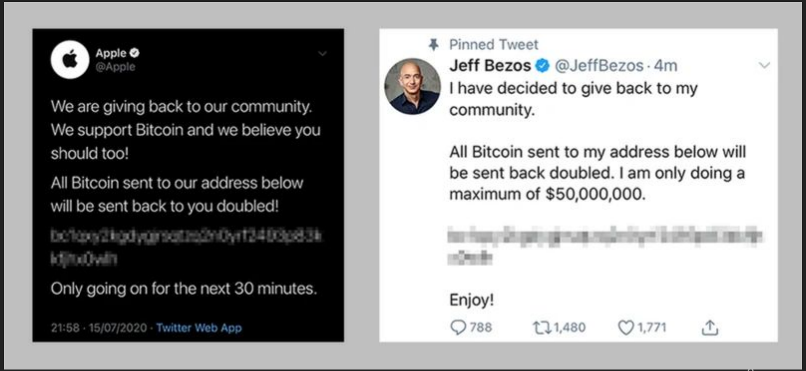
I am giving back to my community due to Covid-19!

All Bitcoin sent to my address below will be sent back doubled. If you send $1,000, I will send back $2,000!

bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh

Only doing this for the next 30 minutes! Enjoy.

22:35 · 7/15/20 · Twitter Web App



Apple @Apple

We are giving back to our community. We support Bitcoin and we believe you should too!

All Bitcoin sent to our address below will be sent back to you doubled!

bc1qxy2kgdygjrsqtzq2n0yrf2493p83kkfjhx0wlh

Only going on for the next 30 minutes.

21:58 · 15/07/2020 · Twitter Web App

Pinned Tweet

Jeff Bezos @JeffBezos · 4m

I have decided to give back to my community.

All Bitcoin sent to my address below will be sent back doubled. I am only doing a maximum of $50,000,000.

[illegible]

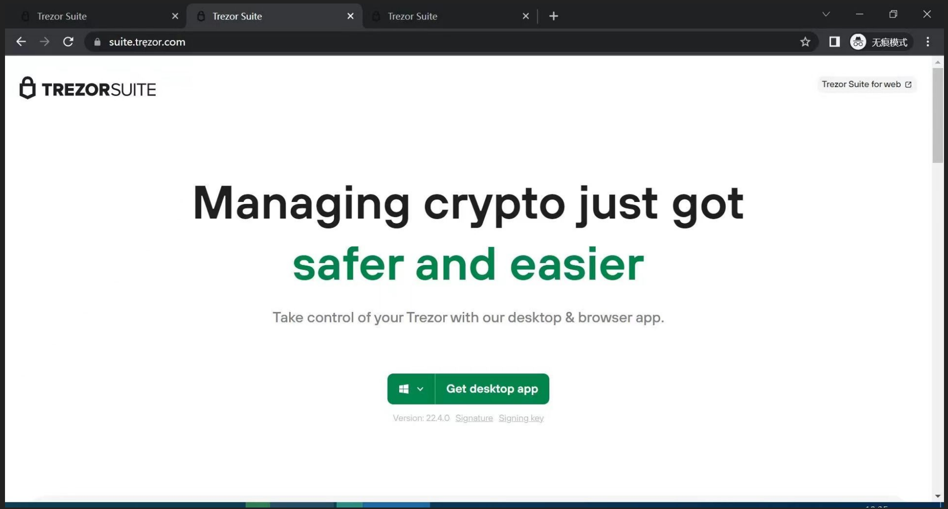
Enjoy!

788 1,480 1,771

CERTIK

Case study: Phishing Website

- In April 2022, Trezor hardware wallet users received phishing emails from the unofficial domain "trezor.us" instead of the legitimate "trezor.io."
- The phishing emails directed users to a deceptive website with a URL like "https://suite.trẹzor.com," using a Unicode character ("ẹ") to mimic the legitimate site.
- Users who entered their seed phrases on the fake site had their wallets compromised, resulting in stolen assets.

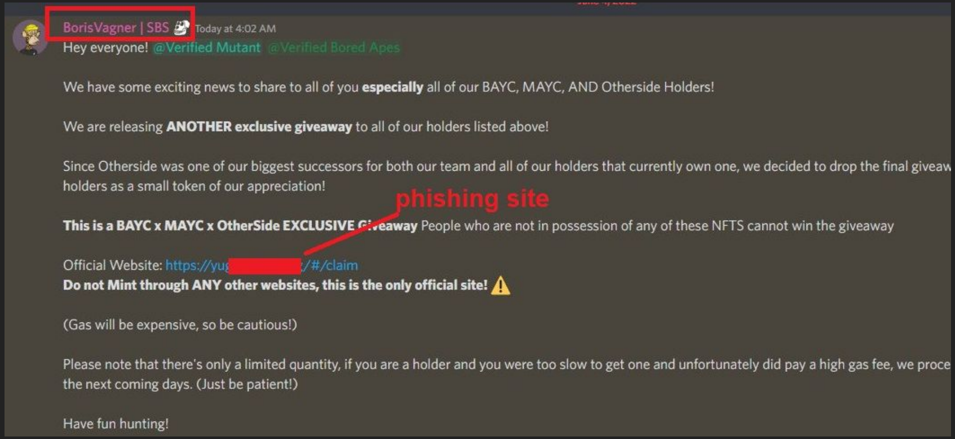


Managing crypto just got safer and easier

Take control of your Trezor with our desktop & browser app.

Case study: Discord Scam Bots

- In June 2022, the Discord servers of popular NFT projects like Bored Ape Yacht Club (BAYC) were compromised.
- Attackers used bots to post phishing links promoting fake exclusive giveaways.
- Users who clicked these links and connected their wallets had their NFTs and tokens stolen.



BorisVagner | SBS Today at 4:02 AM
Hey everyone! @Verified Mutant @Verified Bored Apes

We have some exciting news to share to all of you **especially** all of our BAYC, MAYC, AND Otherside Holders!

We are releasing **ANOTHER exclusive giveaway** to all of our holders listed above!

Since Otherside was one of our biggest successors for both our team and all of our holders that currently own one, we decided to drop the final giveaw
holders as a small token of our appreciation!

This is a BAYC x MAYC x OtherSide EXCLUSIVE Giveaway People who are not in possession of any of these NFTS cannot win the giveaway

Official Website: https://yu[redacted]/#/claim
Do not Mint through ANY other websites, this is the only official site! ⚠

(Gas will be expensive, so be cautious!)

Please note that there's only a limited quantity, if you are a holder and you were too slow to get one and unfortunately did pay a high gas fee, we proce
the next coming days. (Just be patient!)

Have fun hunting!

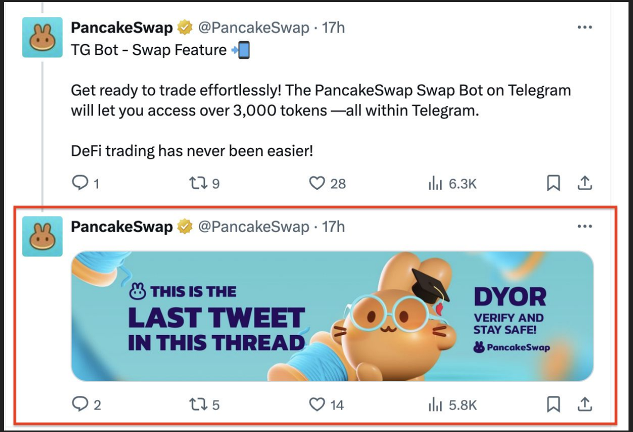
CERTIK

Best Practice and Defense

- **DeFi user**
  - Do not share your private key or seed phrase with anyone.
  - Do not share your private key or seed phrase on any website.
  - Do not click on links from unknown sources.
  - Do not download and execute files from unknown sources.
  - Carefully review transaction data before signing with your wallet.
  - Download applications only from trusted sources, such as the iOS Store or Google Play Store.
  - Be cautious of anything that seems too good to be true.
  - Learn about new phishing attack vectors and avoid them.

Best Practice and Defense

- **DeFi project, developer**
  - Items from the "DeFi" user section
  - Never initiate messages to any user on any communication platform (e.g., Telegram).
  - Routinely identify and remove malicious users from the community channel.
  - Use "The end of thread" on Twitter.
  - Warn users if any project channel (e.g., Twitter, Discord admin account) is compromised.
  - Frequently remind users not to fall victim to phishing attacks.


PancakeSwap @PancakeSwap · 17h
TG Bot - Swap Feature

Get ready to trade effortlessly! The PancakeSwap Swap Bot on Telegram will let you access over 3,000 tokens —all within Telegram.

DeFi trading has never been easier!

1 9 28 6.3K

PancakeSwap @PancakeSwap · 17h

2 5 14 5.8K

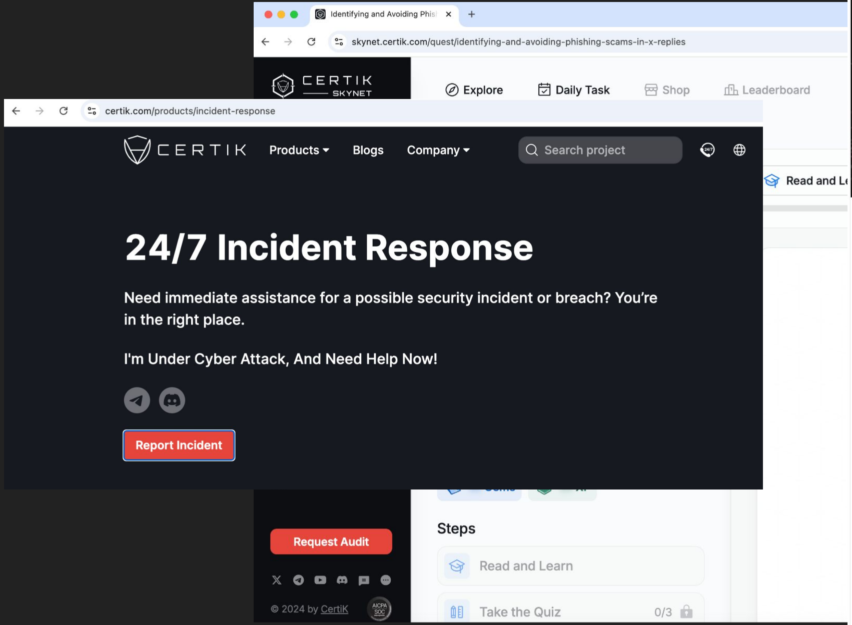
CERTIK

Best Practice and Defense

- **Security firm**
  - Monitor threats in the space and provide public real-time alerts.
  - Establish a phishing reporting channel.
  - Build tools for phishing detection or malicious transaction detection.
  - Publish educational materials for users to learn about recent phishing techniques.
  - Assist users and DeFi projects in tracking stolen funds.

CERTIK

Best Practice and Defense


Identifying and Avoiding Phis

skynet.certik.com/quest/identifying-and-avoiding-phishing-scams-in-x-replies

CERTIK SKYNET

Explore Daily Task Shop Leaderboard

certik.com/products/incident-response

CERTIK Products Blogs Company Search project

24/7 Incident Response

Need immediate assistance for a possible security incident or breach? You're in the right place.

I'm Under Cyber Attack, And Need Help Now!

Report Incident

Read and L

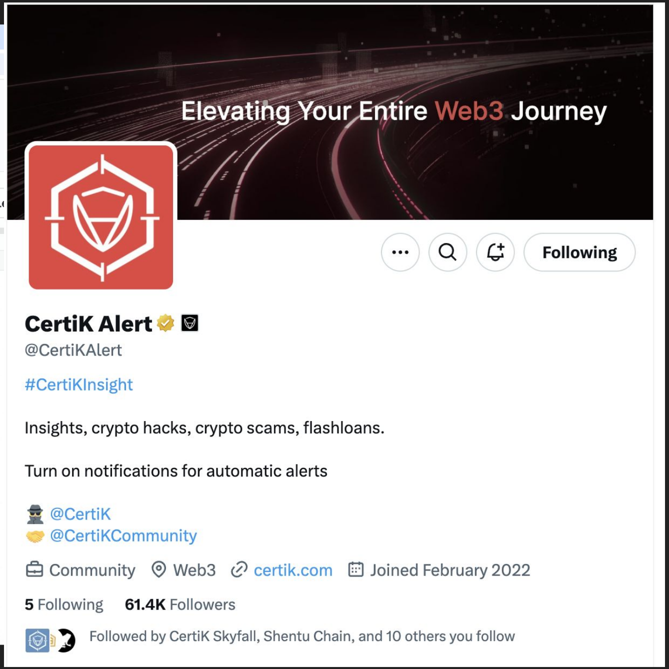
Request Audit

© 2024 by CertiK

Steps

Read and Learn

Take the Quiz 0/3



Following

CertiK Alert

@CertiKAlert

#CertiKInsight

Insights, crypto hacks, crypto scams, flashloans.

Turn on notifications for automatic alerts

🕵 @CertiK
🤝 @CertiKCommunity

Community Web3 certik.com Joined February 2022

5 Following 61.4K Followers

Followed by CertiK Skyfall, Shentu Chain, and 10 others you follow

Rugpull/Exit Scam

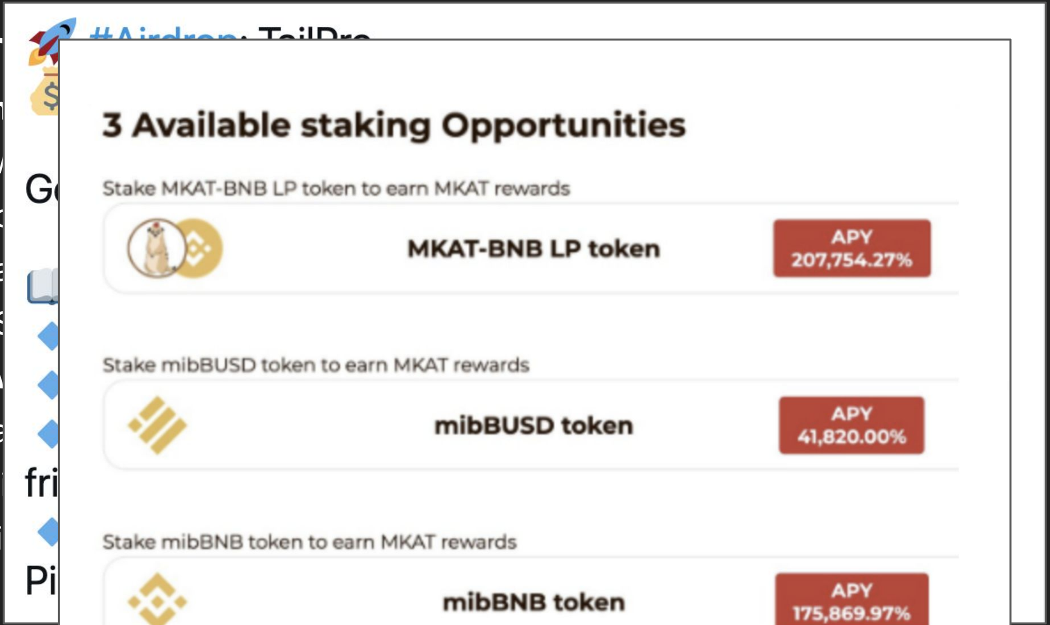
CERTIK

Rugpull and Exit Scam

- **Project teams abandon the project after draining liquidity, dumping tokens, or even stealing deposited user assets, leaving investors with worthless holdings.**
- **Hard Rug**
  - Dump all tokens allotted to the team shortly after the project launch and abandon the project.
  - Use privileged accounts to mint a significant number of tokens and dump them on the market.
  - Use a backdoor function to steal all tokens that users deposit into the project's smart contract.
- **Soft Rug**
  - Slowly selling tokens own by the project team.
  - Delivering only very limited promised new protocol features, then abandon the project.
  - Engaging in insider trading, pump and dump activities.

Rugpull and Exit Scam

- How bad pr
  - Social m
    - Tw
    - Bo
    - Pa
    - BS
  - Attractiv
    - Fa
    - In
    - Ai
  - Audit
    - Fake a
    - Audit by disreputable auditing firm

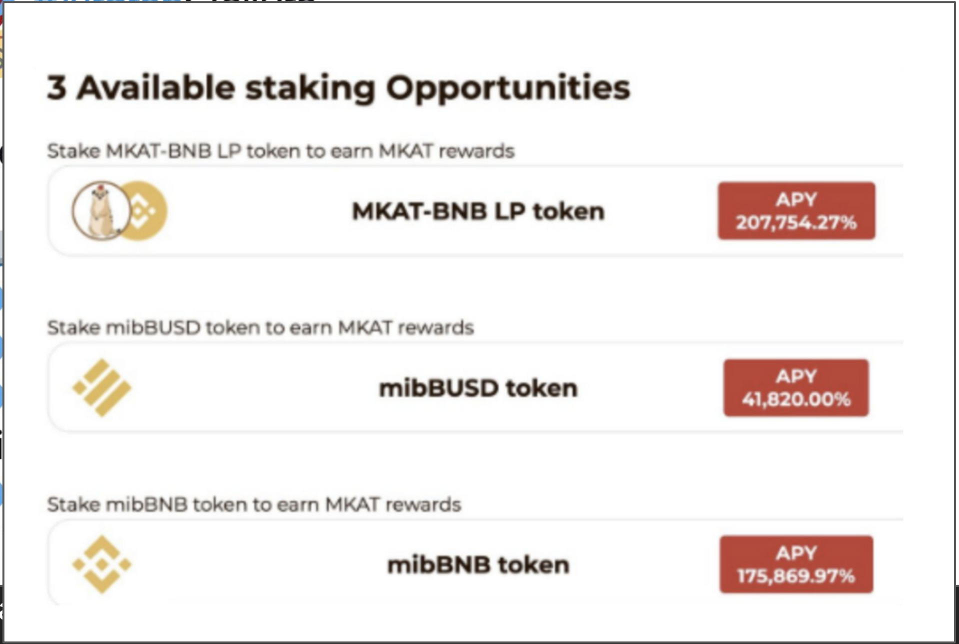


#Airdrop: TailPro

G

fri

Pi



3 Available staking Opportunities

Stake MKAT-BNB LP token to earn MKAT rewards

MKAT-BNB LP token — APY 207,754.27%

Stake mibBUSD token to earn MKAT rewards

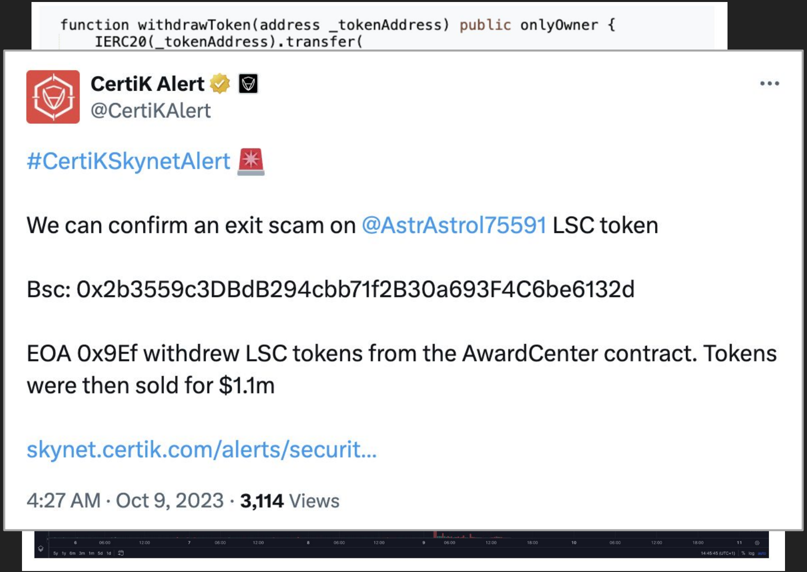
mibBUSD token — APY 41,820.00%

Stake mibBNB token to earn MKAT rewards

mibBNB token — APY 175,869.97%

Case study: Hard Rug

- **Lucky Star Currency Exit Scam**
  - On 9 October 2023, the owner of the "Award Center" and "NFT Merge" contract called withdrawToken.
  - A total of 3,095,977.40 LSC tokens was transferred to the owner.
  - Three million LSC tokens were swapped for 1.1 million USDT, causing an approximate 98% price drop.



```
function withdrawToken(address _tokenAddress) public onlyOwner {
    IERC20(_tokenAddress).transfer(
```

CertiK Alert
@CertiKAlert

#CertiKSkynetAlert 🚨

We can confirm an exit scam on @AstrAstrol75591 LSC token

Bsc: 0x2b3559c3DBdB294cbb71f2B30a693F4C6be6132d

EOA 0x9Ef withdrew LSC tokens from the AwardCenter contract. Tokens were then sold for $1.1m

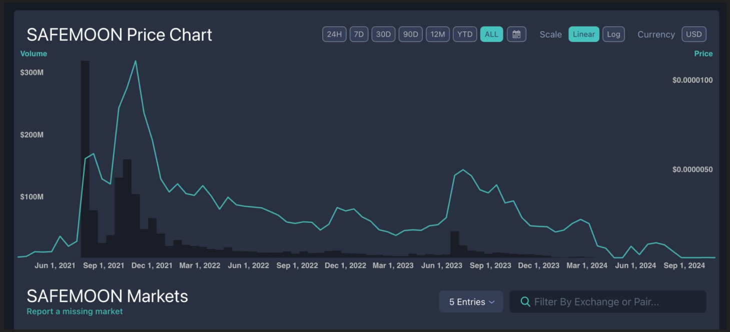
skynet.certik.com/alerts/securit...

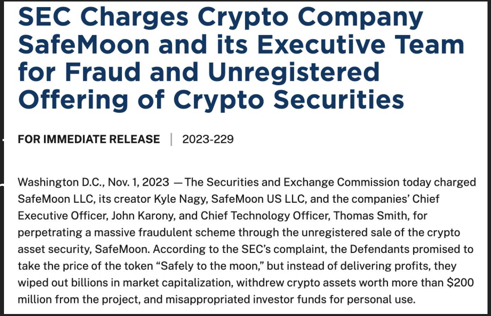
4:27 AM · Oct 9, 2023 · **3,114** Views

CERTIK

Case study: Soft Rug

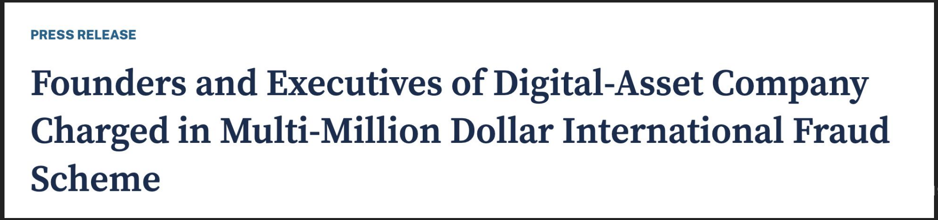
- **Safemoon token**
- **The Good**
  - Shortly after its launch in April 2021, it became the most popular memecoin.
  - The price increased 1,000 times compared to its launch.
  - The Twitter account gained 1.2 million followers.
- **The Bad**
  - The project promised to deliver many new features and products but
  - It launched a V2 token contract that introduced significantly more cen
  - The price dropped 99% compared to its peak.
  - The team dumped tokens and profited from shady activity.
- **The Ugly**
  - DOJ investigation and SEC charges.




SEC Charges Crypto Company SafeMoon and its Executive Team for Fraud and Unregistered Offering of Crypto Securities

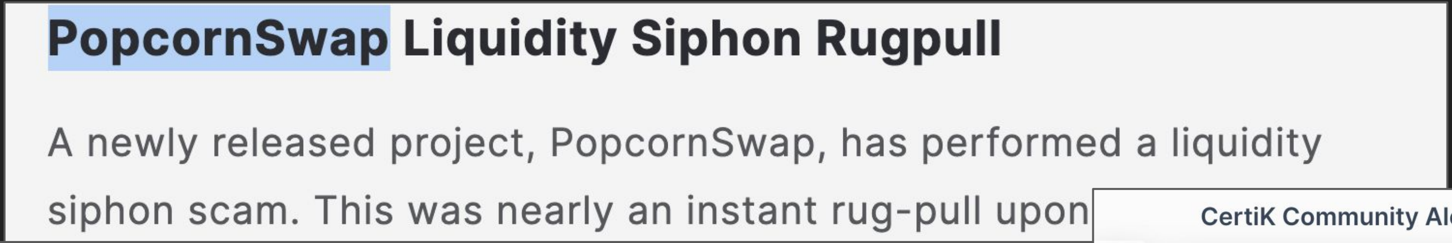
FOR IMMEDIATE RELEASE | 2023-229

Washington D.C., Nov. 1, 2023 —The Securities and Exchange Commission today charged SafeMoon LLC, its creator Kyle Nagy, SafeMoon US LLC, and the companies' Chief Executive Officer, John Karony, and Chief Technology Officer, Thomas Smith, for perpetrating a massive fraudulent scheme through the unregistered sale of the crypto asset security, SafeMoon. According to the SEC's complaint, the Defendants promised to take the price of the token "Safely to the moon," but instead of delivering profits, they wiped out billions in market capitalization, withdrew crypto assets worth more than $200 million from the project, and misappropriated investor funds for personal use.


PRESS RELEASE

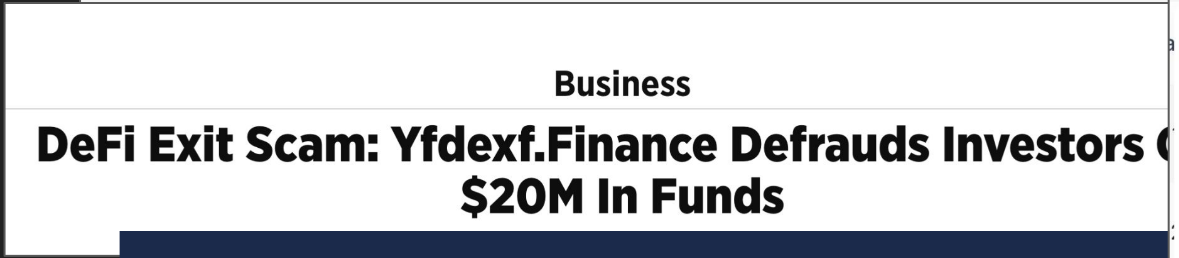
Founders and Executives of Digital-Asset Company Charged in Multi-Million Dollar International Fraud Scheme

Getting hacked is bad, but this is worse…



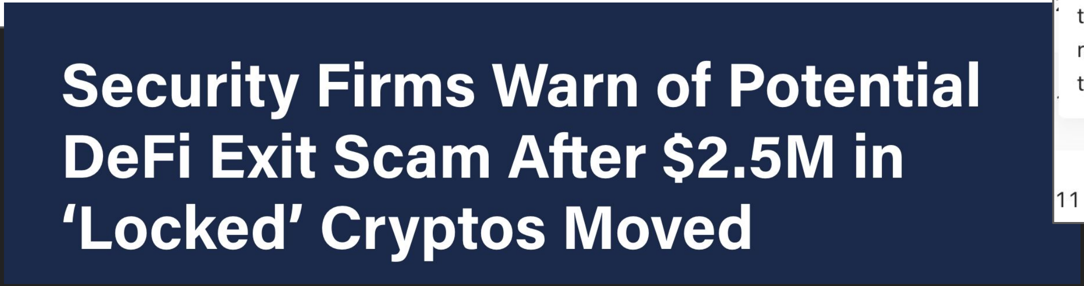
PopcornSwap Liquidity Siphon Rugpull

A newly released project, PopcornSwap, has performed a liquidity siphon scam. This was nearly an instant rug-pull upon

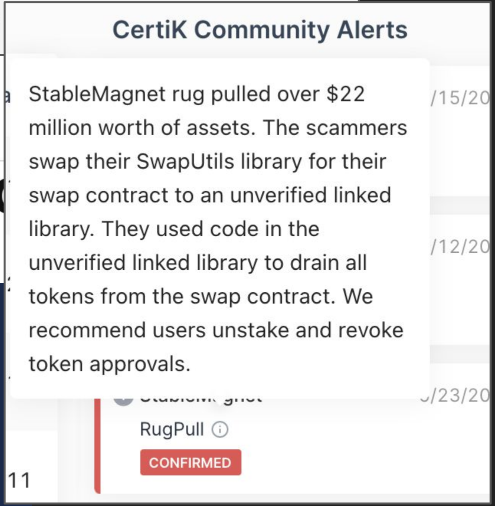


Business

DeFi Exit Scam: Yfdexf.Finance Defrauds Investors
$20M In Funds



Security Firms Warn of Potential DeFi Exit Scam After $2.5M in 'Locked' Cryptos Moved



CertiK Community Alerts

StableMagnet rug pulled over $22 million worth of assets. The scammers swap their SwapUtils library for their swap contract to an unverified linked library. They used code in the unverified linked library to drain all tokens from the swap contract. We recommend users unstake and revoke token approvals.

/15/20

/12/20

/23/20

RugPull

CONFIRMED

11

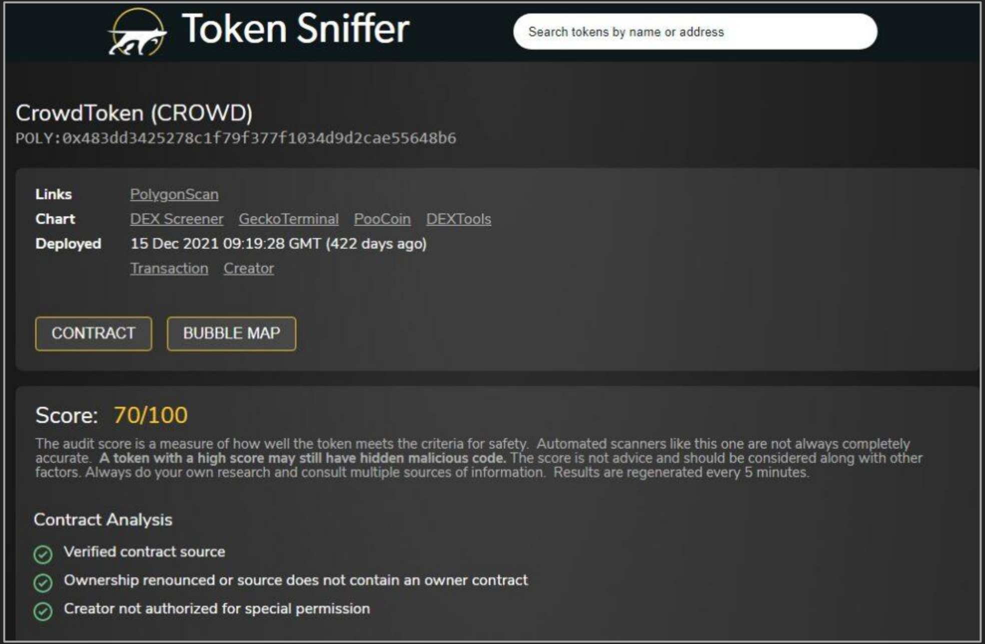
Best Practice and Defense

- **DeFi user**
  - **DYOR (Do Your Own Research) to identify red-flag projects before investing:**
    - Anonymous teams with AI-generated profiles and fake names on the project website.
    - Domains registered through privacy-focused providers like Namecheap.
    - Lack of detailed whitepapers, token distribution plans, or roadmaps.
    - No reliable audit report. Projects without an audit may contain backdoor functions.
    - Unreasonably high APY% returns.
    - Unverified contract.
    - Use tools such as "Token Sniffer."
    - Review audit reports and check for high-risk findings.

CERTIK

Best Practice and Defense

- DeFi user


Token Sniffer

Search tokens by name or address

CrowdToken (CROWD)
POLY:0x483dd3425278c1f79f377f1034d9d2cae55648b6

Links	PolygonScan
Chart	DEX Screener GeckoTerminal PooCoin DEXTools
Deployed	15 Dec 2021 09:19:28 GMT (422 days ago)
	Transaction Creator

CONTRACT BUBBLE MAP

Score: 70/100

The audit score is a measure of how well the token meets the criteria for safety. Automated scanners like this one are not always completely accurate. **A token with a high score may still have hidden malicious code.** The score is not advice and should be considered along with other factors. Always do your own research and consult multiple sources of information. Results are regenerated every 5 minutes.

Contract Analysis

- ✓ Verified contract source
- ✓ Ownership renounced or source does not contain an owner contract
- ✓ Creator not authorized for special permission

Best Practice and Defense

- **DeFi project, developer**
  - Be transparent with users; anonymize team information if necessary.
  - When outsourcing smart contract development
    - Ensure there are no backdoors in the code.
    - Migrate the contract to an owner account controlled solely by the project team, and not share with 3rd party developers.
  - Ensure the privileged account's private key is securely handled.
  - Obtain a smart contract audit from a reputable auditing firm.

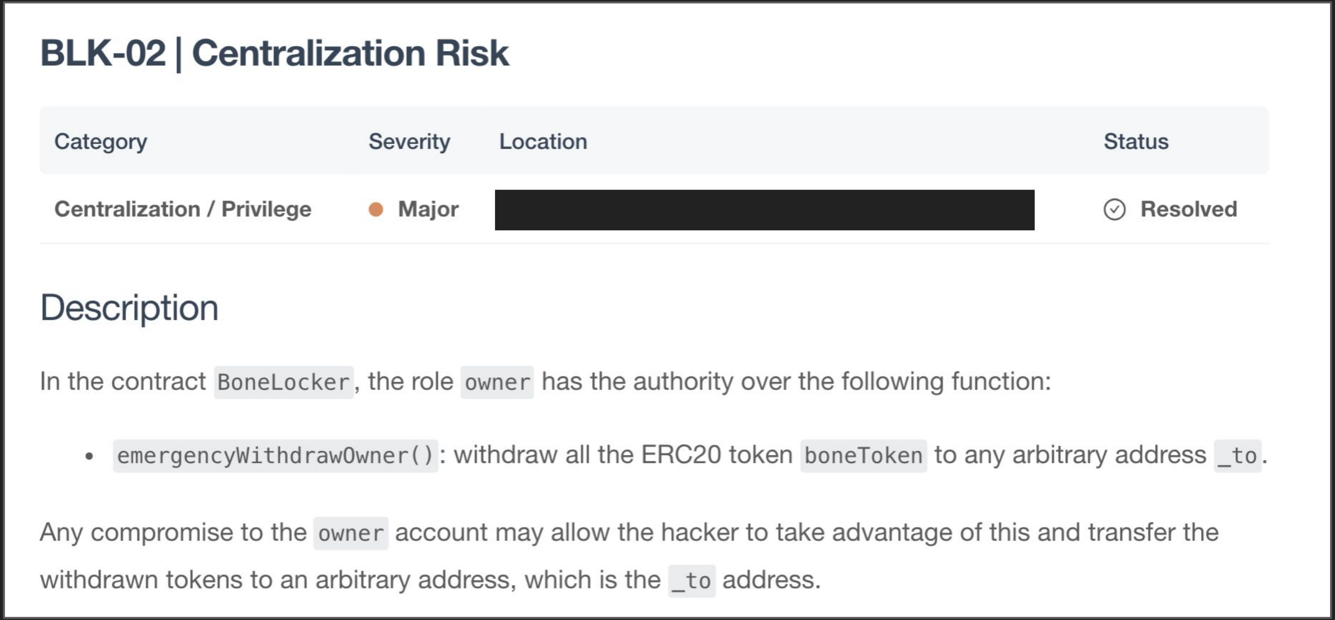
Best Practice and Defense

- **Security firm**
  - Highlight backdoors or risky functions in the smart contract within the audit report.
  - Provide timely alerts to the community to help avoid scam projects.
  - Provide tools to assist users in conducting their research.

CERTIK

Best Practice and Defense

- **Audit report**



BLK-02 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● Major	[redacted]	✓ Resolved

Description

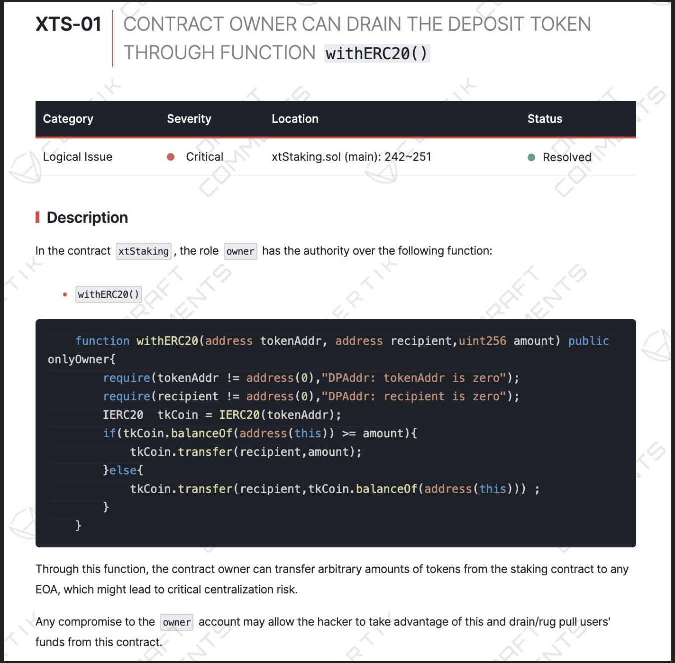
In the contract `BoneLocker`, the role `owner` has the authority over the following function:

- `emergencyWithdrawOwner()`: withdraw all the ERC20 token `boneToken` to any arbitrary address `_to`.

Any compromise to the `owner` account may allow the hacker to take advantage of this and transfer the withdrawn tokens to an arbitrary address, which is the `_to` address.

Best Practice and Defense

- **Audit report**



XTS-01 CONTRACT OWNER CAN DRAIN THE DEPOSIT TOKEN THROUGH FUNCTION `withERC20()`

Category	Severity	Location	Status
Logical Issue	● Critical	xtStaking.sol (main): 242~251	● Resolved

Description

In the contract `xtStaking`, the role `owner` has the authority over the following function:

- `withERC20()`

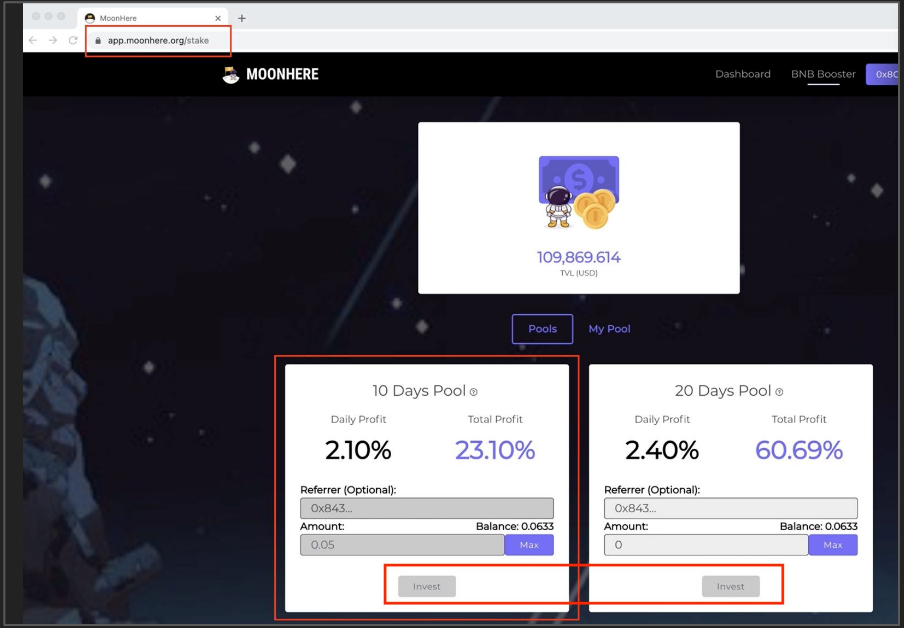
```
    function withERC20(address tokenAddr, address recipient,uint256 amount) public
onlyOwner{
        require(tokenAddr != address(0),"DPAddr: tokenAddr is zero");
        require(recipient != address(0),"DPAddr: recipient is zero");
        IERC20  tkCoin = IERC20(tokenAddr);
        if(tkCoin.balanceOf(address(this)) >= amount){
            tkCoin.transfer(recipient,amount);
        }else{
            tkCoin.transfer(recipient,tkCoin.balanceOf(address(this))) ;
        }
    }
```

Through this function, the contract owner can transfer arbitrary amounts of tokens from the staking contract to any EOA, which might lead to critical centralization risk.

Any compromise to the `owner` account may allow the hacker to take advantage of this and drain/rug pull users' funds from this contract.

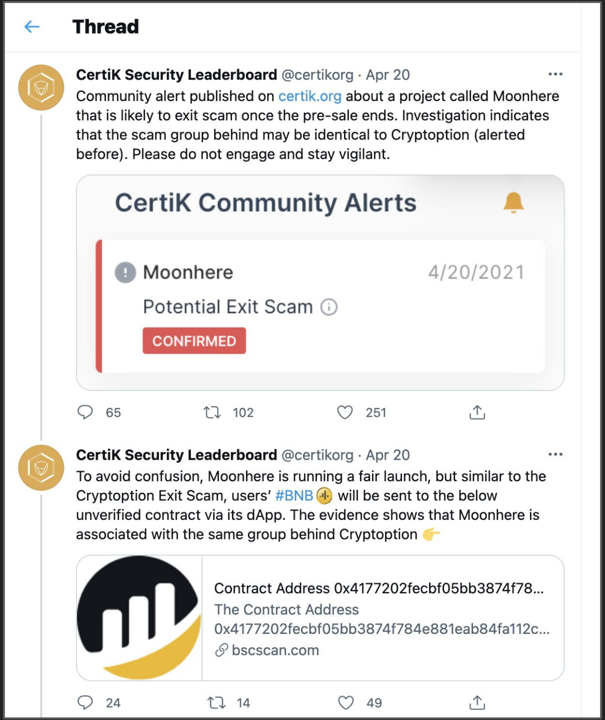
Best Practice and Defense

- A case where our alert helped minimize user losses.



CERTIK

Best Practice and Defense


Thread

CertiK Security Leaderboard @certikorg · Apr 20

Community alert published on certik.org about a project called Moonhere that is likely to exit scam once the pre-sale ends. Investigation indicates that the scam group behind may be identical to Cryptoption (alerted before). Please do not engage and stay vigilant.

CertiK Community Alerts

Moonhere 4/20/2021

Potential Exit Scam

CONFIRMED

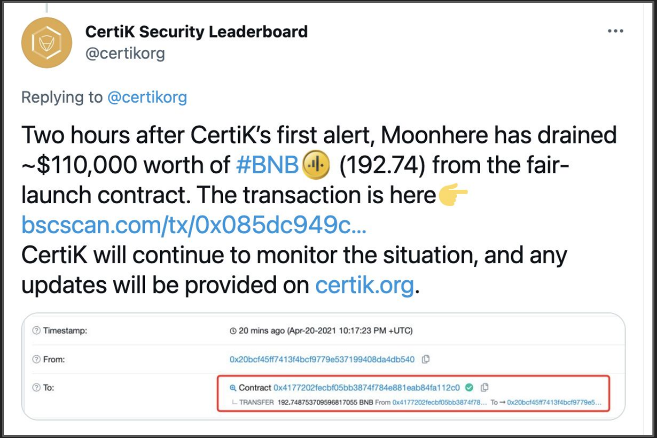
65 102 251

CertiK Security Leaderboard @certikorg · Apr 20

To avoid confusion, Moonhere is running a fair launch, but similar to the Cryptoption Exit Scam, users' #BNB will be sent to the below unverified contract via its dApp. The evidence shows that Moonhere is associated with the same group behind Cryptoption 👉

Contract Address 0x4177202fecbf05bb3874f78...
The Contract Address 0x4177202fecbf05bb3874f784e881eab84fa112c...
bscscan.com

24 14 49

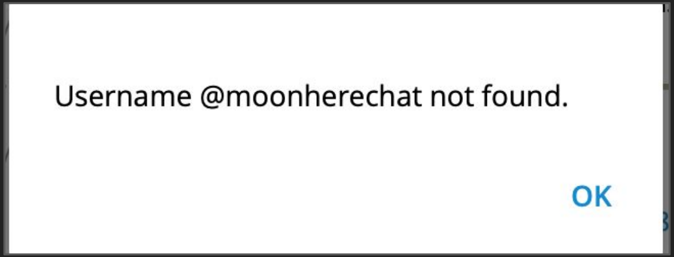

CertiK Security Leaderboard
@certikorg

Replying to @certikorg

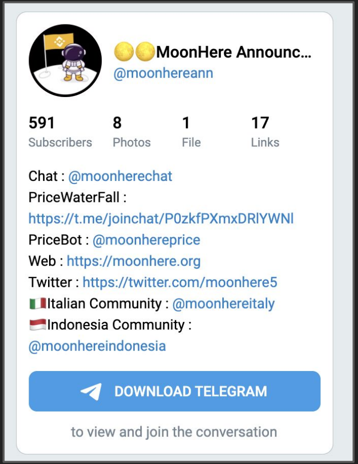
Two hours after CertiK's first alert, Moonhere has drained ~$110,000 worth of #BNB (192.74) from the fair-launch contract. The transaction is here👉 bscscan.com/tx/0x085dc949c...
CertiK will continue to monitor the situation, and any updates will be provided on certik.org.

Timestamp:	20 mins ago (Apr-20-2021 10:17:23 PM +UTC)
From:	0x20bcf45ff7413f4bcf9779e537199408da4db540
To:	Contract 0x4177202fecbf05bb3874f784e881eab84fa112c0 └ TRANSFER 192.748753709596817055 BNB From 0x4177202fecbf05bb3874f78... To → 0x20bcf45ff7413f4bcf9779e5...

Best Practice and Defense


Username @moonherechat not found.

OK

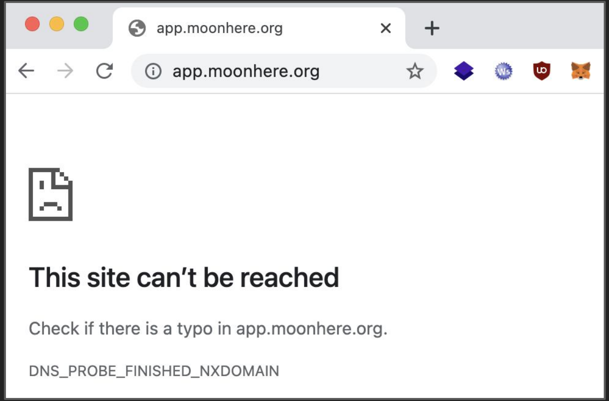

MoonHere Announc...
@moonhereann

591 Subscribers
8 Photos
1 File
17 Links

Chat : @moonherechat
PriceWaterFall :
https://t.me/joinchat/P0zkfPXmxDRlYWNl
PriceBot : @moonhereprice
Web : https://moonhere.org
Twitter : https://twitter.com/moonhere5
🇮🇹Italian Community : @moonhereitaly
🇮🇩Indonesia Community :
@moonhereindonesia

DOWNLOAD TELEGRAM

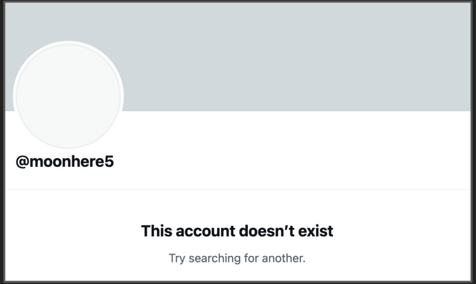
to view and join the conversation


app.moonhere.org

This site can't be reached

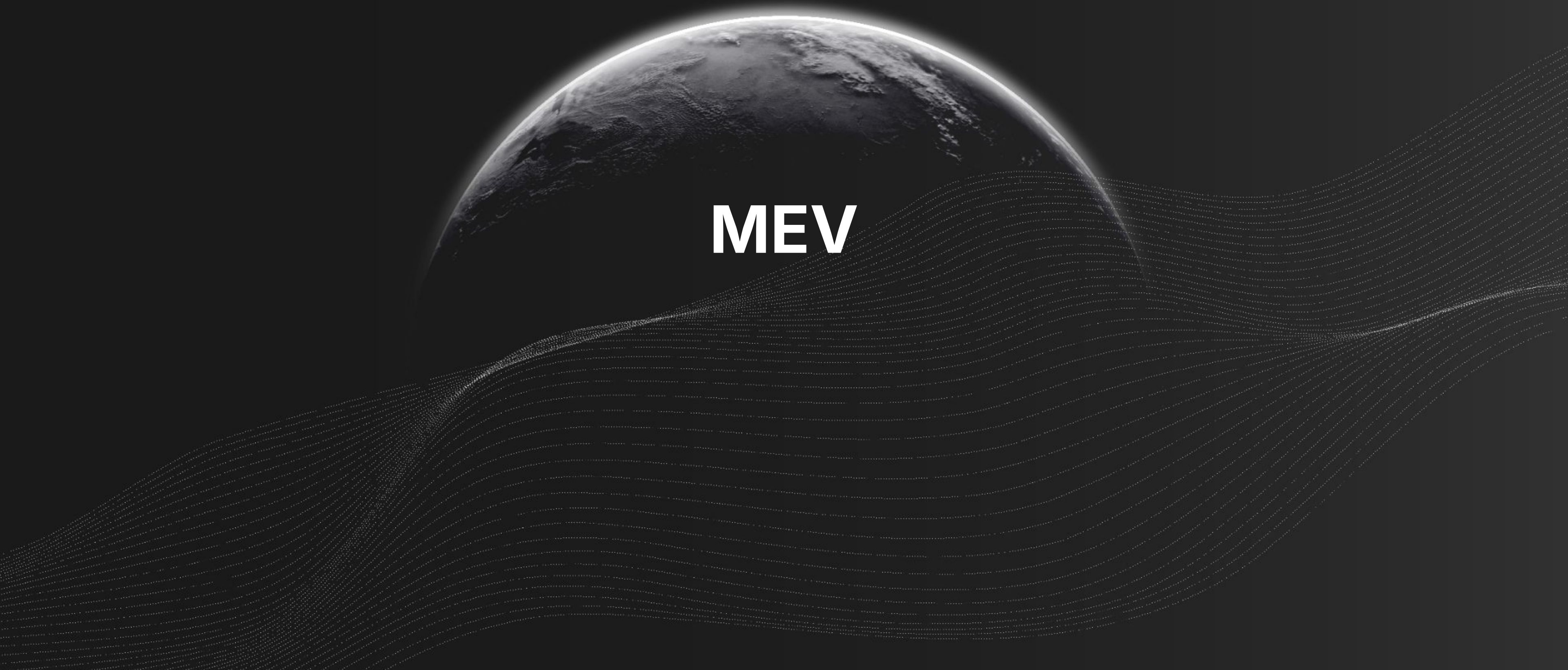
Check if there is a typo in app.moonhere.org.

DNS_PROBE_FINISHED_NXDOMAIN


@moonhere5

This account doesn't exist

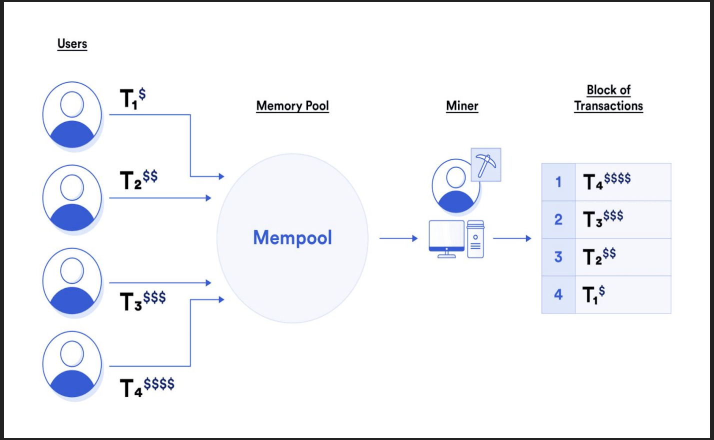
Try searching for another.



MEV

MEV

- **MEV stands for "miner extractable value" or "maximal extractable value"**
- **Reasons that MEV exists**
  - Blockchain transactions in most blockchains are sequential.
  - The order of transactions within the same block matters a lot.
  - Transactions submitted by users are publicly viewable in the "waiting area," called the mempool, where they wait to be picked up by a miner for execution.
  - The miner (block producer) can arbitrarily include, exclude, or reorder transactions as they wish.
  - By default, miners will pick up transactions with higher gas fees first.
- **Common MEV attack vectors**
  - Sandwich attack
  - Front-running
  - Back-running

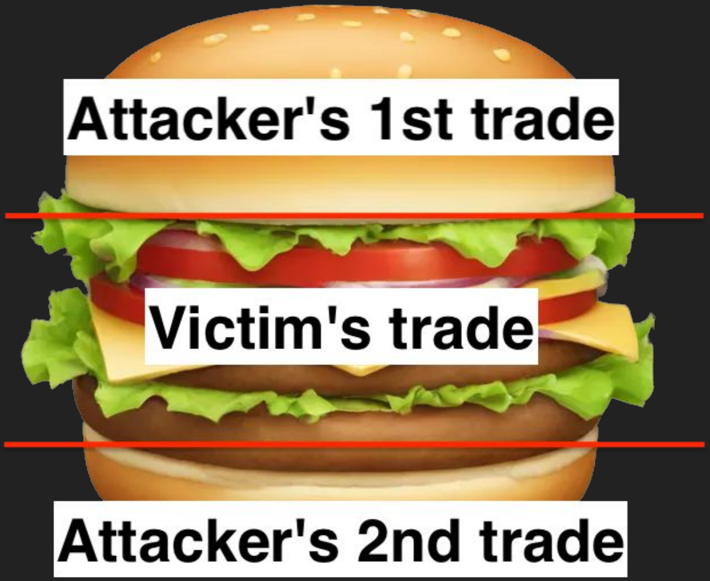


Case Study: Sandwich attack

- Assume there is a BTC <> USDT Pool in the Uniswap AMM DEX (x * y = k)
    - 10 BTC : 1,000,000 USDT
- A user wants to sell 1 BTC into the pool. The user will receive: 1,000,000 - (10 * 1,000,000 / 11) = 90,909 USDT.
    - Pool status: 11 BTC, 909,091 USDT.

CERTIK

Case Study: Sandwich attack



CERTIK

Case Study: Sandwich attack

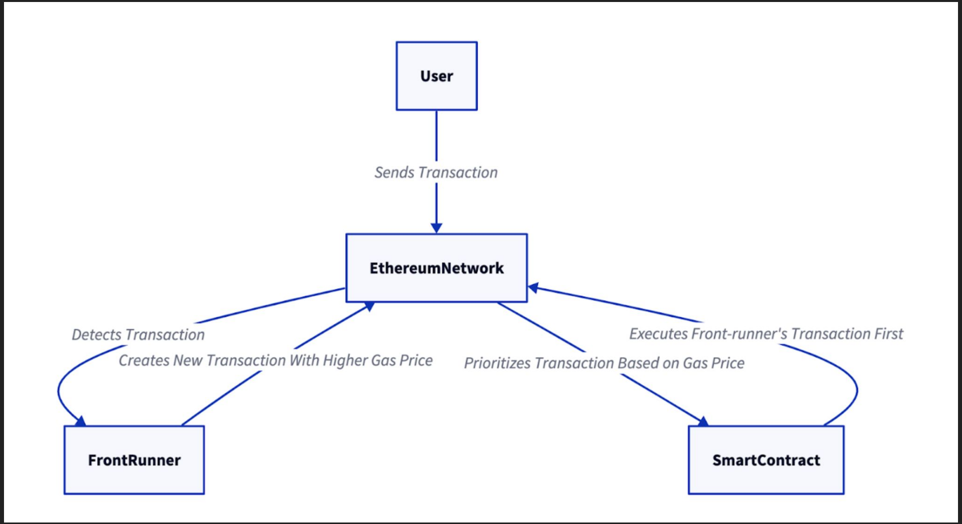
1. An attacker sells 100 BTC into the pool. The attacker will receive 909,091 USDT.
    a. Pool status: [110 BTC : 90,909 USDT]

2. Now, the user trades the 1 BTC in this pool. The user will receive 90,909 - (110 * 90,909 / 111) = **819 USDT**.
    a. Pool status: [111 BTC : 90,090 USDT]

3. The attacker uses all the 909,091 USDT previously obtained to buy Bitcoin. The attacker gains 100.992 BTC.
    a. Pool status: [10.008 BTC : 999,181 USDT]
    b. Profit: 100.992 - 100 = 0.992 BTC.

Case Study: Front-running

- If an attacker notices a profitable pending transaction in the mempool, they may be able to craft a transaction that performs the same action but pays a higher gas fee to front-run the victim's transaction.

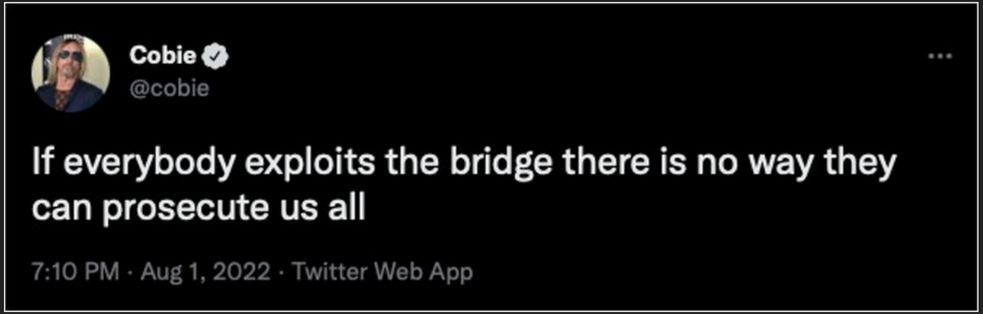

User
Sends Transaction
EthereumNetwork
Detects Transaction
Creates New Transaction With Higher Gas Price
Prioritizes Transaction Based on Gas Price
Executes Front-runner's Transaction First
FrontRunner
SmartContract

Case Study: Front-running

- **Nomad Bridge Exploit**
  - In August 2022, the Nomad Bridge had a vulnerability that allowed attackers to bypass the message verification process.
  - Enabling anyone to make unauthorized withdrawals from the bridge contract.
  - Total lost: $190 million
- **A "not so smart attacker"**
  - The attacker crafted an exploit that only drained ~1% of the tokens from the bridge contract.
  - The exploit transaction was sent to the public mempool.
  - Front-running bots were able to identify this profitable trade and started sending out identical attack transactions.

Case Study: Front-running

- **500+"Copycats" exploiters**
  - 14 addresses each stole over $2,000,000 USD
  - 22 addresses each stole between $1,000,000 and $2,000,000 USD
  - 281 addresses each stole between $100,000 and $1,000,000 USD
  - 215 addresses each stole between $1,000 and $100,000 USD

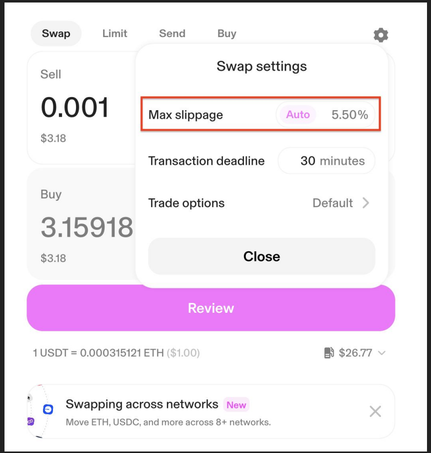

Cobie
@cobie

If everybody exploits the bridge there is no way they can prosecute us all

7:10 PM · Aug 1, 2022 · Twitter Web App

Best Practice and Defense

- **DeFi user**
  - Set an appropriate slippage rate when trading on a DEX or other DeFi platform.
    - If the trade result is less than a preset number, the trade will be canceled in the smart contract.
  - Split a large trade into smaller ones.
  - Trade with aggregators such as 1inch; it splits orders across multiple pools and routes them through the most efficient path.
  - Use a private mempool service for important transactions that might be front-run.



Sell
0.001
$3.18

Buy
3.15918
$3.18

Swap settings

Max slippage Auto 5.50%

Transaction deadline 30 minutes

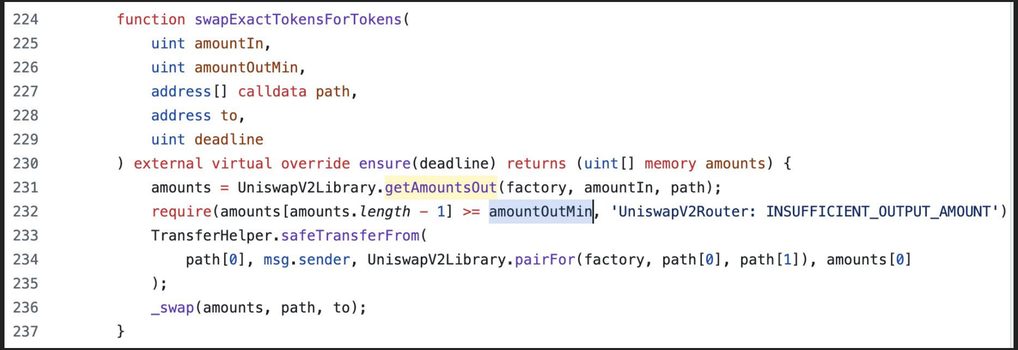
Trade options Default

1 USDT = 0.000315121 ETH ($1.00) $26.77

Swapping across networks New
Move ETH, USDC, and more across 8+ networks.

Best Practice and Defense

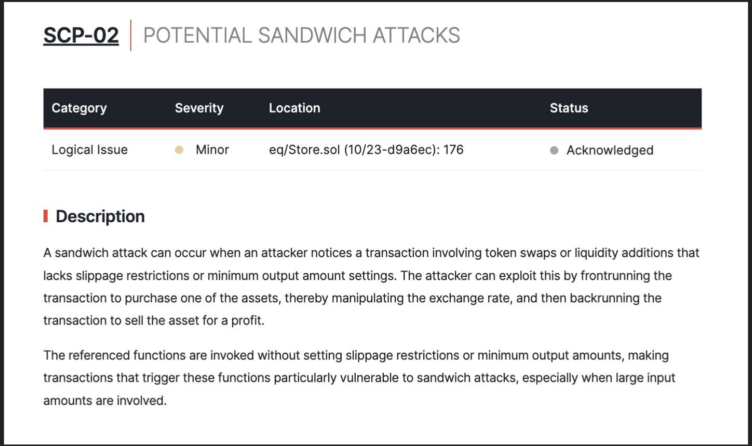
- **DeFi project, developer**
    - Implemented the "slippage" feature in the DEX's smart contract.
        - Warn users to set an appropriate slippage in the Dapp front end.
    - Avoid trading large amounts within the smart contract.
        - It's difficult to prevent sandwich attacks within the smart contract.
    - If a DeFi feature requires multiple transactions, implement it in a way that it cannot be attacked by front-running or back-running.



```
    function swapExactTokensForTokens(
        uint amountIn,
        uint amountOutMin,
        address[] calldata path,
        address to,
        uint deadline
    ) external virtual override ensure(deadline) returns (uint[] memory amounts) {
        amounts = UniswapV2Library.getAmountsOut(factory, amountIn, path);
        require(amounts[amounts.length - 1] >= amountOutMin, 'UniswapV2Router: INSUFFICIENT_OUTPUT_AMOUNT')
        TransferHelper.safeTransferFrom(
            path[0], msg.sender, UniswapV2Library.pairFor(factory, path[0], path[1]), amounts[0]
        );
        _swap(amounts, path, to);
    }
```

Best Practice and Defense

- **Security firm**
  - Look for code vulnerabilities that can be exploited to harm either the user or the DeFi project during smart contract audits.
  - Use a private mempool service for important transactions that might be front-run.
    - Whitehat rescue transaction



SCP-02 | POTENTIAL SANDWICH ATTACKS

Category	Severity	Location	Status
Logical Issue	● Minor	eq/Store.sol (10/23-d9a6ec): 176	● Acknowledged

Description

A sandwich attack can occur when an attacker notices a transaction involving token swaps or liquidity additions that lacks slippage restrictions or minimum output amount settings. The attacker can exploit this by frontrunning the transaction to purchase one of the assets, thereby manipulating the exchange rate, and then backrunning the transaction to sell the asset for a profit.

The referenced functions are invoked without setting slippage restrictions or minimum output amounts, making transactions that trigger these functions particularly vulnerable to sandwich attacks, especially when large input amounts are involved.

Operational Risk

Operational Risk

- **Operational risk is the potential for loss or disruption resulting from failures in a company's internal processes, people, systems, or external events.**
    - Private Key Compromise
    - Compromised Device
    - Human Error in Transaction Handling
    - Unauthorized Access to Internal Resources
    - Social Media Takeover
    - Insider Threats

CERTIK

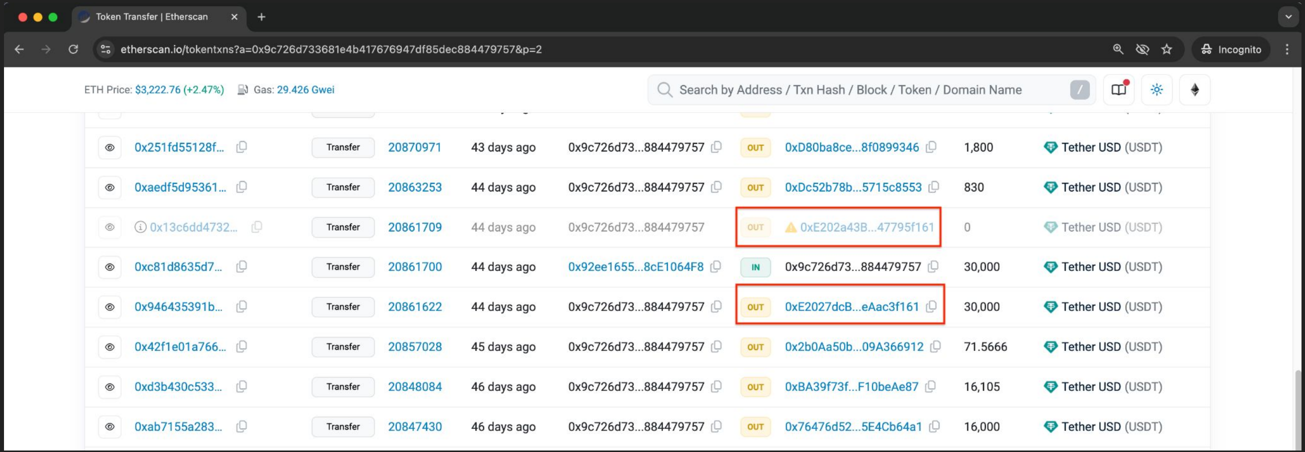
Case study: Compromised Device

- On October 16, 2024, Radiant Capital experienced a security breach resulting in the loss of approximately $50 million USD.
- The attack compromised the devices of three Radiant developers via malware.
- On the compromised devices, the front-end of the Multi-sig Wallet displayed legitimate transaction data while malicious transactions were sent to the developers' **hardware wallets** for signing.
- The attacker ultimately gained three signatures from the compromised developers/devices, allowing them to call the "transferOwnership" function in the smart contract and take control of the contract to withdraw the funds.

CERTIK

Case study: Human Error in Transaction Handling

- **Transaction history poisoning**
  a. DeFi users use blockchain explorers, such as Etherscan, to view transaction history.
  b. Due to the length of a crypto address, explorers show only the first and last few characters of an address.
  c. Users often copy addresses from the transaction log to retrieve recently interacted addresses.
  d. Attackers generate an address with the first and last characters matching the legitimate address.
  e. Users might accidentally copy the wrong address and send funds to it.



Token Transfer | Etherscan

etherscan.io/tokentxns?a=0x9c726d733681e4b417676947df85dec884479757&p=2

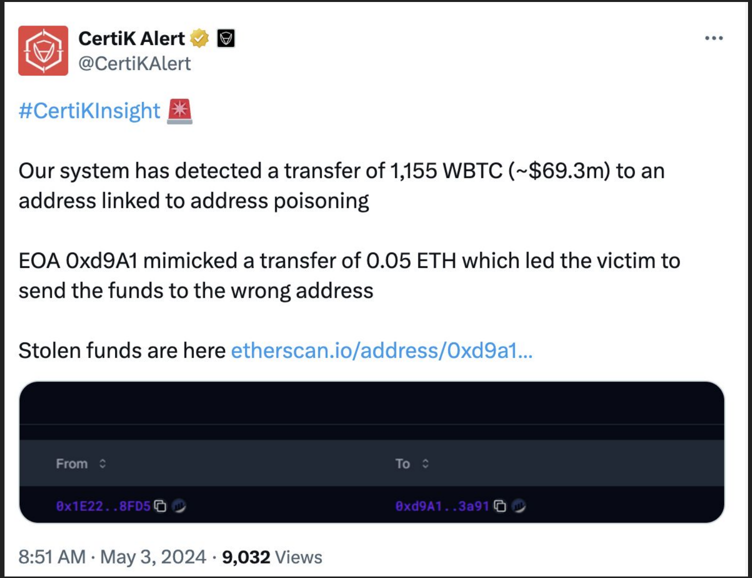
ETH Price: $3,222.76 (+2.47%) Gas: 29.426 Gwei

Search by Address / Txn Hash / Block / Token / Domain Name

0x251fd55128f...	Transfer	20870971	43 days ago	0x9c726d73...884479757	OUT	0xD80ba8ce...8f0899346	1,800	Tether USD (USDT)
0xaedf5d95361...	Transfer	20863253	44 days ago	0x9c726d73...884479757	OUT	0xDc52b78b...5715c8553	830	Tether USD (USDT)
0x13c6dd4732...	Transfer	20861709	44 days ago	0x9c726d73...884479757	OUT	⚠ 0xE202a43B...47795f161	0	Tether USD (USDT)
0xc81d8635d7...	Transfer	20861700	44 days ago	0x92ee1655...8cE1064F8	IN	0x9c726d73...884479757	30,000	Tether USD (USDT)
0x946435391b...	Transfer	20861622	44 days ago	0x9c726d73...884479757	OUT	0xE2027dcB...eAac3f161	30,000	Tether USD (USDT)
0x42f1e01a766...	Transfer	20857028	45 days ago	0x9c726d73...884479757	OUT	0x2b0Aa50b...09A366912	71.5666	Tether USD (USDT)
0xd3b430c533...	Transfer	20848084	46 days ago	0x9c726d73...884479757	OUT	0xBA39f73f...F10beAe87	16,105	Tether USD (USDT)
0xab7155a283...	Transfer	20847430	46 days ago	0x9c726d73...884479757	OUT	0x76476d52...5E4Cb64a1	16,000	Tether USD (USDT)

Case study: Human Error in Transaction Handling

- **Victim Loses $68 Million in Address Poisoning**
  - 1,155 wBTC was transferred to the attacker's address.


CertiK Alert
@CertiKAlert

#CertiKInsight 🚨

Our system has detected a transfer of 1,155 WBTC (~$69.3m) to an address linked to address poisoning

EOA 0xd9A1 mimicked a transfer of 0.05 ETH which led the victim to send the funds to the wrong address

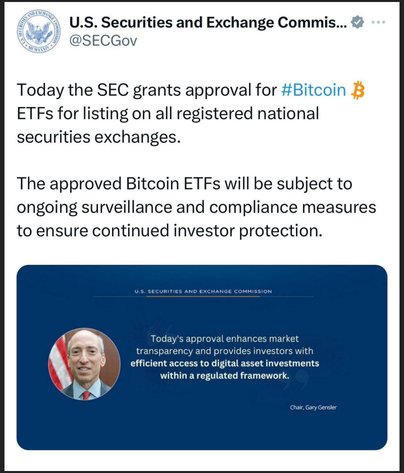
Stolen funds are here etherscan.io/address/0xd9a1...

From	To
0x1E22..8FD5	0xd9A1..3a91

8:51 AM · May 3, 2024 · **9,032** Views

Case study: Social Media Takeover

- **Social media accounts are a valuable asset for attackers.**
  - It is an effective platform to deliver phishing messages or false information.
  - People usually trust the official accounts they follow.
- **Attack vectors**
  - SIM Swap
  - Twitter Permission Phishing
  - Platform Vulnerabilities

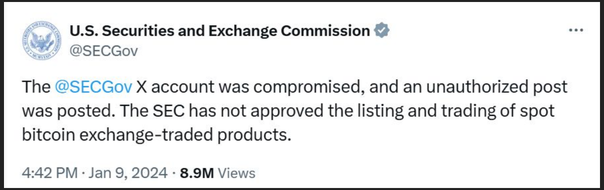


U.S. Securities and Exchange Commis... @SECGov

Today the SEC grants approval for #Bitcoin ₿ ETFs for listing on all registered national securities exchanges.

The approved Bitcoin ETFs will be subject to ongoing surveillance and compliance measures to ensure continued investor protection.

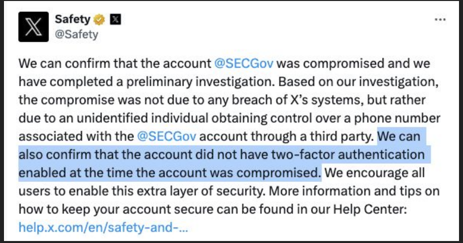
U.S. SECURITIES AND EXCHANGE COMMISSION
Today's approval enhances market transparency and provides investors with efficient access to digital asset investments within a regulated framework.
Chair, Gary Gensler



U.S. Securities and Exchange Commission @SECGov

The @SECGov X account was compromised, and an unauthorized post was posted. The SEC has not approved the listing and trading of spot bitcoin exchange-traded products.

4:42 PM · Jan 9, 2024 · **8.9M** Views



Safety @Safety

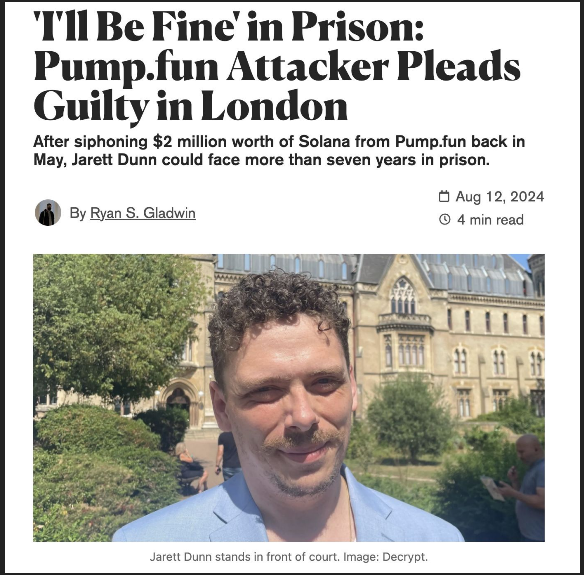
We can confirm that the account @SECGov was compromised and we have completed a preliminary investigation. Based on our investigation, the compromise was not due to any breach of X's systems, but rather due to an unidentified individual obtaining control over a phone number associated with the @SECGov account through a third party. We can also confirm that the account did not have two-factor authentication enabled at the time the account was compromised. We encourage all users to enable this extra layer of security. More information and tips on how to keep your account secure can be found in our Help Center: help.x.com/en/safety-and-...

Case study: Insider Threats

- **Insider Threats**
  - A malicious insider can take advantage of internal access to cause harm to the project or its users.
  - Insider threats are a rising risk in the blockchain space.
  - One of the most effective attack vectors to compromise a project once past the initial barrier.
- **What will insiders often do in Web3?**
  - Steal crypto keys, API keys, cloud access secrets, or any form of secret.
  - Inject malicious code into the application.
  - Steal user information from the backend system.

Case study: Insider Threats

- **Pump.fun Attack**
  - "Pump.fun" is a popular tool for launching meme coins on Solana.
  - A malicious developer used his privileged access to exploit the bonding curve contracts, stealing liquidity with the assistance of a flash loan.
  - The total loss for the protocol is $2M.



'I'll Be Fine' in Prison: Pump.fun Attacker Pleads Guilty in London

After siphoning $2 million worth of Solana from Pump.fun back in May, Jarett Dunn could face more than seven years in prison.

By Ryan S. Gladwin

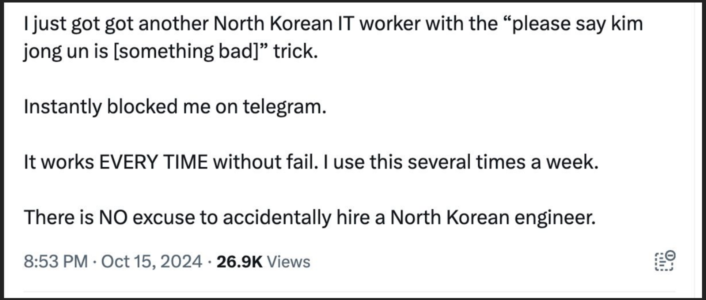
Aug 12, 2024
4 min read

Jarett Dunn stands in front of court. Image: Decrypt.

CERTIK

Case study: Insider Threats

- "Please say kim jong un is [something bad]"



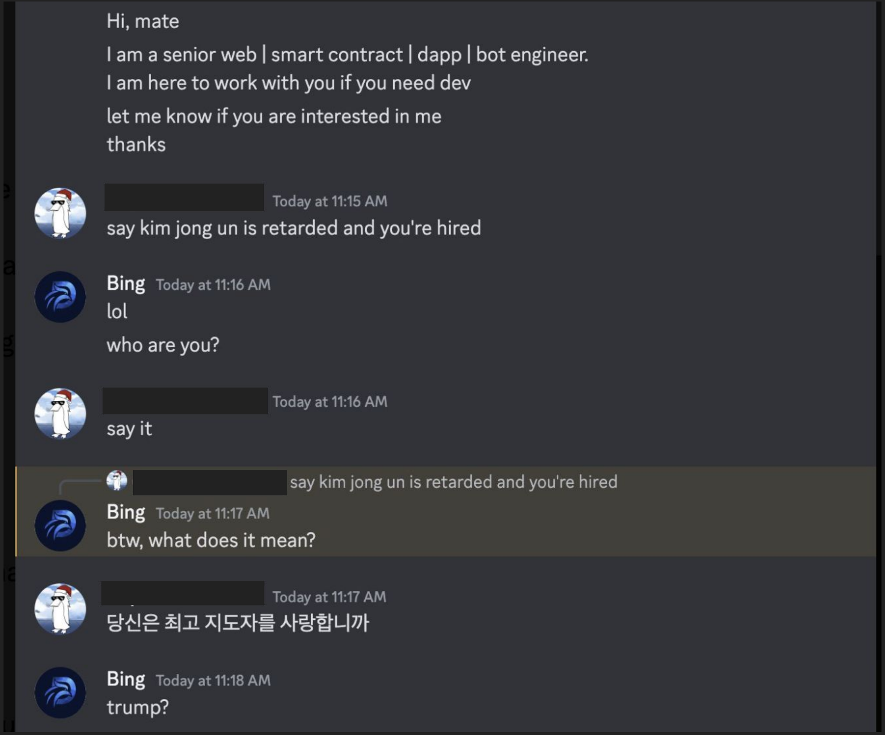
I just got got another North Korean IT worker with the "please say kim jong un is [something bad]" trick.

Instantly blocked me on telegram.

It works EVERY TIME without fail. I use this several times a week.

There is NO excuse to accidentally hire a North Korean engineer.

8:53 PM · Oct 15, 2024 · **26.9K** Views



Hi, mate
I am a senior web | smart contract | dapp | bot engineer.
I am here to work with you if you need dev
let me know if you are interested in me
thanks

[redacted] Today at 11:15 AM
say kim jong un is retarded and you're hired

Bing Today at 11:16 AM
lol
who are you?

[redacted] Today at 11:16 AM
say it

[redacted] say kim jong un is retarded and you're hired
Bing Today at 11:17 AM
btw, what does it mean?

[redacted] Today at 11:17 AM
당신은 최고 지도자를 사랑합니까

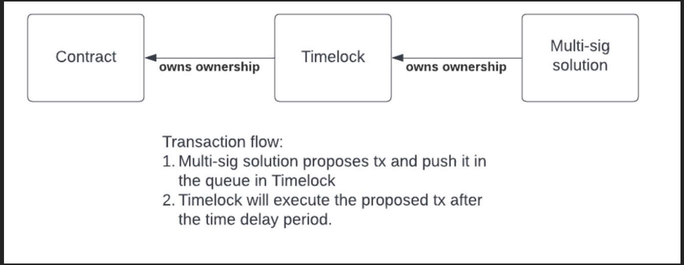
Bing Today at 11:18 AM
trump?

Best Practice and Defense

- **DeFi user**
  - Not much users can do. If you notice anything suspicious, stop interacting with the application immediately.

Best Practice and Defense

- **DeFi project, developer**
  - Use a timelock plus multisig setup for privileged roles in the smart contract.
    - **Timelock**: A sensitive action must wait for a certain period (e.g., 48 hours) before it can be executed on-chain.
    - **Multisig**: A sensitive action on the smart contract requires signatures from multiple wallets.



Transaction flow:
1. Multi-sig solution proposes tx and push it in the queue in Timelock
2. Timelock will execute the proposed tx after the time delay period.

  - Private Key Security: Enforce a security method to handle private keys.
    - Use hardware wallets.
    - Ensure multiple wallet keys are not stored in the same location or kept by the same person.
    - Use a dedicated environment when performing sensitive actions.

Best Practice and Defense

- **DeFi project, developer**
  - Establish strong social media security controls, including MFA, to prevent account takeovers.
  - Perform background checks on employees and enforce strict policies on internal access to sensitive information.
  - Do not copy addresses from blockchain explorers; confirm the payment address with other parties.
  - Regularly audit access logs and conduct internal audits to identify and address unusual or suspicious activity within the team.
  - Enforce incident response and escalation protocols to quickly detect, contain, and recover from potential internal security breaches.
  - Provide employee training on operational security and phishing awareness.

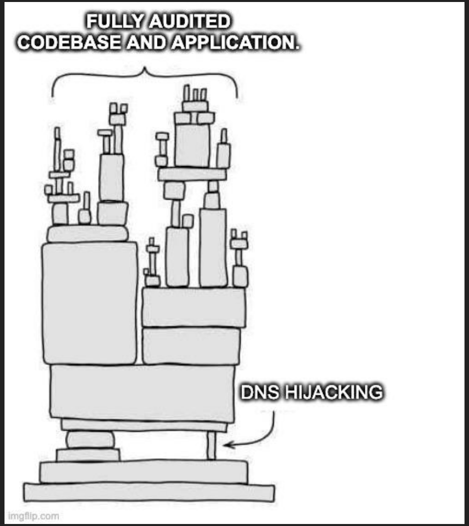
Best Practice and Defense

- **Security firm**
  - Projects that only willing to share limited access
    - Code audit/pentest and provide recommendations in the audit report.
    - Security Consultation
    - KYC Services
  - Project seeking in-depth collaboration
    - Internal security policy review
    - Internal configuration review
    - Red team engagement
    - Phishing training
    - Incident response, investigation, and evidence collection
  - Monitor and provide real-time alerts to the community

Cloud and Infrastructure Risks

Cloud and Infrastructure

- **The underlying technologies and services that support the deployment, hosting, and operation of the applications.**
- **Common Attack surface**
  - **Vulnerabilities in commonly used front-end stacks**
    - Next.js , Netlify, Vercel, Github pages
  - **DNS**
    - DNS/BGP hijacking, Sub-domain takeover
  - **Supply chain attack**
    - npm, yarn
  - **CDN**
    - Malicious JS file
  - **Misconfiguration in cloud service (e.g., S3 bucket)**
  - **Vulnerability in blockchain node server**



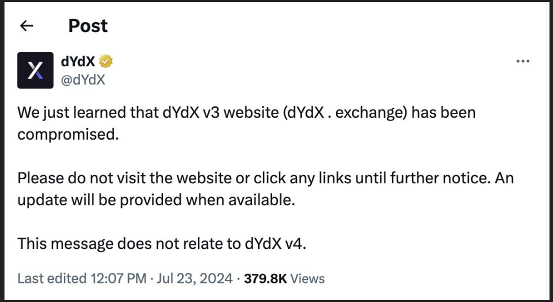
CERTIK

Case Study: DNS hijacking

- **DNS record**
  - Mapping of a domain name to an IP address
  - example.com <> 93.184.215.14
  - It's the address that the user's browser uses to load the website
- **Attack via DNS hijacking**
  - The attacker modifies the DNS record to redirect the user to a malicious site.
  - The malicious website contains JavaScript that tricks users into signing a malicious transaction to steal their assets.
  - Difficult for users to notice the anomaly.
- **Root cause**
  - The project's registrar account is compromised (e.g., GoDaddy, Cloudflare).
  - The DNS provider is compromised (e.g., Route 53, Squarespace).
  - Other undisclosed reasons.

CERTIK

Case Study: DNS hijacking


Post

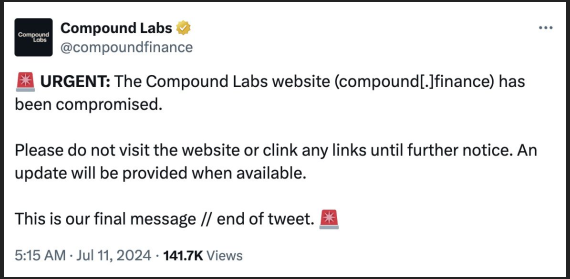
dYdX
@dYdX

We just learned that dYdX v3 website (dYdX . exchange) has been compromised.

Please do not visit the website or click any links until further notice. An update will be provided when available.

This message does not relate to dYdX v4.

Last edited 12:07 PM · Jul 23, 2024 · **379.8K** Views

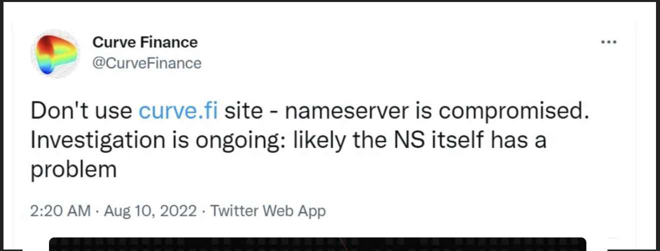

Compound Labs
@compoundfinance

🚨 **URGENT:** The Compound Labs website (compound[.]finance) has been compromised.

Please do not visit the website or clink any links until further notice. An update will be provided when available.

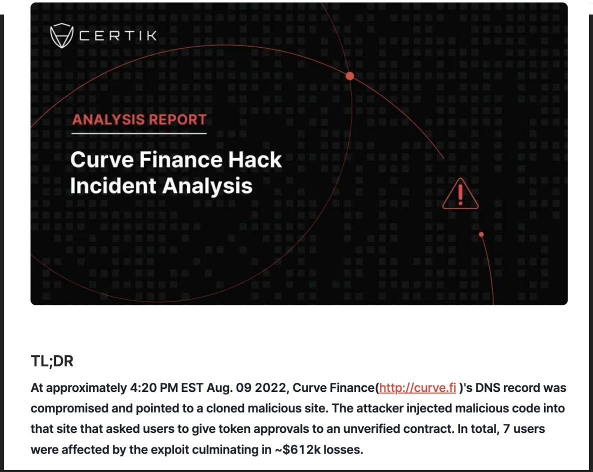
This is our final message // end of tweet. 🚨

5:15 AM · Jul 11, 2024 · **141.7K** Views


Curve Finance
@CurveFinance

Don't use curve.fi site - nameserver is compromised. Investigation is ongoing: likely the NS itself has a problem

2:20 AM · Aug 10, 2022 · Twitter Web App

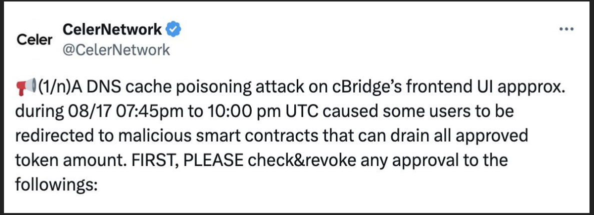

CERTIK

ANALYSIS REPORT

Curve Finance Hack Incident Analysis

TL;DR

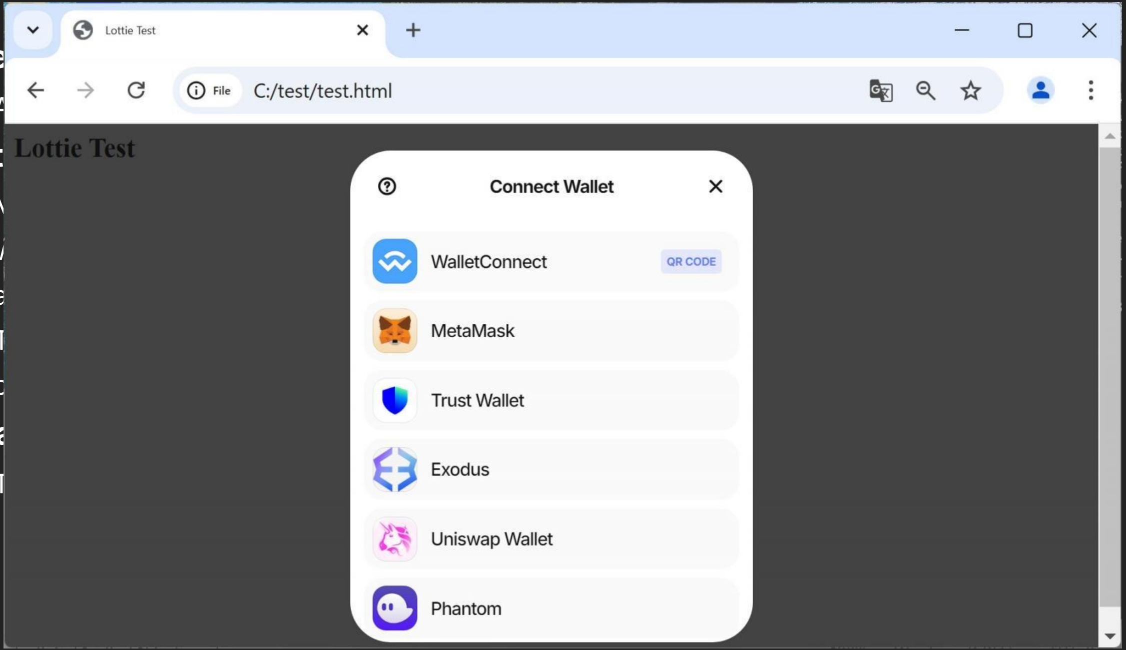
At approximately 4:20 PM EST Aug. 09 2022, Curve Finance(http://curve.fi)'s DNS record was compromised and pointed to a cloned malicious site. The attacker injected malicious code into that site that asked users to give token approvals to an unverified contract. In total, 7 users were affected by the exploit culminating in ~$612k losses.


Celer
CelerNetwork
@CelerNetwork

📢(1/n)A DNS cache poisoning attack on cBridge's frontend UI appprox. during 08/17 07:45pm to 10:00 pm UTC caused some users to be redirected to malicious smart contracts that can drain all approved token amount. FIRST, PLEASE check&revoke any approval to the followings:

Case Study: Supply chain attack

- @lottie
  - A
- Exploit
  1. M
  2. W
     a
  3. T
     c
- Root ca
  - T


Lottie Test

C:/test/test.html

Lottie Test

Connect Wallet

WalletConnect QR CODE

MetaMask

Trust Wallet

Exodus

Uniswap Wallet

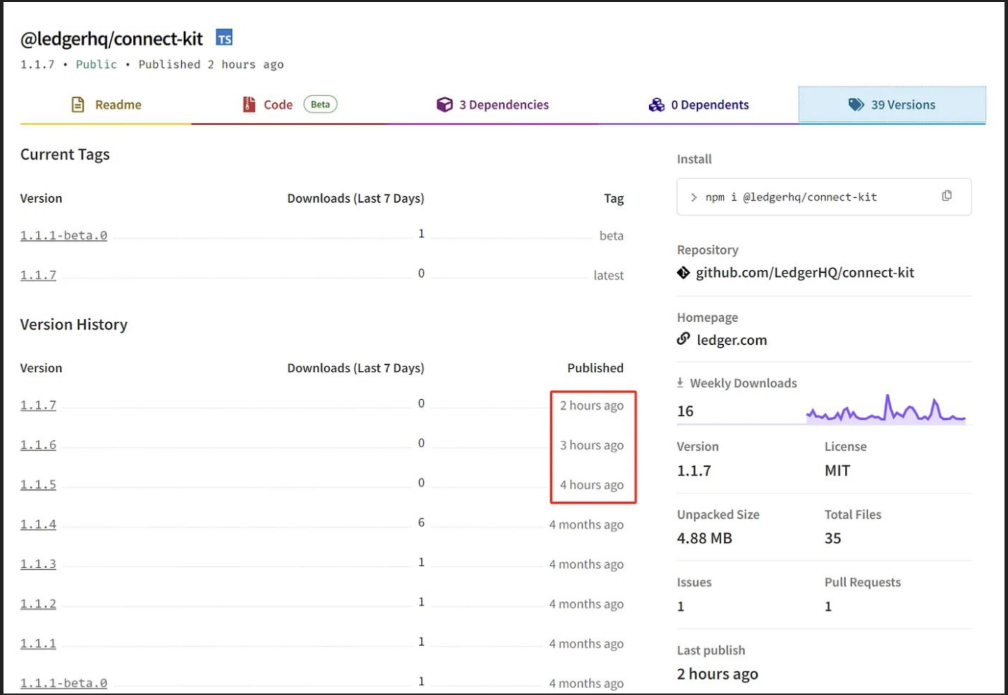
Phantom

Case Study: Supply chain attack

- **Ledger Connect Kit**
  - open-source JavaScript library allowing developers to connect DApps to the Ledger hardware wallet
  - Used in almost all premium DeFi projects.
- **Exploitation**
  1. The attacker phished a former Ledger employee to leverage the individual's access on NPMJS.
  2. The API key was compromised instead of the NPM login account.
  3. The attacker injected malicious code into the library, tricking users into signing malicious transactions.
- **Root cause**
  - The token belonged to a former employee.
  - Access was not manually revoked on NPMJS during the employee offboarding process.
  - The employee fell victim to a phishing attack.

CERTIK

Case Study: Supply chain attack


@ledgerhq/connect-kit TS

1.1.7 • Public • Published 2 hours ago

Readme | Code Beta | 3 Dependencies | 0 Dependents | 39 Versions

Current Tags

Version	Downloads (Last 7 Days)	Tag
1.1.1-beta.0	1	beta
1.1.7	0	latest

Version History

Version	Downloads (Last 7 Days)	Published
1.1.7	0	2 hours ago
1.1.6	0	3 hours ago
1.1.5	0	4 hours ago
1.1.4	6	4 months ago
1.1.3	1	4 months ago
1.1.2	1	4 months ago
1.1.1	1	4 months ago
1.1.1-beta.0	1	4 months ago

Install

```
> npm i @ledgerhq/connect-kit
```

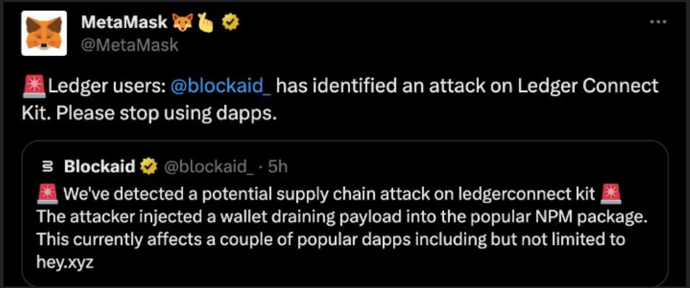
Repository
github.com/LedgerHQ/connect-kit

Homepage
ledger.com

Weekly Downloads
16

Version	License
1.1.7	MIT
Unpacked Size	Total Files
4.88 MB	35
Issues	Pull Requests
1	1

Last publish
2 hours ago

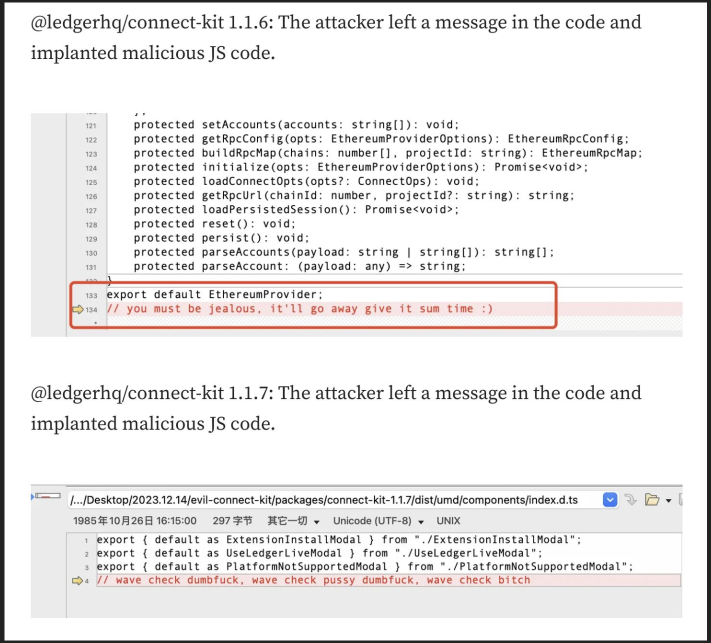

MetaMask 🦊🫰
@MetaMask

🚨Ledger users: @blockaid_ has identified an attack on Ledger Connect Kit. Please stop using dapps.

Blockaid @blockaid_ · 5h
🚨 We've detected a potential supply chain attack on ledgerconnect kit 🚨
The attacker injected a wallet draining payload into the popular NPM package.
This currently affects a couple of popular dapps including but not limited to hey.xyz

Case Study: Supply chain attack

The attacker left some message in the code and implanted malicious JS code.



@ledgerhq/connect-kit 1.1.6: The attacker left a message in the code and implanted malicious JS code.

```
    protected setAccounts(accounts: string[]): void;
    protected getRpcConfig(opts: EthereumProviderOptions): EthereumRpcConfig;
    protected buildRpcMap(chains: number[], projectId: string): EthereumRpcMap;
    protected initialize(opts: EthereumProviderOptions): Promise<void>;
    protected loadConnectOpts(opts?: ConnectOps): void;
    protected getRpcUrl(chainId: number, projectId?: string): string;
    protected loadPersistedSession(): Promise<void>;
    protected reset(): void;
    protected persist(): void;
    protected parseAccounts(payload: string | string[]): string[];
    protected parseAccount: (payload: any) => string;
}
export default EthereumProvider;
// you must be jealous, it'll go away give it sum time :)
```

@ledgerhq/connect-kit 1.1.7: The attacker left a message in the code and implanted malicious JS code.

/.../Desktop/2023.12.14/evil-connect-kit/packages/connect-kit-1.1.7/dist/umd/components/index.d.ts

1985年10月26日 16:15:00   297 字节   其它一切   Unicode (UTF-8)   UNIX

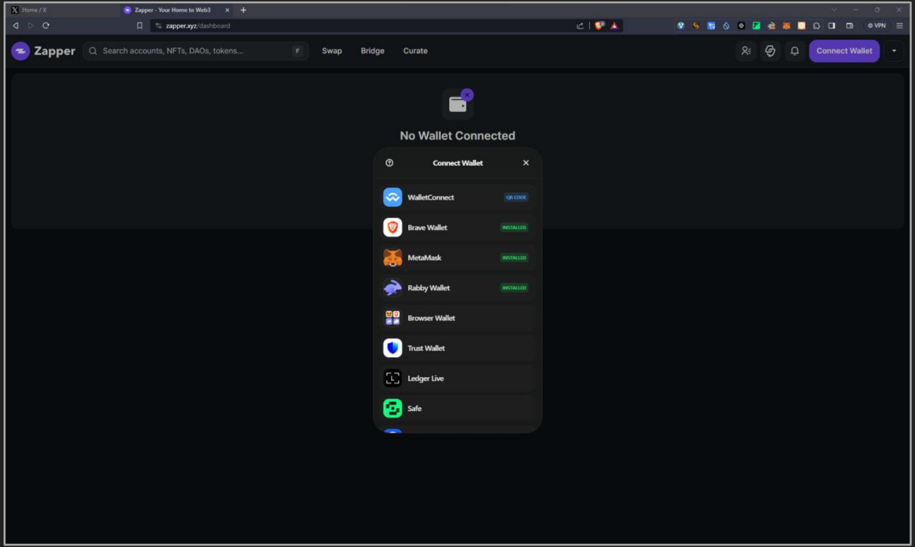
```
export { default as ExtensionInstallModal } from "./ExtensionInstallModal";
export { default as UseLedgerLiveModal } from "./UseLedgerLiveModal";
export { default as PlatformNotSupportedModal } from "./PlatformNotSupportedModal";
// wave check dumbfuck, wave check pussy dumbfuck, wave check bitch
```

Case Study: Blockchain node server RCE(remote code execution)

- **InfStones**
  - Blockchain node/validator infrastructure provider.
  - Launch a blockchain node with just a few clicks.
  - A private key stored on or connected to the server is required to perform block validation.
- **Vulnerability and exploitation**
  - A network service called "Tailon" for reading log files is running on TCP port 55555.
  - The exposed service can be used to run system commands on the server.
  - The researcher found that over 400 nodes owned by the company are vulnerable.
- **Impact**
  - Enables theft of assets from the validator nodes.
  - Can cause the nodes to get "slashed."

Best Practice and Defense

- **DeFi user**
  - If the user notice anything suspicious, stop interacting with the application.
    - Malicious transaction or message signing request from a DNS-hijacked website.



Best Practice and Defense

- **DeFi project, developer**
  - Use a reputable DNS registrar.
  - Employ strong protection over the account that can modify infrastructure configurations (e.g., DNS, software packages).
  - Implement strict Content Security Policies to mitigate risks from malicious JavaScript files served via CDNs.
  - Using a pinned version in software packages*
  - Provide developer training on DevOpsSec best practices and strictly adhere to them.
  - If anything is compromised, immediately notify the community to avoid interaction with the compromised component.
  - Set up a comprehensive incident response plan to quickly resolve issues if exploited.
  - Conduct regular phishing awareness training.
  - Hire a security firm to conduct an infrastructure security assessment.

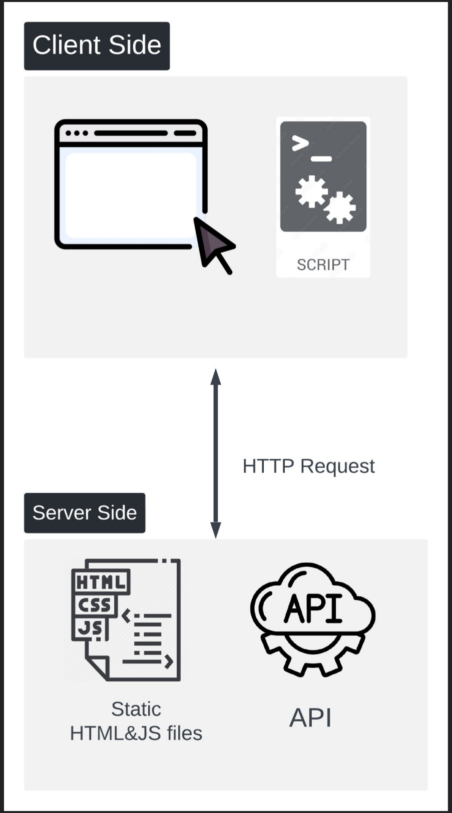
Best Practice and Defense

- **Security firm**
    - Establish a reporting channel for users to submit incidents.
    - Monitor threats in the space and provide public real-time alerts.
        - Compromised website
        - Compromised dependency
        - Reach out to clients to deliver warnings and offer assistance.
    - Perform security audits of cloud infrastructure to help clients identify and resolve vulnerabilities and weakness.
    - Provide training and guidance on best security practices for developers to reduce the risk of human error and insecure practices.
    - Provide phishing awareness training.

Web2 Application Security Risk

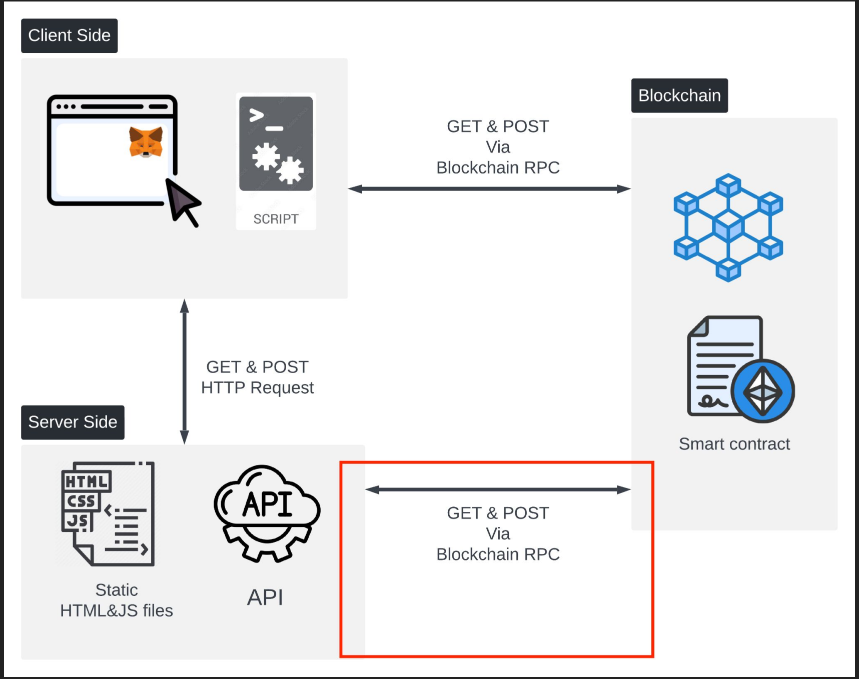
Standard Web2 Application

- **Client-side application**
- **Server-side components**
  - Web server & APIs
  - Database
  - etc



Web3 Dapp Architecture

- **Client-side application**
  - An essential UI for users to interact with the application.
  - Read data from the blockchain (smart contract).
  - Construct a transaction with user input for the wallet to sign.
  - The wallet send the signed tx to the blockchain.
- **Server-side components**
  - Statis HTML/JS files, Server API
  - The server communicates with the blockchain to read and write(via transactions) data
  - The private key of privileged roles (e.g., owner/admin) of the smart contract is stored on the server.



CERTIK

Common AppSec-heavy Web3 product categories

- **Cryptocurrency Exchange(CEX)**
  - Centralized platform where users can buy, sell, and trade cryptocurrencies.
- **Data platform**
  - An application for users to view transaction information, token flows, analytical data, etc. Examples include Etherscan, Arkham, and Dune.
- **Wallet**
  - An application that holds the private key and uses it to perform signing when requested. Wallets can come in various forms such as browser extensions, mobile apps, and web apps.
- **Dapps**
  - A wide range of applications include functionality implemented through the project's on-chain smart contracts.

CERTIK

Web2 Client Side Attack

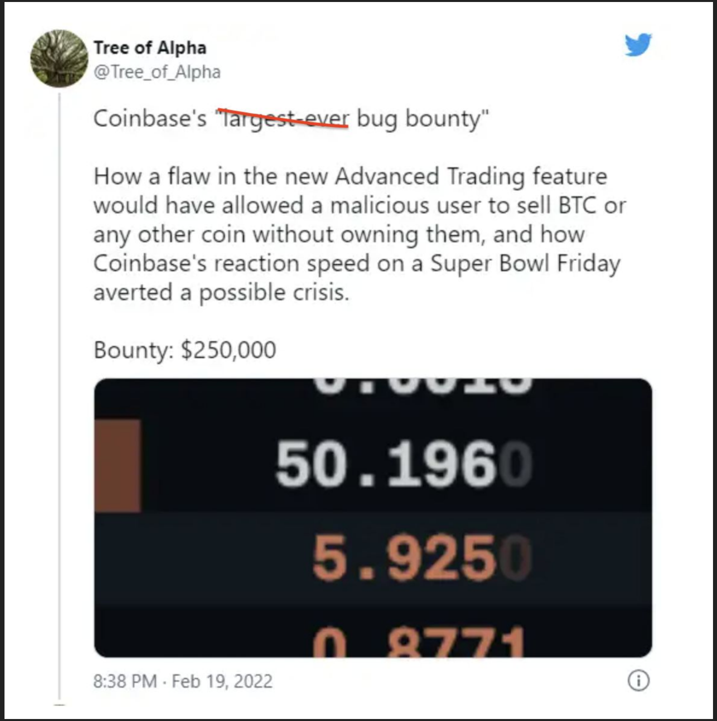
- **Initial malicious transaction**
  - XSS
  - Subdomain takeover
  - DNS hijacking
  - Supply chain attacks
- **Deceive users into signing malicious transactions with UI element tampering.**
  - HTML Injection
  - Clickjacking
- **Redirection to a phishing site**
  - Open redirect

Web2 Server Side Attack

- **Stealing keys from the server**
  - RCE, SQLi, LFI, Directory Traversal, SSRF, Debug page, Server misconfiguration, etc
- **Steal crypto assets from other user accounts or via admin features**
  - Broken access control
  - Privilege escalation
- **Spend/own assets more than the user's actual balance**
  - Business logic error
  - Race condition
- **Take down the server that hosts NFT images and metadata.**
  - Denial of service

Case study: Business Logic Error

- **Business logic error in Coinbases trading API**
  - Allow any users to sell any coin at coinbase with the price of Bitcoin
- The vulnerability was submitted by a whitehat to Coinbase's bug bounty program
  - 250,000$ bounty award


Tree of Alpha
@Tree_of_Alpha

Coinbase's "~~largest-ever~~ bug bounty"

How a flaw in the new Advanced Trading feature would have allowed a malicious user to sell BTC or any other coin without owning them, and how Coinbase's reaction speed on a Super Bowl Friday averted a possible crisis.

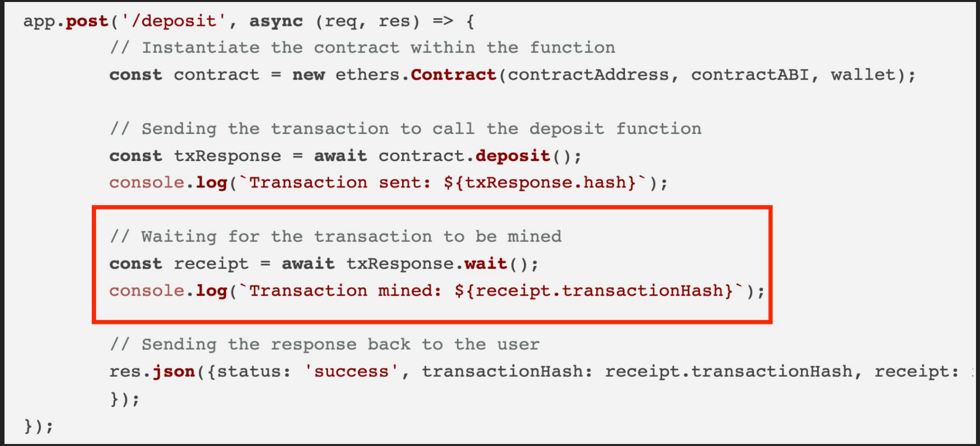
Bounty: $250,000

8:38 PM · Feb 19, 2022

CERTIK

Case study: Denial of Service

- **Finding**
  - For blockchains with long block times (e.g., 12 seconds for Ethereum), the HTTP request and connection will remain open for a considerable amount of time.
  - **The API server can be easily subjected to a DoS attack with concurrent requests sent from a single machine.**



```
app.post('/deposit', async (req, res) => {
        // Instantiate the contract within the function
        const contract = new ethers.Contract(contractAddress, contractABI, wallet);

        // Sending the transaction to call the deposit function
        const txResponse = await contract.deposit();
        console.log(`Transaction sent: ${txResponse.hash}`);

        // Waiting for the transaction to be mined
        const receipt = await txResponse.wait();
        console.log(`Transaction mined: ${receipt.transactionHash}`);

        // Sending the response back to the user
        res.json({status: 'success', transactionHash: receipt.transactionHash, receipt:
        });
});
```

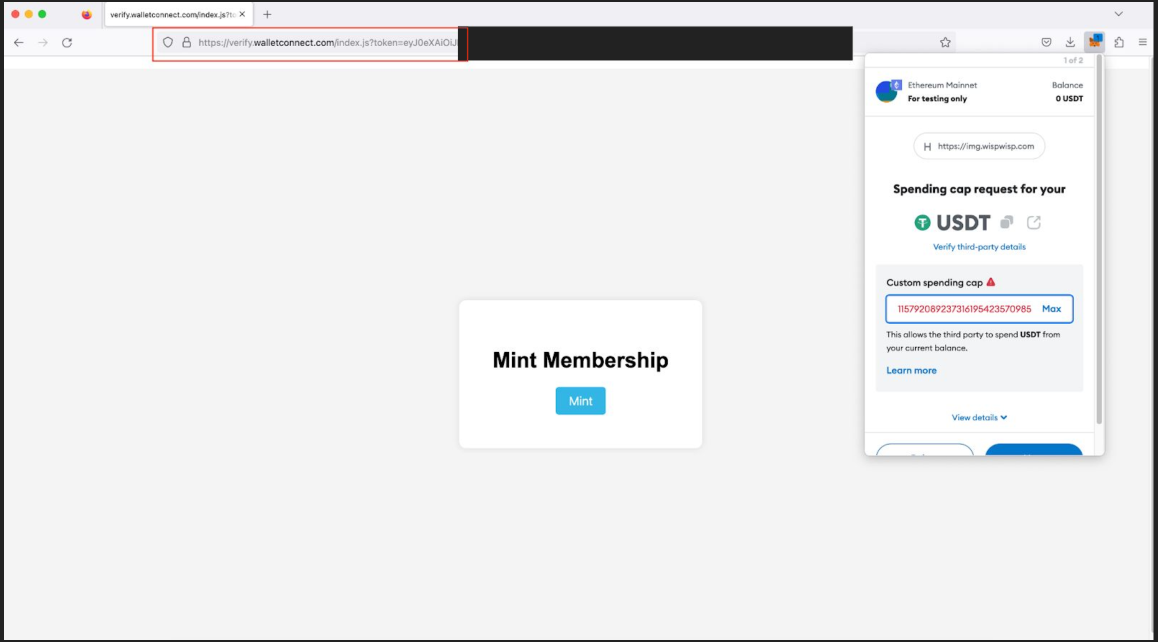
Case study: Client side XSS Attack

- **Cross-Site Scripting Attack in WalletConnect**
  - WalletConnect is one of the most popular tools that connect Dapps with users' wallets.
  - An XSS vulnerability was discovered in one of its APIs.
  - The vulnerability can be used to perform phishing attacks against users, exploiting the trust of the WalletConnect domain.
  - CertiK reported the vulnerability and received a bounty from the team.

CERTIK
Uncovering and Resolving a Cross-Site Scripting Attack in a Popular Wallet Protocol

Case study: Client side XSS Attack

- **Cross-Site Scripting Attack in WalletConnect**


verify.walletconnect.com/index.js?t

https://verify.walletconnect.com/index.js?token=eyJ0eXAiOiJ [redacted]

1 of 2

Ethereum Mainnet
For testing only

Balance
0 USDT

https://img.wispwisp.com

Spending cap request for your

USDT

Verify third-party details

Custom spending cap

115792089237316195423570985 Max

This allows the third party to spend **USDT** from your current balance.

Learn more

View details

Mint Membership

Mint

The state of Dapp security

- With the advancement of the blockchain ecosystem, Dapps are becoming increasingly complex and are starting to integrate with heavy Web2 backend components.
- Often, vulnerabilities are introduced in Dapps because developers may lack knowledge of either Web2 or Web3 security.
- Web2 vulnerabilities in Web3 applications often have increased severity and higher impact when exploited. Nothing is more impactful than the loss of millions of $$$$.
    - Unlike a smart contract exploit, it's publicly viewable. The exploitability of the backend may have far less exposure, but it still happened.
- **To secure a Dapps, especially a Dapp with heavy backend components, it is as important to get a penetration test as it is to audit the smart contract.**

Best Practice and Defense

- **DeFi user**
    - For server-side attacks, there isn't much users can do.
    - For client-side attacks, if the user notice anything suspicious, stop interacting with the application.
    - Avoid low-quality DeFi projects.

CERTIK

Best Practice and Defense

- **DeFi project, developer**
  - Provide developer training on Web2 security vulnerability topics.
  - Obtain Web2 penetration testing from a security firm.
  - Set up a bug bounty for whitehats to report security vulnerability
- **Security firm**
  - Offer penetration testing and security consulting services to DeFi projects.
  - Monitor threats within the ecosystem and alert the community to avoid compromised projects.
  - Conduct security research to understand the latest Dapp development trends and their attack surfaces, to better protect clients.

CertiK Penetration Testing Service Offering

Blackbox Security

- **Approach:** Assessment conducted externally
- **Requirements:** No need for source code; test account access recommended.
- **Focus:** Web portal, Mobile DApp, API.
- **Features:** External attacker simulation, cost-effective

Whitebox Security

- **Approach:** Source level code auditing
- **Requirements:** Source code needed
- **Focus**: Web, backend Infra, Mobile DApp, API, SDK
- **Features:** Comprehensive and high coverage, line-by-line review to uncover hidden bugs

Apply Standards

- ❖ OWASP Web
- ❖ OWASP MASVS + MASTG
- ❖ NIST
- ❖ PTES
- ❖ CVSS

CERTIFICATIONS

OSWE · OSCE · OSCP · aws certified

Tailor Web3 Pentest Solutions For

Crypto Exchange

Exchange pentest that covers beyond Web2 vulnerabilities.

- Infrastructure
- Front-end/client side security
- Server side APIs
- Data protection

Cover Exchange Specific Logics

- ✔ "Fake" deposits
- ✔ KYC/Onboarding process
- ✔ Buying with fiat
- ✔ Asset exchange: Manipulation risks
- ✔ Trading: Price and date tricks
- ✔ Loans: Valuation issues

BingX · CoinW · LATOKEN · OKX

Wallet

Comprehensive wallet security audit with a focus on key safety.

Full Wallet Platforms

- ✔ Web & Mobile
- ✔ Chrome Extension
- ✔ Infrastructure
- ✔ Desktop Application

MPC Wallet Assessment

- ✔ Multi-party Interaction
- ✔ Network Communication
- ✔ Backup Security
- ✔ Share generation & storage
- ✔ MPC Cryptography Impl.

Cover Wallet Risks

- ✔ Randomness
- ✔ Key Handling
- ✔ Use of Cryptography
- ✔ Key Generation & Storage
- ✔ User Protections
- ✔ Transaction Integrity

ZenGo · TRUST · WalletConnect · SAFEHERON · OKX WALLET · BYBIT

DApp

Advanced Web3 DApp Penetration Testing

Specialized pentest expertises with deep knowledge in blockchain and smart contracts, understanding unique Web3 attack vectors.

- Cross Chain Bridge
  - Deposit/Withdraw Logic
  - On-chain Event Processing
- GameFi
  - Game Cheating/Anti-Scripting Measures
- Marketplace/DEX/Lending
  - Client-side vulnerabilities(e.g. XSS)
  - Trading logic assessment

WEMIX · hatom · QTUM · GALA GAMES

Blockchain

Comprehensive Blockchain Dynamic Testing: Elevating Security Beyond Source Code Review

- L1/L2 Blockchain
- Validators Notes
- RPC Endpoints
- Blockchain Network

Assessment Types

- ✔ Network Penetration Testing
- ✔ Architecture Review
- ✔ Cloud Infrastructure Security Review
- ✔ Performance Testing

GALA GAMES · TON

Web3 Contract Exploit

Web3 Contract Exploit

- **Smart contracts refer to programs deployed and run on the blockchain.**
  - The code is often open-source and often visible on the blockchain.
  - Web3 contracts are immutable by default after deployment.
  - Once a transaction is executed, it cannot be undone.
  - DeFi protocols manage massive capital, making them attractive targets for attackers seeking large payouts.
- **Smart contracts Exploits**
  - The attacker crafts a malicious transaction that exploits a vulnerability in the smart contract, with the goal, most of the time, of withdrawing crypto assets from the contract.
  - Occasionally, some critical vulnerabilities can stop a smart contract from functioning, putting it into a locked state that traps all the crypto assets inside the contract.

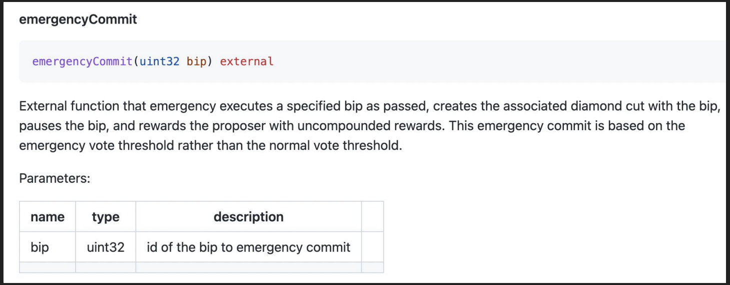
CERTIK

Web3 Contract Exploit

- **Common Web3 Contract Attack Vectors**
  - **Language Specific Risk**
    - Function Visibility, Compiler Version, Event, Low-level Call, Storage Risk etc
  - **Common Security Issues**
    - Input Validation, Reentrancy Attack, Access Control, etc
  - **Business Logic Design Flaw**
    - Abnormal Arbitrages, Inconsistent User Behavior results, funds being locked, malicious calls, etc
  - **Mathematical Operation Risk**
    - Precision Loss, Overflow/Underflow, Incorrect Math Calculation, etc
  - **Price/Balance Manipulation Risk**
    - Unexpected Price Update, Unintended Balance Change, etc
  - **Governance Risk**
    - Private Key Leakage, Voting Power Manipulation, etc
  - **Incentive Mechanisms Design Flaw**
    - Unfair Reward Distribution, etc
  - **Cross-Chain Risk**
    - Vulnerable Proof Verification, Replay Attack, etc)

Case Study: Governance Attack

- A flash loan exploit occurred on April 17, 2022, at Beanstalk Farms. Approximately $182 million was lost. The incident was due to a vulnerability in the governance mechanism, in which the attacker used a flash loan to amplify their voting power and ultimately control the result to pass a malicious proposal, draining the funds.
- **Attack Steps**
  - Create a malicious proposal.
  - Flashloan tokens to vote for this proposal.
  - Trigger emergencyCommit() to immediately execute the proposal.
  - Transfer assets in the contract to the attacker.



emergencyCommit

```
emergencyCommit(uint32 bip) external
```

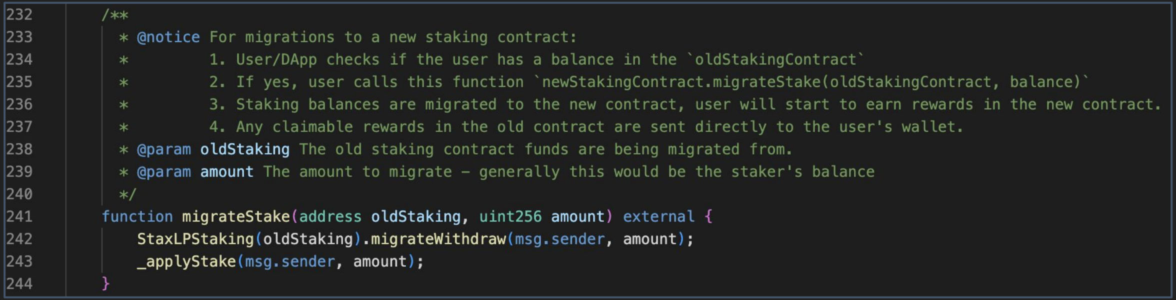
External function that emergency executes a specified bip as passed, creates the associated diamond cut with the bip, pauses the bip, and rewards the proposer with uncompounded rewards. This emergency commit is based on the emergency vote threshold rather than the normal vote threshold.

Parameters:

name	type	description
bip	uint32	id of the bip to emergency commit

Case Study: Lack of Function Parameter Validation

- **TempleDAO Exploit, Oct 11, 2022**
  - The victim contract is a staking contract where users stake LP token and receive rewards
  - The "migrateStake()" function is intended to allow users to migrate their stake from an old staking contract to a new one
  - However, the `oldStaking` address is provided by the user, and the contract does not verify the user provided address
  - By providing a malicious `oldStaking` address, the function caller can receive staking credit for free



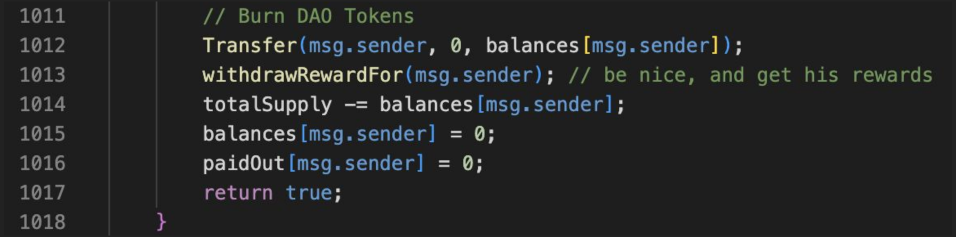
```
    /**
     * @notice For migrations to a new staking contract:
     *         1. User/DApp checks if the user has a balance in the `oldStakingContract`
     *         2. If yes, user calls this function `newStakingContract.migrateStake(oldStakingContract, balance)`
     *         3. Staking balances are migrated to the new contract, user will start to earn rewards in the new contract.
     *         4. Any claimable rewards in the old contract are sent directly to the user's wallet.
     * @param oldStaking The old staking contract funds are being migrated from.
     * @param amount The amount to migrate - generally this would be the staker's balance
     */
    function migrateStake(address oldStaking, uint256 amount) external {
        StaxLPStaking(oldStaking).migrateWithdraw(msg.sender, amount);
        _applyStake(msg.sender, amount);
    }
```

CERTIK

Case Study: Reentrancy

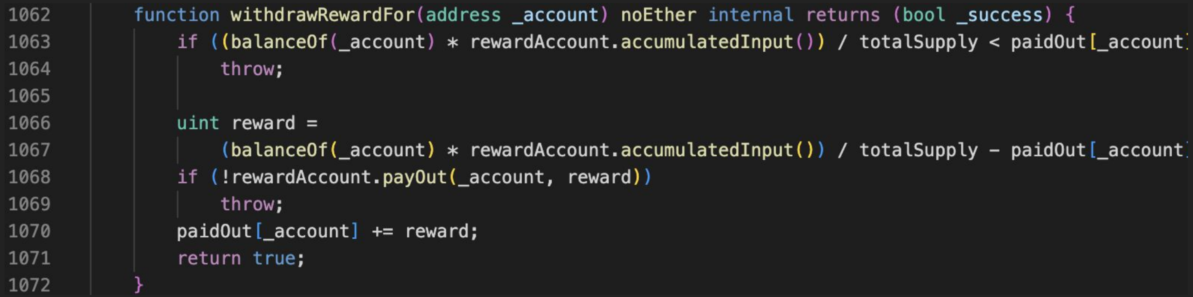
Ethereum DAO Hack, June 2016

- The splitDAO() function allows the caller to withdraw reward
- Note that the balances of the caller is updated after the withdrawRewardFor() function



```
        // Burn DAO Tokens
        Transfer(msg.sender, 0, balances[msg.sender]);
        withdrawRewardFor(msg.sender); // be nice, and get his rewards
        totalSupply -= balances[msg.sender];
        balances[msg.sender] = 0;
        paidOut[msg.sender] = 0;
        return true;
    }
```

- The amount of reward payout is based on the balance of the caller

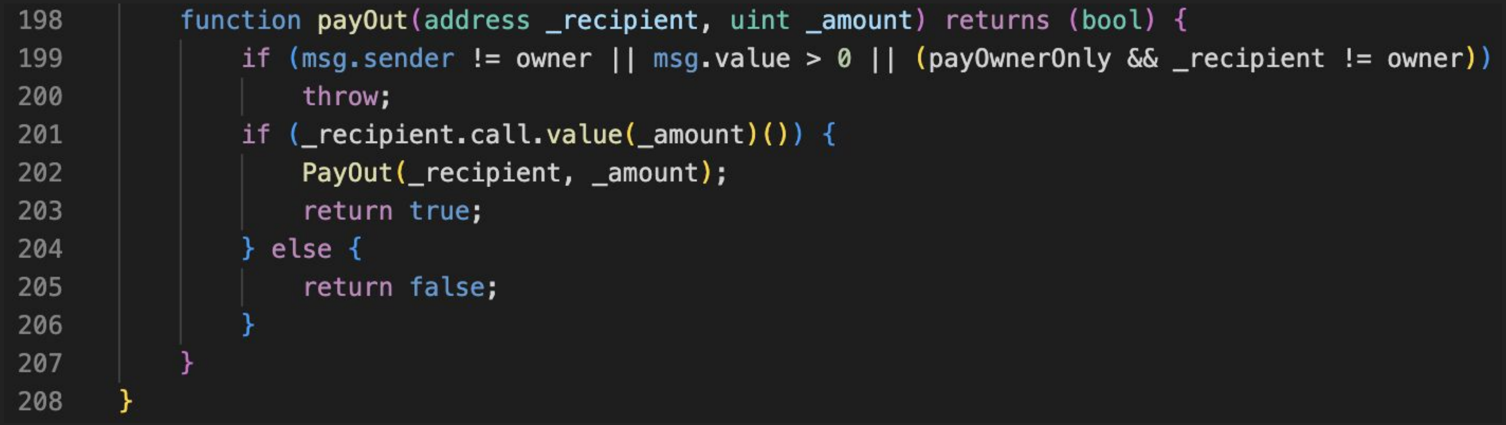


```
function withdrawRewardFor(address _account) noEther internal returns (bool _success) {
    if ((balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply < paidOut[_account
        throw;

    uint reward =
        (balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply - paidOut[_account
    if (!rewardAccount.payOut(_account, reward))
        throw;
    paidOut[_account] += reward;
    return true;
}
```

Case Study: Reentrancy

Ethereum DAO Hack, June 2016 (cont'd)

- The payOut() function contains a low level call to the function caller address containing Ether
- Upon receiving the Ether, the attacker reenters the splitDAO() function

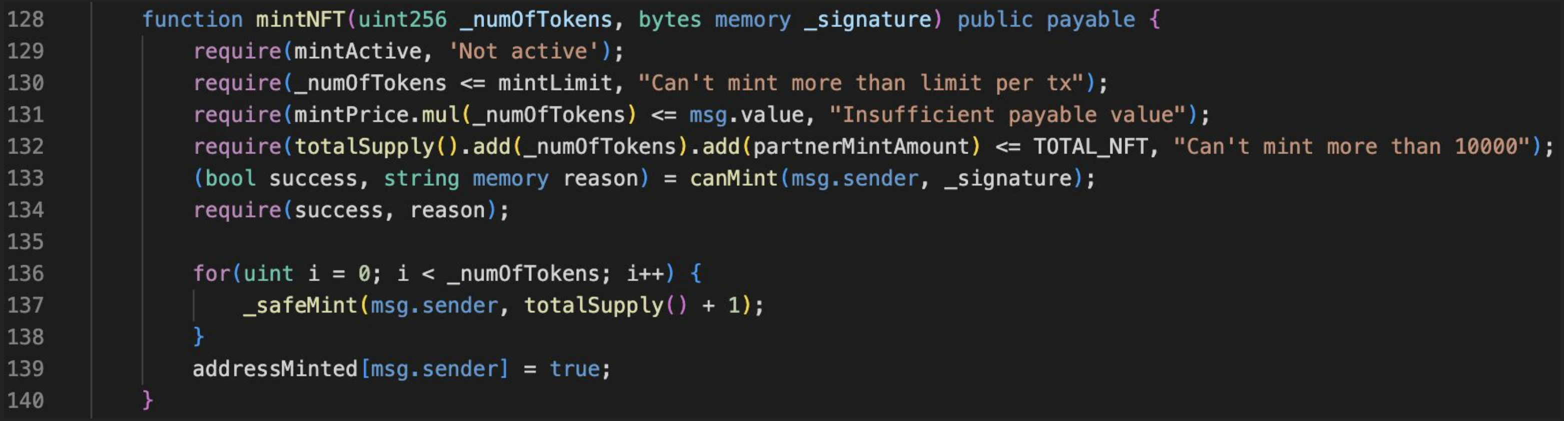


```
    function payOut(address _recipient, uint _amount) returns (bool) {
        if (msg.sender != owner || msg.value > 0 || (payOwnerOnly && _recipient != owner))
            throw;
        if (_recipient.call.value(_amount)()) {
            PayOut(_recipient, _amount);
            return true;
        } else {
            return false;
        }
    }
}
```

CERTIK

Case Study: Reentrancy

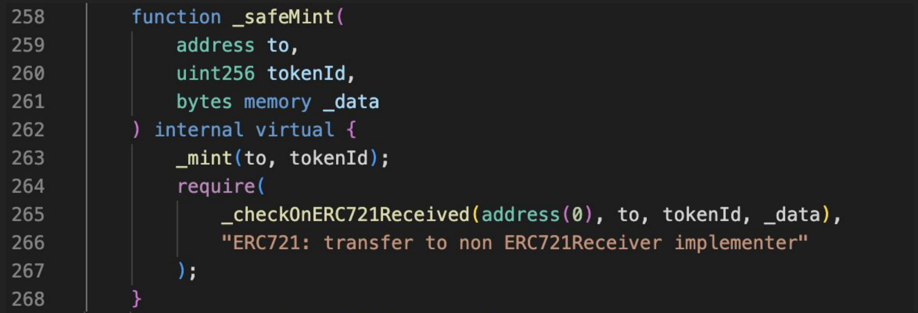
- **HyperBear NFT, Feb 3, 2023**
  - The contract extends the ERC721 contract and allows users to mint the HyperBear NFT
  - The intended design is that a user can only mint once, and if addressMinted[msg.sender] is true, the canMint() function returns false
  - Note that addressMinted[msg.sender] is updated after the _safeMint() function call. Attacker was able to bypass the addressMinted flag check through Reentrancy.



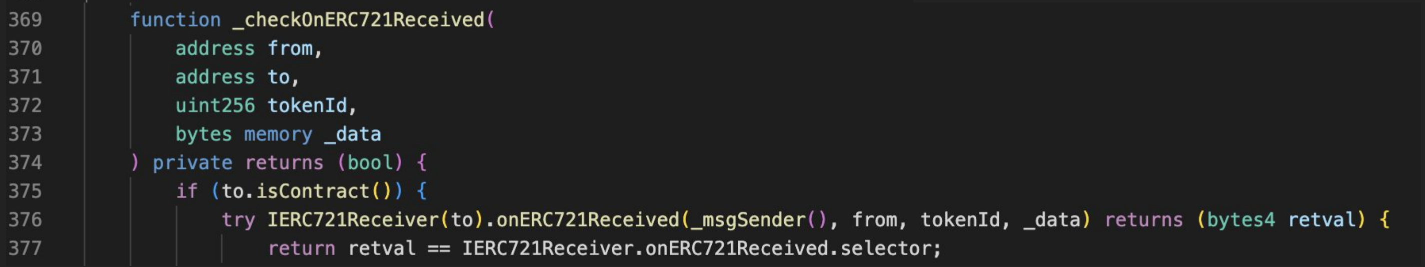
```
function mintNFT(uint256 _numOfTokens, bytes memory _signature) public payable {
    require(mintActive, 'Not active');
    require(_numOfTokens <= mintLimit, "Can't mint more than limit per tx");
    require(mintPrice.mul(_numOfTokens) <= msg.value, "Insufficient payable value");
    require(totalSupply().add(_numOfTokens).add(partnerMintAmount) <= TOTAL_NFT, "Can't mint more than 10000");
    (bool success, string memory reason) = canMint(msg.sender, _signature);
    require(success, reason);

    for(uint i = 0; i < _numOfTokens; i++) {
        _safeMint(msg.sender, totalSupply() + 1);
    }
    addressMinted[msg.sender] = true;
}
```

Case Study: Reentrancy

- **HyperBear NFT, Feb 3, 2023 (cont'd)**
  - The _safeMint() function contains a _checkOnERC721Received() hook to the caller address, which allowed reentrancy to the mintNFT() function



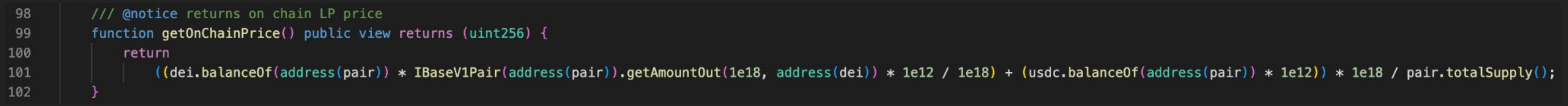
```
function _safeMint(
    address to,
    uint256 tokenId,
    bytes memory _data
) internal virtual {
    _mint(to, tokenId);
    require(
        _checkOnERC721Received(address(0), to, tokenId, _data),
        "ERC721: transfer to non ERC721Receiver implementer"
    );
}
```



```
function _checkOnERC721Received(
    address from,
    address to,
    uint256 tokenId,
    bytes memory _data
) private returns (bool) {
    if (to.isContract()) {
        try IERC721Receiver(to).onERC721Received(_msgSender(), from, tokenId, _data) returns (bytes4 retval) {
            return retval == IERC721Receiver.onERC721Received.selector;
```

Case Study: Price / Oracle Manipulation

- **Deus Finance Exploit, April 28, 2022**
  - Deus Finance provides lending solution based on DEI token collateral
  - The price of DEI measured in USDC is based solely on the liquidity of the DEI-USDC Pair
  - An attacker used flash loan to purchase a massive amount of DEI token, inflating its price, and then used it as collateral for borrowing
  - The attacker then used the borrowed amount to repay the flash loan



```
/// @notice returns on chain LP price
function getOnChainPrice() public view returns (uint256) {
    return
        ((dei.balanceOf(address(pair)) * IBaseV1Pair(address(pair)).getAmountOut(1e18, address(dei)) * 1e12 / 1e18) + (usdc.balanceOf(address(pair)) * 1e12)) * 1e18 / pair.totalSupply();
}
```

CERTIK

Best Practice and Defense

- **DeFi user**
  - Build basic security awareness
  - Always check if a project has been audited by a reputable security firm and review the audit findings to understand potential risks and mitigations. Avoid projects with no audit or unclear security assurances.
  - Follow official security channels and community alerts for the latest information on vulnerabilities or exploits. If a project is compromised, take immediate action to withdraw or secure your funds.
  - Be vigilant when signing approvals, especially with contracts that request high permissions. Limit approvals to necessary amounts and revoke unused permissions regularly to minimize risk.

Best Practice and Defense

- **DeFi project, developer**
  - **Follow coding best practices.**
    - Follow the "Checks-Effects-Interactions" Pattern
    - Apply nonReentrant modifier to critical functions that handle funds or state changes
    - Treat user-supplied addresses as potentially hostile, as they can include arbitrary code or logic. Always validate and restrict user-supplied address usage.
  - **Implement security checks specific to the business logic.**
    - Implement anti-flashloan mechanisms in governance by splitting the voting process across multiple blocks
    - Set Appropriate Voting Thresholds and Timelocks for Proposal
  - **Handle third-party dependency risks.**
    - Choose reliable, on-chain, multi-source price oracles such as Chainlink to secure accurate and robust pricing data. Avoid relying on a single AMM pair or low-liquidity source, as these are vulnerable to manipulation and may not reflect true market value

CERTIK

Best Practice and Defense

- **DeFi project, developer**
  - **Include comprehensive testing in the development lifecycle**
    - Unit testing
    - End-to-end testing
    - Fuzz testing
    - Invariant testing
  - **Provide security training on secure coding practices.**
  - **Ensure all code is audited before deploying on-chain and project launch.**
  - **Set up monitoring and an incident response plan to minimize losses.**
    - Automatically pause the contract to prevent further asset loss.
    - Warn users to withdraw assets from the affected contract.
    - Prepare compensation and recovery plan.

Best Practice and Defense

Best Practice and Defense

- **Security firm**
  - Provide comprehensive security audits for blockchain projects.
  - Develop detailed audit checklist to systematically identify potential attack vectors
  - Provide training and guidance on best security practices for developers to reduce the risk of insecure practices
  - Use fuzz testing, dynamic testing, and static analysis tools to uncover edge cases, simulate attacks, and reduce the likelihood of missed vulnerabilities during the audit process.
  - Establish alert channels to communicate security incidents or exploits to the community and investors promptly, providing users with real-time information to protect or recover their assets.
  - Develop a robust incident response process, including necessary whitehat rescue operation when appropriate to front-run or block attack transactions.

CERTIK

Question?

Securing The **Web3** World

CERTIK

Protect your community and your organization today.

Visit **CertiK.com** Get in touch at **bd@certik.com**