crypto.com

# Attacks and Exploits in DeFi

June 2021

crypto.com

**Research and Insights**

Macro Report

Research Analyst
Leo Leung

# RESEARCH DISCLAIMER

This report alone must not be taken as the basis for an investment decision. The user assumes the entire risk of any use made of this information. The information is provided merely complementary and does not constitute an offer, solicitation for the purchase or sale of any financial instruments, inducement, promise, guarantee, warranty, or as an official confirmation of any transactions or contract of any kind. The views expressed therein are based solely on information available publicly/internal data/other reliable sources believed to be true. This report includes projections, forecasts and other predictive statements which represent Crypto.com's assumptions and expectations in the light of currently available information. These projections and forecasts are based on industry trends, circumstances and factors which involve risks, variables and uncertainties. Opinions expressed therein are our current opinion as of the date appearing on the report only.

No representations or warranties have been made to the recipient as to the accuracy or completeness of the information, statements, opinions or matters (express or implied) arising out of, contained in or derived from this report or any omission from this document. All liability for any loss or damage of whatsoever kind (whether foreseeable or not) which may arise from any person acting on any information and opinions contained in this report or any information which is made available in connection with any further enquiries, notwithstanding any negligence, default or lack of care, is disclaimed.

The reports are not for public distribution. Reproduction or dissemination, directly or indirectly, of research data and reports of Crypto.com in any form is prohibited except with the written permission of Crypto.com. Persons into whose possession the reports may come are required to observe these restrictions.

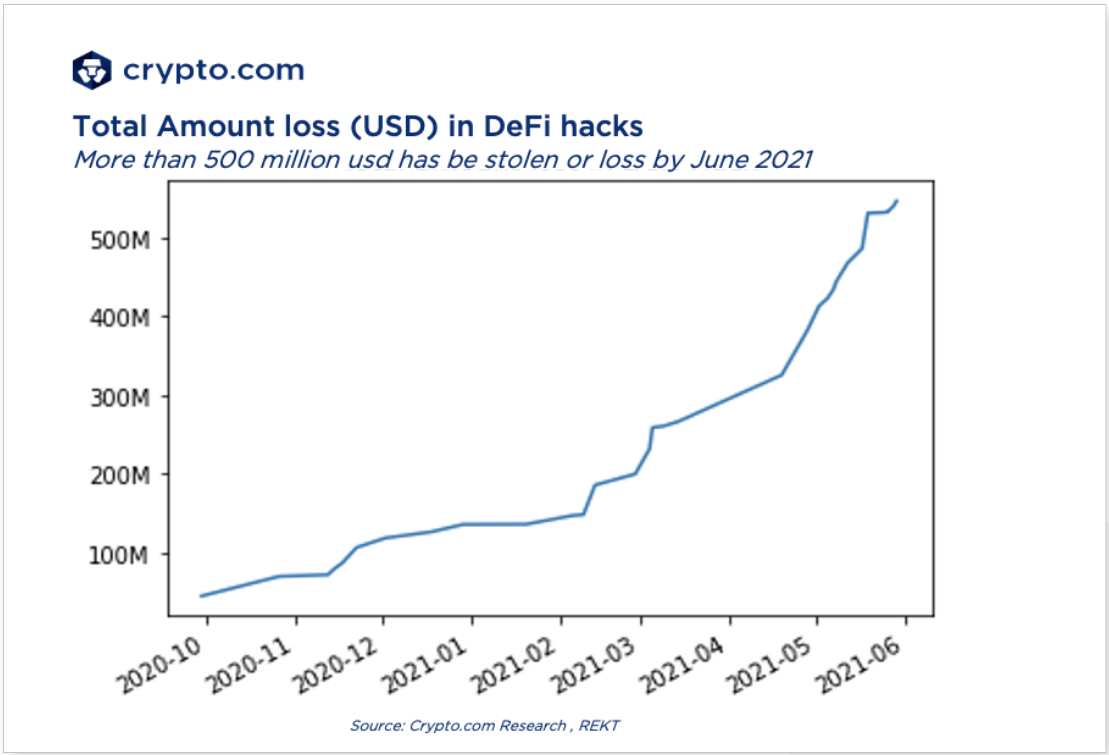
# Content

# Executive Summary

## Key Takeaways

- Decentralized Finance (DeFi) has taken the world by storm thanks to its boundless opportunities and applications. According to DeFi Pulse, the industry has exceeded $57 Billion in total value locked (TVL) as in June 2021.
- DeFi protocols have become appealing targets for hackers as the open and immutable nature of smart contracts has made hacking hard to prevent. More than $500 million worth of assets have been stolen in the last 8 months due to hacks and exploits. The growing number of attacks may slow down the mainstream adoption of decentralized finance solutions.
- This report has covered several common DeFi attack vectors and mitigations, including price oracle manipulation, reentrancy attack and the rug pull.
- Price Oracle manipulation using flash loans is the most common type of exploits in recent years. Malicious parties can use large amounts of unsecured loans to manipulate the assets' apparent prices on decentralized exchanges (DEX). If a protocol relies on one single DEX as the only price oracle, it will become vulnerable to the manipulation of price data.

# Introduction




Decentralized finance (DeFi) has taken the crypto space by storm since 2020 and has become the most popular application on various major blockchains such as Ethereum. Unlike traditional finance that completely relies on intermediaries like banks to process transactions, the open and decentralized nature of DeFi has offered users a way to perform financial activities such as trading, lending, borrowing etc. without middlemen using smart contracts. The total value locked (TVL) - a measure of DeFi transaction value has expanded by more than 14 times in the last twelve months with a total value of $57 Billion as in June 2020.

While DeFi has been gaining an increasing level of market growth in terms of both popularity and liquidity, the openness of DeFi has also led to frequent occurrences of security incidents such as hacks and exploits that always resulted in the loss of participants' funds. About a total of $500 million worth of assets were looted from DeFi platforms between 2020-2021 with an increasing rate as shown in the figure.

In this report, we will explore different major DeFi attack vectors that you should be aware of and understand how such attacks can wipe out millions from the protocols.

# Type of Attacks

## Price Oracle Manipulation

Oracle manipulation is the most common attack in the DeFi space where attackers manipulate the asset price data on decentralized exchanges (DEX) such as Uniswap. If the manipulated exchange is used as the only price feed by a DeFi platform, attackers can buy or sell a certain asset that is below or above the fair market price on that platform.

Since this kind of attack requires a large amount of capital, flash loans have been increasingly employed to attack vulnerable contracts as they allow anyone to access huge sums of capital at a very low cost. For example, in May 2021, the Binance Smart Chain protocol PancakeBunny has lost over 700,000 BUNNY and 114,000 BNB in a flash loan attack, causing its token to plummet by more than 95% and a total loss of over $200M.

Before we dive deeper into how such an attack was carried out, we need to learn some basic ideas about price oracle and decentralized lending.

### Price Oracle

In the DeFi world, price oracles are third-party services that allow smart contracts to receive external price data from outside of their ecosystem. Many DeFi protocols have used centralized price oracles and these feeds draw prices directly from asset pairs on a single decentralized exchange (DEX) such as UniSwap. Developers simply instruct the smart contract to query the oracle which returns the current value of a token quoted in a specified currency such as USD or ETH. If a DeFi protocol relies on price data from a single DEX, any changes in that DEX's price data - whether if it is the fair market price or not - are considered true and accurate by the protocol's smart contracts. Therefore, if an attacker can manipulate the asset price on that single DEX, inaccurate price data can be fed into all protocols which rely on that DEX as a price oracle, making them vulnerable to exploitation.

### What is Decentralized Lending?

In traditional lending, collaterals are forfeited in case of default. The amount of loan that you can borrow depends on the fair market value of your collateral, which is known or estimated by the lender.

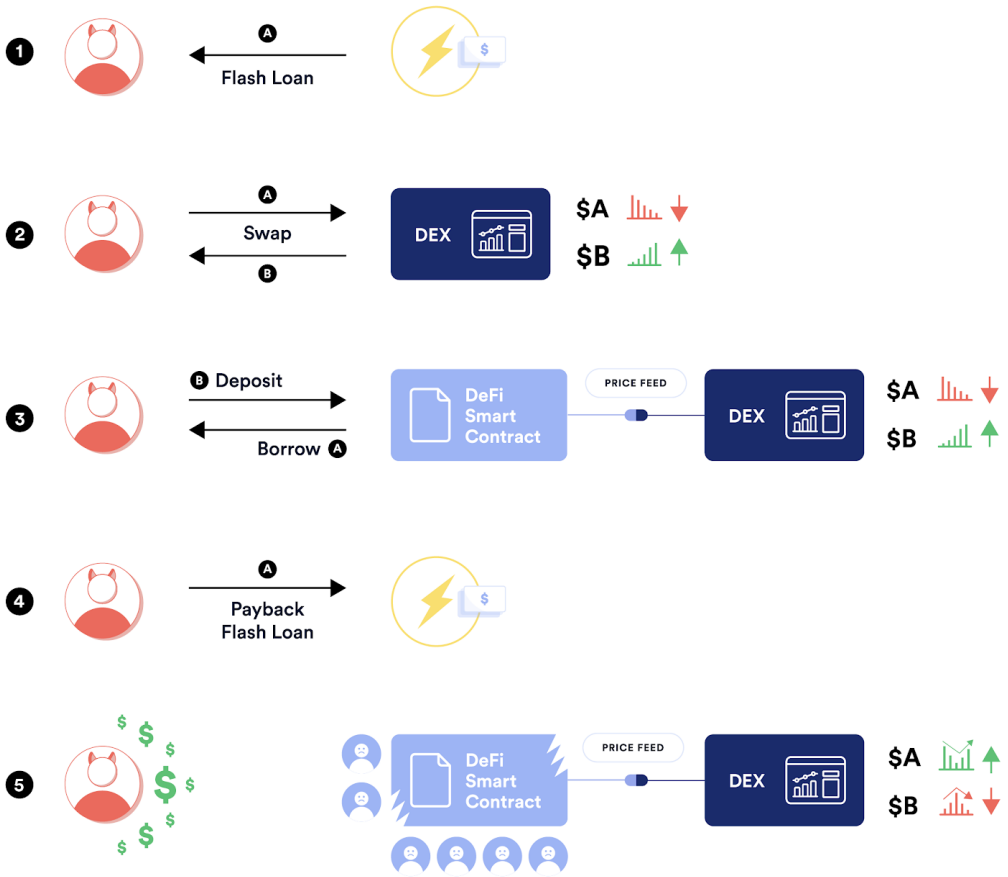
On the other hand, a similar process occurs in decentralized lending except that the lender is the smart contract, which is isolated from the outside world. To get the fair market value of whatever token you are using as the collateral, the smart contract is instructed to query an oracle. You will then be able to take out an under-collateralized loan and make a profit if the oracle returns the manipulated price data.

## Manipulating oracle using Flash Loan

Flash loan is a new type of unsecured loan developed by the DeFi lending protocol AAVE which is enforced by smart contracts. An unsecured loan refers to a loan that doesn't require any collateral. For example, you can borrow money from a bank if you have a good credit score. However, if the amount you intend to borrow is too large, the bank would require you to provide collateral such as a house or a car to mitigate the risk.

On the other hand, a flash loan allows users to borrow as much as they want with zero capital, given that they can repay the loan after doing something with the fund such as arbitrage. Otherwise, the transaction will reverse. For their low-risk, low-cost, and high-reward properties, flash loans are frequently used by malicious hackers.

## Attack Process




*Source: Uniswap*

A flash loan attack on a DeFi lending and borrowing protocol using a DEX-based price feed follows the following sequence:

1. Borrow a large amount of token A from a protocol that supports flash loans

2. Swap token A for token B on a DEX (lowering price of token A and increasing price of token B on the DEX)

3. Deposit the purchased token B as collateral on a DeFi protocol that uses the above DEX as its sole price feed, and use the manipulated pricing to borrow a larger amount of token A that should normally be impossible

4. Use a portion of borrowed token A to fully pay back the original flash loan and keep the remaining tokens, generating a profit off the protocol's manipulated price feed

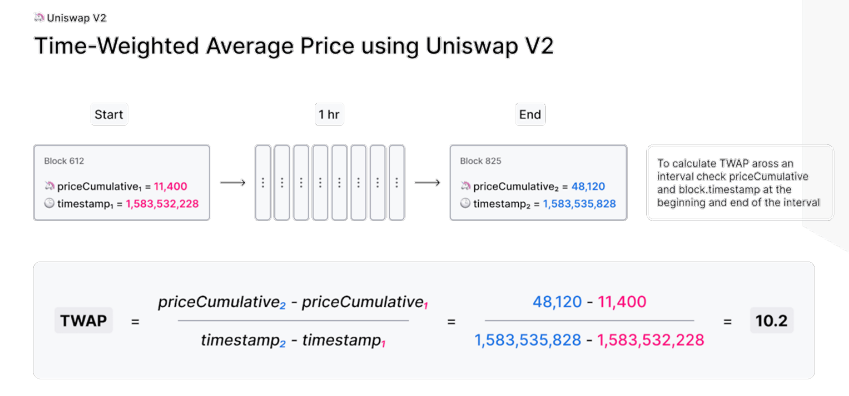
5. As the prices of token A and B on the DEX get arbitraged back to the true market-wide price, the DeFi protocol is left with an undercollateralized position (debt worth more than collateral), directly harming users such liquidity as providers of the pool

## Prevention

From the above example, we can see protocols that are using on-chain centralized price oracles such as a singular DEX have opened themselves up to flash loan exploits. When using a single on-chain exchange as price feed, the data of an asset is extremely limited since it only reflects the market status of a single exchange. Using a decentralized oracle like Chainlink would help reduce the risk as it is powered by a decentralized network of oracles that aggregates price data from multiple sources, such as Coingecko, BraveNewCoin, and more, to get the full market coverage. Therefore, a single flash loan transaction will not affect the price feed.

Uniswap V2 has also introduced multiple improvements for supporting the manipulation-resistant oracle. Instead of storing all the market prices in a block, the price feed will only measure the asset's market price at the beginning of each block before any trade. By doing so, price is harder to be manipulated as it is pre-defined by the previous block. Therefore, attackers need to manipulate the price at the end of the previous block and ensure that they can outperform the arbitrageurs, which is extremely challenging.

In addition, the last price of each block will then be added together to get the time-weighted average price (TWAP) for any desired interval. TWAPs can be used directly or as the basis for moving averages (EMAs and SMAs) as needed.




*Source: UniSwap v2*

# Reentrancy Attack

Reentrancy is one of the most well-known and devastating attacks faced by smart contract developers over the years. As the balance of a smart contract can be completely drained out when the attack is performed by hackers. By definition, a procedure is reentrant if its "execution can be interrupted in the middle, initiated over (re-entered), and both runs can complete without any errors in execution". Hence, it will put a smart contract into an "inconsistent state" and lead to vulnerabilities.

## What is Reentrancy?

Now imagine there is a poorly designed ATM that would only check the balance of your account when you remove your card. What will happen if you keep requesting a withdrawal? Eventually, you can empty all the money in the ATM as it would never discover that the withdrawal amount has potentially exceeded the balance in your bank until you have pulled out your card. This is the main mechanism of the reentrancy exploit which was used in the well-known DAO hack back 2016 (ref).

## Interaction Between Smart Contracts

Smart contracts aren't designed to send more money than the amount being held inside them. However, by exploiting reentrancy, hackers can effectively turn a smart contract into a poorly designed ATM. Before we dive deeper into how smart contracts can be exploited via reentrancy, we must first learn about some basic ideas of interactions between smart contracts.

For simplicity, we will only cover some basic actions of a smart contract in this section. When developing smart contracts using Solidity, two types of functions are mainly called to perform basic tasks:

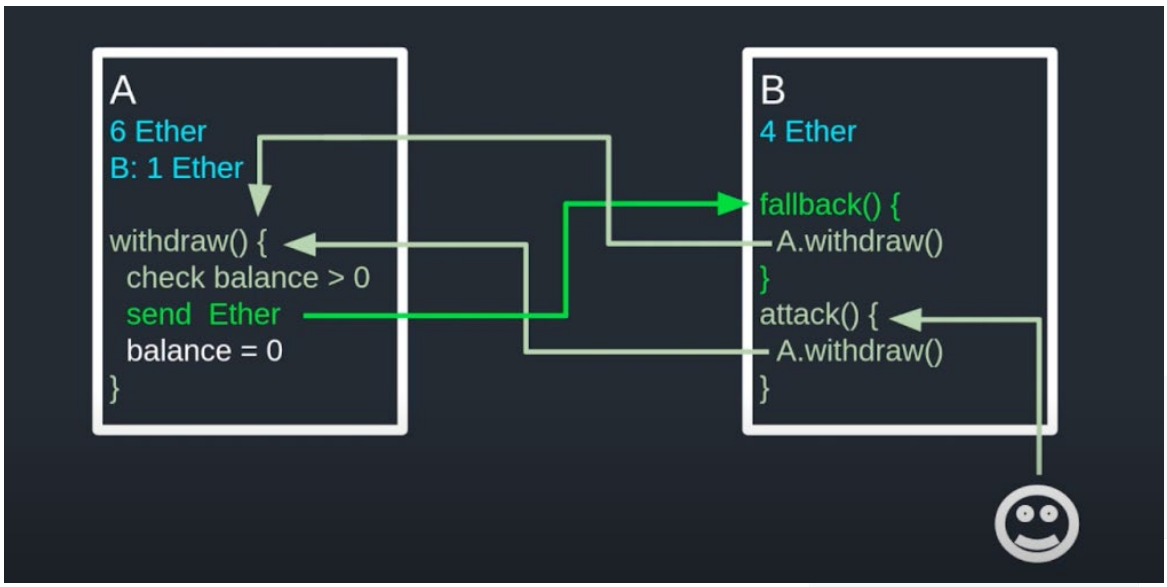
External Call Function - functions that allow contracts to interact with each other and further invoke a function from within the same contract or from a different one, for example

1. Requesting withdrawal from another contract
2. Receiving Ether from another contracts (Payable)
3. Transferring Ether to another contracts (send(), transfer() and call.value() )

Internal Call Function:

1. Internal auditing (i.e Checking balance)

Every Ethereum smart contract contains a default fallback function that is fully customizable and can be overridden by the developer with any arbitrary code. For example, if it is overridden as payable, the smart contract can accept Ether and the function can be executed whenever Ether is transferred to the contract. It can be designed to request a withdrawal from a target contract after an ether is received.

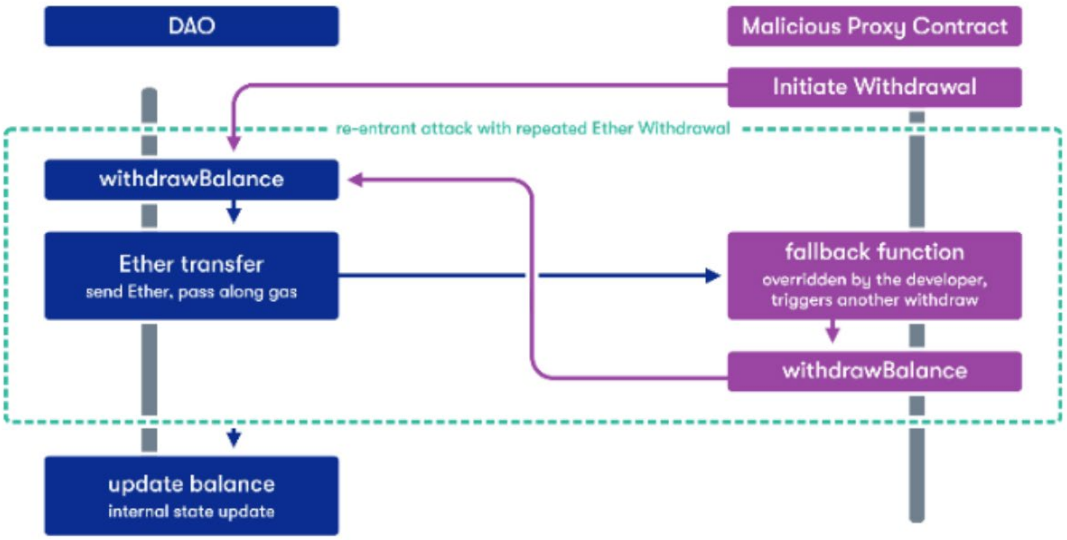



Now let's say there are contracts A and B as shown above, where A is the vulnerable contract and B is the contract used by the attacker. Below is the following sequence of actions:

1. Contract B would ask for a withdrawal from A by calling the attack function
2. A calls the withdraw function and send 1 Ether to B after confirming B has more than 0 Ether in the contract
3. The transfer from A to B has triggered a fallback function.
4. The fallback function would ask A for another withdrawal.
5. The transfer from A to B has triggered a fallback function again
6. The fallback function would ask A for another withdrawal... ( the process repeats itself )

Since the real balance of B in contract A will never be updated until the withdrawal loop has finished, the hacker can call the withdrawal function recursively until A is completely drained out).

## The DAO Hack




The DAO is a popular decentralized investment fund based on smart contracts. In 2016, the DAO smart contract accumulated over $150,000,000 (at the time) of Ether. On June 17th 2016, it was hacked and 3.6 million Ether ($50 Million) were stolen using a cross-function reentrancy attack. The Ethereum Foundation issued a critical update to roll back the hack. This resulted in Ethereum being forked into Ethereum Classic and Ethereum. The hack has taken the advantage of Ethereum's fallback function to perform the attack.

## Prevention

The reentracy attack can be avoided in some generic ways. Solidity supports three ways of transferring Ether between wallets and smart contracts, namely, send(), transfer() and call.value(). These methods only differ by the amount of gas limit when passing the function. Using the functions send() or transfer() instead of call.value() would not allow for recursive withdrawal calls due to the low gas allowance. On top of that, to avoid any recursive withdrawal, a contract should update its internal state (e.g., user balance) before making any transaction.
Another possible approach is to whitelist specific external contracts to restrict any interaction between unknown contracts.

# The Rug Pull

Rug pull is a popular 'exit scam' in the DeFi ecosystem widely used by many malicious developers as this type of attack is technically very simple but

extremely profitable. A rug pull refers to an event when a malicious team suddenly deletes all the traces on social media after abandoning the project and running away with investors' funds. For instance, the team can take away all the buying support of a liquidity pool in a decentralized exchange (DEX). Scammers have full control over the project protocol, so they can create and list tokens without audit, and even have the right to remove the liquidity.

## What is a Rug Pull

Scammers can create a token on a DEX such as Uniswap and Sushiswap and pair it with a leading cryptocurrency such as Ethereum. Then they tend to promise a ridiculously large annual percentage yield (AYP) to retail investors such as yield farmers. Once investors have swapped their ETH to the newly created token and enough funds have been locked into the smart contract, the developers can drain the liquidity pool and run away with the funds. The sudden loss of liquidity will result in a big sell-off of the token as token holders want to save their profit.

## Case 1: Meerkat Finance

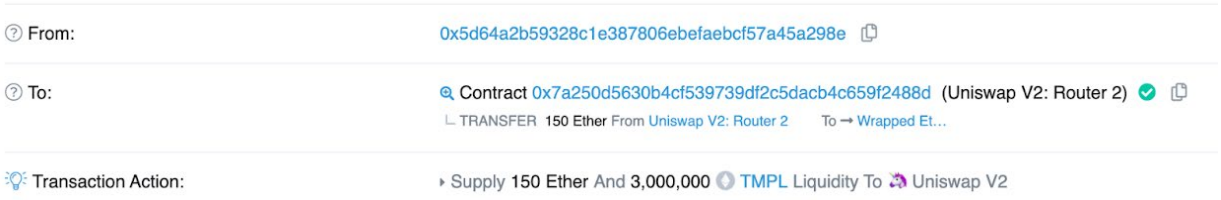
Meerkat Finance is a yield farming protocol on Binance Smart Chain (BSC), which was rugged within a day after going live in March 2021 for about 13 million BUSD and 73,000 BNB, totaling around $31m USD. After the incident, Meerkat's website and Twitter account were both taken down. It was found that the deployers had upgraded the contract to grant themselves the ability to retrieve the assets from the pool shortly before the attack.

## Case 2: TRUAMPL (TMPL) token


*Source: Etherscan.io*

Another example is the famous TMPL rug pull that happened in August 2020 when the creator of the owner of the contract drained out the entire liquidity which contained 154 ETH & 2,926,099 TMPL tokens in just 40 minutes after the token sale went public as shown in the above figure.

The sequence of events can be summarised as follows:




*Source: Etherscan.io*

1. Scammers (0x5d64a2b59328c1e387806ebefaebcf57a45a298e) created a TMPL token pool contract (0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D) on Uniswap and supplied 150 Ether and 3,000,000 TMP liquidity to hype it up

2. Users swap their valuable tokens such as ETH for TMPL and drive the pool participation rate and TMPL price up

3. Scammers remove liquidity from the platform and takes gains on the valuable token

## Prevention

To protect yourself from being rug pulled, you should research the project before putting your money in it. Finding out the purpose and the backer of the project would help verify the team's credibility. In particular, you can look up their track records, social media traces and experiences, etc. Besides, block explorer tools like Etherscan can be used to verify the number of token holders. If the token has very few wallet holders and only be listed on DEX platforms, there is a high possibility of an exit scam.

# Summary

## Key Takeaways

- Decentralized Finance (DeFi) has taken the world by storm thanks to its boundless opportunities and applications. According to DeFi Pulse, the industry has exceeded $57 Billion in total value locked (TVL) as in June 2021.
- DeFi protocols have become appealing targets for hackers as the open and immutable nature of smart contracts has made hacking hard to prevent. More than $500 million worth of assets have been stolen in the last 8 months due to hacks and exploits. The growing number of attacks may slow down the mainstream adoption of decentralized finance solutions.
- This report has covered several common DeFi attack vectors and mitigations, including price oracle manipulation, reentrancy attack and the rug pull.
- Price Oracle manipulation using flash loans is the most common type of exploits in recent years. Malicious parties can use large amounts of unsecured loans to manipulate the assets' apparent prices on decentralized exchanges (DEX). If a protocol relies on one single DEX as the only price oracle, it will become vulnerable to the manipulation of price data.

# References

1. Werner Vermaak (2021 May), What Are Flash Loan Attacks? Retrieved from https://coinmarketcap.com/alexandria/article/what-are-flash-loan-attacks#toc-why-flash-loan-attacks-are-common-in-defi

2. Yixin Cao∗ ,Chuanwei Zou, and Xianfeng Cheng Shanghai Wanxiang Blockchain Inc (2021 Feb) , Flashot: A Snapshot of Flash Loan Attack on DeFi Ecosystem. Retrieved from https://arxiv.org/pdf/2102.00626.pdf

3. Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais (2020 Mar) .Attacking the DeFi Ecosystem with Flash Loans for Fun and Profit. Retrieved from https://hackingdistributed.com/2020/03/11/flash-loans/

4. Everett Muzzy (2020 May), Security Risks in Ethereum DeFi. Retrieved from https://consensys.net/blog/codefi/security-risks-in-ethereum-defi/

5. Known Attacks Retrieved from https://consensys.github.io/smart-contract-best-practices/known_attacks/

6. Adelyn Zhou (2020 NOV),Flash Loans Aren't the Problem, Centralized Price Oracles Are. Retrieved from https://www.coindesk.com/flash-loans-centralized-price-oracles

7 .Rekt Retrieved from https://www.rekt.news/leaderboard/

8 .Quantstamp Lab (2019 Aug), What is a Re-Entracy Attack? Retrieved from https://quantstamp.com/blog/what-is-a-re-entrancy-attack

9. Ivan on Tech (2020 Dec), DeFi Deep Dive – Explaining DeFi Attack Vectors and Prevention. Retrieved from https://academy.ivanontech.com/blog/defi-deep-dive-explaining-defi-attack-vectors-and-prevention#:~:text=DeFi%20Attack%20Vectors%20Price%20Oracle%20Attacks&text=By%20doing%20so%2C%20they%20can,become%20vulnerable%20to%20the%20exploit.&text=Price%20oracles%20are%20what%20protocols%20consult%20for%20asset%20price%20information.

10. Liesl Eichholz (2020 Nov) ,DeFi Attacks: Flash Loans and Centralized Price Oracles Retrieved from https://insights.glassnode.com/defi-attacks-flash-loans-centralized-price-oracles/#:~:text=Price%20Oracle%20Attacks,is%20the%20price%20oracle%20attack.&text=This%20action%20exploits%20a%20vulnerability,which%20can%20then%20be%20arbitraged

11. Uniswap v2 Retrieved from https://uniswap.org/docs/v2/core-concepts/oracles/

12. Etherscan Retrieved from https://etherscan.io/

crypto.com

**e.** contact@crypto.com