QuillAudits
by Quillhash

DeFi Security Synopsis 2020

Contents

Introduction

DeFi is being considered as the new face of finance. Showcasing an unrealistic growth over the past few years, DeFi has attracted millions of people, offering better investment opportunities, higher returns, transparency, and unmatched trust. In the long run, DeFi is bound to become a mainstream system for accessing financial services.

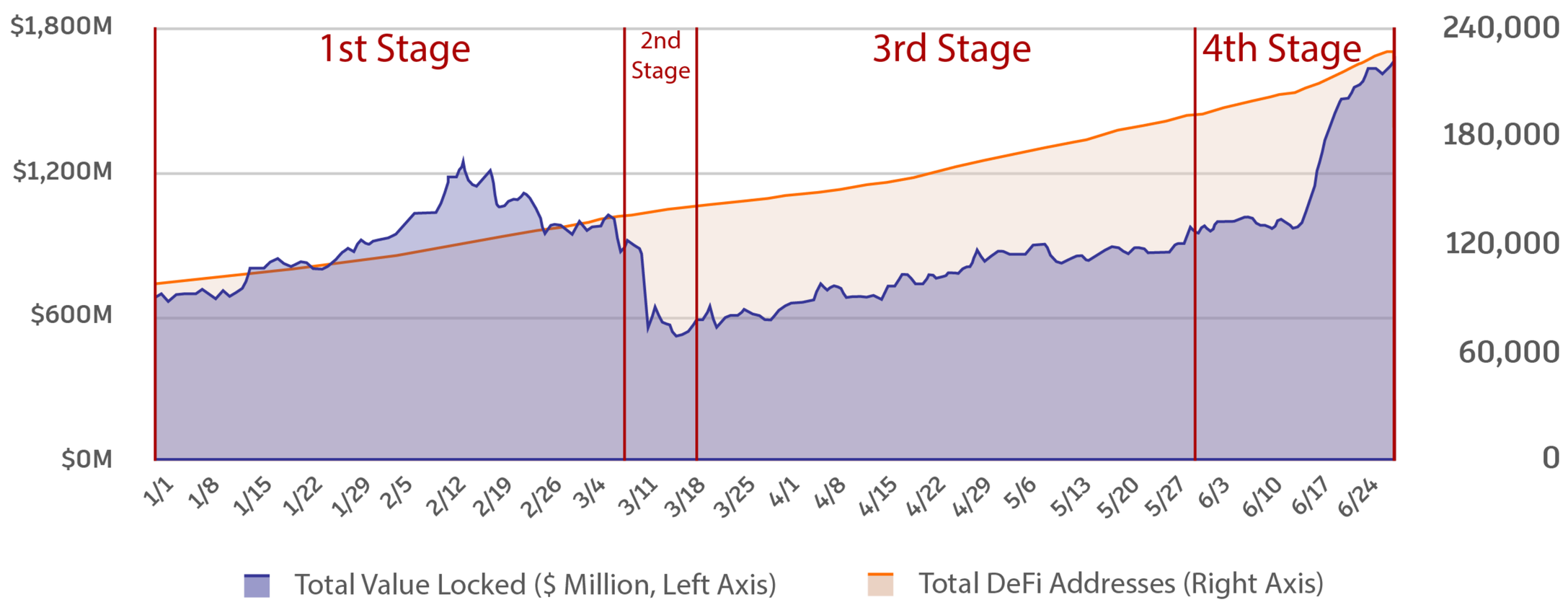
However, as every coin has two sides, DeFi is moving towards a disastrous future because of the potential risks that remain unnoticed across the industry. A proper understanding of these risks is essential to define the path for a better, secured future for the world of finance.

In this report by QuillHash, we take a glance at the existing DeFi market and how it has been affected over the years due to the negligence of risks associated with it and then we discuss about the different possible attacks on DeFi protocols as well as few of the methods that can ensure greater security in the DeFi ecosystem.

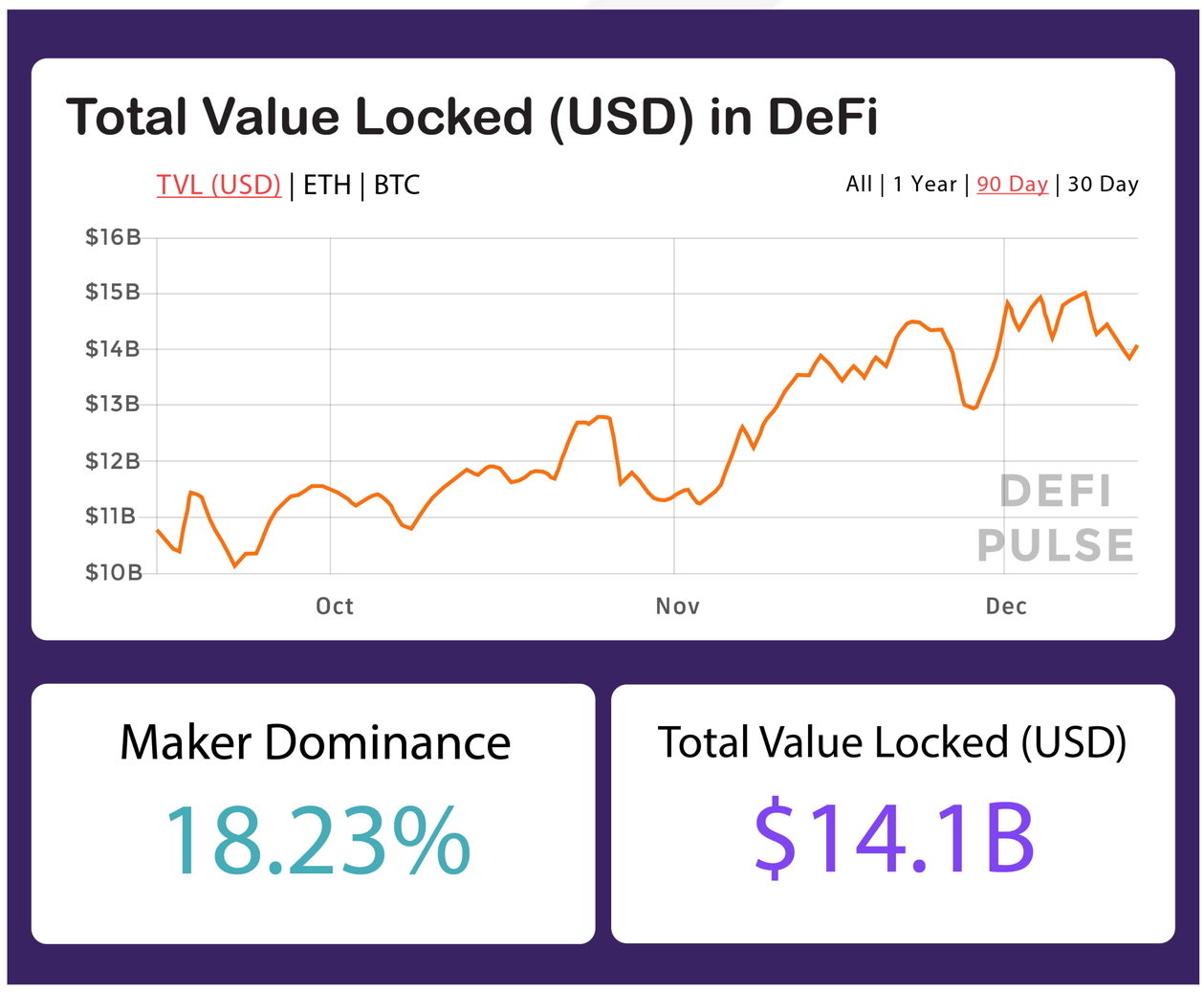
It should be noted that there are several other risks associated with the DeFi ecosystem which are not included in this report. Risks such as economic incentive risk, financial illiteracy risk, and regulatory risk require the opinion of an expert. Moreover, as DeFi is still in a nascent state, there is a high probability that more risks will arise over time as DeFi evolves further. Therefore, a business in the DeFi space needs to be in a constant touch with the ever-changing trends of the DeFi ecosystem and continuously evaluate the new possible risks.

The Current Scenario

As the world got hit by COVID and the economy suffered, DeFi protocols have not only survived but flourished in these times. The total value locked in different DeFi protocols has undergone massive growth. It skyrocketed by 380% from the end of Q2 2020 and topped $10 billion in September 2020.

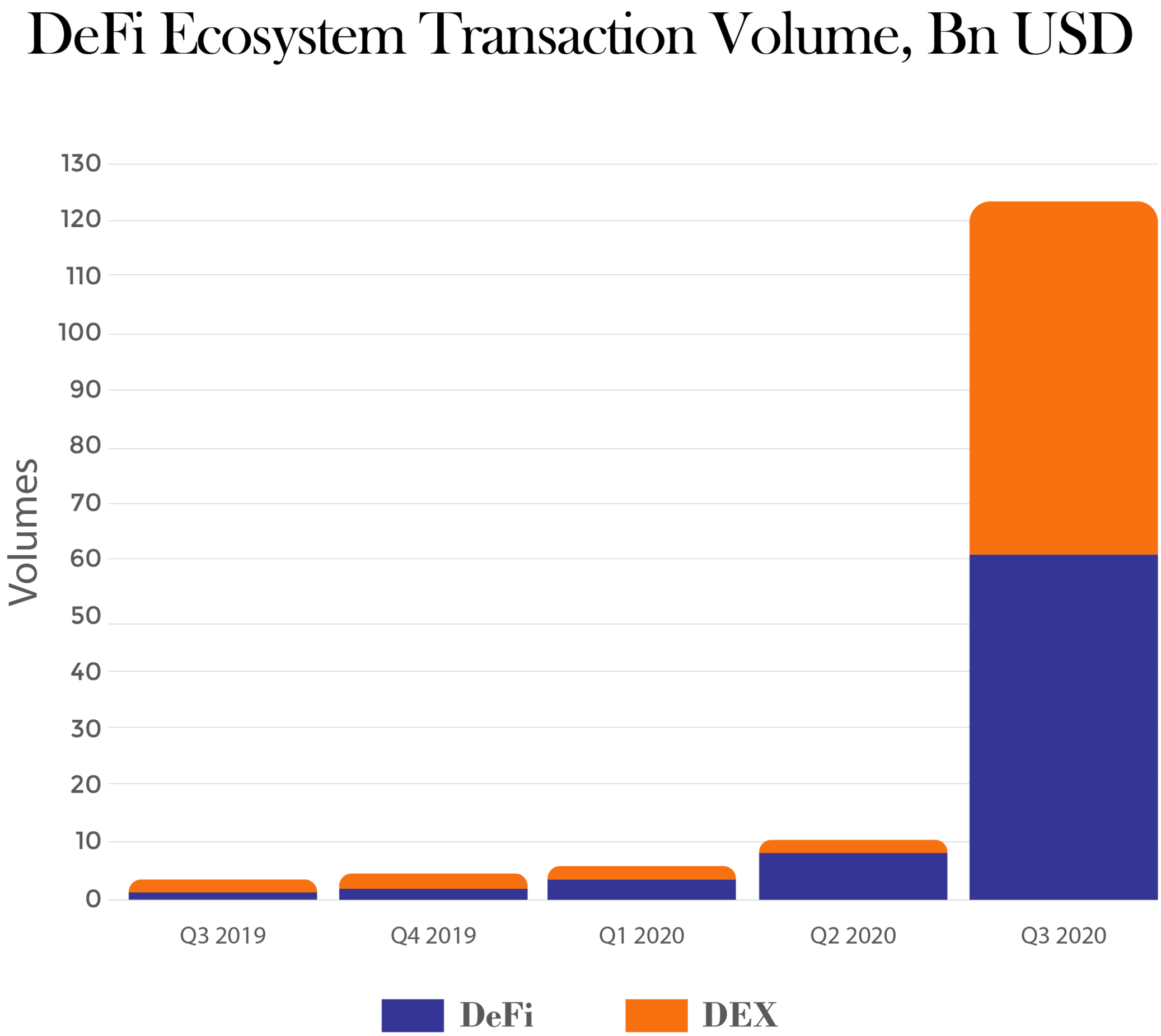


There has been an increase of $1 billion in the total value locked in the DeFi ecosystem which equals an increase of 147% during H1 2020. The growth of DeFi can be monitored through the most popular metrics “Total Value Locked” in USD. In June of 2020, the Total Value Locked in USD for the DeFi ecosystem stood at 1 billion and jumped to more than 1.6 billion in just a matter of 2 weeks, showing unfathomable growth.

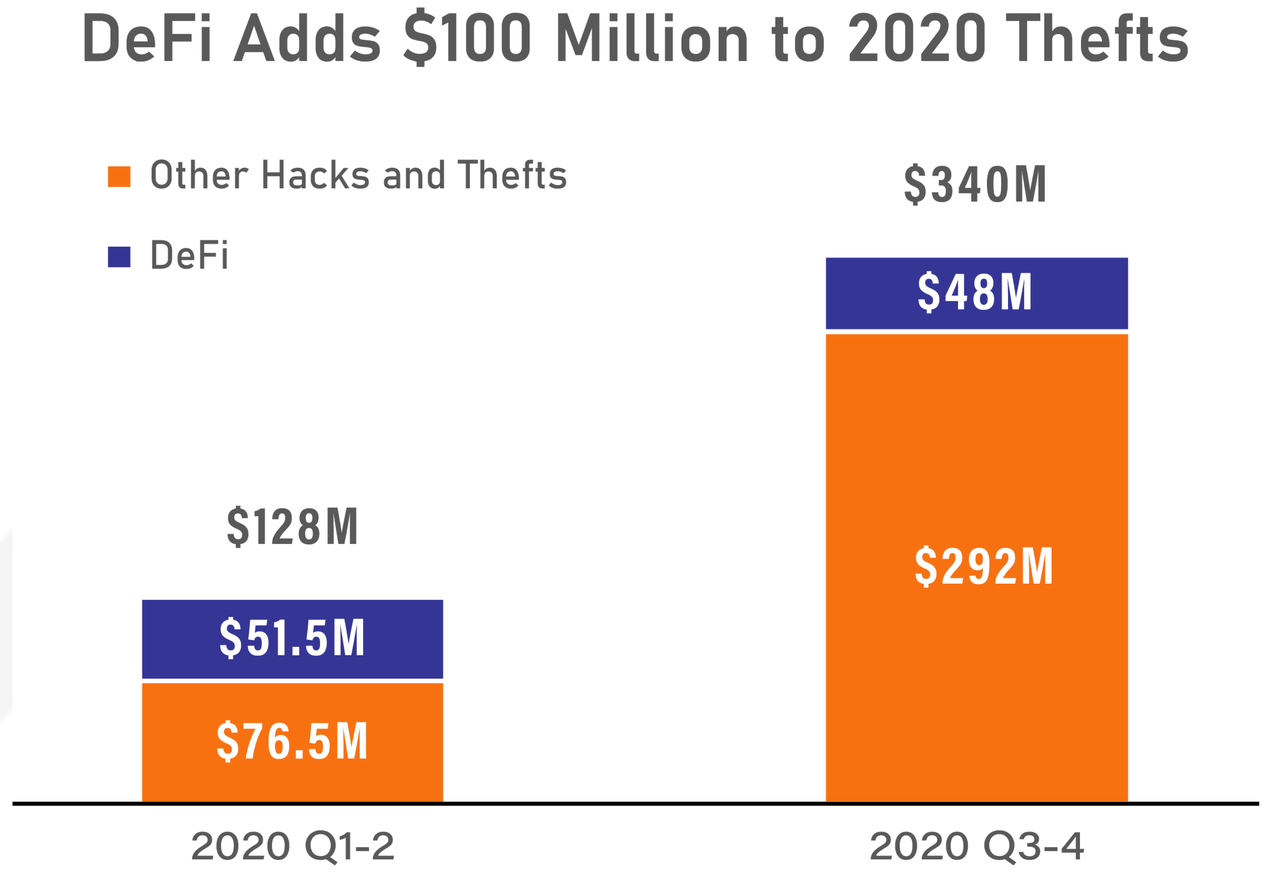


At the forefront of this growth is Ethereum. Being the most widely used Blockchain for building DeFi applications, Ethereum has been a key player in the growth of DeFi. 96% of the transaction volume in the DeFi space is happening over the Ethereum Blockchain and it also accounts for more than 57% of the daily active wallets. Today, there is over $10.4B of TVL located within the Ethereum DeFi ecosystem

Therefore, Q3 2020 is being regarded as the “best quarter for the DeFi ecosystem.”



However, a recent report released by blockchain and crypto security analytics firm CipherTrace shows that the DeFi sector resulted in $100 million lost in thefts and hacking attacks this year alone. Currently, the Total Value Locked (TVL) on DeFi protocols have hit a new all-time high of $13.03 billion after experiencing precipitous growth throughout the year. This has come with consequences as well, seeing as the sector is responsible for 21% of all hacks and theft this year.



Therefore, the DeFi ecosystem needs rigorous auditing as DeFi audits are the only way to ensure that such hacks are just a thing of the past and not the future of DeFi. Before getting into what audits are, why they are needed, and how DeFi applications can benefit from it, let us look into what is DeFi and what role does smart contracts play in the DeFi ecosystem.

What is DeFi

DeFi or Decentralized finance is a means to provide financial services to the people without the intervention of a third party such as a bank. DeFi is a crucial layer of financial products in the modern age and proves beneficial for the people involved in the blockchain industry.
Apart from the high investment potential, DeFi is gradually becoming the new face of finance. Decentralized finance is challenging banks and bringing up people to be self-sovereign.

What are Smart Contracts

Smart contracts are computer protocols that are the fundamental driving force behind the Decentralized Financial System. These contracts are nothing but lines of code that ensures the credibility between entities dealing through a digital medium, without the intervention of a third party.

Over the past few years, the term “Smart Contracts” has gained tremendous popularity by becoming the de facto standard for digital agreements. However, smart contracts are not a revelation but their origin dates back to 1994 when a person named Nick Szabo proposed the concept of smart contracts and created the first smart contract in 1998 to digitalize the agreement terms for transaction of his virtual currency “Bit Gold”.

According to Szabo, smart contracts would streamline the complex process of agreement between two parties without involving any mediator. The roles of the mediator such as validating, authenticating, and verification will be completely based on code written in form of a contract.

There are a number of Blockchains that support smart contracts. Tron, Tezos, EOS, and Algorand are the most common examples of Blockchains which allow the creation and management of smart contracts to build decentralised applications for various purposes. However, Ethereum is the most popular Blockchain platform for this purpose. Founded in 2016, Ethereum is the first Blockchain platform to

facilitate smart contracts. Solidity is the native programming language to write smart contracts in Ethereum and it is similar to javascript.

Smart contracts are the reason why Ethereum is the most popular Blockchain platform especially for the world of decentralised finance as DeFi is a means of providing permissionless financial services to the users over the Internet and smart contracts on Ethereum are most commonly used to create the DeFi applications.

It has become a global alternative against traditional banking systems where users can enjoy various financial services such as loans, savings, trading, insurance and many more.

Role of Smart Contracts in DeFi

DeFi or Decentralized Finance, also known as Open Finance, is considered as the long-pending evolution of the financial sector. The unprecedented growth of DeFi since the past few years has resulted in a boost to the global economy, enabling more inclusivity and trust in the system.

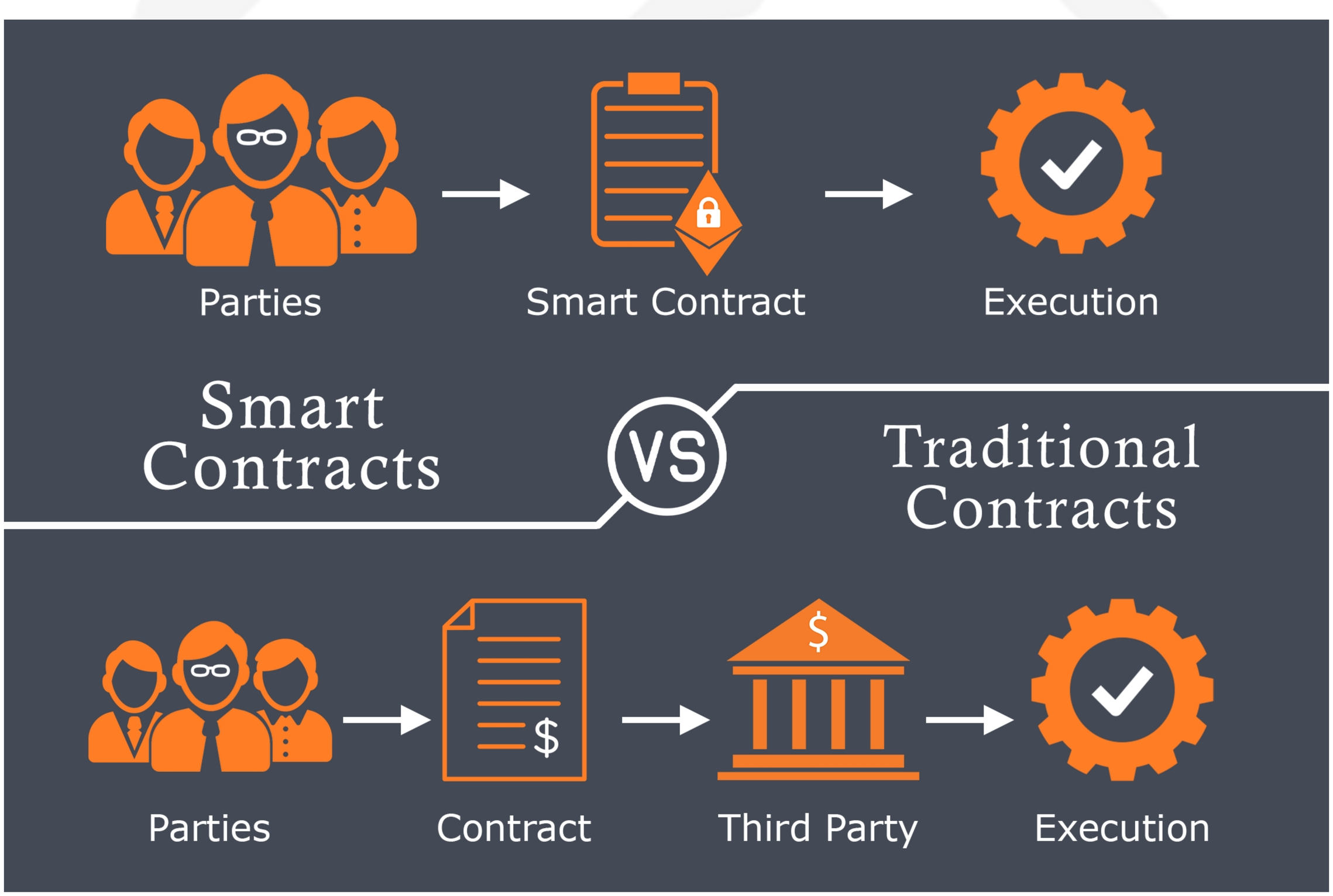
However, the growth of DeFi is still believed to be just the beginning as experts believe that it is an underdeveloped sector whose potential is unfathomable. It has evolved as a unique and dynamic industry that bears great significance for a considerable percentage of the world population.

The main reason for the inclusion of smart contracts in the DeFi ecosystem is to enhance the transparency and credibility of the entire lifecycle of the agreement process.

Smart contracts have played a significant role in the growth of DeFi as the incorporation of smart contracts is the reason behind the appropriateness of DeFi in various use cases. For instance, using decentralised finance to create an ecosystem which synchronizes the needs of buyers and sellers without any third party.

Here, smart contracts will act as the terms of agreements and enforce the predefined, mutually agreed upon rules. Other products of the synergy between DeFi and smart contracts are true digitization, enhanced security, safeguard from external factors such as bribe, unmatched speed, improved accuracy, lower costs such as transaction fee, and utmost transparency.

When it comes to comparing smart contracts with traditional contracts, the main difference is that the governance and execution of smart contracts is not only fast but it is self-dependent. Therefore, the process of smart contracts is entirely digital, which is beneficial for both entities participating in the agreement. This digitalization of the whole process results in saving a tremendous amount of time, energy, and resources. Moreover, smart contracts with DeFi also become a highly secured system which stores the various records providing more auditability and accessibility.

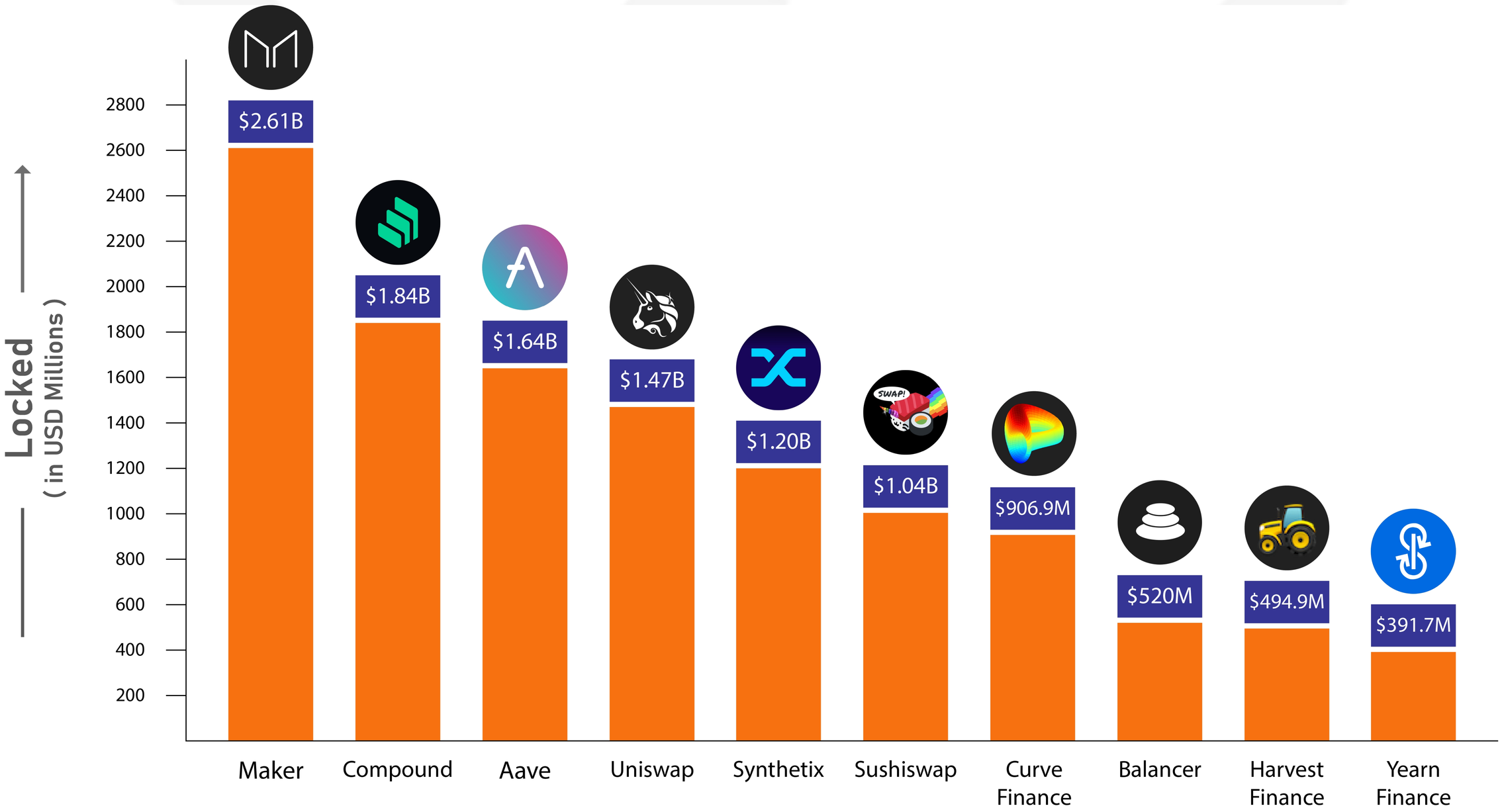


Among many benefits, security remains the most crucial feature of having a DeFi system with smart contracts.

However, with the rapid developments in the DeFi sector, challenges are bound to arise. The need for speed, not only in the process but in terms of the time for going to the market, has made loopholes a common thing among many DeFi projects.

The accuracy of smart contracts is a crucial factor in the popularity and growth of DeFi and as this accuracy is leveraged due to the need for speed, DeFi is bound to face criticism. The completely automated smart contract system should promise the service of exchanging information with efficiency and accuracy in order for DeFi to become the new face of finance.

Top DeFi Projects by Market Valuation



The Vulnerabilities of the DeFi Ecosystem

According to the CipherTrace report, 20 percent of hacks in 2020 belonged to the DeFi ecosystem which amounts to $98 million. Combining this with the fact that there were very few attacks on DeFi in 2019, it can be observed that these attacks have accompanied the exponential growth of DeFi.

Therefore, one possible assumption is that vulnerabilities have always existed in the DeFi ecosystem but only when it gained tremendous popularity, these vulnerabilities are being exploited by the hackers. The major reason behind this exploitation is that the smart contracts that govern the DeFi ecosystem are unaudited. Facing several challenges such as the lack of a standard audit or quality audit, DeFi smart contracts are becoming a hot zone for attacks and 2020 is a proof of that.

The need for DeFi Smart Contracts Auditing

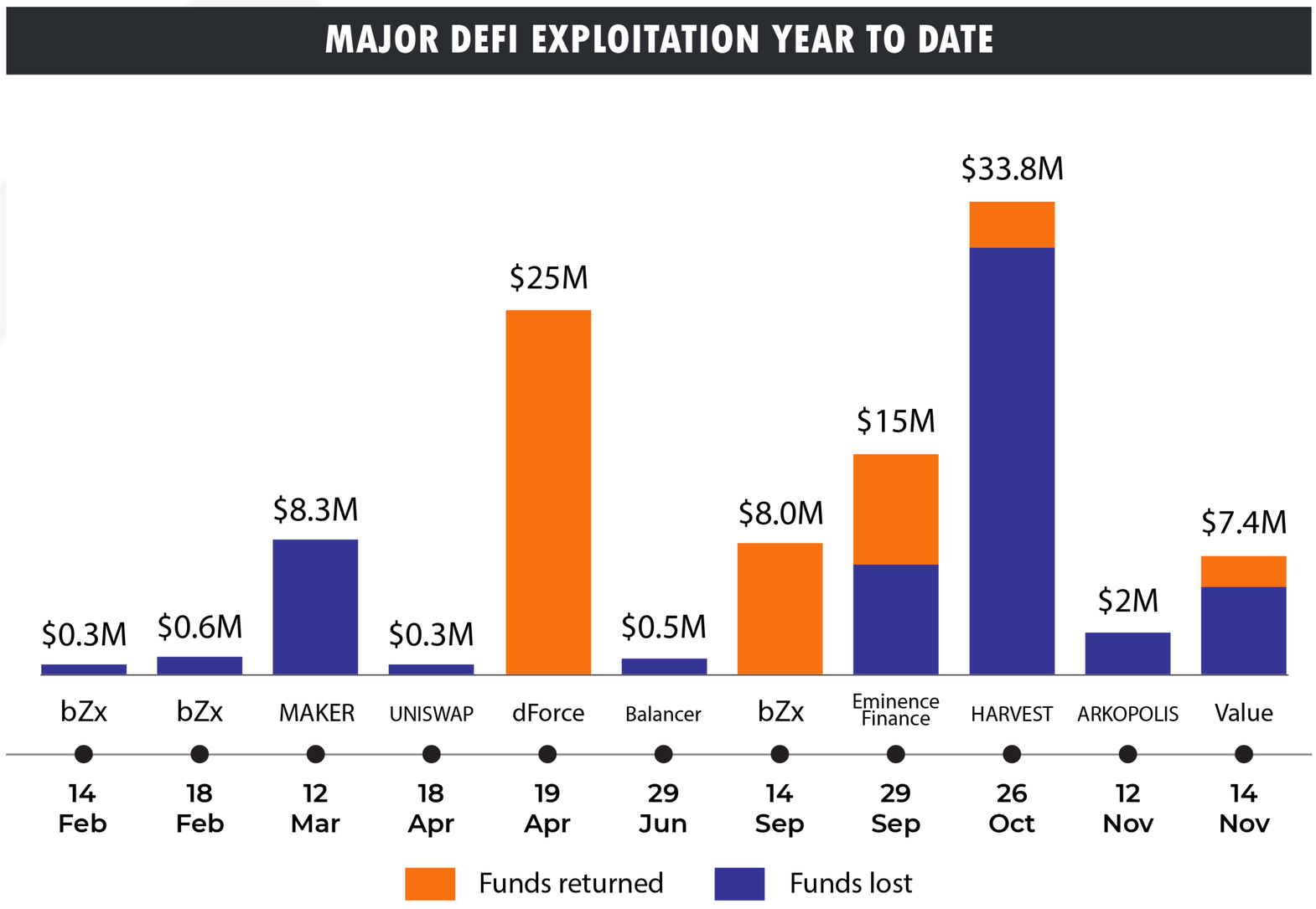
With a significant dependency on smart contracts, DeFi audits become crucial where a third party reviews every line of code and helps to identify the bugs and bottlenecks.

If left unaudited, the DeFi contracts may result in setbacks that include loss of funds and manipulation of the system. Sometimes, it may also lead to the shutdown of the company. Therefore, it becomes necessary to check the quality of the team auditing your DeFi contract and get acquainted with their process of auditing.

Let us examine the recent DeFi attacks which showcase the increasing magnitude of attacks over the past few years. This increasing magnitude serves as a proof that the lack of quality audit is leading us to a dangerous future of DeFi.

Recent DeFi Hacks

2020 was the year of tremendous growth for DeFi but it also proved to be the year which saw most DeFi hacks. Big names like bZx Protocol, Harvest, and Maker experienced big hacks. The Maker platform lost $ 8.3 million for the attack on 12 March, the Acropolis lost $ 2 million on November 12, and the Balancer platform lost half a million dollars in an attack on June 29. bZs, which claimed to be the strongest open finance protocol, lost 300 thousand dollars on February 14th and 600 thousand dollars on February 18th. It was again attacked on September 14th and lost $ 8 million but managed to get all of the lost funds back.



It has been reported that all these attacks occur due to a faulty code in smart contracts.

The greatest loss was suffered by Harvest Finance. By targeting the protocol's liquidity pools and carrying out an arbitrage attack, the hacker stole $ 33.8 million on October 26 in a hack that lasted just 7 minutes. $ 2.5 million out of the stolen $ 33.8 million were returned by the hacker afterwards.

Another DeFi platform that joined the club of being hacked this year is dForce. Losing $ 25 million in a hack attack on April 19th, the security of the platform has been put to question.

On September 29, the Eminence Finance Project experienced a $15 million loss due to a hack. The project backed by the Yearn. finance founder Andre Cronje then got a return of $8 million fund NFT game ecosystem. Another project in the queue is Value protocol as it lost $ 7.4 million in an attack on November 14. Just a few days later, on 21st November, Popular decentralized finance (DeFi) protocol Pickle Finance lost half of its value in a hack of $19.7 million in DAI, a decentralized stablecoin pegged to the U.S. dollar, from a Pickle wallet.

These DeFi attacks beg the question of why capital is being dumped in DeFi smart contracts that have not undergone the proper auditing. In the following sections, we will examine what is a smart contract audit, what type of DeFi attacks are possible without these audits and what are the benefits of smart contract audit.

What is a Smart Contract Audit?

Being decentralized and operational via piece of code it becomes highly crucial to have periodic audits of the smart contracts to find potential bugs. However, these audits not only help in finding any bugs but also the vulnerabilities and possible bottlenecks present in the system and also creates a sense of credibility and faith in the minds of users. A smart contract Audit ensures an uninterrupted functioning and protection for the asset which is stored in the smart contract.

Smart Contract auditing is done by an unbiased third party, who reviews the whole code, line by line, and identifies the potential loopholes.

Why does a Smart Contract need Auditing?

Frequent Auditing of the Defi applications help in ensuring their performance, stability and protection against various hacks. The most recent example is the Flash Loan attack on 14th November, 2020 where Value DeFi's MultiStables vault suffered a net loss of $6 million.

The different DeFi attacks are:

1. **Nested Bots**

   Unlike any financial institution, smart contracts can be bought 24*7. This anytime feasibility has brought in a threat of nested bots. These bots are automated pieces of code that work according to the wish of the person controlling them. They are fast and can buy/sell within milliseconds which allows the person to manipulate the market.

2. **Phishing Flash Loans**

   Flash loans are like instant loans that can be availed without any collateral. Essentially, an attacker can use flash loans to borrow, deposit, and again borrow a large number of tokens. This practice allows the hacker to artificially manipulate the price of a single token on a single exchange like Uniswap or Curve.

3. **Oracle Manipulation**

   Oracles are the third-party services that are responsible for providing data and information outside of the blockchain. The data provided by these services acts as a real-time data for the smart contracts which is used for several purposes such as data regarding the rate of a currency can be used by a smart contract to perform any function. However, using such third party services can cause conflicts as the data from these services is directly fed to smart contracts where the same data is processed without any verification. Thus any latency or manipulation in data supplied from the Oracle can create a point of error in the entire system.

4. **Ramping Market Network Congestion**

   A network congestion or delay in a smart contract system even for a millisecond can give rise to dangerous results for the user. Some organizations, for their own foul purposes, lead to network congestion so that the overall rates can fluctuate and they can be benefited.

5. Token Attacks

In the past couple of years, tokens have become increasingly popular among DeFi applications as they help create an ecosystem to facilitate transactions and exchange of services. However, with their popularity, token have attracted a number of possible attacks mentioned below:

- Token inflation

  In the crypto market, less popular assets are generally low in price which means that less people are utilizing these assets and their value remains low. However, some people may form groups and buy these assets collectively leading to significant popularity on social media. This leads to an overall increase in asset price over a short period of time. The same group of people may sell their tokens once the price increases significantly. After they sell their tokens, the price returns back to normal and the people who have bought the token due to its short term popularity suffer heavy losses. This is known as token inflation.

- Circulating supply dump

  As the prices per asset increases, there are some foul players who sell off their respected owned contracts. Due to such collective selling, a supply dump is created. As a result the foul players get immensely benefited as the price of token comes down considerably and they again buy the token with the profits they have made from the same token.

6. Single-function Reentrancy

Single-function Reentrancy occurs when the attacker identifies a vulnerable function in the smart contract and uses that function recursively. This is one of the most common types of attacks. It occurs when a function in the code makes an external call to some untrusted contract. Before this external call is completed the attacker calls the function recursively and brings the system to a halt.

7. Cross-function Reentrancy

The cross-function reentrancy is a bit more complex than single-function reentrancy and becomes difficult to detect. Essentially, cross-function reentrancy occurs when a function shares its state with another function which holds importance for the attacker.

8. Multiple Contract Reentrancy

- Wash trading

  Wash trading is a market manipulation trick usually done by traders and brokers to buy and sell the assets. This kind of fake trade is usually done to increase liquidity. According to reports, it has been found that about 70% to 90% of bitcoin trade is done through wash trading.

- Quote stuffing

  Quote stuffing is a process by which the cryptocurrency holders buy a large amount of assets and then sell their own assets when prices per token inflate and cancel their buy again within seconds. When the cryptocurrency holders buy a large number of assets, the trading systems of other contenders sense in the demand because of which token prices inflate. By the time prices are higher, the notorious character sells off their assets and cancels down the buy order of assets, gaining the extra profit in the pocket. Such a scenario takes place within seconds.

9. Front Running

Front running attacks are when the central parties are replaced by some untrusted third parties with the same functionality. Its types are:

- Displacement attack

  It is a kind of attack, when the user wants to do the transaction but the attacker steals the request and does it in his name before the actual user.

- Insertion attacks

  Transactions carried out over Defi applications change the state of the contract. If a legitimate user does a transaction after the attacker has changed the state of contract. Then the transaction by the authentic user is done overpriced, creating a profit threshold for the attacker without even having the hold of the assets.

- Suppression attacks

  This type of attack suppresses the role of authentic users in the transaction. The foul player tends to delay the legitimate transaction. After waiting for a while, the response from legitimate users is of no use to the system, thus creating a suppression scenario.

The Current Trends for Auditing

The unprecedented growth of the DeFi ecosystem has resulted in capital being dumped into unaudited smart contracts. The recent hacks, however, have led to the realisation that smart contract audits, especially in the DeFi ecosystem, are paramount. As a result, the current auditing trends can be summarized with words such as escalating and overwhelming demand. DeFi audit firms have a waiting list of months as more and more DeFi platforms are being developed and require thorough auditing before they go to market.

The founder and CEO of Polychain Capital mentioned in a statement -

> I do think it scares me a little bit how much capital is being dumped into contracts that are unaudited. I think that getting security audits is, overall, an important part of maturing any one of these systems.

This statement provides further meaning to the need for audits in the DeFi ecosystem.

Types of Audits

Blockchain technology offers highly secure properties, but it should be considered that the technology works on code and it is individual people that will code the vital programming to incorporate and interface with the blockchain. Therefore, the whole system may be robust but it is not certain that there are no mistakes in the code.

Hence, there are some prerequisites driven by the International Standards on Auditing (ISAs). The auditors need to comprehend the dangers of the reports and how the organization is reacting to these dangers. With the rising reception of blockchain innovation, auditors should increase the expectations by providing progressive administrations and must incorporate advanced changes. Audits help to verify the organization's activities by investigation or assessment to guarantee that the organization complies with all

Interim Audit

Interim audit is necessary for reviewing work and systems where audit testing is performed between the time fiscal reports. It is essential for the audit technique, where the auditor needs to lessen audit works. The audit includes transaction details from all the assessments and a survey of records up to the date. The audit happens in the middle of two other annual audits which leads to a particular period with pronouncing interim dividend.

The auditor reviews the control system in detail and checks for errors and frauds. It causes the administration to evaluate the monetary situation of business during the first half of the year and resolves the error in the latter part of the year.

Full Security Audit

During the full security audit, specific consideration gets designated to the legitimate and administrative necessities explicit to the customer's business and area. The auditor intends to cover all parts of security and the threats, including all the individuals and organizations. The security audit measure follows an example to that of IT Audit administration that is utilizing a mix of mechanized and manual information to assemble the data. The auditor provides an assortment of skills and strategies, for example, partner interviews, organization, and application settings. At the end of the audit, the auditor gives a full report of the activities providing crucial information.

Basic Security Audit

A basic security audit is a day of service contributing to a high-level digital audit for different associations and organizations. It distinguishes the dangers and weaknesses that are faced by the association and also identifies the possible effects and probability that these dangers may have. The auditor will work to determine the strategy or potential measures that have been identified. At that point, the auditor must search for proof that the organization follows the proper measure. When searching for evidence, the auditor will utilize an examining approach. Instead of examining each record to state consistency, they will take a gander at a randomly picked state that makes it easier to complete the audit in a day.

Round the Clock Audits

Because of the necessities and prerequisites, more auditors get chosen daily to provide proper fixings of the errors using the round the clock

audits. Round the clock, audits monitor the all-day and all-night status of the organizations. The external auditors also get contacted to arrange for a visit to ensure the audit is safe as a third party gets involved. The auditor has to check and confirm the records thoroughly to guarantee that all mentioned records get completed reasonably, and no deception or misrepresentation is being followed. After the audit, a report gets transferred into a customer's data system, with round the clock status and the time given to work out the activity plans.

Key Security Attack Vectors

The present cyber status must have all the organizations positioning themselves one step ahead of the hackers to keep up their wellbeing. These consistently begin with distinguishing your shortcomings, seeing how your organization may get traded off, and executing the most appropriate strategies that will assist you with accomplishing digital flexibility. On the whole, you need to comprehend what vectors of attack you can experience that may upset your business and the steps your organization must follow to eliminate the attack.

Attack Vector

The attack vector is a technique utilized by a hacker to access the system's framework. Hackers demand cash from individuals and associations by researching known attack vectors and abusing their weaknesses to enter their system. The hackers can add malicious code that grants them access to control the system or steal important information and different assets.

Financial Risks

In recent years, the financial sector has faced the most extreme number of digital attacks, increasing financial risks all over the world. The explanation for the financial sector being the most targeted for digital hackers is because these organizations hold immense information about

individuals and organizations, which proves very significant.

A cyber-attack can perpetrate unsalvageable harm to the organization and can make clients lose faith. Apart from the financial losses incurred, the organizations will also lose their whole data and crucial information. One can prevent the attacks by taking necessary measures which may

Audit History

The audit history is generally utilized in data frameworks to ensure an organized record of changes gets made to the data. Audit history helps to check and validate operational activities, give verification of consistency, guarantee honesty, and distinguish malicious action. In the circumstance of an attack, the audit history gets used as a beginning step of legal procedure.

The accuracy of the Audit history is crucial to discover the reason for the attack and start appropriate countermeasures. For instance, the initial phase in recognizing the answer for the attack is by obtaining information on the conditions that lead to the crash. The information is acquired through audit history, which can help to form an impression for the states of the event. These help to accurately distinguish the main reason for the issue, for example, bugs and data altering. Besides, after solving the matter, the system can be re-established back to the state before the attack. At long last, audit history can likewise help to identify the interruption by providing valuable data to recognize unapproved access.

Centralization Index

Centralization index alludes to the member who is in the center associating with all the nodes aka the participants on the network. An organization's degree of centralization shows how much time it takes to be a star organization, wherein a member controls all the activity from the center. There was a requirement for standardization, which could

gauge the significance of a given vertex in an organization and would be based on the centralization index. Henceforth, the centralization index measure is dependent on the standardized difference in vertex centrality of any picked measure intending to allow a correlation of organizations on their centralization index scores. Centralization ensures the approval of participation data by favoring contracts that demonstrate the member has a place.

Monitoring and Troubleshooting of Smart Contracts

Developing a smart contract and auditing it is only the first step in ensuring unabated operations. For long term success of the business operations, the smart contracts need to be monitored constantly. Any dynamic analysis that expresses what should happen in which order, is one of the key elements of identification points during execution of smart contracts. Such an analysis done at runtime provides the information to troubleshoot smart contracts which simply means that effective monitoring helps you prepare for effective troubleshooting which can save you millions. One such tool to monitor and troubleshoot your smart contracts is QuillMonitor by QuillHash.

This is a specifically created monitoring tool which can:

- Help in tracking the behaviour of unauthorised calls in the smart contract
- Identify abnormalities in the functions of deployed smart contracts
- Create trust between the Investors and the organiser

Furthermore, this tool can be used to identify unexpected flaws in the contract and monitor the performance of contracts. Fundamentally, this tool keeps the investors as well as the organisation updated regarding the functioning of the smart contract. Any suspicious activity can be easily identified and notified using QuillMonitor. Dapps, Token exchange platforms, and DeFi protocols running on smart contracts are the perfect scenarios for QuillMonitor.

Smart Contracts Security Measures

Developing a fully functional smart contract with no vulnerabilities is a dream that can not come true. It takes a tremendous amount of efforts to not only identify but fix out all threats and bugs in the system. Since, a smart contract is an agreement written in code through which funds/ data can transfer without requiring mutual trust, it becomes a necessity to be 100% secure and bug free. Following are some security measures that developers should keep in their mind while coding the smart contracts.

External Calls

Use external calls carefully. External calls done in the code to the untrusted smart contracts can cause a plethora of risks and errors. A malicious code can get executed over the smart contract or any related smart contract depending upon that particular smart contract. Therefore, every time when an external call is made, it should be treated as a security risk. As an alternative, developers can use below mentioned recommendations in the remaining code to minimize the threat.

- Avoid state change after external calls

  Malicious code can execute while using contract calls or raw calls. The malicious code can run even if the external contract is not malicious at all because of the other contracts it calls. This kind of malicious activity can hinder the entire control system and can cause other vulnerabilities because of reentrancy. The best practice is to avoid a state change of contract after making an untrusted external call. This kind of pattern is often called a checks-effects-interaction pattern.

- Error handling in external calls

  Solidity provides low-level function calls on raw addresses such as address.call(), address.delegatecall(), address.callcode(), and address.send(). These methods of calling do not throw an exception but will return false if

they ever encounter an exception. If the smart contract needs any of the low-level functions, you should handle the possibility of call failure properly by checking the return value.

- For external calls use ‘pull’ instead of ’push’

  Failure in execution of external calls can occur both accidently and intentionally. To minimize harm because of these kinds of failures, always isolate external calls in its own transaction. The recipient of the call can only initiate this transaction. It is often practised where it is better to withdraw funds rather than pushing them. This safety measure also reduces the gas limit problems.

- Mark the untrusted contracts

  While programming smart contracts, always name your variables, methods, and contract interfaces, interacting with external contracts, in a way that they are potentially unsafe. Following this practice, one can easily judge the behavior of the code and the areas of potential threat.

- Never use transfer() or send() functions

  Functions like transfer() and send() forwards exactly 2300 gas. This fixed transfer to the recipient can be inadequate if it is a contract or cost of gas changes. Reentrancy vulnerabilities can be prevented by hard-coding gas, assuming no change in gas costs. An intelligent approach is to use a call() function instead of transfer() and send() functions.

- Don’t delegatecall to untrusted code

  The delegatecall function calls a function from other contracts as they belong to the calling contract. The state of calling address can be manipulated and changed by the caller. This kind of behavior can be highly insecure as the variation in balance and contract destruction can occur.

- On-chain data is public

  Some applications need their data to be private until a point of time so as to work properly. Let us take an example of rock-paper-scissors and sealed-bid auction. In rock-paper-scissors, a hash from both players is submitted before starting. Then the players submit their move one by one. If it does not match, hash throws it out. Similarly, in an auction, the bidders submit their hash for bid values before starting and bid values are submitted afterwards. Thus, if you are building such applications where privacy is the major concern, avoid early publishing of information by users. The best practice is to use 'commitment schemes' with separate phases.

- Possibility of non-returning and offline participants

  While having any transaction over smart contracts, times can come when the user may not return or become offline during an ongoing transaction. In such cases do not make a refund or claim to any specific party. For example, in the game of rock-paper-scissors the result is not given until both the players submit their moves. In such issues, we can do state channel settlement through fixing a time limit for response or procurement of incentives for submission of information.

- Negation of most negative signed integer

  In solidity language a signed integer of N bits can represent values from -2^(N-1) to 2^(N-1)-1. Negation of negative numbers will yield the same number itself for all the integer types i.e. int8, int16, ..., int256. This potential risk is specific to solidity but it can have a different version in a language you are using for your smart contract. It is advisable to check language related risks while coding the smart contract.

Common Challenges of Smart Contracts

Continuous research is being made into the field of cryptocurrencies and smart contracts. We all are very much tempted towards it but various obstacles are associated with the smart contract adoption in the industry. Some downsides of their implementation are:

Strong technical background to understand it

Use of smart contracts as contractual agreements over computer code can become a shortcoming if the user does not possess in-depth technical knowledge. Understanding smart contracts communicating with other contracts and other nodes on the network can be difficult to apprehend. It requires the person to have, at the very least, a basic understanding of the entire blockchain network

Reverse Transactions

Since smart contracts mostly work on public blockchains like Ethereum, they share data publicly with everyone on the network. Also, the validation process does not require permissions by every user. This feature is made possible by sacrificing abilit to reverse transactions and this trade-off is a big challenge in the smart contract world

Interoperability and International Standardization

A very important aspect of any system is its interoperability with different prevailing systems in the market. The nature of smart contract depends upon the distributed ledgers they interact with meaning that different Blockchains have different standards for their smart contracts. For this international standardization, development of blockchains is undergoing (ISO/TC 307). Without a particular standardization system, interoperability with other systems can not be established

Competition

Smart contracts are just becoming viable. Many other ways to codify agreements already exist and are used regularly in the market. For example, peer-to-peer lending via software as a service is more prevalent and growing rapidly. Therefore, it becomes challenging to compete with the existing systems by making the customers understand the benefits of smart contracts

Benefits of DeFi Smart Contract Auditing

- A smart contract audit helps identify the potential risks and vulnerabilities in your DeFi project. This is achieved through the complete and thorough testing of the contract code in various environments highlighting the operational, technical, and cyber risks to which the contract is exposed.
- The audit report generated after the completion of the smart contract audit helps in optimizing the code. It highlights all the bugs and fixes to be implemented in the code which leads to better structure of the contract code.
- During the audit, the performance of the contract is also tested with respect to different factors such as speed and security. This allows the performance optimization of the contract.
- In terms of the operating cost for a smart contract which is dependent on the underlying Blockchain platform, a quality audit helps evaluate the state of the contract and suggest possible optimization.
- A quality audit also helps in determining the compliance of the smart contracts with respect to the different rules and regulations depending on the location or industry.
- Lastly, a thoroughly audited smart contract instills more trust in the investors and other people in the DeFi space. As a result, if the contract is used for an Initial Coin Offering(ICO), Initial Public Offering(IPO), or a Security Token Offering(STO), then these initiatives can be more successful.

References

1. https://defipulse.com/
2. https://www.coindesk.com/defi-audit-firms-swamped
3. https://medium.com/conflux-network/the-overlooked-element-of-defi-adoption-e3b29829e3da
4. https://defiprime.com/defi-smart-contract-audits
5. https://www.coindesk.com/the-defi-flash-loan-attack-that-changed-everything
6. https://news.bitcoin.com/hackers-stole-100-million-defi-projects/
7. https://ciphertrace.com/half-of-2020-crypto-hacks-are-from-defi-protocols-and-exchanges/
8. https://codefi.consensys.net/blog/security-risks-in-ethereum-defi
9. https://academy.ivanontech.com/blog/defi-deep-dive-explaining-defi-attack-vectors-and-prevention
10. https://blog.coinbase.com/a-beginners-guide-to-decentralized-finance-defi-574c68ff43c4

QuillAudits

by Quillhash

448-A EnKay Square, Opposite Cyber Hub,
Gurugram, Harayana, India - 122016

audits.quillhash.com

hello@quillhash.com