Exploring Attacks in the NFT Gaming Industry: A Study of Risks and Mitigation Strategies

by

Zarin Rahman Mohima
19101334
Syed Ziaul Bin Bashar
19101166
Rawnak Muktadir
18201177
Amirah Hossain
19101292
Shafin Mahmud
19101419

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
23 January 2023

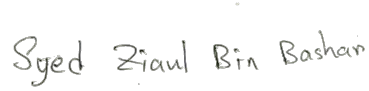
Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Zarin Rahman Mohima
19101334



Syed Ziaul Bin Bashar
19101166

Rawnak Muktadir
18201177

Amirah Hossain
19101292

Shafin Mahmud
19101419

Approval

The thesis/project titled "NFT in gaming: Cross Platform Implementation and Scam Prevention" submitted by

1. Zarin Rahman Mohima (19101334)
2. Syed Ziaul Bin Bashar (19101166)
3. Rawnak Muktadir (18201177)
4. Amirah Hossain (19101292)
5. Shafin Mahmud (19101419)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 23, 2023.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Exploring the decentralized concept of NFT games implemented on blockchain to demonstrate possession of specific game elements, has enabled the integration of blockchain features technology where the developer enables a cross-platform interchange of gaming assets, with cryptocurrency transactions. Blockchain-based games using NFTs have been developing since 2021 and several games released in 2022. As the industry is expected to keep on growing with mass cultural adoption from 2023-2024, it will give rise to many security issues as blockchain is vulnerable to different attacks. Thus, here we try to identify different types of attacks that might happen in the NFT gaming industry and also their possible countermeasures. Moreover, to prevent the fraudulent attacks on the ethereum network, we propose a hybrid fraud detection model using machine learning. Furthermore, we conducted a thorough breakdown of a relevant case study analyzing its hack mechanism.

Keywords: Blockchain; NFT; Blockchain-gaming; Security threats; fraud Detection; ethereum;

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our advisor Dr. Muhammad Iqbal Hossain sir and co-advisor Mr. Rafeed Rahman sir for their kind support and advice in our work. they helped us whenever we needed help.

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

Chapter 1

Introduction

Non-fungible token (NFT) is a form of digital ledger which is a non-interchangeable unit of data stored on a blockchain, it can be sold and traded on different marketplaces. NFTs, as it is growing has proved to be a valuable asset in the gaming industry as buying and selling of game assets can be done with the ownership of the individual. It is a decentralized system where the transaction between two parties will occur in the presence of a trusted third party with smart contracts through wallets. However, with its popularity, it has attracted scammers and malicious attacks can occur and so investors might fall into their trap and lose ownership of their NFTs, have copyright issues and unauthorized access. For instance, the attackers stole approximately $620 million from the network that runs Axie Infinity. Thus, these potential problems need to be solved and countermeasures need to be taken accordingly.

1.1 Research motivation:

NFTs made themselves known to the gaming industry at the end of 2021 and are expected to become one of the trending topics in this industry in near future. The players as well as the buyers both invest time and or money into securing NFT items in an inventory or NFT wallet which is possible for the combined feature of blockchain technology, attracting Web3 enthusiasts, startups, game developers and investors all at once. Such as, in the game Axie Infinity, the players can buy, sell and trade their assets just like we do with physical collectibles or cards and thus the game is attracting the gamer community who spend their precious time in this universe. Therefore, mass adoption of this new industry will lead to targeting scammers. More people will invest, the industry will grow to the maximum, more scamming games will be created and gamers and investors might fall into the trap and might face economic loss. Axie Infinity is also prone to the attack of hackers. The main concerns of the system include breach of privacy, unauthorized access by stealing keys and attack on nodes. Blockchain is vulnerable to identity-based attacks because the interface of blockchain depends on external data sources, which leads to major risks relating to the integrity of the external data and endpoint risks. These attacks can lead to the majority of the nodes getting hijacked in a network and we focused on those loopholes.

1.2 Problem statement

Our system is a decentralized design that deals with NFT in the gaming world. The system deals with privacy and security concerns, maintaining strict protocols and standards to ensure a safe and secure environment. For example, user authorization, owning NFTs, copyright issues, all this private information under security, information, ownership, breach of privacy and unauthorized access. One may fall victim to a malicious attacker, thus solving these potential issues are important to ensure a safe environment for our proposed system.

In the case of blockchain, every transaction occurs digitally in a decentralized manner. The issue with this concept is that a transaction between two parties might end up conflicting because one party might refuse to pay after getting the work done, that is, might scam the other party. A method of smart contract has been introduced where any transaction is watched by a trusted third party who ensures security between the transecting two parties. The third-party ensures the contract has been completed and the transaction has been done. In case of one party, refuses to pay after the deal has been completed, the wallet of that particular party will be frozen. Also, that party will be included in a blacklist which would specify him being a scammer. Thus, any future contract or involvement with this particular party would alert the other party to not trust them.

1.3 Research Objectives

Fraudulent activities are causing a major economic loss in the NFT gaming industry due to different types of attacks, which motivated our research team to find a possible solution that would detect various scams and propose a way to mitigate them. Thus, the aim of this research is to shed light on different types of attacks, their possible countermeasures and propose a hypothetical hybrid machine learning model implementation. The objectives of this research are:

- Understanding the architecture of blockchain framework and its part NFT in cross-platform gaming industry.
- Deeply analyzing NFT based gaming industry and its rapid evolution within a very short span of time.
- Thoroughly finding different types of security threats and vulnerability of them.
- Case study about recent scams in NFT gaming industry and causes behind it.
- Proposing countermeasures based on those security threats to prevent fraudulent attacks.
- A hypothetical hybrid model approach for detecting fraud transactions in the NFT games which are built on Ethereum framework.
- A relevant case study on a NFT based online video game, Axie Infinity which incorporates the main essence of our research.
- Analyzing the probable shortcoming(s) of the proposed model.

1.4 Thesis Structure

Chapter 1

Chapter 1 consists of an introduction to the NFT based games and its growing popularity which is attracting hackers.The research motivation thus focuses on the concerns of getting scammed through different attacks in this industry. Therefore, the problem statement deals with the intentions of the malicious attackers to prey on the potential issues of the system and the research objective is to find about different types of security issues and their countermeasures and to propose a hybrid model approach mitigating risk factors.

Chapter 2

In chapter 2, we provide a detailed literature review on the blockchain architecture that explains how blocks work in the blockchain and tampering with the hash makes the blocks invalid. In addition, we also explain the structure of the Merkle tree. Further, we move on to a brief description of NFT and related study of NFT games.

Chapter 3

Then, in chapter 3, we give a description of different types of security threats of NFT along with their countermeasures. We discuss the existing security threats in NFT and how to mitigate them.

Chapter 4

Chapter 4 deals with the machine learning implementation. Here, we built a model for detection of fraudulent transactions of Ethereum based systems through working on basic classifiers, neural networks and a hybrid model.

Chapter 5

Next, in chapter 5, we provide a case study of a recently hyped NFT based online game, Axie Infinity. We impart a thorough breakdown of the game with the hack mechanism, main reason for it and also the countermeasure of the attack.

Chapter 6

We conclude our research with chapter 6 with the benefits of NFT based gaming and the safe environment it can provide. Moreover, we also discuss the future scopes of our research work.

Chapter 2

Background

2.1 Review on Blockchain

The internet is rapidly shifting towards web 3.0 and blockchain technology seems to be at the core of it. NFT is a digital asset that is based on the blockchain network which then can be used to digitally showcase the ownership of certain intellectual property.

2.1.1 Blockchain Architecture

Blockchain is known as a sequence of blocks where all the transactions are recorded like an open ledger and is authenticated. It is a decentralized network and follows a distributed database architecture where the transactions are ordered records stored and referred to as blocks. Here, each block has a hash that is unique and recognizes the block along with its contents[6]. In each block, there is a block header, a reference to the parent block, and records of the recent transactions. The first block is called the genesis block.

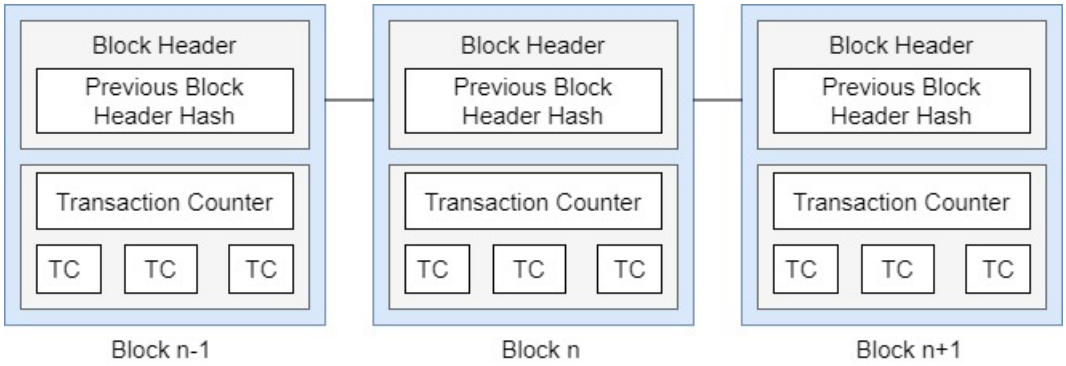


Figure 2.1: Blockchain Architecture

As we know, every block has a unique hash which is calculated during its creation. Any changes within the block will result in a change of the hash. Figure 2.1 has the blockchain architecture with each block having a block header, previous block header hash and transaction counter.

For instance, in this figure 2.2, we have three blocks here with their own and their previous hash except for the genesis block. In figure 2.3, if we tamper with the hash

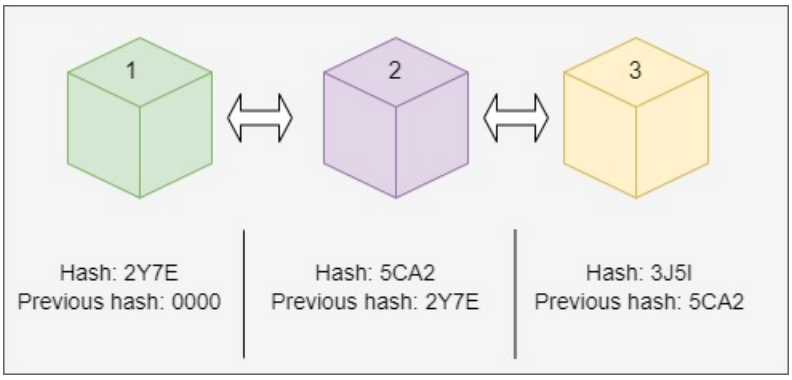


Figure 2.2: Architecture step 1

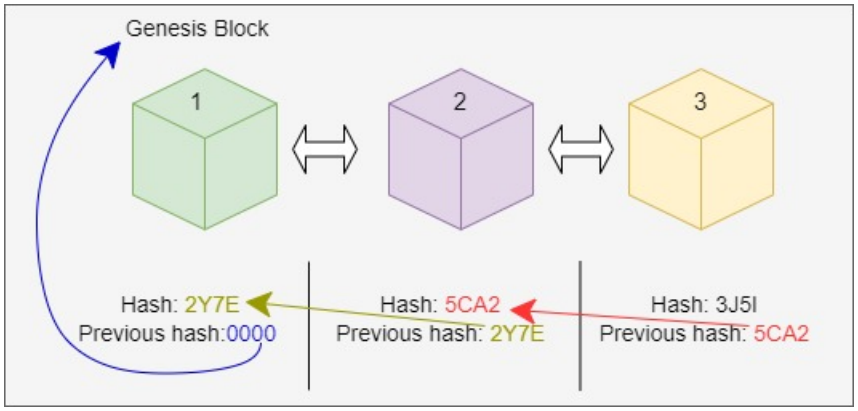


Figure 2.3: Architecture step 2

of the 2nd block, the hash will change making all its following blocks invalid.

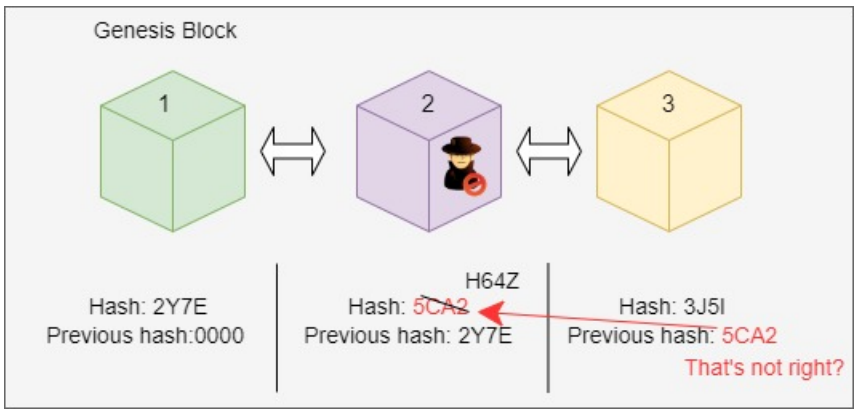


Figure 2.4: Architecture step 3

But modern-day computers can calculate all the following hashes in seconds and can make them valid again. That's why, blockchain maintains proof of work that slows down the creation of new blocks and thus it is difficult to tamper with as shown in the figure 2.4. Thus, blockchain has a distributed system and uses hashes and proof of work making it secure.

2.1.2 Merkle Tree

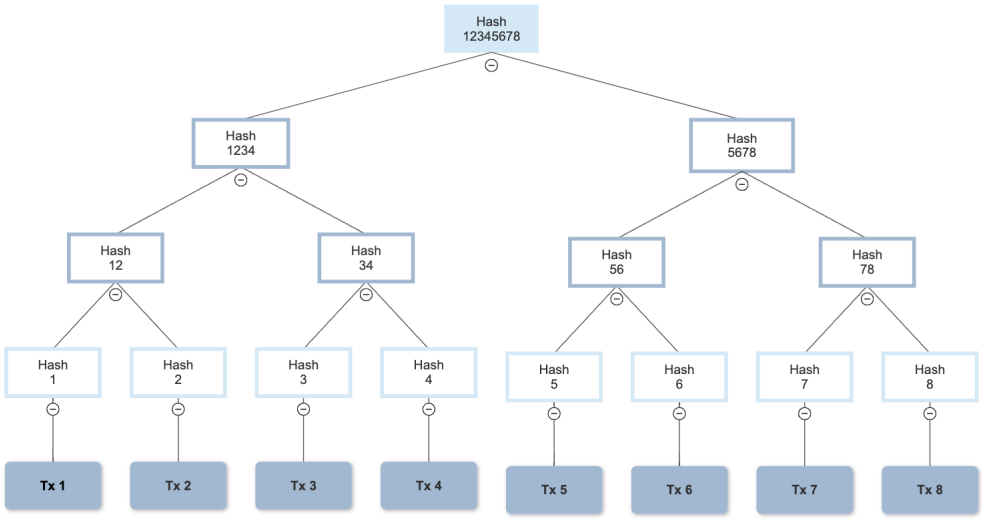


Figure 2.5: Merkle Tree

Merkle tree is very similar to binary search trees, having nodes, and each node holding the values of hashes of the children nodes. The only thing that holds these transaction values are these bottom nodes in figure 2.5. The way these nodes are so secure are because their parents are hashes.

2.2 Analyzing the NFT Gaming Industry

NFT is considered as a form of digital currency that is derived from the concept of smart contracts of ethereum, where the creator has all the rights to easily own the existence and ownership of his assets in the form of images, art, video games, event tickets [4]. NFTs, or non-fungible tokens, are digital assets that are unique and cannot be replicated. They are often used in the gaming industry to represent in-game items, such as weapons, armor, and collectibles. These items can be bought, sold, and traded on blockchain marketplaces and they can also be used to prove ownership and authenticity. NFTs are becoming increasingly popular in the gaming industry as they allow players to own and control their virtual assets in a way that was not possible before.

It is possible for scams to occur in the market for non-fungible tokens (NFTs) used in gaming. These scams can take various forms, such as some scammers may create and sell counterfeit NFTs that purport to be from a particular game or project, but are not actually legitimate. These scammers may create an NFT investment opportunity that promises high returns, but actually offer a scheme where early investors are paid with the funds of later investors. They might launch an NFT gaming project, attract a large number of users and investors and then suddenly shut down the project and disappear with the funds. In addition, these scammers might send fake emails, and messages, or create fake websites that look like legitimate NFT gaming sites in order to steal personal information or funds from users. It's important to be aware of these potential scams and take steps to protect the users.

2.2.1 Interdependency of NFT and Etheruem

The Research team has studied the correlation between NFTs and Ethereum to find significant interdependencies. While Ethereum most widely supports NFT, NFTs can be used to represent the ownership of unique assets and this ownership is secured by the Ethereum blockchain. They can represent real-world items like art and real estate collectibles and we can tokenize them. Initially, a token standard of Ethereum presents the notion of NFT with the objective to distinguish tokens with distinguishable signs. For unique identification of digital assets, we can bind these tokens with them.

2.2.2 Related Study of NFT Gaming

NFT games or blockchain games is a huge paradigm shift in the modern gaming industry about how the gamer can turn his gaming hours into an investment, following some basic criteria. After conducting thorough research on the recent evolution of NFT-based games, it is understood that in early 2021, we saw the NFT world just explode and now game developers are jumping on NFT-based games.

Players who are invested in gaming are trying to secure NFT items within these crypto games.[11] They hold those items in an inventory over time when those items go up by making them able to sell or trade more money for dollars. While massively popular multiplayer online role-playing games are expected to spend a huge amount of their assets, money and efforts, the developers as well as the business owners are predicted to adopt this new industry, accelerating its fame, and overall boosting a huge portion of the economy. This new revolutionary idea might even sound very rewarding to the gamers, because their time, effort and dedication is not going wasted. The game builders are free to develop NFT based games without involving any third party and the gamers get rewarded in exchange of their time and effort as well. This is possible for the combined features of blockchain technology. The total amount spent on NFT transactions has crossed 34,530,649.86 USD by May 2021. The interest in NFT is growing at a rapid pace. As of 2022, the gaming industry has reached a market value of USD 173.70 billion, with a CAGR of 9.64%, expecting to reach well over USD 314 billion by 2027 [16]. We are trying to work on this fastly growing industry where the interest of people is very high currently.

Decentraland: Decentraland is the first game with a fully decentralized virtual world. In this 'decentralized virtual world' players can buy and sell plots of lands using cryptocurrency.[21] This game is built to track real estate parcels which are defined by LAND tokens using the Ethereum Blockchain. Using Ethereum helps the game with tracking ownership of digital assets and holds mana tokens for each individual within Decentraland's ecosystem. This is a whole new dimension of online gaming with a whole virtual universe. Gamers will find ample ways to make an earning playing this game. In the game, users can get employed by a company with a salary, work as a freelancer for clients, or even create items to sell for profit. It is noted that an NFT-based metaverse real estate company bought a plot on the game worth $2.43 million in November 2021. [9]

Aavegotchi: Aavegotchi is also a Crypto collectible game built on Ethereum

Blockchain architecture, in the game the users can participate in purchasing and growing Tamagotchis, Non-Fungible Token (NFT) avatars. These Avatars can be used to explore and play within the game's virtual universe. In this game, each of the avatars has unique digital attributes defined by rarity score. Here you obtain Aavegotchis by first obtaining GHST tokens. Currently each costs around 2.02$(February 1st, 2022), and it is expected that the price is going to reach $3.517 by the first of February 2023 and $9.495 by the end of January 2027.[14]

Skyweaver: Skyweaver is a trading card game where the cards themselves are NFTs. Here, you can own, trade, and gift your cards. This game requires strategic gameplay with a crowded marketplace that lets him win NFT cards on the game by following some strategies.

Rarible: Rarible is most hyped game in the current gaming industry which enables selling and buying digital assets.[17] The game is known to have created the gaming market very competitive and attractive to the users.

Somnium Space: Somnium Space uses its in-game mechanics by implementing cross platform which is available on all major VR headsets enabling virtual land ownership.[18] This system allows the user to buy and customize their own land and to build anything he envisions.

Sorare: Here, the platform has a global fantasy football league which uses blockchain based collectable digital cards, owns 230 clubs, it helps to connect and trade with other clubs in the open marketplace. [19]

SuperRare: SuperRare collects digital art, Buy and sell NFTs from the world's top artists, live auction process. [20]

Splinterlands: It is a free-to-play tradable card game which lets users earn as they play. The user can earn rewards on the condition of winning matches. By registering an account and unveiling the purchased cards on Splinterlands is a must. Here, the users can proceed to battle other players too, which makes them a unique one. [12]

AtomicHub: It is a one stop solution for creating, trading, buying and selling NFTs that is already used by millions of loyal users which started as an organic social collaboration. Within a game, players invest time and/or money in obtaining NFT items. Those items are kept in inventory or in crypto wallets by the gamer and over time the value of those items goes up as more players join the game and demand for those items increase and could be sold for more money. To prevent the illegal copying of copyrighted contents, some core technologies are required such as authentication, access control and encryption.

Chapter 3

Attacks on blockchain system and it's mitigation

To store NFT, Ethereum based blockchain maintains a peer-to-peer network system. In this report, we explore some of the security threats of blockchain, since NFTs are blockchain-based tokens, so the risks are still to be considered that NFT gaming industry can inherit the vulnerabilities of blockchain.

3.1 NFT wash trade trend: Price manipulation

This trend refers to a practice where a price hike or manipulation happens in an unethical way to benefit an individual while he artificially tries to inflate or deflate the price.[13] While this practice is illegal, in recent times it has been reported that almost 2% of global NFT games involve rapid wash trading followed by illegal transactions. This creates a malicious scenario in the NFT industry, making it easier for scammers to take advantage of it. With the intent of manipulating or misleading the market, this high-frequency wash trading practice is expected to create a negative impact on the market making it hard for new users to step in.

3.2 Timejacking Attack

Timejacking attack is a potential vulnerability that exploits the handling of timestamps. By initiating this attack, attackers try to push in inaccurate timestamps while connecting to a node. Once the current state of the network time counter of the node is altered, the attacker adds a deceiving node that will accept alternative blockchains. This leads to creating a possibility for threats like double-spending and creating mishaps in computational resources. [10]

In a blockchain, each of the nodes maintains a time counter which is based on the median time of a node's peers. This time stamp is sent when it tries to connect to a new block in the blockchain. In this case, the attacker tries to plant multiple counterfeit peers in the network and all these fake peers will report inaccurate timestamps, this leads to slowing down the system and speeding up network time.

Timejacking attack mitigation:

While studying a research, [10] we found a few solutions which can help overcome the Timestamp attack perpetuating a Node system. Using this method, we can stop the occurrence of this Timejacking attack. Another way to stop this attack is to restrict the acceptance time range. For example, restricting the timestamp window to 30-60 min replacing the 70 and 140.[9] Hardware-based time sources, such as GPS or radio clocks, are more accurate and less susceptible to tampering compared to software-based time sources. It is to note that using this restricting method will not totally prevent splits entirely, as nodes that have daylight saving time might be left behind. Digital certificates and encryption can be used to authenticate the identity of devices and systems, making it more difficult for attackers to impersonate legitimate devices and disrupt the flow of time.

3.3 Sybil Attack

The Sybil Attack is an orchestrated invasion within a Peer-To-Peer Network, where an individual forges multiple identities, in order to compromise the system [1]. The core reason behind such attacks is to grow an unfair influence within the system. In terms of a Blockchain network, the attacker creates multiple nodes and controls it. As the platform is decentralized, a Sybil attack can always commence given that the attacker has the proper resources and entities' coordination. In launching an attack, the aggressor first identifies the node that it wants to isolate; the attacker will reconnect the victim node from its original peers to the forged nodes until it is isolated. This may halt any transaction from honest-to-honest blocks as it has been isolated from the network. Furthermore, the user then is forced to transact with the forged nodes. However, in a Peer-to-Peer network, an user only needs another honest peer to transact data properly and be connected to the platform safely. This yields to a victim node being safe even if it has been isolated from the network except only connected to one honest peer [2]. Such a countermeasure is implemented in the Bitcoin system and keeps NFT transactions safe.

Finally, in order to placate a Sybil-based DoS attack, a two-party protocol, introduced as Xim, is a design that finds nodes, anonymously, on the basis of ads launched on the decentralized system [2]. The design increases a user's cost proportional to the participants of the network. As Sybil Attack is based on forging multiple identities, Xim not only creates a cost threshold for each account in order to participate but also ensures the cost grows linearly with the sum of mixed participants. This ensures that Sybil attacks' profit margin is not sustainable and decreases in proportion to total mixed nodes.

Sybil attack mitigation

Proof of work protocol prevents a Sybil attack from taking place. So when a new peer connects to the network it can only be connected through a single honest peer, this is for the resistance to Sybil attack shown in the figure 3.1. So if an attacker creates fake nodes/peers, they will fail to connect to the network. So as long as a single honest node is passing through true data, the system will ignore all the at-

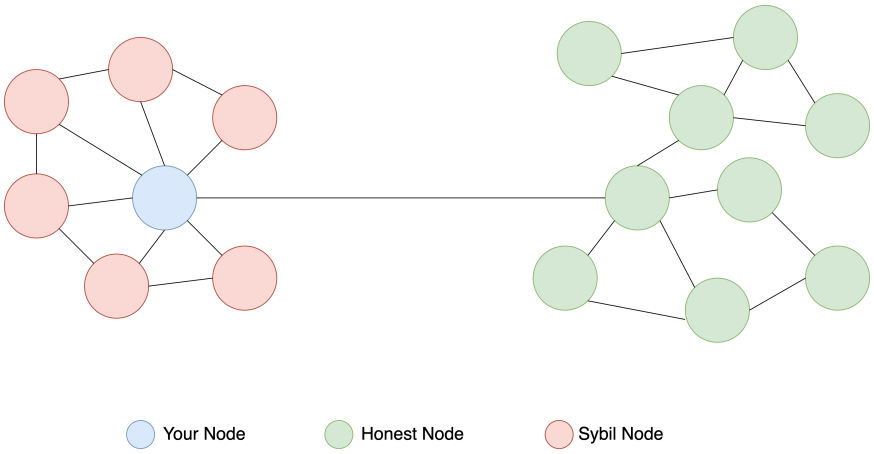


Figure 3.1: Sybil Attack on Honest Nodes

tacks from a Sybil attackers node. Moreover, to ensure an individual does not forge multiple identities, in terms of creating pseudo-anonymous wallets, we propose to verify tokens through a biometric system. In this way, a user can create multiple wallets, but with a token of existence as verification. This way every wallet is verified with individual and illicit transactions or attacks can be traced, in terms of Sybil attack. A reputation system can be used to track the history of each user's interactions within the network and assign a reputation score based on their past behavior. This can help identify and weed out malicious actors.

It is important to note that no single countermeasure is likely to be completely effective at preventing Sybil attacks. Instead, it is often necessary to use a combination of different approaches to providing an adequate level of protection.

3.4 Smart Contract Threats

The smart contracts are designed in such a way so that they will execute autonomous computations following decentralized entities on a blockchain as soon as the conditions are met. To address the wave of smart contract vulnerabilities which are prone to be exploited by cyber-criminals, and run in the Ethereum blockchain [7]. A recent increase in the adoption rate of smart contracts makes us do research about the security issues, coding issues as well as privacy issues of smart contracts.

To find out more about the causes of security breaches, it is important to pen down some of the issues with smart contract programming and its subfields. Advancing the codes, analyzing the security code as well as identifying the correct way of using tools is the best approach to creating a model of security issues taxonomy. While our main objective is to find out the advancement of security breaches, identifying the vulnerabilities by inspecting similar attacks which have happened in recent times is our first research topic.

Attack scenario of smart contract

In this section, we discuss the security-sensitive aspects of smart contract threats for NFT.

I Reentrancy attack:

A reentrancy attack is a specific type of vulnerability that can occur in smart contracts, where a malicious contract repeatedly calls another contract with malicious intent. It occurs when a contract calls another contract and before the called contract has finished executing, the original contract continues execution, potentially calling the same contract again. This can lead to an infinite loop and can also allow the attacker to repeatedly call the contract with malicious intent [8].

For example, an attacker can create a malicious contract that repeatedly calls a contract that manages a bank account and withdraws funds from it. This can happen because the malicious contract can call the bank contract multiple times before the bank contract has a chance to update the balance.

Mitigation: Reentrancy can be mitigated by utilizing a pattern called reentrancy guard, which is a pattern that prevents reentrancy attack by making sure that the state of the contract is updated before the external call is made. Another way to prevent reentrancy is to use a mutex, a synchronization mechanism that allows only one contract to execute at a time. Also, it can be avoided by using 'view' and 'pure' functions in smart contracts which do not modify the state.

II Exception disorders:

Integer underflow or overflow can occur making the bytecode faulty. This situation can lead an attacker to change his current balance to an increased amount that he does not own. Thus, an integer underflow can cause loss of huge tokens and money.

For instance, if the amount of representation a certain variable can hold is X and we write a code where it is instructed to print any value beyond this limit, this will cause an integer overflow.

Example code:

```
#include <stdio.h>

  Int main( int argc,char* argv[]) {
        unsigned char Y=252; // 2^8=256
        Int i;
        for (i=0; i< 20 ; i++ ) {
            printf ( \%hhu  | %hhuX\ Y'' , Y , Y );
            Y++;
    }
    return 0;
}
```

Vulnerability in solidity

III. Call to the unknown:

In solidity, a call to the unknown attack scenario refers to incorporating some unknown functions to invoke regular functions. The "Call to the Unknown" attack is a type of vulnerability that can occur in smart contracts that use external calls to other contracts. The attack occurs when a contract calls another contract without properly checking its address or code. This can allow an attacker to redirect the call to a malicious contract that they control, which can then steal funds or disrupt the normal execution of the original contract.

For example, a contract may have a function to transfer funds to another contract's address, but it does not properly check the address before making the transfer. An attacker can then create a malicious contract with the same function signature and change the address in the original contract to point to the malicious contract. When the original contract calls the transfer function, it will unknowingly send the funds to the attacker's contract.

Mitigation:

To mitigate this type of attack, it is important to always check the address and the code of any external contract before making a call to it. One way to do this is by using the "address.code()" function, which returns the code of a contract at a given address. Additionally, it's important to use a whitelist of known and approved contract addresses before making the call. Another way to mitigate the risk is to use 'delegatecall' or 'staticcall', which are opcodes that can be used to make external calls in a more secure way. These opcodes allow the contract to call external functions while still using its own context and storage, making it less likely for an attacker to tamper with the data.

3.5 Wallet Security Threats

There are two types of wallets, which is Hot wallet and Cold wallet. A hot wallet is always connected to the internet and the private keys act like a password to sign transactions on the blockchain. Thus, keeping them secure is critical and it's better not to keep significant funds there. Whereas, cold wallets are not connected to the internet and private keys are not exposed, so it is more secure and is more like a hardware wallet. People are getting scammed out of their NFTs by keeping too many assets in their hot wallets. The main way to get scammed is by people going to a fake website created by someone which looks exactly like a trusted website and entering their seed phrase, which is the master key to all the wallets and generates private keys. Some methods are more advanced like putting code in a specific PDF, and opening it gathers data from our screen. Hot wallets are easier to interact with certain applications on the internet. But, here private keys are stored online which can be hacked and thus seasoned investors use this strategy.

Countermeasures of wallet security threats:

We need a safe place that is inaccessible to anyone no matter what. Using a hot wallet we can buy an NFT, send and receive crypto and interact with certain games in the metaverse and then we can have a cold wallet that stores the valuable NFTs that we won't sell anytime soon. Also, another effective way is by partitioning wallets. Here, the address of each wallet is unique and protected from the interactions of other addresses even if they were generated on the same hardware wallet. So, for instance, if we interact with a malicious contract on one address, it won't affect others. We can split the hardware wallets into- High-Risk wallet through which risky NFT project mints can be done or a not trusted site can be visited for not missing out on deals, Buying/Selling wallet and Vault which are wallets never interacting with the smart contracts and thus greatly mitigating the risk factors and protecting the valuable NFTs. Furthermore, we can also revoke contracts using the website revoke.cache. Some smart contracts like selling in OpenSea remain open in perpetuity which can be opened up to sign transaction scams. So, we might be thinking that we are signing to verify an asset but what actually happens is that the scammer creates an OpenSea contract to create a private sale for free to their own wallet and then they'll accept that private sale and take our item and we won't even realize what happens. However, these sign transaction scams can only work if they are able to leverage existing open smart contracts.

3.6 Eclipse Attack

An eclipse attack is set to create a network partition between the public peer-to-peer platform and the victim node. The attacker does that by having control over a significant amount of IP addresses [6]. In order to manipulate the node, the attacker fills the peer table of the user with its IP addresses, as opposed to it being filled with public IP addresses. Once the node restarts, it disconnects with its peers and the new connection takes place- effectively isolating the victim from the actual public network and onto the attacker's desired IP. The attacker, then, can exploit the victim as it has created an isolated connection with the node. However, to launch an eclipse attack, the victim node must accept incoming connections. For example, in order to propagate "Tried" keeps the IP addresses that have successfully created both incoming and outgoing connections with the node, and the "new" table keeps the IP addresses obtained from DNS seeders and ADDR messages. Once again, to launch an eclipse attack, the attacker fills the "tried" table with its desired IP addresses. Then, overrides the "new" table with IP addresses that are unallocated, creating a buffer to disconnect the victim upon restarting. Once the node restarts, it randomly selects an IP from either the "tried" or the "new" table. However, in this case, the node will select from the "tried" table as the other one is filled with unallocated IP addresses. Given that the connection with the desired malicious IP is successful, it will be considered as a new connection. Such a case will iterate a total of eight times for a successful TCP outgoing connection. Upon the completion of the loop, the user is now connected to the malicious node and can be exploited [3].

Countermeasure of Eclipse Attack:

Wang F. (2015) [6] suggests accepting only limited connections from the same IPs as opposed to nodes accepting all connections from the same address. Furthermore, users must avoid large incoming unsolicited ADDR messages, which is more than 10, at once. Also, to avoid buffering the "new" table, a node should accept only soliciting address messages from outgoing connections. The author has also suggested developing a random selection algorithm to select addresses from the table.

3.7 Refund Attack: (Marketplace Trader Attack)

The payment protocol of BIP70 was proposed with the public address of the merchant being with their respective X.509 certification. The communication between the customer and merchant will be over HTTP/HTTPS setting. To initiate the transaction, a cryptocurrency URL will be generated from the merchant's website as soon as the customer clicks on "Pay Now". The URL, then, downloads the merchant's Payment Request Message and the customer's wallet verifies the public key written on the X.509 certificate, as well as checking the authenticity of the certification itself. This, then, shows a human-readable name alongside the requested cryptocurrency on-screen for the customer to check. Upon pressing "Send", the customer's wallet successfully transacts cryptocurrency to the merchant and sends a payment message with a refund address to the merchant.
In such Protocol, the instances needed are as follows:

- **Payment Request:** Payment Address, Ap, number of Bitcoin requested, , time of request t1, time of expiration t2, memo message, mM, Payment URL uM, merchant's data for future payments/ reference, dM, the private key, kM, the algorithm implemented for digital signing, and finally, merchant's correlated public key of X.509 certification, µ

  In which, $\mu = \lambda_{KM}(A_M, B, u_M, m_M, d_M, t_1, t_2)$

- **Payment:** Payment transaction τ_B, refund address R_B, and Bitcoin B being listed such that $(R_{AB_1}, B_1), \ldots, (R_{AB_n}, B_n)$ is the refunded Bitcoin to each address, memo message, m_B.

- **Payment Acknowledgement:** A repeat of a buyer's payment message, P'.

Now, let us focus on a simplified figure 3.2 of HTTPS communication:
The lack of authentication is seen in the refund policy of various cryptocurrency payment service providers, as it is not needed to be digitally signed by the customer. This means the integrity of the refund address relies on the HTTP/HTTPS website and the merchant. For this reason, certain wallet applications have a refund policy to use external email to communicate, strictly to seek refunds. Such a deviation from the Payment Protocol has created a new phishing-style attack known as the "Marketplace Trader Attack" [5].

In such an attack, the scammer develops a pseudo-website to sell market products at an attractive price. As the prices are well within the market value, the customers may be suspicious of the scam. To placate suspicion, the trader advertises trusted

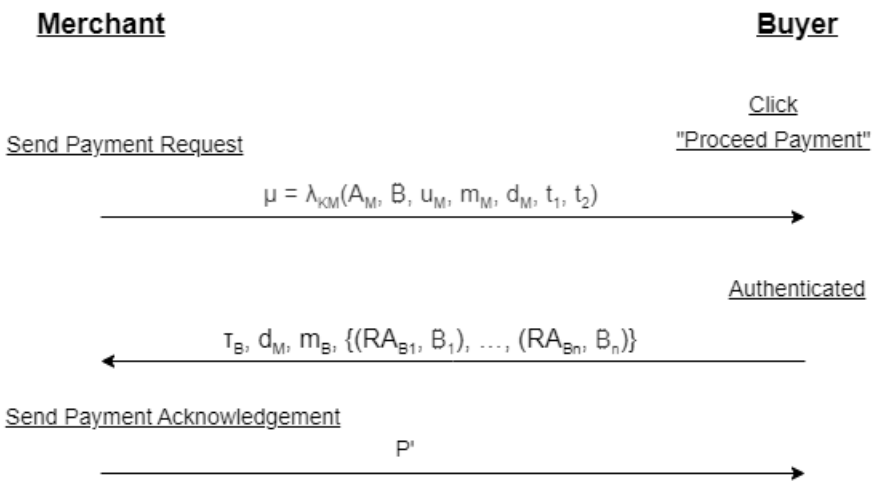


Figure 3.2: Simplified HTTPS communication

platforms to transact, such as Coinbase or Centralized Exchange. As there is no suspicion, the customer can now proceed to buy the desired product(s) and press "Pay Now". To proceed with the payment, the website then fetches an original Payment Request message from the merchant, which the customer's wallet verifies through the X.509 certificate. This yields the customer having a readable name and crypto amount on-screen to pay. As soon as the customer pays for the transaction, the website detects it on the broadcasted network and refreshes the page to confirm the transaction. However, by paying from that pseudo website, the user fell prey to the scammer, as the attacker now utilizes an external communication platform e.g. email, according to the refund policy, to cancel the purchase and seek a refund to another address from the trusted merchant. The scammer then successfully obtains the amount the customer has paid for the product, whereas the customer is not aware of the scam that has occurred yet as there is a lack of information for the customer to identify the refund taking place.

Countermeasures of Refund Attack:

As per the solution, we provide a Proof-of-Approval concept. In our Payment message, a transaction endorsement Bi would correspond to a specific refund address RABi. Bi would have the signature to authenticate payment and associate the public key alongside the index of transactions output, i. Both Bi and RABi would be linked through an endorsement signature algorithm, KBi, in which KBi is the same private key that authenticates Bi. The algorithm would consist of the following: Requested Bitcoin value Bi, the merchant's inscribed digital signature, Payment Request, and additional memo given by the Buyer, mBi, having the Refund address. Thus, the endorsement signature is μBi = KBi(Bi, Bi, mBi, Payment Request)

The proposed Proof-of-Approval for the requested refund would require the identical keyset, inscribed by the merchant, to authenticate the approval by only providing the private key KBi to the customer. In this way, neither the merchant nor the attacker can open the encrypted endorsed digital signature. However, the merchant's wallet application will only be able to open the message once the customer seeks a refund by emailing the private key KBi , which only the customer had upon transaction.

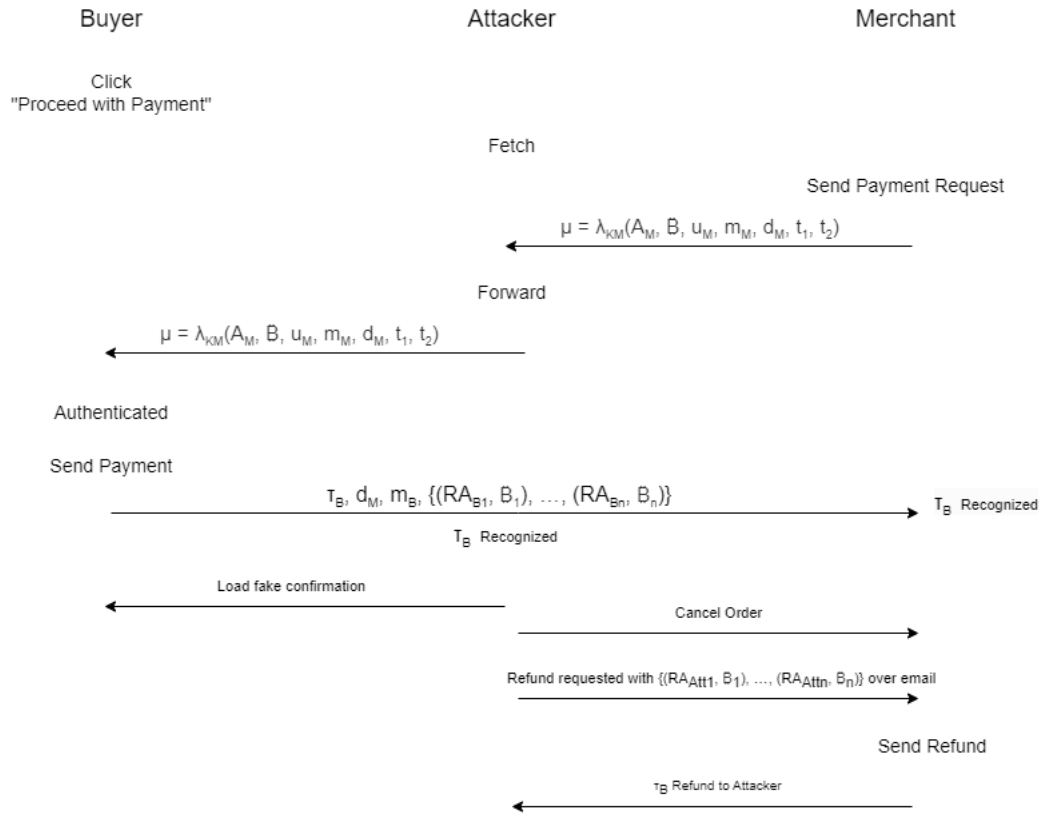


Figure 3.3: Refund Attack

Such corroboration provides a gatekeep to protect the currency of the original owner till notified, regardless of how the attack took place. The attacker need not identify as a co-signer for endorsement to retrieve the refund, as the additional signatures are handled by the wallet instead of the user. Regardless of the human errors or methods of scamming for a refund, our Proof-of-Approval endorsement signature also provides a backup incentive to protect the transacted currency itself, given that the payment service providers developed additional verification methodologies.

3.8 Bribery Attack

Suppose an attacker attempts to launch a double-spending attack against a seller, who controls a minority of the computational power in the Bitcoin network. The attacker does a large transaction with the seller and the transaction is included in the chain. The attacker starts working on a fork privately by creating a block that does not include that transaction. Transaction is confirmed after the attacker creates at least one block in their chain. This block has some special transactions in which the attacker transfers some BTC to new addresses that they have created before. The seller sends the attacker the purchased goods and the attacker releases their block and tries to bribe other miners to mine on their fork after the attacker's transaction to the seller is confirmed. The attacker discloses the private key of the account that they have transferred the money to in the blocks just created. Rational miners may decide to work on the attacker fork who sees the private key. They try to create new blocks on the attacker block after creating a new transaction

to send the BTCs in the disclosed account to their accounts. The attacker confirms a solution by disclosing the next private key as soon as a miner finds a solution. The second branch will take over the main one and the transaction between attacker and seller will be undone if the attacker manages to give sufficient incentives to attract enough mining power to their fork. The attackers' cannot hope for a successful attack without encouraging other miners to work on their fork because they have a small fraction of the network's computational power. Rational miners will join the attackers' fork if mining on the attackers' fork is deemed more profitable by them. While making sure that the attack remains profitable for them at the end they need to keep convincing miners to mine on their fork. In order to initiate the attack, the attacker attacks at least one block in their fork that contains transactions to the account IDs to be published. Assuming that a transaction needs six blocks to be confirmed, the attacker should have at least 14.28% of the network computational power. With such power, the attacker can on average create one block in 70 minutes, that is the time required for the creation of 6 blocks by other miners with 85.72% of the power in the main fork.

Countermeasures of Bribery attacks:

Limiting the amount of BTC that a user can transfer could be one solution to overcome this attack. The limitation is closely related to the amount of block reward. Decreasing the reward increases the attack success chance if the attacker's budget remains constant. Mining algorithms should contain a mechanism that adjusts the limit every 4 years when usually the reward falls in cryptocurrencies like BTC. Since the attacker can transfer BTC with multiple addresses in the same block, this solution is not perfect. Limiting the whole transferred BTC in each block could be a solution to prevent previously mentioned drawbacks. Like previously, there should be a mechanism to adjust the limit in this solution. The situation in which there is no block reward and the only reward is the transaction fee, should be taken into account while implementing these solutions. At least a soft-fork is necessary in all case.

3.9 Validator Nodes attack

Validator nodes are proof of stake blockchains where there are no mining equipment. We buy a bunch of coins, then stake it on a validator node. Instead of a node communicating with a mining network, that validator node does both of the things. It will verify and validate the transactions and it communicates that information and data on the blockchain. In a proof-of-stake (PoS) blockchain, validator nodes are responsible for validating new blocks and for participating in the consensus process. Nodes validate the transactions, where information is received and sent. A validator node attack is a type of attack that targets the validator nodes of a blockchain network. Validator nodes are responsible for validating and processing transactions and maintaining the integrity of the network. Validator nodes can help to enhance the security of a decentralized system by providing a number of important functions, including:

Verification and validation: Validator nodes verify and validate transactions and

other actions within the system, helping to ensure the integrity and accuracy of the system. Verification and validation of validator nodes are important steps in maintaining the security and integrity of a blockchain network. Verification of validator nodes refers to the process of ensuring that a node is who or what it claims to be. This can include checking the node's identity, location and other identifying information. Validation of validator nodes refers to the process of ensuring that a node is following the network's rules and protocols and is not acting maliciously. This can include monitoring the node's behavior for any suspicious activity and checking that the node is properly following the consensus mechanism.

Consensus building: Validator nodes participate in consensus building processes to reach agreement on the state of the system, helping to ensure the system is secure and consistent. Consensus building for validator nodes is the process by which a blockchain network reaches agreement on the state of the ledger among all the validator nodes in the network. The consensus mechanism used by a blockchain network determines how this agreement is reached. For example, in a proof-of-work (PoW) blockchain, such as Bitcoin, validator nodes, also known as miners, compete to solve complex mathematical problems in order to add new blocks to the chain. The first miner to solve the problem gets to add the block and receives a reward. This process is called mining and it secures the network.

In a proof-of-stake (PoS) blockchain, the validator nodes, also known as validators, are selected to validate transactions based on the amount of cryptocurrency they have "staked" as collateral. This process is called staking, and it secures the network. In both cases, the validator nodes reach consensus by following the rules and protocols set by the network's consensus mechanism.

Network security: Validator nodes can help to secure the network by participating in network security protocols, such as proof of stake or proof of work, and by helping to prevent malicious actors from gaining control of the network.

Countermeasure of validator node attack :

Increasing the number of validator nodes in a blockchain network can improve security by making the network more decentralized. With more nodes participating in the validation process, it becomes harder for any single entity or group to control a majority of the network's computing power and manipulate the system. Additionally, spreading the validation process across more nodes can help to reduce the potential for network congestion and improve the overall performance of the system. However, it is important to note that while validator nodes can help to enhance the security of a decentralized system, they are not a guarantee of security. Other security measures, such as implementing robust protocols and technologies, conducting regular security assessments, and providing secure infrastructure and hardware are also important for protecting against hacking and other types of malicious activity.

Chapter 4

Fraud transactions detection of Ethereum based system

Smart contracts on Blockchain provide a way for developers to create decentralized applications. This is particularly useful for NFT games, which often rely on the ability to create, transfer and trade unique digital assets that are owned by the players. Moreover, the Ethereum network is at the core of the creation of global, open marketplaces for NFTs. This allows players to easily buy, sell, and trade their NFTs with other players, regardless of their location.

Therefore, fraud detection on Ethereum can help detect NFT scams by identifying and flagging suspicious activity in the blockchain gaming industry. By monitoring the blockchain for suspicious activity, it is possible to detect and flag any suspicious transactions or addresses that may be associated with a scam. Thus, we tried to implement a Machine learning model that can be trained to identify patterns and anomalies in the transactions of NFT games which are indicative of fraudulent activities.

4.1 The Dataset

For our dataset, we needed data regarding the known valid and fraud transactions made over the Ethereum network [15]. We searched online and found a dataset that matched our needs named "transaction_dataset". Our dataset has 45 parameters. Some of the significant ones include "Address : the address of the ethereum account", "FLAG: whether the transaction is fraud or not", "Sent_tnx : Total number of sent normal transactions", "TotalEtherSent : Total Ether sent for account address", "ERC20AvgTimeBetweenSent_Tnx : Average time between ERC20 token sent transactions in minutes" etc. All the fraud transactions are not available in this dataset as it is near impossible to gather that information and also not all fraud information is available to the general public or known.

4.2 Data Preprocessing

Data preprocessing is a very important step to increase the performance or accuracy of the classifiers. Since our dataset includes a huge number of information in those

45 parameters it is crucial that we choose wisely which parameters will give us the maximum accuracy for our model. First, we select the columns and select them as tuples to perform all the actions. We replace all the spaces then try to find if we can group by address or any other fields. Then, we split the data into X and y and try to remove multicollinearity. We do this just to clean the data for our classifiers to run more accurately.

To group the data, we check for the most useful attributes and also check the occurrence of that attributes by checking their count in the dataset. From this, we find that "ERC20mostsenttokentype" was the most relevant to what we were trying to achieve. So, we use that to group our data. Then, we check for missing values, clear those, and split the data. After this to remove multicollinearity, we create a correlation matrix and check the upper triangle of the matrix. From there, we remove all the highly correlated values. We were looking for anything greater than 0.7. We also dropped the columns that had only one unique value. After doing so, we will split this data into training and testing sets. We are using 70% of our data for training and 30% for testing.

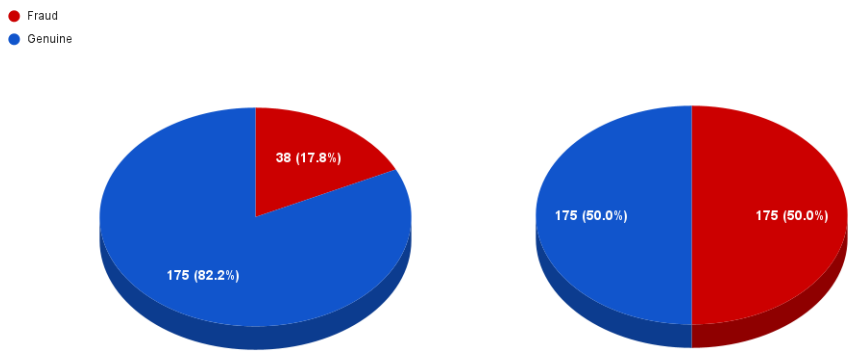


Figure 4.1: Class imbalance

We know imbalanced data is a problem for every machine learning algorithm. The algorithms have a hard time classifying on imbalanced data. Our training set contains 17% fraud transactions and almost 83% genuine transactions as we can see from figure 4.1. Considering our data is highly imbalanced, we need to perform some sort of balancing so that our model could work efficiently and give us the desired output. There are many balancing techniques and SMOTE is known to be popular for the classification problems. Here, we will oversample the minority class and we will use SMOTE to do so. Our preprocessing is almost done with these steps. Now, we can use the chosen classifiers and our proposed model on this pre-processed data.

4.3 Methodology

For the model training, we first tried out eight different popular pre-built classifiers. Using the sci-kit-learn library, the code starts by importing the necessary libraries

and modules for each model, then, it initializes each model with default or specific hyperparameters. After selecting these classifiers (KNN, MLP, AdaBoost, Random Forest, SVC, GaussianNB, Logistic Regression, Decision Tree) we proceeded to train the models using the training set. To get the best results we fine tuned selected models by experimenting with different hyperparameters to improve the performance. To assess the models, we evaluated the performance of each model and included two functions which are evaluation_score and cross_validation. The evaluation_score function calculates the accuracy, precision, recall, and f1-score of the model. The models are then fitted on the training set. Afterwards, the model is used to make predictions on the test set using the predict() method. The cross-validation function performs k-fold cross-validation on the training set to evaluate the model's performance and to reduce the variance in the model's performance. We also generated confusion matrix and ROC curve with AUC score, so that, we could better define which classifiers performed the best.

To achieve better accuracy, after trying out those built-in classifiers we tried to implement a neural network(figure 4.2) approach to the problem, importing the necessary libraries and modules for building a neural network. We built the Neural Network with 3 layers with the input layer having neurons equivalent to the number of features and the hidden layers having 8 neurons and the output layer with 1 neuron. We found that using ReLU in the activation function in the hidden layers of the Neural Network gave the best performance as it was computationally efficient. The sigmoid function was used with the output layer of the network as this is a classification problem.

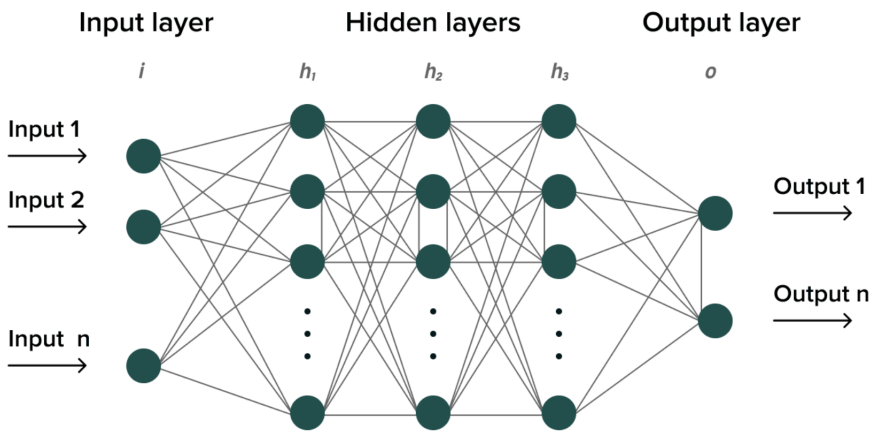


Figure 4.2: Neural Network Architecture

The model is then fit to the training data using the fit() method. The fit() method takes in the training data, the number of epochs, and a verbosity level, which controls how much information is printed during training. As the training is happening the model's parameters are updated and the errors decrease with an iteration of each epoch meaning the model is learning the way it is supposed to. As this training process involves adjusting the model's parameters, such as, the weights and biases of the neurons, the goal is to minimize the error between the predicted and the true output. The error is typically measured using a loss function, such as, mean squared

error. The model's parameters are updated in order to reduce the error and improve the model's performance on the training data, resulting in a reduction of errors with each iteration of an epoch. The process of updating the model's parameters and reducing the error is repeated for multiple iterations until the model reaches a satisfactory level of performance or a stopping criterion is met. So we iteratively tried increasing and decreasing the epoch values to find the most optimal point.

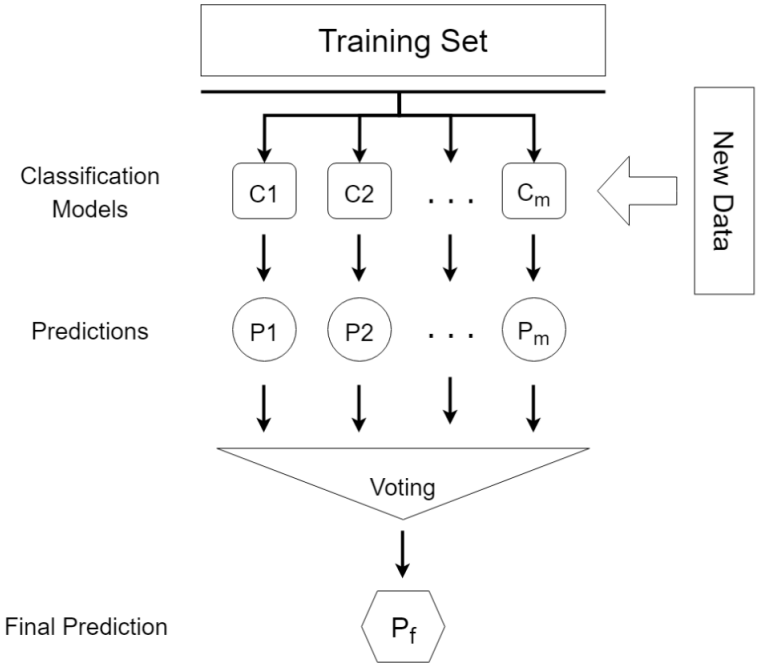


Figure 4.3: EnsembleVoteClassifier

It is assumed that any hybrid model-based approach produces more reliable results compared to any single classifier model. An ensemble model voting classifier is a type of ensemble learning method in which multiple models are trained on the same dataset and their predictions are combined to make a final prediction. The most common way of combining the predictions is through a majority voting system, where the class that receives the most votes is the final prediction as we can see in figure 4.3. This approach can lead to better performance compared to using a single model alone, as the diversity of the different models can reduce overfitting and improve generalization. It is important to note that the ensemble model's performance can be affected by the choice of base classifiers, the ensemble method used and the parameters used in the ensemble model.

Generally, in a voting classifier, each individual classifier in the ensemble makes its own predictions and the ensemble model then makes a final prediction based on the majority vote of the individual classifiers. We used the Random Forest and AdaBoost classifier to get the best optimal performance which is also our proposed model. During the experiment, we also tried out using different classifiers and up to 4 classifiers at once in the ensemble voting. It's expected to perform better than any single individual classifier because it combines the predictions of multiple models, the ensemble is less likely to make errors and can provide a more robust estimate

of the true class label. In a voting classifier, there are different ways to combine the predictions of the individual classifiers, such as by taking a majority vote (hard voting) or by averaging the predicted probabilities (soft voting). By trial and error, we found that by using soft voting we could find the best performance. We set the weights parameter to [2, 1] after some experimentation, which means that the first classifier in the ensemble will have a weight of 2 and the second classifier will have a weight of 1. This means that the predictions of the first classifier will be given twice the importance or weight when making the final ensemble prediction, compared to the predictions of the second classifier. We also tried to use, a weight of 1:1 for the classifier at first but it did not lead to better accuracy. The evaluation_score function is used to evaluate the performance of the ensemble model by calculating the accuracy, precision, recall, and f1-score.

4.4 Result Analysis

From table-3.1 The results of the model evaluation show that the Random Forest model performed the best with an accuracy of 0.93478, followed by AdaBoost with an accuracy of 0.9106. The AUC-ROC score is a measure of the model's ability to distinguish between the positive and negative classes. The Random Forest model has the highest AUC-ROC score of 0.887596, followed by AdaBoost with an AUC-ROC score of 0.875968. Overall, Random Forest gave out the best result out of the popular built-in library classifiers. The Random Forest algorithm works by training multiple decision trees on a dataset and then averaging their predictions to make the final prediction. By increasing the number of decision trees in the forest, the model becomes more robust and accurate. In the Table, we can see that The KNN model has an accuracy score of 0.891125, setting the k-nearest neighbors classification to 3. A smaller value of k will make the model more sensitive to noise in the data but also more flexible, while a larger value of k will make the model more robust to noise but less flexible. We also tried out using nearest points ranging from 2 to 9 but by using 3 we extracted the best optimum performance.

From figure 3.4, we can compare that the neural network model could not outperform AdaBoost or Random Forest. We tried to test the epoch values up to 300 and calculate the loss and accuracy of the model. It was found that the model gave the best accuracy when it is trained between 150 to 200 epochs. While the training process was initiated, we saw the training loss to be 0.5634 for the 1st epoch. But, when it reached the 150th epoch, the loss went down to 0.1692. The loss gradually decreased meaning our model was learning the way it is expected to learn. We also achieved a training accuracy of 0.9437 after the 150th epoch. The highest accuracy score we could get after testing was 0.91304. It is important to consider other evaluation metrics like precision, recall, and F1-score, as well as the confusion matrix, balanced accuracy, AUC-ROC etc when comparing the performance. From the table-3.1, Figures 3.4 and 3.5 we can see that Random Forest beat the Neural Network model on all factors. It had better TP with 0.94 to NN's 0.92, better precision with 0.74 to NN's 0.70, and also a better f1 score of 0.79 to NN's 0.75.

Finally, trying out the ensemble model that combines the Random Forest and AdaBoost model with a weight ratio of 2:1, we got an accuracy of 94.57% which is

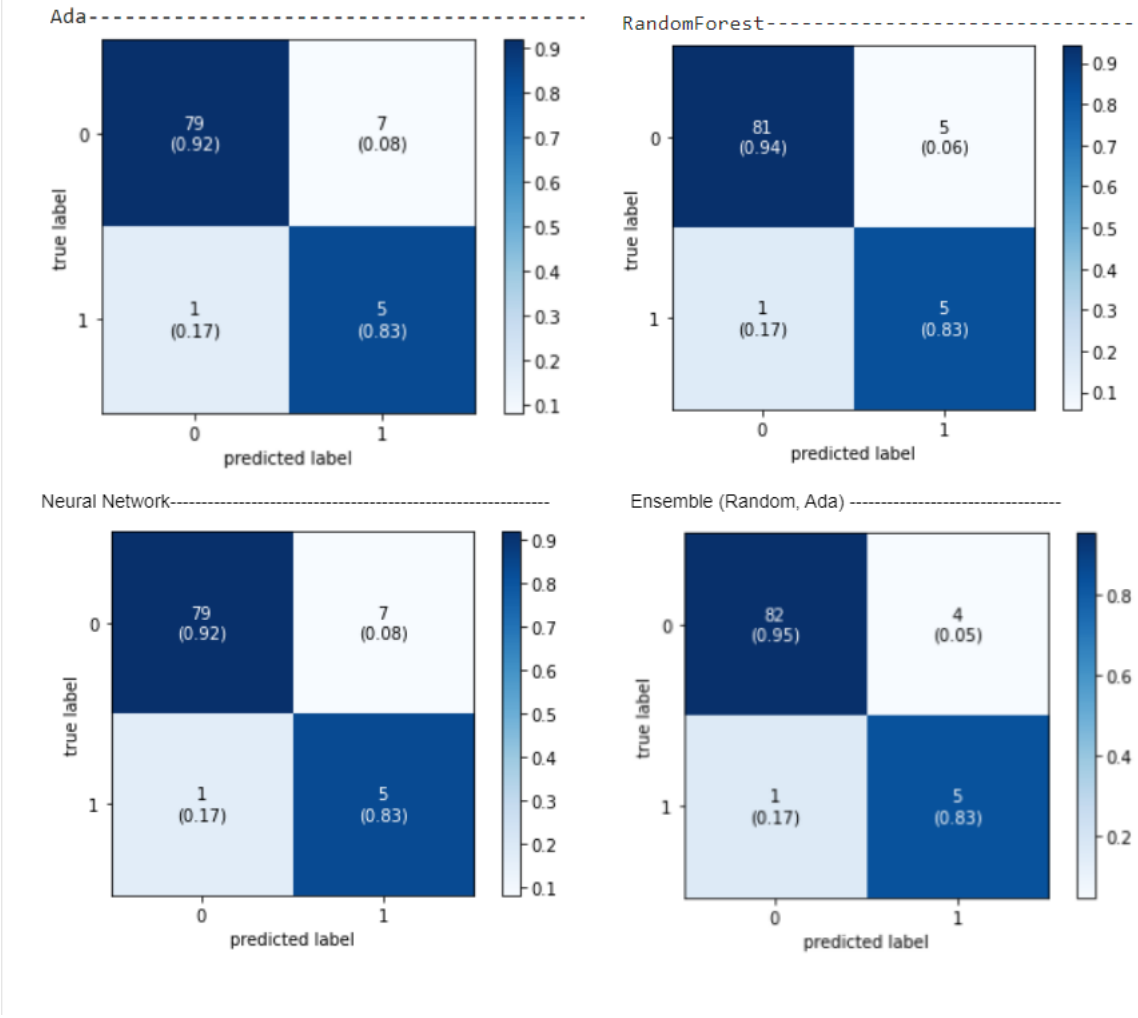


Figure 4.4: Confusion Matrix

better than all the classifiers we tried before. The confusion matrix shows TP of 0.95 and TN of 0.83. The recall accuracy score, also known as the true positive rate, is 95%, which means that out of all the positive instances, the model is able to correctly identify 95% of them. This accuracy score is better than the AdaBoost model where the recall accuracy was around 91%. The precision score, or the ratio of true positives to true positives plus false positives, is 0.77, indicating that the model correctly identifies 77% of the positive instances it predicts. Prior to implementing the EnsembleVoteClassifier, the precision value of 0.70 had a dominance on the table. Implementing the hybrid model had a significant increase of 7% precision against the dominance hierarchy. The F1 score, which is the harmonic mean of the precision and recall scores, is 0.82 which is the highest we achieved. The closer the F1 score is to 1, the better the model is considered to be. Thus, our hybrid model is the closest to this threshold compared to other classifiers as we can see from table 3.1. The AUC-ROC score, measures the model's ability to discriminate between positive and negative classes, is 89.34% which can be seen from figure 3.5. As mentioned before all of the result we have discussed was with a weight ratio of 2:1. But, we also tried with the ratio set to 1:1. In fact, by compiling the 1:1 weighted model, we

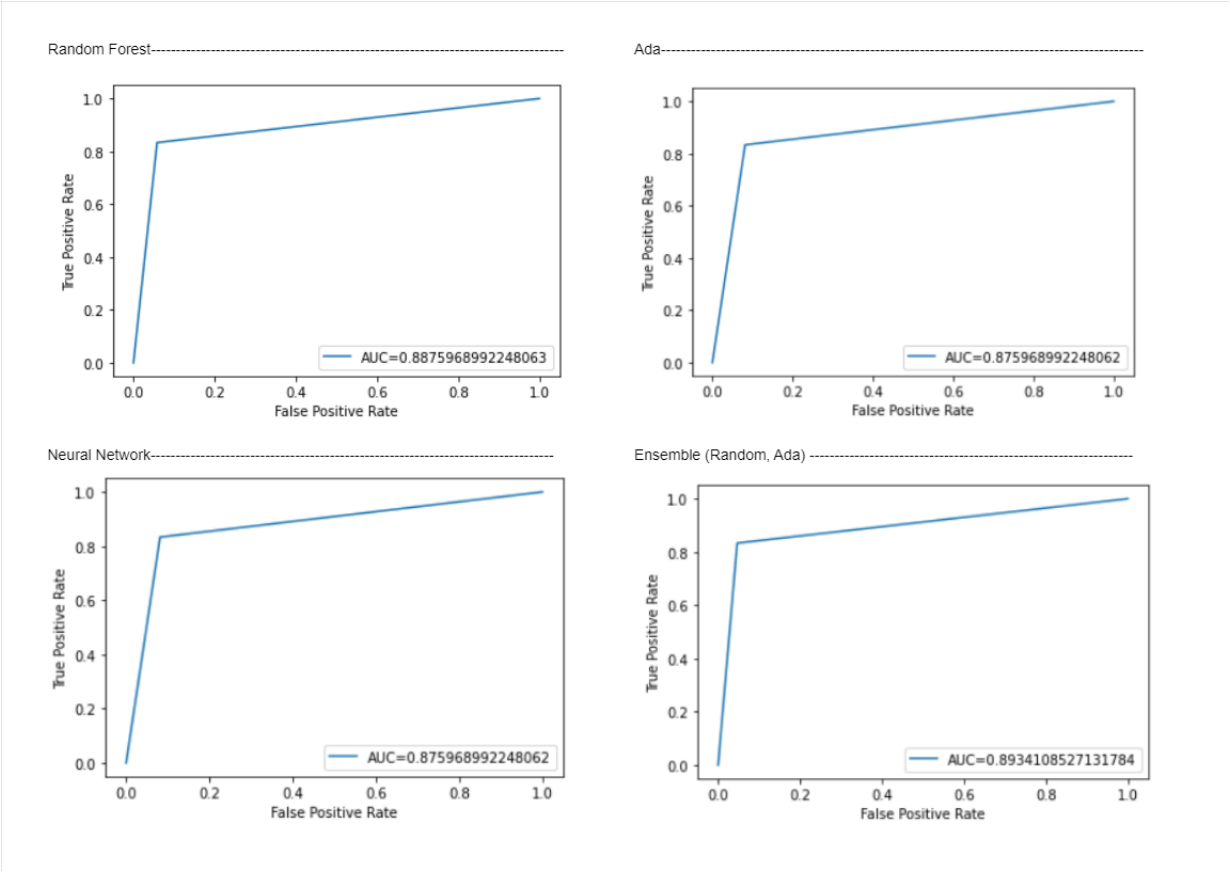


Figure 4.5: ROC curve

see a significant difference in accuracy which decreased from 94.57% to 93.47% and also the F1 score was down to 0.79. The biggest difference we saw while using the 2:1 weighted model was the label '1' precision went up from 0.50 to 0.56 which is a significant upgrade. Overall, the ensemble model combining Random Forest and AdaBoost shows a good performance in terms of accuracy, balanced accuracy, recall accuracy, precision, F1 score, and AUC-ROC score.

Approach	Average Score	F1 Score	Precision	Standard Deviation
Ensemble (Random, Ada)	**0.94565**	0.82	0.77	0.0504
RandomForest	0.93478	0.79	0.74	0.0518
Neural Network	0.91304	0.75	0.70	0.0626
AdaBoost	0.91060	0.75	0.70	0.0721
DecisionTree Classifier	0.87727	0.75	0.70	0.0862
KNN	0.89112	0.74	0.69	0.828
Logistic Regression	0.86320	0.75	0.70	0.1012
SVC	0.86406	0.73	0.69	0.0631
MLP	0.84891	0.77	0.72	0.0837
GaussianNB	0.38961	0.28	0.54	0.1217

Table 4.1: Result Analysis Table

Chapter 5

Preliminary Analysis: Case study

5.1 The game

Axie Infinity is an online video game developed by Sky Mavis which is based on NFT. The in-game economy uses Ethereum as the base of crypto currency. Axis is an in-game pet that represents the players minted and collected NFT's. You can breed these creatures and also send them to battle in the game. Sky Mavis, the developer of this popular non-fungible token (NFT) video game got hacked back on March 23 and lost hundreds of million dollars in assets. The attackers attacked the ronin bridge chain which is a part of the Ronin Network sidechain. So, the bridge is a way to take cryptocurrencies from one network to another in case we need it in both the networks.

5.2 Hack mechanism

It was built on the Ronin chain network and was secured by 9 validators who were responsible for approving any deposit or withdrawal. Out of these 9 validators, 5 of them (majority) had to be in consensus to approve a transaction. Ronin was launched as an ethereum side chain to provide users a fast and cheap transaction throughput.

5.3 Main reason

The role of validator nodes is to communicate with each other and perform votes on different possible chain states which we discussed while explaining validator nodes in chapter 2. Whichever chain has 2/3rd plus votes (maximum support) for any new proposed update, the validator nodes will approve that new change to the chain. Increasing the number of validator nodes in a decentralized system can help to improve the security and integrity of the system. Validator nodes are responsible for verifying and validating transactions and other actions within the system and adding more validator nodes can increase the overall redundancy and resilience of the system. However, it is important to note that increasing the number of validator nodes alone is not a guarantee that the system will be secure and free from hacking. Other security measures, such as implementing robust protocols and technologies, conducting regular security assessments, and providing secure infrastructure and hardware

are also important for protecting against hacking and other types of malicious activity. It is also important to ensure that the validator nodes themselves are secure and properly configured, as a compromised validator node could potentially pose a threat to the system. Ensuring the security of validator nodes may involve measures such as implementing secure coding practices, providing secure infrastructure and hardware and conducting regular security assessments.

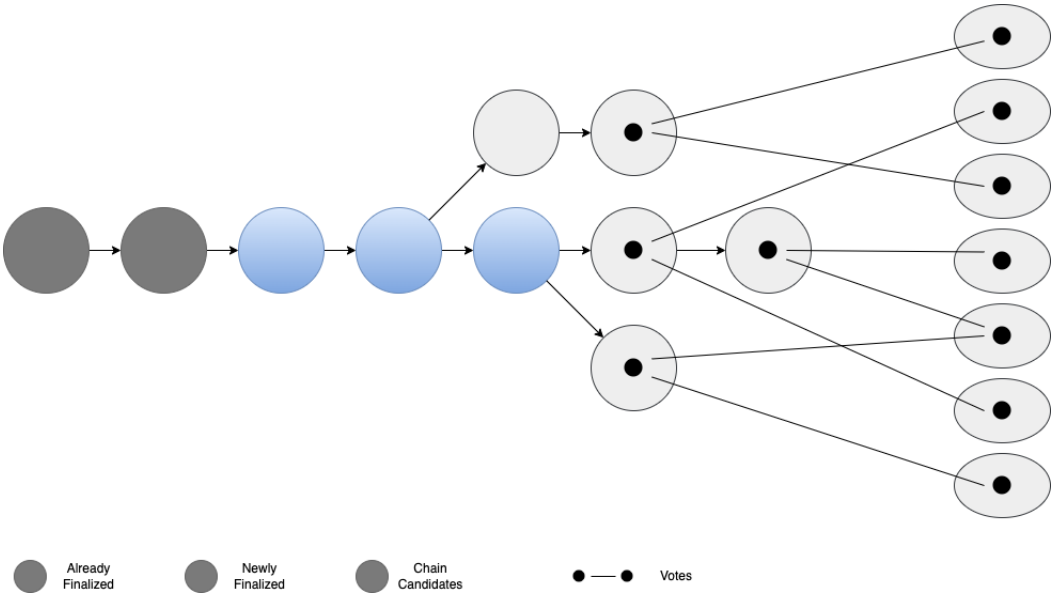


Figure 5.1: Finalizing Nodes

For Axie Infinity, the breach occurred when attackers gained control of a series of those validator nodes. Nine validator nodes existed for recognition of withdrawal and the breach occurred when five of those nine nodes were compromised. The hackers conducted fake withdrawals using those hacked nodes. The hackers had stolen 173,600 Ethereum and 25.5 million USD Coins, which are worth approximately $620 million during that time. They had somehow gained control of five nodes through hacked private keys and a backdoor which is used for a fifth node that is controlled by the decentralized autonomous organization (DAO) of Axie Infinity. They tried to address the vulnerability by adding more nodes and also by imposing a zero-trust policy. The attackers used the vulnerability and the back door was found through the gas-free RPC nodes and got the signature of the DAO validator though the validator key scheme was set up in a decentralized way to prevent such attacks. Since some nodes were not able to catch up with the chain or were stuck in the synching state, five out of nine nodes were selected as threshold initially.

5.4 Snowball Algorithm

Snowball algorithm can be used to identify that increasing the threshold can increase the security of a decentralized system.

pseudo Code:

```
preference => Node1
```

```
consecutiveSuccesses => 0
while not decided:
 ask n sample node their preference
 if (>= x) give the same response:
   preference => a
   if preference == old preference:
     consecutiveSuccesses++
   else:
     consecutiveSuccesses = 1
 else:
   consecutiveSuccesses = 0
 if consecutiveSuccesses > y:
   decide(preference);
```

At the initial point we select the preference to be Node1 over Node2 and set the consecutiveSuccesses to 0. Now the system will query n sample number of nodes and ask them for preference. If x or more Node give the same response to change to a different preference, the preference will be changed to the new preference and set the consecutiveSuccesses to 0. a Is denoted as the quorum size. If the preference is same as the old preference, the consecutiveSuccesses will be set to 1 to indicate it is the old preference.

Every node in the system repeats the preference until they get a quorum for the same response y times in a sequence. So, if one node selects Node1, then every other node following this protocol will also eventually pick Node1.
Here, random change to the preference by random sampling causes a network to prefer one node at a time. This leads the nodes to choose a network preference that can become irreversible and then the nodes can decide which node to further prefer. For example, in the given figure, there is only a choice between Node1 or Node2 to select, but the Snowball can execute consensus on decisions with numerous possible nodes.

The parameters can be set high for a better safety threshold. By increasing the quorum size x, we can increase the safety threshold, but due to this the liveness threshold decreases. This increases the tolerance to protect from malicious nodes and helps to remain the network safe. So, all nodes will eventually acknowledge which node to accept or reject. The expected number of malicious participants that may be allowed before the protocol could no longer advance is known as the liveness threshold.

The snowball can be scaled as the number of nodes on the network can be increased. Regardless of the number of total participants, the number of consensus messages sent stays the same because while querying, a node tend to query around 20 nodes, even though there can be thousands of nodes in the whole network.

5.5 Countermeasure

For preventing any further short-term damage of hackers, the developers increased the validator nodes threshold which was from five till eight. Because the more validator nodes exist, the more difficult it becomes for any hacker to gain access. More validator nodes would have ensured multiple layers of security, making it hard to get access to the private keys. What the validators do is that they start to talk to each other to say that they are the views of different chain states and vote on which is the best chain state. The best chain is determined by (2/3) + 1 vote.

Chapter 6

Conclusion

To conclude, in the gaming industry, NFT is providing uniqueness making the assets more programmable and secure. As all the transactions are made on the ethereum network, building a Machine learning based hybrid model to detect fraud transactions on ethereum can help prevent a lot of these mentioned attack at it's root. The gaming industry will gain a new momentum by ensuring a threat free and secure environment.
The fraud detection model we have implemented has an accuracy of almost 95%. We aim to improve our accuracy and make our model more precise, which will work better to lessen the chances of fraud. Moreover, better hybrid models using more complex structure could be used to improve accuracy and robustness. Deep learning could also be used to create a better fraud detection system instead of using general classifiers or simple hybrid models. Another lack of our research was the lack of data related to fraud transactions on Ethereum. With more data the results could be better with our solution and all the possible solutions mentioned above. Furthermore, we have discussed different types of attacks and their mitigations theoretically. So, our idea is that a blockchain based implementation can be done where all the mitigations of these attacks will be implemented for ensuring a fully secure system. While implementing we need to take into account the capabilities of blockchain systems, as they are restricted to handle limited number of transections per second. Also due to lack of interoperability between blockchain systems which we have seen in case of Axie Infinity it can be difficult for different systems to communicate with each other simultaneously. In addition to that, we have shown the hack mechanism of a related case study by increasing or decreasing the validator nodes throughout the snowball algorithm. By implementing the snowball algorithm on a blockchain system we can further test it in real time. Finally in our research, we could not give any practical implications of NFT authentication, so we can also work on this.

Bibliography

[1] J. R. Douceur, "The sybil attack," in *Peer-to-Peer Systems*, Springer Berlin Heidelberg, 2002, pp. 251–260. DOI: 10.1007/3-540-45748-8_24. [Online]. Available: https://doi.org/10.1007/3-540-45748-8_24.

[2] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, "Sybil-resistant mixing for bitcoin," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, ser. WPES '14, Scottsdale, Arizona, USA: Association for Computing Machinery, 2014, pp. 149–158, ISBN: 9781450331487. DOI: 10.1145/2665943.2665955. [Online]. Available: https://doi.org/10.1145/2665943.2665955.

[3] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on {bitcoin's}{peer-to-peer} network," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 129–144.

[4] A. Dika, "Ethereum smart contracts: Security vulnerabilities and security tools," M.S. thesis, NTNU, 2017.

[5] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *International conference on financial cryptography and data security*, Springer, 2017, pp. 357–375.

[6] Q. Wang, J. Yu, S. Chen, and Y. Xiang, "Sok: Diving into dag-based blockchain systems," *CoRR*, vol. abs/2012.06128, 2020. arXiv: 2012.06128. [Online]. Available: https://arxiv.org/abs/2012.06128.

[7] S. Linoy, S. Ray, and N. Stakhanova, "Etherprov: Provenance-aware detection, analysis, and mitigation of ethereum smart contract security issues," in *2021 IEEE International Conference on Blockchain (Blockchain)*, 2021, pp. 1–10. DOI: 10.1109/Blockchain53845.2021.00014.

[8] Z. Pan, T. Hu, C. Qian, and B. Li, "Redefender: A tool for detecting reentrancy vulnerabilities in smart contracts effectively," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, 2021, pp. 915–925. DOI: 10.1109/QRS54544.2021.00101.

[9] M. . Astardjieva. "Decentraland guide: How to make money in the play-to-earn crypto game..." (Mar. 17, 2022), [Online]. Available: https://www.the-sun.com/tech/4494680/decentraland-how-to-make-money-crypto/ (visited on 09/18/2022).

[10] A. . Boverman. "Timejacking bitcoin." (Sep. 18, 2022), [Online]. Available: http://culubas.blogspot.com/2011/05/timejacking-bitcoin_802.html (visited on 09/18/2022).

[11] A. . Rustaggi. “For the uninitiated, what are the risks and rewards associated with web3 and nft gaming.” (Mar. 30, 2022), [Online]. Available: https://timesofindia.indiatimes.com/blogs/voices/for-the-uninitiated-what-are-the-risks-and-rewards-associated-with-web3-and-nft-gaming/ (visited on 09/18/2022).

[12] “Top nft games (updated for 2022).” (Apr. 25, 2022), [Online]. Available: https://coinmarketcap.com/alexandria/article/top-nft-games (visited on 09/18/2022).

[13] “2% of non-fungible token (nft) trades globally are manipulated; how to safely tread the market.” (), [Online]. Available: https://www.outlookindia.com/business/2-of-non-fungible-token-trades-globally-are-manipulated-how-to-safely-tread-the-market-news-219849.

[14] “Aavegotchi - enter the gotchiverse.” (), [Online]. Available: https://www.aavegotchi.com/ (visited on 09/18/2022).

[15] “Ethereum fraud detection dataset. kaggle.” (), [Online]. Available: https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset (visited on 09/18/2022).

[16] H. Patel. “Isolation of non-fungible tokens from the decentralize applications business center: The case of ethereum block chains.” (), [Online]. Available: https://www.irjet.net/archives/V9/i11/IRJET-V9I11171.pdf.

[17] “Rarible: Low fee nft marketplace — buy nfts on 5 blockchains.” (), [Online]. Available: https://rarible.com/ (visited on 09/18/2022).

[18] Somnium Space Ltd. “Somnium space.” (), [Online]. Available: https://somniumspace.com/ (visited on 09/18/2022).

[19] “Sorare.” (), [Online]. Available: https://sorare.com/ (visited on 09/18/2022).

[20] “Superrare — nft art — nft art marketplace — digital art.” (), [Online]. Available: https://superrare.com/ (visited on 09/18/2022).

[21] “Welcome to decentraland.” (), [Online]. Available: https://decentraland.org/ (visited on 09/18/2022).