Understanding Security Issues in the NFT Ecosystem

Dipanjan Das
University of California, Santa Barbara
Santa Barbara, California, USA
dipanjan@cs.ucsb.edu

Priyanka Bose
University of California, Santa Barbara
Santa Barbara, California, USA
priyanka@cs.ucsb.edu

Nicola Ruaro
University of California, Santa Barbara
Santa Barbara, California, USA
ruaronicola@cs.ucsb.edu

Christopher Kruegel
University of California, Santa Barbara
Santa Barbara, California, USA
chris@cs.ucsb.edu

Giovanni Vigna
University of California, Santa Barbara
Santa Barbara, California, USA
vigna@cs.ucsb.edu

ABSTRACT

Non-Fungible Tokens (NFTs) have emerged as a way to collect digital art as well as an investment vehicle. Despite having been popularized only recently, NFT markets have witnessed several high-profile (and high-value) asset sales and a tremendous growth in trading volumes over the last year. Unfortunately, these marketplaces have not yet received much security scrutiny. Instead, most academic research has focused on attacks against decentralized finance (DeFi) protocols and automated techniques to detect smart contract vulnerabilities. To the best of our knowledge, we are the first to study the market dynamics and security issues of the multi-billion dollar NFT ecosystem.

In this paper, we first present a systematic overview of how the NFT ecosystem works, and we identify three major actors: marketplaces, external entities, and users. We then perform an in-depth analysis of the top 8 marketplaces (ranked by transaction volume) to discover potential issues, many of which can lead to substantial financial losses. We also collected a large amount of asset and event data pertaining to the NFTs being traded in the examined marketplaces. We automatically analyze this data to understand how the entities external to the blockchain are able to interfere with NFT markets, leading to serious consequences, and quantify the malicious trading behaviors carried out by users under the cloak of anonymity.

CCS CONCEPTS

• **Security and privacy → Web application security**.

KEYWORDS

Non-Fungible Token (NFT); Blockchain; Decentralized Finance (DeFi)

ACM Reference Format:
Dipanjan Das, Priyanka Bose, Nicola Ruaro, Christopher Kruegel, and Giovanni Vigna. 2022. Understanding Security Issues in the NFT Ecosystem. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3548606.3559342

CCS '22, November 7–11, 2022, Los Angeles, CA, USA.
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9450-5/22/11.
https://doi.org/10.1145/3548606.3559342

1 INTRODUCTION

A *Non-Fungible Token* (NFT) is an ownership record stored on a blockchain (such as the Ethereum blockchain). While digital items, such as pictures and videos, are the most common assets traded as NFTs, the sale of physical assets, *e.g.*, postal stamps [29, 41], gold [43], real estate [33], physical artwork [16], *etc.*, is also steadily gaining popularity. In the cryptocurrency world, an NFT is the equivalent of a conventional proof-of-purchase, such as a paper invoice or an electronic receipt. Among other things, what make NFTs attractive are *verifiability* and *trustless transfer* [67]. Verifiability means that sales are recorded as blockchain transactions, which makes tracking of ownership possible. In addition, the NFT concept allows for the trading of digital assets between two mutually distrusting parties, as both the crypto payment and the asset transfer happen atomically in a single transaction.

Several NFT marketplaces (NFTMs), *e.g.*, OpenSea, Rarible, and Axie, emerged in recent years to facilitate buying and selling NFTs. This has sparked the interest of both crypto art collectors and traders. To put things into perspective, OpenSea, the largest NFTM, collected $236M USD in platform fees generated out of a trading volume of $3.5B USD [11] in August 2021 alone. This is around half of the volume [21] generated by the e-commerce giant eBay during the same period. And the all-time combined trading volume of the top three NFTMs—OpenSea, Axie, and CryptoPunks—surpassed $10B USD in September 2021 [9]. Individual NFT sales have also skyrocketed in recent months [59], with nine out of ten of the most expensive sales [1] taking place between February and August 2021. For example, the media widely reported on the digital artist Beeple, who sold an art piece for $69.3M USD; as another example, the first tweet of Twitter CEO Jack Dorsey was sold for $2.9M USD. Also, NFTMs have surfaced as the most *gas*-eating Ethereum contracts. For example, OpenSea made it to the top of the list of *gas-guzzlers* in Etherscan [45], consuming around 20% of the gas spent by the network.

As the NFT space exploded with multi-million dollar sales, cybercriminals and scammers have inevitably flocked to the markets to make quick profits and cheat unsuspecting users. As a result, numerous NFT scams also made recent headlines.

Legitimacy is one of the big issues with NFTs, as nothing prevents an impostor from "tokenizing" and selling someone else's art, while the creator remains oblivious of the fraud. With the current state of affairs, the onus of verifying the token is on the buyer. Unfortunately, this is not always easy. For instance, in August 2021, a perpetrator impersonated the popular British graffiti artist Banksy and sold an NFT [15] that featured a "fake" art piece by the artist for $336K USD through an online auction. While NFTMs try to thwart such attacks by mandating account validation, typically through an artist's social media presence, another scammer punched a hole through Rarible's verification process and managed to get a fake account associated with the renowned artist Derek Laufman verified [27]. Counterfeits NFTs, also called *copycats* or *parody* projects, resemble reputable collections and purport to have been created by reputable sources. For example, the early NFT project CryptoPunks has numerous clones, such as CryptoPhunks. In some scenarios, scammers set up unauthorized customer support channels and social media accounts that pretend to be affiliated with NFTMs in an effort to steal customer information and compromise accounts [35]. Also, there is evidence of *rug-pulls*, where the owner/creator of an NFT unscrupulously hypes an asset in order to inflate its value, only to cash out, leaving others to suffer from the subsequent decline in value. One such example is the Eternal Beings collection, which was promoted by the popular American rapper Lil Uzi Vert through his Twitter account with 8.5M followers. Soon after the initial investment by the buyers, he deleted all of his tweets, causing the token values to plummet [36].

With enormous funds flowing into *decentralized finance* (DeFi) applications, scams have become lucrative money-making opportunities. Previous research studied several different aspects of crypto-economic attacks, *e.g.*, financial repercussions due to transaction reordering [47, 49, 61, 71], flash loan abuse [62], arbitrage opportunities [70], and pump-and-dump schemes [51, 56, 68]. Besides protocol attacks, there also exists a substantial body of work on automated detection of smart contract vulnerabilities, *e.g.*, reentrancy, transaction order dependence, integer overflows, and unhandled exceptions [12, 26, 44, 46, 50, 52–55, 58, 60, 64, 69].

To the best of our knowledge, however, the existing literature has not explored the security challenges in the emerging NFT ecosystem, or performed a systematic and comprehensive analysis of the associated threats. Our work fills that void. First, we identify three components constituting the NFT ecosystem. We then analyze each component to discover security, privacy, and usability issues, as well as economic threats. We hope that our work will be helpful both for NFT marketplaces and their users. We envision this paper as a guide to help NFTMs to avoid mistakes while making users aware of the perils of the NFT space. In particular, we make the following contributions:

Anatomy of the NFT ecosystem. We systematize the NFT ecosystem, looking at the participating actors —marketplaces, external entities, and users— and we analyze their mutual interactions (Section 3).

Comprehensive data collection. We leverage multiple sources of data, including the Ethereum blockchain, as well as asset and event data sourced from the NFTM dApps, to paint a holistic picture of how the ecosystem operates (Section 4).

Identifying irregularities in NFTMs. We identify flaws in the NFTM designs, which, if abused, pose a significant financial risk.

Identifying issues with external entities. We identify the off-chain external entities connected to the NFT ecosystem, and how such entities can pose threats to users (Section 6).

Uncovering malicious user behaviors. We discover and quantify trading malpractices, such as wash trading, shill bidding, and bid shielding, which are taking place in the top marketplaces. The insight drawn from our analysis sheds light on some of the prime factors responsible for driving up the recent NFT frenzy (Section 7).

Releasing code and data. We will open-source our analysis framework along with the data we collected to help researchers uncover further interesting insights about the emerging NFT economy.

2 BACKGROUND

In this section, we introduce the building blocks of the Ethereum ecosystem, with an emphasis on non-fungible tokens (NFTs) and the economy that has grown around them.

The Ethereum Blockchain. Ethereum is the technology powering the cryptocurrency Ether (ETH) and thousands of decentralized applications (dApps). The Ethereum blockchain is a distributed, public ledger where transactions are mined into *blocks* by *miners* who solve cryptographic Proof of Work (PoW) challenges. In this ecosystem, an *account* is an entity represented by an address that is capable of submitting transactions. There are two types of accounts in Ethereum: externally owned accounts (EOA), which are controlled by anyone holding the corresponding private key, and contract accounts, which contain executable pieces of code, called smart contracts. A smart contract is a program run by the Ethereum Virtual Machine (EVM), which leverages the blockchain to store its persistent state. A *transaction* is the transfer of funds between accounts, or an invocation of a contract's public method. The address that sends the funds or interacts with the contract is denoted by `msg.sender`.

Non-Fungible Token (NFT). In the real world, tokens are representations of facts, such as the position in a queue or the authorization to access a facility. In Ethereum, tokens are digital assets built on top of the blockchain. Unlike Ether, which is the native (built-in) cryptocurrency of the Ethereum blockchain, tokens are implemented by specialized smart contracts. There are two main types of tokens: fungible and non-fungible. All the copies of a *fungible* token, usually conforming to the ERC-20 interface [66], are identical and interchangeable. Such tokens can act as a secondary currency within the ecosystem, or can represent someone's stake in an investment. On the other hand, all the copies of *non-fungible* tokens, usually conforming to the ERC-721 [48] interface, are unique, and each token represents someone's ownership of a specific digital asset, such as ENS domains [14] and CryptoKitties [7], or a physical asset, like a gold bar.

ERC-721 [48] is by far the most popular standard for implementing non-fungible tokens on Ethereum. The standard interface defines a set of mandatory and optional API methods that a token contract needs to implement. Figure 1 presents a few of those API methods relevant to our discussion.

```
setApprovalForAll(address _operator,
                  bool _approved) external
approve (address _approved,
         uint256 _tokenId) external payable
transferFrom (address _from, address _to,
              uint256 tokenId) external payable
tokenURI (uint256 _tokenId)
          external view returns (string)
```

Figure 1: Important methods defined in ERC-721.

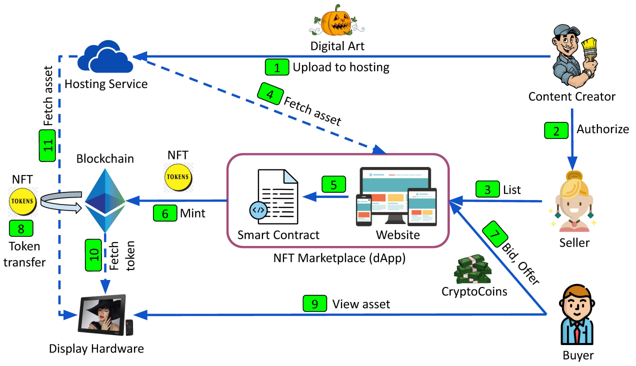

Figure 2: Anatomy of the NFT ecosystem showing all the marketplace actors, and their mutual interaction. The dotted and solid lines indicate data-centric and command-centric communication channels, respectively.

Each NFT has its own ID (to keep track of these unique tokens), which is referred to as _tokenId. In ERC-721, an *operator* is an entity that can manage all of an NFT owner's assets. In other words, an NFT owner can delegate the authority to act on her assets to an operator. Depending on whether the _approved argument is set, the setApprovalForAll() method either adds or removes the address _operator from/to the set of the operators authorized by the msg.sender (the NFT's owner). Unlike an operator, who can operate on all the assets of an owner, ERC-721 defines a *controller* as an entity who is authorized to operate on one single asset held by an owner. The approve() method approves the address _approved as the controller of the asset _tokenId. An operator, a controller, or the owner can call the transferFrom() method to transfer the token _tokenId from the current owner's _from address to the _to address.

When an NFT is created (minted), the creator can optionally associate a URL with the NFT. That URL, called metadata_url, should point to a JSON file that conforms to the ERC-721 Metadata JSON Schema [48]. The JSON file stores the details of the asset, *e.g.*, its name and description, and also contains an image field storing a URL, called image_url, that points to the asset. In this way, an NFT essentially connects an asset with the record of its ownership. Given a _tokenId, the associated metadata_url can be retrieved by querying the tokenURI() API of the contract. Interestingly, the creation and destruction of NFTs ("minting" and "burning") are not a part of the standard. Typically, mint() is defined as a public function restricted to the contract creator, and invoked by passing metadata_url as an argument. Minting can also be done during contract creation by calling mint() through the contract's constructor.

InterPlanetary File System (IPFS). IPFS [23] is a distributed, peer-to-peer, permissionless file system. Anyone can join the IPFS overlay network. A data item d is assigned a unique immutable address, also known as content identifier (CID): $cid = H(d)$, which is the hash H of the file's content d. Therefore, when the content of the file changes, the CID changes as well. The content of a file is first split into blocks. All the storage elements, *i.e.*, a directory, the files inside the directory, and the blocks within those files, are stored in a directed acyclic graph structure called a Merkle DAG. IPFS maintains a distributed hash table (DHT) split across all the nodes in the network to store provider records, which locate those peers that store the requested content. To retrieve a data item d, a node first looks up the providers $P(d)$ in the DHT, and then requests d from the members of $P(d)$.

3 ANATOMY OF THE NFT ECOSYSTEM

In this section, we provide an overview (Figure 2) of the economy that has developed around NFTs. Specifically, we identify the actors that participate in the ecosystem and the components they interact with.

Users. NFTs are often used to sell digital collectibles and artwork, *e.g.*, images, audio files, and videos. The users in the NFT ecosystem belong to one of three categories: *content creator*, *seller*, and *buyer*. First, the creators create digital content and upload it 1 to hosting services (an *external entity*) to make the art publicly available. When it comes to selling the content, some creators are not technical enough to turn their art into an NFT, and put it as a token on the blockchain. Therefore, they authorize 2 sellers to mint NFTs 6 and offer it on marketplaces. In other cases, a content creator is also taking the role of the seller. Once listed on a marketplace 3, buyers can buy the artwork at a listed price, make offers, or place bids 7. If their offer is accepted or they win an auction, the NFT is transferred 8 by invoking the transferFrom() API (Section 2) from the seller to the buyer to reflect the change in ownership.

Marketplaces. NFT marketplaces (NFTM) are dApp platforms where NFTs (also referred to as *assets*) are traded. There are typically two main components of an NFTM—a user-facing web frontend, and a collection of smart contracts that interact with the blockchain. Users interact with the web app, which, in turn, sends transactions to the smart contracts on their behalf 5. Primarily, there are two types of contracts: **(i)** marketplace contracts, which implement the part of the NFTM protocol that interacts with the blockchain, and **(ii)** token contracts, which manage NFTs. Marketplaces typically allow users to perform the following activities: **(a)** user authentication, **(b)** token minting, **(c)** token listing, and **(d)** token trading. The token-related activities are collectively called *events*. Depending on where these events are stored, three broad types of NFTM protocol design are possible: **(i)** *on-chain*: all the events live on the blockchain. Since every action costs gas, this design makes the NFTM operationally expensive for the users. NFTMs that follow this design include Axie, CryptoPunks, Foundation, and SuperRare. **(ii)** *off-chain*: the events are recorded in a centralized, off-chain database managed by the NFTM. Users perform various activities by interacting with the web app, not the blockchain, and, therefore, this design is gas-friendly. Nifty is an example of an off-chain NFTM. **(iii)** *hybrid*: depending on their type, events are stored either on-chain or off-chain. To ensure the integrity of the operation, on-chain and off-chain events are tied together with a cryptographic check. OpenSea and Rarible follow this model.

► **User authentication.** Users first need to register with the NFTMs to access their services. Post-registration, two different authentication workflows are possible: **(a)** classic credentials-based (username/password), or **(b)** signature-based. With the latter, the user is first asked to sign a challenge string. Then, the marketplace recovers [34] the address of the signer (user) from the elliptic-curve signature. OpenSea, Rarible, Foundation, CryptoPunks, and SuperRare follow this model. Since Ethereum private keys are essentially unguessable [24], this authentication method is generally more secure than traditional passwords (passwords are typically drawn from a limited set of characters, shorter in length, and easier to brute-force).

► **Token minting.** A token is *minted* (created) 6 by calling the appropriate method of the token contract, which generally complies with the ERC-721 or ERC-1155 standard. A single token contract can manage the ownership of a number of NFTs. Every NFT is assigned an integer called `_tokenId`. Therefore, an NFT is uniquely identified by the ⟨`token_contract_address, _tokenId`⟩ pair on the blockchain. A "family" of NFTs, which are either similar, or based on a common theme, called a *collection*, *e.g.*, CryptoPunks. An NFT can be minted in many different ways: **(a)** *default contract*: the token is minted as part of a pre-deployed, designated token contract managed by the marketplace. NFTMs like OpenSea, Foundation, SuperRare, *etc.*, provide a default contract to hold NFTs when no custom contract is deployed by the creator. **(b)** *replica contract*: the NFTM itself deploys a contract on behalf of the creator to manage the collection that the NFT is a part of. Deployed contracts have identical bytecode, but are customized through initialization parameters. Examples of such NFTMs includes Nifty and Rarible. Since both *default* and *replica* contracts are managed by the NFTM, together they are called *internal* token contracts. **(c)** *external contract*: the creator independently deploys a custom contract to manage the collection, and later imports it to the marketplace. To be interoperable with the NFTMs, external contracts must follow a well-established token standard. Otherwise, a custom integration is needed. OpenSea and Rarible allow external contracts on their platforms. A single token contract can manage one or more collection. Typically, *replica* or *external* contracts manage a single collection, while the marketplace *default* contract manages several. In the latter case, the NFTM dApp maintains an off-chain association between the set of `_tokenId`s and the collection those belong to.

► **Token listing.** Once created, a seller lists their assets for sale 3. To list an NFT on a platform, some NFTMs, *e.g.*, Foundation, SuperRare, Nifty, mandate either the seller or the entire collection (that the NFT is a part of) to be verified. Even for the NFTMs where verification is optional, for example, OpenSea, Rarible, getting an artist or a collection verified provides credibility and increases buyers' confidence. NFTMs display special badges on verified profiles of artists and collections, which helps in building a brand, and receive preferential treatment to boost sales – such as search priority and safe-listing to suppress safety-related alerts before the purchase.

► **Token trading.** Buyers can make offers, or place bids 7 on the assets on sale. When an offer is accepted, or an auction is settled, the NFTM transfers 8 assets from the seller's account to the buyer's. Usually, this is when the NFTMs charge a fee for the service they offer. A few key aspects of the NFTM bidding system are discussed below: **(i)** *Pricing protocol*: The bid price can either increase or decrease with every bid. In an English auction, the bid opens at a reserve price, which is the minimum price the seller is willing to accept for an NFT. Subsequent bids from the buyer gradually increase the price. The NFT goes to the highest bidder. The English auction approach is used by most NFTMs, *e.g.*, OpenSea, Foundation, and SuperRare. In a Dutch auction, the bid opens at a high price. Subsequent bids from the seller gradually decrease the price. The NFT goes to the bidder who first accepts a bid. Axie follows the Dutch auction pattern. **(ii)** *Bid storage*: Bids can be stored either on-chain, *e.g.*, CryptoPunks, Foundation, SuperRare, or off-chain, *e.g.*, Nifty, Rarible, OpenSea. There are protocols, such as Wyvern used by OpenSea, which keep both the sell order (listing) and the bids off-chain for gas efficiency, though the order matching and the NFT transfer happen on-chain. Therefore, the marketplace contract cryptographically verifies the buy order against the associated sell order to prevent a malicious buyer from either buying an item that is not on sale, or tampering with an existing sell order. **(iii)** *Active bids*: Some NFTMs disallow multiple active bids on the same asset. For example, in CryptoPunks, Foundation, or SuperRare, when a bidder outbids the current top bidder, the latter gets automatically refunded. **(iv)** *Bid withdrawal*: Some NFTMs, such as CryptoPunks, allow the withdrawal of bids, while others, for example, Foundation, do not. **(v)** *Bid settlement*: Bid settlement does not require seller's intervention in most cases, *i.e.*, the asset automatically goes to the highest bidder. However, for some NFTMs like CryptoPunks, the bid has to be explicitly accepted by the seller.

When an item is sold by a seller other than the creator, it is called a secondary sale. Royalty is the payment made to the creator for every such secondary sale. Before the first (primary) sale takes place, the creator specifies the royalty amount, which is then deducted from every secondary sales and given to the creator. The deduction happens either **(i)** on-chain, where royalty is calculated by the marketplace contract during the *buy* transaction, or **(ii)** off-chain, where the NFTM dApp keeps track of the royalty accumulated from all the sales.

External entities. External to both NFTMs and blockchain, there are services and devices that provide the necessary infrastructure for the system to work. For example, creators store 1 their artwork on web servers or storage services such as Amazon S3 or IPFS. When buyers purchase the NFT, they can exercise their *bragging right* by displaying the art on photobook-style websites or digital NFT photo-frames. The websites, photo-frames 11, and NFTMs 4 fetch tokens from the blockchain 10, and respective artwork from those services.

4 ANALYSIS APPROACH

This paper studies scams, malpractice, and security issues in the NFT ecosystem. In particular, we investigate the following research questions related to the three entities identified in the previous section, *i.e.*, users, marketplaces, and external entities: **(RQ1.)** Are

Marketplace	Trading Volume	Assets	Events	Asset Collection	Event Collection
OpenSea [30]	4.32B	12,215,650	349,911,634	A, B	A
Axie [3]	1.75B	891,238	487,486	B	B
CryptoPunks [8]	1.18B	9,999	172,157	B	B
Rarible [32]	199.42M	72,509	1,864,997	B	B, W
SuperRare [39]	106.87M	28,676	198,848	B	B
Sorare [37]	97.42M	298,219	1,392,292	B	W, B
Foundation [17]	68.19M	112,120	508,349	B	B
Nifty [28]	300.12M	-	-	-	-

Table 1: Characteristics of the marketplace dataset. A: API access, W: Web scraping, B: Blockchain parsing.

there weaknesses in the way NFTMs operate today, and can those be exploited (Section 5)? **(RQ2.)** How and to what extent do external entities pose a threat to the NFT ecosystem (Section 6)? **(RQ3.)** Are users involved in any fraud or malpractice resulting in the financial loss for others (Section 7)?

We used a hybrid (both qualitative and quantitative) approach to answer RQ1, and a quantitative approach for both RQ2 and RQ3. The rest of this section discusses how we collected the data for the quantitative analysis, and we provide a rationale for choosing the specific NFT marketplaces that we examined in more detail.

Marketplace selection. In line with previous work [57], we use DappRadar [9], a popular tracker of dApps, to select the most relevant marketplaces. We selected 8 out of a total of 35 marketplaces (Table 1) listed in DappRadar. This selection was based on the following two criteria: **(a)** backed by the Ethereum blockchain, and **(b)** the "all-time" trading volume is over 50M USD as of June 15, 2021.

Data collection. We collect two different types of data: **(a)** information about the NFTs (assets), *e.g.*, collection name, asset URI, metadata URI, *etc.*, traded on the different marketplaces, **(b)** NFT-related events, such as mint, buy, sell, auction creation, placing of a bid, acceptance of a bid, transfer, *etc.*, generated as a result of marketplace activity. To conduct differential analysis for one of the studies, we needed to monitor how the details of certain assets change over a period of time. Therefore, we crawled the same set of assets three times with a three-month interval between two subsequent crawls: in June 2021, September 2021, and finally in December 2021. Moreover, we collected event information continuously between the June and September crawls. We use CoinGecko [6] API to fetch historical prices of the cryptocoins to convert the pricing information to their equivalent USD value.

Asset and event information: The first step to collect asset and event information is *asset enumeration*, *i.e.*, obtaining the list of assets traded on a marketplace. Once enumerated, we collect the asset and event information for those assets. For both the steps, we employ three different strategies, subject to marketplace restrictions:

1) API access: If a marketplace exposes an appropriate API, we use it to retrieve the list of assets and events. Unfortunately, the APIs are often record-limited, *e.g.*, for a specific query, OpenSea's API returns at most 10, 000 assets. However, the total number of assets listed on their website was 18.2M at the time of crawling. As a workaround, we generate API requests with combinations of `sort` and `filter` parameters to fetch different sets of assets with every request.

2) Web scraping: If a marketplace does not provide an API interface, but its terms and conditions (T&C) do not disallow scraping of their web interface, we crawl the assets and events data from the website.

3) Blockchain parsing: If a marketplace neither provides an API nor allows web scraping, we retrieve asset and event data directly from the blockchain, if possible. Trading activities of a decentralized marketplace are handled by smart contracts that are well-known. Leveraging the ABI (Application Binary Interface) of the contracts published in Etherscan, we parse historical transactions, *e.g.*, `atomicMatch()` in case of OpenSea, to retrieve asset and event details.

Our asset collection is best-effort, as it is impossible to enumerate all the listed assets in a marketplace. This is due to various reasons mentioned above, such as the absence of marketplace APIs, their rate limits, and T&C prohibiting any crawling activity. Table 1 shows the number of assets and events collected for each marketplace, and the strategies used to collect data. Since Nifty does not provide an API, prohibits web scraping through T&C, and stores events off-chain, we were unable to collect data on the marketplace activities.

Measurement study. We utilize the asset and event data we collected to perform several measurement studies, which are described in the subsequent sections. We would like to emphasize that we attain reasonable coverage, *e.g.*, OpenSea, the largest NFTM that accounts for 89.63% of assets in our dataset, listed 18.2M assets in their website at the time of crawling. We crawled 12.2M assets, which is 66.94% of the size of the marketplace. Since OpenSea contributes to the most number of assets in our dataset, we use only OpenSea in Section 5 (unless the study requires cross-NFTM analysis) and Section 6, as only that dataset would be representative enough to capture the extent of the issues we quantified in those sections. However, since we measured the occurrences of trading malpractices per NFTM in Section 7, we used assets from all the marketplaces.

5 ISSUES IN NFT MARKETPLACES

In this section, we identify weaknesses in the design of NFTMs, which, when abused, pose a significant risk in the form of financial loss to both the marketplaces and its users. For this part of the study, we gathered information from public security incidents, attacks, and abuses reported on various blogs and technical reports, direct interactions with individual marketplaces, and marketplace documentation. We have systematized our findings by connecting those issues with the marketplace activities discussed in Section 3, and then quantified, whenever possible, the prevalence/impact of those issues. Lastly, we systematically evaluated the existence of each of the issues across all the marketplaces (Table 2).

5.1 User Authentication

(U1) Identity verification. Art in the physical world has been used in money laundering schemes [2]. NFTs might make this process easier, as trades are executed by anonymous users, and there are no physical artworks to be transported. Identity verification is the first step to deter such criminals. Major crypto exchanges, such as Coinbase and Binance US, are highly regulated. To create

Issues	Marketplaces							
	OpenSea	Axie	CryptoPunks	Rarible	SuperRare	Sorare	Foundation	Nifty
User authentication								
U1. Identity verification	✗	✗	✗	✗	✗	✗	✗	✗
U2. Two-factor authentication	N	✗	N	N	N	O	N	✓
Token minting								
M1. Verifiability of token contracts	✗	N	N	✗	N	N	N	✗
M2. Tampering token metadata								
M2.1 Changing metadata url	P	✓	✗	✗	P	P	✗	✗
M2.2 Decentralized metadata	O	✗	✗	O	O	✗	M	O
Token listing								
L1. Principle of least privilege	✓	✓	✓	✓	P	✓	✗	✗
L2. Invalid caching	✓	N	N	✓	N	N	N	N
L3. Seller / collection verification	O	N	N	O	M	N	M	M
Token trading								
T1. Lack of transparency	✗	✗	✗	✗	✗	✗	✗	✓
T2. Fairness in bidding	✗	✓	✓	✗	✓	✗	✓	✗
T3. Royalty and fee evasion								
T3.1 Cross-platform	✗	✗	N	✗	✗	✗	P	✗
T3.2 Post-sales modification	✓	✗	N	✓	✗	✗	✗	✗

Table 2: Issues in the NFT marketplaces. O: Optional, M : Mandatory, P: Partial, N: Not applicable, ✓: Exists, ✗: Does not exist.

an account with these exchanges, one needs to provide personally identifiable information (PII), *e.g.*, name, residential address, social security number (SSN), along with supporting documents confirming these details. Without getting the identity verified, it is either impossible to use the platform, or it can only be used with tight financial restrictions in place. To investigate if the NFTMs impose similar regulatory restrictions, we interacted with them by creating accounts. We discovered that no NFTM has made any steps towards enforcing KYC (Know Your Customer) rules nor implemented AML/CFT (Anti-Money Laundering/Combating the Financing of Terrorism) measures. As a result, apart from being able to hide the identity, a user can create several accounts on the platform that are hard to be traced back to one single entity.

(U2) Two-factor authentication. Enabling 2FA (Two-Factor Authentication) greatly enhances the security of a password-based authentication workflow. While traditional financial institutions like banks, brokerages, and cryptocurrency exchanges, such as Coinbase and Binance, provide 2FA as an option, it is not yet a ubiquitous option for NFTMs. Sorare manages a user's wallet on her behalf. As a result, an attacker who is able to login into an account can download the user's Ethereum private key associated with the wallet, and transact on behalf of her. Though Sorare does support 2FA, it is not enabled by default. 2FA was also optional for Nifty users until the infamous hack [19] that compromised a number of accounts in March 2021. According to their initial assessment, none of the impacted accounts used 2FA when the hack took place.

5.2 Token Minting

(M1) Verifiability of token contracts. A token contract is considered "verifiable" if its source code is submitted to Etherscan. Given the functional complexity of these token contracts, source code is much easier to audit than bytecode. Verifiability of external token contracts is crucial as they can be malicious or buggy. As an example, OpenSea users complained about a malicious token contract that did not transfer tokens after purchase. Also, to make a particular NFT valuable, sometimes NFT projects promise to circulate only a certain number (rarity) of that token. A malicious token contract can be abused to mint more tokens than the *rarity* threshold, thus dropping the token's price, which hurts the buyers. A malfunctioning contract can burn gas without even doing any real work, *e.g.*, almost all `Purchase` events of the `CelebrityBreeder` contract failed with errors. Ideally, an NFT project should make the source of the underlying token contract available for public scrutiny before the NFTs are minted to make sure that they are neither malicious nor buggy. Unfortunately, none of the NFTMs that support external token contracts mandates such contracts to be open-source.

► **Quantitative analysis.** To enumerate how abundant closed-source NFT tokens are, we queried Etherscan API for every token contract in our dataset to check if its source is present. Out of 11,339 token contracts, 8,122 (71.63%) were open-source, while the remaining 3,217 (28.37%) were closed-source, of which 7,850 (96.65%) and 3,209 (99.75%) tokens belong to OpenSea, respectively.

Further, we intended to evaluate if closed-source tokens are more likely to exhibit malicious behavior than open-source ones. Since NFTMs take down NFTs when they observe or receive a report of either an abuse or a violation of the T&C, we consider "take-down" as an indirect (yet strong) indication of a token being found malicious. According to our observation, 1,765 (55.00%) closed-source tokens were taken down by OpenSea between June and December, which account for $328.8M USD in trading volume. On the contrary, only 606 (7.72%) open-source tokens were taken down during the same span.

(M2) Tampering with token metadata. The metadata of a token holds the pointer to the corresponding asset. Hence, if the metadata changes, the token loses its significance. The ERC-721 standard for NFTs actually allows for the possibility to change a token's metadata. However, when an NFT represents a particular asset (such as a piece of art) that is sold, changing the metadata violates the expectation of the buyer. The location and the content of the metadata are decided at the time of minting. A malicious creator/owner $\mathcal{A}$ can alter the metadata by manipulating either of the two post-minting: **(i)** by changing the `metadata_url`, and **(ii)** by modifying the metadata itself. Even if **(i)** can be disallowed at the contract level, metadata hosted on third-party (web) domains can be freely modified by $\mathcal{A}$, if she controls the domain. This second attack can be prevented if the metadata is hosted in IPFS. Since the URL of an object stored in IPFS includes the hash of its content, the metadata cannot be modified while retaining the same URL recorded in the NFT.

For internal token contracts, CryptoPunks, Foundation, Rarible, and Nifty offer no way to update the `metadata_url` of an NFT. Axie allows the creator to modify the URL at any time. OpenSea, SuperRare, and Sorare allow modification by the creator until the first sale. Since only Foundation mandates storing the metadata on IPFS, other NFTMs are susceptible to the second attack for the internal contracts. Since no NFTM supporting external token contracts employs any check to prevent metadata tampering, both attacks are feasible.

► **Quantitative analysis.** We performed a differential analysis to determine the change in the `metadata_urls` of external assets over a period of time. Specifically, we monitored the `metadata_urls` of all 9,064,767 external OpenSea assets three times over a span of six months in an uniform interval—in June 2021, September 2021, and December 2021, respectively. Since ERC-721 metadata extensions are optional (explained in Section 6), `metadata_urls` were completely missing for some of the assets. Also, OpenSea took down some assets during this time period, which is why their `metadata_urls` could not be retrieved in the subsequent crawl. After excluding these two kinds of assets, we were left with 3,079,139 assets that had `metadata_urls` in all three crawls. According to our observation, the `metadata_urls` of 89,089 (2.89%) and 35,446 (1.15%) assets changed between the first two and the last two crawls, respectively.

5.3 Token Listing

(L1) Principle of least privilege. While listing an NFT, the NFTM takes control of the token so that when a sale is executed, it can transfer the ownership of the NFT from the seller to the buyer. To this end, the NFTM needs to be either **(i)** *the owner* of the NFT: that is, the current owner transfers the asset to an escrow account $\mathcal{E}$ during listing, or **(ii)** *a controller*: an Ethereum account C that can manage that specific NFT on behalf of the owner, or **(iii)** *an operator*: an Ethereum account O that can manage all the NFTs in that collection. The escrow model in case **(i)** is risky because one single escrow contract/wallet $\mathcal{E}$ managed by the NFTM holds all assets being traded on the platform. Therefore, the security of all assets in a marketplace depends on the security of the escrow contract or the external account that manages such contract. This design essentially violates the principle of least privilege. As a result, either a vulnerability in the contract or a leak of the private key of the external account could compromise the security of all the stored NFTs. Nifty, Foundation, SuperRare follow this approach. A safer alternative would be to adopt **(ii)** or **(iii)**, where a proxy contract C or O deployed by the NFTM becomes the controller of the NFT, or the operator of the entire NFT collection, respectively. As enforced by the marketplace contract, the NFTM is able to transfer an NFT only when it has been put on sale and the required amount is first paid to the seller. This ensures the safety of the NFT token even in case of a marketplace hack. If the private key of a seller (owner of an NFT) gets leaked, it can, at most, compromise the safety of that specific NFT or collection, as opposed to all the NFTs as in the case of the escrow model.

► **Quantitative analysis.** Among the NFTMs in our dataset, Foundation holds tokens in an escrow contract, while Nifty uses an Externally Owned Account (EOA) as escrow wallet. SuperRare escrows tokens only when an auction is ongoing. The larger the number of NFTs held in escrow, the greater is the risk. On December 31, 2021, SuperRare, Foundation and Nifty held 55, 64079, and 90988 NFTs in their escrow accounts, respectively.

			Count	**Total Sales**	**Average Sales**	**Taken Down**
OpenSea	Seller	Verified	502	$114.5M	$228,028	-
		Non-verified	124,398	$2.7B	$22,001	-
	Collection	Verified	1,805	$3.3B	$1,824,882	88
		Non-verified	234,112	$403.4M	$1,723	11182

Table 3: Number of verified and non-verified sellers and collections, along with corresponding sales volumes.

(L2) Invalid caching. While displaying an NFT on sale, OpenSea and Rarible leverage a local caching layer to avoid repeated requests to fetch the associated images. If the image is updated, or disappears, the cache goes out of sync. This could trick a buyer into purchasing an NFT for which the asset is either non-existent or different from what the NFTM displays using its stale cache.

► **Quantitative analysis.** To understand the potential impact of this caching issue, we measured how many `image_urls` in our OpenSea dataset are inaccessible (non-200 HTTP response code), but OpenSea still serves the corresponding cached versions. Out of total 12,215,650 NFTs, `image_urls` of 3,945,231 (32.30%) tokens were inaccessible. However, OpenSea still cached 2,691,030 (68.21%) of those inaccessible images, thus creating the illusion that the asset linked to the NFT is still alive. One such broken collection is Gods Unchained, a verified collection with an overall trading volume of 19.8K Ethers.

(L3) Seller and collection verification. Listings by verified sellers/collections are not only given preferential treatment by the NFTMs, but they also attract greater attention from the buyer community. However, the verification mechanism is typically ad-hoc, and the final decision is at the discretion of the NFTMs. Common requirements include sharing the social media handles of the sellers and proving their ownership, sharing contact information, collections needing to reach certain trading volume, submitting the draft files of the digital artworks, *etc.* Marketplaces such as Foundation adopt a stricter policy by mandating verification of all the sellers on their platform. However, there are NFTMs, *e.g.*, OpenSea, Rarible, where verification is optional. Buyers are expected to exercise self-judgment when trading on these platforms, which, unfortunately, puts them at greater risk.

Since verification comes with financial benefits, it has been abused in different ways: **(i) Forging verification badge.** Scammers forged profile pictures with an image of the verification badge overlaid on them, making the profiles appear visually indistinguishable from the verified ones at a cursory glance. **(ii) Impersonation.** Abusing weak verification procedures, scammers got their fake profiles verified by just submitting social media handles, without actually proving the ownership of the corresponding accounts [27]. **(iii) Wash trading.** One of the requirements of OpenSea to verify a collection is to have at least 100 ETH in trading volume [20], which is possibly hard to attain for a newly launched collection. Historically, this requirement has incentivized people to perform *wash trading*, *i.e.*, performing fictitious trades between multiple accounts that are all under the control of the attacker, to artificially inflate sales volumes.

► **Quantitative analysis.** To highlight the economic incentive behind verification abuse, we present the number of sales and the sales volume generated by the verified and non-verified sellers and collections in OpenSea in Table 3. Though only 0.40% sellers and

0.77% collections of OpenSea are verified, the average sales per verified seller and collection are 10 and 1,059 times more that their non-verified counterparts, respectively.

Next, we measure how effective the NFTM verification mechanisms are in preventing abuse. Had the verification mechanism been foolproof, then a verified collection could not be malicious, and in turn, it should never have been taken down. However, we observed that 4.88% of the verified and 4.78% of the non-verified OpenSea collections were taken down in six months (between June and December 2021). This indicates that though verification attempts to reduce abuse, it fails to eliminate it completely. The fact that the verified collections are still taken down shows that bad actors do "slip through" the system and verify their collections.

5.4 Token Trading

(T1) Lack of transparency. NFTs are asset-ownership records that should be stored on the blockchain to allow for public verifiability. In a decentralized setting, an NFT sale is handled by a marketplace contract C_m that invokes the `transfer()` API of the token contract C_t to transfer the token from the seller to the buyer. Every *sale* transaction and the associated *transfer*, for example, the `atomicMatch()` call in case of OpenSea, is visible on the blockchain. Among other things, each transaction includes the following information: **(i)** address of the seller (current owner), **(ii)** address of the buyer (new owner), **(iii)** how much the NFT was sold for, **(iv)** time of ownership transfer. Querying for ownership has further been made easier by ERC-721 `ownerOf()` API that returns the current owner of a token. The sales records, in conjunction with the API, permit one to reconstruct the precise sales and ownership history of an NFT.

On the other hand, if sales records and transactions are stored off-chain, it becomes impossible to verify any trades and the ownership history of an NFT. Moreover, a malicious NFTM can abuse this fact to forge spurious sales records to inflate the trading activity and volume. Off-chain records are susceptible to tampering, censorship, and prone to disappear if the NFTM database goes down. Among the NFTMs we surveyed, only Nifty maintains off-chain records. When an item is listed, Nifty takes control of the NFT by first having it transferred (T_1) to an escrow wallet. Thereafter, multiple trades can take place while Nifty holds the custody of the asset, but no sales record is ever emitted on the blockchain. If and when the owner decides to take the NFT out of Nifty, the marketplace transfers (T_2) the token back to the owner's account. Since only T_1 and T_2 are visible from the blockchain, no intermediate ownership and sales activity can be verified.

(T2) Fairness in bidding. NFTMs implement bidding either **(i)** on-chain, through a smart contract that requires the bid amounts to be deposited while placing the bid, or **(ii)** off-chain, through the NFTM dApp which maintains an orderbook without requiring any upfront payment. Off-chain bidding is *unfair* as it can be abused by both the NFTM and the users. Since bids are not visible from the blockchain, NFTMs can inflate the bid volume to create hype. Also, placing bids is inexpensive, as there is no money transfer involved. Therefore, such NFTMs are more susceptible to *bid pollution*, a form of abuse where a large number of *casual* bids are placed on items. Since no money is locked, most of these bids are likely to fail due to a shortage of funds in the bidder's account at the time of execution. Since on-chain bidding costs gas to place/cancel bids, it deters scammers from placing spurious bids, making abuses less frequent. Moreover, on-chain bids reserve the bid amount upfront. Therefore, such bids invariably succeed during settlement. In OpenSea, we observed sellers complain that (attempted) sales of their items fail because the WETH balances of the winning bidders drop below the offered amounts.

► **Quantitative analysis.** Unless a bid fails due to lack of funds, an NFT gets transferred to the highest bidder at the end of the auction. To measure the extent of bid pollution in the NFTMs, we enumerated the auctions where the highest bidder did not receive the item. This is unfair to the seller, because the bid immediately below might be a lowball offer. Our analysis uncovered 16,215 and 15,368 such instances out of 48,862 and 19,109 total auctions in OpenSea and Rarible, respectively. We did not find any evidence of the same in the Foundation, Axie, and SuperRare marketplaces.

(T3) Royalty distribution and marketplace fee evasion. If a royalty is set, every trade should earn a fee for the creator. However, we identified ways in which users can potentially abuse the royalty implementations: **(i) Cross-platform.** As explained in Section 3, royalty is enforced by either the marketplace contract or the dApp, both of which are specific to an NFTM. Also, NFTMs do not share royalty information with each other. Therefore, royalty set on one platform is not visible from the other. Leveraging this lack of coordination, a malicious seller can evade royalty by trading the NFT through a platform where royalty is not set, though it is set on another. **(ii) Non-enforcement.** Neither royalty nor marketplace fees are enforced in ERC-721 token contracts. A malicious seller can thus avoid both payments by transferring (ERC-721 `transfer()`) the NFT to the buyer directly and settling the payment off-platform. Both royalty and fees could be levied inside the `transfer` method of the token contract, though the additional logic makes the API more expensive. **(iii) Post-sales modification.** OpenSea and Rarible allow the creator to modify the royalty amount even after the primary sale. Now, the royalty is calculated on the price listed by the seller. In a potential abuse scenario, a creator can first lure a buyer B by setting a low royalty and then increasing it post-sales. During secondary sales, B may not notice this change at all, and may end up giving more royalty to the creator than initially advertised.

► **Quantitative analysis.** We discovered potential abuses of unconditional token transfer **(case ii)** to evade NFTM fees and royalty. The question of evasion appears when a seller S lists an NFT on a marketplace to gain popularity, but executes the trade off-platform, entirely bypassing the marketplace protocol. There could be two possible cases. Seller S might trust the buyer B and, therefore, transfers the NFT first. After that, B settles the payment. In the other case, the order is reversed. For the assets listed in each NFTM, we counted the number of occurrences on the blockchain where an address (seller) S transferred the NFT to another address (buyer) B, and B sent a payment to S on-chain within 15 minutes (before or after) the transfer transaction. We found 56920, 302, 2777, 5, 814, 56, and 0 such instances for assets listed in OpenSea, SuperRare, Rarible, Foundation, CryptoPunks, Sorare, and Axie, respectively. Note that this estimate is conservative, because the payment could be made either off-chain or outside the time window that we considered for our analysis.

We also measured how often creators abuse sellers by increasing the royalty after the primary sale **(case iii)**. For each OpenSea asset, we enumerated the "sell" events in increasing order of time, and counted the number of times the royalty was increased with respect to the previous sale. We discovered 157,450 instances of such royalty modifications across 20,802 (8.81%) collections.

6 ISSUES RELATED TO EXTERNAL ENTITIES

The asset (picture, video) that an NFT points to must be accessible for this NFT to be "meaningful." NFTs can point to assets in two ways. If the NFT contract is ERC-721-compliant and implements the metadata extension, then the token includes a `metadata_url` on-chain, which points to a metadata record (JSON). This record, in turn, includes an `image_url` field that points to the actual digital asset. Many older tokens, on the other hand, are not standard-compliant and do not contain any on-chain `image_url`. Instead, they use some ad-hoc, off-chain scheme to link to an asset. For such NFTs, NFTMs implement custom support so that they can generate valid image URLs. Since both the metadata record and the asset are stored off-chain, those do not enjoy the same guarantee of immutability as the NFT itself. When any URL becomes inaccessible, that breaks the link between the NFT and the corresponding asset. In practice, the URLs frequently point to a distributed storage service, *e.g.*, IPFS, or centralized storage, *e.g.*, a web-domain or Amazon S3 bucket. For IPFS URLs, if the NFT owner is aware, she can keep the NFT "alive" by pinning the resource (*i.e.*, storing it persistently). Even that could also be problematic, because NFTs do not store the hash value of the actual resource but rather store URLs that point to an IPFS gateway web service. If the gateway becomes unavailable, the NFT "breaks." In general, NFTs that include URLs that point to domains outside the control of the NFT owners risk getting invalidated when the corresponding domains go away.

► **Quantitative analysis.** We performed an analysis to quantify the number of OpenSea NFTs that were "lost" due to the reasons outlined above. As of June 15, 2021, out of our 12,215,650 assets from OpenSea, there were only 3,175,644 assets with a valid `metadata_url` field. Querying OpenSea's API, we obtained 8,363,550 assets with non-empty `image_url` fields. The remaining 3,860,607 assets did not have an `image_url` field, which means that they are hosted directly on OpenSea (content creators have the option to leave the image URL field empty, in which case OpenSea handles the hosting). We first check whether the image and metadata URLs point to resources hosted on IPFS. Next, we check whether the URLs are still accessible. To this end, we perform an `HTTP HEAD` query. If the query returns with a response code other than 200 (`OK`), we perform an `HTTP GET` query next. If that also returns a non-200 response code, we mark that URL as *inaccessible*. We take this two-step approach to optimize for performance, and not to generate false negatives due to web servers that do not respect `HEAD` queries. Also, the servers hosting the assets could be offline at the time of testing, but later come back up online. To account for this possibility, we repeated the above URL-check three times in a span of 15 days. Only the assets marked as *inaccessible* in the previous attempt were tested for accessibility each time. An asset is finally marked as *inaccessible* only if all three attempts agree.

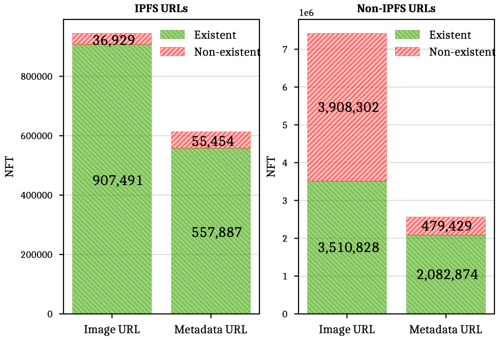

Figure 3: Validity of image and metadata URLs

Figure 3 reports our findings. Two important observations are: **(i)** Only 3.91% of the assets (images) and 9.04% of metadata records hosted on IPFS have disappeared in our dataset between June and December; as expected, NFTs hosted on IPFS are less likely to disappear than those hosted on non-IPFS domains. **(ii)** Though IPFS is supposed to be more resilient to disappearance of the assets, a majority of asset URLs (88.71%) as well as metadata URLs (80.69%) are hosted on non-IPFS domains. Looking at all lost NFTs, they have generated a staggering amount of $160,761,805 USD in revenue from 118,294 transactions. Not only that, due to the caching issue we discussed in Section 5, it is very well possible that a fraction of them are still in circulation. As this analysis shows, persistence is a pressing issue in the NFT space.

7 FRAUDULENT USER BEHAVIORS

In this section, we study the impact of various fraudulent user activities that occur in NFTMs. In particular, we look at counterfeit NFT creation as well as trading malpractices, such as wash trading, shill bidding, and bid shielding.

7.1 Counterfeit NFT Creation

The authenticity of an NFT is endorsed by the smart contract managing the collection. Therefore, to ensure that the token one is buying is legitimate, buyers are advised to verify the contract address of the collection from official sources, *e.g.*, the project's web page, before making a purchase. Unfortunately, buyers are not always aware of the existence of counterfeits, or of how they can verify an NFT's authenticity. Instead, they only rely on the names and visual appearances of items in the marketplaces. This makes it possible for malicious users to offer "fake" NFTs. We observed the following types of counterfeits:

(i) Similar collection names. There are fake NFTs that use the name of a collection or individual piece that resembles the original (victim) one. A common trick is to substitute ASCII characters in the original name with non-ASCII characters that look alike. To prevent such abuse, OpenSea restricts users from using popular collection names and certain special characters. Still, it is often possible to circumvent these limitations, *e.g.*, by adding a dot(.) at the end of the name or substituting an upper-case character with a lower-case one, *e.g.*, a fake of "CryptoSpells" collection used the name "Cryptospells." Moreover, restrictions can cause problems for legitimate users, *e.g.*, French users complained about not being able to use the accented characters in collections.

(ii) Identical image URLs. Some fake NFTs point to existing assets, *i.e.*, they simply copy the `image_urls` of legitimate NFTs. For example, CryptoPunks is a well-known collection. Of course, nothing prevents a scammer from deploying her own token contract on the blockchain and mint tokens that point to CryptoPunks. A buyer who just looks at the appearance of items in a collection will see the CryptoPunks images and might mistake the NFTs for the originals.

(iii) Similar images. Instead of copying the `image_url`, a scammer might copy the digital asset and then mint an NFT that points to this copy. As of now, no NFTM runs any similarity check to detect if a media file has already been used by other NFTs.

► **Quantitative analysis.** We looked for each type of counterfeits present in the OpenSea dataset comprising of 12,215,650 NFTs spread across 236,057 collections.

(i) To check for (potential) counterfeits that abuse similar collection names, we compute the *Levenshtein distance*, an *edit distance* metric between pairs of collection names (strings). Since a shorter distance indicates greater similarity, we considered a maximum distance of 2 characters, which means that we only consider collection names as similar if they differ in at most two characters. We considered 52,399 collections that have names longer than 7 characters, and a minimum of 10 NFTs in it (collections with fewer NFTs could be insignificant) to avoid spurious matches. Given that it is more beneficial to imitate verified collections, we only considered collection pairs that include one verified collection (and the other one is considered to be its *replica*).

Our analysis found 322 collection pairs with similar names. We noticed that the names of most of the replica collections were minor modifications of the names of the respective verified collections, for example, pluralizing a noun, adding whitespace at hard-to-notice positions, *etc.*, which indicates a potential intent to mislead. We then randomly picked 100 pairs and checked if those replica collections indeed contain images that are similar to the verified ones and that could mislead buyers. Since judging the similarity visually could be subjective, two researchers independently performed the assessment, and a pair was marked "visually similar" only if both the decisions agreed. We discovered 11 such collections, which we reported to OpenSea requesting a take-down. Moreover, we identified an additional 11 collections that were already taken down by OpenSea (which indicates wrongdoing) between June and December 2021.

(ii) To check for counterfeits that leverage identical `image_urls`, we first collected 8,363,550 `image_urls` comprising of 944,420 IPFS, and 7,419,130 non-IPFS URLs from our dataset. Objects on IPFS are accessed through IPFS gateways, which are web services. An IPFS URL is typically of the form: `http(s)://<gateway>/<ipfs_hash>`. Any `gateway` can be used to access the object pointed to by `<ipfs_hash>`. Therefore, we pre-processed those URLs to extract only the hash component. In the last step, we performed a string comparison between every pair of IPFS hashes and non-IPFS URLs, which reported 356,377 and 2,082,119 identical IPFS, and non-IPFS URLs with at least one duplicate, respectively.

(iii) To find potential counterfeits due to image similarity, we crawled the images pointed by the `image_url` for all NFTs in our dataset. Since the individual assets linked to NFTs can be very large, we decided to focus on downloading just the smaller resolution version of an asset generated and cached by OpenSea. We then used the *perceptual* algorithm [31] of ImageHash [22], a popular (2.1K GitHub stars) image hashing tool, to compute a "fuzzy" hash that is tolerant to small perturbations of the images. Lastly, we compare every pair of hashes to find similar images. We refrain from comparing hashes of the images that are part of the same collection, as they are likely similar (but not counterfeits). We downloaded 9,991,013 images, and we discovered 59,425 hash collision pairs. We randomly picked 100 such pairs, and manually verified that 90% of those image pairs are indeed visually identical.

7.2 Trading Malpractices

In this section, we explore illicit trading practices, specifically, wash trading, shill bidding, and bid shielding [4, 38, 42]. We first discuss how these malpractices are relevant in the context of NFTMs, and then build heuristic models to detect such attacks. Finally, we apply these models to all 13,628,411 assets and 354,535,763 events we collected (Section 4). The goal is to measure the extent and impact of these trading activities on the top 7 NFTMs.

Data modeling. From the event data and the Ether flows collected from blockchain transactions, we extract actions (such as transfers, sales, bids, ...) that operate on NFTs. Figure 7 shows the types of predicates (actions) that we record for users u, assets a, auctions id and prices p. These predicates capture relationships that we use to build four different graphs: A sales graph $\mathcal{G}_s$ (sale), a bidding graph $\mathcal{G}_b$ (auction, bid, cancel_bid, win), a payment graph $\mathcal{G}_p$ (paid), and an asset transfer graph $\mathcal{G}_t$ (transfer). $\mathcal{G}_b$ contains two types of nodes: users (u) and assets (a), and directed edges from u to a annotated with property tuples of the form (p, t, id). All of $\mathcal{G}_s$, $\mathcal{G}_p$, and $\mathcal{G}_t$ contain only one type of node: users (u), and directed edges from u_1 to u_2. Edges in $\mathcal{G}_s$, $\mathcal{G}_p$, and $\mathcal{G}_t$ are annotated with property tuples of the form (a, p, t), (e), and (a), respectively.

7.2.1 *Wash Trading.*

In wash trading, the buyer and the seller collude to artificially inflate the trading volume of an asset by engaging in spurious trading activities. In NFTMs, users wash trade to either create the illusion of demand for a specific asset, artist, *etc.*, or to inflate metrics that are of their financial interest, such as getting a profile/asset verified, or collecting rewards. For example, Rarible users are incentivized by $RARI governance tokens where the more a user spends, the more tokens they receive [25]. It is suspected that many high-value NFT sales related to popular projects such as CryptoKitties [7] and Decentraland [10] are instances of wash trading [42].

Detection. In the NFT space, wash traders primarily intend to increase the sales volume of NFT collections. To detect wash trading, given a set of assets $\mathcal{A} = \{a_1, a_2, ..., a_n\}$ that are part of a collection, we check for a set of users (addresses) $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ who heavily trade those assets with each other. We assume a limited number of colluding users to make the problem tractable (we use an empirical threshold of 50 users). In other to generate wash trades, these users repeatedly trade that set of assets among them, which often results in cycles in the sales graph $\mathcal{G}_s$. Hence, we check $\mathcal{G}_s$

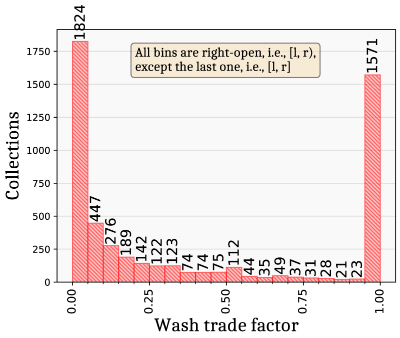

Figure 4: Distribution of wash trading factors across collections

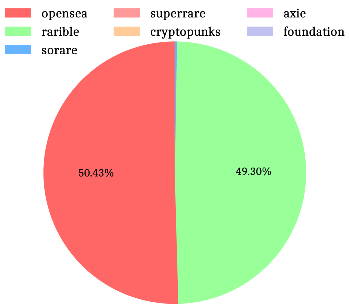

Figure 5: Relative volumes of wash trading in different marketplaces

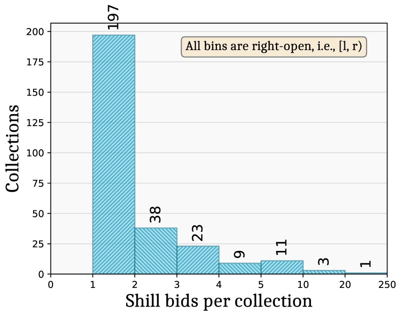

Figure 6: Distribution of shill bidding across collections

$\mathrm{sale}(u_1, a, p, t, u_2)$:– u_1 sold a to u_2 at price p at time t

$\mathrm{auction}(u, p, t, id, a)$:– u started auction with id id at time t with starting price p

$\mathrm{bid}(u, p, t, id, a)$:– u placed bid p on a at time t on an auction with id id

$\mathrm{cancel\_bid}(u, p, t, id, a)$:– u canceled bid p on a at t on an auction with id id

$\mathrm{win}(u, p, t, id, a)$:– u won auction id on a at time t with price p

$\mathrm{paid}(u_1, e, u_2)$:– u_1 transfered e ethers to u_2

$\mathrm{transfer}(u_1, a, u_2)$:– u_1 transfered a to u_2

Figure 7: Relationships in graphs $\mathcal{G}_s, \mathcal{G}_b, \mathcal{G}_p, \mathcal{G}_t$.

for the existence of cyclic relationships among these users. In a strongly connected component (SCC) of a graph, there exist paths between all pairs of vertices. Therefore, this type of wash trade can be detected [65] by checking if two users: u_1 and u_2 appear in any SCC of the sales graph $\mathcal{G}_s$. In other words, if $\mathrm{SCC}(u_1, u_2, \mathcal{G}_s)$ holds, it means that both the users are involved in round-trip trades, *i.e.*, there exist either direct, or indirect sale relations between them in both the directions. Now, two users being a part of an SCC can be accidental, and does not indicate the frequency of trades between them. However, in a wash trade, users are involved in frequent sales. Therefore, we only consider SCCs where the number of sale relationships between every two intermediate users is above (indicating 'heavy' trading volume) an empirically determined threshold (ϵ). We use $\epsilon = 10$ in our analysis.

However, bad actors can come up with more intricate strategies to conceal apparent connections so that such simple detection can be evaded. We manually analyzed the blockchain transactions history and found two evasion strategies that would throw off the prior analysis. In the first case, when we investigated an otherwise legitimate-looking sale relation $u_i \rightarrow u_j \rightarrow u_k$, we realized that both u_j and u_k are funded (Ether transfer) by the same "parent" user u_i. We capture this case by checking if two users: u_1 and u_2 appear in any weakly connected component (WCC) of the payment graph $\mathcal{G}_p$. In other words, if $\mathrm{WCC}(u_1, u_2, \mathcal{G}_p)$ holds, it means that direct or indirect Ether-flow exists between those two users in either direction. In the second case, for a sale relation $u_i \rightarrow u_j \rightarrow u_k$, we identified multiple unconditional asset transfers (ERC-721 `transfer()`) from u_i to u_k, giving a strong indication of a close tie between those users. We capture this case by checking if two users: u_1 and u_2 appear in any WCC of the transfer graph $\mathcal{G}_t$. In other words, if $\mathrm{WCC}(u_1, u_2, \mathcal{G}_t)$ holds, it means that direct or indirect unconditional asset transfer relationships exist between those two users in either direction.

To summarize, our model considers any $\mathrm{sale}(u_1, \_, \_, \_, u_2)$ relation a potential wash trade if: $\mathrm{SCC}(u_1, u_2, \mathcal{G}_s) \vee \mathrm{WCC}(u_1, u_2, \mathcal{G}_t) \vee \mathrm{WCC}(u_1, u_2, \mathcal{G}_p)$.

► **Quantitative analysis.** We detected 9,393 instances of wash trading that generated $96,858,093 USD in trading volume across 5,297 collections involving 17,821 users in all NFTMs except Axie, Foundation, and CryptoPunks. Moreover, out of 238,180 collections in our dataset, only 8,869 collections had more than $2K in trading volume, out of which 2,569 (28.97%) collections show signs of wash trading.

We define *wash_trade_factor* (WTF) as the fraction of the total trading volume of a collection generated by wash trading, *i.e.*, if WTF is 1, then all the trades are wash trades. In Figure 4, we show the distribution of the *wash_trade_factor* across collections where wash trading has been detected. Of all the wash traded collections, 1,824 (34.43%) collections had less than 5% (WTF < 0.05) of the trades generated by wash trades. Interestingly, we discovered 1,571 (29.66%) collections which were heavily abused, because more than 95% of all of their trades are wash trades, totaling $3,407,284 USD in the trading volume. Figure 5 shows the relative volumes of wash trades that have happened in different NFTMs. Though nearly equal volume of wash trades were discovered in both Rarible (49.30%) and OpenSea (50.43%), given that the overall trading volume of OpenSea is 21 times more (Table 1) than that of Rarible, it seems that wash trading is significantly more frequent in Rarible than OpenSea. Our finding is also corroborated by discussions we saw on Rarible Discord, which indicates a heavy amount of past wash trading incidents as malicious users attempted to secure $RARI tokens.

► **Manual analysis.** In our analysis, the size of a connected component represents the number of addresses involved in a wash trade. We observed that 98.88% (9,288 of 9,393) of the reports had a component size at most 10. Therefore, for our manual analysis, we randomly selected 100 reports, and checked whether one of the following two conditions holds: **(i)** if a addresses are involved in t transactions on n NFTs, then both $t \geq 2a$ and $n \ll t$ need to hold. The intuition is that if a set of users "heavily" trades on only a small number of assets, then those are likely to be wash trades. Alternatively, **(ii)** the addresses involved in trading are all funded by a common, on-chain funding source. This is true when the "supposed" buyers, in reality, are all funded directly by the seller, or by a seller-controlled address, before making (pseudo) purchases. If one of these two conditions holds, we consider a detected wash trade instance as a true positive. We determined all the sampled instances as true positives.

Limitation. Ethereum mixers (informally "tumblers"), such as Bitmix [5], ETH Mixer [13], and Tornado Cash [40], are anonymity services that help to conceal the true source of a payment by breaking the link between the receivers and the sender of the funds. Specifically, these services accept Ethers from a user, and either route it to a smart contract, or relay it through a complex, large network of addresses by splitting the amount into a number of micro-transactions; essentially mingling that fund with hundreds of other users. Since our wash trade detection strategy leverages information about Ether flows between two addresses, mixers can lead to false negatives.

7.2.2 Shill Bidding. Shill bidding is a common auction fraud where a seller artificially inflates the final price of an asset either by placing bids on her own asset, or colluding with other bidders for placing spurious bids with increasingly higher bid amounts. This can lead to honest bidders paying higher prices than they would have otherwise. With high-value bids on assets becoming increasingly common, it is suspected that many sales suffer from artificial price inflation [38].

Detection. Detecting shill bidding is difficult when looking at a single auction in isolation. It becomes even harder when malicious users take turns, placing bids on each others' auctions so that the seller-bidder relation changes. In this paper, we only consider the simple case where a specific user repeatedly places bids in auctions, yet never (or rarely) purchases anything. Moreover, we check whether there is some relationship between this user and the seller. Thus, our findings should be viewed as a lower bound on the actual number of shill bidding occurrences in NFTMs. Our detection mechanism draws on our insight from the manual analysis of NFTM activities and prior work [63].

Let $\mathsf{bid}(u_b, p_i, t_i, id, a)$ denote the i-th bid placed by user u_b with amount p_i at time t_i on asset a in an auction with id id created by the seller u_s. Then, user u_b is a shill bidder if:

Rule 1. u_b places at least n bids on an asset a auctioned by u_s with monotonically increasing bid amounts. That is, $\forall i \in [1, n]$, $\mathsf{bid}(u_b, p_i, t_i, id, a)$, the following holds: $\forall i, \forall j, \quad t_i > t_j \implies p_i > p_j$

Rule 2. u_b never buys the asset a, *i.e.*, $\mathsf{win}(u_b, \_, \_, id, a)$ is false .

Rule 3. u_b has limited buying/selling activity, *i.e.*, $|\{\mathsf{sale}(u_b, \_, \_, \_, \_)\} \cup \{\mathsf{sale}(\_, \_, \_, \_, u_b)\}| < \sigma$, where σ is an empirically determined threshold. We set $\sigma = 10$ for our analysis.

Rule 4. u_b is "connected" to the seller u_s either through Etherflows ($\mathcal{G}_p$) or asset transfers ($\mathcal{G}_t$). That is, $\mathsf{WCC}(u_b, u_s, \mathcal{G}_t) \vee \mathsf{WCC}(u_b, u_s, \mathcal{G}_p))$ holds.

Rule 5. We define *shill score* as the ratio of the number of times u_b participates in an auction created by u_s and the total number of auctions that u_b participated in. In our detection approach, the *shill score* must be greater than μ, another empirically determined threshold. We set $\mu = 0.8$ for our analysis.

► **Quantitative analysis.** We detected 703 instances of shill bidding across 282 collections involving 1,211 users in all NFTMs except Axie and CryptoPunks. We estimate *shill_profit* as the profit made by the seller due to shill bidding. Specifically, assume legitimate bidders place bids on an item first, and then shill bidding drives the price up. If b_l is the offer made by the last legitimate bidder before shill bidding starts, and the item is finally sold at b_s due to artificial inflation, we compute $(b_s - b_l)$ as the *shill_profit*. According to our analysis, malicious sellers have collected a cumulative profit of $13,014,662 USD from all the shill bidding instances detected. In Figure 6, we show the frequency of shill bidding instances discovered across collections where shill bidding has been detected. The majority (197) of the collections have just one instance of shill bidding, while almost all collections (281) have fewer than 20 shill bids.

The one exception is the official collection of Foundation, which seems to be heavily affected by shill bidding. With 212 instances (30.16% of all instances detected) of shill bids in that collection alone, it becomes the one with the most number of shill bids on any individual collection. Our model also reports frequent shill bidding activity on the official collection of SuperRare (15 instances) and CryptoVoxels (11 instances), which is an OpenSea verified collection with 5.8K items and a cumulative trading volume of 19.2K ETH.

► **Manual analysis.** Since shill bidding often closely resembles legitimate bidding behavior, it is harder to detect than other malpractices. Therefore, to remain conservative during ground-truth determination, we looked for the following conditions: **(i)** the shill bidder S placed at least 3 bids in an auction, and **(ii)** if the average price of the items that S bought is p, and the average bid that S placed in that auction is b, then $p \ll b$, and **(iii)** S never bought any NFT from that seller. We manually verified 100 reports that we randomly selected from the instances that our approach detected. Out of these 100 cases, 61 show strong indications of being instances of shill bidding. For the remaining 39, we could not draw any definitive conclusion from the trading patterns alone. We observed an interesting shill bidding case in Foundation, where the initial reserve price of an NFT was 2 ETH. The item was targeted by a shill bidder who bid $[3.3, 4.4, 5.5, 6.71, 8.14]$ ETH on that item, thereby making the item finally sell at 9 ETH. However, all the NFTs owned by the bidder were worth between $(0, 2]$ ETH, and the bidder never bought any items from that seller.

7.2.3 Bid Shielding. In bid shielding, a malicious bidder u_2 guards a low bid, possibly from a colluding bidder u_1, with a bid high enough to deter legitimate bidders from placing any additional bids. Immediately before the auction ends, u_2 retracts the bid, thus uncovering the low bid from u_1 to let her win the auction.

Detection. We apply the following heuristics to detect instances of bid shielding in NFTMs. If for two users u_1 and u_2, $\mathsf{bid}(u_1, p_1, t_1, id, a)$, $\mathsf{bid}(u_2, p_2, t_2, id, a)$ and $\mathsf{cancel\_bid}(u_2, p_2, t_3, id, a)$ hold, then u_2 is shielding a bid from u_1 if:

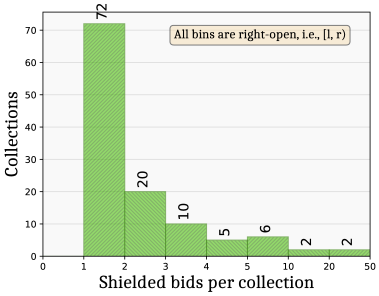

Figure 8: Distribution of bid shielding across collections

Rule 1. For all bids $\{\text{bid}(u_i, \_, t_i, id, a)\}_{i=1}^{n}$ placed on asset a, $t_3 > t_i$ holds, *i.e.*, no new bid was placed after u_2 retracted her bid on asset a.

Rule 2. u_1 won the auction with id id. That is, $\text{win}(u_1, p_1, t_4, id, a)$ holds, and $u_1 \neq u_2 \wedge p_1 < p_2$.

► **Quantitative analysis.** We detected a total 316 instances of bid shielding across 117 collections involving 471 users only in OpenSea. It is expected, because other NFTMs implement bidding policies (Section 3) to deter such malpractices, for example, on-chain bids, removal of the the previous bid when outbid, *etc.* We compute *shielded_bid_difference*, the difference in the bid amounts of the two colluding parties, the potential bid shielder and the auction winner. While the minimum *shielded_bid_difference* amount was $200.77 USD, the maximum was as high as $152,606.31 USD for one of the token in Mirandus Vaults, a verified collection. Additionally, all 316 instances together shielded a total of $942,061 USD worth of bids. Figure 8 shows the number of instances of bid shielding discovered across collections where bid shielding has been detected. For most of the collections (113 out of 117), we find less than ten instances of bid shielding per collection. Ethereum Name Service (ENS), a popular Ethereum name lookup service, makes it to the top of the list with 49 bid shielding instances. Another notable finding in this category was the CryptoVoxels collection. We noticed several complaints by CryptoVoxels collectors on their Discord server about the recent increase of bid shielding activity. According to our analysis, $24,519.27 USD worth of bids were shielded by 35 instances of bid shielding, which corroborates this prior observation. Our results show that bid shielding is frequent in verified collections as 66.67% (78 out of 117) of the bid shielded collections were verified.

► **Manual analysis.** We have manually verified randomly chosen 100 instances flagged by our analysis. During manual analysis, we mark an instance as a *true positive* if **(i)** the potential bid shielder B and the (colluding) auction winner W are the last two highest bidders on that auction in that order, and **(ii)** B cancels her bid just before (≤ 2h) the auction ends, and **(iii)** during the auction, they never outbid each other. Out of the 100 instances, our manual analysis confirms 90 such instances as true positives. Our detection model produced some false positives because it does not take into account the last condition listed above. Let b_i and w_i be the bids from B and W, respectively. Now, first they outbid each other, *i.e.*, $b_1 \longrightarrow w_1 \longrightarrow b_2 \longrightarrow w_2 \longrightarrow b_3$, and then B removes b_3 at the last moment (possibly B just changes her mind). This is *not* a bid shielding scenario, as the bids from B drove up the price for W. This would not happen in a bid shielding scenario as B and W are colluding. However, the first two conditions are still met, and therefore our model incorrectly flags this case. We also observed that most of bid shielding activities are performed in verified collections, such as CryptoVoxels, ENS, *etc.*, as they are popular and in high demand.

8 RELATED WORK

To the best of our knowledge, we are the first to perform an in-depth study of security and privacy risks in the NFT ecosystem. Our paper fits into the recent line of work on cryptoeconomic attacks in decentralized finance (DeFi) systems. The transparency of blockchains opens up the possibility of launching economic attacks by manipulating the market. Since uncommitted Ethereum transactions and their gas bids are visible to other network participants, an attacker can offer a higher gas price to get their malicious transactions mined early in a block, before the victim transaction. This behavior is called *front-running* [49]. The authors in Flash-Boys [47] demonstrated how arbitrage bots front-run transactions in decentralized exchanges (DEX) to generate non-trivial revenues. *Sandwich attacks* take this idea a step further by both front- and back-running victim transactions. Zhou *et al.* [71] quantified the probability of being able to perform such an attack and the profits it can yield. In fact, a recent paper [61] reported the profit extracted from the blockchain to be a staggering $28.8M USD in just two years, leveraging sandwiching, liquidation, and arbitrage. The authors also measured the prevalence of other profit-making operations, *e.g.*, *clogging* and *private mining*. Another DeFi trading instrument, *flashloans*, allows a borrower immediate access to a large amount of funds without offering any collateral, under the condition that the loan needs to be repaid in the same transaction. Qin *et al.* [62] analyzed how flashloans have been used to execute arbitrage and oracle manipulation attacks, and they presented a constrained optimization framework to cleverly choose the attack parameters that maximize the profit. DefiPoser [70] proposes trading algorithms to generate profit by crafting complex DeFi transactions, both with and without flashloans. Recent research [51, 56, 68] has also characterized and quantified *pump-and-dump*, a price manipulation scheme that attempts to inflate the price of a crypto asset by spreading rumors and misinformation.

9 CONCLUSION

This paper conducts the first systematic study of the emerging NFT ecosystem on 8 top NFT marketplaces (NFTM). First, we perform a large-scale data collection from various sources, *viz.*, Ethereum mainnet, NFTM websites, and their documentation. We compile a comprehensive list of design weaknesses originating from the NFTMs and external entities, which often lead to financial consequences. Further, we develop models to detect common trading malpractices, and quantify their prevalence in these marketplaces. For additional discussion and supplementary materials, please refer to the full version of our paper [18].

ACKNOWLEDGMENTS

This material is based upon work supported in part by a Security, Privacy and Anti-Abuse Award from Google and an award from Intel, Corp. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the Google or Intel.

REFERENCES

[1] The 15 most expensive nfts ever sold. https://decrypt.co/62898/most-expensive-nfts-ever-sold.

[2] The art industry and u.s. policies that undermine sanctions. https://www.hsgac.senate.gov/imo/media/doc/2020-07-29%20PSI%20Staff%20Report%20-%20The%20Art%20Industry%20and%20U.S.%20Policies%20that%20Undermine%20Sanctions.pdf.

[3] Axie infinity. https://axieinfinity.com.

[4] Behaviors in the nft ecosystem that we hope will decrease in 2020. https://nonfungible.com/blog/bad-behaviors-nft-blockchain.

[5] Bitmix. https://bitmix.online.

[6] Coingecko. https://www.coingecko.com.

[7] Cryptokitties. https://www.cryptokitties.co.

[8] Cryptopunks. https://www.larvalabs.com/cryptopunks.

[9] Dappradar. https://dappradar.com.

[10] Decentraland. https://market.decentraland.org/.

[11] Dune analytics—opensea. https://dune.xyz/rchen8/opensea.

[12] Echidna. https://github.com/crytic/echidna. [accessed 07/27/2020].

[13] Eth mixer. https://eth-mixer.com.

[14] Ethereum name service (ens) as nft. https://docs.ens.domains/dapp-developer-guide/ens-as-nft.

[15] Fake banksy nft sold through artist's website for £244k. https://www.bbc.com/news/technology-58399338.

[16] Flipkick. https://www.flipkick.io.

[17] Foundation. https://foundation.app.

[18] Full version of this paper. https://arxiv.org/pdf/2111.08893.pdf. 2022/08/26.

[19] Hackers stole nfts from nifty gateway users. https://www.theverge.com/2021/3/15/22331818/nifty-gateway-hack-steal-nfts-credit-card.

[20] How do i get a blue checkmark? https://support.opensea.io/hc/en-us/articles/360063519133-How-do-I-get-a-blue-checkmark-.

[21] How nft giant opensea's $3 billion month compares to amazon, ebay and etsy. https://decrypt.co/79789/opensea-3b-month-ethereum-nft-sales-amazon-ebay-etsy.

[22] Imagehash. https://github.com/JohannesBuchner/imagehash.

[23] Interplanetary file system (ipfs). https://ipfs.io.

[24] Mastering ethereum. https://www.oreilly.com/library/view/mastering-ethereum/9781491971932/ch04.html.

[25] More rari tokens rewarded than the actual art purchase. https://twitter.com/bergleeuw62/status/1293502649308979200?utm_source=nonfungible.

[26] Mythril. https://github.com/ConsenSys/mythril.

[27] Nft mania is here, so are the scammers. https://www.theverge.com/2021/3/20/22334527/nft-scams-artists-opensea-rarible-marble-cards-fraud-art.

[28] Nifty. https://niftygateway.com.

[29] Now postage stamps are getting the nft treatment. https://decrypt.co/61963/now-postage-stamps-are-getting-the-nft-treatment.

[30] Opensea. https://opensea.io.

[31] Perpetual image hash. http://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html.

[32] Rarible. https://https://rarible.com.

[33] Real estate tokenization. https://www.blockchainappfactory.com/real-estate-tokenization.

[34] Signing and verifying ethereum signatures. https://yos.io/2018/11/16/ethereum-signatures.

[35] Social engineering nft users through unauthorized support channel. https://www.theverge.com/22683766/nft-scams-theft-social-engineering-opensea-community-recovery.

[36] Solana nft project accused of rug pull after lil uzi deletes tweets. https://cryptobriefing.com/solana-nft-project-accused-of-rug-pull-after-lil-uzi-deletes-tweets.

[37] Sorare. https://sorare.com.

[38] The specter of shill bidding around nfts - tokensmart nft humpday report 15. https://nft.substack.com/p/the-specter-of-shill-bidding-around.

[39] Superrare. https://superrare.com.

[40] Tornado cash. https://tornado.cash.

[41] Usps certifies casemail as first blockchain generated epostage. https://www.prnewswire.com/news-releases/usps-certifies-casemail-as-first-blockchain-generated-epostage-301267842.html.

[42] What is "wash trading" and why is it negative for non-fungible tokens? https://nonfungible.com/blog/wash-trading-and-why-its-negative-for-non-fungible-tokens.

[43] You can now buy gold-backed nfts with the mining carbon footprint offset. https://cointelegraph.com/news/you-can-now-buy-gold-backed-nfts-with-the-mining-carbon-footprint-offset.

[44] Manticore. https://github.com/trailofbits/manticore/, 2016.

[45] Etherscan. https://etherscan.io/, 2018.

[46] Priyanka Bose, Dipanjan Das, Yanju Chen, Yu Feng, and Christopher Kruegel Giovanni Vigna. Sailfish: vetting smart contract state-inconsistency bugs in seconds. In *Proc. IEEE Symposium on Security and Privacy*, 2022.

[47] Philip Daian, Steven Goldfeder, T. Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and A. Juels. Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. In *Proc. IEEE Symposium on Security and Privacy*, 2020.

[48] William Entriken, Dieter Shirley, Jacob Evans, and Nastassia Sachs. Eip-721: Erc-721 non-fungible token standard, ethereum improvement proposals, no. 721. https://eips.ethereum.org/EIPS/eip-721.

[49] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. Sok: Transparent dishonesty: Front-running attacks on blockchain. In *Proc. Financial Cryptography and Data Security*, 2020.

[50] Joel Frank, Cornelius Aschermann, and Thorsten Holz. ETHBMC: A bounded model checker for smart contracts. In *Proc. USENIX Security Symposium*, 2020.

[51] Neil Gandal, JT Hamrick, Tyler Moore, and Tali Oberman. Price manipulation in the bitcoin ecosystem. *Journal of Monetary Economics*, 95, 01 2018.

[52] Neville Grech, Michael Kong, Anton Jurisevic, Lexi Brent, Bernhard Scholz, and Yannis Smaragdakis. Madmax: surviving out-of-gas conditions in ethereum smart contracts. In *Proc. International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 2018.

[53] Shelly Grossman, Ittai Abraham, Guy Golan-Gueta, Yan Michalevsky, Noam Rinetzky, Mooly Sagiv, and Yoni Zohar. Online detection of effectively callback free objects with applications to smart contracts. In *Proc. Symposium on Principles of Programming Languages*, 2018.

[54] Bo Jiang, Ye Liu, and W. K. Chan. Contractfuzzer: fuzzing smart contracts for vulnerability detection. In *Proc. International Conference on Automated Software Engineering*, 2018.

[55] Sukrit Kalra, Seep Goel, Mohan Dhawan, and Subodh Sharma. ZEUS: analyzing safety of smart contracts. In *Proc. The Network and Distributed System Security Symposium*, 2018.

[56] Josh Kamps and Bennett Kleinberg. To the moon: defining and detecting cryptocurrency pump-and-dumps. *Crime Science*, 7(1):18, Nov 2018.

[57] Kai Li, Jiaqi Chen, Xianghong Liu, Yuzhe Tang, Xiaofeng Wang, and Xiapu Luo. As strong as its weakest link: How to break blockchain dapps at rpc service. In *Proc. The Network and Distributed System Security Symposium*, 2021.

[58] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proc. Conference on Computer and Communications Security*, 2016.

[59] Matthieu Nadini, Laura Alessandretti, Flavio Di Giacinto, Mauro Martino, Luca Maria Aiello, and Andrea Baronchelli. Mapping the nft revolution: market trends, trade networks, and visual features. *Scientific Reports*, 2021.

[60] Tai Nguyen, Long Pham, Jun Sun, Yun Lin, and Minh Quang Tran. sfuzz: An efficient adaptive fuzzer for solidity smart contracts. In *Proc. International Conference on Software Engineering*, 2020.

[61] Kaihua Qin, Liying Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? *ArXiv*, abs/2101.05511, 2021.

[62] Kaihua Qin, Liying Zhou, B. Livshits, and Arthur Gervais. Attacking the defi ecosystem with flash loans for fun and profit. In *Proc. Financial Cryptography and Data Security*, 2021.

[63] Jarrod Trevathan and Wayne Read. *Detecting Shill Bidding in Online English Auctions*, pages 446–470. 01 2008.

[64] Petar Tsankov, Andrei Marian Dan, Dana Drachsler-Cohen, Arthur Gervais, Florian Bünzli, and Martin T. Vechev. Securify: Practical security analysis of smart contracts. In *Proc. Conference on Computer and Communications Security*, 2018.

[65] Friedhelm Victor and Andrea Marie Weintraud. Detecting and quantifying wash trading on decentralized cryptocurrency exchanges. In *Proc. The Web Conference*, pages 23–32, 2021.

[66] Fabian Vogelsteller and Vitalik Buterin. Eip-20: Erc-20 token standard, ethereum improvement proposals, no. 20. https://eips.ethereum.org/EIPS/eip-20. 2021/07/11.

[67] Qin Wang, Rujia Li, Qi Wang, and Shiping Chen. Non-fungible token (nft): Overview, evaluation, opportunities and challenges. *arxiv*, abs/2105.07447, 2021.

[68] Jiahua Xu and Benjamin Livshits. The anatomy of a cryptocurrency pump-and-dump scheme. In *Proc. USENIX Security Symposium*, 2019.

[69] Mengya Zhang, Xiaokuan Zhang, Yinqian Zhang, and Zhiqiang Lin. TXSPECTOR: Uncovering attacks in ethereum from transactions. In *Proc. USENIX Security Symposium*, 2020.

[70] Liying Zhou, Kaihua Qin, Antoine Cully, Benjamin Livshits, and Arthur Gervais. On the just-in-time discovery of profit-generating transactions in defi protocols. In *Proc. IEEE Symposium on Security and Privacy*, 2021.

[71] Liying Zhou, Kaihua Qin, C. F. Torres, D. Le, and Arthur Gervais. High-frequency trading on decentralized on-chain exchanges. In *Proc. IEEE Symposium on Security and Privacy*, 2020.