Hindawi
Security and Communication Networks
Volume 2022, Article ID 3448950, 12 pages
https://doi.org/10.1155/2022/3448950

Research Article

Attacker Traceability on Ethereum through Graph Analysis

Hang Zhu,[1] **Weina Niu**,[1] **Xuhan Liao**,[1] **Xiaosong Zhang**,[1,2] **Xiaofen Wang**,[1] **Beibei Li**,[3] **and Zheyuan He**[1]

[1]*School of Computer Science and Engineering, Institute for Cyber Security, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*
[2]*Blockchain Research Lab of UESTC, Chengdu Jiaozi Financial Holding Group Company Ltd, Chengdu, China*
[3]*School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Weina Niu; niuweina1@126.com

Received 29 July 2021; Revised 12 December 2021; Accepted 28 December 2021; Published 27 January 2022

Academic Editor: Yu Yao

Since the Ethereum virtual machine is Turing complete, Ethereum can implement various complex logics such as mutual calls and nested calls between functions. Therefore, Ethereum has suffered a lot of attacks since its birth, and there are still many attackers active in Ethereum transactions. To this end, we propose a traceability method on Ethereum, using graph analysis to track attackers. We collected complete user transaction data to construct the graph and analyzed data on several harmful attacks, including reentry attacks, short address attacks, DDoS attacks, and Ponzi contracts. Through graph analysis, we found accounts that are strongly associated with these attacks and are still active. We have done a systematic analysis of these accounts to analyze their threats. Finally, we also analyzed the correlation between the information collected through RPC and these accounts and finally found that some accounts can find their IP addresses.

1. Introduction

Ethereum is a distributed platform based on an open source blockchain called Blockchain 2.0 [1, 2]. At the beginning of 2016, the technology of Ethereum was recognized by the market. The price began to soar, attracting many people other than developers to enter the world of Ethereum.

As the price of Ether gradually soared, it also attracted a group of uninvited guests, that is, attackers [3]. They are taking advantage of the loopholes in the Ethereum contract and the loopholes in the EVM to carefully design a variety of attack methods to steal Ethereum or disrupt the Ethereum market. Just like the 2016 DAO attack [4], the attacker exploited long-term lease loopholes to steal approximately 60 million dollars. In July 2017, a loophole in the Parity Wallet contract resulted in a loss of 31 million USD [5]. In April 2018, MyEtherWallet wallet became a victim of BGP and DNS hijacking attacks, enabling hackers to steal US 17 million dollar [6]. On April 18, 2020, hackers took advantage of the compatibility issues between Uniswap and ERC777. When conducting ETH-imBTC transactions, hackers used multiple iterations in ERC777 to call tokensToSend to achieve a reentry attack and exhaust the imBTC pool on Uniswap (ImBTC is an ERC-20 token with 1 : 1 anchoring to Bitcoin). On August 11, 2020, the Ethereum-based token project NUGS had a security problem. There were security loopholes in its smart contract, which caused massive inflation in its token system. Since the security vulnerability of the smart contract cannot be repaired, the NUGS project officially announced the decision to abandon the project, and the tokens deposited in it cannot be withdrawn.

The attackers of Ethereum have existed for a long time along with the development of Ethereum. Even now, there are still many attackers active on Ethereum. There are many ways to detect Ethereum contract vulnerabilities [7–12], and some people have studied the behavior of users on DAPPs [13]. However, few people trace the origin of the attackers.

To this end, after conducting systematic research on the connection between attackers, attack scale, duration, losses, and recent activity, we have proposed to trace the attackers.

In this article, through some previous work, we found some well-known data sets of attacks on Ethereum, including DDoS,

reentry attacks, short address attacks, and Ponzi contracts. After obtaining all the data, we processed it and obtained its direct attack account. Then we inquired about the account related to it through graph analysis. Through the analysis of the inquired account, we have the following findings:

(i) In the same type of attack, many attack accounts are connected, indicating that they are all from an attacker or attack team

(ii) Ether still exists in these linked accounts, but most accounts have only a small balance, and a small number of accounts have a large amount of Ether

(iii) Except for DDoS attacks, all other attacks have accounts active on recent Ethereum blocks

Then we use the RPC mechanism to link the attacker data obtained from the graph analysis with the IP. In summary, we make the following major contributions:

(i) We use graph analysis technology to systematically analyze the attacker and analyze the behavior characteristics of the attack.

(ii) Through the study of other documents, a relatively complete data set of several types of attacks with more research value has been collected. Perform graph analysis on these data sets. We found a correlation between the vast majority of attackers and found that many accounts are still active on Ethereum.

(iii) We have conducted statistics on accounts with strong correlation, calculated the damage caused to Ethereum, and analyzed their overall activity.

(iv) We propose a method to trace the source of the attack. By using the data obtained from the RPC call, we can associate the attacker with a specific IP address.

The rest of this article is organized as follows: Section 2 introduces some background work of this article. Section 3 introduces the method of this paper. Section 4 introduces the data collection method. Section 5 introduces how to establish graph relationships, Section 6 introduces the experimental results, Section 7 discusses a flaw and future work of this paper, and Section 8 summarizes the full paper.

2. Background

The blockchain is essentially an immutable distributed transaction ledger with ever-increasing data [14]. Distribution is reflected in the absence of a clear central organization in the blockchain system. All the complete client nodes in the system are equal, and they all keep a copy of the same ledger. The ledger contains all confirmed transactions in the blockchain system. These transactions are usually stored in the form of a Merkle tree and packed into each block. Each block points to the header hash value of the previous block. In this way, all the blocks are connected into a long chain. The chain formed by this block is called a blockchain [15].

Ethereum is a decentralized virtual machine using blockchain technology. Smart contracts are programs written in Turing-complete bytecode language that can run on it [2]. Smart contracts can build various applications, from simple wallet applications to complex financial systems in the banking industry. Accounts can create new contracts on the Ethereum network, call contract functions, and implement functions such as transferring Ether. However, at the same time, there will be many vulnerabilities in smart contracts, and these vulnerabilities may be solidity design flaws or negligence in the process of writing [9].

Chen [10] has designed a set of frameworks on which to write corresponding APPs to detect different types of attacks. Using this framework can obtain a large number of attack data sets [16]. After detecting the RPC attack, this paper simply traced the source of the attacker. This may be due to the different attack methods. There is not much value. Reference [17] traced the source from the network layer; this method is not suitable for blockchain [18]. This article systematically analyzes and investigates the Ponzi contract. We also borrowed some of his indicators when we started our work. Below we will introduce the type of attack on the data set we used in the experiment:

2.1. Reentry Attack. Ethereum smart contracts can call and utilize the code of other external contracts. Contracts usually also deal with Ether, so Ether is sent to various external user addresses. The operation of calling an external contract or sending Ether to an address requires the contract to submit an external call. These external calls can be hijacked by an attacker, thereby forcing the contract to execute more codes (that is, through the fallback function), including callbacks to the original contract itself. Therefore, the contract can be "reentered" during the execution of the contract code, which is a bit like an indirect recursive function call in a programming language [19]. In the infamous DAO incident, hackers used this attack, which eventually led to the hard fork of Ethereum [20]. This attack may occur when the contract sends Ether to an unknown address. An attacker can carefully construct a contract at an external address that contains malicious code in the fallback function. Therefore, when the contract sends Ether to this address, the malicious code will be activated. Usually, malicious code executes functions on vulnerable contracts, which is an operation that developers did not expect. The name "Reentrancy" comes from the reality that the external malicious contract calls back a function on the attacked contract and "reenters" code execution anywhere on the attacked contract. Because the programmer of the original contract may not have anticipated that the contract code can be "reentered," the contract will exhibit unpredictable behavior [21].

2.2. DDoS Attack. Denial-of-service attacks are designed to destroy the access to the network or resources of a specific target. A denial-of-service attack launched by attackers in multiple locations simultaneously is called a distributed denial-of-service attack [22]. Once a distributed denial-of-service attack occurs in the blockchain, the entire blockchain network may face paralysis. Sending transactions in Ethereum requires a certain amount of gas (transaction fee). This mechanism can resist DoS attacks to a certain extent. However, because the gas consumption price of Ethereum's

EXTCODESIZE opcode is too low, the attacker has successfully launched a DoS attack on Ethereum. The EXTCODESIZE opcode is used to read the wise contract code size. When EXTCODESIZE is called, the node needs to read the status information of the disk. Since EXTCODESIZE only consumes 20 gases, an attacker can perform 50,000 EXT-CODESIZE operations in one transaction. This kind of attack can consume a lot of computing resources and network resources of the blockchain network, leading to congestion or even paralysis of the blockchain network. Some attackers also used the SUICIDE opcode to launch DoS attacks [23]. The attackers used the SUICIDE operation to generate 19 million empty accounts. Since empty accounts need to be stored in the state tree, a large number of empty accounts waste disk resources. In this attack mode, the node synchronization speed and transaction processing speed of the blockchain network have dropped significantly. Excessive authority of smart contract token owners may also lead to denial-of-service attacks. If the token contract owner keeps freezing the contract, other users in the contract will not be able to conduct transactions:

```
assembly{
let seed := calldataload(4)
let iterations :=calldataload(36)
let target :=seed
loop:
target := add(target,seed)
pop(call(0,div(target,0x100000000000000000000000),0,0,0,0,0))
iterations := sub(iterations,1)
DDoS code
```

2.3. Short Address Attack. A short address attack means that the attacker makes a contract call by constructing an address with zero at the end and deliberately discards the zero at the end of the address in the call parameters, thereby using the virtual machine's automatic completion mechanism for the data to perform the second parameter [3]—shift zoom. Short address attacks usually occur in exchanges. Suppose a user initiates a transfer operation to a contract on the exchange and uses a maliciously constructed short address as the target. If the exchange does not verify the length of the user input, the actual transfer amount will be enlarged several times due to the short address loophole, which will cause a large amount of capital loss. The root of the short address vulnerability is that when the virtual machine reads the input of the contract call, it automatically fills the fields with zeros at the end, which causes the ambiguity of the data and the shift and expansion of the parameters.

CALLDATALOAD reads 32 bytes from the input data. If the input data is not enough 32 bits, it will add 0 bytes. Attackers can use this design to perform short address attacks to steal tokens [24].

2.4. Ponzi Contract. Ponzi scheme is a traditional investment scam. Its typical feature is to use funds provided by new investors to pay the so-called returns to exist investors [25]. In the Ethereum smart contract, the Ponzi scheme has some new features. The most obvious is that it is based on the anonymity of the blockchain. Researchers cannot know the true identity of the contract initiator, cannot associate their credit information, and can only analyze the information disclosed on the Ethereum official website. In addition, its code is public, immutable, and automatically enforced, so investors will have a sense of trust in it and reduce awareness of prevention. Because of this, Ponzi schemes on smart contracts are also emerging endlessly. Many investors mistakenly invested in the Ponzi scheme because they could not understand the source code of the smart contract and ultimately suffered heavy losses [11]. In addition, due to the anonymity of the blockchain and the difficulty of traceability, the defrauded funds are basically impossible to be recovered successfully, and investors can only suffer losses. Many blockchain Ponzi schemes have been exposed to obtain huge profits. Between 2013 and 2014 alone, Bitcoin's Ponzi scheme gained 7 million dollars in illegal benefits. Therefore, it is necessary to strengthen the supervision of the blockchain market. Ponzi contract is a contract that defrauds investors of Ether through its own code logic. Specifically meet the following four characteristics: 1. Require the contract to allocate funds to investors, that is, users who join the contract by remittance to the contract. 2. It is required to obtain only the money collected in the contract from investors. 3. Require every investor who wants to make money, provided that the new investor continues to remit money to the contract. 4. As the plan progresses, investors will be lost [18].

Timestamp dependence [26] and block high dependence: Timestamp dependence means that intelligent contracts use strict block timestamps in their code to make important control flow decisions, thereby introducing security vulnerabilities. The block timestamp refers to the timestamp when the block to which the current contract call transaction belongs is packaged. The block timestamp seems to have a specific contingency. However, it can be manipulated by miners within a specific range of values. Suppose the precise timestamp is used as an essential decision parameter in the contract, although it is irresistible to ordinary attackers. In that case, the attacker with the miner's identity can easily bypass it by constructing a malicious timestamp within the value range—restrictions on the use of timestamp design in contracts. These two dependency issues mean that certain judgments in the contract are based on these two, and the miner can control the block number and timestamp. Therefore, miners can influence the execution of smart contracts [27]. Table 1 lists some attacks on Ethereum.

2.5. Motivation. Although there are many researches on the traceability of blockchains [28], few people study the traceability of attacks on the blockchain. The reason is that, in Ethereum, when an attack is detected, the Ethereum account is anonymous, and the attacker may control or use other "intermediate smart contracts or intermediate users"

Table 1: Some attacks on Ethereum.

Date	Money	Events
6/17/2016	100 million USD	The DAO
1/26/2018	1080.6	Coincheck exchange was attacked
2/10/2018	917.4	BitGrail exchange XRB stolen
3/7/2018	825.607	Ethereum "smuggling" vulnerability
3/30/2018	409.929	OKEx exchange suffered abnormal transactions
4/9/2018	429.251	Coinsecure exchange wallet stolen
4/22/2018	640.767	Smart contract BEC US chain was attacked
4/24/2018	708.875	Smart contract SMT was attacked
4/25/2018	707.062	MyEtherWallet wallet website was attacked
5/23/2018	651.636	Smart contract EDU was attacked
6/10/2018	594.345	Coinrail exchange stolen
6/20/2018	541.015	Bithumb exchange stolen
7/8/2018	503.199	Bancor exchange private key leaked and attacked
11/27/2019	5000	Hackers attacked UpBit exchange
8/2019	—	Zerocoin attacks Citex

to attack. Therefore, even if an attack threat is detected in the blockchain, it cannot be traced back to the attacker. Therefore, there is an urgent need to study reasonable traceability technology to effectively, quickly, and accurately trace the source of the attacker and realize the statistics and traceability of asset losses on the blockchain.

Because the relationship presented by users based on Ethereum is a kind of graph, it is more reasonable to use graphs to map the flow of funds, call relationships, and establish relationships when studying attackers' attack behavior.

3. Methodology

In this paper, our method consists of three stages, which are described in the following sections. The first stage is data collection, collecting all attack data, including reentry attacks, DDoS, short address attacks, and Ponzi contracts mentioned in this article. These four have a more significant impact on the Ethereum ecosystem and have a far-reaching impact [29].

As shown in Figure 1, in the second stage, the framework in [10] is used to collect complete transaction data, including internal transactions and external transactions, and then build different relationship graphs based on the attack data set of the first stage because different attack types have very different natures. For example, DDoS attacks do not involve profit from the attack, so we do not need to pay too much attention to the flow of money. We only need to establish a call relationship graph to perform a correlation analysis [30].

In the third stage, based on the correlation analysis of the statistical data of these graphs, we can obtain a group closely related to the attacker. Analyze the attacker group: how many associated attackers are there? After obtaining these data, we use etherscan to investigate its current activity, focusing on the amount of Ether held by these accounts and the activity of recent transactions.

In addition to all the work mentioned above, we are still trying to trace the IP addresses of these attackers [31]. The specific method is to refer to [16]. This paper used a honeypot method to collect a large number of attack calls sent by the attacker. The parameters in the request included the attacker's account address. Then we performed a correlation analysis between the account addresses of these attackers and the relationship graph we established when analyzing the contract attackers. The basis for this is that the attacker may also perform RPC call attacks while making contract attacks. It may be said that the account addresses used by the attackers on both sides are the same address. Therefore, it can be found through correlation analysis to the IP address of the account [32, 33].

In this paper, the critical issue is the creation of graphs. Our method is to collect all transaction data during the operation of the blockchain and extract the three main activities of the attacker, namely, transfers and smart contracts. Creation and smart contract invocation, respectively, construct the money flow graph, smart contract creation graph, and smart contract invocation graph. With these three graphs, we can conduct a comprehensive analysis of the attacker.

4. Data Collection

The first thing that needs to be collected is offensive behavior. Many existing papers do not have a complete collection of offensive transactions. Most of the articles are used to detect contract vulnerabilities. The framework proposed in [10] can not only be used to detect contracts with vulnerabilities but also detect transactions that exploit contract vulnerabilities. A total of eight types of vulnerability transaction detection are proposed, including reentry, Unexpected Function Invocation, Invalid Input Data, Incorrect Check for Authorization, No Check after Contract Invocation, Missing the Transfer Event, Strict Check for Balance, Timestamp Dependency, and Block Number Dependency. We selected the type of vulnerability that has a significant and far-reaching impact on Ethereum, that is, reentry attacks. A total of 3291 transactions were received, for which reentry attacks were detected. Secondly, there is Invalid Input Data, which is a short address attack.

Then there is the Ponzi contract. The impact of the Ponzi contract is relatively long-term and far-reaching. The detection of the Ponzi contract is mainly from the Ponzi

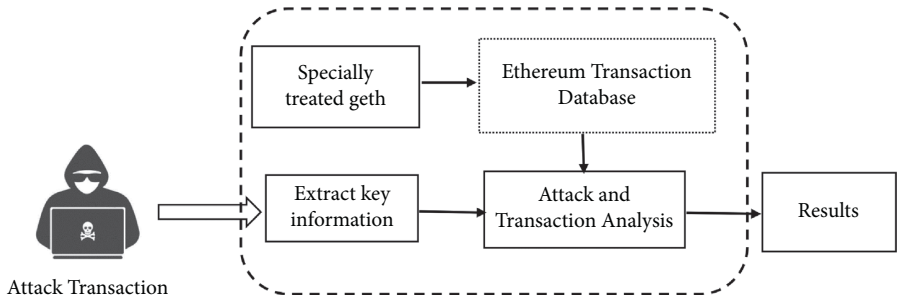

Figure 1: The overview of our system.

contract mentioned in [18]. There is a total of 140 Ponzi contracts. DDoS attacks are mainly contracts for DDoS attacks that existed before 2.3 million and 2.8 million.

After getting these attacks, we still need transaction information about these attacks, including the creator of the contract, the caller of the contract, and the internal transaction information in the completed transaction [34]. This information is obtained by using the framework mentioned in [10] to obtain complete transaction information. Because this framework can write related plug-ins on its basis, we will specifically do it by calling the interface provided by the framework in the plug-in. The information that the interface can provide is each opcode executed after a transaction starts, and then each different opcode can get corresponding information. The sign of the start of an external transaction is "EXTERNALINFOSTART," and then the block number, transaction hash, both parties of the transaction, and the amount of the transaction can be obtained in the interface and stored in a structure. When it encounters "EXTERNALINFOEND," the external transaction ends. The information that can be obtained at this time is whether the transaction is successful and all the gas spent by the exchange, and then all the data is written [35]. Then the signs of the start of internal transactions are mainly "CALLSTART," "CREATESTART," "CREATE2START," "CALLCODESSTART," "DELEGATECALLSTART," "STATICCALLSTART," and then collect the transaction amount of both parties during this period. When encountering "CALLEND," "CREATEEND," "CREATE2END," "CALLCODESEND," "DELEGATECALLEND," "STATICCALLEND," it is the end of the transaction, and then save all the collected results to the local [36].

In addition to collecting complete transaction data in this way, we can also obtain complete data in a transaction and all internal transactions through the Ethereum data website etherscan [37]. Nevertheless, there will be a problem when using etherscan because etherscan considers the bandwidth issue when designing the API, so if the transaction volume of the contract is more than 10,000, the list of returned transactions will not return all transactions [38]. This leads to a problem that the data is incomplete.

If we use web3 to connect to geth, we can get a complete list of transactions, but we cannot get the internal transactions of each transaction [39]. For example, in a reentry attack, many of the transfers are performed in internal transactions. Therefore, transactions cannot be obtained in this way.

In addition, we can obtain relevant data through Google's bigquery [40]. Bigquery has a large amount of complete data (such as blocks, tokens, contracts, transactions, etc.), which can be used to obtain all contract addresses on Ethereum (including open source and closed source addresses). However, for transactions, only general information about a transaction can be obtained, and a complete list of transactions cannot be obtained.

In the process of constructing the attacker relationship graph, the existence of the trading market will have a significant impact, so in the process of constructing the graph, it is necessary to remove the trading market in the graph. We collect trading markets, mainly through the list of trading markets given in the website [41]. Then search for the name of the trading market in etherscan to get its hash address.

5. Graph Construction

This part mainly refers to the method used to analyze Ethereum in [30, 42]. It mentioned the construction of three relationship graphs CCG, MFG, and CIG. It can analyze important security issues, including attack forensics, anomaly detection, and deanonymization. In this article, there are similarities with [30] in attack forensics. However, paper [30] did not specifically do a lot of traceability analysis. There are many complicated situations in actual attacks, and simple backtracking cannot get complete traceability data. Moreover, there will be a trading market during the trading process, which will significantly affect the accuracy of the final result.

As shown in Figure 2, nodes with different colors mean different types of attacker groups. These groups are closely connected. This article proposes different ways to trace the source of different types of attacks on its basis. In this article, the most important types of attacks are divided into four categories. The first category is the contract with the nature of the attack. This category mainly refers to the attacker calling the contract created by himself to achieve the purpose of the attack, such as the DDoS attack contract.

The second category refers to attacks on contracts with vulnerabilities. This type of attack mainly involves vulnerabilities in the victim's contract. The attacker constructs corresponding parameters to steal tokens or eth, such as short address attacks and integer overflow attacks.

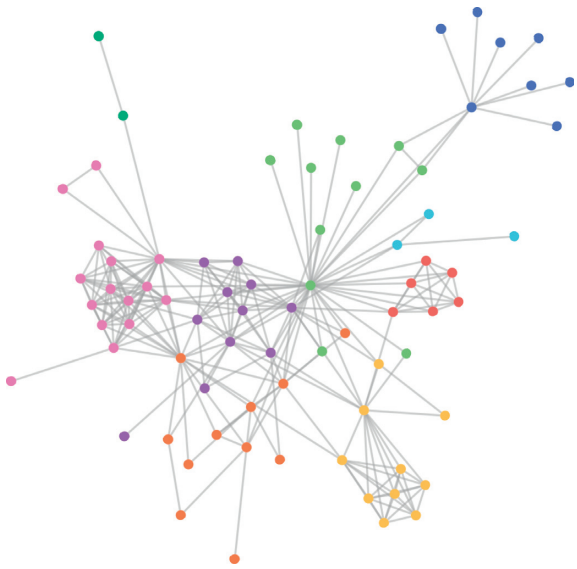

Figure 2: Example diagram of Ethereum account transaction.

The third type of attack is to call the contract attacker to attack the vulnerable contract to steal token or eth, such as the famous reentry attack.

The fourth type of attack refers to phishing contracts, in which transaction initiators are victims, such as Ponzi contracts. The essential characteristics of these four different types of attacks are quite different. Therefore, it is necessary to adopt different methods to trace the source. Traceability methods are as follows.

5.1. DDoS. The method here is to query the creator of the collected DDoS attack contract list. If the contract account is created, then continue to trace back and record all accounts with transaction records. Backtracking until it is an EOA account, all contract accounts during this period must record the accounts with which transactions are made. This is to complete the second round of inquiries and then perform the third round of transaction records inquiries for all addresses obtained in this round of inquiries. During the inquiry process, if a trading market is encountered, the backtracking of the branch is terminated. Then compose a graph of its creator with the results obtained. After completing the creator relationship diagram, the second and third rounds of backtracking queries are performed on the contract caller. Unlike the creator, the caller does not need to go back to the end of EOA.

5.2. Short Addresses. First, we need to collect the data of the detected attack behavior and find all the account addresses that initiated the attack. If the attacker's account is a contract address, it must be traced back to its creator until it is an EOA account. This requires recording the addresses in the middle and then starting the second and third rounds of backtracking together. If it is an EOA account, start the retrospective query directly. Unlike the first type, this type of transaction involves the transfer of funds. Therefore, transactions involving the transfer of funds will be highlighted.

5.3. Reentry Attacks. The characteristic of this type of attack is that the attacker first creates a particular contract and then calls this contract to attack the vulnerable contract. Therefore, for this type of attack, the contract's creator must first be backtracked. Then there are often a large number of internal transactions in this type of attack. The flow of funds is also involved in the internal transactions. So the difference from the previous two is that the external transaction call needs to be recorded and the internal transaction caller. Then go back to the second and third rounds.

5.4. Ponzi. Because it is a phishing attack, the measures taken against this type of attack are mainly aimed at the creator of the contract and the final flow of funds. Like the aforementioned categories, it is necessary to perform a backtracking query on the creator to create a creator relationship graph. Then find the beneficiaries in the transaction list of the Ponzi contract, and then conduct the second and third rounds of retrospective inquiry.

6. Result Analysis

Our experimental computer environment is 32G memory, Intel(R)Core(TM)i9-10920X@3.50 GHz CPU, 1 TB SSD. This section analyzes the source tracing results of four types of attacks. These four attacks are selected for analysis because these four attacks cause more significant harm to Ethereum, rarely false positives, and the amount of data is relatively small for data processing. We will describe the results first and then introduce the information we get from the results and then analyze the results according to the indicators.

6.1. DDoS. Through the analysis of DDoS transactions, a total of 25 attacker contract contracts were collected. The total transaction volume was 109,607, and the internal transaction volume was about 188,680,133. The internal transaction distribution is shown in Figure 3. The first block is 2,286,910, and the last block is 2,720,289. We found that there are 14 creators in these 25 contracts, which means that many contracts have the same creator. Then we scanned the callers of these 25 contracts and found that there were 841 callers.After our analysis, we found that a large number of callers were invalid calls. The contract creators are all EOA users. Then we put these users into the results of the first round of retrospective query for analysis and found that these 25 contracts can be divided into four groups and related. Then we put the creator data into the second-round retrospect and found that the 25 data can be divided into two groups of related data. The first group contains 23 contracts as shown in Figure 4 and the second group has only two contracts. To prove that the remaining two sets of data are connected, we used etherscan to conduct manual analysis. However, it turns out that there is no correlation between the

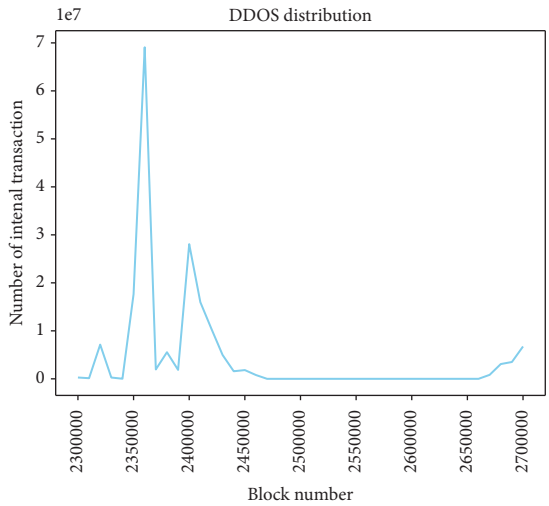

Figure 3: The internal transaction distribution of DDos attack.

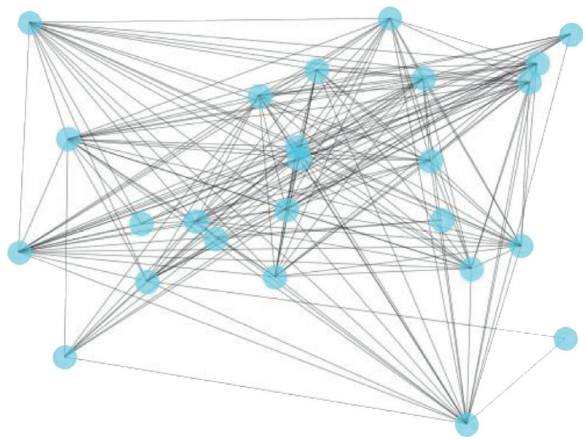

Figure 4: The relation of DDos attack accounts.

two sets of data, and there is a big gap between the two sets of DDoS attack contract codes. Moreover, the data of the first group is distributed from 2.2 million to 2.7 million. In contrast, the data of the second group only exists in the block after 2.67 million. We investigated all of its associated accounts and found that all of its Ether was transferred to the trading market in the block number about 3 million. These accounts only had a small amount of Ether, and no more transactions were conducted afterward.

Ethereum is developing rapidly, and the sudden DDoS attack caused significant losses to the initial Ethereum. During this period, the transaction volume and block generation speed have significantly decreased, but this also shows that Ethereum is very powerful. The Ethereum developers subsequently succeeded in patching the vulnerabilities, which can be seen from the 25 contracts that stopped the attack after 2.7 million. For attackers, there are indeed some hackers who cheat in order to obtain bonuses in a particular game, but this should be relatively short-lived. The 25 contracts mentioned in this article have well-designed code and extensive deployment and invested a lot of money and time to launch attacks. This is more like a business competitor with Ethereum.

6.2. Short Address Attack. We detected a total of 270 short address attack transactions. After deduplication, we found that these short addresses had 52 different account addresses. Compared with DDoS, short address attacks are less costly, easier to implement, and relatively less harmful. We initially thought that there were more people involved in the attack from the perspective of organizing an attack, and there would not be much correlation between them. However, the result of our experiment was beyond our expectations.The active blocks of short address attacks ranged from 7934974 to 8175517. Among the 270 attacked transactions, only 160 were successfully executed. Through graph analysis technology, we found that, in the first round of backtracking results, we found that there are only a few interrelated situations among the 52 account addresses. Among them, all accounts can be divided into 12 groups according to the connection situation. The largest group contains 16 accounts. At this time, we were a bit lost, thinking that there would be no surprises. Then we checked the second round of backtracking results and found that all accounts are related, as shown in Figure 5. That is to say, these accounts are likely to come from a hacker or an organization. We took out all the account addresses at the link between the attacked accounts, a total of 3,465,302. After our analysis, we found that most of the addresses here are EOA addresses. Among them, many users' recent transaction block is after 10 million.

6.3. Reentry Attack. In the reentry attack, the first transaction detected was in the 1718497 block. It has been continuously discovered since then: until the 2130761 block, a total of 1849 reentry attack transactions were detected. A total of 18 attackers were involved. The reentry attack caused a total loss of 21207208.88995413 Ether and a token worth 225136933535.86 dollars. We found that, through graph analysis technology, we found that, in the first round of retrospective results, we found that these 18 addresses can be divided into two groups of irrelevant data. However, in the second round of backtracking results, we were pleasantly surprised to find that these 18 attacker addresses are all related to each other, and the transactions between them are very frequent as shown in Figure 6. Therefore, we have reason to believe that these 18 attackers are all from the same organization or individual.

We found through graph analysis technology that, in the first round of retrospective results, we found that these 18 addresses can be divided into two groups of irrelevant data. However, in the second round of backtracking results, we were pleasantly surprised to find that these 18 attacker addresses are all related to each other, and the transactions between them are very frequent. Therefore, we have reason to believe that these 18 attackers are all from the same organization or individual. We got all the suspicious accounts and found a total of 3,208,251, most of which are EOA addresses, and there is a small amount of trading market. After removing the trading market, there are a large number of users whose recent trading blocks are after 10 million, but only a small amount of funds exists on these accounts.

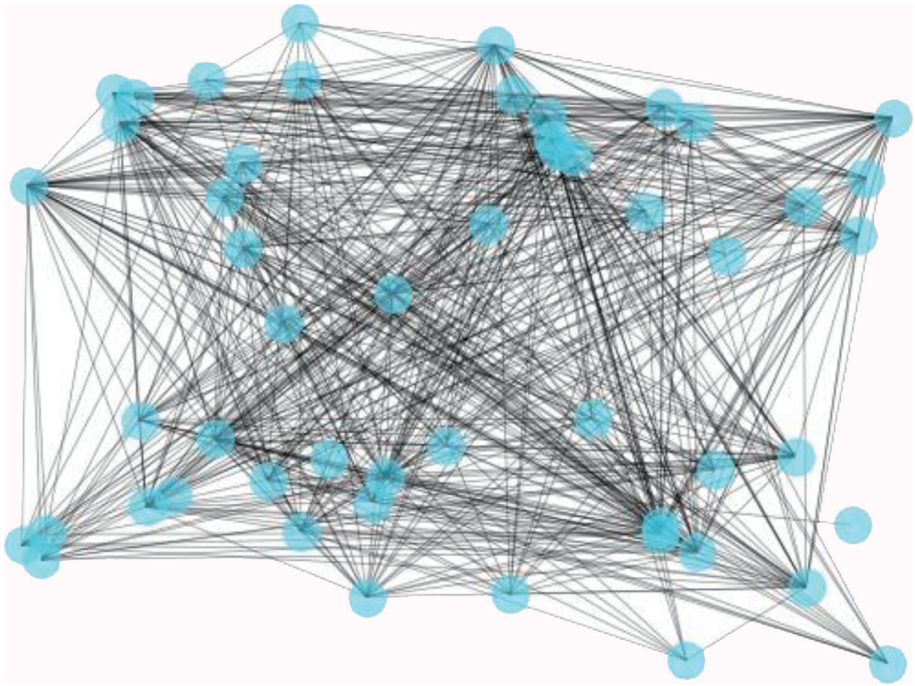

Figure 5: The relation of short address attack accounts.

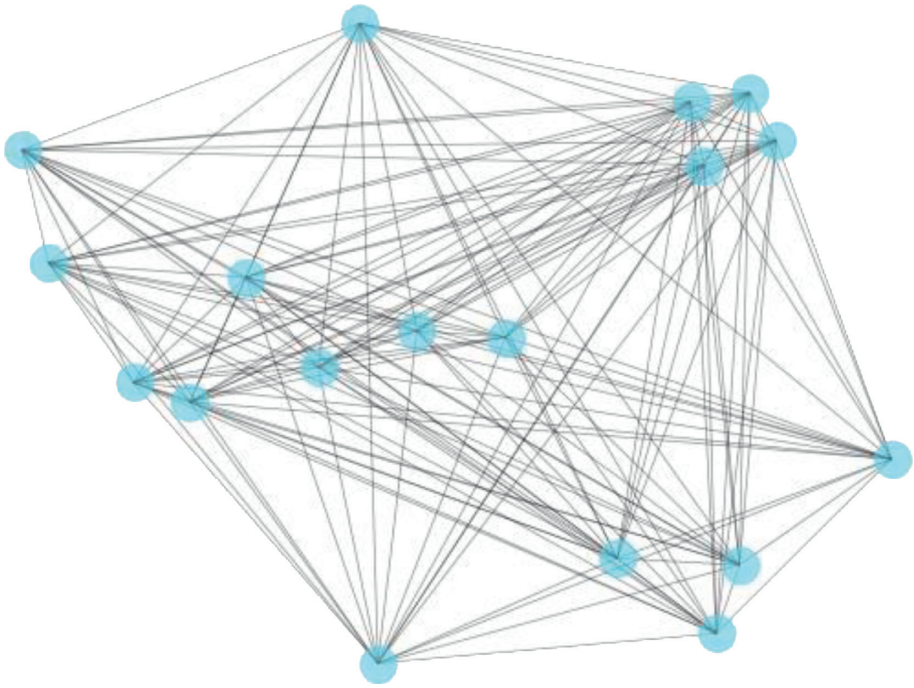

Figure 6: The relation of reentry attack.

6.4. Ponzi Contract. There are a total of 140 data sets of Ponzi contracts. There are 74 creators of Ponzi contracts in the past. In the traceability process of Ponzi contracts, only the creators need to be paid attention to. These 140 contracts have resulted in more than 600,000 contracts. Economic losses: the number of victims reached more than 2,000, which has had a considerable impact on the Ethereum ecosystem. Among these 74 creators, through graph analysis technology, we did not find an excellent correlation between the creators in the first round of backtracking results. It is found that these 74 creators are all related in the second round. The strength of the association between these accounts is shown in Figure 7.

We then took out the account addresses at the link between the attacked accounts. There were only 599 account addresses in the first round, but the second round was vast. A

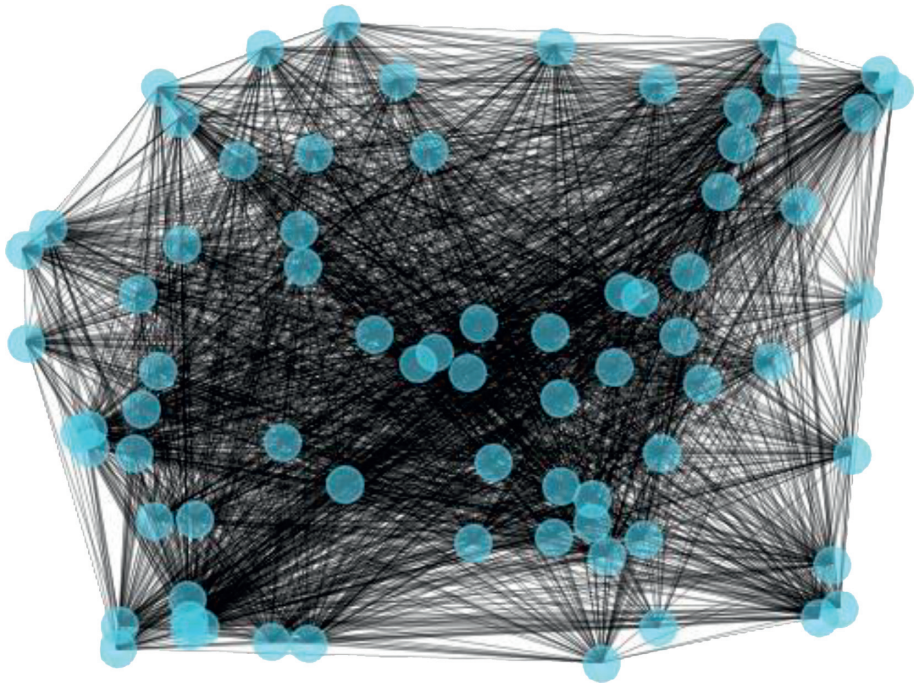

Figure 7: The relation of Ponzi creators.

total of 11,331,544 account addresses were found in these addresses. Among them, most of them are EOA addresses, and there is many trading markets. After removing the trading market, our sample survey found that 80% of these accounts were active in the block about 2 million, and then there was no activity. Some users' recent trading blocks are after 10 million. However, there are only a tiny amount of funds on these accounts.

6.5. Attacker Activity Analysis. This subsection mainly discusses the account balance, the latest transaction block distribution of the account, the distribution of transaction volume, and the nature of the account. We used sampling methods to randomly extract part of the data from the data set and remove the trading market from it for the associated accounts we found from DDoS, Ponzi contracts, short addresses, and reentry attacks. Among them, 1446 accounts were extracted from DDoS, including 911 EOA accounts, 2430 accounts were extracted from the Ponzi contract, including 2430 EOA accounts, 2712 were extracted from short address attacks, and 2712 were EOA accounts, and reentry attacks were extracted. One thousand one hundred accounts were extracted from reentry attacks, including 1100 EOA accounts. It can be seen that there are a large number of EOA accounts in these accounts. It can be seen in Table 2 that there are still a large number of funds in these accounts, but due to the relatively large account base, there is not much Ether on average. Active accounts refer to accounts that have sent transactions after the 9 million blocks. Only two accounts were found in DDoS, accounting for only 0.13%. Even so, there is a large number of funds in these two accounts. In the data associated with the creator of the Ponzi contract, it was found in the sample that 15.9% of the accounts are still active on Ethereum, and there is a large amount of Ether on these accounts. This is the most active type of attack. 12.3% of accounts associated with short address attacks are active on Ethereum, but the amount of funds is relatively small. 7% of the linked accounts in the reentry attack is still active on Ethereum, and there is also a large amount of Ether.

As can be seen from Figure 8, in addition to DDoS, other types of accounts are active in recent transactions.

6.6. Connections between Various Types of Attacks. We are curious about whether there is a connection between the attackers of various attacks, so we have analyzed the correlation between the suspicious account data sets in the various attacks. It turns out that there is indeed a correlation among the four types of attacks. There are 1446 suspicious accounts in our sample in DDoS, 2427 in Ponzi, 1097 in reentry attacks, and 2712 in short address attacks. In the end, it was discovered that there were 10 accounts related to DDoS and Ponzi, three of DDoS and reentry attacks, and two of Ponzi and reentry. Therefore, it can be seen that various types of attackers are also connected, and they may come from the same attacker or attack organization.

6.7. IP Traceability. We perform a correlation analysis between the data set obtained from graph analysis and the data set obtained from the RPC honeypot [16]. We finally found that two addresses are related. Then we took out all the accounts involved between these two addresses for analysis and found that there may be a trading market.

Table 2: The balance of related account.

	DDoS	Ponzi	Short address	Reentry
Average balance	0.7523620102737771	2.7769269276979367	0.7410036026692691	1.916695408323441
Active accounts (%)	0.13	15.9	12.3	7
Active accounts' average balance	18.909574226213486	7.97303457016362	2.4134015471282093	15.944400954564028

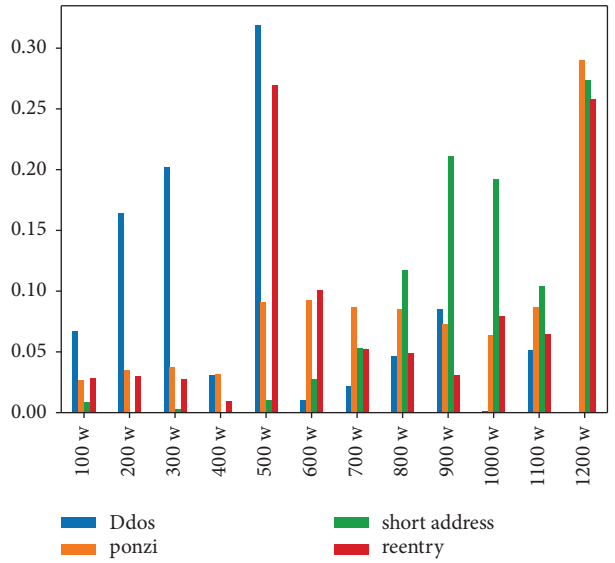

Figure 8: Block distribution of the latest transactions of suspicious accounts.

7. Discussion

In this paper, although we can connect the attacker with his IP address, we still need to make improvements. After our analysis, there are the following points.

First, there is a gap between the types of attacks, which means that people who use the four attack methods mentioned in this paper will not be RPC attacks. Second, the data set obtained in the RPC honeypot we used is too small.

For these two points, in future work, we will evaluate the behavior of the attacker and find a data set with a higher probability. Then, it is to expand the data set obtained in the RPC honeypot, extend the honeypot collection time, or collect IP in other ways.

8. Conclusion

For Ethereum research, a lot of current work is based on smart contract vulnerabilities and some problems of Ethereum itself. This article carried out research on the attacker and proposed a way to trace the attacker through graph analysis. Through the traceability analysis of reentry attacks, DDos, short address attacks, and Ponzi contracts, it is found that the attackers behind a large number of attacks are interconnected, and there are still a large number of active accounts. Finally, by using the RPC mechanism, we can find the IP addresses of certain attackers. With these results, we can study the behavior of some attackers in a targeted manner, so that the security of Ethereum can be further improved [43].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. U19A2066), the Opening Project of Guangdong Provincial Key Laboratory of Information Security Technology (no. 2020B1212060078), the Sichuan Science and Technology Plan Projects, Key Research and Development Projects (no. 2020YFG0294), and Chengdu Science and Technology Project, Key R&D Support Program, Major Science and Technology Application Demonstration Project (no. 2019-YF09-00048-CG).

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, https://bitcoin.org/bitcoin.pd.

[2] V. Buterin, "Ethereum: a next generation smart contract and decentralized appli-cation platform," 2013, https://github.com/ethereum/wiki/wiki/White-Paper.

[3] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: vulnerabilities, attacks, and

defenses[J]," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–43, 2020.

[4] P. Daian, "Analysis of the DAO exploit," 2016, http://hackingdistributed.com/2016/06/18/analysisof-the-dao-exploit/.

[5] P. Santiago, "The parity wallet hack explained," 2017, https://blog.zeppelin.solutions/on-theparity-wallet-multisig-hack-405a8c12e8f7.

[6] L. Poinsignon, "BGP leaks and crypto currencies," 2018, https://blog.cloudflare.com/bgp-leaks-andcrypto-currencies/.

[7] J. Huang, S. Han, W. You et al., "Hunting vulnerable smart contracts via graph embedding based bytecode matching," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2144–2156, 2021.

[8] T. Hu, X Liu, T Chen et al., "Transaction-based classification and detection approach for Ethereum smart contract," *Information Processing & Management*, vol. 58, no. 2, Article ID 102462, 2021.

[9] P. Praitheeshan, L. Pan, J. Yu, J. Liu, and R. Doss, "Security Analysis Methods on Ethereum Smart Contract Vulnerabilities: A survey[J]," 2019, https://arxiv.org/pdf/1908.08605.pdf.

[10] T. Chen, R. Cao, T. Li et al., "SODA: a generic online detection framework for smart contracts," in *Proceedings of the Network and Distributed System Security Symposium*, 23 February 2020.

[11] S. Fan, S. Fu, H. Xu, and X. Cheng, "Al-SPSD: anti-leakage smart Ponzi schemes detection in blockchain," *Information Processing & Management*, vol. 58, no. 4, Article ID 102587, 2021.

[12] N. Anita and M. Vijayalakshmi, "Blockchain security attack: a brief survey," in *Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, Kanpur, India, 6 July 2019.

[13] Yu Wang, Z. Li, G. Gou, G. Xiong, C. Wang, and Z. Li, "Identifying DApps and user behaviors on ethereum via encrypted traffic," in *Proceedings of the International Conference on Security and Privacy in Communication Systems*, Springer, Washington, DC, USA, 21 October 2020.

[14] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger[J]," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[15] D. Vujičić, D. Jagodić, and S. Randić, "Blockchain technology, bitcoin, and Ethereum: a brief overview," in *Proceedings of the 2018 17th international Symposium Infoteh-Jahorina (Infoteh)*, pp. 1–6, IEEE, East Sarajevo, Bosnia and Herzegovina, 21 March 2018.

[16] Z. Cheng, X. Hou, R. Li et al., "Towards a first step to understand the cryptocurrency stealing attack on ethereum," in *Proceedings of the 22nd international Symposium on research in Attacks, Intrusions and Defenses (RAID 2019)*, Usenix, Beijing, China, 23 September 2019.

[17] M.-H. Yang, J.-N. Luo, M. Vijayalakshmi, and S. M. Shalinie, "Hybrid multilayer network traceback to the real sources of attack devices," *IEEE Access*, vol. 8, pp. 201087–201097, 2020.

[18] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, "Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact," *Future Generation Computer Systems*, vol. 102, pp. 259–277, 2020.

[19] M. I. Mehar, C. L. Shier, A. Giambattista et al., "Understanding a revolutionary and flawed grand experiment in blockchain," *Journal of Cases on Information Technology*, vol. 21, no. 1, pp. 19–32, 2019.

[20] V. Dhillon, D. Metcalf, and M. Hooper, "The DAO hacked [M]," in *Blockchain Enabled Applications*, pp. 67–78, Apress, Berkeley, CA, USA, 2017.

[21] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov, "Smartcheck: static analysis of ethereum smart contracts[C]," in *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, pp. 9–16, ACM, Gothenburg Sweden, May 2018.

[22] Y. N. Aung and T. Tantidham, "Review of Ethereum: smart home case study," in *Proceedings of the 2017 2nd International Conference on Information Technology (INCIT)*, pp. 1–4, IEEE, Nakhonpathom, Thailand, November 2017.

[23] T. Chen, X. Li, Y. Wang et al., "An adaptive gas cost mechanism for ethereum to defend against under-priced dos attacks[C]," in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 3–24, Springer, Melbourne, Australia, 13 December 2017.

[24] P. Vessenes, "The ERC20 short address attack explained," 2017, https://vessenes.com/the-erc20-short-address-attack-explained/.

[25] W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: towards healthier blockchain technology[C]," in *Proceedings of the 2018 World Wide Web Conference*, pp. 1409–1418, Lyon, France, April 2018.

[26] A. Mense and M. Flatscher, "Security vulnerabilities in ethereum smart contracts[C]," in *Proceedings of the 20th International Conference on Information Integration and Web-Based Applications & Services*, pp. 375–380, Yogyakarta Indonesia, November 2018.

[27] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, *Making Smartcontracts Smarter*, inCCS, Padappai, Tamil Nadu, Indai, 2016.

[28] S. A. Pranesh, K. V. Vignesh, N. Viswanathan, and M. Vijayalakshmi, "Design and analysis of incentive mechanism for ethereum-based supply chain management systems," in *Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, Kharagpur, India, July 2020.

[29] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)[C]," in *Proceedings of the International conference on principles of security and trust*, pp. 164–186, Springer, Uppsala, Sweden, 24 April 2017.

[30] T. Chen, Y. Zhu, Z. Li et al., "Understanding ethereum via graph analysis," in *Proceedings of the IEEE INFOCOM 2018-IEEE conference on computer communications*, IEEE, Honolulu, HI, USA, April 2018.

[31] Bok, "Ethereum network attackers IP address is traceable," 2016, https://www.bokconsulting.com.au/blog/ethereum-network-attackers-ip-address-is-traceable/.

[32] E. F. Kfoury and D. J. Khoury, "Secure end-to-end volte based on ethereum blockchain[C]," in *Proceedings of the 2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pp. 1–5, IEEE, Athens, Greece, July 2018.

[33] Z. Li, J. Hou, H. Wang, C. Wang, C. Kang, and P. Fu, "Ethereum behavior analysis with NetFlow data[C]," in *Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–6, IEEE, Matsue, Japan, September 2019.

[34] P. Zheng, Z. Zheng, J. Wu, and H.-N. Dai, "XBlock-ETH: extracting and exploring blockchain data from ethereum," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 95–106, 2020.

[35] S. Bistarelli, G. Mazzante, M. Micheletti, L. Mostarda, D. Sestili, and F. Tiezzi, "Analysis of ethereum smart contracts and opcodes[C]," in *Proceedings of the International*

Conference on Advanced Information Networking and Applications, pp. 546–558, Springer, Matsue, Japan, March 2019.

[36] W. Chan and A. Olmsted, "Ethereum transaction graph analysis[C]," in *Proceedings of the 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 498–500, IEEE, Cambridge, UK, December 2017.

[37] E. Team, "Etherscan: the ethereum block explorer[J]," 2017, https://etherscan.io/?downloadpipe.com_id_1055536_question.

[38] Etherscan, "Ethereum developer APIs," 2018, https://etherscan.io/apis%23accounts.

[39] V. P. Ranganthan, R. Dantu, A. Paul, P. Mears, and K. Morozov, "A decentralized marketplace application on the ethereum blockchain[C]," in *Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pp. 90–97, IEEE, Philadelphia, PA, USA, October 2018.

[40] G. A. Oliva, A. E. Hassan, and Z. M. J. Jiang, "An exploratory study of smart contracts in the Ethereum blockchain platform [J]," *Empirical Software Engineering*, vol. 25, pp. 1–41, 2020.

[41] https://dapptotal.com/eth.

[42] C. Zhao and Y. Guan, "A Graph-based investigation of bitcoin transactions," in *Proceedings of the IFIP International Conference on Digital Forensics*, Springer, Orlando, FL, USA, January 2015.

[43] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Du, "Accurate decentralized application identification via encrypted traffic analysis using graph neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2367–2380, 2021.