Characterizing Ethereum Address Poisoning Attack

Shixuan Guan
sguan4105@sdsu.edu
San Diego State University
San Diego, CA, USA

Kai Li
kli5@sdsu.edu
San Diego State University
San Diego, CA, USA

Abstract

This paper presents the first comprehensive analysis of the address poisoning attack surged on the Ethereum blockchain. This phishing attack typically exploits the address shortening feature of Ethereum explorers and digital wallets (e.g., Etherscan and MetaMask) by crafting token transfer events with a seemingly correct address to poison victims' transfer history, waiting for them to mistakenly transfer assets to the attacker's address.

To systematically detect and characterize the address poisoning attack, we developed a detection system named *Poison-Hunter*, which can recognize the attacker's crafted transfers and detect the phishing addresses controlled by the attacker. By applying *Poison-Hunter* to Ethereum blocks produced from Nov. 2022 to Feb. 2024, we have detected millions of phishing transfers and phishing addresses. Our analysis shows that the attacker has predominantly targeted USDC and USDT token holders and used a phishing address that looks highly similar to a benign one. We also find that the sender of legitimate transfers was the primary target of this attack. Furthermore, by tracing the transaction history of the detected phishing addresses, we reveal that over 1,800 victim addresses have lost crypto assets, with a potential financial loss of up to $144 million US dollars. Among them, about $90 million of loss are confirmed by this work. Finally, our analysis suggests that 98% of phishing addresses are controlled by four entities, which collected nearly 92% of the total profits.

Overall, this paper sheds light on the tactics utilized in the address poisoning attack and its scale and impact on the Ethereum blockchain, emphasizing the urgent need for an effective detection and prevention mechanism against such a phishing activity.

CCS Concepts

• **Security and privacy → Distributed systems security**.

Keywords

Address Poisoning; Ethereum; Phishing Attack; Blockchain.

ACM Reference Format:
Shixuan Guan and Kai Li. 2024. Characterizing Ethereum Address Poisoning Attack. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), October 14–18, 2024, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3658644.3690277

CCS '24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-0636-3/24/10
https://doi.org/10.1145/3658644.3690277

1 Introduction

The flourishing of smart contract-based blockchains has led to an increasing adoption of various decentralized applications in domains such as finance, gaming, and health. As the largest smart contract-based blockchain, Ethereum enables the issuance and circulation of customized tokens, which has proliferated various ERC-20 tokens [4] and stable coins [23] in the cryptocurrency markets. However, the rise of cryptocurrency markets has also brought new threats targeting cryptocurrency users and causing financial loss to them. Among them, one emerging threat is the notorious cryptocurrency scams [37, 44, 47–50, 54, 57, 57, 59] and phishing attacks [33, 36, 41, 45] that playing tricks to deceive cryptocurrency holders and steal their funds.

In this work, we identified a new phishing attack named Ethereum address poisoning and conducted the first comprehensive analysis to dissect it and evaluate the impact. Based on an initial investigation, we found that the attacker exploited the address shortening feature of Ethereum explorer and digital wallets (e.g., Etherscan [13] and MetaMask [20]) by using a seemingly correct address to craft token transfer records in a victim's transfer history, which can potentially deceive victims and lead them to transfer assets to the attacker. To launch the attack at a large scale, the attacker generated a large number of Ethereum addresses and used them to craft three types of phishing transfers (dust-value, zero-value, and fake token transfers). Our initial investigation also shows that the attacker controlled two sets of addresses. The first set is used to initiate phishing transactions and pay the transaction fee (**funding address**). The second set is used to interact with a victim address and serve as the payload of the phishing transaction (**phishing address**). Given such a new phishing activity, in this paper, we aim to systematically study the address poisoning attack and answer the following research questions: **RQ1.** How prevalent is the address poisoning attack on the Ethereum blockchain, and how many users have been targeted by the attacker? **RQ2.** How many victims have been deceived by this attack, and how much financial loss has been caused? **RQ3.** What are the attacker's behaviors, and what strategies have been adopted to increase the attack's success rate?

Detection system: To answer the above research questions, we developed an attack detection system, *Poison-Hunter*, to detect phishing transfers crafted by the attacker and recognize its phishing address. Specifically, *Poison-Hunter* first collects a comprehensive dataset of ERC-20 token transfer events recorded on the Ethereum mainnet and then separates them into legitimate transfers and suspicious transfers based on the pattern of each type of phishing transfer. After that, *Poison-Hunter* matches the suspicious transfers with legitimate transfers by comparing the involved addresses through an address similarity scoring mechanism, which can accurately recognize the phishing addresses. To avoid flagging benign addresses that the attacker mistakenly or deliberately entered into

the phishing transfers, *Poison-Hunter* also employs a benign address sifting approach based on the activation timestamp of the address on Ethereum, which can filter out benign addresses and ensure that all of the flagged addresses are indeed controlled by the attacker.

Research findings: We have applied *Poison-Hunter* to analyze Ethereum blocks produced from Nov. 2022 to Feb. 2024. In summary, we collected over 14 million phishing transfers that target more than 1.44 million benign addresses, predominantly belonging to USDT and USDC token holders. From the phishing transfers, we identified over 6 million phishing addresses and 8,000 funding addresses controlled by the attacker. In addition, we also collected 2,300 fake tokens and 2,900 batching contracts deployed by the attacker to reduce the transaction cost (**answers to RQ1**). By leveraging the blockchain's transparency to trace transactions transferring assets to the phishing addresses, our work disclosed that over 1,800 victim addresses lost approximately $144 million US dollars (USD) to the attacker, of which $81.96 to $89.93 million USD were confirmed from the victims targeted in the phishing transfers. Compared to the $25.5 million USD paid by the attacker as the transaction fee for crafting the phishing transfers, the return on investment of this attack is above 220%. Furthermore, by clustering the phishing addresses based on their associated activities with other addresses, such as the funding address and batching contract, we identified four big clusters that controlled 98% of phishing addresses and gained over 90% of the total profits. Among them, the largest cluster controlled over 4.6 million addresses and profited over $60 million USD (**answers to RQ2**). Moreover, our results suggested that the attacker's phishing addresses bear a high degree of similarity with the benign addresses, most containing more than 7 similar hexadecimal characters in the beginning and ending segments. Besides, we also discovered that the attacker tends to target the sender of legitimate transfers, as proved by the fact that more than 90% phishing transfers target the "from" address in a legitimate transfer. Finally, our analysis showed that the attacker's money flow followed a common pattern, which involved mixing services, decentralized exchanges, and centralized exchanges (**answers to RQ3**).

Contributions: Our work makes the following contributions.

- **A comprehensive analysis:** To the best of our knowledge, our work is the first comprehensive study on the Ethereum address poisoning attack. We dissected the attacker's strategies in crafting phishing transfers, including dust-value transfers, zero-value transfers, and fake token transfers.
- **New attack detection system:** We developed an attack detection system named *Poison-Hunter* that leverages the unique pattern of each phishing transfer type to collect phishing transfers and detect the attacker's phishing address. The evaluation result on the ground-truth dataset shows that *Poison-Hunter* can achieve 100% precision and 97.3% recall.
- **New understandings:** *Poison-Hunter* has led to the discovery of millions of phishing transfers and addresses controlled by the attacker. Our work reveals that the attacker has targeted more than 1.44 million benign addresses and generated more than 6 million phishing addresses that contain 7 or more similar characters to the benign addresses. We also found that the attacker tends to target the sender of legitimate transfers.
- **Quantification of financial loss:** We quantified that more than 1,800 victim addresses have suffered a financial loss of up to $144 million USD to the attacker, with $90 million USD of the loss being confirmed from the targeted benign addresses. This compelling result calls for a more comprehensive mitigation against the address poisoning attack.

Road-map: The rest of this paper is organized as follows. Sec. 2 provides the necessary background of the address poisoning attack. Sec. 3 illustrates the strategies adopted in the address poisoning attack. Sec. 4 details how *Poison-Hunter* is designed and implemented to detect phishing transfers. Sec. 5 presents our analysis of the detected phishing transfers and addresses. Sec. 6 discusses the robustness of *Poison-Hunter* and possible countermeasures. Related work is discussed in Sec. 7, followed by a conclusion in Sec. 8.

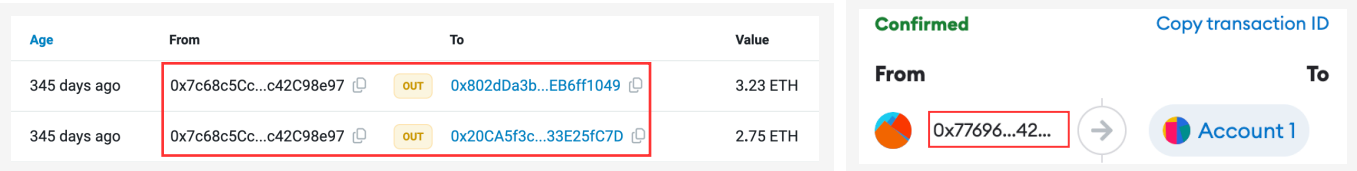
2 Background

2.1 Ethereum Blockchain and ERC-20 Tokens

As the largest smart contract-based blockchain platform, Ethereum allows users to develop programs with arbitrary logic (a.k.a, smart contracts) and execute them in a decentralized and trustworthy manner. With the smart contract feature, various customized tokens have been deployed on Ethereum. As of this writing, there are over 1,300 valuable tokens [10] deployed on Ethereum worth $20.4 billion USD[1]. In order to trade on the Ethereum blockchain, users can create an account with the Elliptic Curve Digital Signature Algorithm (ECDSA) [46], which features a pair of public key and secret key. The public key is shared with other users to help them identify the owner, while the secret key should be kept private by the owner, who then uses it to sign transactions. The last 20 bytes of the hash of the user's public key, a sequence of 40 hexadecimal characters, is also called the user's address on Ethereum.

ERC-20 token: On Ethereum, ERC-20 [4] is a token standard defined for Fungible tokens, which specifies the necessary functions (i.e., *transfer*, *transferFrom*) for owners to trade tokens, as well as events (i.e., *Transfer*) to log the token operations. Specifically, in each ERC-20 token contract, a mapping data structure named *balance* is used to track the token balance of every owner indexed by their Ethereum addresses. When the *transfer* or *transferFrom* function is executed successfully, a *Transfer* event will be emitted to record the token movement, including the sender's address, receiver's address, and the transferred amount. Any smart contract implementing the ERC-20 token standard is considered an ERC-20 token. In today's market, popular ERC-20 tokens include stable coins such as USDC [27], USDT [28], DAI [11], whose value is always pegged to $1 USD. It is worth noting that in these popular ERC20 tokens, as long as the transferred amount is zero, a successful *Transfer* event will be emitted even if the transaction initiator is not the token owner. In addition, when deploying an ERC-20 token contract on Ethereum, the contract deployer is allowed to give the token an arbitrage name and symbol, as the Ethereum blockchain does not impose restrictions on the token name and symbol. As a result, two token contracts could have the same name and symbol, which renders fake ERC-20 tokens possible on Ethereum, as shown in the recent work [44].

[1] https://coincodex.com/cryptocurrencies/sector/ethereum-erc20/



(a) Etherscan (b) Metamask

Figure 1: The address shortening feature employed by Etherscan and Metamask.

Ethereum transaction batching: Unlike Bitcoin's Unspent Transaction Output (UTXO) model, Ethereum adopts an account-based transaction model where each transaction only binds one sender address with one receiver address. Hence, Ethereum natively does not support transfering ETH to multiple receivers or invoking multiple smart contracts in a single transaction. However, users can work around this limitation with a batching contract. For example, to transfer ETH in batch, users can utilize a batching contract that takes multiple recipient addresses as the input and then distributes the received ETH to individual recipients through internal transfer calls. Likewise, users can also utilize batching contracts to invoke multiple smart contracts through internal function calls. According to the recent work [55, 56], batching ETH transfers or smart contract invocations can save users' transaction costs.

2.2 Ethereum Address Shortening

Due to the large computation and storage cost, it is unaffordable for an average user to run a full node to manage their accounts and access the Ethereum blockchain. To solve the problem, third-party Web3 services have emerged as a gateway to connect users to the Ethereum blockchain, including digital wallets, RPC services [5, 9, 19, 22], and Ethereum explorers [7, 13, 21]. With a digital wallet like MetaMask [20], users can create an account and send transactions to the Ethereum blockchain. Once a transaction is executed successfully, it will be recorded on both the sender and receiver's Metamask App. Besides, users can also visit Etherscan [13], the most popular Ethereum explorer, to check their balance and browse the transaction history. Due to the long sequence of Ethereum addresses, when displaying transactions, most Web3 services would shorten the user's address, as presented in Fig. 1. For instance, Fig. 1a shows a user's two transactions displayed on Etherscan. As highlighted in the red box, Etherscan shortens both the sender and receiver's address by only showing the first 8 characters and the last 9 characters. Similarly, Metamask also shortens the user's address by only presenting the first 5 characters and 2 characters in the middle, as highlighted in Fig. 1b.

3 Threat: Address Poisoning Attack

Due to the address shortening feature employed by Web3 services such as Etherscan and Metamask, users can only use the prefix or suffix to differentiate Ethereum addresses. In this work, we discovered that attackers had exploited such an address shortening feature to launch the so-called address poisoning attack. By investigating online reports from multiple sources, including Twitter [1], MetaMask [32], and Etherscan [31], overall, we found that the attacker typically generates a phishing address to impersonate a benign address and uses it to interact with an address who has previously interacted with the benign address, aiming to craft a similar transfer in the address' transaction history to deceive the address's owner. As a result, the owner may copy the attacker's address and make subsequent transfers to it. Below, we discuss how the attack is launched in practice.

3.1 Attack Preparation

For the address poisoning attack, the key to success is launching it on a large scale and targeting many benign addresses so that the attacker has a higher chance to successfully deceive one of them. To do so, the attacker typically generates a large number of Ethereum addresses before using them to interact with a benign address, which we denoted as phishing addresses. Since these generated phishing addresses do not have enough balance to pay the transaction fee, we found that the attacker also controls a set of Ethereum addresses that will be used to initiate the phishing transaction and pay the transaction fee, which we denoted as funding addresses. After generating the phishing addresses, the attacker then actively monitors transactions recorded on the Ethereum blockchain to find a benign address similar to one of its phishing addresses. Upon finding a similar benign address, the attacker then uses the funding address and phishing address to send a transaction to craft a phishing transfer in a victim's transaction history.

3.2 Attack Strategy

Based on the analysis of the online reports, we found that attackers adopted three attack strategies to craft phishing transfers in a victim's transaction history, as presented in Fig. 2. These strategies generally follow a similar pattern and only differ at step ❷. Below, we elaborate on each strategy in detail.

Dust-value token transfer: After finding a legitimate token transfer between two benign addresses (❶), say Alice (0x1234) sends 10 USDC to Bob (0x0D8C), the attacker uses its funding address (0x5678) to transfer a small amount of USDC to the phishing address (0x0DBC), which then immediately transfers it to Alice (❷a). The phishing address looks highly similar to Bob's address. Both the legitimate transfer and the dust-value transfer will be captured by Etherscan or a digital wallet and added to Alice's transfer history. When Alice decides to make another transfer to Bob and visits Etherscan to browse the token transfer history to find Bob's address (❸), because the attacker's phishing address looks similar to Bob's, she could copy the phishing address and transfer tokens to it (❹), resulting in a financial loss.

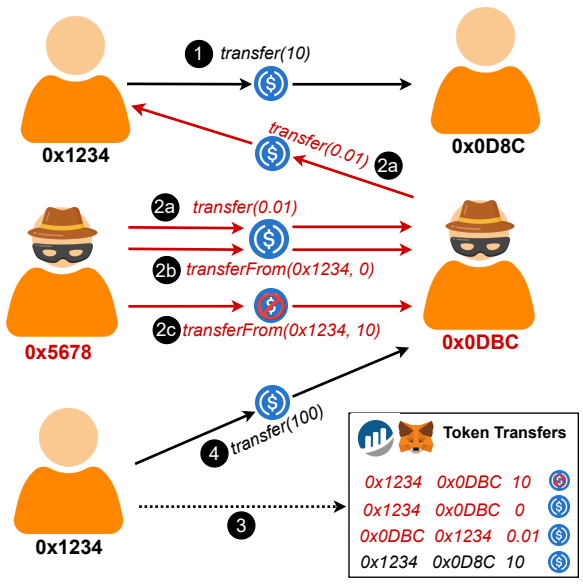


Figure 2: The workflow of the address poisoning attack. For a legitimate transfer from a benign address (1), the attacker crafts phishing transfers with a similar recipient address (2a, 2b, 2c). The crafted transfers are added to the benign address's transfer history (3). The benign address's owner then copies the recipient address from the phishing transfers and transfers funds to the attacker (4).

Zero-value token transfer: In this strategy, the attacker calls the `transferFrom` function in the legitimate token contract to transfer zero-value tokens. That is, in step 2b, the attacker uses the funding address to call the token contract as follows, `transferFrom(Alice, Eve, 0)`, where `Eve` is the phishing address. The sanity checks in the token contract (`allowance and balance`) would allow the transfer to pass since the transferred amount is 0. After that, a `Transfer` event is emitted to log the zero-value transfer and added to Alice's token transfer history. When Alice browses its transfer history (3), she could copy the phishing address and transfer tokens to it (4), resulting in a financial loss[2].

Fake token transfer: In the above strategy, the attacker cannot manipulate the transferred amount, which must be 0 and could be easily detected. To overcome such a drawback, the third strategy is to deploy a fake ERC-20 token that can emit `Transfer` events of arbitrary transferred amounts. That is, the attacker removes the sanity check logic and then deploys the fake token contract. Then, in step 2c, the attacker uses the funding address to call the fake token contract as `transferFrom(Alice, Eve, 1,000)`, which then emits a `Transfer` event that looks the same as the legitimate transfer. Such a fake transfer event will be added to Alice's transfer history and displayed to Alice (3). In this strategy, the attacker can give different symbols to the fake tokens, which generally can be summarized in two categories. The first category uses the symbol of popular ERC-20 tokens such as USDC and USDT, and the second category utilizes a self-defined symbol such as "ETH."

In summary, each attack strategy can craft a phishing transfer record in the victim's transfer history. Compared to the first strategy, the last two strategies are cheaper, as the cost only includes the transaction fee. In this study, we found that attackers also leveraged batching contracts to save the transaction fee. With a batching contract, the attacker can include multiple phishing transfer payloads in one transaction, which can dispatches the payloads to emit multiple transfer events simultaneously. The batching contract allows the attacker to save the basic transaction fee (21,000 Gas), thus further reducing the attack cost.

4 Detection System: *Poison-Hunter*

To collect the phishing transfers crafted by the attacker and detect the involved phishing addresses, we developed an attack detection system named *Poison-Hunter*. Our detection system consists of three modules: *Data Collector, Token Analyzer, Address Filter*. An overview of the detection workflow is presented in Fig. 3.

4.1 Data Collector

Our first module, *Data Collector*, aims to extract ERC-20 token contracts and their associated token transfers. To accomplish this task, we set up an Ethereum full node and synchronized it with the Ethereum mainnet. After that, we leverage the Ethereum-ETL [16] tool to extract the deployed ERC-20 token contracts and their associated token transfer events. Specifically, for each ERC-20 token contract, we collect the token address, name, symbol, and the token deployer's address. Thereafter, we collect the token transfer events emitted by each ERC-20 token contract, including the sender, receiver, transferred amount, and the token address.

4.2 Token Analyzer

After collecting ERC-20 token contracts and token transfer events, we run the second module, *Token Analyzer*, to separate highly suspicious transfers from legitimate token transfers. As described before, the attacker can craft three types of phishing transfers: dust-value transfer, zero-value transfer, and fake transfer. Hence, there are three types of **suspicious transfers**. Below, we elaborate on how we separate them from legitimate transfers.

Dust-value transfer: We leverage the ground-truth[3] of popular ERC-20 token addresses [15] to collect the transfer events emitted by each legitimate token. After that, we filter the transfer events by the transferred amount. That is, if the transferred amount is a dust value (e.g., 0<value<1), we deem it suspicious and save it to the suspicious transfer dataset. Since a dust-value transfer could also be sent by a benign user who just wants to verify a receiver or forgets to set the correct amount, we will describe how we further filter the suspicious transfer dataset to detect phishing dust-value transfers in the next section.

Zero-value transfer: Similar to the above, if the transferred amount is 0, we deem the token transfer event a suspicious transfer. Since some zero-value transfers could also be sent by benign users,

[2]Likewise, the attacker can poison Bob's token transfer history by using an address similar to Alice's.

[3]It is well-known that the legitimate USDC and USDT token address is `0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48` and `0xdAC17F958D2ee523a2206206994597C13D831ec7`.

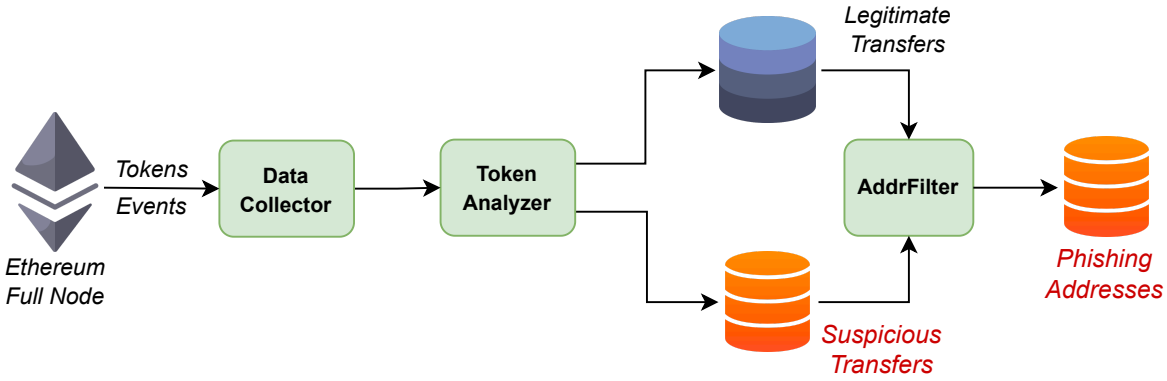


Figure 3: The detection workflow of *Poison-Hunter*.

we filter them as follows. For each zero-value transfer, we verify whether the transaction sender is the sender of the token. If not, we deem it highly suspicious and save it to the suspicious transfer dataset. This is because, unlike a phishing dust-value transfer where the victim is the receiver of the token, the phishing zero-value transfer often places the victim as the sender of the transfer while using a funding address as the transaction's sender. Hence, if a zero-value transfer transaction's sender is not the sender of the transfer, we consider it a highly suspicious transfer. After that, we further filter the suspicious transfer dataset to identify phishing zero-value transfers, as will be described in the next section.

Fake token transfer: To seperate this type of transfer, we need to first identify the fake token deployed by the attacker. There are two types of fake token contracts: a fake token using the same symbol as a popular token and a fake token using a self-defined symbol. To detect the first type of fake token, we collect the symbols of top ERC-20 tokens by the market cap to discover fake tokens. Specifically, we collect the contract address and symbol of the top 50 ERC-20 tokens listed on Etherscan [15]. After that, we search in the collected token contracts for fake tokens that have the same symbol but different addresses with the legitimate ERC-20 tokens. Then, we save all the transfer events emitted by the fake tokens to the suspicious transfer dataset. To identify the second type of fake tokens, we utilized a token validation method. In this method, we use a locally generated address to send a testing transaction to execute the *transfer* and *transferFrom* functions in the collected token contracts. If the testing transaction succeeds and a *Transfer* event is emitted, we deem the token contract a suspicious token and save all the emitted transfer events into the suspicious transfer dataset. This is because should the token contract employ the correct logic to check the allowance and balance, our testing transaction would fail, and no *Transfer* event would be emitted, as our locally generated address shall not have a balance in the token. Hence, any token violating the logic could be used to craft token transfer events, which will be detected by our token validation technique.

Finally, by applying the *Token Analyzer* module, we obtain two token transfer datasets: a legitimate transfer dataset where all the transferred amounts are larger than or equal to 1, and a suspicious transfer dataset that includes dust-value transfers, zero-value transfers, and fake token transfers.

4.3 Address Filter

After seperating suspicious and legitimate token transfers, we apply the *Address Filter* module to filter the suspicious transfer dataset and detect the attacker's phishing address. Below, we describe how we achieve the goal in three steps.

4.3.1 Matching token transfers by identical address. In this step, we match the suspicious transfers with legitimate transfers to filter some non-phishing transfers from the suspicious transfer dataset. We know that in the address poisoning attack, a phishing transfer particularly targets a previously included legitimate transfer and uses a phishing address to interact with one of the addresses in the legitimate transfer. If we can find a previous legitimate transfer that shares an identical address (**requirement A**) and a highly similar address (**requirement B**) with the suspicious transfer, then the suspicious transfer is likely a phishing transfer. In light of these requirements, we first use requirement A to match the suspicious transfers with legitimate transfers. Specifically, for each suspicious transfer recorded at block T, we search in the legitimate transfers for one transfer recorded before block T that has the same "from" (or "to") address with the suspicious transfer. Additionally, if the transfer is a fake token transfer, we also require that they have the same transferred amount. In this matching phase, one suspicious transfer could match with multiple legitimate transfers. If so, we retain all matched legitimate transfers and filter them in the next step. In the case that a suspicious transfer does not match with any legitimate token transfer, we remove it from the suspicious transfer dataset as it is not related to the address poisoning attack.

4.3.2 Filtering by address similarity. After matching suspicious transfers with legitimate transfers that share an identical address, we need to verify further if they meet requirement B, which is sharing a highly similar address. To accomplish the goal, we propose an address similarity scoring mechanism to determine if two addresses are highly similar. Specifically, our address similarity scoring mechanism works as follows. We use a score to represent the similarity of two addresses and continuously update the score by comparing the hexadecimal characters of two addresses bidirectionally. At the beginning, we compare the first and last characters of two addresses and see if they are identical or similar. If so, we increase the score by 2 and move to the next position (the second

and the last second character) to continue the comparison process. If only in one direction two characters are identical or similar, we then increase the score by 1 and move to the next position in that direction. The comparison process terminates until no characters in both directions are identical or similar. When comparing hexadecimal characters, we consider that letters A-F are case insensitive (A=a, B=b, etc.). In addition, we also consider the following three number-to-letter pairs to be similar due to their visual similarity: (0, "D"), (6, "b"), and (8, "B"). By applying our address similarity scoring mechanism on a matched suspicious transfer and legitimate transfer, we can assess whether they share a similar address. In this work, we require that the first 2 characters and the last 2 characters must be the same or similar in the matched transfers (threshold = 4). If their similarity score is above the threshold, we consider the suspicious transfer as a phishing transfer.

When a phishing transfer is matched with multiple legitimate transfers after filtering by the similarity score, we further calculate their block distance and retain the closest one. The reason for keeping the closest legitimate transfer is that in this attack, the attacker needs to include their phishing transfers right after the legitimate transfer so that they can be adjacent in the victim's transfer history, leading to a high chance of deceiving the victim.

4.3.3 Sifting benign addresses. After the above two steps, we end up collecting multiple pairs of suspicious and legitimate transfers that share an identical address and a similar address. Now, we can locate the victim's address (the identical address) and the phishing address (the similar address in the suspicious transfer). However, treating all the similar addresses in the suspicious transfers as phishing can cause a new problem. That is, we could mistakenly label a benign address that happens to meet all the requirements as phishing. Indeed, in our initial results, we found that some benign addresses (e.g., a popular address) were entered into the suspicious transfers, either due to a coincidence or a mistake made by the attacker, or even due to a countermeasure adopted by the attacker for confusing a detection system. To solve this problem, we propose to sift the similar addresses in the suspicious transfers as follows: (1) if the similar address has not sent or received any transactions or legitimate tokens at the time the suspicious transfer is emitted, we deem it a phishing address; (2) for the other addresses not meeting the first criteria (e.g., has sent or received transactions or legitimate tokens), if all of their transactions were interacting with one or another similar address in the suspicious transfers, we treat them as phishing addresses. The first condition guarantees that the similar address never appears in the blockchain at the time it was entered into the suspicious transfer, giving us high confidence that they must be locally generated by the attacker. The second condition ensures that attackers cannot evade our detection even if they use multiple addresses to aggregate or distribute funds for payment mixing or money laundering. With the two filtering conditions, we can ensure that benign addresses are removed and that all the addresses left are indeed controlled by the attacker.

4.4 Evaluation of *Poison-Hunter*

To evaluate the performance of *Poison-Hunter* in detecting phishing addresses involved in the address poisoning attack, we collected Ethereum addresses from reliable sources to build a ground-truth dataset and measure the precision and recall metrics.

Table 1: Evaluation of *Poison-Hunter* on the ground-truth dataset.

	Ground-truth		*Poison-Hunter*					
# Address	Phishing (P)	Benign (N)	TP	TN	FP	FN	Precision	Recall
	5,890	1,154	5,729	1,154	0	161	100%	97.3%

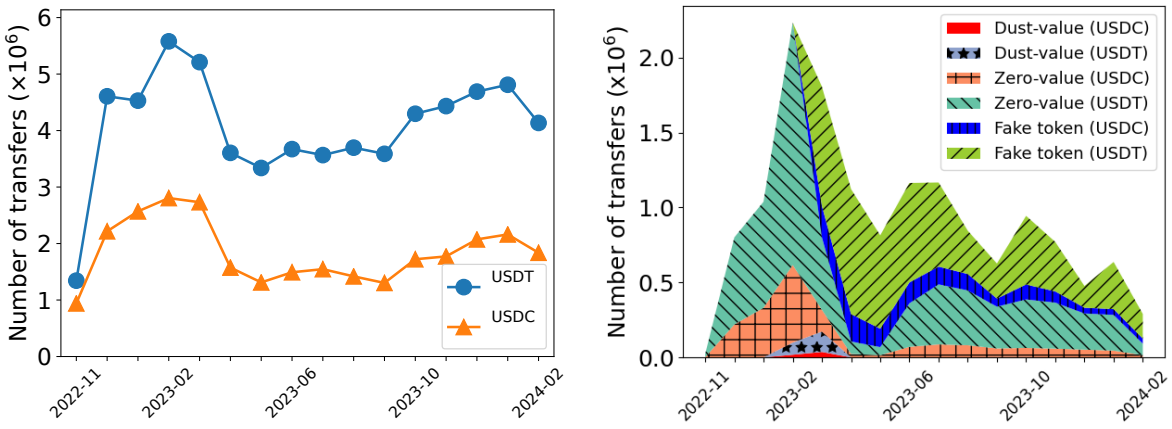
Benign addresses: Due to the blockchain's anonymity, it is challenging to attribute an address to an individual or entity. Hence, no ground-truth benign address dataset is available. However, thanks to Etherscan's public label and name tag features, we can build our own benign address dataset by downloading addresses with benign public labels and name tags from Etherscan. The public labels and name tags are added by Etherscan based on the publicly disclosed information[4]. For this task, we leveraged Etherscan's *Label Word Cloud* API [14] to obtain labels and name tags related to decentralized finance (DeFi) and then chose the top 15 labels that own the most addresses, including Aave, Bancor, Coinbase, Compound, etc. We then selected the top 100 addresses within each label by the number of transactions that transfer ERC-20 tokens. In total, we obtained 1,154 benign addresses.

Phishing addresses: To collect phishing addresses, we combined the reports from Etherscan and Forta [17], which are the only two services that publish addresses involved in the address poisoning attack. Specifically, Etherscan has assigned the phishing address a label of "Fake_Phishing" and published a note of "the address may be attempting to impersonate a similar looking address" to warn users. Similarly, Forta also published a phishing address dataset involved in the address poisoning attack based on intelligence from multiple detection bots. Indeed, Forta used a combined logic to determine whether a reported address should be labeled as phishing, including the detection bot's trustworthiness, the manual analysis result from the Forta community, the contracts deployed by the address, and the address's association with other phishing addresses, etc. For this task, we combined the phishing addresses reported by Etherscan and Forta to build our phishing address dataset. In total, we collected 5,890 phishing addresses from them.

Evaluation results: We ran *Poison-Hunter* to detect phishing addresses from our collected transfer events and compared the results with the ground-truth dataset. The evaluation result is shown in Table 1. We can see that our *Poison-Hunter* did not label any benign addresses as phishing, achieving a precision of 100%. For the 5,890 phishing addresses, *Poison-Hunter* detected 5,729 of them, leading to a recall of 97.3%. We randomly checked certain phishing addresses missed by *Poison-Hunter* and found that they were targeting less popular ERC-20 tokens such as TrueUSD [29] and BUSD [8], which *Poison-Hunter* has not considered. Nevertheless, the evaluation result on the ground-truth dataset still shows that *Poison-Hunter* has achieved a good performance in detecting phishing addresses involved in the address poisoning attack.

Coverage of *Poison-Hunter*: We also show the performance of *Poison-Hunter* in uncovering new phishing addresses that Etherscan and Forta have missed. Specifically, we selected the 450K

[4]Etherscan assigns a corresponding label to an address that is claimed to be owned by an entity or organization.



(a) Legitimate transfers.

(b) Phishing transfers.

Figure 4: Timeline of legitimate and phishing transfers in the lifespan of the address poisoning attack.

Table 2: Comparison of *Poison-Hunter* with existing services.

	Poison-Hunter	Covered by Existing Service	
		Forta (%)	Etherscan (%)
# Address	450,844	33,685 / 7.5%	7 / 0.002%

phishing addresses *Poison-Hunter* detected from transfer events emitted in June 2023 and then queried their status on Etherscan and Forta. As shown in Table 2, Forta only flagged 7.5% (33,685) of our reported addresses, while Etherscan only flagged 7 of them. Such a result suggests that over 92.5% of phishing addresses were uniquely detected by *Poison-Hunter*, indicating the strong potential of *Poison-Hunter* in uncovering new phishing addresses involved in the address poisoning attack.

5 Detection Results

We applied *Poison-Hunter* to analyze Ethereum blocks produced from Aug. 2022 to Feb. 2024 (block number 14, 880, 000 to 19, 350, 000). In total, we collected hundreds of millions of legitimate transfers, tens of millions of phishing transfers, and millions of phishing addresses. Our analysis suggests that the earliest phishing transfer dated back to Nov. 2022, and USDC and USDT tokens were the predominant target of this attack due to their popularity, which accounts for more than 95% of the phishing transfers. We thereby focus on analyzing the phishing transfers targeting the USDC and USDT tokens. Below, we illustrate the characteristics of our collected phishing transfers and phishing addresses.

5.1 Characteristics of phishing transfers

We first show an overview of the collected phishing transfers and the fake tokens in Table 3. Specifically, from the transfer events emitted by legitimate USDC/USDT token contracts, we collected 42.3/82 million legitimate transfers transfers, 55K/222K phishing dust-value transfers, as well as 1.78/6.07 million phishing zero-value transfers. Then by combining the token symbol searching and token validation approaches, we respectively detected 1,130 fake USDC tokens and 1,203 fake USDT tokens, and from which 1.22/5.43 million fake token transfers were identified.

Table 3: Overview of collected token transfers.

	# Legitimate Transfers	# Phishing Transfers			# Fake Tokens
		Dust-value	Zero-value	Fake token	
USDC	42.3M	55.4K	1.78M	1.22M	1,130
USDT	82.0M	221.9K	6.07M	5.42M	1,203
Total	124.3M	277.3K	7.85M	6.64M	2,333

Based on the results in Table 3, we can obtain several interesting findings. First, the trading frequency of legitimate USDT is twice of USDC, indicating that cryptocurrency holders have a higher interest in trading USDT than trading USDC. Second, compared to USDC, USDT has led to 3X to 4X more phishing transfers in dust-value, zero-value, and fake token transfers due to having more legitimate transfers. However, the fake tokens of USDC and USDT seem to have a similar scale, both with more than 1,100 fake ERC-20 tokens deployed on Ethereum. Third, the number of zero-value transfers and fake token transfers is much larger than the dust-value transfers, indicating that the attackers prefer to craft zero-value transfers and fake token transfers. This can be explained as transferring dust-value incurs a higher cost, rendering the attacker select the cheaper options by transferring zero-value and fake tokens.

Timeline of phishing transfers: Fig. 4 presents the timeline trend of legitimate and phishing transfers in different months from Nov. 2022 to Feb. 2024. We first show the monthly volume of legitimate transfers in Fig. 4a. It can be seen that the number of legitimate transfers in both USDT and USDC exhibited a similar trend, with the peak occurring in Feb. 2023, followed by a sharp decline towards May 2023. After that, legitimate transfers experienced a steady growth towards Feb. 2024. In addition, in Fig. 4b, which shows the monthly volume of phishing transfers, we also have several interesting observations. First, the figure shows that the lifespan of dust-value transfers in both USDT and USDC was

relatively short, which only spanned Dec. 2022 to Apr. 2023 and reached a peak in Mar. 2023. After that, dust-value transfers were no longer of the attacker's interest. Second, we can see that the trend of zero-value transfers in USDC and USDT seems to follow the trend of the legitimate transfers, with the peak also occurring in Feb. 2023, followed by a sharp decline towards Apr. 2023. Third, for fake token transfers, the earliest transfer in both USDC and USDT was in Mar. 2023, which was four months later than the earliest zero-value transfer. Since then, the number of zero-value transfers has decreased. Such an observation suggests that in Mar. 2023, the attacker began to prioritize fake token transfers over zero-value transfers. In addition, the two figures indicate that the total number of all three phishing transfers in each month was roughly 1/4 of the legitimate transfers, implying that a legitimate transfer could have a 25% chance of being targeted by the attacker.

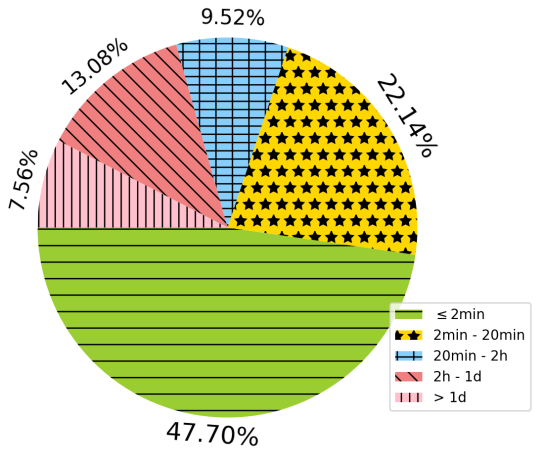


Figure 5: The distribution of the block delay between the matched legitimate transfers and phishing transfers.

Timeliness of phishing transfers: Since a phishing transfer particularly targets a legitimate transfer included in the previous blocks, it will be interesting to understand how promptly the attacker was able to craft a phishing transfer upon finding a suitable legitimate transfer. To obtain the result, we calculate the block distance between each matched legitimate transfer and phishing transfer and summarize the distribution in Fig. 5. We can see that the attacker was able to craft phishing transfers within 2 minutes (≈10 blocks) for more than 47% of legitimate transfers, and 22% of phishing transfers were crafted in a delay between 2 and 20 minutes. In contrast, only 7.56% of phishing transfers were crafted with a delay of more than one day (≈7,200 blocks). Such an observation indicates that the attacker was able to craft phishing transfers for most legitimate transfers in a small amount of time. For the cases where the block delays were over one day, we suspect that it is because when the attackers launched the attack, they also searched legitimate transfers included in historical blocks instead of monitoring transfers in the current block, in the hope of targeting more legitimate transfers and increasing the attack's success rate.

Batched phishing transfers: We also analyzed the transactions emitting the phishing transfer to see if a batching contract was used. Table 4 summarizes the analysis result. We can see that over 14.72 million phishing transfers are created through batching contracts, which accounts for 99.7% of the phishing transfers. Besides, the attacker has deployed nearly 3,000 batching contracts and utilized more than 3,700 funding addresses to send a total of 1.5 million transactions to batch the phishing transfers, which costs the attacker over 7,800 ETH ($25.4 million USD) of transaction fee. In comparison, only a tiny portion of the phishing transfers (0.3%) are directly sent through a single transaction with 4,800 funding addresses, which costs 39.5 ETH (127.5K USD). Based on the result, we can see that without a batching contract, the attacker has to spend an average of 2.55 USD to craft a phishing transfer. In contrast, using a batching contract reduces the average transaction fee to 1.73 USD, saving the attacker's cost by 32%. Such an advantage has motivated the attacker to predominantly utilize batching contracts. In addition, the total number of distinct funding addresses in both batched and non-batched transfers is 8,422, implying that (3,746+4,823) - 8,422 = 147 addresses have been utilized to send both non-batched and batched phishing transfers.

Table 4: Overview of batched and non-batched phishing transfers.

	# Phishing Transfers	# Txs	# Funding Addrs	# Batching Contracts	Tx Fee (ETH/USD)
Batched	14.72M	1.50M	3,746	2,914	7,879.6 / 25.4M
Non-batched	50.0K	50.0K	4,823	N/A	39.5 / 127.5K
Total	14.77M	1.55M	8,422	2,914	7,918.2 / 25.5M

5.2 Characteristics of phishing addresses

As described in Sec. 4, we can locate both the targeted benign address and the phishing address from the matched legitimate and phishing transfers. In total, we have extracted 6.09 million phishing addresses and 1.44 million benign addresses targeted by the attacker, as summarized in Table 5.

Overview: From Table 5, we can obtain several interesting insights. First, the benign and phishing addresses contributed by USDT are ≈3X of USDC, which can be attributed to its larger number of phishing transfers. In addition, in both USDC and USDT, the number of phishing addresses is ≈4X of the number of benign addresses, indicating that one benign address could be targeted by four phishing addresses on average. Besides, by combining the benign addresses and phishing addresses, we found that 0.12 million benign addresses have been targeted by both USDC and USDT phishing transfers, and 0.13 million phishing addresses have been utilized to craft both USDC and USDT phishing transfers, indicating that some attackers were actually monitoring multiple tokens' legitimate transfers. Moreover, the table also presents the distribution of the targeted benign addresses based on their role in the legitimate transfer. The result shows that in 91% - 93% of USDC and USDT legitimate transfers, the benign address served as the sender of the transfer, indicating that compared to the receiver, the sender of legitimate token transfers has a much higher probability of being targeted by the attacker. This can be explained by the attacker's belief that the sender is more likely to make a subsequent transfer to the same recipient after sending the first transfer. Indeed, it is a common practice in the real world for cryptocurrency users to

make a small transfer to a recipient before sending a large transfer in order to validate the recipient. Therefore, attacking the sender can increase the possibility that the sender may make a mistake and transfer assets to the attacker, hence improving the attack's success rate.

Table 5: Overview of phishing addresses and the targeted benign addresses.

	# Benign	# Phishing	Role of Benign Address	
			Sender (%)	Receiver (%)
USDC	0.45M	1.56M	92.7%	6.3%
USDT	1.11M	4.66M	91.4%	8.6%
Total	1.44M	6.09M	NA	NA

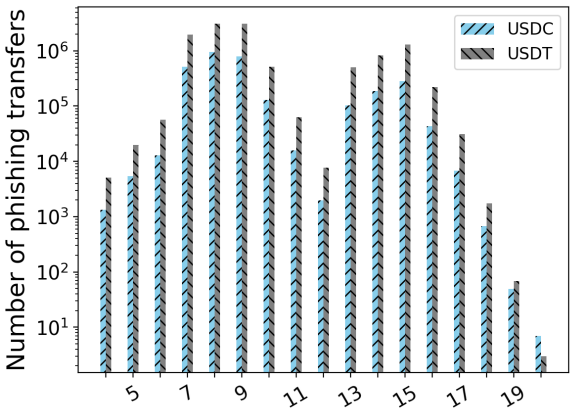


Figure 6: The distribution of address similarity score between the phishing address and benign address.

Address similarity: We further analyzed the similarity between the phishing and benign addresses to understand the attacker's ability to generate highly similar phishing addresses. We applied our address similarity scoring mechanism to all the matched phishing and legitimate transfers. The similarity score distribution is shown in Fig. 6. In both USDC and USDT phishing transfers, the similarity scores of their phishing addresses bear a similar distribution and vary between 4 and 20. In addition, their similarity scores do not follow the normal distribution but center around two separate ranges, one from 7 to 10 and the other from 13 to 16. We analyzed phishing addresses in such two ranges and found a clear separation in the timeline. Before Mar. 2023, all the similarity scores were smaller than 14. After that, the similarity scores in the second range (13 - 16) started accumulating. Such a separation in the timeline may be explained by the change in Etherscan's GUI. In early 2023, Etherscan only displayed 14 hex characters (first 6 and last 8). Then Etherscan expanded the address field to display 17 hex characters (first 8 and last 9). We suspect such a change may have caused attackers to generate phishing addresses with more similar characters to benign ones. In our results, the total number of phishing transfers falling into such two ranges is 10.5 million and 3.3 million, which respectively account for 71.1% and 22.3% of the phishing transfers, implying that the attacker can generate highly similar addresses to impersonate a benign address, which could make them difficult to be distinguished by users.

5.3 Victim Transactions and Financial Loss

In this section, we discuss our analysis of the victim transactions and the profits gained by the attackers.

5.3.1 Identification of victim transactions. In general, there are three types of victim transactions. The first type is a victim sends a basic transaction that transfers Ether to the attacker. The second type is a victim using a smart contract to internally transfer Ether. The third type is a victim transfers legitimate ERC-20 tokens to the attacker. To identify each type of victim transaction, we trace the transaction history of the detected phishing addresses to search for both external and internal transactions that transfer Ether to them, as well as transactions that transfer legitimate ERC-20 tokens to them. After that, we group the collected transactions by the phishing addresses. To remove false positives, we filter the collected transactions with the following conditions: (1) we remove a transaction if it is sent by one of the phishing addresses or funding addresses; (2) we remove a transaction if it is included earlier than the earliest phishing transfer crafted from the phishing address. After applying two conditions, we further verify the left transactions and assign them to one of the following two categories if the transaction sender meets the associated condition:

- **Confirmed victim transaction:** The transaction sender was targeted in one of the phishing transfers.
- **Potential victim transaction:** The transaction sender was not targeted in the phishing transfers and only transferred funds to the phishing addresses less than three times.

The reason we assign a transaction as a potential victim transaction even though the sender was not targeted in the phishing transfer is that a victim may own multiple addresses and use a different address not targeted in the phishing transfers to transfer funds to the attacker. Moreover, we also assume that victims of this attack cannot be deceived more than twice.

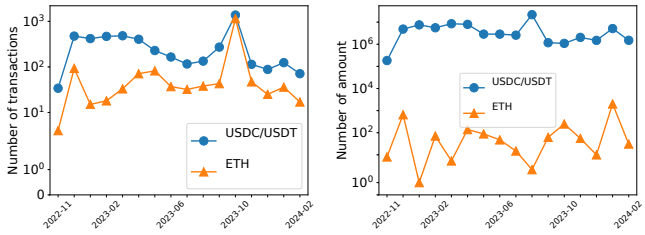
5.3.2 Overview of victim transactions. After tracing the transaction history of each phishing address, we obtained two groups of victim transactions, one group representing the confirmed victim transactions and the other group representing the potential victim transactions. Below, we focus on discussing the analysis of the confirmed victim transactions.

Confirmed victim transactions: Table 6 shows the statistics of the confirmed victim transactions and the profits gained by the phishing addresses. From the table, we can see that the majority of victim transactions are transferring stablecoins (USDT/USDC) to the attacker. There are more than 1,700 victim addresses that have sent over 76 million stablecoins to 2,900 phishing addresses in 4,900 transactions, resulting in a financial loss of up to $76.79 million USD. Among the 4,900 victim transactions, the minimal lost amount is 5, and the maximum is 20 million (the Binance case reported in [1]). The average and median lost amount is 15,500 and 500. In addition to transferring stablecoins, we also found that over 200

Table 6: Overview of confirmed victim transactions and the associated financial loss.

Crypto Asset	#Phishing Addr.	#Victim Addr.	#Victim Txs	# Lost Amount Total	# Lost Amount Min / Max / Mean / Median	Total USD Value (Min~Max[5])
USDT/USDC	2,925	1,715	4,963	76.79M	5 / 20M / 15.5K / 500	76.79M
ETH	1,485	205	1,680	3,231	0.0001 / 2,000 / 0.005 / 1.97	5.17M ~ 13.14M
Total	3,149	1,858	6,643	NA		81.96M ~ 89.93M

victim addresses have transferred more than 3,200 ETH to 1,400 phishing addresses through 1,600 transactions, leading to a financial loss varying from $5.1 million to $14 million USD. The maximum lost amount is 2,000 ETH. Based on the result, it can be seen that the attacker's primary profits are stablecoins, which is reasonable as the attacker only crafts token transfers to deceive users, who are more likely to make subsequent token transfers rather than making ETH transfers. By combining two categories of victim transactions, we obtained over 1,800 unique victim addresses and 3,100 unique phishing addresses. Such a result indicates that the attack's success rate is around (1,858/1.44M)≈0.1%, and (1,715+205-1,858) = 62 victim addresses (3.3%) have sent both stablecoins and ETH to the attacker, and (2,925+1,485-3,149) = 1,261 phishing addresses (40%) have profited in both stablecoins and ETH. Overall, the total lost amount in two categories of victim transactions is worth $81.96 million to $89.93 million USD. Compared to the attacker's investment of $25.5 million USD in the transaction fee, the attacker's return on investment (ROI) is above 220%.



(a) Monthly victim transactions. (b) Monthly financial loss.

Figure 7: Timeline of victims' transactions and financial loss in the lifespan of the attack.

Timeline of victim transactions and financial loss: We show the monthly victim transactions and lost amount in Fig. 7. From Fig. 7a, we can see that the number of victim transactions in two types of crypto assets almost stays stable in the entire lifespan of the attack, except for a spike in Oct. 2023 totaling more than 1,000 victim transactions. However, as shown in Fig. 7b, the victims' monthly financial loss in two types of crypto assets does not follow a similar trend. For USDC and USDT, the highest loss occurred in Aug. 2023, which was caused by the Binance case. For ETH, the highest loss happened in Jan. 2024, which was caused by the 2,000 ETH loss case. Besides, for each type of crypto asset, its monthly lost amount does not follow the trend of monthly victim transactions. This is because the lost amount in victim transactions varied significantly, leading to a diversified total lost amount in different months.

Table 7: Overview of potential victim transactions and the associated financial loss.

Crypto Asset	#Phishing Addr.	#Victim Addr.	#Victim Txs	# Total Amount	Total USD Value (Min~Max)
USDT/USDC	10,853	659	24,039	8.93M	8.93M
ETH	370	504	1,041	11,102	17.76M ~ 45.14M
Total	11,666	1,137	25,080	NA	26.69M ~ 54.07M

Potential victim transactions: We also present the statistics of potential victim transactions in Table 7 and briefly discuss the result. We found that over 1,100 addresses have potentially lost funds to the attacker. Among them, 650 addresses transferred more than 8.9 million stablecoins to the phishing addresses, and over 500 addresses transferred more than 10,000 ETH. Compared to the confirmed victim transactions, potential victim transactions transferred more ETH to the phishing addresses. The total transferred assets in potential victim transactions are worth $26 million to $54 million USD.

Summary: Overall, our analysis of the victim transactions yields several interesting findings. First, the address poisoning attack achieved a small success rate, as only 0.1% of the targeted addresses have been successfully deceived and lost funds. However, despite such a low success rate, a deceived victim could transfer a large amount of assets to the attacker, leading to a significant financial loss. Second, the primary loss of victims is caused by stablecoins such as USDC and USDT, which are also the attacker's primary target. Third, we found that some victims lost both stablecoins and ETH to the attacker due to mistakenly copying the attacker's address from the phishing transfers. Lastly, the potential profits of the attacker reached ≈$144 million USD, of which ≈$90 million are confirmed from victims targeted in the phishing transfers. Such a compelling result calls for a more comprehensive countermeasure to mitigate such a phishing attack.

5.4 Attacker Clusters

Our previous analysis clearly shows that many phishing addresses are likely controlled by the same entity, e.g., using the same funding address or interacting with the same batching contract. Hence, we can cluster the phishing addresses based on their associated activities with other addresses in this attack. Below, we describe our cluster criterion and present the cluster result.

Cluster criterion: We use the following conditions to cluster the phishing addresses based on their associated activities: (1) if two phishing addresses are utilized by the same funding address, we add them along with the funding address to the same group; (2) if two

[5]We measure the min and max USD value based on the daily closure price of ETH during this study.

Table 8: The statistics of the top 4 attacker clusters.

Cluster #	# Phishing Addr.	Profits	
		USDC/USDT	ETH
1	4,608,594	60.65M	2,407.1
2	689,865	1.70M	1.2
3	276,235	3.70M	10.5
4	185,361	3.48M	651.4
Sum	5,958,322	69.86M	3,080.5

funding addresses have interacted with the same batching contract, we merge their belonged groups; (3) if two batching contracts are deployed by the same address, we merge their associated funding and phishing addresses, and add the contract deployer to the group; (4) if two phishing addresses in two groups are emitted by the same fake token contract, we merge such two groups and add the fake token deployer to the group; (5) if two groups have the same fake token deployer, we merge such two groups.

Cluster result: By applying the above cluster criterion, our result shows that 6 million phishing addresses are formed into 90 clusters, which account for 99.9% of the total phishing addresses. For phishing addresses that do not form a cluster, we found most of them come from the dust-value transfers where the funding address is the same as the phishing address. Among the 90 clusters, there are 11 big clusters that all control more than 10,000 phishing addresses, with the biggest one controlling over 4.6 million addresses. In these 11 clusters, we found the top 4 clusters all gained more than 1 million stablecoins. In Table 8, we show the number of controlled addresses and the profits gained by them. It can be seen that the largest cluster is also the most profitable one, which gained over 60 million stablecoins and 2,400 ETH, respectively accounting for 78.5% and 74.5% of the total confirmed profits. For the other 3 clusters, their profits vary between 1.7 million and 3.5 million on stablecoins and 1 to 652 on ETH. Overall, our clustered result suggests that there are four attacker entities that collected nearly 92% of the total confirmed profits on stablecoins and ETH, where the largest cluster controlling 4.6 million addresses has profited approximately 78% of total confirmed profits.

5.5 Attacker's Money Flow

In this section, we discuss the attacker's money flow during the lifespan of the attack. As previously mentioned, the attacker typically controls two sets of addresses, one serves as the funding address and pays the transaction fee, and the other serves as the phishing addresses to deceive victims. To analyze the attacker's money flow, we downloaded the transaction history of the attacker's funding addresses and phishing addresses. Specifically, for funding addresses, we looked into the transactions that send funds to them. For phishing addresses, we investigated the transactions that transfer funds out. By tracing the transaction history of all the collected funding addresses and phishing addresses that have gained profits, our analysis suggests that the attacker's money flow actually follows a common pattern, which is presented in Fig. 8.

As shown in the figure, we found that the attacker actually purchased ETH from mixing services such as Tornado Cash [25] to top up the funding addresses. Because using those mixing services

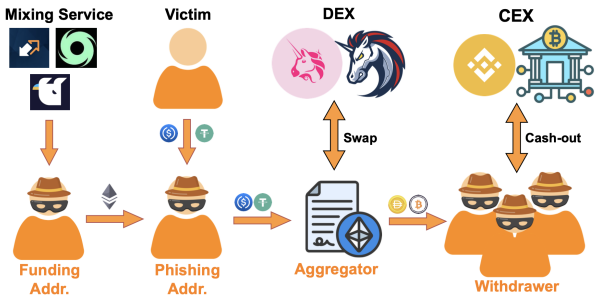


Figure 8: The attacker's money flow in the lifespan of the attack.

does not require users' real identity, it protects the attacker's privacy and anonymity. Then, the attacker uses the funding address to craft phishing transfers and include the phishing address as the payload in the transaction. When the phishing address gains profits from a victim, since the address does not have ETH to cover the transaction fee, the attacker then transfers a small amount of ETH from the funding address to the phishing address. After that, the attacker sends transactions to move the profited funds from the phishing address to an aggregator contract, which then splits the funds and swaps them on decentralized exchanges (DEX) such as Uniswap [26] and 1inch [3] for other types of crypto assets. Thereafter, the attacker transfers the swapped assets from the aggregator contract to multiple withdrawer addresses. In the last step, each withdrawer address interacts with centralized exchanges (CEX) such as Binance [6] and Derbit [12] to cash out the crypto assets. Based on such a money flow pattern, we believe countermeasures can be taken on the CEX site, as they require users' identity to cash out cryptocurrencies, which can provide assistance to help victims recover the financial loss.

6 Discussion

6.1 Ethical Consideration

We have contacted Etherscan to report all the phishing addresses identified in this work in order to protect their users. Meanwhile, we also reported the phishing addresses to the blockchain alerting services such as Chainabuse [2] and HashDit [18]. We are currently working on integrating our detection system with Forta to report phishing addresses involved in this attack continuously. In this paper, we also tried our best to protect the anonymity of the victim and attacker by shortening their addresses presented in the case study in Appendix A. Though the data we collected from the Ethereum blockchain are already part of the public ledger, we also discarded them after accomplishing the paper's analysis and writing.

6.2 Robustness of *Poison-Hunter*

In this address poisoning attack, the key technique explored by the attacker is to use a similar address to craft token transfer records to poison a victim's transfer history. In light of this feature, our detection system relies on matching suspicious transfers with legitimate transfers and comparing the address similarity to detect

phishing addresses. However, due to the freedom of entering arbitrary addresses into the phishing transfers, it is important for a detection system to defend against the defamation attack. That is, if the detection system flags all similar addresses involved in the suspicious transfers as phishing, then the attacker could deliberately enter benign addresses to slander them, rendering the detection system mistakenly flag benign addresses. *Poison-Hunter* avoids this problem with the benign address sifting approach. In this approach, we sift suspicious addresses by checking if they have been activated (e.g., have sent or received transactions or tokens) at the time they are entered into the suspicious transfer. The rationale comes from the fact that if the attacker defames a benign address, the address must have been activated on the blockchain so that the attacker can find it. Hence, the benign address sifting approach can help *Poison-Hunter* detect such a case and prevent the defamation attack. While the attacker could enter a random address into the phishing transfer, the random address is not actually owned by an existing user and remains unknown who will own it in the future, still making the defamation attack ineffective. In addition, *Poison-Hunter* also makes it hard for the attacker to evade future detection. To evade detection, the attacker must transfer funds to the locally generated address before launching the attack. However, this would cause a substantial financial burden to the attacker due to the high transaction cost incurred. For example, in order to activate all of our detected addresses, the estimated cost can be 15 million USD. Therefore, we believe the high monetary cost can prevent the attacker from launching the attack on a large scale, which is the key to the success of this attack.

6.3 Limitations of *Poison-Hunter*

While *Poison-Hunter* has detected millions of phishing transfers and addresses, it also has the following limitations. First, *Poison-Hunter* has focused on detecting phishing transfers from the top 50 ERC-20 tokens, which could miss the phishing transfers involved in some less popular tokens, such as TrueUSD and BUSD, as indicated in the evaluation results on the ground-truth dataset. However, the problem can be solved by expanding the data collection sources and adding those tokens to *Poison-Hunter*. Second, the purpose of employing a benign address sifting procedure in *Poison-Hunter* is to filter out benign addresses and reduce the false positives. However, this may cause some phishing addresses to be removed from our dataset if they have been previously activated before being utilized in the phishing activity. Hence, the scale of our detected phishing addresses may not represent the full spectrum. Nevertheless, we believe such cases are rare given the low success rate of this attack, and the attacker has to generate a large number of inactivated addresses. Third, for the profit analysis, we only consider ETH and the two popular stablecoins, USDC and USDT. Therefore, if certain victims have transferred other forms of cryptocurrencies to the phishing addresses, the real profits lost by the victims could be even higher than what we reported in this paper.

6.4 Countermeasures

Here, we discuss the countermeasures already adopted by Etherscan and recommend more comprehensive countermeasures.

As seen in this work, the attacker typically exploits the address shortening feature employed by Web3 services such as Etherscan and MetaMask. Given their dominant popularity, it is thus necessary for them to take proactive countermeasures to protect its users. Throughout this study, we observed that Etherscan adopted gradually improved countermeasures. For example, in Aug. 2023, Etherscan started to flag fake tokens with red asterisks and published a "low-reputation" text to warn users. Later, in Nov. 2023, Etherscan began to warn users with the pop-up window when they copy the address from transfer events emitted by a flagged fake token. Then, in Feb. 2024, Etherscan employed a new countermeasure, which was to hide zero-value transfers and fake token transfers for users. Users must tune the settings to display such suspicious transfers on the website. However, we believe all of these countermeasures can be easily bypassed if attackers use a self-defined symbol in the token contracts, making it difficult for Etherscan to detect and flag.

To more effectively mitigate this threat, we discuss more countermeasures that could be adopted by Web3 services and individual users. For example, Web3 services may redesign their graphical user interface (GUI) to separate legitimate and suspicious transfers instead of aggregating them together, which could reduce the chance that users mistakenly copy a phishing address. One may also suggest that Web3 services develop tools to better differentiate Ethereum addresses, such as hashing the address with some randomness or randomly displaying a part of the address. While these tools can effectively mitigate the attack, they could hurt usability, as it would be difficult for users to locate their own addresses and other benign addresses in the transaction history. In addition, we recommend that cryptocurrency users take cautious actions when copying and pasting addresses across different services. It is always a good strategy to verify each character and ensure the address belongs to the desired recipient. Another countermeasure that users may adopt is to request an Ethereum Name Record (ENR) for their addresses and use it as a nickname to transfer assets, especially when they need to transfer a large amount of assets or make regular payments to the same recipient.

7 Related Work

Existing works have studied various phishing scams on public blockchains, including Ponzi Schemes [34, 35, 37, 40, 47, 57], fraudulent Initial Coin Offering [43, 51, 52, 60], fake exchange scams [58], phishings [33, 41, 45], giveaway scams [49, 50, 54, 57], honeypot contract scams [42, 53], scam tokens [44, 59], and token theft [38].

Some of the existing works also proposed detection systems to identify phishing addresses on the blockchain, including Chen et al. [41], Chen et al. [39], and He et al. [45]. Specifically, Chen et al. [41] and Chen et al. [39] have proposed to identify a phishing address based on the address's transaction graph on the blockchain through different machine learning models, such as Cascade Feature Extraction Method and Graph Convolutional Networks. However, it remains unknown whether they can be applied to detect the phishing addresses involved in the address poisoning attack, as the phishing addresses are typically inactive and just have a few transactions recorded on the blockchain. The most relevant work to *Poison-Hunter* is He et al. [45], which developed *TxPhishScope* to

detect transaction-based phishing attacks launched on fake websites. In this phishing attack, the victims are attracted to visit a fake website and sign transactions that would send crypto assets to the attacker. *TxPhishScope* detected such an attack by monitoring Certificate Transparency Log [30] to identify suspicious domains and then visiting the suspicious website to trigger the transaction signing operations to detect phishing addresses. Compared to the *TxPhishScope*, our detection system *Poison-Hunter* focuses on a different phishing attack in which the victims are deceived by the phishing transfer records and copy the phishing address to make a transfer. In addition, *Poison-Hunter* also employs different techniques to detect phishing addresses, including matching the suspicious transfers with legitimate transfers and comparing the similarity of the involved addresses.

8 Conclusion

In this paper, we present the first comprehensive analysis of the Ethereum address poisoning attack, a new phishing activity that has crafted more than 14 million phishing transfers from 6 million phishing addresses. Our analysis shows that the attacker has targeted 1.4 million benign addresses and profited nearly $90 million USD from more than 1,800 victims. Our work sheds light on the scale and impact of the address poisoning attack on the Ethereum blockchain, emphasizing an urgent need to effectively prevent such a phishing activity.

Acknowledgement

We thank the anonymous reviewers for their constructive feedback in improving this work. The authors were partially supported by NSF grant CNS-2347486 and one Ethereum academic grant.

References

[1] Binance's loss in the address poisoning attack. https://twitter.com/cz_binance/status/1686764372616515585.
[2] Chaiabuse: Report a scam. https://www.chainabuse.com/report.
[3] 1inch network - leading high capital efficient defi protocols. https://1inch.io/, .
[4] Erc-20 token standard. https://ethereum.org/en/developers/docs/standards/tokens/erc-20/, .
[5] Alchemy - the web3 development platform. https://www.alchemy.com/, .
[6] Binance - cryptocurrency exchange for bitcoin, ethereum. https://www.binance.com/en, .
[7] Blockchain explorer. https://www.blockchain.com/explorer/assets/eth, .
[8] Binance usd (busd). https://www.binance.com/en/research/projects/binance-usd, .
[9] Chainstack: Managed blockchain services. https://chainstack.com/, .
[10] Ethereum tokens. https://cryptoslate.com/blockchain/ethereum/, .
[11] Makerdao - an unbiased global financial system. https://www.circle.com/en/usdc, .
[12] Deribit - crypto options and futures exchange for bitcoin. https://www.deribit.com/, .
[13] Etherscan: Ethereum (eth) blockchain explorer. https://etherscan.io, .
[14] Label word cloud - etherscan. https://etherscan.io/labelcloud, .
[15] Token tracker (erc-20). https://etherscan.io/tokens, .
[16] Ethereum etl. https://github.com/blockchain-etl/ethereum-etl, .
[17] Scam detector docs. https://docs.forta.network/en/latest/scam-detector-bot/, .
[18] Hashdit - securing bnb chain. https://www.hashdit.io/en, .
[19] Ethereum & ipfs apis. develop now on web 3.0. https://infura.io/, .
[20] Metamask: The ultimate crypto wallet for defi, web3 apps. https://metamask.io/, .
[21] Intelligent web3 data platform. https://www.oklink.com/, .
[22] Blockchain infrastructure powering secure, decentralized innovation. https://www.quicknode.com/, .
[23] Stablecoins. https://ethereum.org/en/stablecoins/, .
[24] https://docs.tally.xyz/knowledge-base/managing-a-dao/gnosis-safe, .
[25] Introduction to tornado cash. https://docs.tornadoeth.cash/, .
[26] Uniswap protocol. https://uniswap.org/, .
[27] Usdc - digital dollars backed 1:1 with usd. https://www.circle.com/en/usdc, .
[28] Tether (usdt). https://tether.to/, .
[29] Trueusd (tusd). https://tusd.io/.
[30] Crt.sh. https://certificate.transparency.dev, Retrieved Feb, 5, 2024.
[31] Address poisoning attacks. https://info.etherscan.com/what-is-address-poisoning, Retrieved Feb, 5, 2024.
[32] Address poisoning scams. https://support.metamask.io/privacy-and-security/staying-safe-in-web3/address-poisoning-scams, Retrieved Feb, 5, 2024.
[33] Emad Badawi, Guy-Vincent Jourdan, Gregor Bochmann, and Iosif-Viorel Onut. An automatic detection and analysis of the bitcoin generator scam. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 407–416. IEEE, 2020.
[34] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. Data mining for detecting bitcoin ponzi schemes. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 75–84. IEEE, 2018.
[35] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. Dissecting ponzi schemes on ethereum: identification, analysis, and impact. *Future Generation Computer Systems*, 102:259–277, 2020.
[36] Massimo Bartoletti, Stefano Lande, Andrea Loddo, Livio Pompianu, and Sergio Serusi. Cryptocurrency scams: analysis and perspectives. *Ieee Access*, 9:148353–148373, 2021.
[37] Lingyu Bian, Linlin Zhang, Kai Zhao, Hao Wang, and Shengjia Gong. Image-based scam detection method using an attention capsule network. *IEEE Access*, 9: 33654–33665, 2021.
[38] Jiaqi Chen, Yibo Wang, Yuxuan Zhou, Wanning Ding, Yuzhe Tang, XiaoFeng Wang, and Kai Li. Understanding the security risks of decentralized exchanges by uncovering unfair trades in the wild. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 332–351. IEEE, 2023.
[39] Liang Chen, Jiaying Peng, Yang Liu, Jintang Li, Fenfang Xie, and Zibin Zheng. Phishing scams detection in ethereum transaction network. *ACM Transactions on Internet Technology (TOIT)*, 21(1):1–16, 2020.
[40] Weili Chen, Zibin Zheng, Jiahui Cui, Edith Ngai, Peilin Zheng, and Yuren Zhou. Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In *Proceedings of the 2018 world wide web conference*, pages 1409–1418, 2018.
[41] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, and Yutong Lu. Phishing scam detection on ethereum: Towards financial security for blockchain ecosystem. In *IJCAI*, volume 7, pages 4456–4462, 2020.
[42] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, Yutong Lu, and Yin Li. Honeypot contract risk warning on ethereum smart contracts. In *2020 IEEE International Conference on Joint Cloud Computing*, pages 1–8. IEEE, 2020.
[43] Tiffany Chiu, Victoria Chiu, Tawei Wang, and Yunsen Wang. Using textual analysis to detect initial coin offering frauds. *Journal of Forensic Accounting Research*, 7(1):165–183, 2022.
[44] Bingyu Gao, Haoyu Wang, Pengcheng Xia, Siwei Wu, Yajin Zhou, Xiapu Luo, and Gareth Tyson. Tracking counterfeit cryptocurrency end-to-end. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(3):1–28, 2020.
[45] Bowen He, Yuan Chen, Zhuo Chen, Xiaohui Hu, Yufeng Hu, Lei Wu, Rui Chang, Haoyu Wang, and Yajin Zhou. Txphishscope: Towards detecting and understanding transaction-based phishing on ethereum. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 120–134, 2023.
[46] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, 1:36–63, 2001.
[47] Tyler Kell, Haaroon Yousaf, Sarah Allen, Sarah Meiklejohn, and Ari Juels. Forsage: Anatomy of a smart-contract pyramid scheme. *arXiv preprint arXiv:2105.04380*, 2021.
[48] Kai Li, Shixuan Guan, and Darren Lee. Towards understanding and characterizing the arbitrage bot scam in the wild. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(3):1–29, 2023.
[49] Kai Li, Darren Lee, and Shixuan Guan. Understanding the cryptocurrency free giveaway scam disseminated on twitter lists. In *2023 IEEE International Conference on Blockchain (Blockchain)*, pages 9–16. IEEE, 2023.
[50] Xigao Li, Anurag Yepuri, and Nick Nikiforakis. Double and nothing: Understanding and detecting cryptocurrency giveaway scams. 2023.
[51] Daniel Liebau and Patrick Schueffel. Crypto-currencies and icos: Are they scams? an empirical study. *An Empirical Study (January 23, 2019)*, 2019.
[52] Kenny Phua, Bo Sang, Chishen Wei, and Gloria Yang Yu. Don't trust, verify: The economics of scams in initial coin offerings. *Available at SSRN 4064453*, 2022.
[53] Christof Ferreira Torres, Mathis Steichen, et al. The art of the scam: Demystifying honeypots in ethereum smart contracts. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1591–1607, 2019.
[54] Iman Vakilinia. Cryptocurrency giveaway scam with youtube live stream. In *2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0195–0200. IEEE, 2022.
[55] Yibo Wang, Qi Zhang, Kai Li, Yuzhe Tang, Jiaqi Chen, Xiapu Luo, and Ting Chen. ibatch: saving ethereum fees via secure and cost-effective batching of

smart-contract invocations. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 566–577, 2021.

[56] Yibo Wang, Kai Li, Yuzhe Tang, Jiaqi Chen, Qi Zhang, Xiapu Luo, and Ting Chen. Towards saving blockchain fees via secure and cost-effective batching of smart-contract invocations. *IEEE Transactions on Software Engineering*, 49(4): 2980–2995, 2023.

[57] Pengcheng Xia, Haoyu Wang, Xiapu Luo, Lei Wu, Yajin Zhou, Guangdong Bai, Guoai Xu, Gang Huang, and Xuanzhe Liu. Don't fish in troubled waters! characterizing coronavirus-themed cryptocurrency scams. In *2020 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–14, 2020. doi: 10.1109/eCrime51433.2020.9493255.

[58] Pengcheng Xia, Haoyu Wang, Bowen Zhang, Ru Ji, Bingyu Gao, Lei Wu, Xiapu Luo, and Guoai Xu. Characterizing cryptocurrency exchange scams. *Computers & Security*, 98:101993, 2020.

[59] Pengcheng Xia, Haoyu Wang, Bingyu Gao, Weihang Su, Zhou Yu, Xiapu Luo, Chao Zhang, Xusheng Xiao, and Guoai Xu. Trade or trick? detecting and characterizing scam tokens on uniswap decentralized exchange. *Proc. ACM Meas. Anal. Comput. Syst.*, 5(3), dec 2021. doi: 10.1145/3491051. URL https://doi.org/10.1145/3491051.

[60] Dirk A Zetzsche, Ross P Buckley, Douglas W Arner, and Linus Föhr. The ico gold rush: It's a scam, it's a bubble, it's a super challenge for regulators. *University of Luxembourg Law Working Paper*, (11):17–83, 2017.

A Case Study

This section presents a case study on the top phishing addresses that have made the most profits. We rank the phishing addresses by their profits and show the top 10 profited addresses of stablecoins and ETH in Table 9 and Table 10.

Table 9: Top profited phishing addresses by stablecoins.

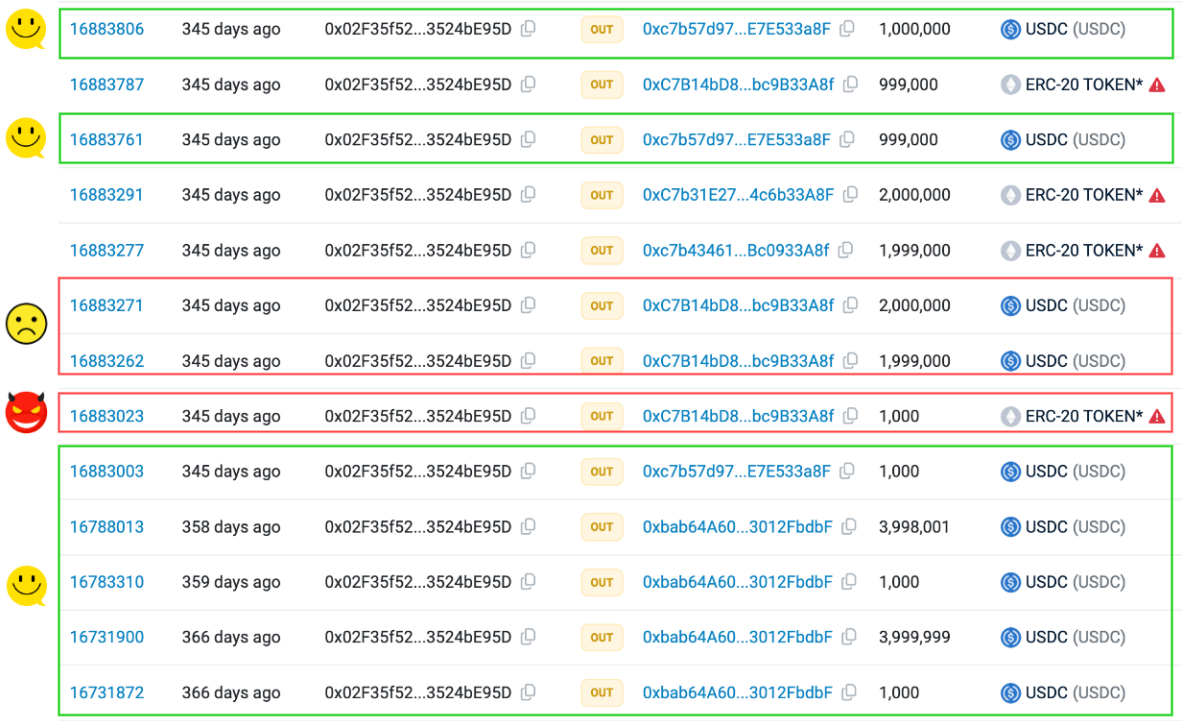
Rank	Address	Profit (USDC/USDT)
1	0xa7bf4874******a9e90570	20,000,000
2	0xc7b14bd8******c9b33a8f	3,999,000
3	0x1cbb23db******269b758a	3,554,610
4	0x74c9bdbe******8560e1ca	2,030,000
5	0xbb2edba8******dd619455	2,000,000
6	0xcba796e8******1134c994	1,200,000
7	0x9cadec5b******4ebac282	1,107,010
8	0x73435a47******2bca79f7	1,045,150
9	0x9e5c0ec6******ea8671c3	1,000,992
10	0x80d707f2******7e4dbea2	1,000,000

Profits of stablecoins: In Table 9, all 10 phishing addresses have profited at least 1 million stablecoins. Among them, 2 addresses collected over 2 million stablecoins, 2 addresses collected over 3 million stablecoins, and 1 address collected 20 million. The largest profit was gained by `0xa7bf4874******a9e90570`. The financial loss has been reported on X (formerly Twitter) and confirmed by Binance[1]. In this paper, we skip this case and analyze the second-highest financial loss (3.999 million) collected by the phishing address `0xc7b14bd8******c9b33a8f`. For this phishing address, we looked into its transaction history and found that all of the profits are collected from one victim whose address is `0x02F35f52******524bE95D`. We investigated the victim's transaction history and presented it in Fig. 9. We found that the victim initially received a total of 20 million USDC from GnosisSafeProxy[24] on Feb. 15, 2023. Then, on Mar. 1, 2023, and Mar. 9, 2023, each day, the victim transferred nearly 4 million USDC to `0xbab64A60******012FbdbF` in 2 transactions, with the first transaction transferring a small amount of 1,000 followed by another transaction transferring 3.999 million. During this period, no phishing transfer was sent by the attacker. Then, on Mar. 22, 2023 at 12:02, the victim transfered another 1,000 to a new recipient `0xc7b57d97******7E533a8F`. Such a transfer was observed by the attacker, who then immediately crafted a phishing transfer at 12:06, using a highly similar address `0xC7B14bD8******c9B33A8f`. Comparing such two recipient addresses, the first 3 and last 5 hexadecimal characters are the same. After, at 12:54 and 12:55, the victim respectively transferred 1.999 million and 2 million USDC to the attacker's address, resulting in a total financial loss of 3.999 million USD. After two hours, the victim realized the mistake and the intended recipient address `0xc7b57d97******7E533a8F` did not receive the USDC, then the victim transferred 3.999 million USDC to the correct recipient address in 3 transactions, respectively sending 0.999 million, 1 million, and 2 million USDC. In this process, we can see that when sending a large amount of UDSC, every time the victim would send a small amount to verify the recipient before sending the large amount. In this case, after the victim transferred the first 1,000 to the correct recipient address, the attacker crafted a fake transfer with the same transferred amount. Unfortunately, the victim was deceived and thought the attacker's address was the correct recipient. Then, the victim transferred the remaining 3.999 million USDC to the attacker. However, there is no way for the victim to recover the loss. So, the victim had to transfer another 3.999 million USDC to the correct recipient address in 3 transactions. During these three transactions, though there were new fake transfers targeting the victim, the victim already realized the attack and never made the same mistake again.

Table 10: Top profited phishing addresses by ETH.

Rank	Address	Profit (ETH)
1	0xba8ba758******2a05c0e6	2,000
2	0xbac63481******a32caeae	645
3	0x437eef72******a11a9117	130
4	0x70ad93d0******f846eef2	100
5	0x52a083a4******81a33a49	42.726452
6	0xaee355bd******6108fad4	38.837000
7	0x46ab2d74******70c58231	37.334990
8	0xe387029c******35ae0035	30
9	0xb9d472fc******d10a9b4b	20.335611
10	0x041c6ff7******ee4db633	18.474804

Profits of ETH: Table 10 shows the top 10 phishing addresses that have profited at least 18 ETH. Among them, 6 addresses collected 18 to 43 ETH, 2 addresses collected over 100 ETH, 1 address collected 645 ETH, and 1 address collected 2,000 ETH. The largest profit was gained by `0xba8ba758******2a05c0e6`. We looked into this phishing address's transaction history and found the loss was from the victim `0x01BEF997******7cf83Ec0`. Below, we illustrate the phishing process against the victim in detail. We show both the victim's transaction history and token transfer history in Fig. 10. In token transfer history, the victim initially transferred a total of 200K USDC to the benign recipient address `0xbA83cE92******3305c0E6` on Sep. 2, 2023. After a few minutes, the attacker crafted three phishing transfers respectively in 6, 14, 17 blocks. All the recipient addresses look highly similar to the benign recipient address. After



16883806	345 days ago	0x02F35f52...3524bE95D	OUT	0xc7b57d97...E7E533a8F	1,000,000	USDC (USDC)
16883787	345 days ago	0x02F35f52...3524bE95D	OUT	0xC7B14bD8...bc9B33A8f	999,000	ERC-20 TOKEN*
16883761	345 days ago	0x02F35f52...3524bE95D	OUT	0xc7b57d97...E7E533a8F	999,000	USDC (USDC)
16883291	345 days ago	0x02F35f52...3524bE95D	OUT	0xC7b31E27...4c6b33A8F	2,000,000	ERC-20 TOKEN*
16883277	345 days ago	0x02F35f52...3524bE95D	OUT	0xc7b43461...Bc0933A8f	1,999,000	ERC-20 TOKEN*
16883271	345 days ago	0x02F35f52...3524bE95D	OUT	0xC7B14bD8...bc9B33A8f	2,000,000	USDC (USDC)
16883262	345 days ago	0x02F35f52...3524bE95D	OUT	0xC7B14bD8...bc9B33A8f	1,999,000	USDC (USDC)
16883023	345 days ago	0x02F35f52...3524bE95D	OUT	0xC7B14bD8...bc9B33A8f	1,000	ERC-20 TOKEN*
16883003	345 days ago	0x02F35f52...3524bE95D	OUT	0xc7b57d97...E7E533a8F	1,000	USDC (USDC)
16788013	358 days ago	0x02F35f52...3524bE95D	OUT	0xbab64A60...3012FbdbF	3,998,001	USDC (USDC)
16783310	359 days ago	0x02F35f52...3524bE95D	OUT	0xbab64A60...3012FbdbF	1,000	USDC (USDC)
16731900	366 days ago	0x02F35f52...3524bE95D	OUT	0xbab64A60...3012FbdbF	3,999,999	USDC (USDC)
16731872	366 days ago	0x02F35f52...3524bE95D	OUT	0xbab64A60...3012FbdbF	1,000	USDC (USDC)

Figure 9: The token transfer history of the victim address 0x02f35f52****524be95d.**

Token

18048260	240 days ago	0x01BEF997...97cf83Ec0	OUT	0xbA83cE92...A3305c0E6	7,800,015.897626	Tether USD (USDT)
18048243	240 days ago	0x01BEF997...97cf83Ec0	OUT	0xBA4B5D1A...D7725C0E6	200,000	ERC-20 TOKEN*
18048240	240 days ago	0x01BEF997...97cf83Ec0	OUT	0xbA4583d3...9Ef05C0E6	200,000	ERC-20 TOKEN*
18048232	240 days ago	0x01BEF997...97cf83Ec0	OUT	0xbA8BA758...A2A05C0e6	200,000	ERC-20 TOKEN*
18048226	240 days ago	0x01BEF997...97cf83Ec0	OUT	0xbA83cE92...A3305c0E6	200,000	Tether USD (USDT)

ETH

18927070	117 days ago	0x01BEF997...97cf83Ec0	OUT	0xbA83cE92...A3305c0E6	3,563.88989526 ETH	0.00252
18926719	117 days ago	0x01BEF997...97cf83Ec0	OUT	0xbA8BA758...A2A05C0e6	2,000 ETH	0.0021
18926665	117 days ago	0x01BEF997...97cf83Ec0	OUT	0xbA83cE92...A3305c0E6	300 ETH	0.000588
18688295	150 days ago	OKX 3	IN	0x01BEF997...97cf83Ec0	4,999.9979 ETH	0.00064665

Figure 10: The transaction and token transfer history of victim address 0x01BEF997****7cf83Ec0.**

90 days, the victim received more than 4,999 ETH from `0xA7EFAe72******8dD593f3`. Then, in 33 days, the victim transferred 300 ETH to the same benign recipient address. Later on the same day, the victim made a mistake and sent 2,000 ETH to the attacker's phishing address `0xba8ba758******2a05c0e6`, which was copied from the first phishing transfer. After 315 blocks (≈1 hour), the victim realized the mistake, and the benign recipient `0xbA83cE92******3305c0E6` did not receive the ETH. So, the victim transferred another 3,563 ETH to the benign recipient in one transaction. In this case, the victim has previously interacted with the same benign recipient in both token transfers and ETH transfers. However, when the victim intended to make another ETH transfer to the same benign recipient, the victim visited its token transfer history and ended up copying the phishing address and sending ETH to it, resulting in a financial loss.