Unmasking the Shadow Economy: A Deep Dive into Drainer-as-a-Service Phishing on Ethereum

Bowen He*
bowen_os@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China
Mohamed bin Zayed University of Artificial Intelligence
Abu Dhabi, United Arab Emirates

Yufeng Hu
yufenghu@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Zhuo Chen
hypothesiser.hypo@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Yuan Chen
yuanchen96@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Ting Yu
ting.yu@mbzuai.ac.ae
Mohamed bin Zayed University of Artificial Intelligence
Abu Dhabi, United Arab Emirates

Rui Chang
crix1021@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Lei Wu
lei_wu@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Yajin Zhou†
yajin_zhou@zju.edu.cn
Zhejiang University
Hangzhou, Zhejiang, China

Abstract

The prosperity of Ethereum gives rise to a new type of transaction-based phishing scam. Specifically, users are tempted to visit phishing websites and sign phishing transactions that allow scammers to withdraw their tokens. Meanwhile, to accelerate the deployment of phishing websites, scammers have introduced a business model, Drainer-as-a-Service (DaaS). In this model, drainer operators focus on crafting specialized phishing toolkits, named "wallet drainers", while drainer affiliates handle the deployment and promotion of phishing websites. After stealing victims' tokens, they will distribute profits. In this paper, we present the first systematic study of DaaS on Ethereum. To begin with, we propose a snowball sampling approach to build the first large-scale DaaS dataset, including **1,910** profit-sharing contracts, **56** operator accounts, **6,087** affiliate accounts, and **87,077** profit-sharing transactions. Then, we analyze the scale of DaaS from the perspectives of victims, operators, and affiliates, and perform clustering analysis to uncover dominant DaaS families. Finally, we reported DaaS accounts in the dataset and **32,819** phishing websites deployed with DaaS toolkits to the community. Our work aims to serve as a guide for Ethereum service providers to enhance user protection against DaaS.

*Part of this work was done when the author was a research intern at BlockSec.
†Corresponding author: Yajin Zhou (yajin_zhou@zju.edu.cn).

IMC '25, Madison, WI, USA

ACM ISBN 979-8-4007-1860-1/2025/10
https://doi.org/10.1145/3730567.3764476

CCS Concepts

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

Keywords

Decentralized Finance, Ethereum, phishing detection

ACM Reference Format:
Bowen He, Yufeng Hu, Zhuo Chen, Yuan Chen, Ting Yu, Rui Chang, Lei Wu, and Yajin Zhou. 2025. Unmasking the Shadow Economy: A Deep Dive into Drainer-as-a-Service Phishing on Ethereum. In *Proceedings of the 2025 ACM Internet Measurement Conference (IMC '25), October 28–31, 2025, Madison, WI, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3730567.3764476

1 Introduction

Since the rise of Decentralized Finance (DeFi), Ethereum has consistently attracted significant capital and investments from users. By April 2025, the Total Value Locked (TVL) on Ethereum had surpassed $46.4 billion, accounting for 51.2% of the overall TVL in the DeFi sector [50]. However, this influx of capital has been accompanied by a surge in cryptocurrency phishing scams [7, 11, 16, 22–24, 36, 42, 45–47, 54]. In these scams, users are first tricked into visiting fraudulent websites that pose as legitimate Ethereum projects. Once there, they connect their wallets and unknowingly sign phishing transactions, which leads to unauthorized token withdrawals. Such phishing scams have become a significant threat to DeFi users. Between September and November 2024, victims lost over $60 million in total due to phishing websites [60, 61, 64].

At the same time, a toolkit known as the *wallet drainer* [38] was exposed by the security community. As the name suggests, this tool ultimately *drains users' wallets*. Specifically, scammers developed this toolkit to accelerate and scale the deployment of phishing websites. It automatically prompts users to connect their wallets,

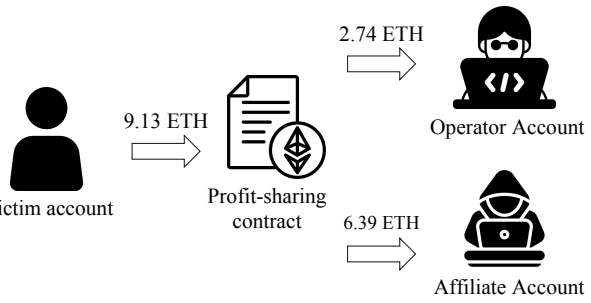

Figure 1: Example of a Profit-sharing Transaction [12]. The victim account mistakenly transferred 9.13 ETH to the profit-sharing contract. Then the profit-sharing contract distributed 30% to the operator account and 70% to the affiliate account.

scans their tokens, and generates phishing transactions. The first well-known wallet drainer, `Monkey Drainer` [23], operated from August 2022 to February 2023, causing approximately $13 million in losses. Currently, the Web3 security community receives hundreds of daily reports about drainer phishing websites, underscoring the severity of this issue [20, 26, 29, 59].

Building on the use of wallet drainers, scammers have developed a new business model known as *Drainer-as-a-Service* (DaaS) [59, 62]. This model comprises two key roles: *operators and affiliates*. *Operators* are responsible for developing the toolkit of wallet drainers, which includes *phishing websites* and *profit-sharing smart contracts*. *Affiliates* acquire the toolkit from the operators, deploy phishing websites, and promote them to potential victims. Once tokens are stolen from victims through these phishing websites, the *profit-sharing contracts* automatically distribute tokens between the affiliates and operators. In this paper, we collectively refer to operator accounts, affiliate accounts, and profit-sharing contracts as DaaS accounts.

The collaboration between operators and affiliates is carried out in four steps. First, operators promote this business model on social media and invite affiliates to join their Telegram groups [14, 32, 37, 39, 53, 56]. Next, affiliates must demonstrate their ability to spread phishing websites to a wide user base and provide an Ethereum account to receive shared profits. Subsequently, operators customize a wallet drainer for each affiliate and create private Telegram groups to share real-time information, such as the number of tokens stolen from various users. Finally, affiliates deploy the phishing websites with wallet drainers and distribute them on social platforms. Victims are then lured to visit the phishing websites, connect their wallets, and sign phishing transactions. After draining victims' wallets, the profit-sharing contract launches transactions to distribute the profits to the operator and affiliates. Figure 1 illustrates an example of profit-sharing transactions. In this example, upon receiving the victim's ETH, the profit-sharing contract distributes the tokens in a 3:7 ratio. In profit-sharing transactions, operators often provide affiliates with a larger share of the profits as an incentive to promote phishing websites, which will be detailed in Section 4.3.

Despite the significant threat posed by DaaS, so far no comprehensive studies have focused on it. While we have discussed DaaS in several industry blogs, they mainly analyze individual exploits or focus on specific DaaS families (e.g., `Inferno Drainer`) [70]. However, a thorough understanding of this shadowy domain is still lacking within the community, which is crucial for developing effective mitigation strategies. Therefore, there is an urgent need for empirical research in this area.

To address this gap, our work presents the first empirical study of DaaS on Ethereum, aiming to raise awareness of this emerging threat and contribute to the mitigation of such threats. Specifically, we aim to answer the following four research questions: First, what are the characteristics of DaaS and key features of profit-sharing transactions? Second, how can we build a large-scale dataset of the addresses involved in DaaS, including profit-sharing contracts, operator accounts, and affiliate accounts? Third, what is the current scale of DaaS, including victim losses and the corresponding profits of operators and affiliates? Fourth, what are the relationships among DaaS accounts, and how many DaaS families exist?

Large-scale Dataset Construction. The absence of a publicly available DaaS dataset poses a significant challenge for our study. We address this challenge by leveraging the foundational behavior of the DaaS family, *profit-sharing*, to build a large-scale DaaS dataset. Specifically, the profit has to be shared between the operators and affiliates. This dataset will not only establish a robust foundation for our subsequent measurement work but also serve as a valuable resource for all research on Web3 phishing.

Based on this insight, we take an iterative approach to construct the dataset. First, we collect phishing contracts from public sources. We then analyze the historical transactions of these contracts. If these contracts exhibit profit-sharing behavior, we treat them as profit-sharing contracts and extract the operator and affiliate accounts from the historical transactions, establishing the initial seed dataset. After that, we can expand the dataset by discovering new profit-sharing smart contracts from the identified operator and affiliate accounts, since one operator or affiliate may leverage different profit-sharing smart contracts. We iteratively perform this analysis until no new profit-sharing contracts are discovered.

Building upon the dataset construction approach described earlier, from March 1, 2023 to April 1, 2025, we identified **1,910** profit-sharing contracts, **56** operator accounts, **6,087** affiliate accounts, and **87,077** profit-sharing transactions. In total, the operator accounts earned **$23.1 million**, while the affiliate accounts earned **$111.9 million**. Additionally, we assembled a team of experts to validate the accuracy of our dataset and found no false positives. Since DaaS accounts are densely connected through profit-sharing transactions, our snowball sampling approach effectively leverages this connectivity to expand the dataset. However, a limitation of our approach is that accounts not connected to the seed dataset through transactions may be overlooked. To mitigate this issue, we utilized four different dataset sources and conducted an extended dataset collection to improve its coverage. Although we cannot guarantee that all DaaS accounts on Ethereum have been captured, we acknowledge that this is the first such dataset, and the scale of our dataset provides a solid foundation for subsequent measurement studies. We will release the dataset to the research community.

Scale of DaaS. After obtaining the dataset, we first analyze the scale of DaaS from the perspectives of victim, operator, and affiliate accounts. First, a significant number of individuals fall victim to DaaS, with incidents exceeding **100** cases daily. Although **83.5%**

of the victim accounts experience losses of less than **$1,000**, the tokens stolen from numerous victims still contribute substantial profits to DaaS accounts. Second, while there are a large number of operator and affiliate accounts, the majority of the profits are concentrated in a few key accounts. Specifically, **25.0%** of the operator accounts accounted for **75.7%** of the total operator profits, and **7.4%** of the affiliate accounts received **75.6%** of the total affiliate profits. Third, we observe that there are fund flow transfers between several operator accounts. It motivates us to cluster them based on their fund flow relationships in Section 7.

DaaS Family Clustering. To identify and categorize DaaS accounts into distinct families, we develop an algorithm that first groups operator accounts based on their transactions. Next, we organize profit-sharing and affiliate accounts according to their associated operator accounts. Ultimately, we classify all DaaS accounts into **nine** distinct DaaS families. Among these families, `Angel Drainer`, `Inferno Drainer`, and `Pink Drainer` stand out as the dominant ones, accounting for **93.9%** of all profits. We then proceed to compare these three leading DaaS families from the perspectives of *drainer toolkits, contract implementation, contract lifecycle, affiliate requirements, and affiliate management.*

Contributing to the Anti-phishing Community. To safeguard users, we report DaaS accounts in the dataset, of which only **10.8%** have been labeled on Etherscan, the leading Ethereum blockchain explorer. Additionally, we report **32,819** phishing websites deployed using drainer toolkits, to the Web3 security community. Specifically, we leverage the fact that over 70% of phishing sites are secured with TLS connections [72, 79], and newly issued X.509 certificates can be tracked through Certificate Transparency Logs [57]. Our approach begins by collecting 867 drainer toolkits from Telegram groups and reported phishing websites. We then extract suspicious domains from newly issued certificates using keyword similarity matching. Finally, we crawl these websites to verify if they are deployed with drainer toolkits. If confirmed, we classify them as phishing websites and report them accordingly.

Contributions. We summarize our main contributions as follows.

- **Anatomy of Drainer-as-a-Service on Ethereum.** Through sample data analysis, we systematize Drainer-as-a-Service on Ethereum, covering both the whole operational pipeline and the profit-sharing process.
- **First large-scale dataset.** We build the first large-scale DaaS dataset for our community. We make efforts to ensure the accuracy and integrity of the dataset and share it with the research community. [1]
- **First in-depth measurement study.** We analyze the scale of DaaS and conduct a clustering analysis of DaaS accounts. Additionally, we provide a comprehensive comparison of the dominant DaaS families, offering valuable insights into the entire ecosystem.
- **Actions to protect users.** We identify **32,819** phishing websites deployed with drainer toolkits in the wild. To protect users, we reported both DaaS accounts and these phishing websites to the community, receiving a bounty of $800 and their recognition in return.

[1] https://github.com/blocksecteam/DaaS_dataset

2 Background

2.1 Ethereum

Ethereum is an open-source blockchain platform that enables the creation of decentralized applications (DApps). Unlike Bitcoin, which simply serves as digital currency, Ethereum is designed to function as a general-purpose programmable blockchain [71].

Accounts. In Ethereum, there are two types of accounts: Externally Owned Accounts (EOAs) and Contract Accounts (CAs) [27]. EOAs are managed by private keys to launch transactions. By contrast, CAs are governed by the code within their smart contracts, which remains immutable once deployed.

Transactions. Ethereum transactions are signed network messages utilized to modify the states of accounts. Typically, the transaction parameters consist of a sender, recipient, value, data payload, and other metadata [71].

Tokens. Within the Ethereum ecosystem, tokens refer to the digital assets or units of value that serve various purposes such as fundraising, rewards, or voting. ETH is the native token used to facilitate transactions and pay for computations. Generally, tokens can be categorized into fungible and non-fungible tokens. The majority of fungible token contracts adhere to the ERC-20 standards [1], commonly referred to as ERC-20 tokens. While most non-fungible token (NFT) contracts conform to ERC-721 [3] and ERC-1155 standards [2].

The core operations involved in ERC-20 and ERC-721 token management can be categorized into two types: approval and transfer functions. Specifically, approval functions grant another account the authority to control user tokens, while transfer functions move user tokens to a different account.

Wallets. Ethereum users typically rely on digital wallets to manage the private keys of their accounts and launch transactions. Among these wallets, MetaMask [33] is the most widely used one with more than 30 million monthly active users [4].

2.2 Wallet Drainers on Ethereum

As users engage in token trading through Ethereum transactions, a new type of phishing scam has surfaced. To begin with, users are lured to visit a fraudulent Ethereum service website. Then they mistakenly sign phishing transactions that withdraw all of their tokens. This type of transaction-based phishing scam happens almost every day [26] and has inflicted millions of dollars in losses [8].

To maximize profits, scammers develop phishing toolkits that automatically search users' tokens and generate phishing transactions. Due to its objective of draining users' tokens entirely, the Ethereum anti-phishing community has dubbed the phishing toolkit as "wallet drainers" [38].

2.3 Drainer-as-a-Service on Ethereum

Cybercrime-as-a-Service. Cybercrime poses a significant threat to the security of Internet users. According to the FBI Internet Crime Report [5], the total losses of cybercrime in 2022 exceeded $10.2 billion. More than just an illicit pastime, cybercrime has evolved into a means for cybercriminals to make a livelihood [82]. And Cybercrime-as-a-Service has emerged as a new business model in the underground market [88]. In detail, the affiliate can purchase

cybercrime services provided by the operator, who is responsible for developing and maintaining the cybercrime infrastructure [6]. The payment methods encompass monthly subscription fees, one-time payments, and profit-sharing agreements [82].

Drainer-as-a-Service. The business model of Cybercrime-as-a-Service can also be applied to the wallet drainers, which we refer to as Drainer-as-a-Service (DaaS). More precisely, drainer affiliates launch phishing websites via wallet drainers developed by operators. The revenue model for DaaS can be categorized into three types. The first type [15, 25, 34, 37, 53] involves a one-time payment for the wallet drainer, without any profit-sharing arrangement between the operators and affiliates. The second type [21] requires both a subscription fee and a percent of the profits. The third type [14, 32, 56] only entails sharing profit after the successful theft of users' tokens. In this paper, we focus on the last two scenarios, where the profit-sharing process can be observed through Ethereum transactions.

3 Study Design

In this work, we seek to address the following research questions (RQs) to gain a comprehensive understanding of DaaS on Ethereum:

- RQ1: **What are the characteristics of DaaS?** Although DaaS has caused significant losses for users on Ethereum, the community still lacks a fundamental understanding of this emerging business model. In particular, the key features of DaaS include its operational pipeline and profit-sharing mechanisms. Moreover, accurately defining the characteristics of DaaS is crucial for identifying them on a large-scale.
- RQ2: **Can we build a large-scale dataset of DaaS?** To the best of our knowledge, there is currently no comprehensive dataset available that captures DaaS phishing, including profit-sharing contracts, operator accounts, affiliate accounts, and related transactions. Creating such a dataset is vital not only for conducting detailed measurement studies but also for contributing to the broader research community by open-sourcing the dataset.
- RQ3: **What are the harmful effects of DaaS?** Using the large-scale dataset, we aim to first assess the basic scale of DaaS. Specifically, what is the total volume of profit-sharing transactions, and how are the associated losses and gains distributed among victim accounts, operator accounts, and affiliate accounts? Answering these questions will shed light on the overall magnitude of DaaS phishing.
- RQ4: **How many DaaS families dominate this underground economy?** To gain deeper insights into the structure and dynamics of the overall ecosystem, we should analyze the relationships between different accounts in the dataset and uncover the majority of DaaS families.

In our work, we begin by joining several Telegram groups of "wallet drainers" and summarizing the profit-sharing process. Next, leveraging the key characteristics of profit-sharing transactions, we develop a system to construct the first large-scale dataset of DaaS on Ethereum. This dataset includes **1,910** profit-sharing contracts, **56** operator accounts, **6,087** affiliate accounts, and **87,077** profit-sharing transactions. All accounts in the dataset have been manually verified and subsequently reported to Etherscan and relevant security teams. Utilizing the dataset, we conduct an in-depth analysis of DaaS phishing attacks from the perspectives of victim accounts, operator accounts, affiliate accounts, and DaaS families.

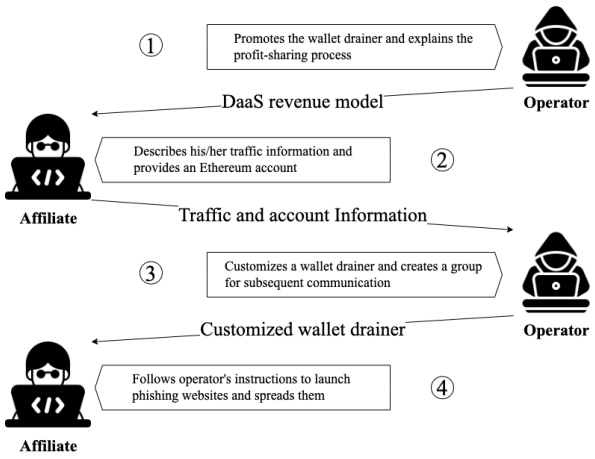

Figure 2: Workflow of DaaS. In this collaborative model, the operator provides the wallet drainer toolkit, while the affiliate supplies the traffic to attract users.

4 Anatomy of DaaS

To gain a preliminary understanding of DaaS, we joined several related Telegram groups [14, 21, 32, 37, 56], communicated with operators, acquired wallet drainers, and summarized the profit-sharing process. These groups can be discovered by searching for keywords like "wallet drainer" on platforms such as Twitter, Github, and Telegram. In this section, we will first describe the operational pipeline of DaaS (Section 4.1). Following that, we will analyze the profit-sharing process of DaaS (Section 4.2). Lastly, we will present an overview of profit-sharing transactions on Ethereum (Section 4.3). These efforts serve as a foundation for detecting DaaS accounts.

4.1 Operational Pipeline of DaaS

As depicted in Figure 2, the complete DaaS workflow consists of four steps.

Operator promotes the DaaS revenue model. To attract prospective affiliates, the operator introduces the wallet drainer and its potential profits on social media [55]. Subsequently, the affiliate joins the Telegram group [14, 21, 32, 37, 56] linked in the message. The operator then provides a comprehensive explanation of the collaboration process, showcasing real-time profits and outlining the affiliate requirements.

Affiliate provides traffic and account information. To demonstrate their ability to reach a significant number of Ethereum users and generate profits, the affiliate is required to present traffic information. For example, the affiliate may manage a popular Web3 Telegram group and have access to thousands of potential victims. Additionally, the affiliate must provide an Ethereum account, which the operator uses to customize the wallet drainer for the affiliate.

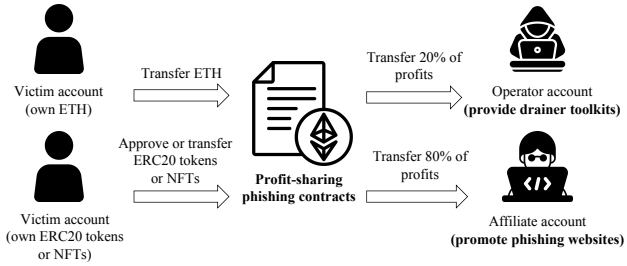

Figure 3: Profit-sharing Scenarios of DaaS. The profit-sharing ratio can vary between 10%, 20%, 30%, and other percentages. [2]

Operator customizes the wallet drainer. To track the phishing profits generated by each affiliate, the operator customizes the wallet drainer for each individual. Specifically, the drainer toolkits log information about the associated affiliate, which is crucial for the profit-sharing process after stealing users' tokens. Moreover, the operator creates Telegram groups where affiliates receive real-time information on phishing websites and updates regarding the wallet drainer.

Affiliate launches phishing websites. Following the operator's guidance, the affiliate deploys phishing websites and promotes them on social platforms such as Twitter, Telegram, Instagram, and Discord. These websites deceive users into visiting and signing fraudulent transactions. Once the users' tokens are stolen, the phishing program sends messages to record the associated affiliate accounts and phishing transactions. The profit-sharing process then occurs automatically.

4.2 Profit-sharing Process of DaaS

The profit-sharing process starts after the victim, lured by the phishing websites, connects their wallet and mistakenly signs the phishing transaction. To analyze the details of the profit-sharing process, we collect 30 profit-sharing contracts from large-scale phishing incidents [9, 10, 17, 30, 31, 40, 41, 43, 44, 48, 49, 62] and examine their transactions. Figure 3 illustrates three distinct profit-sharing scenarios in DaaS, determined by the token types in the victim's account.

The Native Token (ETH). Victims are tricked into directly transferring the native token i.e., ETH, to the profit-sharing contract. To entice potential victims, the drainer operator will deploy a profit-sharing contract containing a payable function with names such as "claim" or "mint". [3] Listing 1 demonstrates an example of the profit-sharing function. Invoking these functions of the profit-sharing contract directly transfers ETH and returns nothing. Subsequently, the obtained ETH will be transferred to the operator and affiliate accounts, respectively.

ERC20 Tokens. If the victim account holds ERC20 tokens, the phishing transaction authorizes their tokens to a profit-sharing contract. The operator then triggers the profit-sharing contract to

[2]In this figure, we have simplified the profit-sharing process to ensure ease of understanding.
[3]Example: 0x5e0102e6448b602fcd955fcfc7ceea9a36e7e5f0.

```
function claimRewards(address affiliate_account) public
    payable {
  uint256 operator_profit = msg.value * 20 / 100;
  uint256 affiliate_profit = msg.value * 80 / 100;
  payable(operator_account).transfer(operator_profit);
  payable(affiliate_account).transfer(affiliate_profit);
  // operator_account is defined upon contract deployment
}
```

Listing 1: Simpified Source Code of a Profit-sharing Function. The "claimRewards" function transfers 20% and 80% of the victim's ETH to the operator and affiliate accounts.

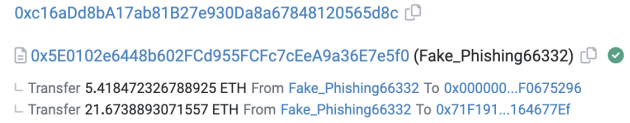

Figure 4: Example of a Profit-sharing Transaction. The victim account was lured to transfer 27.1 ETH to the profit-sharing contract. Then 5.4 ETH and 21.7 ETH were forwarded to operator and affiliate accounts [13].

invoke the *TransferFrom* function of the token contract, transferring a portion of the victim's tokens to the operator and affiliate accounts.

NFTs. Similar to ERC20 token phishing, when the victim holds NFTs, the phishing transaction requests approval or transfer of the NFTs. Since NFTs cannot be divided, they are sold on marketplaces like Blur [18] or OpenSea [35] in exchange for ETH, which is then distributed between the operator and affiliate accounts.

4.3 Profit-sharing Transactions of DaaS

After gaining control over the victims' tokens, the profit-sharing contracts promptly launches profit-sharing transactions. These transactions typically involve two transfers: one to the operator and one to the affiliate, with a ratio such as 1:4. When victims visit phishing websites, they are tricked into transferring or approving tokens to the phishing contract. Once the tokens are received, the phishing contract is triggered, and the profit-sharing process occurs.

Figure 4 illustrates an example of a profit-sharing transaction. The victim account was deceived into transferring 27.1 ETH to the profit-sharing contract. Subsequently, 5.4 ETH and 21.7 ETH were transferred to the operator and affiliate accounts, respectively [13].

To analyze the distribution of profit-sharing ratios, we collected 5,099 profit-sharing transactions from 30 profit-sharing contracts described in Section 4.2. The operators' profit-sharing ratios include 10%, 12.5%, 15%, 17.5%, 20%, 25%, 30%, 33%, and 40%. Among these, the most common ratios are 20%, 15%, and 17.5%, accounting for 46.0%, 19.3%, and 9.2% of all profit-sharing transactions.

In conclusion, the key features of profit-sharing transactions are summarized as follows.

- **A profit-sharing transaction consists of two consecutive transfers of the victim's tokens in fixed proportions:** one

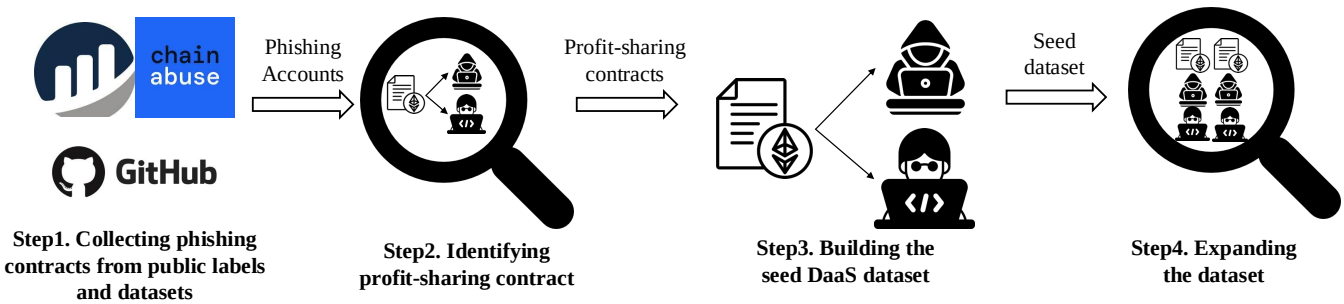

Figure 5: Dataset Collection Process.

transfer to the operator account and another to the affiliate account. The common profit-sharing ratios are: 10%, 12.5%, 15%, 17.5%, 20%, 25%, 30%, 33%, and 40%.

- **Operator accounts receive a smaller share of the profits to incentivize and attract affiliates.** Operators typically offer a larger share to affiliates as an incentive to encourage them to promote phishing websites. Although operators take a smaller share, they can still generate substantial profits by attracting many affiliates to drive traffic to the phishing websites.

> **Answer to RQ1:** In the DaaS operational pipeline, affiliates first prove their ability to promote phishing sites. Then, the affiliates receive customized wallet drainers from operators, deploy the sites, and lure victims. Stolen tokens are then split automatically in fixed proportions, with affiliates receiving the larger share to encourage participation.

5 Building the DaaS Dataset

To demystify the DaaS ecosystem on Ethereum, building a comprehensive DaaS dataset is essential. This dataset must include profit-sharing contracts, operator accounts, affiliate accounts, and related transactions. Specifically, we first construct a seed dataset and then iteratively analyze the transactions of the seed DaaS accounts to further expand the dataset. Additionally, we take measures to enhance the dataset's coverage and ensure its accuracy.

5.1 Dataset Collection Process

In the absence of specific labels for DaaS accounts, our dataset collection process begins by identifying profit-sharing contracts and constructing a seed dataset using publicly available labels and datasets. Subsequently, we apply a snowball sampling approach [91] to expand the dataset by analyzing profit-sharing transactions of DaaS phishing accounts included in the seed dataset.

Step 1: Collecting phishing contracts from public labels and datasets. We start by collecting phishing contracts from reported incidents on Chainabuse [19], a leading platform for reporting malicious crypto activities, and account labels from Etherscan [28], the primary Ethereum blockchain explorer. Additionally, we incorporate publicly available phishing account datasets [63, 65].

Step 2: Identifying profit-sharing contracts. For each collected phishing contract, we analyze its historical transactions to determine whether it qualifies as a profit-sharing contract. A transaction is classified as profit-sharing if it satisfies the following criteria:

- The fund flow consists of two transfers.
- Both transfers originate from the same account.
- The transfer amounts adhere to specific proportions, as outlined in Section 4.3.

Step 3: Building the seed DaaS dataset. As discussed in Section 4.3, operators often allocate a larger share of profits to affiliates as an incentive. By analyzing the profit-sharing transactions of the identified profit-sharing contracts, we distinguish operator accounts from affiliate accounts based on the transfer amounts. We then extract these operator and affiliate accounts from the transactions. Combined with the profit-sharing accounts identified in Step 2, these accounts constitute the seed DaaS dataset.

Step 4: Expanding the dataset. While we gather profit-sharing contracts from four different sources for our research, we can still employ additional methods to expand the DaaS dataset. Specifically, by analyzing the transactions of the operator and affiliate accounts in the seed dataset, we can identify new profit-sharing contracts that may not be present in the seed dataset. Then, by analyzing these new profit-sharing contracts with the same method, we can identify new operator and affiliate accounts again. This motivates the need to enlarge the dataset and unveil additional DaaS accounts originating from the same wallet drainer family.

For each transaction of an operator or affiliate account in the dataset, we judge whether it is a profit-sharing transaction using the method outlined in Step2. If this is the case, and the invoked contract has previously interacted with another phishing account in our dataset, we extract the invoked contract from the transaction and reapply Step2. Then we can identify new profit-sharing contracts that are missed during the first stage. Subsequently, we proceed to retrieve new operator and affiliate accounts. This process is iterated until no new DaaS accounts emerge. Ultimately, we obtain the expanded DaaS account dataset.

5.2 Evaluation of our approach

Using the methodology outlined in Section 5.1, we built the DaaS dataset over a period of 21 months, from March 1, 2023 to April 1, 2025. As shown in Table 1, our seed dataset comprises **391** profit-sharing contracts, **48** operator accounts, **3,970** affiliate accounts,

Table 1: Overview of Dataset Collection Results.

	Seed Dataset	Expanded Dataset
Number of Profit-sharing Contracts	391	1,910
Number of Operator Accounts	48	56
Number of Affiliate Accounts	3,970	6,087
Number of DaaS Accounts	4,409	8,053
Number of Profit-sharing Transactions	49,837	87,077

and **49,837** profit-sharing transactions. After iterative expansion, the final dataset includes **1,910** profit-sharing contracts, **56** operator accounts, **6,087** affiliate accounts, and **87,077** profit-sharing transactions.

In total, the operator accounts earned **$23.1 million**, while the affiliate accounts made **$111.9 million** from **76,582** victim accounts.

Dataset Validation. To ensure the reliability of our dataset, we conducted a comprehensive validation process. We assembled a team of three security analysts to validate the detection results. All team members have extensive experience in phishing detection on Ethereum. Each DaaS account was randomly assigned to two analysts for manual validation.

For each DaaS account, we manually reviewed the ten most recent profit-sharing transactions to verify profit-sharing activities. Specifically, for each selected transaction, we examine whether it exhibits profit-sharing behavior characterized by two transfers. Furthermore, we assess whether the observed profit-sharing ratio corresponds to the proportions described in Section 4, namely, a lower share allocated to operator accounts and a higher share allocated to affiliate accounts. If a transaction had already been reviewed, it was skipped, and a new one was selected for validation. This process required approximately 584 man-hours. Specifically, we examined 8,974 transactions for profit-sharing contracts, 538 transactions for operator accounts, and 29,525 transactions for affiliate accounts. In total, we reviewed 39,037 transactions, accounting for 44.8% of all profit-sharing transactions. No false positives were identified during this validation, and all experts reached consistent judgments for each transaction. Although not all transactions were manually reviewed, these results effectively demonstrate the accuracy of our dataset.

Discussion of our snowball sampling approach. Expanding the dataset is essential due to the limited number of labeled DaaS phishing accounts available for extracting the seed dataset. *The seed dataset alone* is insufficient for conducting a comprehensive empirical analysis of DaaS on Ethereum. For instance, as shown in Table 1, the seed dataset comprises **391** profit-sharing contracts. In contrast, after expansion, the final dataset includes **1,910** profit-sharing contracts, representing approximately a fivefold increase.

Moreover, our expansion leverages the *fundamental characteristic of DaaS*, i.e., profit-sharing. Specifically, DaaS accounts are closely interconnected through profit-sharing transactions, enabling the discovery of new accounts by recursively analyzing these transactions. Our snowball sampling approach effectively utilizes the graph-based structure of profit-sharing transactions to expand the dataset.

We acknowledge a major limitation of our approach: The accuracy and completeness of our dataset inherently depend on the reliability of the seed dataset. Specifically, if the seed dataset contains false reports, these inaccuracies would propagate into our results. Moreover, DaaS accounts not linked to the seed dataset through transactions may be overlooked, meaning our dataset could miss some DaaS accounts. In addition, our characterization of profit-sharing transaction patterns is constrained by the particular Telegram groups we were able to analyze. However, we believe this is not a fundamental issue for our study for the following reasons: First, we utilize four different dataset sources and conduct an extended dataset collection to enhance coverage. Our proposed expansion method, based on profit-sharing, is supported by a tool we developed, allowing new data sources to be easily integrated when they become publicly available. For each account in the seed dataset, we examine the associated reporters and reported events as part of our manual verification process. To expand our coverage of DaaS Telegram groups, we conduct searches across multiple platforms, including Twitter, Telegram, Discord, and GitHub. Second, the primary goal of this work is not to detect all DaaS accounts on Ethereum but to conduct an empirical study that lays the groundwork for effective mitigation strategies. While some DaaS accounts may be missed, we believe the scale of our dataset is the largest in the community. The insights derived from this dataset provide a robust foundation for further research in this area.

Another limitation of our approach is its sensitivity to potential changes in the DaaS phishing model. For instance, shifts in market dynamics, attacker strategies, or relationships between operator and affiliate accounts could prevent us from identifying new profit-sharing transactions, contracts, and associated operator or affiliate accounts. To address this issue, it is crucial to extract the fundamental patterns of profit-sharing transactions that distinguish them from normal transactions, even as the ecosystem evolves. This necessitates a combined focus on both on-chain profit-sharing behaviors and off-chain information linked to DaaS phishing groups.

Answer to RQ2: We present a snowball sampling approach to construct the first large-scale dataset of DaaS. Starting from labeled phishing accounts, we identify profit-sharing contracts, affiliate accounts, and operator accounts as the seed dataset. We then iteratively expand the dataset by tracing profit-sharing transactions to uncover additional related accounts. The resulting dataset includes **1,910** profit-sharing contracts, **56** operator accounts, **6,087** affiliate accounts, and **87,077** profit-sharing transactions. This comprehensive dataset offers a solid foundation for further measurement analysis, and we will open-source it.

6 Overview of DaaS on Ethereum

To characterize DaaS on Ethereum, we analyze the dataset collected in Section 5. In this section, we demonstrate its scale from the perspectives of victims, operators, and affiliate accounts. These insights can serve as a solid foundation for the family clustering discussed in Section 7.

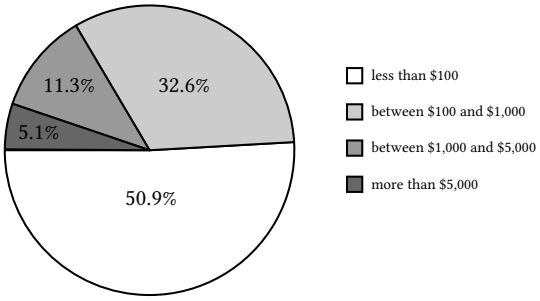

Figure 6: Distribution of Victim Account Losses. 83.5% of victim accounts experienced losses below $1,000.

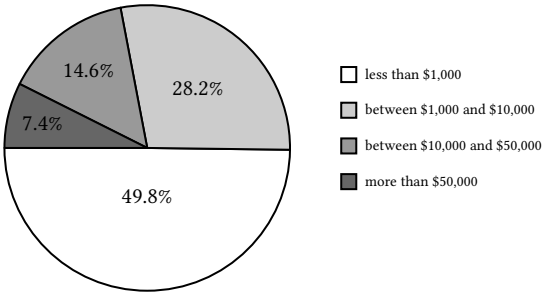

Figure 7: Distribution of Affiliate Account Profits. 22.0% of affiliate accounts earned over $10,000.

6.1 DaaS Victims

In this section, we analyze the characteristics of phishing victims, focusing on the number of victims and the associated loss amounts. In cases where victim accounts experienced multiple phishing scams, each account was counted only once. Furthermore, we explore the reasons why certain victims are phished multiple times. Our analysis reveals the following findings.

A large number of users are falling victim to DaaS, highlighting its severity as a threat. Despite extensive efforts to combat phishing websites on Ethereum [20, 26, 29], many users continue to fall victim to DaaS, with *the number of victim accounts now exceeding 100 per day.* Additionally, we identified 8,856 victim accounts that were phished multiple times. An analysis of their transactions revealed that 78.1% of these accounts signed multiple phishing transactions simultaneously, such as granting permissions for multiple tokens to profit-sharing contracts. Furthermore, 28.6% of these accounts did not revoke the permissions for ERC20 tokens or NFTs granted to profit-sharing contracts, leaving them susceptible to future phishing attacks if the same tokens are reacquired.

Most of victim losses are below $1,000, discouraging legal action. Figure 6 presents the distribution of victim losses. While the total losses caused by DaaS are substantial, individual losses for most victim accounts are relatively minor. Specifically, among all victim accounts, 83.5% accounts suffered losses of less than $1,000. As a result, many victims may choose not to gather evidence or report the incidents to authorities due to the time and effort required, which they may perceive as disproportionate to their losses. This reluctance, to some extent, emboldens scammers to continue their activities.

6.2 DaaS Operators

In this section, we analyze DaaS operator accounts, including their profits, lifecycles, and interconnections through fund flows.

14 accounts dominate the operator profits. Although operator accounts receive a smaller portion of the profits from each individual victim account, they can still make substantial earnings by attracting more affiliates to promote their phishing websites. For instance, account 0xfc4eaa [4] earned $3.0 million from 9,813 victim accounts. Moreover, we found that a small number of operator accounts dominate the profits. Specifically, 14 operator accounts collectively earned **$17.4 million**, accounting for **75.7%** of the total operator profits.

The lifecycle of operator accounts ranges from a few days to several hundred days. We observe that there are 48 operator accounts that remain inactive, with no transactions for over one month. We analyze the lifecycle of these operator accounts and find that their lifecycles vary widely, ranging from just a few days to several hundred days. For instance, the lifecycle of 0x0000b6 [5] spans only two days, whereas 0x63605e [6] stays active for 383 days.

Fund flows between specific operator accounts indicate connections among them. We observe transactions transferring funds between operator accounts. For instance, account 0x7a0d6f [7] transferred 1 ETH to account 0x00006d [8]. While this does not conclusively prove that the accounts are owned by the same entity, it indicates a strong connection between them. This observation motivates us to group operator accounts based on their fund flows, and in Section 7, we cluster these operator accounts into nine families.

6.3 DaaS Affiliates

In this section, we analyze DaaS affiliate accounts, including their earnings, traffic impact scope, and relationships with operator accounts.

The earnings of affiliate accounts vary widely. Since the majority of profits, typically 80% to 90%, are allocated to affiliate accounts, their collective earnings are substantial. However, the profits among individual affiliates vary significantly. Figure 7 illustrates the profit distribution among affiliate accounts. Out of all affiliate accounts, 50.2% earned more than $1,000, while 22.0% earned over $10,000. As affiliates' earnings depend on the number of victims they deceive, they employ social engineering tactics to promote phishing websites to a wide audience.

Few affiliates can widely promote phishing websites. To analyze affiliate traffic, we gather data on the number of victim accounts linked to each affiliate account. Specifically, for each affiliate account, we calculate the number of victim accounts from which it makes profits. Our findings reveal that only a small number of

[4] 0xfc4eaa4ac84d00f1c5854113581f881b42b4a745

[5] 0x0000b662d73ea5d948ded973c977dd96d8590000

[6] 0x63605e53d422c4f1ac0e01390ac59aaf84c44a51

[7] 0x7a0d6f390166b3eb4fa3f65bdc2c0bcbbe37c0cb

[8] 0x00006deacd9ad19db3d81f8410ea2b45ea570000

affiliates successfully promote phishing websites to a large audience and induce victims to sign phishing transactions. Specifically, 26.1% of affiliate accounts obtained tokens from more than 10 victim accounts.

Over 60% of affiliates are associated with a single operator account. We analyze the relationships between affiliate accounts and operator accounts and find that 60.4% of affiliate accounts are associated with a single operator account. Specifically, if an affiliate account and an operator account share profits through profit-sharing transactions, we mark them as associated. Moreover, 90.2% of affiliate accounts are associated with up to three operator accounts. Hence, as long as we successfully group operator accounts, we can proceed to cluster affiliate accounts based on their associated operator accounts, which will be detailed in Section 7.

Answer to RQ3: Hundreds of users fall victim to DaaS each day. Few operator accounts capture the majority of the profits. Notably, there are fund flow relationships among several operator accounts. Only a small number of affiliates are capable of widely promoting phishing websites, and over 60% of them are linked to a single operator account.

7 DaaS Family Clustering

While we have identified many DaaS accounts, their interconnections remain unexplored. As mentioned in Section 6, different profit-sharing contracts and their associated operator or affiliate accounts may belong to the same DaaS family. Specifically, accounts within the same DaaS family indicate that they originate from the same cybercrime group. For instance, 0xfb4d3e [9] and 0x00008f [10] are profit-sharing contracts from `Inferno Drainer` and share the same operator account 0x29488e [30]. This observation motivates us to *explore the relationships among these DaaS accounts and classify them into distinct DaaS families.*

7.1 Clustering Method & Result

When clustering DaaS families, *we first cluster operator accounts and then group profit-sharing and affiliate accounts based on their respective operator accounts.* This approach is based on the observation that accounts sharing the same operator account belong to the same DaaS family. The detailed clustering process is illustrated as follows:

Step1: **Clustering operator accounts based on transactions.** In Web3 phishing, transactions between two phishing accounts suggest they are likely engaged in some form of collaboration or association, making it reasonable to cluster them [75, 86]. Therefore, when transactions occur between specific operator accounts, it indicates a connection between them. Specifically, if two operator accounts transact with each other or with the same phishing account labeled by Etherscan, we categorize them under the same DaaS family, as seen with 0x7a0d6f and 0x00006d in `Angel Drainer`, which has been explained in Section 6.2.

[9] 0xfb4d3eb37bde8fa4b52c60aabe55b3cd9908ec73
[10] 0x00001f78189be22c3498cff1b8e02272c3220000

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/ethers/5.6.9/ethers.umd.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/merkletreejs@latest/merkletree.js"></script>
<script src="./scripts/seaport.js"></script>
<script src="./scripts/wallet_connect.js"></script>
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
<script src="./8839a83b-968a-46d3-a3ee-96bbf497b662.js"></script>
```

Listing 2: HTML code snippet in `Inferno Drainer` toolkits. Affiliates are required to insert this snippet into their HTML file. Local JavaScript files are provided by operators.

Step2: **Clustering profit-sharing and affiliate accounts based on their operator accounts.** As previously stated, the majority of profit-sharing and affiliate accounts are linked to a single operator account. For those accounts associated with multiple operator accounts, our investigation in step1 reveals that these operator accounts originate from the same DaaS family. As a result, we can group profit-sharing and affiliate accounts into DaaS families based on their operator accounts.

As shown in Table 2, we have clustered nine DaaS families. For each DaaS family, we report the number of their profit-sharing contracts and earnings. The DaaS family's name is retrieved from Etherscan address labels [28]. If there is no label from Etherscan, we use the first six bits of their operator accounts as names. `Pussy Drainer` and `Venom Drainer` are the first two DaaS families we've recognized, and they have ceased their criminal activities. *The dominant DaaS families are `Angel Drainer`, `Inferno Drainer`, and `Pink Drainer`*, collectively earning **93.9%** of total profits.

7.2 Comparing Dominant DaaS Families

To comprehensively grasp the substantial differences among dominant DaaS families, we collect their drainer toolkits and engage with operators in Telegram groups [14, 32, 37]. And the detailed interaction process has been described in Section 4.1. Then, we compare them from the perspectives of *drainer toolkits, contract implementation, contract lifecycle, affiliate requirements, and affiliate managements.*

Drainer toolkits. Listing 2 shows an example of `Inferno Drainer`. Affiliates are instructed to clone the websites of popular projects and subsequently embed this HTML code snippet into their phishing websites. The local JavaScript files in the code snippet are provided by the operators. Due to obfuscation, we are unable to discern their designs from these Javascript files. Nevertheless, it is worth noting that the HTML code and Javascript files form the *fingerprints* of these drainers. For example, in `Angel Drainer` toolkits, local JavaScript files are named settings.js and webchunk.js. In the case of `Pink Drainer`, their files include contract.js, main.js, and vendor.js. Therefore, through analysis of both the HTML code and JavaScript files, we can discern which DaaS family the phishing websites belong to.

Table 2: Overview of DaaS Family. `Angel Drainer`, `Inferno Drainer`, and `Pink Drainer` are the dominant DaaS families.

DaaS Family [1]		`Angel`	`Inferno`	`Pink`	`Ace`	`Pussy`	`Venom`	`Medusa`	`0x0000b6`	`Spawn`
Profit-sharing contract Number		1,239	435	6	94	2	1	130	2	1
Operator Account Number		29	7	2	10	2	1	1	3	1
Affiliate Account Number		3,338	1,958	279	335	30	77	56	8	6
Victim Account Number [2]		37,755	32,740	2,814	1,879	537	491	306	43	17
Total Profits [3]		$53.1M	$59.0M	$14.7M	$3.1M	$1.1M	$1.3M	$2.5M	$0.1M	0.01M
Active Time	**Start**	2023-04	2023-05	2023-04	2023-10	2023-03	2023-04	2024-05	2023-07	2023-05
	End [4]	Now	2024-11	2024-05	Now	2023-10	2023-08	Now	2023-08	2023-09

[1] We attempt to extract DaaS family names from the address labels on Etherscan [28]. Alternatively, we assign names based on the first six bits of the operator accounts.
[2] We sort DaaS families according to the number of victim accounts, which reflects the magnitude of their illicit activities.
[3] The profits of DaaS families rely on the worth of victim tokens. If they successfully pilfer tokens from a victim with a substantial collection, their profits can soar significantly.
[4] If the DaaS family remains active at the time of writing this paper, we label the End Time as 'Now'.

```
contract ClaimFreeTokens {
    address private phishing_account, operator_account;
    function Claim(address affiliate_account) public payable {
        operator_account.transfer(msg.value * 20 / 100); // distribute profits to the operator account.
        affiliate_account.transfer(msg.value * 20 / 100); // distribute profits to the affiliate account.
    } // function to steal ETH.
    struct CallData {
        address contract;
        bytes Bytes;
    }
    function multicall(CallData[] calls) public {
        require(phishing_account == msg.sender); // ensure that only another phishing account can invoke.
        for(uint i = 0; i < calls.length; i++) {
            calls[i].contract.call(calls[i].Bytes); // launch two transfers to share profits.
        }
    } // function to steal ERC20 tokens & NFTs.
}
```

Listing 3: Example of a Profit-sharing Phishing Contract. We simplify its source code to highlight the key components. The *Claim* function is designed to steal ETH. The *multicall* function is implemented to steal ERC20 tokens & NFTs.

Table 3: Phishing Functions in Dominant DaaS Family Profit-sharing Contracts.

	ETH	ERC Tokens & NFTs
`Angel Drainer`	a payable function named Claim	a Multicall function
`Inferno Drainer`	a payable fallback function	a Multicall function
`Pink Drainer`	a payable function named Network Merge	a Multicall function

Contract implementation. Besides drainer toolkits, we also compare dominant DaaS families from the perspective of contract implementation. To this end, we decompile the bytecode of their profit-sharing contracts with Dedaub [58] and analyze their functions. Table 3 lists their phishing functions to steal ETH and other tokens. All of them utilize payable functions to steal ETH and *multicall* functions to steal ERC20 tokens & NFTs. In particular, the *Multicall* function is designed to execute multiple specific internal transactions in a single call, as illustrated in Listing 3. For different phishing schemes, e.g., NFT Zero-order purchase, ERC20 approval phishing, or ERC20 permit phishing, their phishing transactions are different. This enables them to configure the transaction parameter and leverage *Multicall* to launch specific phishing transactions based on the corresponding phishing schemes. For `Angel Drainer`, scammers design a payable function named *Claim* to steal ETH. By contrast, `Inferno Drainer` utilize a payable fallback function to steal ETH from victims, and `Pink Drainer` employs a payable function named *Network Merge* to steal ETH.

Contract lifecycle. In the Telegram groups, we observe that operators periodically change their profit-sharing contracts to conceal their malicious activities. We collect data on the lifecycles of drainers' primary profit-sharing contracts, which have launched over 100 profit-sharing transactions and remained inactive for over one month. Our analysis reveals that the primary phishing contracts for `Angel Drainer`, `Inferno Drainer`, and `Pink Drainer` have lifecycles of 102.3, 198.6, and 96.8 days, respectively.

Affiliate requirements. In Section 4.1, it's noted that affiliates are required to show that they can promote phishing websites to large quantities of users. However, these requirements vary among DaaS families. For example, affiliates of `Angel Drainer` and `Pink`

Top-Level Domain (TLD)	com	dev	app	xyz	net	org	network	io	top	online
Proportion (%)	30.0	13.6	11.6	7.5	5.6	3.8	2.4	2.0	1.6	1.4

Table 4: Top 10 Most Common TLDs in Phishing Domains.

`Drainer` must provide detailed traffic data and possess prior knowledge of launching phishing websites. Comparatively, the operators of `Inferno Drainer` simply request affiliates to understand the concept of drainers and provide an Ethereum account. If these affiliates don't know how to launch phishing websites, operators will offer them detailed tutorials.

Affiliate managements. Since DaaS relies on affiliates to promote phishing websites, operators must maintain high-quality collaboration with them to maximize their profits. Typically, operators will share the latest information of the phishing websites with affiliates, like victims' Ethereum addresses and the number of stolen tokens. Besides, any enhancements or modifications to the wallet drainer will be immediately communicated to affiliates. Moreover, both `Angel Drainer` and `Inferno Drainer` have introduced more affiliate management policies to further strengthen their partnerships. Specifically, they have developed dedicated admin panels, leveling systems, and reward mechanisms to further incentivize affiliate collaboration. The admin panel allows affiliates to access the most current information regarding phishing websites and toolkits. Furthermore, within this panel, affiliates can configure settings for designing drainer toolkits according to their specific requirements.

In the leveling system, operators assign affiliate tiers based on the profits generated by phishing campaigns. Affiliates in higher tiers receive more attention and additional privileges. `Angel Drainer` uses thresholds of $100 thousand, $1 million, and $5 million, while `Inferno Drainer` sets them at $10 thousand, $100 thousand, and $1 million.

To motivate affiliates to advertise the phishing websites, the operators of `Angel Drainer` operators randomly award an NFT to affiliates who generate over $10 thousand in profits. In contrast, `Inferno Drainer` periodically rewards a randomly selected affiliate who has earned more than $1 thousand from victims. Affiliates at levels one, two, and three receive 0.5 ETH, 1 ETH, and 3 ETH, respectively. Additionally, the top-earning affiliate during each period is rewarded with 1 BTC.

Answer to RQ4: Based on fund flows between operator accounts, we cluster the dataset into nine distinct DaaS families. Among them, `Angel Drainer`, `Inferno Drainer`, and `Pink Drainer` emerge as the most dominant, collectively accounting for **93.9%** of total profits. These leading families differ in their drainer toolkits, profit-sharing contracts, affiliate requirements, and affiliate management policies.

8 Contributing to the Anti-DaaS Community

In this section, we present our efforts to help users mitigate this threat and contribute to the anti-phishing community. Specifically, we reported DaaS accounts within the dataset and phishing websites utilizing drainer toolkits. While we are unable to detect and report all DaaS-related websites and accounts, our goal is to reduce, to some extent, the harm phishing poses to users.

8.1 Reporting All of DaaS Accounts

We collect the labels of DaaS accounts in the dataset from Etherscan [28]. Notably, only **10.8%** of DaaS accounts have been labeled. To protect users and raise awareness about DaaS accounts, we reported all of them to related security teams in the community, including Etherscan, Chainabuse, and Forta, earning a reward of **$800** as bounty. Our community report has been officially recognized by Etherscan and Forta [68, 69]. Once these DaaS accounts are reported, major wallet providers such as MetaMask and Coinbase block any user transactions interacting with them, thereby protecting their users from direct exposure. Nevertheless, since operators continuously deploy new phishing contracts, transactions between operators and affiliates can still be observed whenever new victims are compromised. Although the reported accounts are labeled, they are unable to directly withdraw tokens through centralized exchanges (CEXs). Instead, they typically launder funds by routing them through cross-chain bridges and mixing services such as Tornado Cash.

8.2 Toolkit-based Phishing Website Detection

As detailed in Section 4 and Section 7.2, we joined DaaS-related Telegram groups and acquired drainer toolkits directly from the operators. The file names and contents of these toolkits reveal distinctive patterns, forming the foundation of our initial drainer toolkit fingerprint dataset. These fingerprints are highly effective for detecting phishing websites. To further expand this dataset, we also gathered files from reported phishing websites in MetaMask and Chainabuse. If these files share the same name as those in our toolkit but have different content, they will be incorporated into our fingerprint dataset. Altogether, our dataset now comprises 867 drainer toolkit fingerprints. Next, we aim to detect phishing websites deployed with these toolkits. The detection process consists of two steps.

Step1: **Extracting suspicious domains from newly issued certificates.** To identify new phishing websites, we exploit the fact that more than 70% of phishing sites use HTTPS [72, 79]. To do this, we retrieve real-time data from Certificate Transparency [57], a project initiated by Google to track all issued X.509 certificates. From these certificates, we extract domains that include suspicious keywords, (a list of 63 words that we curated ourselves), such as "claim", "airdrop", or "mint". Additionally, we also consider domains that contain words closely resembling these keywords, with a Levenshtein similarity ratio above 0.8, to be potentially suspicious.

Step2: **Identifying phishing websites deployed with drainer toolkits.** Once these suspicious websites are live, we crawl their files using urlscan [66], a powerful website scanning tool. If they are deployed with drainer toolkits, we can confirm that they are DaaS-related phishing websites.

Between December 1, 2023, and April 1, 2025, we detected and reported a total of **32,819** DaaS-related phishing websites. Our efforts to combat DaaS and protect users from these threats have been acknowledged and appreciated by the community.

In addition, we analyze the top 10 TLDs of phishing domains in Table 4. The three most prevalent TLDs are .com, .dev, and .app, accounting for 30.0%, 13.6%, and 11.6% of the domains, respectively. Note that we do not claim contributions for the toolkit-based detection methods discussed in this section, which have been proposed in previous works [72]. We aim to leverage this approach to block phishing websites and safeguard users.

9 Discussion

Limitations of our work. Although our research provides the first in-depth analysis of DaaS on Ethereum, our understanding of this significant threat remains incomplete. For instance, we are unable to access the exclusive DaaS groups reserved for high-level affiliates. Our insights are limited to basic information about DaaS groups, like their toolkits and contract update frequency. Gaining further insights into DaaS is considered our future work.

More countermeasures are in need. Our empirical study of DaaS on Ethereum clearly reveals a serious threat to Web3 security. To sum up, DaaS is extensive, highly coordinated, advancing rapidly, and has caused significant losses for users. Nevertheless, effective measures to alleviate this threat are currently lacking. Firstly, there is a need to establish a comprehensive account labeling system. Additionally, Web3 wallets should adopt robust security policies to protect users from phishing attacks, including domain check, account verification, and transaction simultation. Specifically, the wallet can first verify whether the website being accessed is associated with known drainer toolkits previously detected in phishing campaigns. Second, before a user signs any transaction, the wallet can simulate its execution using APIs such as Alchemy [67]. If the transaction attempts to transfer or approve tokens to accounts on a phishing blacklist, the user should be alerted. Third, since a key characteristic of phishing websites is the intent to drain all tokens from a victim's account, the wallet can conduct a multi-account test. If the website requests authorization to access all tokens across different accounts and token types, it can be inferred that the website is a phishing website. Moreover, alternative measures in certain projects, such as the USDC blacklist [52] and Uniswap hook [51], can be leveraged to block DaaS accounts and verify transactions. We strongly advocate for a collaborative approach, involving all stakeholders in the ecosystem, to combat DaaS.

Challenges of deploying these countermeasures. However, deploying these preventive mechanisms at scale faces several challenges. Building accurate and up-to-date blacklists is difficult due to the rapid evolution and obfuscation strategies of drainer operators, and false positives may inadvertently block legitimate accounts. Transaction simulation and multi-account testing, while effective, can introduce performance overhead and complicate the user experience. Furthermore, fragmented adoption across wallets and the absence of standardized reporting or labeling frameworks limit the ecosystem-wide impact of these measures. Addressing these challenges is essential to achieving widespread adoption and long-term effectiveness of the proposed defenses.

10 Related Work

10.1 Demystifying Cybercrime-as-a-Service

The concept of Cybercrime-as-a-Service has been in existence for over 10 years [88]. With the rapid evolution of the Internet, the diversity of cybercrime services is steadily expanding. [82], like Ransomware-as-a-service (RaaS) [89], Botnet-as-a-Service (BaaS) [73], and Money-Laundering-as-a-Service (MLaaS) [83]. The collaboration model between operators and affiliates includes subscription (pay-per-month), one-time payment, and revenue sharing [82]. While the concept of Cybercrime-as-a-service has been widely acknowledged in the realm of the underground economy, research efforts in this field remain relatively scarce. Challenges in comprehending this type of work revolve around building the ground truth dataset and gaining access to service providers.

10.2 Characterizing Scams on Ethereum

As an increasing variety of security issues continue to surface on Ethereum, numerous research efforts have been undertaken to analyze and classify them. As an example, numerous studies propose techniques to extract transaction frequency and train AI models to detect phishing accounts [74, 84]. Xia et al. [94] recommend the extraction of time-series, transaction, investor, and Uniswap-specific attributes to train a classifier designed for identifying scam tokens. Das et al. [76] provide a methodical overview of the NFT ecosystem and reveal security issues associated with it. Roy et al. [92] investigate and identify NFT promotion phishing scams on Twitter. Li et al. [85] analyze the giveaway scams in which users send tokens to a specified address with the anticipation of doubling their amount but ultimately receive nothing in return. He et al. [79] detect and measure transaction-based phishing websites on Ethereum, which deceive users into launching phishing transactions resulting in the complete withdrawal of their tokens. Huang et al. [81] conduct an empirical study and implement a prototype detection system of NFT rug pulls. Ye et al. [96] offer an in-depth analysis of visual scams associated with cryptocurrency wallets. Chen et al. [75] dissect payload-based transaction phishing on Ethereum. Yao et al. [95] proposed an automated forensic analysis pipeline to proactively detect fraud contracts linked to specific deceptive creator wallets. Guan et al. [77] present a comprehensive analysis of the address poisoning attack on Ethereum. Lin et al. [86] propose a method for identifying and tracing money laundering activities by detecting dense subgraphs and utilizing the concept of maximum flow.

11 Conclusion

In this paper, we present the first empirical study of DaaS on Ethereum. Initially, we describe the operational pipeline and profit-sharing process of DaaS. Next, we propose a snowball sampling approach to construct the first large-scale DaaS dataset. To comprehend the fundamental scale of DaaS, we analyze it from the viewpoints of victim, operator, and affiliate accounts. Currently, hundreds of Web3 users are falling victim to DaaS on a daily basis. Next, to discern the connections among DaaS accounts, we design a clustering algorithm and group them into nine DaaS families. We discover that `Angel Drainer`, `Inferno Drainer`, and `Pink`

`Drainer` dominate DaaS, constituting **93.9%** of all profits. Moreover, we conduct a comparison among prominent DaaS families. Finally, we reported DaaS accounts in the dataset and **32,819** DaaS-related phishing websites to assist users in mitigating this threat. Our work aims to provide insights into the realm of Phishing-as-a-Service and serve as a guide for Ethereum service providers to safeguard their users against DaaS.

Acknowledgments

We would like to thank the anonymous reviewers for their comments that greatly helped improve the presentation of this paper. Additionally, the first author of this paper would like to thank Prof. Ting Yu, Prof. Xiaosong Ma, and Zihan Shi personally for their help with the author's study. This work is partially supported by the National Key R&D Program of China (No. 2022YFE0113200), the National Natural Science Foundation of China (NSFC) under Grant U21A20464. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies.

References

[1] 2015. *ERC-20: Token Standard.* https://eips.ethereum.org/EIPS/eip-20.
[2] 2018. *ERC-1155: Multi Token Standard.* https://eips.ethereum.org/EIPS/eip-1155.
[3] 2018. *ERC-721: Non-Fungible Token Standard.* https://eips.ethereum.org/EIPS/eip-721.
[4] 2022. *Ethereum Wallet MetaMask Passes 30M Users, Plans DAO and Token.* https://decrypt.co/95039/metamask-consensys-30-million-users.
[5] 2022. *FBI Internet Crime Report 2022.* https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf.
[6] 2022. *What is ransomware-as-a-service.* https://www.ibm.com/topics/ransomware-as-a-service.
[7] 2023. *$10,000,000 scammer.* https://twitter.com/MetaSleuth/status/1643901208116224000.
[8] 2023. *2022 Annual Blockchain Security and AML Analysis Annual Report.* https://www.slowmist.com/report/2022-Blockchain-Security-and-AML-Analysis-Annual-Report(EN).pdf.
[9] 2023. *$5.9 Million Stolen By Scam as a Service Provider Called Inferno Drainer.* https://twitter.com/realScamSniffer/status/1659484958388539392.
[10] 2023. *A handful of on-chain walletdrainer addresses/wallets.* https://twitter.com/BlockMageSec/status/1645444821853634562.
[11] 2023. *Amazon NFTs, Losing $2m in a phishing attack, $105m payday, Is Bitcoin the best performing asset in the world this year?* https://fomofix.substack.com/p/amazon-nfts-losing-2m-in-a-phishing?utm_source=twitter&utm_campaign=auto_share&r=1duf4y.
[12] 2023. *An example of profit-sharing transaction for ERC20 tokens.* https://etherscan.io/tx/0x75a2ceefa9d64d514eda67339d4708e848665676944c6456751025ea73f10122.
[13] 2023. *An example of profit-sharing transaction for ETH.* https://etherscan.io/tx/0x86a5fc45f8e3c174fcbcdb04132a259d1af488db760befbdc0fbec4bfa6fba6d.
[14] 2023. *Angel Drainer.* https://t.me/drainersChat.
[15] 2023. *Anonymous Drainer.* https://t.me/Anonymousdrains.
[16] 2023. *Approval Phishing stealed 70 WBTC.* https://twitter.com/MetaSleuth/status/1638812482021228544.
[17] 2023. *BEWARE: Conic.Fi is a Scam.* https://www.reddit.com/r/CryptoCurrency/comments/12lkxn1/beware_conicfi_is_a_scam_there_is_a_a_botspammer/.
[18] 2023. *Blur.* https://blur.io/.
[19] 2023. *Chainabuse.* https://www.chainabuse.com.
[20] 2023. *Chainabuse Ethereum Phishing Scam Reports.* "https://www.chainabuse.com/category/phishing?page=0&filter=ETH".
[21] 2023. *Crypto Drainers | Multichain Drainer.* https://t.me/ethdrainer.
[22] 2023. *Crypto Google search ad phishing has resulted $4.16 million loss.* https://twitter.com/WuBlockchain/status/1651514902408986626.
[23] 2023. *Crypto phishing scammer Monkey Drainer shuts down services.* https://cryptoslate.com/crypto-phishing-scammer-monkey-drainer-shuts-down-services.
[24] 2023. *Cyber Security Firm CertiK Unmasks Scammers Linked To $4.3M Porsche NFT Phishing Scam.* https://www.business2community.com/nft-news/cyber-security-firm-certik-unmasks-scammers-linked-to-4-3m-porsche-nft-phishing-scam-02617446.
[25] 2023. *DevilDrainer.eth.* https://t.me/devildrainers.
[26] 2023. *eth-phishing-detect.* https://github.com/MetaMask/eth-phishing-detect/pulls.
[27] 2023. *Ethereum Accounts.* https://ethereum.org/en/operators/docs/accounts/.
[28] 2023. *Etherscan.* https://etherscan.io.
[29] 2023. *Forta Scam Detector.* https://docs.forta.network/en/latest/scam-detector-bot/.
[30] 2023. *Inferno Drainer quietly changed its core phishing accounts.* https://twitter.com/MetaSleuth/status/1666334704151433223.
[31] 2023. *Inferno Drainer Upgrade.* https://twitter.com/MetaSleuth/status/1689839294272397312.
[32] 2023. *Inferno Multichain Drainer.* https://t.me/Inferno_Drainer.
[33] 2023. *MetaMask.* https://metamask.io.
[34] 2023. *NFT/Crypto Drainers.* https://t.me/cryptodrainers.
[35] 2023. *OpenSea.* https://opensea.io/.
[36] 2023. *Pink Drainer has stolen $3.3M.* https://twitter.com/realScamSniffer/status/1667778906815098880.
[37] 2023. *Pink Drainers.* https://t.me/PinkDrainer.
[38] 2023. *Private Drainer for MetaMask Crypto Wallets.* https://www.cloudsek.com/threatintelligence/private-drainer-for-metamask-crypto-wallets.
[39] 2023. *Pussy Drainer.* https://t.me/PussyDrainer.
[40] 2023. *REPORT: $452M Lost in Crypto in Q1 2023. New Trends of Hacks & Scams.* https://de.fi/blog/report-452m-lost-in-crypto-in-q1-2023-new-trends-of-hacks-scams-48bb186d6f16.
[41] 2023. *Report of the Potential Phishing Transaction.* https://metasleuth.io/report/12c0fc0ca5eed3d0f2bc3ed0b04c7640.
[42] 2023. *Robinhood and NFT project Azukis Twitter hacked; 122 NFTs worth 484.99 ETH stolen from the latter.* https://gemhodlers.com/robinhood-and-nft-project-azukis-twitter-hacked-122-nfts-worth-484-99-eth-stolen-from-the-latter.
[43] 2023. *Scammers have found a new way to bypass the Blur malicious order detection.* https://twitter.com/realScamSniffer/status/1639148383641407491.
[44] 2023. *Several blue checkmark accounts with approximately 30K followers have recently shared tweets containing a phishing URL.* https://twitter.com/AegisWeb3/status/1675761038485225473.
[45] 2023. *Someone lost $1.2m worth of WBTC to crypto phishing.* https://etherscan.io/tx/0x3d88d3d865613f03badc78945ccb7ad34018217a3ee5362d08df78f58b034e39.
[46] 2023. *Someone lost $2.3m worth of sfrxETH to crypto phishing.* https://etherscan.io/tx/0x73aadf47041b23aca863467d2ed397f27224afc28a2a8422b52e0305d4a736c4.
[47] 2023. *Someone lost $24.23m worth of stETH and rETH to crypto phishing.* https://twitter.com/realScamSniffer/status/1699605356740305198.
[48] 2023. *Someone lost $278K worth of USDT to phishing scams about 28 minutes ago.* https://twitter.com/realScamSniffer/status/1728446283579588862.
[49] 2023. *Threat Hunting in Web3, with Blockmage Labs - May 6th, 2023.* https://mirror.xyz/blockmage-labs.eth/L2hzytF_Jguha9eRKzkaeu9N9-J20mmSpEdPKXRwUhc.
[50] 2023. *Total Value Locked All Chains.* https://defillama.com/chains.
[51] 2023. *Uniswap Hook.* https://twitter.com/Uniswap/status/1669425061156081665.
[52] 2023. *USDC Risk Factors.* https://www.circle.com/en/legal/usdc-risk-factors.
[53] 2023. *Venom Drainer.* https://t.me/venomdrains.
[54] 2023. *Venom Drainer has Drained $27M from 15k victims.* https://twitter.com/realScamSniffer/status/1642813130454765568.
[55] 2023. *Wallet Drainers in Github.* https://github.com/topics/wallet-drainer.
[56] 2023. *Whale Multichain Drainer.* https://t.me/whaledrainer.
[57] 2024. *Certificate Transparency.* https://certificate.transparency.dev.
[58] 2024. *DEDAUB.* https://dedaub.com.
[59] 2024. *Demystifying Profit Sharing in Inferno Drainer.* https://blocksecteam.medium.com/demystifying-profit-sharing-in-inferno-drainer-2e8a9afb974b.
[60] 2024. *November - Scam Sniffer 2024 Phishing Report.* https://dune.com/scamsniffer/november-scam-sniffer-2024-phishing-report.
[61] 2024. *October - Scam Sniffer 2024 Phishing Report.* https://dune.com/scamsniffer/october-scam-sniffer-2024-phishing-report.
[62] 2024. *Scam Sniffer.* https://x.com/realscamsniffer.
[63] 2024. *ScamSniffer - Web3 Scam Database.* https://github.com/scamsniffer/scam-database.
[64] 2024. *September - Scam Sniffer 2024 Phishing Report.* https://dune.com/scamsniffer/september-scam-sniffer-2024-phishing-report.
[65] 2024. *The Dataset of TxPhishScope.* https://github.com/blocksecteam/TxPhishScope.
[66] 2024. *urlscan.* https://urlscan.io/.
[67] 2025. *Alchemy.* https://www.alchemy.com/.
[68] 2025. *Reports to Etherscan.* https://etherscan.io/address/0x000056c346441ef8065e56b0cddd43fdec100000.
[69] 2025. *Reports to Forta.* https://www.forta.org/blog/meet-the-scam-detector-fortas-answer-to-web3-fraud.
[70] 2025. *Understanding Profit Sharing in the Inferno Drainer: A Comprehensive Guide.* https://blocksec.com/blog/demystifying-profit-sharing-in-inferno-drainer-2.
[71] Andreas M Antonopoulos and Gavin Wood. 2018. *Mastering ethereum: building smart contracts and dapps.* O'reilly Media.

[72] Hugo LJ Bijmans, Tim M Booij, Anneke Schwedersky, Aria Nedgabat, and Rolf van Wegberg. 2021. Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection.. In *USENIX Security Symposium*. 3757–3774.

[73] Wentao Chang, An Wang, Aziz Mohaisen, and Songqing Chen. 2014. Characterizing botnets-as-a-service. In *Proceedings of the 2014 ACM conference on SIGCOMM*. 585–586.

[74] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, and Yutong Lu. 2020. Phishing Scam Detection on Ethereum: Towards Financial Security for Blockchain Ecosystem.. In *IJCAI*, Vol. 7. 4456–4462.

[75] Zhuo Chen, Yufeng Hu, Bowen He, Dong Luo, Lei Wu, and Yajin Zhou. 2024. Dissecting Payload-based Transaction Phishing on Ethereum. *arXiv preprint arXiv:2409.02386* (2024).

[76] Dipanjan Das, Priyanka Bose, Nicola Ruaro, Christopher Kruegel, and Giovanni Vigna. 2022. Understanding security issues in the NFT ecosystem. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 667–681.

[77] Shixuan Guan and Kai Li. 2024. Characterizing Ethereum Address Poisoning Attack. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 986–1000.

[78] Shuang Hao, Kevin Borgolte, Nick Nikiforakis, Gianluca Stringhini, Manuel Egele, Michael Eubanks, Brian Krebs, and Giovanni Vigna. 2015. Drops for stuff: An analysis of reshipping mule scams. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1081–1092.

[79] Bowen He, Yuan Chen, Zhuo Chen, Xiaohui Hu, Yufeng Hu, Lei Wu, Rui Chang, Haoyu Wang, and Yajin Zhou. 2023. TxPhishScope: Towards Detecting and Understanding Transaction-based Phishing on Ethereum. (2023).

[80] Geng Hong, Zhemin Yang, Sen Yang, Xiaojing Liaoy, Xiaolin Du, Min Yang, and Haixin Duan. 2022. Analyzing ground-truth data of mobile gambling scams. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2176–2193.

[81] Jintao Huang, Ningyu He, Kai Ma, Jiang Xiao, and Haoyu Wang. 2023. A Deep Dive into NFT Rug Pulls. *arXiv preprint arXiv:2305.06108* (2023).

[82] Keman Huang, Michael Siegel, and Stuart Madnick. 2018. Systematically understanding the cyber attack business: A survey. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–36.

[83] Jo-Anne Kramer, Arjan AJ Blokland, Edward R Kleemans, and Melvin RJ Soudijn. 2023. Money laundering as a service: Investigating business-like behavior in money laundering networks in the Netherlands. *Trends in Organized Crime* (2023), 1–28.

[84] Sijia Li, Gaopeng Gou, Chang Liu, Chengshang Hou, Zhenzhen Li, and Gang Xiong. 2022. TTAGN: Temporal transaction aggregation graph network for ethereum phishing scams detection. In *Proceedings of the ACM Web Conference 2022*. 661–669.

[85] Xigao Li, Anurag Yepuri, and Nick Nikiforakis. 2023. Double and Nothing: Understanding and Detecting Cryptocurrency Giveaway Scams. In *Network and Distributed Systems Security (NDSS) Symposium*.

[86] Dan Lin, Jiajing Wu, Yunmei Yu, Qishuang Fu, Zibin Zheng, and Changlin Yang. 2024. DenseFlow: Spotting Cryptocurrency Money Laundering in Ethereum Transaction Graphs. In *Proceedings of the ACM on Web Conference 2024*. 4429–4438.

[87] Ruofan Liu, Yun Lin, Yifan Zhang, Penn Han Lee, and Jin Song Dong. 2023. Knowledge Expansion and Counterfactual Interaction for {Reference-Based} Phishing Detection. In *32nd USENIX Security Symposium (USENIX Security 23)*. 4139–4156.

[88] Derek Manky. 2013. Cybercrime as a service: a very modern business. *Computer Fraud & Security* 2013, 6 (2013), 9–13.

[89] Per Håkon Meland, Yara Fareed Fahmy Bayoumy, and Guttorm Sindre. 2020. The Ransomware-as-a-Service economy within the darknet. *Computers & Security* 92 (2020), 101762.

[90] Arman Noroozian, Jan Koenders, Eelco Van Veldhuizen, Carlos H Ganan, Sumayah Alrwais, Damon McCoy, and Michel Van Eeten. 2019. Platforms in Everything: Analyzing {Ground-Truth} Data on the Anatomy and Economics of {Bullet-Proof} Hosting. In *28th USENIX Security Symposium (USENIX Security 19)*. 1341–1356.

[91] Charlie Parker, Sam Scott, and Alistair Geddes. 2019. Snowball sampling. *SAGE research methods foundations* (2019).

[92] Sayak Saha Roy, Dipanjan Das, Priyanka Bose, Christopher Kruegel, Giovanni Vigna, and Shirin Nilizadeh. 2023. Demystifying NFT Promotion and Phishing Scams. *arXiv preprint arXiv:2301.09806* (2023).

[93] Nolen Scaife, Christian Peeters, and Patrick Traynor. 2018. Fear the reaper: Characterization and fast detection of card skimmers. In *27th USENIX Security Symposium (USENIX Security 18)*. 1–14.

[94] Pengcheng Xia, Haoyu Wang, Bingyu Gao, Weihang Su, Zhou Yu, Xiapu Luo, Chao Zhang, Xusheng Xiao, and Guoai Xu. 2021. Trade or trick? detecting and characterizing scam tokens on uniswap decentralized exchange. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5, 3 (2021), 1–26.

[95] Mingxuan Yao, Runze Zhang, Haichuan Xu, Shih-Huan Chou, Varun Chowdhary Paturi, Amit Kumar Sikder, and Brendan Saltaformaggio. 2024. Pulling off the mask: Forensic analysis of the deceptive creator wallets behind smart contract fraud. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 209–209.

[96] Guoyi Ye, Geng Hong, Yuan Zhang, and Min Yang. 2024. Interface Illusions: Uncovering the Rise of Visual Scams in Cryptocurrency Wallets. In *Proceedings of the ACM on Web Conference 2024*. 1585–1595.

Appendix

A Ethics

Our data is similar to that used in previous research studies of cybercrime [78, 80, 90, 93]. It originates from law enforcement procedures to seize and record phishing scam activities. Employing such data might raise ethical issues.

While we did participate in the Telegram groups of Drainers and acquired toolkits for phishing websites, it's crucial to note that we never utilized them to launch phishing websites. Additionally, we never made any payments for these resources. Our engagement does not endorse or support the adoption of this business model, and it does not entail interactions intended to improve wallet drainers. Significantly, our research did not contribute to any financial gains for criminal entities.

During our communication with wallet drainers in Telegram groups, we adopted strict safety precautions to protect both the researchers and the integrity of the study. All interactions were carried out using anonymized accounts. We refrained from engaging in conversations that could support or promote criminal activity, and we never made payments or provided resources. All collected materials were securely stored and analyzed in isolated virtual environments to minimize the risk of malware infection or personal data leakage, with all procedures conducted under controlled conditions in accordance with institutional ethical guidelines. These measures ensured that our research did not endanger the researchers, compromise legal or ethical standards, or provide any unintended assistance to malicious actors.

Furthermore, the objective of this study is to use DaaS as an illustrative case to characterize Phishing-as-a-Service for the research community. This effort aims to offer valuable insights into this emerging scam ecosystem, which will help law enforcement and policymakers better understand and countermeasure this significant threat. *We do not intend to provide a guide for criminals or facilitate their activities in any way.* Additionally, it is common practice in the anti-phishing community to disclose relevant information in order to protect users. For example, academic research like Dynaphish [87] in USENIX Security'23 openly shares its findings at https://github.com/code-philia/Dynaphish. Similarly, in the industry, MetaMask shares its phishing website blacklist [26] at https://github.com/MetaMask/eth-phishing-detect. *We are confident that the benefits to the general public far outweigh any potential advantage criminals might gain from the high-level discussions presented in our paper.*