# CtPhishCapture: Uncovering Credential-Theft-Based Phishing Scams Targeting Cryptocurrency Wallets

Hui Jiang[†‡], Zhenrui Zhang[‡], Xiang Li[§], Yan Li[†],
Anpeng Zhou[†], Chenghui Wu[‡], Man Hou[¶✉], Jia Zhang[†], Zongpeng Li[†✉]
[†]Tsinghua University, [‡]Baidu Inc., [§]Nankai University, [¶]Zhongguancun Laboratory
Email: {jiang-h23, li-yan21, zap23}@mails.tsinghua.edu.cn; {zhangzhenrui, wuchenghui}@baidu.com;
lixiang@nankai.edu.cn; houman@zgclab.edu.cn; {zhangjia2017, zongpeng}@tsinghua.edu.cn

***Abstract*—Due to substantial financial incentives, credential-theft-based cryptocurrency wallet phishing (CtPhish) scams have become among the most pervasive threats in the cryptocurrency ecosystem. In such attacks, victims are deceived into visiting CtPhish websites or applications and divulging their credentials, enabling attackers to steal cryptocurrency assets. Although various phishing detection approaches exist, they are often ineffective for CtPhish or suffer from significant limitations.**

**To bridge this gap, we propose CtPhishCapture, a large-scale detection system designed to identify CtPhish websites and applications. CtPhishCapture actively visits suspicious websites, employs large language model (LLM)-based detection methods to identify CtPhish websites, and downloads and analyzes potential CtPhish applications for further verification. During a six-month deployment, CtPhishCapture detected 5,138 CtPhish websites and 10,612 CtPhish applications. Notably, only 17% of the websites and 21% of the applications were previously reported by the community, indicating that CtPhishCapture newly discovered 83% of the websites and 79% of the applications—establishing it as the largest known CtPhish detection system to date.**

**Leveraging the collected dataset, we conduct a comprehensive end-to-end measurement and analysis of the CtPhish ecosystem. Our study reveals how attackers attract victims, build trust, and ultimately exfiltrate cryptocurrency assets. Additionally, we further characterize the associated websites and applications, examining their features, evasion strategies, and estimated financial impact. Finally, we deploy CtPhishCapture in collaboration with Baidu. By integrating its detection results, the weekly user complaints about CtPhish are reduced by a factor of 5.8.**

## I. Introduction

In recent years, the cryptocurrency ecosystem has experienced substantial expansion, driven by advancements in blockchain technology and the broader Web3 infrastructure. By mid-2024, the total market capitalization of cryptocurrencies reached USD 2.55 trillion. As a core component of this ecosystem, hundreds of cryptocurrency wallets have emerged, enabling the secure storage and exchange of digital assets.

However, the substantial economic incentives have also fostered the proliferation of credential-theft-based cryptocurrency wallet phishing (CtPhish) scams, now among the most pervasive malicious activities in the cryptocurrency domain. Between 2017 and 2025, we observed a sharp increase in CtPhish websites and applications, most of which exfiltrate victims' digital assets through credential theft [1]–[6]. Notably, documented incidents involving counterfeit versions of Curve Wallet [7], Rabby Wallet [8], and Ledger Wallet [9] collectively caused at least USD 120,000 in financial losses, underscoring the widespread prevalence and severe impact of CtPhish scams within the cryptocurrency ecosystem.

**CtPhish.** Our analysis of reported cases reveals that CtPhish attacks typically proceed in three stages. First, attackers employ techniques such as black-hat SEO to increase the visibility of CtPhish websites in search engine results. Most of these websites directly offer download links to CtPhish applications. Second, attackers design CtPhish websites and applications to closely mimic the appearance and interaction logic of legitimate counterparts, thereby gaining victim's trust. Finally, once the victim is convinced of the legitimacy of website or application, they are prompted to enter their cryptocurrency wallet credentials. Attackers harvest this information and subsequently transfer the victims' cryptocurrency assets.

**Limitations of Existing Approaches.** We highlight the limitations of current approaches by analyzing two state-of-the-art methods for detecting cryptocurrency phishing websites.

One of the most relevant works is CES [10], which targets phishing websites and applications associated with cryptocurrency exchanges. However, CES exhibits three limitations:

- First, CES generates phishing website candidates by applying typo-squatting techniques to known legitimate domains and then verifies their authenticity. However, among the 5,138 CtPhish websites identified by our system, 43% could not be generated through typo-squatting, indicating CES's limited coverage of CtPhish websites.
- Second, CES identifies fake wallet applications by comparing digital signatures of applications with the same name to legitimate ones in app stores. However, our findings show that 28% of the 10,612 CtPhish applications we identified use different name from their legitimate counterparts. Moreover, CES collects applications exclusively from app stores, while many CtPhish applications are not distributed via official app stores but are instead made available through download links on CtPhish websites.

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.242854
www.ndss-symposium.org

- Third, CES does not provide an end-to-end measurement of phishing workflows, such as how victims are lured to phishing websites.

TxPhishScope [11] represents the state-of-the-art approach in detecting transaction-based cryptocurrency phishing activities. However, TxPhishScope exhibits three limitations:

- First, TxPhishScope focuses solely on website detection using Certificate Transparency (CT) logs [12] and lacks mechanisms for identifying fake wallet applications. On the one hand, CT logs exclude domains without HTTPS support. However, we found that over 10% of CtPhish websites still operate without HTTPS. On the other hand, CT logs do not capture full URL paths, even though many phishing websites are hosted in specific subdirectories. As a result, attackers can evade detection by leveraging non-HTTPS domains or obfuscated URL structures.
- Second, more critically, TxPhishScope's detection mechanism relies on observing on-chain transaction. However, CtPhish attacks primarily involve credential theft rather than transaction-based deception, meaning no on-chain activity occurs. Consequently, TxPhishScope is inherently ineffective for detecting CtPhish.
- Third, TxPhishScope also does not provide an end-to-end measurement of the phishing lifecycle and therefore fails to capture how attackers lure, deceive, and exploit victims throughout the full attack process.

In summary, existing phishing detection approaches are either unsuitable for CtPhish or suffer from critical limitations. Moreover, no prior work has conducted a comprehensive and systematic measurement of the CtPhish ecosystem. Consequently, there is an urgent need for both effective CtPhish detection mechanisms and holistic analyses that provide actionable insights into its operational characteristics and impact.

**CtPhishCapture**. In this paper, we propose CtPhishCapture to address the aforementioned challenges and effectively identify nearly all CtPhish websites and applications across the web. Specifically, CtPhishCapture first collects candidate websites using CT logs and real-time URL snapshot datasets from search engines. It then applies multi-feature fusion and probabilistic inference–based filtering to isolate cryptocurrency-related websites, substantially reducing the candidate pool. Next, CtPhishCapture employs LLM-based detection to accurately identify CtPhish websites from the filtered set. Finally, it attempts to extract potential CtPhish applications from the identified CtPhish websites. The system confirms true CtPhish applications by comparing the digital signatures and evaluating name similarity with small edit distances.

From June 23 to December 15, 2024, CtPhishCapture was deployed continuously for six months, identifying 5,138 CtPhish websites and 10,612 CtPhish applications with zero false positives. Notably, only 17% of the websites and 21% of the applications were previously reported by the community, indicating that CtPhishCapture newly discovered 83% of the websites and 79% of the applications. To further assess its effectiveness, we deployed CtPhishCapture in collaboration with Baidu. By integrating its detection results, the weekly user complaints about CtPhish are reduced by a factor of 5.8. To the best of our knowledge, CtPhishCapture represents the largest and most effective real-time detection system for CtPhish websites and applications to date.

**Measurement and Analysis**. Beyond detection, we conduct a comprehensive end-to-end measurement and analysis of the CtPhish ecosystem. Our study examines how attackers attract victims to CtPhish websites and applications, gain their trust, and ultimately exfiltrate cryptocurrency assets. We find that attackers leverage black-hat SEO techniques to enhance the search engine visibility of CtPhish websites, enabling easy victim access. These websites and applications are carefully designed to replicate the interfaces and interaction patterns of legitimate services, thereby deceiving victims into trusting them. Once victims are convinced of the site's legitimacy, attackers employ credential theft techniques—such as capturing keystrokes and covert scanning—to exfiltrate wallet credentials and drain assets. Over the six-month study period, we estimate that the ten largest CtPhish campaigns generated at least $600,000 in illicit gains.

In addition, we measure and analyze the structural and behavioral characteristics of CtPhish websites and applications, as well as the evasion strategies employed. We find that over 90% of abused URLs originate from high-reputation domains, such as `google.com` and `github.io`. CtPhish websites typically adopt cost-effective but efficient approaches, including the use of inexpensive top-level domains (TLDs), shared parent domains, and free TLS certificates. Fake wallet applications often closely resemble their legitimate counterparts while requesting fewer permissions. All compromised app stores were third-party marketplaces, with 92% of CtPhish applications hosted on stores based in China. Furthermore, attackers deploy multiple evasion strategies, including short-lived domains, user-agent spoofing, and dynamic application updates, to evade detection and prolong campaign lifespans.

**Contributions.** While prior works have explored phishing detection, they are either unsuitable for CtPhish or suffer from critical limitations. Moreover, no prior work has conducted a comprehensive and systematic measurement of the CtPhish ecosystem. This paper fills these gaps through the following key contributions:

- We propose CtPhishCapture, the largest known real-time detection system for CtPhish websites and applications. Among all dentified entities, only 17% of websites and 21% of applications were previously reported by the community, indicating that CtPhishCapture newly discovers 83% of websites and 79% of applications.
- We conduct a comprehensive end-to-end measurement and analysis of the CtPhish attack workflow, detailing how attackers lure victims, build trust, and exfiltrate cryptocurrency assets.
- We characterize the structural and behavioral features of CtPhish websites and applications, analyze their evasion strategies, and estimate the financial impact of operations.

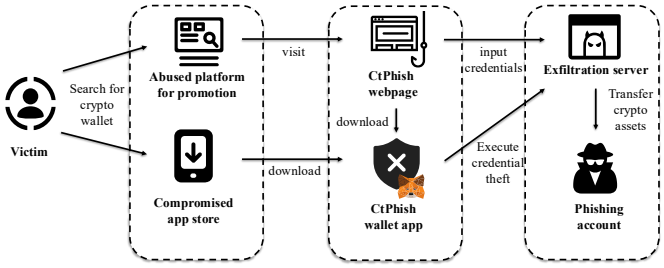



Fig. 1: The operational workflow of CtPhish scams.

- We validate CtPhishCapture through real-world deployment in collaboration with Baidu. By integrating its detection results, the weekly user complaints about CtPhish are reduced by a factor of 5.8.

## II. Dissecting CtPhish

To investigate how users fall victim to CtPhish attacks, we analyze a collection of anecdotal reports, including testimonials from both prominent cryptocurrency community members and affected victims. Our analysis identifies two primary phishing vectors: CtPhish websites and CtPhish applications. As shown in Fig. 1, CtPhish attack comprises three stages:

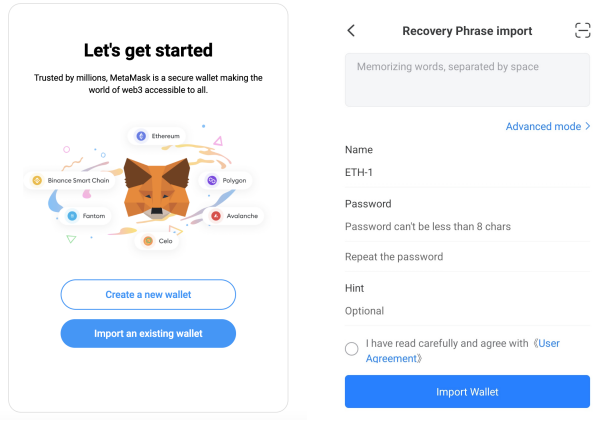
**Luring victims to CtPhish websites and applications.** According to analyzed reports, victims typically search for or download cryptocurrency wallets via search engines and app stores. As a result, attackers exploit these behaviors through two primary distribution channels. First, they post phishing content on reputable public platforms such as Google Sites and Medium, employing black-hat SEO techniques to artificially elevate the ranking of phishing pages in search results. Second, they distribute CtPhish applications through app stores with weak or absent review mechanisms.

**Deceiving victims into trusting CtPhish websites and applications.** After victims access a CtPhish website or install a CtPhish application, they encounter interfaces that closely replicate the appearance and interaction patterns of legitimate wallets, thereby fostering a false sense of trust. Fig. 2 illustrates examples of phishing interfaces from both websites and mobile applications. Similar to official wallet websites, many phishing pages also provide direct links for downloading CtPhish applications.

**Stealing victim credentials and exfiltrating cryptocurrency assets.** Once victims trust the fraudulent website or application, attackers prompt them to visit credential-stealing pages or perform malicious wallet import operations. Any entered credentials, such as private keys or seed phrases, are transmitted to the attackers, who later use them to exfiltrate the victims' cryptocurrency assets. In some cases, when victims bypass credential import interfaces, attackers deploy additional malicious techniques to extract sensitive information directly from the victims' devices.

## III. CtPhishCapture

As discussed in Section I, while prior works have explored phishing detection, they are either unsuitable for CtPhish or




(a) CtPhish website (b) CtPhish application

Fig. 2: Example of a CtPhish scam designed to obtain user credentials.

suffer from critical limitations. To address these gaps and enable accurate detection of CtPhish websites and applications, we design and implement a detection system called CtPhishCapture. Fig. 3 provides an overview of its architecture, which consists of five core components: the website collector, website filter, CtPhish website detector, CtPhish app collector, and CtPhish app detector.

### A. Website Collector

**Limitations of Prior Approaches.** Rapid detection and takedown of phishing websites are essential for mitigating their impact [13], [14]. Previous studies primarily rely on newly submitted CT logs to identify suspicious domains [11], [15]–[17]. However, this approach suffers from two key limitations. First, CT logs only record domains with valid TLS certificates, while more than 10% of phishing websites still operate without HTTPS [18]. Second, CT logs exclude full URL paths, preventing the detection of phishing content hosted in subdirectories of otherwise legitimate domains.

**Insight.** We observe that most websites are accessed through search engine indexing [19]. Therefore, we continuously crawl real-time snapshots of newly submitted URLs from search engines, ensuring coverage of both HTTPS and non-HTTPS domains. To complement this, we also collect real-time CT logs for two reasons: (1) domains newly submitted to CT logs may not yet be indexed by search engines, and (2) incorporating CT data allows direct comparison with prior CT-based detection approaches. Notably, our analysis shows that 88% of collected websites would be missed if detection relied solely on CT logs. Therefore, CtPhishCapture integrates two complementary data sources:

- **Certificate Transparency Logs.** Maintained by Certificate Authorities (CAs), CT logs record newly issued TLS certificates. They are a valuable resource for identifying suspicious domains as soon as certificates are issued.
- **Search Engine URL Snapshots.** The search engine URL snapshot dataset represents the initial stage of website indexing, encompassing both user-submitted sites and

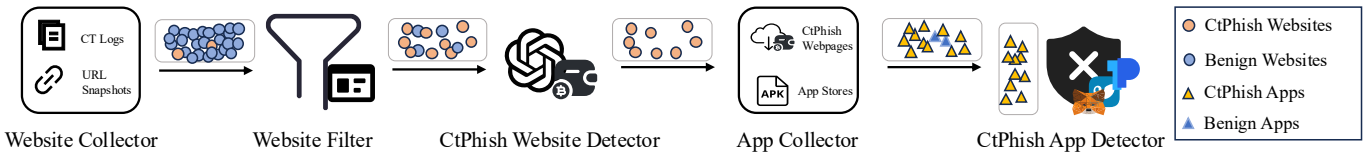



Fig. 3: Methodological framework for identifying CtPhish websites and applications.

those discovered by search engine crawlers. We collect newly indexed URLs in real time, covering both HTTPS and non-HTTPS domains. Each record contains the URL, corresponding HTML content, and a timestamp. On average, we obtain 60 million entries per day.

**Discussion:** It is worth noting that we utilize both CT logs and search engine URL snapshots to ensure comprehensive website coverage. However, this does not imply that CtPhishCapture fully relies on search engine URL snapshots. Even when operating exclusively on CT logs, CtPhishCapture can detect the majority of CtPhish instances. This is because most websites are recorded in CT logs, while only approximately 10%—those not adopting HTTPS—are absent from CT logs but appear in search engine URL snapshots. Moreover, as shown in Section III-F, we independently evaluate CtPhishCapture using CT logs and search engine URL snapshots, further verifying that our system's performance is not contingent on search engine data.

### *B. Website Filter*

**Limitations of Prior Approaches.** Given the vast number of websites collected daily, an efficient filtering mechanism is crucial to identify those related to cryptocurrency wallets. Existing studies primarily employ keyword-based filtering [11], [17], which presents four major limitations:

- ***Limitation-I: False Positives from Ambiguous Keywords.*** Common keywords such as *wallet* frequently appear on unrelated pages (e.g., news or blogs), leading to misclassifications and increased downstream workload.
- ***Limitation-II: Equal Weighting of All Keywords.*** Prior methods treat all keywords uniformly, ignoring their varying discriminative power. For example, `metamask` strongly indicates a wallet-related site, whereas `coin` is overly generic. This uniform weighting exacerbates classification errors.
- ***Limitation-III: Absence of Structural Feature Analysis.*** Keyword-based methods overlook technical features that reveal wallet functionality, such as the use of external libraries (`web3.js`) or HTML elements (`<input type=password id=private-key>`).
- ***Limitation-IV: Binary Classification Rigidity.*** Purely binary decisions lack flexibility. Incorporating confidence scores would allow more nuanced filtering and better trade-offs between precision and recall.

**Insight.** To address these challenges, we design a multi-feature probabilistic filtering algorithm, as outlined in Alg. 1. The algorithm integrates three key components to enhance both precision and adaptability.

**Algorithm 1** Filtering cryptocurrency wallet-related websites through multi-feature fusion and probabilistic inference.

**Input:** $snapshot\_urls$: URLs from CT logs and snapshot dataset; $codes$: HTML code of each URL; $keywords$: collected wallet-related keywords
**Output:** $url\_list$: list of wallet-related URLs

initialize $url\_list = []$
initialize $automaton = ahocorasick.Automaton()$
**for** $keyword$ in $keywords$ **do**
  $automaton.add(keyword)$
**end for**
**for** $url$ in $snapshot\_urls$ **do**
  **for** $code$ in $codes[url]$ **do**
    parse $code$ and extract feature vector $\mathbf{x} = [\mathbf{x}_{\text{TF-IDF}}, \mathbf{x}_{\text{struct}}]$
    $detected\_keywords = automaton.iter(code)$
    **if** $detected\_keywords.length > 0$ **then**
      compute TF-IDF$_k$ weights for matched keywords and update $\mathbf{x}_{\text{TF-IDF}}$
      estimate posterior probability $P(\text{wallet} \mid \mathbf{x})$ using a pre-trained Bayesian model
      **if** $P(\text{wallet} \mid \mathbf{x}) \geq \theta$ **then**
        $url\_list.add(url)$
      **end if**
    **end if**
  **end for**
**end for**
**return** $url\_list$

- ***Insight-I: TF-IDF Weighted Keyword Matching (Limitation-I & II).*** We adopt a term frequency–inverse document frequency (TF-IDF) weighting scheme to prioritize semantically relevant keywords. For each keyword $k$, its importance is defined as:

$$\text{TF-IDF}_k = \text{TF}_k \cdot \log\left(\frac{N}{\text{DF}_k}\right),$$

where $\text{TF}_k$ is the frequency of $k$ on a given page, $N$ is the total number of pages, and $\text{DF}_k$ denotes the number of pages containing $k$. This approach amplifies discriminative wallet-related keywords (e.g., `MetaMask`, which is frequent in wallet pages but rare overall) while down-weighting generic ones (e.g., `coin`, which is globally frequent but less relevant).
- ***Insight-II: HTML Structural Feature Analysis (Limitation-III).*** Each page is represented by a binary structural feature vector $\mathbf{x}_{\text{struct}} = [x_1, x_2, \ldots, x_m]$, where

$x_i$ = 1 if a wallet-related HTML element is detected (e.g., `<input type=''password''>` or external libraries such as `web3.js`). These features capture implementation-level indicators of wallet functionality, enhancing filtering precision.

- ***Insight-III: Bayesian Confidence Scoring (Limitation-IV).*** We employ a Bayesian posterior probability model to produce a continuous confidence score and dynamically adjust the classification threshold. Assuming a binary classification setting, the probability that a page is wallet-related given its features $\mathbf{x}$ is computed as:

$$P(\text{wallet}|\mathbf{x}) = \frac{P(\mathbf{x}|\text{wallet}) \cdot P(\text{wallet})}{P(\mathbf{x})},$$

where $P(\text{wallet})$ is the prior estimated from historical data, and the likelihood $P(\mathbf{x}|\text{wallet})$ is modeled as the product of independent Gaussian distributions:

$$P(\mathbf{x}|\text{wallet}) = \prod_{k=1}^{d} \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_k - \mu_k)^2}{2\sigma_k^2}\right),$$

where $\mu_k$ and $\sigma_k$ represent the mean and standard deviation of feature $x_k$ among wallet-related pages.

**Results.** Leveraging the three aforementioned techniques, our filtering algorithm substantially reduces the volume of candidate websites, retaining approximately 3,000 URLs per day. This performance surpasses that of existing keyword-based methods. However, the filtered set still contains a mix of legitimate and CtPhish websites, necessitating a subsequent deep detection stage powered by large models to achieve higher precision.

### *C. CtPhish Website Detector*

**Limitations of Prior Approaches.** Distinguishing CtPhish websites from legitimate ones requires detecting deceptive misuse of brand assets aimed at credential theft. Although the state-of-the-art phishing detection system PhishLLM [20] leverages URL, HTML, and visual screenshots to analyze both visual and linguistic cues via LLMs, it is not tailored for CtPhish detection. Consequently, it faces two major limitations:

- ***Limitation-I: Absence of Domain-Specific Knowledge.*** PhishLLM employs a zero-shot learning paradigm without integrating cryptocurrency wallet–specific knowledge, reducing its accuracy by 37% when identifying subtle misuse of wallet brand elements, such as minor logo, color, or layout variations.
- ***Limitation-II: Suboptimal Prompt Design.*** PhishLLM's prompts overlook cryptocurrency wallet–specific indicators and fail to prioritize visual over semantic anomalies. Moreover, it lacks a structured framework for analyzing social engineering behaviors, resulting in low recall—53% of its false negatives correspond to CtPhish cases.

**Insight.** To address these challenges, CtPhishCapture introduces three innovations to enhance CtPhish website detection.

- ***Insight-I: Domain Knowledge Integration (Limitation-I).*** We integrate cryptocurrency wallet–specific brand knowledge as structured input to strengthen LLM-based inference. A knowledge base of official brand assets for mainstream wallets (e.g., MetaMask, Trust Wallet, Coinbase Wallet) is constructed, containing both vector-format logos (e.g., SVGs) and semantic descriptions (e.g., "fox icon" for MetaMask). Each entry is jointly encoded using the CLIP model into multimodal embeddings that capture both semantic and visual relationships. This enables the system to match not only literal text but also semantic visual similarity, thereby enhancing the detection of subtle visual impersonation.
  We further employ the FAISS library to build a similarity search index over these embeddings, allowing millisecond-level retrieval of relevant reference assets. When analyzing a suspicious webpage, potential brand-related image and text elements are extracted and encoded via CLIP into query vectors. The top-$k$ most similar brand templates are retrieved and incorporated into PhishLLM as retrieval-augmented prompts. This retrieval-guided enhancement introduces domain-specific priors into the model, improving its sensitivity and accuracy in detecting brand impersonation within CtPhish websites.
- ***Insight-II: Prompt Specialization (Limitation-II).*** We redesign LLM prompts to address the unique characteristics of cryptocurrency wallet phishing by integrating both visual and semantic analyses.
  - **Visual Analysis.** The process begins with brand logo verification, where all detected logos on a webpage are extracted and compared against official references retrieved via retrieval-augmented generation (RAG). Discrepancies in design, scale, or metadata indicate potential forgery. In addition, suspicious QR codes and download buttons are identified, and their coordinates and linked destinations are recorded, prioritizing those associated with unverified or malicious sources.
  - **Semantic Analysis.** This component involves 1) *Form Field Enumeration*, which identifies all input fields—especially those requesting sensitive information such as passwords or seed phrases—and flags fields that are redundant or contextually inconsistent; and 2) *Textual Content Analysis*, which examines on-page text for social engineering indicators, including urgency, coercion, or deceptive persuasion tactics designed to manipulate user behavior.
- ***Insight-III: Adoption of GPT-4o.*** In addition, PhishLLM employs GPT-4v for website analysis, which exhibits high visual-processing latency (3.7 seconds per page) and limited comprehension of complex webpage layouts. These limitations hinder its scalability and suitability for real-time CtPhish detection. To address this, we adopt the GPT-4o architecture, which features improved visual–language alignment. This enhancement reduces processing time to 1.8 seconds per page while substantially improving the model's capability to interpret intricate phishing content.

**Output.** Based on the extracted evidence, CtPhishCapture assigns each webpage a phishing probability score ranging from 0 to 100%, reflecting its overall risk level. The system also automatically extracts and summarizes key supporting evidence in JSON format to facilitate downstream analysis and interpretation.

### *D. CtPhish Application Collector*

**Limitations of Prior Approaches.** Existing methods [21], [22] primarily rely on official app stores to collect malicious applications. However, our analysis shows that 92.8% of phishing applications are not distributed through app stores but instead disseminated directly via phishing websites.

**Insight.** As shown in Fig. 3, CtPhishCapture collects fraudulent cryptocurrency wallet applications from two complementary sources: official app stores and CtPhish websites. First, we compile a list of well-known app store domains from publicly available datasets [23], [24] and use them to identify candidate download pages from cryptocurrency-related websites. Second, we crawl the CtPhish websites identified in Section III-C to retrieve potentially malicious wallet applications directly hosted and distributed on these sites.

**Web Crawler Implementation.** We implement the crawling system using the Playwright library [25] to emulate realistic user interactions during the download process. Upon visiting a URL, the crawler extracts all hyperlinks from `<href>` tags and downloads files ending with the `.apk` extension. Since many download links are hidden behind redirections or interactive elements, the crawler also parses and interacts with `<a>`, `<button>`, and `<div>` tags to reveal embedded links. Given the frequent use of `<iframe>` elements on CtPhish websites, the crawler recursively inspects all iframes to locate hidden downloads. For sites distributing applications via QR codes, full-page screenshots are captured and manually scanned to retrieve the corresponding APK files for analysis.

**Evasion-Resilient Downloading.** Certain websites employ evasion tactics such as IP-based filtering and device-specific rendering to hinder automated collection. To mitigate these effects, we deploy a proxy pool of 12 rotating IP addresses, with each URL accessed through a randomly selected proxy. Additionally, to simulate diverse client environments, HTTP requests are issued with varied user-agent headers representing both desktop and mobile devices.

### *E. CtPhish Application Detector*

**Collection of Legitimate Cryptocurrency Wallets.** To detect CtPhish applications, we employ a signature-based comparison approach. This process begins by construction of a comprehensive corpus of legitimate cryptocurrency wallets. Using Coincarp [26], a trusted source of cryptocurrency intelligence, we obtain an initial list of popular wallets. We further expand this list by manually querying "cryptocurrency wallet" across mainstream app stores, including Google Play [27], the iOS App Store [28], and the Chrome Web Store [29]. Through manual verification, we identify 176 legitimate wallets and their official websites, covering Android, iOS, desktop, browser extensions, and hardware platforms. These 176 wallets encompass nearly all mainstream wallets available on the market. Notably, even wallets without mobile applications are often impersonated by fraudulent applications through deceptive naming and iconography. Accordingly, all 176 wallets are retained as detection targets. For those offering mobile applications, we collect 97 verified APKs and extract their package names and certificate signatures to establish ground-truth references for counterfeit detection.

**Limitations of Prior Approaches.** Existing methods [10] typically detect counterfeit applications by comparing certificate signatures among applications with identical names. However, we find that 28% of phishing applications employ names that are only partially similar to legitimate ones, significantly reducing the accuracy of name-based matching.

**Insight.** To address this, we design a multi-stage detection framework that correlates downloaded applications with legitimate wallets through both textual and visual similarity, followed by certificate verification. Unlike prior approaches, our framework does not require exact name matching. The detection pipeline comprises three stages:

- ***Insight-I: Name Similarity.*** We first calculate the edit distance between collected applications names and legitimate wallet names. Applications with an edit distance below two are flagged as potential wallet candidates. This relaxed criterion enables the detection of applications with minor spelling variations or name modifications, improving recall over strict equality matching.
- ***Insight-II: Icon Similarity.*** Next, we assess icon similarity using a Siamese neural network trained for image similarity recognition. The model effectively identifies visually consistent logo variants. Applications showing neither textual nor visual resemblance are discarded, while those exhibiting similarity in either dimension are retained as wallet candidates.
- ***Insight-III: Signature Verification.*** Finally, we compare the certificate signature of each candidate application with the verified signature of its legitimate counterpart. Since digital signatures uniquely authenticate and ensure the integrity of applications, any mismatch provides strong evidence of impersonation. Applications with inconsistent signatures are thus classified as CtPhish applications.

### *F. Evaluation of CtPhishCapture*

**Evaluation Setup.** We evaluate the website filter and the CtPhish website detector using two datasets: (1) CT logs and (2) webpage snapshots obtained from Baidu. The website filter processes raw webpages from both datasets, while the CtPhish website detector analyzes cryptocurrency wallet-related pages identified by the filter. For the website filter, we randomly sample 10,000 webpages from each dataset and manually verify their labels to compute accuracy and recall. The data is divided into three subsets: (1) *Training set* (60%) for fitting the TF-IDF vectorizer and Naive Bayes classifier, (2) *Validation set* (20%) for tuning the classification threshold, and (3) *Test set* (20%) for final evaluation. To assess the CtPhish

TABLE I: Evaluation results of CtPhishCapture in the filtering and detection stages using (a) CT log and (b) URL snapshot datasets.

| Dataset | Stage | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| CT Logs | Filter | 99.93% | 80.00% | 96.00% | 0.873 |
| | Detector | 95.00% | 100.00% | 93.80% | 0.968 |
| URL Snapshots | Filter | 99.86% | 83.58% | 94.92% | 0.889 |
| | Detector | 92.70% | 100.00% | 92.00% | 0.958 |

website detector, we randomly sample 1,000 filtered webpages from each dataset, manually annotate their ground truth, and compute standard detection metrics.

**Evaluation Results.** Table I presents the overall performance of CtPhishCapture across both datasets and detection stages. The website filter achieves high precision, effectively excluding non-cryptocurrency websites and substantially reducing downstream workload. This confirms the filter's efficiency. The CtPhish website detector attains 100% precision, accurately identifying all CtPhish websites without false positives, demonstrating its exceptional reliability.

### *G. In-the-Wild Discovery of CtPhish*

To identify CtPhish instances in the wild, we conducted a six-month detection continuous detection campaign. As shown in Fig. 4, between June 23, 2024 and December 15, 2024, CtPhishCapture identified 5,138 CtPhish websites and 10,612 CtPhish applications, averaging 29 new phishing websites and 60 phishing applications per day. To minimize false positives, all detections underwent daily manual verification, resulting in a confirmed precision rate of 100%.

Among the detected CtPhish websites, only 17% had been previously reported by community-based systems (VirusTotal [30], Chainabuse [31]) or open-source threat intelligence (TI) feeds (URLhaus [32], BlackWeb [33], StopForumSpam [34], and DynDNS [35]) at the time of discovery. Consequently, 83% were novel detections first identified by CtPhishCapture, demonstrating its effectiveness as the most comprehensive and large-scale CtPhish detection system to date. We further conducted a retrospective analysis of historical search engine click data for these sites and observed sustained user traffic even for those unreported by the community. This finding indicates that these sites were already being actively exploited by attackers and had targeted a substantial number of users.

CtPhishCapture also establishes the largest dataset of CtPhish applications. Prior studies [10] identified only 323 phishing apps, whereas CtPhishCapture uncovered 10,612 unique cases—representing a 30-fold increase over existing efforts.

### *H. Real-World Governance*

CtPhishCapture has been deployed within Baidu and has become an integral component of its security system. To date, it has operated effectively, with a significant reduction in related user complaints. Integration of its detection results led to a 5.8-fold reduction in weekly user complaints related to CtPhish

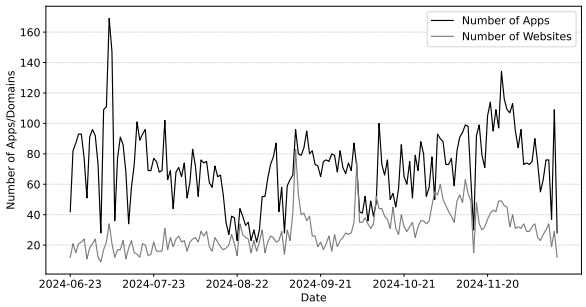



Fig. 4: Number of CtPhish websites and applications detected between June 23 and December 15, 2024.

TABLE II: An example of CtPhish information extracted from user compliant.

| | |
|---|---|
| **Fraud Details** | Mobile search for "imtoken wallet download" returns bbs.zhiyoo.com impersonating a virtual wallet, but the PC version does not show virtual currency content |
| **Involved Amount** | 30000 |
| **Scam Category** | Cryptocurrency Phishing |
| **Fraud URL** | bbs.zhiyoo.com |
| **Complaint Time** | 2024-03-19 18:10:57 |
| **Complainant UID** | 979******454 |

incidents, as summarized in Table II. As shown in Fig. 5, the number of user complaints declined markedly following deployment, with the blue-shaded regions indicating the post-deployment period. Before deployment, the system received an average of 20.55 complaints per week. After blocking the indexing of websites identified by CtPhishCapture, the weekly average fell to 3.55, with observed values ranging from a peak of 108 to zero.

## IV. End-to-End Analysis of CtPhish

In this section, we present a comprehensive end-to-end measurement and analysis of CtPhish, focusing on how attackers attract victims to CtPhish websites and applications, deceive them into trusting these fraudulent services, and ultimately steal their cryptocurrency assets.

### *A. How Victims Are Lured to CtPhish Scams*

We begin by analyzing the techniques attackers employ to direct victims to CtPhish websites and applications—the first stage of the phishing process. As summarized in Table III, these techniques include domain typosquatting, black-hat search engine optimization (SEO), and redirection for websites, as well as the distribution of malicious applications through fraudulent websites and compromised stores. We find that 92.1% of CtPhish websites and 96.7% of applications use at least one of these methods to attract victims.

**Domain Typosquatting.** Attackers commonly exploit domain typosquatting to deceive victims who mistype legitimate URLs, thereby redirecting them to CtPhish sites. To detect

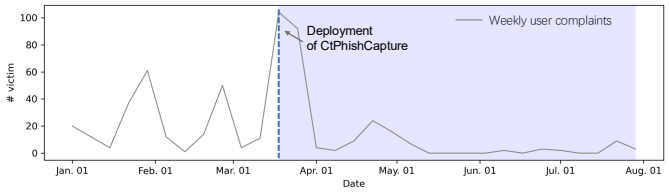



Fig. 5: Number of user complaints received between January and August 2025.

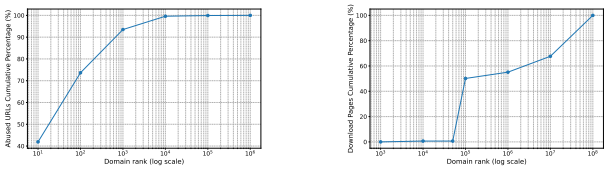
TABLE III: Techniques used to lure users. Applications downloaded from CtPhish websites are assumed to adopt the same promotional strategies as their corresponding hosting sites.

| Technique | # Websites (Proportion) | # Apps (Proportion) |
|---|---|---|
| Domain Typosquatting | 2,867(55.8%) | 6,708(63.2%) |
| Black-Hat SEO | 2,243(43.7%) | 6,241(49.4%) |
| Redirection | 1,206(23.5%) | 2,451(23.1%) |
| Abuse of App Marketplaces | - | 764(7.2%) |

such behavior, we compute the edit distance between CtPhish domains and legitimate wallet domains; domains with an edit distance of two or less are classified as typosquatting. Our analysis shows that 55.8% of the 5,138 detected CtPhish websites exhibit typosquatting behavior—a rate higher than that observed for the most frequently typosquatted domain category (advertisement landing pages, 50.55%) [36]. This underscores attackers' strong reliance on typosquatting to drive traffic to CtPhish websites.

**Black-Hat SEO.** In addition to domain typosquatting, attackers employ black-hat SEO techniques to increase the visibility of CtPhish websites in search results. They achieve this by exploiting the reputation of legitimate websites to boost malicious site rankings. To detect such abuse, we use the cryptocurrency wallet-related websites collected in Section III-C. Excluding known CtPhish sites, we identify approximately 6,000 wallet-related URLs per day. If a CtPhish domain appears in the HTML source code of these URLs, the host domain is classified as abused. Table IV lists the five most frequently abused domains. To analyze the relationship between domain reputation and abuse frequency, we rank the abused hosts using the latest Tranco top-1M list [37]. The cumulative distribution function (CDF) in Fig. 6(a) shows that over 90% of the abused domains rank within the top 1,000, indicating that attackers intentionally target high-reputation domains to promote CtPhish websites. Furthermore, 33.7% of these domains were found to rotate the specific CtPhish sites they promoted over time, suggesting continuous adaptation in their evasion strategy.

**Redirection.** Redirection is another common tactic used by attackers. For example, Google Sites [38] allows users to create free web pages under the `sites.google.com` domain and offers a URL redirection service. A typical redirection path follows the format `/url?q=https%3A%2F%2F<phishing_site>`, enabling attackers to obscure the true destination of their phishing websites.




(a) CDF of abused hosts (b) CDF of compromised app stores

Fig. 6: The cumulative distribution function (CDF) of domain rankings is analyzed for hosts exploited by CtPhish websites and compromised app stores distributing CtPhish applications. Domains absent from the Tranco ranking are uniformly assigned a default rank of 10 million.

TABLE IV: Top 5 domains exploited to lure users to CtPhish websites.

| Abused domain | Tranco rank | Category of abused pages | #Abused URLs | #Promoted CtPhish sites |
|---|---|---|---|---|
| google.com | 1 | Website builder | 3,362 | 1,045 |
| wordpress.com | 84 | Website builder | 1,451 | 434 |
| 163.com | 579 | News & blog | 1,322 | 762 |
| weebly.com | 287 | Website builder | 554 | 112 |
| sohu.com | 509 | News & blog | 467 | 256 |

**Abuse of App Stores.** CtPhish applications are primarily distributed through CtPhish websites and compromised third-party app stores. Many CtPhish websites provide direct download links. In addition, we identify 18 app stores hosting 764 unique CtPhish applications. The most affected store, `crsky.com`, hosts 377 samples and ranks within the top 100,000 domains. However, the majority of compromised stores are non-mainstream platforms. As shown in Fig. 6(b), over 30% of them do not appear in the Tranco top-1M list, reflecting attackers' preference for less-regulated stores. Major platforms such as Google Play impose stringent review mechanisms that limit fraudulent uploads, whereas smaller third-party stores often lack such safeguards, making them more vulnerable to exploitation. A longitudinal analysis reveals frequent updates to download links for malicious applications. For example, a counterfeit imToken app was updated 11 times during our observation period, with each new link replacing the previous one within three days. This pattern suggests that while some third-party stores intermittently remove detected malicious applications, they lack robust pre-screening mechanisms to prevent repeated re-uploads.

**Finding-I.** CtPhish scams attract victims through domain typosquatting, black-hat SEO, deceptive redirections, and the distribution of malicious applications via untrusted websites and compromised app stores.

### *B. How Victims Are Deceived into Trusting CtPhish Scams*

Once victims are directed to a phishing website or application, attackers exploit victims' trust in reputable brands by closely replicating the user interface (UI) and behavioral logic of legitimate services to reinforce credibility.

TABLE V: Top 10 targeted cryptocurrency wallets. "#Download" indicates the number of times an application has been downloaded from Google Play.

| Wallet | #CtPhish website | #CtPhish app | #Download |
|---|---|---|---|
| TokenPocket [45] | 2,557 | 8,601 | 5 million |
| MetaMask [46] | 1,024 | 719 | 10 million |
| imToken [47] | 454 | 694 | 1 million |
| Bitpie Wallet [48] | 64 | 319 | 1 million |
| Trust Wallet [49] | 336 | 72 | 50 million |
| Phantom Wallet [50] | 232 | 0 | 10 million |
| TronLink Pro [51] | 102 | 39 | 5 million |
| Keplr Wallet [52] | 127 | 0 | 500 thousand |
| Coinbase Wallet [53] | 46 | 25 | 10 million |
| Bitget Wallet [48] | 58 | 6 | 10 million |

**Brand Trust.** Our analysis reveals that attackers primarily impersonate well-known cryptocurrency wallet brands [39]–[44]. To assess targeting patterns and attacker preferences, we examine the distribution of impersonated brands across CtPhish websites and applications. Results show that CtPhish websites mimic 28 distinct wallet brands, while CtPhish applications impersonate 16. Table V lists the 10 most frequently targeted brands based on the combined number of associated phishing websites and applications. The three most impersonated wallets—TokenPocket, MetaMask, and imToken—account for 78.5% of CtPhish websites and 94.4% of CtPhish applications. Although these brands are not globally dominant (e.g., compared to Coinbase), their exclusive focus on wallet functionality makes them especially attractive targets for cryptocurrency-oriented attackers.

**UI Imitation.** Attackers frequently replicate the visual design of legitimate websites and applications to enhance credibility. Through systematic sampling and manual inspection of CtPhish websites, we observe a consistent pattern of visual mimicry. As detailed in Section V-B, 96.3% of CtPhish applications display a high degree of visual resemblance to their legitimate counterparts.

**Behavioral Imitation.** In addition to UI imitation, CtPhish applications often imitate the functional logic of legitimate applications. To assess the effectiveness of such deception, we conducted a controlled within-subjects usability study with eight volunteers in an Android testing environment. Participants interacted with one legitimate wallet and five CtPhish variants from Section V-B, completing standard tasks such as wallet registration and import. After each session, participants rated the app's perceived legitimacy on a continuous scale from 0 (certainly fake) to 1 (certainly legitimate). CtPhish applications achieved a mean confidence score of 0.85, demonstrating strong behavioral resemblance and a high potential for user deception.

**Finding-II.** Attackers exploit brand trust by closely replicating both the user interface and operational behavior of legitimate platforms, effectively fostering a false sense of authenticity among victims.

```
1  URL realUrl = new
   URL("https://api1.***.com/api/openapi/getauthorize?type=MAINNET&e=1&pri=");
2  String strData = "AppID=0x789&Ver=293&Type=IT&Mem=" + strUserType +
   "&VerToken=" +
   URLEncoder.encode(MnemonicUtil.publicKeyEncrypt(strMnemonic, 1), "UTF-8");
3  byte[] postData = strData.getBytes("UTF-8");
4  HttpURLConnection connection = (HttpURLConnection) realUrl.openConnection();
5  connection.setRequestMethod(ShareTarget.METHOD_POST);
6  DataOutputStream wr = new DataOutputStream(connection.getOutputStream());
7  wr.write(postData);
8  wr.flush();
9  connection.getResponseCode();
```

Fig. 7: Sending mnemonic upon input.

### C. How Attackers Successfully Defraud Victims

After victims reach and trust CtPhish websites or applications, attackers typically employ credential-exfiltration and covert-scanning techniques to harvest sensitive information and subsequently drain victims' cryptocurrency assets.

Many mainstream wallets adopt the `BIP-39` standard [54]. `BIP-39` enables users to create or import wallets using a mnemonic recovery phrase, which can be generated in one wallet application and imported into another for daily use. This interoperability greatly benefits attackers, as they can exploit stolen recovery phrases without needing to determine which wallet originally generated them.

To identify the backend endpoints used for credential exfiltration, we combine static and dynamic analysis. In static analysis, we locate HTTP-related code segments using regular expressions and manually inspect them for malicious exfiltration logic. To mitigate the impact of potential code obfuscation techniques, we perform dynamic analysis by executing applications in an instrumented mobile sandbox with network monitoring. We then import a test mnemonic or private key and observe outbound traffic; the presence of the test credentials in network requests constitutes definitive evidence of data leakage. Our analysis identifies three primary techniques used to steal credentials and assets:

**Credential Exfiltration Upon Input.** We find that 81% of CtPhish applications exfiltrate victim credentials during wallet creation or import, enabling attackers to seize control of victims' funds. In total, we identify 604 backend URLs associated with credential theft. A representative case is shown in Fig. 7.

**Covert Credential Scanning.** Approximately 27.4% of applications covertly collect local images, apply Optical Character Recognition (OCR) to screenshots, and upload them to remote servers when mnemonic-related keywords are detected. Fig. 8 illustrates an example of this technique. This method exploits users' tendency to capture screenshots during wallet setup, allowing attackers to harvest previously unused credentials for further illicit gains.

**Fake Wallet Activation via Cryptocurrency Transfer.** A small fraction (0.2%) of applications deceive users by requesting cryptocurrency transfers under the pretense of wallet activation, targeting inexperienced users for direct financial gain. We identify one Bitcoin (BTC), one Ethereum (ETH),

```
@Override // com.google.android.gms.tasks.h
/* renamed from: a, reason: merged with bridge method [inline-methods] */
public void c(b.a.c.b.c.b bVar) {
    if (bVar.a().contains("助记词")
        || bVar.a().contains("Mnemonic")
        || bVar.a().contains("memorizing")
        || bVar.a().contains("Memorizing")
        || bVar.a().contains("recovery phrase")) {
        new Thread(new a(new com.ddhooker.tokenpocket.c())).start();
    }
}
```

Fig. 8: Covert credential scanning.

TABLE VI: Top 5 certificate registrars associated with CtPhish websites.

| Certificate Registrar | # | Proportion | Free |
|---|---|---|---|
| Let's Encrypt | 2,173 | 42.3% | ✔ |
| Google Trust Services | 2,081 | 40.5% | ✔ |
| GoDaddy | 134 | 2.6% | ✗ |
| ZeroSSL | 108 | 2.1% | ✔ |
| GlobalSign | 87 | 1.7% | ✗ |

and one TRON (TRX) address associated with this scheme. The BTC address conducted 28 transactions totaling approximately 0.01 BTC, while the ETH address processed a single transaction of 0.0035 ETH. Notably, funds from the BTC address were later transferred to another address holding 53.15 BTC, suggesting aggregation of illicit proceeds.

**Finding-III.** Attackers employ input-time credential exfiltration and covert scanning techniques to capture sensitive information and misappropriate victims' cryptocurrency assets.

## V. Analysis of CtPhish Websites and Applications

In addition to conducting end-to-end measurements of the CtPhish attack lifecycle, we perform a comprehensive analysis of the characteristics and evasion techniques exhibited by CtPhish websites and applications.

### A. Analysis of CtPhish Websites

To better understand CtPhish websites and inform anti-phishing efforts, we conduct an in-depth analysis of the websites detected by CtPhishCapture. Our results identify three prevalent characteristics: the use of economically shared top-level domains (TLDs), shared parent domains, and free TLS certificates. Furthermore, Section V-C examines the evasion strategies employed by these sites, including short lifespans and user-agent cloaking.

**Economically Shared Top-Level Domains (TLDs).** Attackers deploy CtPhish websites using numerous low-cost domains. Specifically, 41.3% of CtPhish domains are registered under at least five distinct TLDs. For instance, the domain `imtOken-in` appears under 11 TLDs. These phishing domains typically avoid costly TLDs such as `.com` (2.1%) and `.org` (2.6%), instead favoring cheaper alternatives like `.pro` (16.3%), `.life` (13.7%), and `.club` (11.8%). Cost analysis

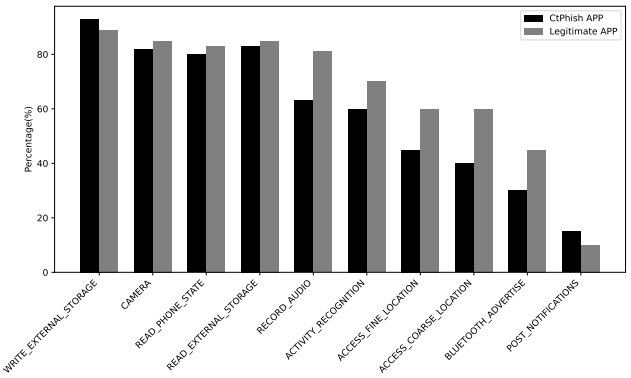



Fig. 9: Comparison of dangerous permissions requested by CtPhish and legitimate applications.

indicates that such domains incur only 5.4% of the registration cost compared to those using more expensive TLDs.

**Shared Parent Domains.** To further reduce registration expenses, many CtPhish websites operate under shared parent domains. Among the 5,138 analyzed websites, 2,034 (39.7%) were associated with 345 parent domains. For instance, 879 CtPhish websites were hosted under `tokenpockit.com`, registered through the provider GNAME [55], with all TLS certificates issued for free by Let's Encrypt. Given GNAME's annual registration fee of approximately $200 for `tokenpockit.com`, the effective per-site cost is only $0.22.

**Free TLS Certificates.** Our analysis reveals that attackers predominantly utilize free TLS certificates to secure CtPhish websites. As shown in Table VI, 85.7% of phishing wallet websites use free certificates issued by authorities such as Let's Encrypt, Google Trust Services, and ZeroSSL. Including non-TLS websites, 94.9% (4,877 out of 5,138) of CtPhish websites incur no certificate-related costs.

**Finding-IV.** CtPhish websites extensively exploit inexpensive TLDs, shared parent domains, and free TLS certificates to enable large-scale, low-cost deployment while obscuring attribution.

### B. Analysis of CtPhish Applications

In addition to analyzing websites, we conduct an in-depth analysis of 10,612 CtPhish applications detected by CtPhishCapture. Our findings highlight two prominent characteristics: strong similarity to legitimate wallet applications and limited permission requests. Furthermore, the evasion strategies employed by these applications (e.g., user-agent cloaking and dynamic updates) are discussed in Section V-C. To the best of our knowledge, this constitutes the largest analysis of CtPhish applications to date—approximately 30 times larger than prior work [10], which examined only 323 samples.

**Application Similarity.** CtPhish applications demonstrate substantial similarity to their legitimate counterparts. To quantify this resemblance, we conduct a code-level similarity analysis using SimiDroid [56], which supports pairwise comparisons of resource and code similarity. We generate a similarity matrix for all applications and apply clustering to group samples with similarity scores exceeding 0.80. This process yields

103 distinct clusters: 89 correspond to unique applications, while the remaining 14 clusters—comprising 10,219 samples (96.3%)—target 14 legitimate wallet applications.

**Permission Usage.** CtPhish applications generally request fewer Android Open Source Project (AOSP) permissions [57]. Using Apktool [58], we extract and analyze all declared permissions, identifying 452 unique entries—141 standard AOSP permissions and 311 custom ones defined by developers or third-party libraries. To evaluate privacy implications, we focus on dangerous AOSP permissions. As shown in Fig. 9, CtPhish applications request an average of 7.2 dangerous permissions, compared to 11.8 for legitimate applications. This indicates that legitimate applications, due to their broader functionality, may pose higher privacy risks. For instance, the official TokenPocket app declares 20 dangerous permissions (e.g., `READ_MEDIA_IMAGES`, `READ_MEDIA_AUDIO`), whereas CtPhish TokenPocket variants average 12.1. The reduced permission usage in CtPhish applications reflects their narrower focus on credential theft and their intent to minimize suspicion.

**Finding-V.** CtPhish applications emulate legitimate applications in both appearance and behavior while deliberately minimizing permission requests to enhance stealth and avoid detection.

### C. Bypass Techniques

As phishing detection mechanisms advance, attackers continuously refine their evasion strategies to sustain their operations. This section examines the primary techniques employed by CtPhish websites and applications to circumvent detection.

**Short Lifespan.** Over 82.9% of CtPhish websites exhibit lifespans shorter than one month. We evaluate each website's lifetime based on four temporal indicators: WHOIS domain registration date, detection date, and deactivation date. Our analysis shows that 69.8% of domains become inactive within two weeks of detection. When defining a website's lifespan as the interval between WHOIS registration and deactivation, more than 82.9% of CtPhish websites were found to be short-lived (i.e., less than one month). This underscores the importance of timely detection and takedown of CtPhish websites to protect users from fraud.

**User-Agent Cloaking.** Approximately 22.1% of CtPhish websites employ user-agent–based cloaking. When accessed via a desktop browser, these sites redirect users to legitimate download pages on Google Play or the Apple App Store. In contrast, when accessed from a mobile device, the same websites embed direct download links to the CtPhish applications behind the icons of Google Play or the App Store. This cloaking strategy is particularly effective in black-hat SEO. Beyond improving search engine ranking, it also allows attackers to better target and filter potential victims from mobile users.

**Dynamic Updating.** As discussed in Section V-B, we identify six clusters containing 8,821 CtPhish applications with intra-cluster similarity scores of 1.0, indicating that applications within each cluster are completely identical except for their application signatures. Further examination revealed that these wallets are dynamically generated by modifying application package names. For example, in one cluster, the CtPhish website domain follows the pattern `[a-z]{6}.tokenpockit.com`, and the application download domain follows the pattern `[a-z]{6}.tkfptavj.top`. The downloaded APKs follow the naming convention `TokenPocket_[0-9]{6}.apk`, with package names in the format `[a-z]{8}.[a-z]{10}`. This dynamic updating strategy complicates detection by security vendors, as it renders simple signature-based comparisons based on application names ineffective. We even suspect that this design may have been developed explicitly to bypass prior research efforts, thereby validating our choice to use similarity-based clustering rather than relying solely on matching application names or signatures.

**Finding-VI.** CtPhish websites and applications employ evasion techniques, including short-lived lifespans, user-agent cloaking, and dynamic content updating, to circumvent detection and impede analysis.

## VI. Analysis of CtPhish Campaigns and Profits

In this section, we first analyze the data exfiltration servers used by CtPhish. We then examine attacker behaviors across related scam campaigns. Finally, we trace attacker-controlled wallet addresses to estimate their illicit gains.

### A. Data Exfiltration Server Analysis

As shown in Fig. 1 and discussed in Section IV-C, CtPhish data exfiltration servers collect victims' credentials from two primary sources: direct user input on CtPhish websites and credentials stolen by CtPhish applications. Using user behavior simulation, dynamic analysis, and manual verification, we identify 873 server domains exfiltrating sensitive data such as seed phrases and private keys.

Only 10.2% of these servers share a parent domain with their associated CtPhish websites, and 11.7% share one with the corresponding application download URLs. According to VirusTotal [59], merely 21.2% of these domains are flagged as malicious or phishing-related by at least one security vendor, compared to a 42.3% detection rate for CtPhish websites. This indicates that attackers intentionally decouple exfiltration servers from primary phishing infrastructure to improve stealth and evade detection.

### B. Campaign Analysis

We cluster attacker campaigns based on shared characteristics extracted from CtPhish websites and applications. Entities with overlapping attributes are grouped into the same campaign.

*1) Common eTLD+1:* The eTLD+1 [60] generally represents a registrable domain controlled by a single registrant. Some applications are not distributed from the same domain as their associated CtPhish websites. CtPhish websites, application download links, and data exfiltration servers sharing the same eTLD+1 are clustered into one campaign.

TABLE VII: Top 10 CtPhish campaigns.

| Campaign | #CtPhish domain | #CtPhish app | Exfiltration server | Target wallet(s) | Minmum income($) |
|---|---|---|---|---|---|
| C1 | 4 | 4,046 | bit***wallet.com | MetaMask,TokenPocket,imToken | 12,000.43 |
| C2 | 1 | 1,068 | crypto***wallet.com | TokenPocket | 5,644.32 |
| C3 | 15 | 910 | 1***p.com | TokenPocket,imToken | 256.41 |
| C4 | 48 | 610 | web.to***-cn.top | MetaMask,TokenPocket,imToken | 591,439.87 |
| C5 | 62 | 574 | token***ert.net | TokenPocket | 8278.00 |
| C6 | 26 | 547 | fox***.cc | Trust Wallet,TokenPocket,imToken,Bitpie | 24.05 |
| C7 | 7 | 472 | tg***soft.one | TokenPocket,imToken | - |
| C8 | 10 | 337 | a***trust.com | TokenPocket,imToken | - |
| C9 | 59 | 227 | qidian***.cn | MetaMask,imToken | 4,343.62 |
| C10 | 3 | 163 | itoo***.com | TokenPocket,Bitpie | - |

*2) Common Application Certificate Signature:* Android developer certificates uniquely identify application publishers. Applications signed with the same certificate are attributed to the same attacker and classified within a single campaign.

A cluster qualifies as a campaign if it includes at least one CtPhish website and one CtPhish application. Using these features, we identify 81 distinct campaigns, comprising 4,198 (81.7%) CtPhish domains and 9,554 (90.0%) CtPhish applications. Table VII presents the 10 largest campaigns by application count. Notably, 8 of these campaigns target multiple cryptocurrency wallets, suggesting that attackers frequently reuse phishing templates across wallets to harvest universally applicable recovery phrases.

### C. Revenue Estimation

The primary objective of CtPhish attackers is the illicit acquisition of cryptocurrency assets. However, due to the scarcity of confirmed attacker-controlled wallet addresses, accurately estimating total campaign revenue remains challenging. To address this, we deployed honeypot wallets to monitor attacker transactions. Specifically, we created 50 honeypot wallets, each funded with $2 via MetaMask, and imported them into five counterfeit wallet applications using their recovery phrases. Among these, 26 honeypots were compromised, and their funds were transferred to 14 distinct wallet addresses linked to 7 campaigns.

Once attackers obtained the recovery phrases, they modified wallet configurations to establish co-ownership, enabling unilateral fund withdrawal without the victim's involvement.

Using historical transaction data collected during the study period, we estimated the minimum revenue for each campaign, as summarized in Table VII. No wallet address was shared across multiple campaigns. Overall, the campaigns accumulated at least $611,186.90 from 104 transactions, corresponding to an average loss of $5,876.79 per victim. The most profitable campaign (C4) generated nearly $600,000. Notably, campaign revenue showed no strong correlation with the number of associated CtPhish websites or applications, indicating that attackers likely exploit opportunistic rather than large-scale operations for profit.

**Finding-VII.** We identify 81 distinct CtPhish campaigns, with the most profitable (C4) generating nearly $600,000 in illicit revenue.

## VII. Discussion

### A. Limitations

**App Collection.** All fake wallet applications were collected from publicly accessible sources (official app stores and direct-download websites) by correlating newly observed cryptocurrency wallet-related web domains with store domains (Section III-D). This approach inherently limits our ability to examine only those fake apps that were available during the period of our research. Our monitoring indicates that the average lifespan of these applications in app stores is less than three months. We therefore believe the six-month collection window captured the majority of active fake apps present in app stores, but transient or rapidly rotated samples could be underrepresented. Additionally, the current dataset primarily focuses on Android applications, which is due to the closed ecosystem and privacy restrictions of the iOS platform. To improve coverage in future work, we plan to: (1) collaborate with mobile security companies to obtain iOS threat intelligence and privately distributed samples, and (2) develop platform-compliant crawlers to identify CtPhish iOS applications disseminated through social platforms such as X and Telegram.

**Evasion of Detection.** CtPhish operators employ dynamic evasion techniques that can defeat automated collection and analysis. Examples include HTML- or user-agent–based cloaking, conditional content rendering, and delivery of payloads only after specific interactions or environmental checks. Such behaviors can bypass both DOM- and screenshot-based detectors and impede APK retrieval. Resource constraints limit exhaustive countermeasures against every evasion variant. To mitigate these risks, we implemented several defenses: a Playwright-based browser emulator to exercise interactive elements, a rotating residential proxy pool, and mixed mobile/desktop crawling profiles (Section V-C). Despite these measures, some evasive campaigns may remain undetected, and our reported counts should be interpreted as conservative lower bounds.

**Limited Data on Wallet Addresses.** Due to funding constraints and ethical compliance requirements, we deployed only a limited number of honeypot wallets. Thus, the reported attack profits represent a conservative lower bound. Nevertheless, our honeypot experiments identified 14 addresses collectively responsible for over $0.6 million in illicit gains. In future work, we plan to enhance financial analysis by:

(1) expanding honeypot coverage to include multiple on-chain wallets and cross-platform accounts, and (2) integrating blockchain graph–based fund tracing methods (e.g., as in TxPhish [11]) to more accurately quantify attack profits.

### B. Responsible Disclosure

We responsibly disclosed our findings to affected cryptocurrency wallet providers and shared relevant threat intelligence. Both imToken and OKX acknowledged receipt, expressed appreciation for our efforts, and indicated their intention to strengthen defenses against CtPhish-related scams.

## VIII. Related works

**Giveaway Scams.** Giveaway scams involve fraudsters impersonating influential figures or official organizations, claiming to distribute or multiply victims' cryptocurrency to induce transfers [61]. Li et al. [17] developed CryptoScamTracker, which leverages CT logs to detect likely giveaway scams. By analyzing phishing pages and associated wallet accounts, they estimated attacker profits in the tens of millions of dollars. Similarly, He et al. [11] examined a new class of transaction-based scam websites, termed TxPhish, which deceive users into signing malicious transactions that result in unauthorized cryptocurrency transfers.

**Cryptocurrency Scams on Social Media.** Social media platforms are also frequently exploited for cryptocurrency-related fraud. Li et al. [62] identified over 9,000 giveaway scams on Twitter, estimating total losses of approximately $872,000. Roy et al. [63] detected more than 300 fraudulent NFT projects on the same platform, while Acharya et al. [64] investigated cryptocurrency-based technical support scams using automated interaction systems. YouTube has likewise been misused to promote fraudulent "bot contract" schemes that drain victims' funds [65]. In contrast, our findings reveal limited use of social media for promoting phishing websites. Instead, attackers predominantly leverage web hosting services—such as `sites.google.com` and `wordpress.com`—to disseminate their campaigns.

**Phishing Scams.** Numerous studies have examined phishing activities targeting cryptocurrency ecosystems, including exchanges, wallets, and tokens. The work most relevant to ours is by Xia et al. [10], which detected phishing campaigns targeting cryptocurrency exchanges. Using typosquatting-based domain generation techniques, they identified over 1,500 scam domains and more than 300 fake applications. In contrast, our real-time dataset reveals 5,138 CtPhish websites and 10,612 CtPhish applications—substantially expanding the observed scope of such attacks. Wang et al. [66] investigated cryptocurrency-themed malicious browser extensions, detecting 65 phishing extensions. Regarding token-related scams, Gao et al. [67] identified 2,117 counterfeit ERC-20 tokens imitating top ERC-20 assets. Ye et al. [68] further explored visually deceptive wallet scams, uncovering over twenty thousand victims defrauded through fraudulent wallet interfaces.

## IX. Conclusion

In this paper, we present a comprehensive study of CtPhish, a large-scale credential-theft phishing ecosystem targeting cryptocurrency users. We begin by reviewing existing phishing detection approaches and identifying their limitations in addressing CtPhish threats. To bridge these gaps, we develop CtPhishCapture, a detection system capable of identifying both CtPhish websites and applications. During a six-month deployment, CtPhishCapture detected 5,138 websites and 10,612 applications, of which 83% and 79%, respectively, were previously unreported—establishing CtPhishCapture as the largest known detection framework for this threat.

Using CtPhishCapture's detection results, we conduct an end-to-end measurement of the CtPhish ecosystem, examining attacker strategies for luring victims, establishing credibility, and exfiltrating cryptocurrency assets. We further analyze the technical characteristics of the CtPhish websites and applications, the evasion techniques employed, and the financial impact of these scams.

We deploy CtPhishCapture in collaboration with Baidu. By integrating its detection results, the weekly user complaints about CtPhish are reduced by a factor of 5.8.

## X. Ethical Considerations

Our analysis strictly followed established ethical guidelines, including the Belmont Report [69] and the Menlo Report [70]. All user complaint data were manually reviewed, and any personally identifiable information was anonymized using MD5 hashing before access. Consequently, the complaint analysis involved no personal data. The snapshot dataset contained only URLs and HTML files, ensuring that no user-specific information was included. For the user study, no personal data were collected, and all records were securely stored on the researcher's local machine. Informed consent was obtained from all participants, who were clearly informed of their right to withdraw from the study at any time without consequence.

## Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful feedback. This work was in part supported by the National Key Research and Development Program of China (No. 2023YFC3321303), the Zhongguancun Laboratory, the Quan Cheng Laboratory (Grant No. QCL20250108), and the Research Project of Provincial Laboratory of Shandong (Grant No. SYS202201). Authors from Nankai University were also supported by the National Natural Science Foundation of China (No. 62502236).

## References

[1] SecurityAffairs, "Three fake bitcoin wallet apps were removed from the official google play," https://securityaffairs.com/67123/malware/fake-bitcoin-wallet-apps.html, 2017.

[2] Protos, "Fake crypto wallet in app store for four years drained $120k in stacks," https://protos.com/fake-crypto-wallet-in-app-store-for-four-years-drained-120k-in-stacks/, 2024.

[3] Cointelegraph, "iphone user blames apple for $600k bitcoin theft via fake app," https://cointelegraph.com/news/iphone-user-blames-apple-for-600k-bitcoin-theft-via-fake-app, 2021.

[4] Trendmicro, "Watch out for fake crypto wallet apps, $4.3m stolen — metamask, imtoken, bitpie, trust wallet, and more!" https://news.trendmicro.com/2022/01/20/watch-out-for-fake-crypto-wallet-apps-4-3m-stolen-metamask-imtoken-bitpie-trust-wallet-and-more/, 2022.

[5] Financefeeds, "Over 10 million targeted by fake crypto app ads," https://financefeeds.com/over-10-million-targeted-by-fake-crypto-app-ads/, 2025.

[6] Cyble.com, "Over 20 crypto phishing applications found on the play store stealing mnemonic phrases," https://cyble.com/blog/crypto-phishing-applications-on-the-play-store/, 2025.

[7] Cointelegraph, "Fake curve finance app listed on apple store," https://cointelegraph.com/news/fake-curve-finance-app-listed-apple-store, 2024.

[8] ——, "Fake rabby wallet wreaks havoc after listing on apple app store," https://cointelegraph.com/news/apple-yet-to-remove-fake-rabby-wallet-app-despite-users-being-drained, 2024.

[9] ——, "Fake ledger live app sneaks into microsoft's app store, $588k stolen," https://cointelegraph.com/news/fake-ledger-live-app-sneaks-into-microsoft-app-store-as-victims-lose-half-a-million, 2023.

[10] P. Xia, H. Wang, B. Zhang, R. Ji, B. Gao, L. Wu, X. Luo, and G. Xu, "Characterizing cryptocurrency exchange scams," *Comput. Secur.*, vol. 98, p. 101993, 2020. [Online]. Available: https://doi.org/10.1016/j.cose.2020.101993

[11] B. He, Y. Chen, Z. Chen, X. Hu, Y. Hu, L. Wu, R. Chang, H. Wang, and Y. Zhou, "Txphishscope: Towards detecting and understanding transaction-based phishing on ethereum," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, W. Meng, C. D. Jensen, C. Cremers, and E. Kirda, Eds. ACM, 2023, pp. 120–134. [Online]. Available: https://doi.org/10.1145/3576915.3623210

[12] C. Transparency, "Certificate transparency," https://certificate.transparency.dev/, 2024.

[13] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, and G. Ahn, "Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale," in *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, S. Capkun and F. Roesner, Eds. USENIX Association, 2020, pp. 361–377. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/oest-sunrise

[14] A. Oest, Y. Safaei, A. Doupé, G. Ahn, B. Wardman, and K. Tyers, "Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists," in *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*. IEEE, 2019, pp. 1344–1361. [Online]. Available: https://doi.org/10.1109/SP.2019.00049

[15] M. A. Sabah, M. Nabeel, Y. Boshmaf, and E. Choo, "Content-agnostic detection of phishing domains using certificate transparency and passive DNS," in *25th International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2022, Limassol, Cyprus, October 26-28, 2022*. ACM, 2022, pp. 446–459. [Online]. Available: https://doi.org/10.1145/3545948.3545958

[16] E. Fasllija, H. F. Eniser, and B. Prünster, "Phish-hook: Detecting phishing certificates using certificate transparency logs," in *Security and Privacy in Communication Networks - 15th EAI International Conference, SecureComm 2019, Orlando, FL, USA, October 23-25, 2019, Proceedings, Part II*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, S. Chen, K. R. Choo, X. Fu, W. Lou, and A. Mohaisen, Eds., vol. 305. Springer, 2019, pp. 320–334. [Online]. Available: https://doi.org/10.1007/978-3-030-37231-6_18

[17] X. Li, A. Yepuri, and N. Nikiforakis, "Double and nothing: Understanding and detecting cryptocurrency giveaway scams," in *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society, 2023. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/double-and-nothing-understanding-and-detecting-cryptocurrency-giveaway-scams/

[18] W3Techs, "Usage statistics of Default protocol https for websites," https://w3techs.com/technologies/details/ce-httpsdefault, 2024.

[19] H. S. Xavier, "The web unpacked: A quantitative analysis of global web usage," in *Proceedings of the 20th International Conference on Web Information Systems and Technologies, WEBIST 2024, Porto, Portugal, November 17-19, 2024*, F. J. García-Peñalvo, K. Aberer, and M. Marchiori, Eds. SCITEPRESS, 2024, pp. 183–190. [Online]. Available: https://doi.org/10.5220/0012905900003825

[20] R. Liu, Y. Lin, X. Teoh, G. Liu, Z. Huang, and J. S. Dong, "Less defined knowledge and more true alarms: Reference-based phishing detection without a pre-defined reference list," in *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*, D. Balzarotti and W. Xu, Eds. USENIX Association, 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/liu-ruofan

[21] K. Li, S. Guan, and D. Lee, "Towards understanding and characterizing the arbitrage bot scam in the wild," in *Abstracts of the 2024 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS/PERFORMANCE 2024, Venice, Italy, June 10-14, 2024*, M. Garetto, A. Marin, F. Ciucu, G. Fanti, and R. Righter, Eds. ACM, 2024, pp. 89–90. [Online]. Available: https://doi.org/10.1145/3652963.3655088

[22] Y. Liu, Y. Zhang, B. Liu, H. Duan, Q. Li, M. Liu, R. Li, and J. Yao, "Tickets or privacy? understand the ecosystem of chinese ticket grabbing apps," in *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*, D. Balzarotti and W. Xu, Eds. USENIX Association, 2024. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/liu-yijing

[23] alternativeto, "APK Stores," https://alternativeto.net/lists/12450/apk-stores/, 2019.

[24] Chinaz, "Software Download Website Ranking ," https://top.chinaz.com/hangye/index_wangluo_ruanjian.html, 2024.

[25] Playwright, "Playwright enables reliable end-to-end testing for modern web apps." https://playwright.dev/, 2024.

[26] CoinCarp, "Compare bitcoin, ethereum & other cryptocurrency walletso," https://www.coincarp.com/wallets/, 2024.

[27] GooglePlay, "Google Play," https://play.google.com/, 2025.

[28] AppStore, "App Store," https://www.apple.com/app-store/, 2025.

[29] ChromeWebStore, "Chrome Web Store," https://chromewebstore.google.com/, 2025.

[30] VirusTotal, "Virustotal," https://www.virustotal.com/, 2025.

[31] chainabuse, "Scam reports," https://www.chainabuse.com/reports, 2024.

[32] Urlhaus, "Urlhaus," https://urlhaus.abuse.ch/, 2025.

[33] BlackWeb, "Blackweb," https://github.com/marav ento/blackweb/, 2025.

[34] S. F. Spam, "Stop forum spam," https://www. stopforumspam.com/, 2025.

[35] D. DNS, "Dyn dns," http://security-research. dyndns.org/pub/malware-feeds/, 2025.

[36] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis, "Seven months' worth of mistakes: A longitudinal study of typosquatting abuse," in *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015*. The Internet Society, 2015. [Online]. Available: https://www.ndss-symposium.org/ndss2015/seven-months-worth-mistakes-longitudinal-study-typosquatting-abuse

[37] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, ser. NDSS 2019, Feb. 2019.

[38] G. Sites, "Google sites," https://sites.google.com/, 2025.

[39] MetaMask, "eth-phishing-detect," https://github.com/MetaMask/eth-phishing-detect, 2024.

[40] ——, "phishing sites report," https://github.com/MetaMask/eth-phishing-detect/pull/13717, 2024.

[41] imToken, "imtoken on security: How to tell if a website or an app is not fake?" https://support.token.im/hc/en-us/articles/4405264410393-imToken-on-security-How-to-tell-if-a-website-or-an-App-is-not-fake, 2024.

[42] TokenPocket, "How to verify whether the wallet is genuine," https://help.tokenpocket.pro/en/security-knowledge/security-measure/verify, 2024.

[43] TrustWallet, "How to avoid fake trust wallet apps," https://trustwallet.com/blog/how-to-avoid-fake-trust-wallet-apps, 2024.

[44] MetaMask, "How do i recognize the real metamask?" https://support.metamask.io/privacy-and-security/staying-safe-in-web3/how-do-i-recognize-the-real-metamask-/, 2024.

[45] TokenPocket, "Tokenpocket," https://www.tokenpocket.pro/, 2025.

[46] MetaMask, "Metamask," https://metamask.io/, 2025.

[47] imToken, "imtoken," https://token.im/, 2025.

[48] Bitget, "Bitget wallet," https://www.bitget.com/, 2025.

[49] TrustWallet, "Trust wallet," https://trustwallet.com/, 2025.
[50] Phantom, "Phantom wallet," https://phantom.com/, 2025.
[51] TronLink, "Tronlink pro," https://www.tronlink.org/, 2025.
[52] Keplr, "Keplr wallet," https://www.keplr.app/, 2025.
[53] Coinbase, "Coinbase wallet," https://www.coinbase.com/, 2025.
[54] BitcoinWiki, "Bip 0039," https://en.bitcoin.it/wiki/BIP_0039, 2013.
[55] GNAME, "GNAME," https://www.gname.com/, 2024.
[56] L. Li, T. F. Bissyandé, and J. Klein, "Simidroid: Identifying and explaining similarities in android apps," in *2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, Australia, August 1-4, 2017*. IEEE Computer Society, 2017, pp. 136–143. [Online]. Available: https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.230
[57] AOSP, "Android Open Source Project," https://source.android.com/, 2025.
[58] Apktool, "Apktool," https://apktool.org/, 2025.
[59] VirusTotal, "Virustotal api v3 overview," https://virustotal.readme.io/reference/overview, 2024.
[60] MDN, "etld," https://developer.mozilla.org/en-US/docs/Glossary/eTLD, 2025.
[61] R. Phillips and H. Wilder, "Tracing cryptocurrency scams: Clustering replicated advance-fee and phishing websites," in *IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020, Toronto, ON, Canada, May 2-6, 2020*. IEEE, 2020, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ICBC48266.2020.9169433
[62] K. Li, D. Lee, and S. Guan, "Understanding the cryptocurrency free giveaway scam disseminated on twitter lists," in *IEEE International Conference on Blockchain, Blockchain 2023, Danzhou, China, December 17-21, 2023*. IEEE, 2023, pp. 9–16. [Online]. Available: https://doi.org/10.1109/Blockchain60715.2023.00012
[63] S. S. Roy, D. Das, P. Bose, C. Kruegel, G. Vigna, and S. Nilizadeh, "Unveiling the risks of NFT promotion scams," in *Proceedings of the Eighteenth International AAAI Conference on Web and Social Media, ICWSM 2024, Buffalo, New York, USA, June 3-6, 2024*, Y. Lin, Y. Mejova, and M. Cha, Eds. AAAI Press, 2024, pp. 1367–1380. [Online]. Available: https://doi.org/10.1609/icwsm.v18i1.31395
[64] B. Acharya, M. Saad, A. E. Cinà, L. Schönherr, H. D. Nguyen, A. Oest, P. Vadrevu, and T. Holz, "Conning the crypto conman: End-to-end analysis of cryptocurrency-based technical support scams," *CoRR*, vol. abs/2401.09824, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2401.09824
[65] K. Li, S. Guan, and D. Lee, "Towards understanding and characterizing the arbitrage bot scam in the wild," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 7, no. 3, pp. 52:1–52:29, 2023. [Online]. Available: https://doi.org/10.1145/3626783
[66] K. Wang, Y. Ling, Y. Zhang, Z. Yu, H. Wang, G. Bai, B. C. Ooi, and J. S. Dong, "Characterizing cryptocurrency-themed malicious browser extensions," in *Abstract Proceedings of the 2023 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2023, Orlando, FL, USA, June 19-23, 2023*, E. Smirni, K. Avrachenkov, P. Gill, and B. Urgaonkar, Eds. ACM, 2023, pp. 91–92. [Online]. Available: https://doi.org/10.1145/3578338.3593529
[67] B. Gao, H. Wang, P. Xia, S. Wu, Y. Zhou, X. Luo, and G. Tyson, "Tracking counterfeit cryptocurrency end-to-end," in *SIGMETRICS '21: ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems, Virtual Event, China, June 14-18, 2021*, L. Huang, A. Gandhi, N. Kiyavash, and J. Wang, Eds. ACM, 2021, pp. 33–34. [Online]. Available: https://doi.org/10.1145/3410220.3456282
[68] G. Ye, G. Hong, Y. Zhang, and M. Yang, "Interface illusions: Uncovering the rise of visual scams in cryptocurrency wallets," in *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, T. Chua, C. Ngo, R. Kumar, H. W. Lauw, and R. K. Lee, Eds. ACM, 2024, pp. 1585–1595. [Online]. Available: https://doi.org/10.1145/3589334.3645348
[69] J. M. Sims, "A brief review of the belmont report," *Dimensions of critical care nursing*, vol. 29, no. 4, pp. 173–174, 2010.
[70] M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan, "The menlo report," *IEEE Security & Privacy*, vol. 10, no. 2, pp. 71–75, 2012.