TL;DR

: Sybil attacks undermining the integrity of retrospective airdrops in Web3. Greedy actors create fake accounts to unfairly earn more airdropped tokens. The article discusses different Sybil resistance approaches like proof-of-personhood and community reporting, highlighting their limitations. It then introduces Trusta's AI and machine learning powered framework to systematically analyze on-chain data and identify suspicious Sybil clusters. The 2-phase approach first uses graph mining algorithms to detect coordinated communities, then refines results with user behavior analysis to reduce false positives. Examples demonstrate how Trusta identified real onchain Sybil clusters. The article advocates AI-ML as a robust sybil resistance solution that preserves user privacy and permissionless participation.

Introduction

Sybil Attacks Undermine the Integrity of Retrospective Airdrops

Since Uniswap began using airdrops in 2020 to reward early users, airdrops have become very popular in Web3. Airdrops

refer to distributing tokens to current or past users' wallets to spread awareness, build ownership, or retroactively reward early adopters. However, the original intent of airdrops can be undermined by Sybil attacks. Sybil attacks

happen when dishonest actors generate fake accounts and manipulate activities to unfairly earn more airdropped tokens. Therefore, identifying the Sybil accounts forged by airdrop farmers and attackers has become a critical issue.

Proof-Of-Personhood VS. AI-Powered Machine Learning Algorithms

Proof-of-personhood

methods like biometric scans (e.g. iris scanning in World Coin Project) or social media verification check humanities by requiring identity confirmation. However, permissionless and pseudonymous participation are core Web3 values. While proof-of-personhood prevents Sybil creation, it also adds friction for users and compromises privacy. There is a need for solutions that stop airdrop farming without undermining privacy or independence.

Onchain activities represent a user's unique footprint, providing massive datasets where data scientists can gain insights. Trusta leverages big data and expertise in AI and machine learning to address the Sybil problem. Comparing the two approaches, AI-powered machine learning (AI-ML

) Sybil identification has advantages over proof-of-personhood:

1. AI-ML preserves privacy

as users don't provide their bio-information and their identities in Web2. Proof-of-personhood compromises anonymity by requiring identity confirmation.

1. AI-ML comprehensively analyzes massive onchain data to reduce vulnerability

. Proof-of-personhood is vulnerable as verified identities can be exploited.

1. AI-ML is inherently permissionless

as anyone can analyze the same public onchain data.

1. Sybil judgements can be publicly double verified

due to the transparent analysis.

Gitcoin passport incorporates both methods. It mainly uses proof-of-personhood but added Trusta's AI-ML TrustScan score before GG18, combining their advantages for reliable Sybil resistance.

Project Airdrops and Sybil Resistance Approaches

[

1400×627 133 KB

](https://ethresear.ch/uploads/default/original/2X/c/c7435d8788c94c5005ebd4193270414eb05a80f1.png)

Recent major airdrops reveal gaps in anti-sybil expertise. Aptos lacked anti-sybil rules when launching their airdrop. Airdrop hunters claimed many $APT tokens, pumped the price after exchange listing, then massively dumped tokens. Researchers found sybil addresses accounted for 40% of tokens deposited to exchanges.

Some projects like HOP and Optimism encouraged community reporting for Sybils from eligible addresses. This shifted sybil resistance responsibility to the community. Although well-intended, the program sparked controversy. Reported Sybil accounts even threatened to poison other wallets, which could disrupt the entire community-led sybil resistance effort.

Since 2023, AI-ML Sybil resistance has grown more popular. Zigzag uses data mining to identify similar behavioral sequences. Arbitrum based allotment on onchain activity and used community detection algorithms like Louvain to identify Sybil clusters.

Trusta‘s AI-ML Sybil Resistance Framework

The Sybils automate interactions across their accounts using bots and scripts. This causes their accounts to cluster together as malicious communities. Trusta’s 2-phase AI-ML framework identifies Sybil communities using clustering algorithms:

- Phase 1 analyzes asset transfer graphs (ATGs) with community detection algorithms like Louvain and K-Core to detect densely connected and suspicious Sybil groups.
- Phase 2 computes user profiles and activities for each address. K-means refines clusters by screening dissimilar addresses to reduce false positives from Phase 1.

In summary, Trusta first uses graph mining algorithms to identify coordinated Sybil communities. Then additional user analysis filters outliers to improve precision, combining connectivity and behavioral patterns for robust Sybil detection.

Phase I: Community Detection on ATGs

Trusta analyzes asset transfer graphs

(ATGs) between EOA accounts. Entity addresses such as bridge, exchanges, smart contracts are removed to focus on user relationships. Trusta has developed proprietary analytics to detect and remove hub addresses from the graphs. Two ATGs are generated:

1. The general transfer graph

with edges for any token transfer between addresses.

1. The gas provision network

where edges show the first gas provision to an address.

The initial gas transfer activates new EOAs, forming a sparse graph structure ideal for analysis. It also represents a strong relationship as new accounts depend on their gas provider. The gas network’s sparsity and importance makes it valuable for Sybil resistance. Complex algorithms can mine the networks while gas provision links highlight meaningful account activation relationships.

[

1130×621 50.3 KB

](https://ethresear.ch/uploads/default/original/2X/e/ea699ba6efc229ba29580ba1ea56df75ca0cadfc.png)

ATG patterns detected as suspicious Sybil clusters

Trusta analyzes asset transfer graphs to detect Sybil clusters through:

1. Clusters are generated by partitioning ATGs into connected components like P1+P2. Community detection algorithms then break down large components into densely connected subcommunities, like P1 and P2 with few edge cut, to optimize modularity.
2. Trusta identifies Sybil clusters based on known attack patterns, shown in the diagram
3. The star-like divergence attacks: Addresses funded by the same source
4. The star-like convergence attacks: Addresses sending funds to the same target
5. The tree-structured attacks: Funds distributed in a tree topology
6. The chain-like attacks: Sequential fund transfers from one address to the next in a chain topology.

Phase 1 yields preliminary Sybil clusters based solely on asset transfer relations. Trusta further refines results in Phase 2 by analyzing account behavior similarities.

Phase II: K-Means Refinement Based on Behaviour Similarities

Transaction logs reveal address activity patterns. Sybils may exhibit similarities like interacting with the same contracts/methods, with comparable timing and amounts. Trusta validates Phase 1 clusters by analyzing onchain behaviors across two variable types:

Transactional variables

: These variables are derived directly from on-chain actions and include information such as the first and latest transaction dates and the protocols or smart contracts interacted with.

Profile variables

: These variables provide aggregated statistics on behaviors such as interaction amount, frequency, and volume.

[

768×554 20.3 KB

](https://ethresear.ch/uploads/default/original/2X/4/4c8d6cd358cd9085d58414054d9eab90cd957523.png)

A K-means-like procedure to refine Sybil clusters

To refine the preliminary cluster of Sybils using the multi-dimensional representations of addresses behaviors, Trusta employs a K-means-like procedure. The steps involved in this procedure are repeated until convergence, as shown in the diagram:

Step 1: Compute the Centroid

of the clusters:

1. For continuous variables, calculate the mean of all the addresses within each cluster.
2. For categorical variables, determine the mode of all the addresses within each cluster.

Step 2: Refine the cluster by excluding the addresses that are far from the Centroid by a predefined threshold:

1. Addresses that are located far from the Centroid, beyond a specified threshold, are excluded from the cluster.
2. The cluster membership is then updated or refreshed based on the refined set of addresses.

These two steps are iteratively performed until convergence is achieved, resulting in refined clusters of Sybils.

Examples

Within the 2-Phase framework, we have identified several example Sybil clusters on Ethereum. These clusters are not only visualized via ATGs, but we also provide reasoning based on the behavioral similarities among the addresses in each cluster. The three clusters can be found via the link.

StarLike Asset Transfer Graph

Cluster 1 has 170 addresses which have completed 2 interactions on Ethereum, namely deposit

and purchase

. The two interactions all happened on Dec 5, 2021 and Feb 26, 2023. All the addresses got funded for the first time from the Binance address.

[

1400×827 253 KB

](https://ethresear.ch/uploads/default/original/2X/7/783f9f491e07426e7e26e32ff94d70c09d7ba248.jpeg)

[

1400×333 119 KB

](https://ethresear.ch/uploads/default/original/2X/9/91b8512730041675d9a4f8474e01147aedff8ab3.png)

ChainLike Asset Transfer Graph

Cluster 2 has 24 addresses which have completed a sequence of similar interactions on Ethereum.

[

1280×774 21.5 KB

](https://ethresear.ch/uploads/default/original/2X/5/5b75344baeab900e9b56551bcf64fe6155485cf3.png)

[

1280×569 266 KB

](https://ethresear.ch/uploads/default/original/2X/1/1c79294e749ca0f0495a423b1e5230d1f8def668.png)

TreeLike Asset Transfer Graph

Cluster 3 has 50 addresses which could be regarded as 2 sub-clusters, performing a sequence of similar interactions on Ethereum respectively.

[

1400×840 42.8 KB

](https://ethresear.ch/uploads/default/original/2X/d/db2e2ad83fe7379a891446e6e1054578fa9c623a.jpeg)

[

1280×405 183 KB

](https://ethresear.ch/uploads/default/original/2X/c/c2715ba4fd2b63c73b33ce5baae8702e2e68fe26.png)

[

1280×399 185 KB

](https://ethresear.ch/uploads/default/original/2X/9/9056bc8c379cf22ad3c608aaddde57f324a6ffad.png)

Discussion

The clustering-based algorithms for Sybil resistance are the optimal choice at this stage for several reasons:

1. Relying solely on historical Sybil lists like HOP and OP Sybils is insufficient because new rollups and wallets continue to emerge. Merely using previous lists cannot account for these new entities.
2. In 2022, there were no benchmark Sybil labelled data sets available to train a supervised model. Training on static Sybil/non-Sybil data raises concerns about the precision and recall of the model. Since a single dataset cannot encompass all Sybil patterns, the recall is limited. Additionally, misclassified users have no means to provide feedback, which hampers the improvement of precision.
3. Anomaly detection is not suitable for identifying Sybils since they behave similarly to regular users.

Therefore, we conclude that a clustering-based framework is the most suitable approach for the current stage. However, as more addresses are labeled, Trusta will certainly explore supervised learning algorithms such as deep neural network-based classifiers.