STEALTH

Bitcoin Wallet Privacy Analyzer

A read-only audit engine that surfaces wallet exposure at the UTXO level before funds move.

No keys

UTXO-level findings

Self-hostable

The Problem

VISIBILITY GAP

Bitcoin privacy leaks are invisible to users

- Companies like **Chainalysis** can analyze wallet privacy
- **Users cannot**
- People may expose: full transaction history, identity links, and behavioral fingerprints

Companies can analyze your privacy better than you can.

Why This Happens

Privacy is broken by **patterns**, not hacks

Common wallet patterns that leak privacy:

- Multi-input transactions (CIOH / consolidation)
- Combining coins
- Address reuse
- Sending change to same input address
- Dust UTXOs
- Exchange linkage / taint signals

Visibility Imbalance

Chainalysis users can see wallet-linkage signals that the average user cannot see about themselves.

Chainalysis
user

Privacy Parity

With Stealth, users gain visibility closer to institutional-grade analysis.

Chainalysis

user

STEALTH

What Stealth Does

INPUT

- Paste a wallet descriptor
- Supports `wpkh`, `pkh`, `sh(wpkh)`, `tr`, multisig

OUTPUT

- Structured findings plus warnings
- Type, severity, description, and evidence
- Severity badges mapped directly from detectors

```
# one click
wpkh([xpub...]/0/*) → Analyze
```

`⟹ Full report with actionable, spend-aware insights`

Vulnerabilities Detected

DETECTOR TYPE	MEANING
`ADDRESS_REUSE`	Repeated receive address links payment history
`CIOH`	Multi-input ownership clustering signal
`DUST` / `DUST_SPENDING`	Dust + normal co-spend linkage pattern
`CHANGE_DETECTION`	Payment and change outputs become distinguishable
`CONSOLIDATION` / `CLUSTER_MERGE`	Input histories merged into one traceable cluster
`SCRIPT_TYPE_MIXING`	Mixed script families create a wallet fingerprint
`UTXO_AGE_SPREAD`	Old/new spread leaks dormancy behavior
`EXCHANGE_ORIGIN`	Probable exchange withdrawal origin signature
`TAINTED_UTXO_MERGE`	Tainted + clean merge propagates contamination
`BEHAVIORAL_FINGERPRINT`	Consistent transaction style re-identifies wallet

How It Works

01

Parse

- Extract addresses from descriptor
- Normalize all common formats

02

Fetch

- Load on-chain history per address
- Use Bitcoin node or indexed API source

03

Analyze

- Apply privacy heuristics and warnings
- Flag each UTXO with findings and evidence

Architecture

```
stealth/
|-- frontend/   # React + Vite: input, loading, report
`-- backend/    # Java/Quarkus: parsing, chain data, analysis
```

SECURITY MODEL

Read-only

No private keys, no descriptor storage, no hidden transmission path.

DEPLOYMENT

Self-hostable

Point to your own node for maximum privacy and deterministic trust.

Demo Flow

1. **Input** Paste descriptor and trigger analysis
2. **Load** Fetch + parse + detect in one pipeline
3. **Report** Summary bar: findings / warnings / transactions
4. **Inspect** Expand finding cards for severity and evidence payloads

Demo

Roadmap

EXPANDED HEURISTICS

- `LEGACY_SCRIPT_EXPOSURE` — old script usage (`p2pkh` / nested-only flows) shrinking anonymity set
- `ADDRESS_GAP_LEAK` — sparse derivation usage exposing wallet generation behavior
- `AMOUNT_FINGERPRINT` — repeated denomination templates across spends
- `TIME_PATTERN_FINGERPRINT` — recurring timing cadence linking sessions

IMPROVEMENTS

- **Mainnet Support**
- **Mobile Support**
- **Cluster Visualization**
- **One-click solution**

Roadmap detectors are additive and keep the same read-only, no-key security model.

THANK YOU

STEALTH

Bitcoin Wallet Privacy Analyzer

Protect privacy before you broadcast intent.

Appendix — Supported Descriptors

- `wpkh(...)` — native SegWit
- `pkh(...)` — legacy
- `sh(wpkh(...))` — nested SegWit
- `tr(...)` — Taproot
- Multisig variants

All analysis relies only on publicly available on-chain data.