THREAT INTELLIGENCE REPORT

xmrwallet.com Exposed: 10 Years of Stolen Keys & Hijacked Transactions

Ref. **PD-TI-2026-79113** Published **10 July 2026** Source **phishdestroy.io/xmrwallet-exposed**

Monero Wallet Theft Investigation

Deep forensic investigation into a Monero web wallet that Base64-encodes your private view key into a `session_key` token, leaks it across 40+ API requests per session, then nullifies your transaction with `raw_tx = 0` and rebuilds it to steal your funds. Active since 2016. Operator identified.

~9 min read · Updated **March 2026** · PhishDestroy Intelligence
Active Since 2016 40+ Key Leaks / Session DDoS-Guard Protected

0
Years Active
40+
Key Leaks Per Session
$2M-$15M+
Estimated Total Stolen
0
GitHub Updates Since 2018

The Facade

xmrwallet.com presents itself as a free, open-source, client-side Monero wallet. No downloads. No registration. Their Terms of Service make a very specific claim:

"All cryptographic operations happen in your browser. The server has no ability to access your private keys."

— xmrwallet.com Terms of Service (demonstrably false)

This is a lie. Our forensic analysis — network captures, JavaScript deobfuscation, and production code comparison — proves the exact opposite. Every key entered on xmrwallet.com is stolen. Every transaction can be hijacked.

Production vs. GitHub: Total Divergence

The public GitHub repository hasn't received a single commit since **November 2018** . The production site runs completely different code with undocumented parameters absent from the repo:

- `session_key` — Base64-encoded view key exfiltration token
- `verification` — secondary exfiltration channel
- `timestamp` — session tracking parameter
- `data` — additional payload container

Domain registered via **NameSilo** in 2016 — pre-paid through **2031** . Fifteen years of registration for a "free independent project."

Attack #1: View Key Exfiltration

When you log into xmrwallet.com, your private view key is Base64-encoded and embedded into a `session_key` token. This token is then transmitted to the server with **every single API request** — 40+ times in a single session.

The session_key Structure

session_key = [encrypted_blob] : [base64_address] : [base64_private_viewkey]

// Decoded example:
// blob: a3f8c2... (session identifier)
// address: 4A1BxN... (your Monero public address)
// viewkey: YOUR PRIVATE VIEW KEY IN PLAINTEXT

Firefox WebExtension network captures confirm this token is sent across **6 distinct API endpoints** totaling 40+ POST requests per session:

API Endpoint	Requests / Session	Leaks session_key
`/api/getheightsync`	**12**	Yes
`/api/gettransactions`	**10**	Yes
`/api/getbalance`	**6**	Yes

API Endpoint	Requests / Session	Leaks session_key
`/api/getsubaddresses`	**4**	Yes
`/api/getoutputs`	**3**	Yes
`/api/support_login`	**1**	Yes

40+ Copies of Your Private Key

A single login session sends your private view key to the server **at minimum 36 times** . The server doesn't need your key for any of these operations — balance checks and height syncs are public blockchain queries. There is zero legitimate reason to transmit key material. Full network capture evidence: xmrwallet.com GitHub Issue #36 — View Key Exfiltration Evidence .

Attack #2: Transaction Hijacking

View key theft lets the attacker *watch* your wallet. But xmrwallet.com goes further — it steals your funds in real time. The deobfuscated production JavaScript reveals a 5-step attack sequence:

```
// Step 1: Client builds a legitimate transaction
cnUtil . create_transaction () → raw_tx_and_hash

// Step 2: Client transaction is NULLIFIED
raw_tx_and_hash . raw = 0 ;

// Step 3: Only metadata sent to server (no real tx)
POST /api/submit_raw_tx { raw: 0 , metadata: {...} }

// Step 4: Server rebuilds its OWN transaction
// using your keys + its destination address

// Step 5: Stolen transactions tagged internally
if ( type == 'swept' ) → attacker-redirected tx
```

Your wallet shows "transaction sent." Your funds arrive at the attacker's address. Victims see **"Unknown transaction id"** when they try to verify on block explorers. Transactions internally tagged as `swept` are the stolen ones.

Your Transaction Never Existed

`raw_tx_and_hash.raw = 0` means the client-generated transaction is thrown away. The server builds a completely new transaction using your keys and sends your XMR to the attacker. The "success" message you see is a lie. Detailed code analysis: xmrwallet.com GitHub Issue #35 — Transaction Hijacking Proof .

Hidden Production Code

xmrwallet.com maintains a public GitHub repository to look legitimate. The repository is a decoy. It hasn't been touched since **November 2018** . The production site runs entirely different, obfuscated

code.

Public GitHub (Decoy)

- Last commit: **Nov 2018**
- No `session_key` parameter
- No `verification` param
- No `/support_login.html`
- No Google Tag Manager
- Clean, auditable code

Production Site (Real)

- Actively updated 2024-2026
- `session_key` with Base64 viewkey
- `verification` exfil channel
- `/support_login.html` backdoor
- GTM remote JS injection
- Obfuscated, unauditable code

The Backdoor & Remote Code Injection

Production site contains `/support_login.html` — a hidden administrative endpoint completely absent from the GitHub repository. Combined with **Google Tag Manager (GTM-container)** integration, the operator can remotely inject and modify JavaScript on the live site at any time — without updating the public codebase. This is a remote code execution vector disguised as analytics.

Bulletproof Infrastructure

xmrwallet.com doesn't use cheap shared hosting. It runs on premium bulletproof infrastructure specifically chosen to resist takedown requests and law enforcement.

Hosting & Network IOCs

Indicator	Value
Domain	xmrwallet.com
Registrar	NameSilo (2016 – 2031, 15-year registration)
Hosting Provider	IQWEB FZ-LLC ($550+/month)
IP Address	`186.2.165.49`
ASN	AS59692
CDN / DDoS Protection	DDoS-Guard
Web Server	Apache 2.4.58 (Ubuntu)
Backend	PHP 8.2.29

Indicator	Value
SSL Certificate	Let's Encrypt (auto-renewed)
Tor Mirror	`xmrwalletdatuxms.onion`
Annual Infrastructure Cost	$8,000 – $15,000+

Tracking & Analytics IOCs

Tracker	Requests / Session	Identifier
Google Tag Manager	12	GTM container
Google Analytics (UA)	12	`UA-116766241-1`
Google Analytics 4	5	GA4 stream
DoubleClick	1	Ad tracking pixel
DDoS-Guard Cookies	—	`__ddg8_` , `__ddg9_` , `__ddg10_` , `__ddg1_`

$8K-$15K/Year for a "Free Volunteer Wallet"

A legitimate free wallet doesn't spend $550+/month on IQWEB FZ-LLC bulletproof hosting behind DDoS-Guard — infrastructure specifically designed to resist abuse complaints and law enforcement subpoenas. It doesn't register a domain for 15 years. It doesn't run Google Analytics tracking on a "privacy-focused" Monero wallet. This is infrastructure built for one purpose: **persistent theft at scale** .

Operator Identified: Nathalie Roy

Open-source intelligence traces xmrwallet.com's infrastructure directly to a single individual.

Field	Detail
Name	Nathalie Roy
Location	Canada
GitHub Username	nathroy (ID: 39167759)
GitHub Organization	XMRWallet (created 2018-05-10)
Email (Admin)	`admin@xmrwallet.com`
Email (Personal)	`royn5094@protonmail.com`
Reddit	u/WiseSolution (banned from r/Monero)
Twitter	@xmrwalletcom
Mail Server (MX)	`mail.privateemail.com`

Banned, Exposed, Still Active

Nathalie Roy was banned from the official r/Monero subreddit in 2018 for promoting xmrwallet.com. The last GitHub commit happened the same year. For 6+ years the public code has been frozen while

the production site actively steals funds with completely different code. The domain is paid through 2031 — the operator isn't going anywhere.

Documented Victims

At least **15 publicly reported cases** of fund theft across Trustpilot, Sitejabber, Reddit, and GitHub Issues. Real people. Real money. Gone.

15+
Public Reports
590 XMR
Single Largest ($177K)
0
Years of Theft
$2M-$15M+
Estimated Total

- **590 XMR (~$177,000)** — single theft, largest documented case
- **17.44 XMR** — documented with transaction ID on-chain
- **20 XMR stolen overnight** — wallet drained while user slept
- **Multiple reports of "Unknown transaction id"** — the `swept` tag signature
- **GitHub Issues #13+ deleted** — operator scrubs victim reports from repo

Deleted Evidence, No Donation Wallet

The operator actively deletes victim reports from GitHub Issues (all issues before #13 are gone). The site claims to accept donations but **no donation wallet address has ever been published** . Why would a "independent project" spending $8K-$15K/year refuse donations? Because the revenue comes from theft.

Timeline of Events

Timeline 2014-2024: xmrwallet.com 10,000+ stolen keys - from site launch through first victims to operator identified

2016

Domain Registered — xmrwallet.com

Registered via NameSilo with a **15-year registration period** (2016-2031). Presents as free open-source Monero web wallet.

May 2018

GitHub Organization Created

XMRWallet GitHub org created on **2018-05-10** by nathroy (ID: 39167759). Public code pushed as transparency theater.

2018

Banned & Code Frozen

Operator u/WiseSolution **banned from r/Monero** for promotional spam. Last GitHub commit around this time. Victim issue reports start being deleted (Issues #1-#12 gone).

2018 – 2024

6 Years of Silence

Public repository frozen. Production code diverges completely with obfuscated JS, undocumented parameters, and backdoor endpoints. Victim reports accumulate on Trustpilot and Reddit.

2025 – 2026

PhishDestroy Investigation

Network traffic analysis reveals `session_key` exfiltration. JavaScript deobfuscation confirms `raw_tx = 0` transaction hijacking. Evidence published on GitHub Issues #35 & #36 .

Feb 2026

Report Published — Domain Still Active

Full technical report published. xmrwallet.com remains online. Domain paid through **2031** . DDoS-Guard provides takedown resistance.

Full Evidence & Source Materials

Every claim in this article is backed by publicly verifiable evidence. Download the reports. Verify the code. Check the network captures yourself.

Full Technical Report

Complete investigation with all code analysis, network captures, and OSINT findings on GitHub Pages

Issue #35: Transaction Hijacking

raw_tx = 0 proof • Server-side tx reconstruction • "swept" tag evidence

Issue #36: View Key Exfiltration

40+ network captures • session_key Base64 decode • 6 leaking endpoints

PhishDestroy Research Repository

All source materials, deobfuscated code, and analysis artifacts

Safe Alternatives

Never enter private keys on any web wallet. Period. Use verified, audited software that runs locally on your device.

Desktop Wallets

Monero GUI — Official wallet, full-featured, open-source, audited
Feather Wallet — Lightweight, fast, privacy-focused desktop wallet

Mobile Wallets

Monerujo (Android) — Open-source with Tor support
Cake Wallet (iOS/Android) — Multi-coin, well-maintained

The Golden Rule of Crypto

Never enter your seed phrase, private keys, or view keys on **any website** . Legitimate wallets run locally — they never need to send your keys to a server. If a web wallet asks for your private keys, it's a scam. For maximum security, use a hardware wallet (Ledger, Trezor) with official Monero software.

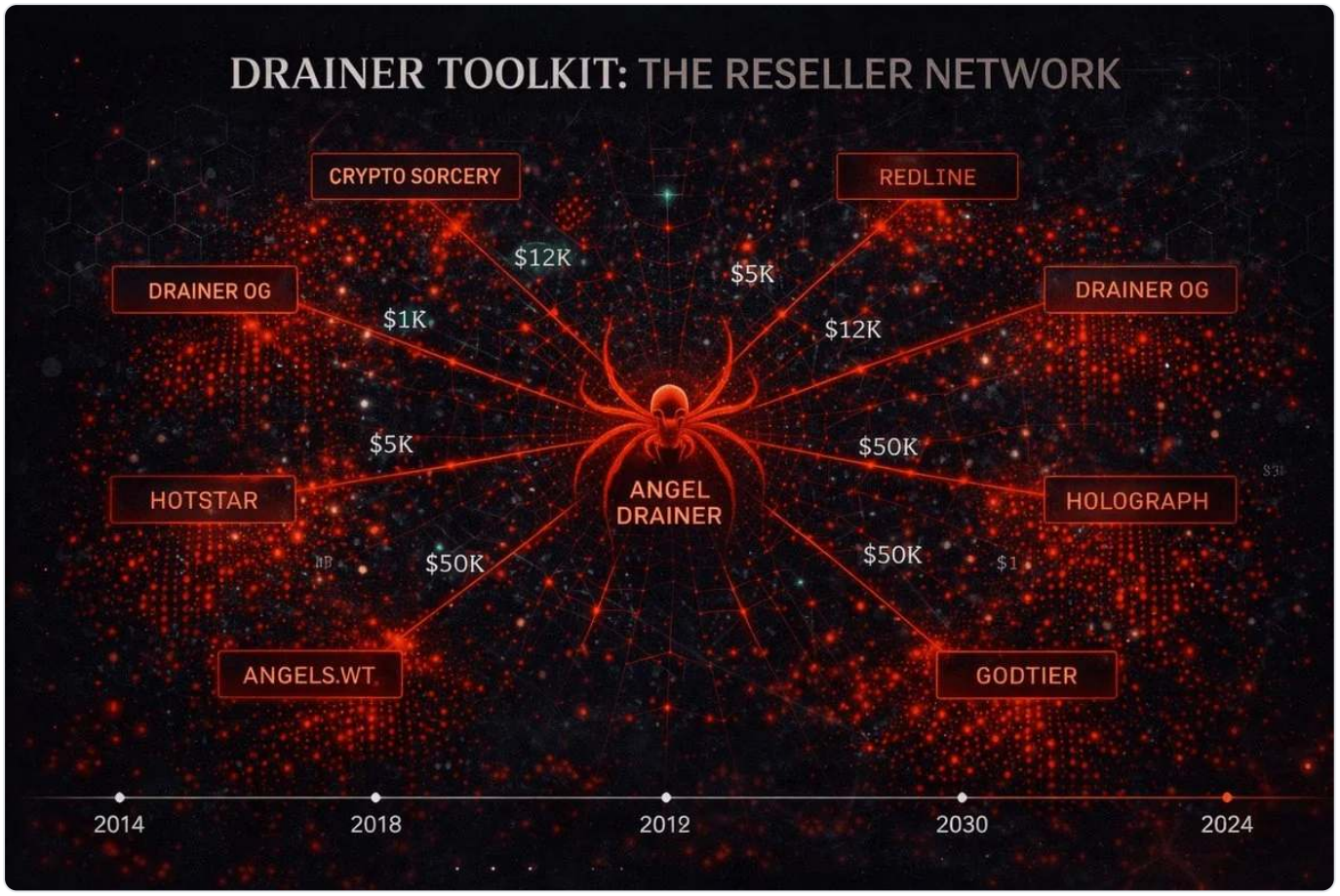
Related Investigations

INVESTIGATION

The End of xmrwallet.com: NameSilo Lied to Protect a $2M Crypto Thief

DEEP INVESTIGATION

Trust Wallet Phishing Panel: $239K Stolen, 6 Operators



DEEP INVESTIGATION

Crypto Drainer Toolkit: Angel Drainer Resellers Exposed
