VALIDIFY PROTOCOL

White Paper v2.0

End-to-End Verification for Web3 Communities

Discord Bots Protect Servers. Validify Protects Users.

December 2025

validifyprotocol.com

 |

Table of Contents

Note: Update the Table of Contents in Word by right-clicking and selecting 'Update Field'

1. Executive Summary

Web3 communities lost $2.5 billion to security incidents in 2024, with $494 million attributed to phishing attacks alone—a 67% increase year-over-year. The attack pattern is consistent: compromise one admin account, post one malicious link, drain hundreds of wallets.

Current security solutions share a fatal flaw: they are reactive. Blocklist-based bots detect malicious links ***after*** they're posted. By definition, a new phishing domain isn't on any blocklist. More critically, existing protection ends when users click a link and leave the Discord server—precisely where the actual wallet drain occurs.

Validify introduces end-to-end verification: a multi-signature approval system that prevents unauthorized links from being posted, combined with a browser extension that verifies links at the point of wallet connection. We don't just protect servers—we protect users throughout their entire journey, from announcement to transaction.

Core Innovation:

- **Multi-Signature Verification:** M-of-N admin approval required before any link-containing announcement is published
- **Browser Extension:** Verification continues when users leave Discord, warning them before wallet connection
- **On-Chain Attestations:** Immutable record of every approval decision via Ethereum Attestation Service (EAS)
- **Cross-Community Intelligence:** Shared threat data that gets stronger with every participating community

Key Differentiator: Discord bots protect servers. Validify protects users—from announcement to transaction.

2. The Problem: Compromised Trust

2.1 The Attack Pattern

The anatomy of a Discord-based Web3 attack is remarkably consistent:

1. **Reconnaissance:** Attackers identify high-value targets—NFT projects with upcoming mints, DeFi protocols, or DAOs with significant treasuries.
2. **Credential Compromise:** Through phishing, SIM swaps, malware, or social engineering, attackers gain access to a single admin account.
3. **Malicious Announcement:** The compromised admin posts a fake mint link, airdrop claim, or urgent security notice to an announcements channel.
4. **User Migration:** Trusting users click the link and leave Discord, arriving at a convincing phishing site.
5. **Wallet Drain:** Users connect their wallets and sign malicious transactions. The attack is complete.

The entire attack takes minutes. By the time moderators react and delete the malicious post, hundreds of wallets may already be compromised.

2.2 Why Current Solutions Fail

The Web3 security market offers numerous Discord security bots, but they share fundamental limitations:

Reactive Detection

Blocklist-based systems only recognize threats that have been previously identified. Attackers generate fresh domains for each campaign—by definition, these aren't on any blocklist. The system detects the threat only after damage is done.

Protection Boundary Problem

Existing security bots protect the Discord server itself—they can detect suspicious activity, lock channels, or flag malicious links after posting. But their protection **ends at the server boundary**. When a user clicks a link and leaves Discord, they're on their own. The actual wallet drain happens in the browser, not Discord—exactly where current solutions provide zero protection.

Single Point of Failure

Discord's permission model gives administrators broad authority. A compromised admin has full posting privileges—no additional verification is required. Existing solutions add 2FA to admin actions within Discord, but they cannot prevent a legitimately authenticated (but compromised) admin from posting.

Isolated Defense

When one community is attacked, the threat intelligence dies with that incident. There's no systematic way to share attack patterns, malicious domains, or attacker behaviors across the Web3 ecosystem. Each community defends alone.

2.3 The Protection Gap

The critical insight is this: the attack doesn't succeed in Discord. It succeeds in the browser. Current security solutions create a false sense of safety by protecting the server while leaving users vulnerable at the actual point of attack.

Current Security Bots	Validify Protocol
1. Malicious link posted	1. Multi-sig blocks unauthorized links
2. User clicks → leaves Discord	2. Verified link posted with attestation
3. User on fake site (NO PROTECTION)	3. Browser extension verifies before connect
4. Wallet drained	**4. User protected**

3. The Validify Solution

Validify is not another security bot. It's an end-to-end verification protocol that protects users throughout their entire journey—from the moment an announcement is created to the moment they're about to connect their wallet.

3.1 Multi-Signature Verification

The same security model that protects billions in crypto treasury funds now protects Discord announcements.

How It Works:

1. **Admin Submits:** An administrator attempts to post a link-containing message. Validify intercepts and holds the content.
2. **Approvers Notified:** Designated approvers receive the message preview and a unique, time-limited cryptographic code.
3. **Threshold Reached:** Once the configured threshold (e.g., 3 of 5 admins) submit their codes, verification is complete.
4. **Published & Attested:** The verified message is published with a Validify badge. All approvals are recorded on-chain.

Key Design Principles:

- **Urgency is the red flag—friction is the feature.** Legitimate announcements (mints, partnerships) are planned in advance. "I need to post this NOW without verification" is exactly what a compromised account would say.
- **Configurable thresholds** allow each project to set their security/convenience balance.
- **Hardware wallet binding** ensures approvers prove identity through cryptographic signatures, not just Discord accounts.

3.2 Browser Extension

This is Validify's critical differentiator. Protection doesn't end when users leave Discord—it follows them to the transaction.

How It Works:

When a user visits a link that claims to be from a Web3 project, the Validify browser extension automatically checks:

- **Attestation Check:** Was this link approved through Validify's multi-sig verification?
- **Contract Verification:** Does the smart contract address match what the project registered?
- **Threat Indicators:** Are there any known threats associated with this domain or contract?

User Interface:

Users see a clear trust indicator before connecting their wallet:

- 🟢 **GREEN:** Verified by Validify—link was approved by multiple admins
- 🟡 **YELLOW:** Not verified—proceed with caution, no Validify attestation found

- 🔴 **RED:** Threat detected—known malicious indicators present

The extension is free for users. Value flows from projects who want their legitimate links to display as verified.

3.3 On-Chain Attestation Layer

Every security event is recorded on Ethereum using the Ethereum Attestation Service (EAS), creating an immutable audit trail.

What Gets Attested:

- Approval signatures from each admin who verified an announcement
- Message content hashes (not the content itself, preserving privacy)
- Timestamps and verification window compliance
- Threat reports and community flags

Benefits:

- **Verifiable security history** that projects can point to as evidence of security practices
- **Dispute resolution evidence** for when questions arise about who approved what
- **Training data** for AI-driven anomaly detection in future phases
- **Privacy-preserving identifiers** that prevent enumeration of protected communities

3.4 Cross-Community Threat Intelligence

When one community encounters a threat, every community is protected. This is the shared immune system Web3 never had.

How It Works:

- Threat reports are anonymized and shared across the network
- Behavioral patterns (not individual data) are analyzed to identify emerging attack vectors
- Risk scores are computed based on network-wide threat intelligence
- Real-time alerts propagate to all participating communities

Network Effect: More participating communities = more threat data = better protection for everyone. Defense gets stronger as adoption grows.

4. The Two-Sided Network Effect

Validify creates a powerful two-sided network that benefits both projects and users:

4.1 For Projects

- Use Validify to attest their legitimate announcements
- Their verified links display as "trusted" to users with the extension
- Build verifiable security reputation through on-chain attestation history
- More users with the extension = more value from verification

4.2 For Users

- Install the free browser extension
- See verification status before connecting wallet to any site
- Protected from unverified links even if they don't know about Validify
- More projects using Validify = more value from the extension

4.3 The Flywheel

Projects sign up because users have the extension → Users install because projects use Validify → More projects sign up → More users install. This creates a self-reinforcing adoption cycle that compounds over time.

5. Technical Architecture

5.1 System Components

Discord Bot

- Intercepts link-containing messages in protected channels
- Manages approval workflows and verification windows
- Publishes verified messages with attestation badges
- Minimum permissions required (no Administrator privilege needed)

Browser Extension

- Chrome extension (Firefox/Safari planned)
- Queries attestation status for visited domains
- Displays trust indicators before wallet connection
- No user data collection—verification is stateless

Attestation Service

- Built on Ethereum Attestation Service (EAS)
- Stores approval signatures and verification metadata
- Privacy-preserving hashed identifiers
- Queryable by browser extension for verification status

Threat Intelligence API

- Aggregates threat reports from participating communities
- Computes risk scores based on behavioral patterns
- Provides real-time threat alerts to all nodes

5.2 Security Model

Validify is designed with the principle of minimum trust:

- Never asks for seed phrases or private keys
- Cannot post on behalf of projects—can only prevent unauthorized posts
- Even if Validify systems were compromised, attackers couldn't use it to publish malicious content
- Worst-case scenario is verification going offline, not active exploitation
- All smart contracts undergo third-party security audits before mainnet

6. 2026 Development Roadmap

Q1 2026: Foundation

- Deploy multi-signature verification protocol to Ethereum mainnet
- Launch Discord integration with admin action monitoring
- Complete third-party smart contract security audit
- Begin onboarding founding communities

Q2 2026: Intelligence

- **Launch Validify browser extension (Chrome)**
- Activate cross-community threat intelligence network
- Launch on-chain attestation layer for verifiable security events
- Expand protocol to Telegram ecosystem

Q3 2026: Expansion

- Release public API and developer SDK
- Launch Project Trust Score with on-chain verification badges
- Establish security researcher partnership program
- Complete first third-party platform integration

Q4 2026: Prediction

- Deploy AI-driven anomaly detection
- Introduce predictive risk scoring for pending announcements
- Launch DAO governance for protocol standards
- Publish 2027 roadmap including token utility framework

7. Token Economics

Token utility is planned for 2027, after the core protocol proves value.

Rationale:

We're focused on building real security infrastructure first. Token design will be informed by actual usage patterns, not speculation. Launching a token before product-market fit creates misaligned incentives and distracts from the core mission of protecting Web3 communities.

Planned Utility (2027):

- Subscription payments for premium features
- Staking for enhanced threat intelligence access
- Governance voting on protocol standards and threat classifications
- Rewards for security researchers and threat reporters

Full token economics will be published in the 2027 roadmap, informed by real protocol usage data.

8. Pricing Model

Final pricing will be announced closer to mainnet launch. The model is designed to make serious security accessible to communities of all sizes.

Principles:

- Browser extension is free for users
- Project pricing scales with community size and feature requirements
- Founding communities receive extended free protection periods
- USD pricing in 2026; token payment option in 2027

9. Conclusion

The Web3 security landscape is fundamentally broken. Current solutions protect servers while leaving users vulnerable at the actual point of attack—the browser, where wallet drains occur.

Validify introduces a new paradigm: end-to-end verification that follows users from Discord announcement to wallet connection. By combining multi-signature approval, browser-side verification, on-chain attestations, and cross-community threat intelligence, we create protection that gets stronger as adoption grows.

> **Discord bots protect servers. Validify protects users.**

We're building the security infrastructure Web3 deserves—one that prevents attacks rather than detecting them after the damage is done. Join us in creating a safer ecosystem for everyone.

10. Contact & Resources

Website: validifyprotocol.com

Discord: discord.gg/validify

Twitter: @validifyprotocol

Email: contact@validifyprotocol.com

— End of Document —