

Web3 Security Checklist

Introduction

Overview

The blockchain space has seen over **\$6 billion in hacks and exploits** in recent years, from smart contract vulnerabilities to infrastructure breaches. This security checklist, derived from **real-world incidents and industry best practices**, covers three critical defense layers: smart contract security, infrastructure protection, and audit strategy.

Whether you're launching a DeFi protocol, NFT platform, or any Web3 application, implementing these measures can help protect your project from common attack vectors and insider threats.

Table of content

1 Smart Contract Security	3
Code quality, testing practices, and architectural patterns.	
2 Infrastructure Security	5
Access control, deployment processes, and role management	
3 Auditing Strategy	8
Multi-layered audit approach from specialized to comprehensive reviews	

Code Quality and Testing

☐ Implement mathematical invariants

Rule: Verify total balances match holdings.

Example: During the **Euler Finance hack (2023)**, attackers exploited a vulnerability to borrow assets far exceeding their collateral's actual value. An invariant checking that the total value of outstanding loans was always less than or equal to the collateral deposited would have prevented the loss of **\$197 million**.

☐ Unit test across all scenarios

Rule: Test every possible input.

Example: In the **TinyMan DEX hack**, testing withdrawals with same-token pairs (like BTC-BTC instead of BTC-USDC) would have revealed the vulnerability that led to a **\$3M loss**.

☐ Minimize contract endpoints

Rule: Develop smart contracts with a minimal number of endpoints.

Example: Instead of having one large farming contract with many features, break it into smaller contracts. A basic staking contract can be just **60 lines** instead of **1600**.

Architecture

☐ **Implement micro-contract pattern**

Rule: Separate asset custody functions from complex contract logic

Example: For farming contracts, use individual escrow accounts for each user's deposits separate from the reward distribution logic. This prevents the entire TVL from being at risk if the **reward logic is compromised**.

☐ **Keep dependencies up to date**

Rule: Regularly update and verify all smart contract dependencies

Example: The **\$320M Wormhole bridge hack** could have been prevented by updating to the latest Solana crate version, which had patched the vulnerability four months earlier.

Access Control

☐ **Enforce 2FA with hardware keys**

Rule: Use two-factor authentication with a physical key.

Example: Cream Finance's DNS hijacking could have been prevented with hardware-based 2FA - instead, shared credentials led to unauthorized DNS changes.

☐ **Never share credentials**

Rule: Enforce unique credentials for each team member.

Example: In 2024, **WazirX lost \$234.9M** due to compromised shared credentials. Unique access could have prevented unauthorized wallet access.

☐ **Implement comprehensive logging**

Rule: Log every DNS change, deployment, and code commit.

Example: The **BitPay wallet hack** through a malicious dependency could have been caught earlier if there was proper logging of package updates and changes.

Deployment Security

☐ **Automate the deployment process**

Rule: Production builds should only occur in CI.

Example: The **MyAlgo wallet hack** occurred because someone could manually modify code in the content delivery network. Automated deployments would have prevented this unauthorized modification.

☐ **Manage infrastructure as code**

Rule: Manage DNS and CDN within your infrastructure code.

Example: For unified access and logging, keep **DNS and content delivery** within the same provider, like AWS, instead of using separate services such as Cloudflare, to maintain **unified access control and logging**.

Role Management

☐ Define the CTO's responsibilities

Rule: The CTO should lead DevOps and control access.

Example: In one documented case, having 5+ people with admin access led to the inability to track down who made malicious changes. Instead, **limit superadmin privileges** to just the CTO.

☐ Rebuild infrastructure after leadership changes

Rule: Rebuild infrastructure from scratch when changing CTO.

Example: In June 2024, **Holograph Protocol lost millions** when a former contractor exploited leftover admin access. Rebuilding infrastructure after leadership changes could have prevented this.

Multi-Step Process

☐ **Step 1: Hire a freelance auditor**

Rule: Consider this option if involving another developer in smart contract development isn't feasible.

Example: Choose a **cost-effective freelance auditor** to identify major issues early. Even basic audits can prevent catastrophic failures - the CashIO protocol lost **\$28M** because they had **no audits at all**.

☐ **Step 3: Secure a reputable audit**

Rule: Ensure smart contract architecture is finalized before this stage.

Example: The final audit should focus exclusively on logic issues, with auditors not needing to address foundational design flaws. Architecture should be complete and stable at this point.

☐ **Step 2: Engage a specialized audit firm**

Rule: Work with a smaller auditing firm specializing in your niche or ecosystem.

Example: Projects on specific blockchains, like **Aptos** or **Substrate**, need auditors familiar with their unique features, such as distinct storage types and validation mechanisms.

Behind the Audit



Paweł Rejkowicz

Security Researcher

Has been auditing smart contracts since 2021, starting with Kudelsky Security and high-profile Solana projects like Magic Eden, Switchboard, and Solend. He later expanded to blockchains like Algorand, Flow, and Aptos, working with various custom languages.

Now focused on EVM-based blockchains, Paweł is impressed by the Foundry tool. With 10+ years in IT, he brings deep expertise in computer science and blockchain economics. He's experimenting with formal proofs for smart contracts and aiming for a PhD in mathematics.

Secure Your Smart Contracts Today

Protect your Web3 product from costly vulnerabilities with a comprehensive security audit from ULAM LABS.

[Contact us for an audit](#)

50+

Smart Contract Audits

Our extensive experience ensures your smart contracts are in expert hands.

 alephium

 SOLANA

 METAGRAVITY

 ethereum

APTOS 


Featured Audits



● Yamato

Euphrates

 TsunamiX

 SuperStable

xBacked

 deflex