PROTOFIRE

February 2026

Smart contract audit

rain.deploy

Content

Project description

The rain.deploy repository implements a Solidity library and tooling designed to manage the deterministic deployment of smart contracts across multiple EVM-compatible networks (such as Arbitrum, Base, Flare, and Polygon). Utilizing the Zoltu deterministic deployment proxy, the project ensures that contracts are deployed to identical addresses on all supported chains.

Unlike typical on-chain registries, this project focuses on secure "dev-ops" workflows within the Foundry environment. It provides logic to verify network dependencies, validate bytecode integrity (hash checks), and execute permissionless multi-chain deployments, ensuring that production environments match local compilations byte-for-byte.

Executive summary

Type	Library
Languages	Solidity
Methods	Architecture Review, Manual Review, Unit Testing, Functional Testing, Automated Review
Documentation	README.md
Repositories	https://github.com/rainlanguage/rain.deploy

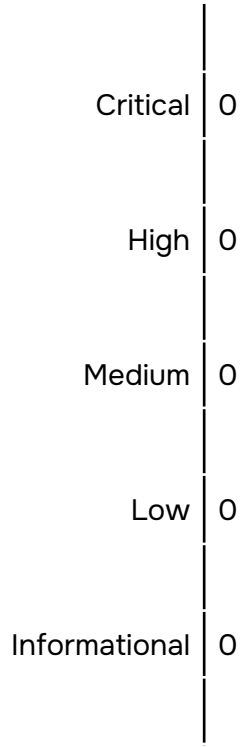
Reviews

Date	Repository	Commit
09/02/26	rain.deploy	1af8ca2a981c2f67844f343e224f4c7b1969de1c

Scope

Contracts
src/lib/LibRainDeploy.sol

Technical analysis and findings



Security findings

**** Critical

No critical severity issue found.

*** High

No high severity issue found.

** Medium

No medium severity issue found.

* Low

No low severity issue found.

General Risks / Assumptions

- Singleton Deployment Limitation: The library utilizes the Zoltu factory with a fixed salt (implied 0). This enforces a singleton pattern, meaning a specific bytecode configuration can only be deployed once per network. Users cannot deploy multiple instances of the exact same bytecode using this tooling without modifying the source or the library.

- Front-running / Griefing: Because deployment addresses are deterministic and known in advance, external actors can front-run the deployment transaction. While the library handles pre-existing code by skipping deployment (checking expectedAddress.code.length), a front-run deployment could still disrupt scripts that rely on the deployer being the msg.sender during the constructor phase, or cause unexpected revert noise in CI/CD pipelines.

- Factory Dependency: The system strictly requires the Zoltu deterministic deployment proxy (0x7A0D...) to exist at the specific hardcoded address on the target network. Deployment is impossible on any chain (e.g., a new private testnet or a fresh hard fork) where this factory has not yet been deployed.

Approach and methodology

To establish a uniform evaluation, we define the following terminology in accordance with the OWASP Risk Rating
Methodology:

	**Likelihood** Indicates the probability of a specific vulnerability being discovered and exploited in real-world scenarios
	**Impact** Measures the technical loss and business repercussions resulting from a successful attack
	**Severity** Reflects the comprehensive magnitude of the risk, combining both the probability of occurrence (likelihood) and the extent of potential consequences (impact)

Likelihood and impact are divided into three levels: High H, Medium M, and Low L. The severity of a risk is a blend of these two factors, leading to its classification into one of four tiers: Critical, High, Medium, or Low.

When we identify an issue, our approach may include deploying contracts on our private testnet for validation through testing. Where necessary, we might also create a Proof of Concept PoC to demonstrate potential exploitability. In particular, we perform the audit according to the following procedure:

	**Advanced DeFi Scrutiny** We further review business logics, examine system operations, and place DeFi-related aspects under scrutiny to uncover possible pitfalls and/or bugs
	**Semantic Consistency Checks** We then manually check the logic of implemented smart contracts and compare with the description in the white paper.
	**Security Analysis** The process begins with a comprehensive examination of the system to gain a deep understanding of its internal mechanisms, identifying any irregularities and potential weak spots.