Security Audit

February 2024

3DNS

Audited by
Ethan Bennett

Contents

About Darklinear Solutions

Darklinear Solutions provides unrivaled security for blockchain applications, from the bytecode to the browser. With years of experience in smart contract development and traditional software engineering, we find the bugs that others miss. Learn more at darklinear.com.

Introduction

3DNS is a web domain registrar built on Optimism. This review consists of issues discovered during the course of Cantina's competitive audit for 3DNS in February 2024. It does not represent a full and exhaustive audit of the protocol.

The findings described below are classified according to Cantina's standards.

Findings Summary

In addition to the two listed findings below, three low severity vulnerabilities were submitted. However, since Cantina only published the high and medium severity issues, the low severity findings have also been omitted from this report.

Classification	Finding
High risk (H-1)	Insufficient validation of `txData` risks loss of funds
Medium risk (M-1)	User maintains control of domain when offchain payments are cancelled

Findings

H-01: Commitments of type `TRANSFER` can be DOSed

Severity: High risk

Description: All commitments of type `TRANSFER` can be instantly revoked by any user, which would allow a malicious actor to block the transfer of any domain.

This exploit is possible because of two distinct vulnerabilities:

1. `CommitmentOrderflow.revoke` does not validate that `msg.sender` is the registrant of the commitment they want to revoke
2. `CommitmentStorage.initialize` does not set the `transferCommitmentHalfLife`, so it remains at the default value of `0` (note that none of the half life values can be changed after deployment)

Although Optimism's mempool is private, the commitment type of any commitment will be visible once the transaction has been included in a block — not to mention the value of `revokableAt_` in the `PendingCommitment` event, which will equal `block.timestamp` when a commitment of type `TRANSFER` has been made. By either means, it would be trivial for an attacker to identify a commitment they can DOS, or to set up a bot to `revoke` any `TRANSFER` commitment automatically.

The additional difficulty of frontrunning on Optimism also does not mitigate this risk, since the attacker would actually be backrunning `makeCommitmentV2` (as opposed to frontrunning `processCommitment`). The L2 would, however, keep transaction costs low enough for an attacker to conceivably block all attempted `TRANSFER` commitments.

Proof of Concept: This test demonstrates how any arbitrary account can `revoke` a valid transfer immediately after the commitment is made.

Recommendation: Both of the abovementioned issues should be fixed: `revoke` should authenticate `msg.sender`, and `initialize` should set `COMMITMENT_HALF_LIFE__TRANSFER` to a nonzero value.

The former might create a problem if the user is attempting to keep the `registrant` secret (by way of the `secretHash`), but `CommitmentStorage` could instead store a mapping of a commitment hash to the account that sent the `makeCommitmentV2` transaction. This would allow users to register a new domain from a throwaway account, maintaining the privacy of the `registrant` but also allowing for some validation in `revoke`.

M-01: User maintains control of domain when offchain payments are cancelled

Severity: Medium risk **Description:** In order to prevent fraud from offchain payments, `CommitmentOrderflow.issueDomain` locks the domains it creates for 30 days. This will prevent a user from transferring their domain during this period, but it does not restrict any other privileges associated with domain ownership. Locked or not, the user retains control of the token — and if their offchain payment is cancelled more than 30 days after registering the domain, even the existing restrictions can be avoided.

It is also likely that a 30 day period would be insufficient to deal with the problem of fraud: in the US, federal law mandates that banks allow at least 60 days to dispute charges, and most allow for it up to 120 days after the payment occurred.

Typically, 0.5-1% of credit card payments are disputed. That is a significant number of unpaid-for domains, particularly if these domains were registered for many years. And if word were to spread of the ease of this exploit, that percentage could be higher in practice, which could cause payment processors to refuse to work with 3DNS altogether.

Although chargebacks themselves are difficult to prevent entirely, steps could be taken to allow for such domains to be recovered by the protocol.

Recommendation: Restrictions on locked domains do not necessarily need to be tightened, but the protocol should have a way to revoke the domain token from a user if (and only if) their offchain payment is cancelled. Since it will not be possible to validate this on-chain, any admin-level token revocation power should be carefully restricted to locked domains.

Additional checks and balances on this authority would be great to include, but some amount of centralized control over payment clearance is unavoidable when offchain payments are involved. As long as this is limited to the period when a charge could be cancelled and to registrants who paid offchain, such admin functionality would not undermine the self-soverign nature of the legitimate domains represented in the protocol.