Enzyme GSN Forwarder Audit

Smart Contract Security Assessment

Oct 8, 2021

ABSTRACT

Dedaub was commissioned to perform a security audit on a modified version of GSN's Forwarder.sol smart contract on repository enzymefinance/gsn-trusted-forwarder, branch "master" at commit hash 187ca596138d9e22b68e6db7526c608888367358. The audit examined contract AcceptsContractSignaturesForwarder.sol but also any other material highly related to its functionality. Specifically, EIP-1271 and EIP-2770. Two auditors worked over the codebase for one day.

SETTING & CAVEATS

Ethereum GSN (Gas Station Network) abstracts away transactions' fees allowing for gasless clients, meaning clients able to perform transactions without having to pay for the corresponding fees in ETH.

A gasless client interacts with a dapp's relay server by creating a "meta-transaction", which essentially is a request for a transaction to be performed on its behalf. This request contains all the information needed for the transaction to be executed as well as the client's signature on this transaction. Later on, the protocol requires a "*Forwarder whose responsibility is to validate transaction signatures on-chain and expose the signer to the destination contract*". The meta-transactions Forwarder contract is specified in EIP-2770. Current implementation of GSN supports only meta-transactions originated by EOAs.

Smart contract AcceptsContractSignaturesForwarder is a modified version of GSN's Forwarder contract, which also supports meta-transactions originated by smart contracts. Due to lack of a private key, contracts cannot sign transactions, so AcceptsContractSignaturesForwarder relies upon EIP-1271 of Standard Signature Validation Method for Contracts to mitigate this problem. We found no security issues in the modified contract. It is important to clearly document that contracts that intend to perform meta-transactions need to comply with EIP-1271. Finally, support for sending ETH (either as part of the meta-transaction itself, or as a post-processing step) has been dropped, which simplifies the original Forwarder implementation.

VULNERABILITIES & FUNCTIONAL ISSUES

This section details issues that affect the functionality of the contract. Dedaub generally categorizes issues according to the following severities, but may also take other considerations into account such as impact or difficulty in exploitation:

Category	Description
CRITICAL	Can be profitably exploited by any knowledgeable third party attacker to drain a portion of the system's or users' funds OR the contract does not function as intended and severe loss of funds may result.
HIGH	Third party attackers or faulty functionality may block the system or cause the system or users to lose funds. Important system invariants can be violated.
MEDIUM	Examples: 01) User or system funds can be lost when third party systems misbehave. 02) DoS, under specific conditions. 03) Part of the functionality becomes unusable due to programming error.
LOW	Examples: 01) Breaking important system invariants, but without apparent consequences. 02) Buggy functionality for trusted users where a workaround exists. 03) Security issues which may manifest when the system evolves.

Issue resolution includes “dismissed”, by the client, or “resolved”, per the auditors.

CRITICAL SEVERITY

[No critical severity issues]

HIGH SEVERITY:

[No high severity issues]

MEDIUM SEVERITY:

[No medium severity issues]

LOW SEVERITY:

[No low severity issues]

OTHER/ ADVISORY ISSUES:

This section details issues that are not thought to directly affect the functionality of the project, but we recommend addressing them.

ID	Description	STATUS
A1	Compiler known issues	**INFO**

The contracts were compiled with the Solidity compiler v0.8.7 which, at the time of writing, has a known bug (SignedImmutables). We believe that it does not affect the code: no immutable signed integer variables are declared.

DISCLAIMER

The audited contracts have been analyzed using automated techniques and extensive human inspection in accordance with state-of-the-art practices as of the date of this report. The audit makes no statements or warranties on the security of the code. On its own, it cannot be considered a sufficient assessment of the correctness of the contract. While we have conducted an analysis to the best of our ability, it is our recommendation for high-value contracts to commission several independent audits, as well as a public bug bounty program.

ABOUT DEDAUB

Dedaub offers technology and auditing services for smart contract security. The founders, Neville Grech and Yannis Smaragdakis, are top researchers in program analysis. Dedaub's smart contract technology is demonstrated in the contract-library.com service, which decompiles and performs security analyses on the full Ethereum blockchain.