Attack Writeup

Share Inflation Attack

Ether-Fi

January 2025

Prepared for EtherFi

certora

ether.fi

Table of content

Background

During a scheduled Audit for Ether.Fi in **January 2025**, A share explosion attack vector was mapped. The following writeup elaborates on a different vector that exploits the same issue that was later uncovered and fixed.

The Fix has been reviewed and acknowledged, and to the best of our knowledge no funds were lost due to this issue.

The Attack

Share explosion attacks occur when a user can change the ratio between shares and the underlying assets, usually this is done through first deposit attacks or gifts to the protocol. Under most scenarios this attacks either end in:

- Denial of Service as each share becomes unobtainably high.
- Drain from other users as the first depositor can inflate the price of shares they already have in the pool.
- Gifting, raising the amount of assets each shares is worth.

That third point usually has a low impact as giving money to other people never seems like it would harm them, however as the ecosystem adapts we notice how seemingly harmless vectors become harmful indeed.
When a third party contract uses share tokens, and it allows for leverage or speculation on those shares value, by raising the value of each share and leveraging the shares you already have in the system you can reach a point where you make money from this action, by have a 10x or 100x or any value of leverage on your given shares your expected yield is that leverage multiplier times the per share raise. If that yield is greater than the gift to the system to increase the shares value. Then you can drain the third party contract, which in turn means that our contract would be abandoned by those third parties.

In EtherFi, the burnShares method allows for a user to burn their own shares, which does not affect the totalAssets in the system, increasing the value of each share.
Which would allow for that third attack mode described above.

```
function burnShares(address _user, uint256 _share) external {
    require(msg.sender == address(liquidityPool) || msg.sender == _user, "Incorrect Caller");
    require(shares[_user] >= _share, "BURN_AMOUNT_EXCEEDS_BALANCE");
    shares[_user] -= _share;
    totalShares -= _share;

    emit Transfer(_user, address(0), liquidityPool.amountForShare(_share));
    emit TransferShares(_user, address(0), _share);
}
```

We recommend not allowing users to burn their own shares through this API or to modify the totalAssets accordingly.

Disclaimer

Even though we hope this information is helpful, we provide no warranty of any kind, explicit or implied. The contents of this report should not be construed as a complete guarantee that the contract is secure in all dimensions. In no event shall Certora or any of its employees be liable for any claim, damages, or other liability, whether in an action of contract, tort, or otherwise, arising from, out of, or in connection with the results reported here.

About Certora

Certora is a Web3 security company that provides industry-leading formal verification tools and smart contract audits. Certora's flagship security product, Certora Prover, is a unique SaaS product that automatically locates even the most rare & hard-to-find bugs on your smart contracts or mathematically proves their absence. The Certora Prover plugs into your standard deployment pipeline. It is helpful for smart contract developers and security researchers during auditing and bug bounties.

Certora also provides services such as auditing, formal verification projects, and incident response.