Eralend Suffers $3.4 Million in Losses from Attack. Will Hackers Return the Funds?

BEOSIN
Blockchain Security
Eralend Suffers $3.4 Million in Losses from Attack. Will Hackers Return the Funds?

On July 25, 2023, according to data from Beosin EagleEye, the ZkSync-based decentralized lending protocol, **EraLend, suffered an attack resulting in a loss of approximately $3.4 million USD.**

EraLend is a decentralized lending protocol designed to maximize capital efficiency while minimizing risks associated with external liquidity and oracle dependencies. The security team at Beosin has provided the following analysis of the incident:

Incident Details:

- Attacking Transactions:
- 0x7ac4da1ea1b0903dfabda56f713ea5e4a960a3fc34467a844d037f86ee8bfe98
- 0x99efebacb3edaa3ac34f7ef462fd8eed85b46be281bd1329abfb215a494ab0ef

- Attacker's Address: 0xf1D076c9Be4533086f967e14EE6aFf204D5ECE7a

- Attacking Contracts:
- 0xC5c668DcD437b901DFE877DC99329Ac2ba338035
- 0x7d8772DCe73cDA0332bc47451aB868Ac98F335F0

- Target Contract:
- 0x1181D7BE04D80A8aE096641Ee1A87f7D557c6aeb

Vulnerability Analysis:

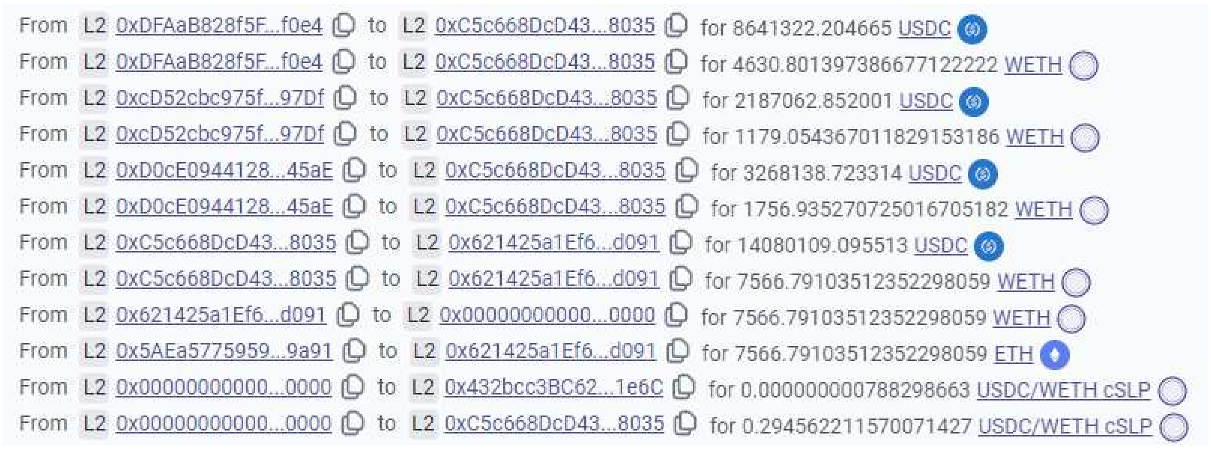
The main vulnerability exploited in this attack was a read reentrancy in the price oracle, causing a discrepancy between the ctoken contract's lending value calculation and its liquidation value calculation in the EraLend. This discrepancy allowed the attacker to borrow a higher amount than they needed to repay, enabling them to make profits from the borrowing and liquidation process. The attacker leveraged multiple contracts to carry out these operations and obtain a substantial amount of USDC.

Attack Process:

Taking transaction 0x7ac4da1ea... as an example:

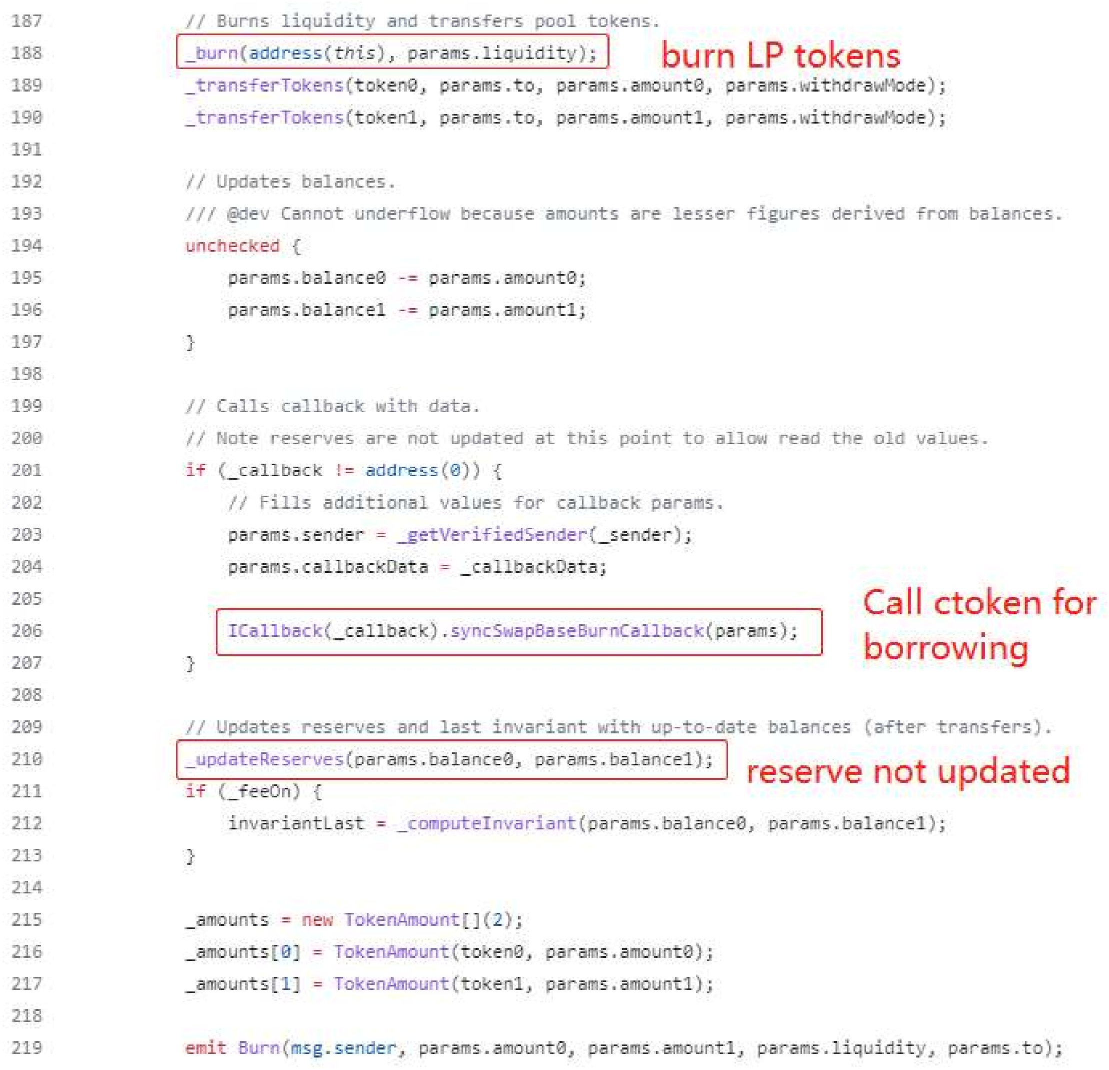
Attack Preparation:
1. The attacker extracted approximately 1.68 ETH from OKX and MEXC to cover the deployment of the attack contract and transaction gas fees.

2. The attacker borrowed 14.08 million USDC and 7566 ETH through a flash loan, staked them to the 0x621425 contract, and obtained 0.29 LP tokens to prepare for the attack.



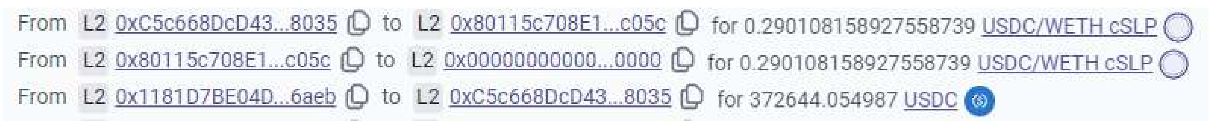
```
From L2 0xDFAaB828f5F...f0e4 to L2 0xC5c668DcD43...8035 for 8641322.204665 USDC
From L2 0xDFAaB828f5F...f0e4 to L2 0xC5c668DcD43...8035 for 4630.801397386677122222 WETH
From L2 0xcD52cbc975f...97Df to L2 0xC5c668DcD43...8035 for 2187062.852001 USDC
From L2 0xcD52cbc975f...97Df to L2 0xC5c668DcD43...8035 for 1179.054367011829153186 WETH
From L2 0xD0cE0944128...45aE to L2 0xC5c668DcD43...8035 for 3268138.723314 USDC
From L2 0xD0cE0944128...45aE to L2 0xC5c668DcD43...8035 for 1756.935270725016705182 WETH
From L2 0xC5c668DcD43...8035 to L2 0x621425a1Ef6...d091 for 14080109.095513 USDC
From L2 0xC5c668DcD43...8035 to L2 0x621425a1Ef6...d091 for 7566.7910351235229805 9 WETH
From L2 0x621425a1Ef6...d091 to L2 0x00000000000...0000 for 7566.79103512352298059 WETH
From L2 0x5AEa5775959...9a91 to L2 0x621425a1Ef6...d091 for 7566.79103512352298059 ETH
From L2 0x00000000000...0000 to L2 0x432bcc3BC62...1e6C for 0.000000000788298663 USDC/WETH cSLP
From L2 0x00000000000...0000 to L2 0xC5c668DcD43...8035 for 0.294562211570071427 USDC/WETH cSLP
```

Attack Phase:
1. The attacker called the burn function of the SyncSwap (0x8011) contract, which destroyed their obtained LP tokens and reclaimed the added liquidity. However, the burn function triggered a specific function (_callback) at the callback address (line 206), which updated the liquidity reserve information only after the callback.



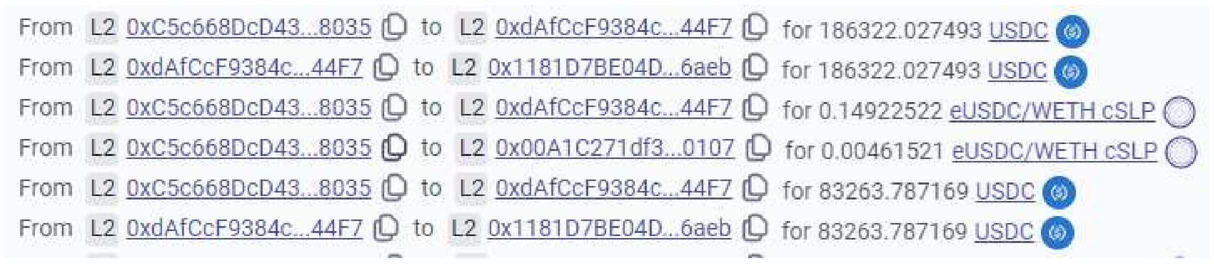
```
        // Burns liquidity and transfers pool tokens.
        _burn(address(this), params.liquidity);
        _transferTokens(token0, params.to, params.amount0, params.withdrawMode);
        _transferTokens(token1, params.to, params.amount1, params.withdrawMode);

        // Updates balances.
        /// @dev Cannot underflow because amounts are lesser figures derived from balances.
        unchecked {
            params.balance0 -= params.amount0;
            params.balance1 -= params.amount1;
        }

        // Calls callback with data.
        // Note reserves are not updated at this point to allow read the old values.
        if (_callback != address(0)) {
            // Fills additional values for callback params.
            params.sender = _getVerifiedSender(_sender);
            params.callbackData = _callbackData;

            ICallback(_callback).syncSwapBaseBurnCallback(params);
        }

        // Updates reserves and last invariant with up-to-date balances (after transfers).
        _updateReserves(params.balance0, params.balance1);
        if (_feeOn) {
            invariantLast = _computeInvariant(params.balance0, params.balance1);
        }

        _amounts = new TokenAmount[](2);
        _amounts[0] = TokenAmount(token0, params.amount0);
        _amounts[1] = TokenAmount(token1, params.amount1);

        emit Burn(msg.sender, params.amount0, params.amount1, params.liquidity, params.to);
```

2. When the burn function reached the attacker's malicious contract, it executed pre-prepared malicious code in a specific function. This malicious code reentered the ctoken (0x1181) contract's functions, specifically the repayBorrow and borrow functions, to pledge LP tokens and borrow USDC. However, at this point, the SyncSwap contract's reserve information had not been updated, and the ctoken's borrowing amount calculation depended on the reserve of SyncSwap. Consequently, the attacker borrowed 372,644 USDC based on the reserve before reclaiming liquidity.



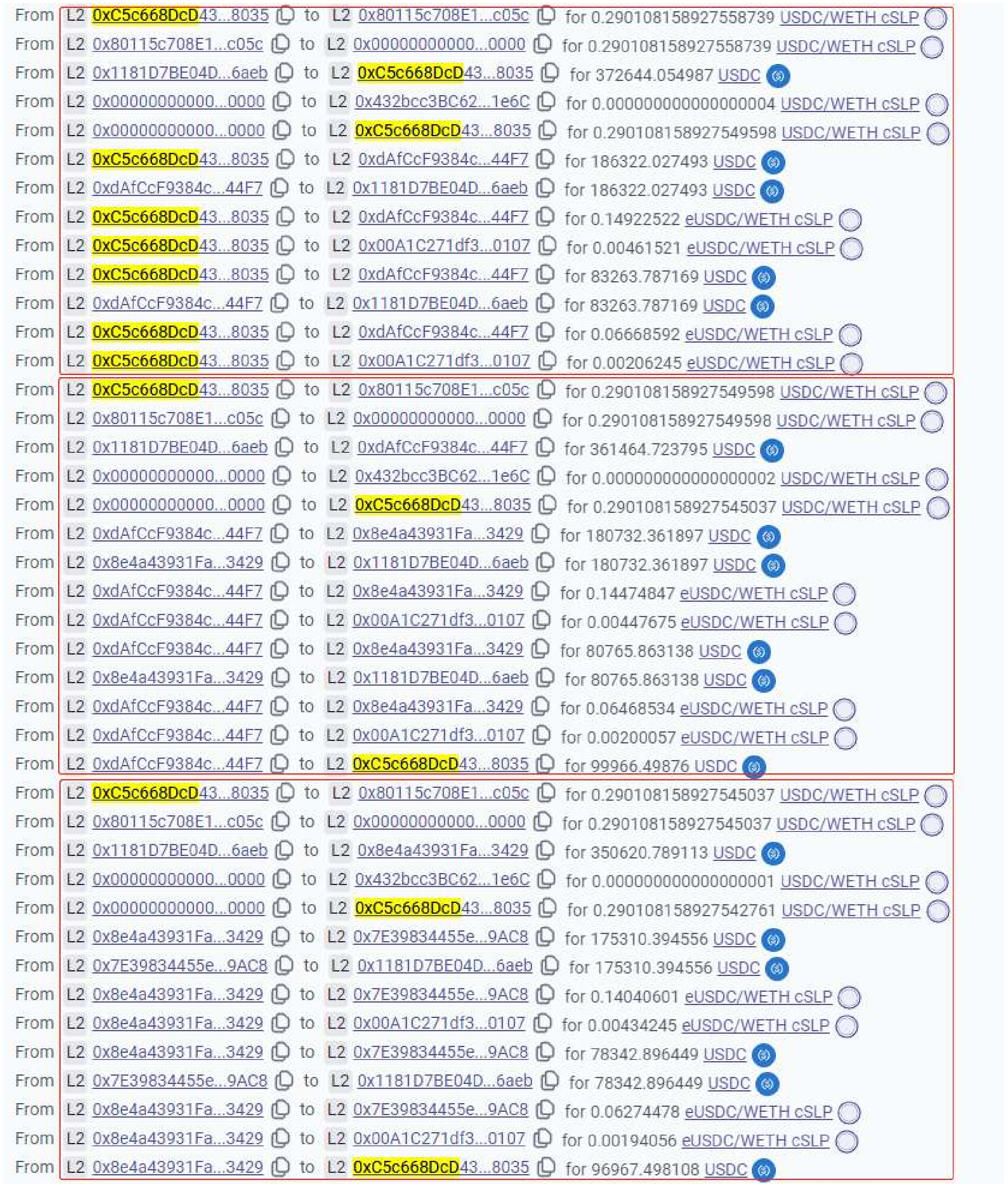
```
From L2 0xC5c668DcD43...8035 to L2 0x80115c708E1...c05c for 0.290108158927558739 USDC/WETH cSLP
From L2 0x80115c708E1...c05c to L2 0x00000000000...0000 for 0.290108158927558739 USDC/WETH cSLP
From L2 0x1181D7BE04D...6aeb to L2 0xC5c668DcD43...8035 for 372644.054987 USDC
```

3. After the reentrancy ended, the SyncSwap reserve was updated, and the attacker repaid the borrowed amount. Since the updated reserve was used for the calculation, the attacker was able to repay the loan with an amount lower than the borrowed tokens. As a result, the attacker returned 269,585 USDC and made a profit of 103,059 USDC.



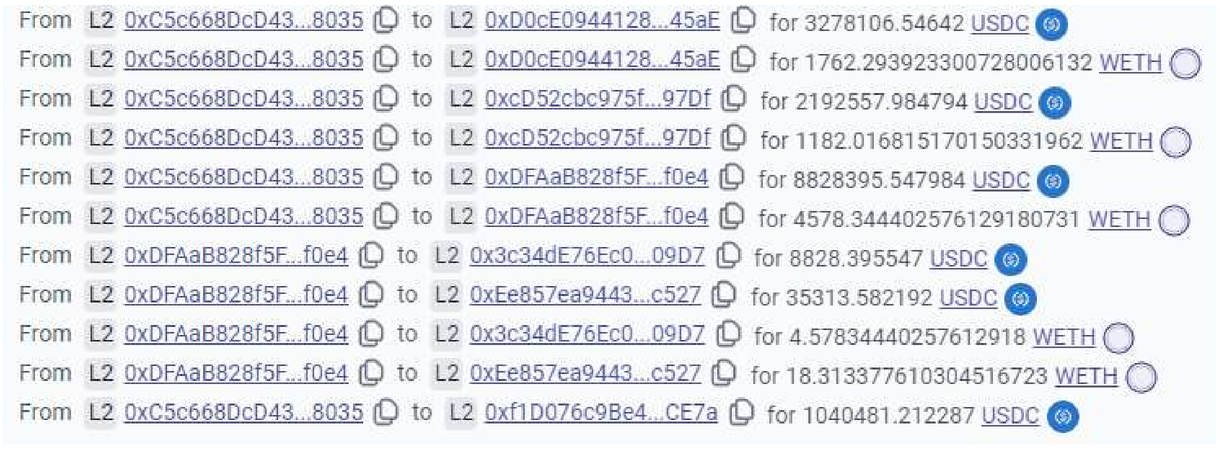
From L2 0xC5c668DcD43...8035 to L2 0xdAfCcF9384c...44F7 for 186322.027493 USDC
From L2 0xdAfCcF9384c...44F7 to L2 0x1181D7BE04D...6aeb for 186322.027493 USDC
From L2 0xC5c668DcD43...8035 to L2 0xdAfCcF9384c...44F7 for 0.14922522 eUSDC/WETH cSLP
From L2 0xC5c668DcD43...8035 to L2 0x00A1C271df3...0107 for 0.00461521 eUSDC/WETH cSLP
From L2 0xC5c668DcD43...8035 to L2 0xdAfCcF9384c...44F7 for 83263.787169 USDC
From L2 0xdAfCcF9384c...44F7 to L2 0x1181D7BE04D...6aeb for 83263.787169 USDC

4. The attacker repeated the above process using multiple contracts.



From L2 0xC5c668DcD43...8035 to L2 0x80115c708E1...c05c for 0.290108158927558739 USDC/WETH cSLP
From L2 0x80115c708E1...c05c to L2 0x00000000000...0000 for 0.290108158927558739 USDC/WETH cSLP
From L2 0x1181D7BE04D...6aeb to L2 0xC5c668DcD43...8035 for 372644.054987 USDC
From L2 0x00000000000...0000 to L2 0x432bcc3BC62...1e6C for 0.000000000000000004 USDC/WETH cSLP
From L2 0x00000000000...0000 to L2 0xC5c668DcD43...8035 for 0.290108158927549598 USDC/WETH cSLP
From L2 0xC5c668DcD43...8035 to L2 0xdAfCcF9384c...44F7 for 186322.027493 USDC
From L2 0xdAfCcF9384c...44F7 to L2 0x1181D7BE04D...6aeb for 186322.027493 USDC
From L2 0xC5c668DcD43...8035 to L2 0xdAfCcF9384c...44F7 for 0.14922522 eUSDC/WETH cSLP
From L2 0xC5c668DcD43...8035 to L2 0x00A1C271df3...0107 for 0.00461521 eUSDC/WETH cSLP
From L2 0xC5c668DcD43...8035 to L2 0xdAfCcF9384c...44F7 for 83263.787169 USDC
From L2 0xdAfCcF9384c...44F7 to L2 0x1181D7BE04D...6aeb for 83263.787169 USDC
From L2 0xC5c668DcD43...8035 to L2 0xdAfCcF9384c...44F7 for 0.06668592 eUSDC/WETH cSLP
From L2 0xC5c668DcD43...8035 to L2 0x00A1C271df3...0107 for 0.00206245 eUSDC/WETH cSLP
From L2 0xC5c668DcD43...8035 to L2 0x80115c708E1...c05c for 0.290108158927549598 USDC/WETH cSLP
From L2 0x80115c708E1...c05c to L2 0x00000000000...0000 for 0.290108158927549598 USDC/WETH cSLP
From L2 0x1181D7BE04D...6aeb to L2 0xdAfCcF9384c...44F7 for 361464.723795 USDC
From L2 0x00000000000...0000 to L2 0x432bcc3BC62...1e6C for 0.000000000000000002 USDC/WETH cSLP
From L2 0x00000000000...0000 to L2 0xC5c668DcD43...8035 for 0.290108158927545037 USDC/WETH cSLP
From L2 0xdAfCcF9384c...44F7 to L2 0x8e4a43931Fa...3429 for 180732.361897 USDC
From L2 0x8e4a43931Fa...3429 to L2 0x1181D7BE04D...6aeb for 180732.361897 USDC
From L2 0xdAfCcF9384c...44F7 to L2 0x8e4a43931Fa...3429 for 0.14474847 eUSDC/WETH cSLP
From L2 0xdAfCcF9384c...44F7 to L2 0x00A1C271df3...0107 for 0.00447675 eUSDC/WETH cSLP
From L2 0xdAfCcF9384c...44F7 to L2 0x8e4a43931Fa...3429 for 80765.863138 USDC
From L2 0x8e4a43931Fa...3429 to L2 0x1181D7BE04D...6aeb for 80765.863138 USDC
From L2 0xdAfCcF9384c...44F7 to L2 0x8e4a43931Fa...3429 for 0.06468534 eUSDC/WETH cSLP
From L2 0xdAfCcF9384c...44F7 to L2 0x00A1C271df3...0107 for 0.00200057 eUSDC/WETH cSLP
From L2 0xdAfCcF9384c...44F7 to L2 0xC5c668DcD43...8035 for 99966.49876 USDC
From L2 0xC5c668DcD43...8035 to L2 0x80115c708E1...c05c for 0.290108158927545037 USDC/WETH cSLP
From L2 0x80115c708E1...c05c to L2 0x00000000000...0000 for 0.290108158927545037 USDC/WETH cSLP
From L2 0x1181D7BE04D...6aeb to L2 0x8e4a43931Fa...3429 for 350620.789113 USDC
From L2 0x00000000000...0000 to L2 0x432bcc3BC62...1e6C for 0.000000000000000001 USDC/WETH cSLP
From L2 0x00000000000...0000 to L2 0xC5c668DcD43...8035 for 0.290108158927542761 USDC/WETH cSLP
From L2 0x8e4a43931Fa...3429 to L2 0x7E39834455e...9AC8 for 175310.394556 USDC
From L2 0x7E39834455e...9AC8 to L2 0x1181D7BE04D...6aeb for 175310.394556 USDC
From L2 0x8e4a43931Fa...3429 to L2 0x7E39834455e...9AC8 for 0.14040601 eUSDC/WETH cSLP
From L2 0x8e4a43931Fa...3429 to L2 0x00A1C271df3...0107 for 0.00434245 eUSDC/WETH cSLP
From L2 0x8e4a43931Fa...3429 to L2 0x7E39834455e...9AC8 for 78342.896449 USDC
From L2 0x7E39834455e...9AC8 to L2 0x1181D7BE04D...6aeb for 78342.896449 USDC
From L2 0x8e4a43931Fa...3429 to L2 0x7E39834455e...9AC8 for 0.06274478 eUSDC/WETH cSLP
From L2 0x8e4a43931Fa...3429 to L2 0x00A1C271df3...0107 for 0.00194056 eUSDC/WETH cSLP
From L2 0x8e4a43931Fa...3429 to L2 0xC5c668DcD43...8035 for 96967.498108 USDC

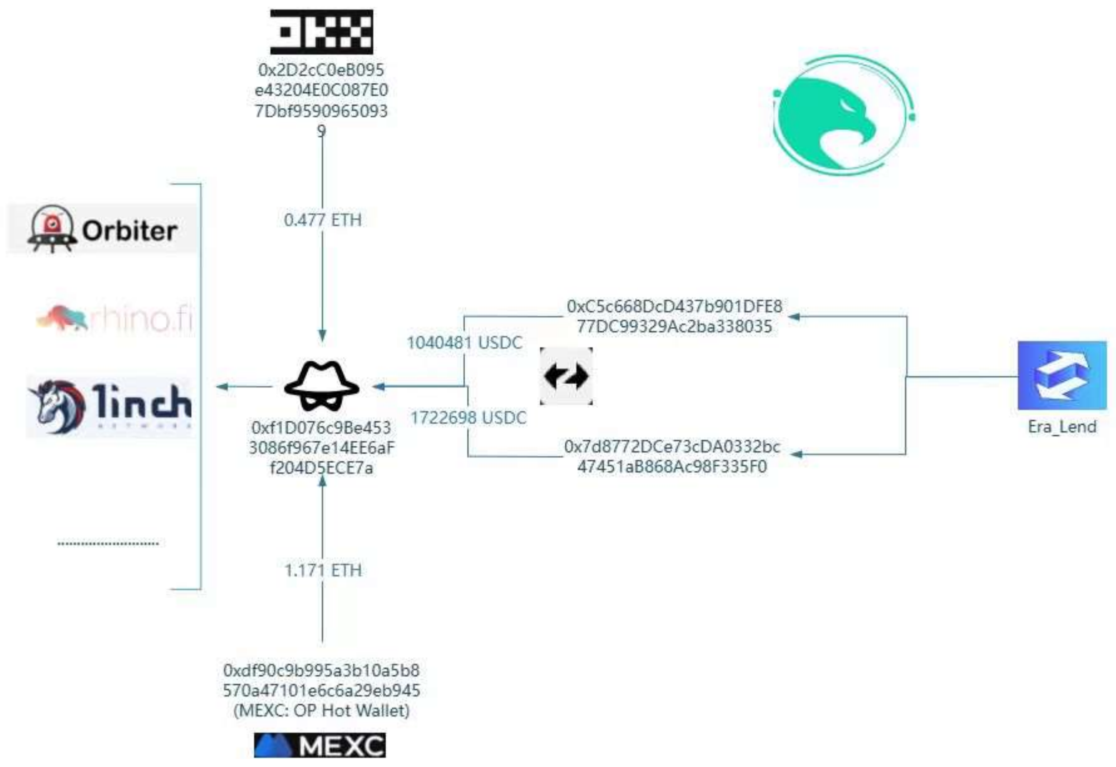
5. The attacker repaid the flash loan and transferred the attack profits to his/her address.



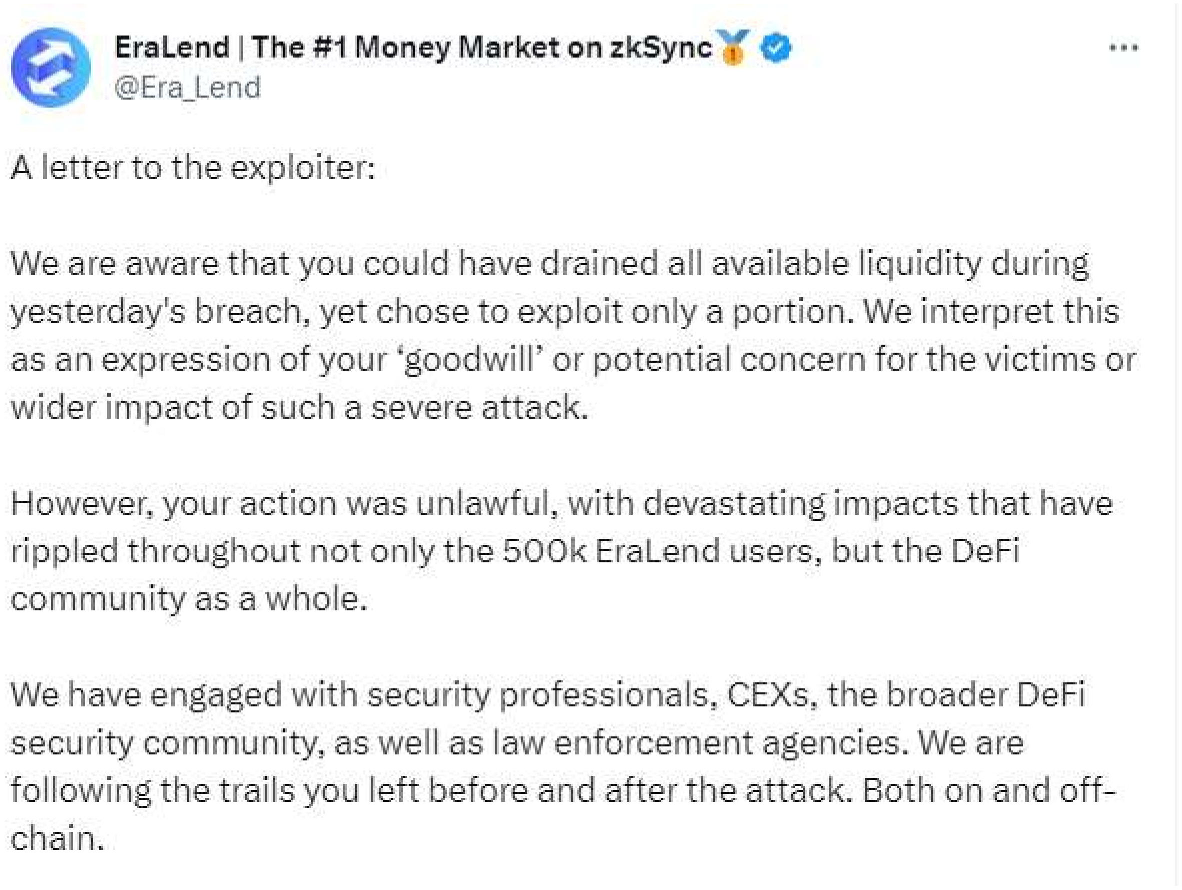
From L2 0xC5c668DcD43...8035 to L2 0xD0cE0944128...45aE for 3278106.54642 USDC
From L2 0xC5c668DcD43...8035 to L2 0xD0cE0944128...45aE for 1762.293923300728006132 WETH
From L2 0xC5c668DcD43...8035 to L2 0xcD52cbc975f...97Df for 2192557.984794 USDC
From L2 0xC5c668DcD43...8035 to L2 0xcD52cbc975f...97Df for 1182.016815170150331962 WETH
From L2 0xC5c668DcD43...8035 to L2 0xDFAaB828f5F...f0e4 for 8828395.547984 USDC
From L2 0xC5c668DcD43...8035 to L2 0xDFAaB828f5F...f0e4 for 4578.344402576129180731 WETH
From L2 0xDFAaB828f5F...f0e4 to L2 0x3c34dE76Ec0...09D7 for 8828.395547 USDC
From L2 0xDFAaB828f5F...f0e4 to L2 0xEe857ea9443...c527 for 35313.582192 USDC
From L2 0xDFAaB828f5F...f0e4 to L2 0x3c34dE76Ec0...09D7 for 4.57834440257612918 WETH
From L2 0xDFAaB828f5F...f0e4 to L2 0xEe857ea9443...c527 for 18.313377610304516723 WETH
From L2 0xC5c668DcD43...8035 to L2 0xf1D076c9Be4...CE7a for 1040481.212287 USDC

Funds Tracking

As of the time of posting, Beosin KYT/AML tracking found that the stolen funds have been transferred to other chains via various cross-chain bridge services such as Orbiter and 1inch.



On the morning of July 27, EraLend posted a letter to the hacker on social media suggesting that the hacker return 90% of the funds and keep 10% of the stolen funds as a white hat bounty, at the time of writing, the hacker has not yet made any restitution.


EraLend | The #1 Money Market on zkSync
@Era_Lend

A letter to the exploiter:

We are aware that you could have drained all available liquidity during yesterday's breach, yet chose to exploit only a portion. We interpret this as an expression of your 'goodwill' or potential concern for the victims or wider impact of such a severe attack.

However, your action was unlawful, with devastating impacts that have rippled throughout not only the 500k EraLend users, but the DeFi community as a whole.

We have engaged with security professionals, CEXs, the broader DeFi security community, as well as law enforcement agencies. We are following the trails you left before and after the attack. Both on and off-chain.

Summary

In response to this incident, Beosin security team recommends:
1. When relying on the SyncSwap project's real-time reserve volume for price calculation, read reentry scenarios should be considered to prevent inconsistent price calculations for related functions in the same transaction.
2. Before the project goes live, it is recommended to choose a professional security audit company to conduct a comprehensive security audit to avoid risks.

Beosin is a leading global blockchain security company co-founded by several professors from world-renowned universities and there are 40+ PhDs in the team, and set up offices in 10+ cities including Hong Kong, Singapore, Tokyo and Miami. With the mission of "Securing Blockchain Ecosystem", Beosin provides "All-in-one" blockchain security solution covering Smart Contract Audit, Risk Monitoring & Alert, KYT/AML, and Crypto Tracing. Beosin has already provided security for 2000+ blockchain companies, audited more than 3000 smart contracts and protected our customers' assets worth of $500