HackerGold Bug Analysis

DEMIAN BRENER | JANUARY 9, 2017

Security Audits

On Jan 4th, Zack Coburn submitted a vulnerability report on EtherCamp's implementation of the StandardToken contract. EtherCamp, in developing their own implementation, inadvertently introduced a small typo which happened to be a valid operation.

The typo is found in line 77, in the `transferFrom()` function:

```
balances[to] =+ value;
```

where `=+` (equal to) should have been `+=` (transfer to).

> Please note that this vulnerability is not present in Zeppelin's implementation of StandardToken.

HackerGold token did not hold any ether balance so no funds are at risk. The EtherCamp team is working on solving this issue, and will probably deploy a new fixed token. They should be able to amend the situation shortly.

As Atul Gawande describes in The Checklist Manifesto, we must distinguish between two types of errors. The first type are errors we commit because we don't know too much. Second type errors are committed when knowledge exists, yet we fail to apply it correctly.

The vulnerability found on EtherCamp's StandardToken.sol was due to a human error. The `=+` looks more like a typo than a conceptual mistake.

In this particular case, the vulnerability would have been avoided by using the SafeMath contract included in Zeppelin. This contract provides functions to perform mathematical operations with safety checks.

In our public code audit to EtherCamp's Hacker Gold, we recommended the EtherCamp team to use the SafeMath library:

> **Use safe math**
> There are many unchecked math operations in the code. We couldn't find any related attack vectors on the HKG contract, but it's always better to be safe and perform checked operations. Consider using a safe math library, or performing pre-condition checks on any math operation.

By following this recommendation, the problem would have been prevented.

Rewriting the same StandardToken contract for every project is prone to bring these kind of problems. That's the reason why we created Zeppelin as an open-source framework of reusable and secure smart contracts in the Solidity language.

Instead of developing their smart contracts from scratch, developers and projects can make use of Zeppelin's community-vetted templates to build from. We actually provide a secure implementation of StandardToken which uses SafeMath.

For general information about smart contract security, check out our thoughts here.

Related Posts

Beefy

BRUSHFAM
OpenBrush Contracts

Linea

Beefy Zap Audit

BeefyZapRouter serves as a versatile intermediary designed to execute users' orders through routes...

Security Audits

OpenBrush Contracts Library Security Review

OpenBrush is an open-source smart contract library written in the Rust programming language and the...

Security Audits

Linea Bridge Audit

Linea is a ZK-rollup deployed on top of Ethereum. It is designed to be EVM-compatible and aims to...

Security Audits