Bug Analysis Report

Project Information

Project: [Your Project Name]
Developer: @0xSpider
Date of Payment: October 20, 2025
Date of Delivery: October 25, 2025
Amount Paid: 1.2 ETH
Project Treasury at Risk: 5 ETH

1. Summary

On October 20th, the client paid 1.2 ETH to @0xSpider to develop a custom staking contract.
The code was delivered on October 25th, and the client deployed it with 5 ETH in the treasury.
Within 24 hours, an attacker exploited a critical vulnerability in the contract, resulting in the loss of the entire treasury.

2. Vulnerability Description

Type: Reentrancy bug
Impact: Complete loss of 5 ETH in project treasury
Discovery: Post-attack analysis revealed the vulnerability.
Details: The contract allowed external calls before updating internal balances, creating an opportunity for attackers to recursively drain funds. This is a well-known issue in smart contract development.

3. Developer Response

The developer, @0xSpider, acknowledged the existence of the bug but claimed it was not his responsibility, citing that the client "approved" the code.

4. Consequences

- Total loss of project treasury (5 ETH)
- Project halted due to lack of funds
- Trust damage between developer and client

5. Recommendations

1. Conduct a full audit of any smart contract before deployment.
2. Avoid deploying contracts with significant funds without peer review or professional audit.
3. Implement standard security patterns like reentrancy guards.
4. Consider insurance or multisig mechanisms for high-value treasuries.

6. Conclusion

The incident highlights the critical importance of secure smart contract development and independent audits. Negligence in basic security patterns can lead to catastrophic financial loss.