**Analysis of the Grim Finance Hack**



On December 19, 2021, Our team at SlowMist was notified of a Reentrancy attack on the Grim Finance project within the Fantom Blockchain. At the time of writing this, over $30M in funds were transferred to [this](https://ftmscan.com/address/0xdefc385d7038f391eb0063c2f7c238cfb55b206c) address. This attack was investigated by our team, and this is our analysis of the incident.

**Let’s first explain what is a Reentrancy**

Reentrancy is when a program execution is interrupted mid-operation, re-initiated, and both run completely without any errors in the execution. Here is a quick explanation by [Rugdoc.io](https://twitter.com/RugDocIO/status/1472293712185151492)

1. The attacker borrows WFTM and BTC tokens via a flash loan and adds liquidity to Spirit Swap to obtain Spirit-LP certificates.

2. The attacker then uses the certificates as collateral with Grim Finance via the depositFor() in the GrimBoostVault contract. The DepositFor() allows users to specify the token to be deposited and transfer the user-specified token into the Grim Boost Vault via safeTransferFrom(). This creates collateral for the user based on the difference between the contract and the funds received by the policy pool before and after the transfer (In this case SPIRIT-LP).

3. However, because the depositFor() does not check the validity of the funds the user specified, the attacker passes the token contract address. When the GrimBoostVault calls the transferFrom function of the malicious contract through the safeTransferFom function, the contract reinvokes the depositFor function. The contract is able to make multiple reentries and use it for collateral into the SPIRIT-LP certificates. This ensures the difference between the tokens expected to be received by the GrimBoostVault before and after the reentry exists. The depositFor function then calculates the difference and provides the corresponding collateral for the attacker.

4. Since the attacker has re-entered the GrimBoostVault contract many times, this gave them access to more collateral. Allowing them to withdraw more SPIRIT-LP liquidity certificates in the GrimBoostVault contract than previously provided. The hacker then uses the SPIRIT-LP liquidity certificate to remove liquidity within the pool for WFTM and BTC tokens and repay the flashloan.

This attack was caused by the depositFor function of the GrimBoostVault contract. It failed to check the validity of the token passed in by the user and did not have a reentrancy guard. As a result, a smart contract could reenter the depositFor() and obtain far more than the expected collateral. The SlowMist security team recommends that the parameters passed in by users should be checked to see if they meet expectations, and external calls in functions should be controlled for risks such as reentrancy attacks caused by external calls.

Reference attack deals: <https://ftmscan.com/tx/0x19315e5b150d0a83e797203bb9c957ec1fa8a6f404f4f761d970cb29a74a5dd6>