

SushiMaker Smart Contract Audit Report

This report summarizes the findings of a security audit conducted on the SushiMaker smart contract using

1. Front-Running Vulnerability

- **Severity**: Medium
- **Description**: The `convert` function is vulnerable to front-running attacks. An attacker could place a transaction before the user's transaction to manipulate the outcome.
- **Impact**: Users could receive less value for their tokens than expected if the attacker successfully manipulates the transaction order.
- **Mitigation**:
 - Implement a commit-reveal pattern to prevent front-running.
 - Utilize an off-chain matching engine for order matching.

2. Potential Reentrancy Attack (LLaMA)

- **Severity**: Medium
- **Description**: While the `SafeERC20` library is used to mitigate reentrancy risks, the contract's reliance on external calls for token transfers remains a potential vulnerability.
- **Impact**: An attacker could potentially execute a reentrancy attack, leading to the loss of funds or manipulation of the contract's state.
- **Mitigation**:
 - Implement a reentrancy guard like the `ReentrancyGuard` library.
 - Develop custom reentrancy protection mechanisms within the contract.

3. Potential Overflow Attack (LLaMA)

- **Severity**: Low-Medium
- **Description**: Although the contract utilizes `SafeMath` to prevent arithmetic overflows, complex calculations involving multiple external calls could potentially lead to overflow vulnerabilities.
- **Impact**: An attacker could exploit overflows to manipulate the contract's logic or potentially drain funds.
- **Mitigation**:
 - Thoroughly review and audit all calculations involving multiple external calls for potential overflow vulnerabilities.
 - Employ additional safeguards and validation checks to prevent overflows.

Overall Assessment

While the SushiMaker contract appears well-structured and uses libraries like `SafeERC20` and `SafeMath` to mitigate common vulnerabilities, the audit identified several areas for improvement, particularly regarding front-running, reentrancy, and overflow protection.

Recommendations

- Prioritize implementing the recommended mitigations for the front-running vulnerability.
- Conduct a thorough review of the contract's logic, especially calculations involving external calls, to address potential overflow vulnerabilities.
- Consider implementing additional security measures like logging, monitoring, and custom reentrancy protection mechanisms.

Disclaimer

This report represents the findings of a security audit and should be considered as a starting point for security improvements. It does not constitute a guarantee of security.