DeFi-Yield-Y — Internal Security Report

Подробный internal report с PoC, моделированием атак, тестами и патчами. Замените <...>

1. Введение и scope

Контракты: StakingPool.sol, RewardManager.sol, OracleAdapter.sol, Admin.sol
Объём: ~1200 строк
Архитектура: proxy upgradeable (EIP-1967), использует Uniswap v2 pools для spot price; c
single EOA на момент аудита.

2. Метаданные воспроизводимости

Commit before: <hash-before>
Commit after: <hash-after>
Test harness: Foundry & Hardhat, forked mainnet @ block <number>
Инструменты: Slither vX, Mythril vY, Tenderly, Echidna, Aave v3 fork для flash loan sims
Тесты: unit 92%, integration 48 tests, fuzz 10k cases

3. Executive Summary

Ключевые находки: реентранси и oracle manipulation —
исправлены; centralization (emergencyWithdrawAll) — accepted by team, но требует срочно
Рекомендация: публикация proof-of-mitigation (адреса multisig/timelock и deadlines) пре

4. Сводный список находок

ID Серьёзность Описание Статус Score
DFY-CRIT-001 Critical Reentrancy in withdraw Fixed 9.1
DFY-HIGH-002 High Oracle manipulation via AMM Fixed 8.4
DFY-HIGH-003 High Centralization: emergencyWithdrawAll Accepted (mitigation requ
DFY-MED-004 Medium Upgradeability governance Recommended/Fixed 6.8
DFY-MED-005 Medium Missing circuit breaker Implemented 6.0
DFY-LOW-006 Low Events/logging Partially fixed 3.2

5. Подробные технические разборы

DFY-CRIT-001 — Reentrancy in withdraw (Fixed)

Описание: balance updated after external transfer; vulnerable to reentrancy.
Remediation: Checks-Effects-Interactions, ReentrancyGuard usage; пример:

solidity

```
function withdraw(uint256 amount) external nonReentrant {
    uint256 userBalance = balances[msg.sender];
```

```
    require(userBalance >= amount, "Insufficient");
    balances[msg.sender] = userBalance - amount;
    stakingToken.safeTransfer(msg.sender, amount);
    emit Withdraw(msg.sender, amount);}
```

Tests/PoC: Tenderly PoC trace <tenderly_link>;
unit & integration tests cover reentrancy attempts (all pass post-fix).

DFY-HIGH-002 — Oracle manipulation via AMM (Fixed)

Описание: spot price from Uniswap v2 made possible intra-tx manipulation.
Remediation: primary Chainlink feeds; AMM prices only via TWAP (window = 1800s); maxPric
Tests: simulated flash loan attacks via forked Aave; after fix manipulation infeasible u

DFY-HIGH-003 — Centralization: emergencyWithdrawAll controlled by single EOA

Описание: emergencyWithdrawAll callable by single EOA — rug-pull vector.
Status: Accepted by team; mitigation required.
Required remediation (urgent):
    Migrate control to Gnosis Safe multisig(3/5).
   Deploy Timelock contract with min delay 48h for critical admin operations.
   Publicly announce multisig/timelock addresses and deadlines.
Tests: unit test to ensure only multisig/timelock can call emergencyWithdrawAll.

DFY-MED-004 — Upgradeability & Governance

Описание: возможность upgrade требует governance flow и ограничений.
Recommendation: use Timelock-governor pattern; restrict upgrades to timelock/multisig a

DFY-MED-005 — Circuit breaker & Monitoring

Описание: реализован circuit breaker для аномалий (withdraw spikes, price deviations).
Config examples:
    hourlyWithdrawThreshold = 0.05 * TVL
    priceDeviationThreshold = 0.20 (20% in 10 min)
Monitoring: on-chain keepers + off-chain alerts (Prometheus/Slack).

6. Экономическое моделирование атак

Вложены скрипты: scripts/attack_sim.py — расчёт required ETH для 10% shift в целевом пул
Результат: после TWAP+Chainlink, capital_required >> realistic attacker capacity; flash fix.

7. Patch snippets & Tests

Reentrancy fix snippet (см. выше).

Oracle adapter snippet:

solidity

```
function getPrice() public view returns (uint256) {
    if (chainlinkAvailable()) {
        return uint256(chainlink.latestRoundData().answer);
    } else {
        return twapPrice();
}}
```

Admin modifier:

solidity

```
modifier onlyMultisigOrTimelock() {
   require(msg.sender == multisig || msg.sender == timelock, "Not authorized");
    _
;}
```

Список тестов: withdraw_reentrancy_attempt_reverts, oracle_twap_resistant_to_flashloa

8. Operational roadmap & recommendations

```
Immediate (7 days):
   Migrate emergencyWithdrawAll to multisig(3/5) + Timelock(48h). Provide addresses and
   Deploy patched contracts to staging and run full integration tests.
Short term (30 days):
    Circuit breakers, monitoring, bug bounty program.
Medium term (1–3 months):
   Formal on-chain governance, formal verification of reward math, independent second au
```

9. Action items & owners

Action Owner Deadline Evidence Migrate emergencyWithdrawAll to multisig DevOps multisigAddress: <0x...> Deploy reentrancy fixes SmartContracts Done Commit Integrate Chainlink/TWAP OraclesTeam Done Commit 10. Приложения

```
Flattened contracts (post-fix) — attachments
Test reports (unit/integration) — attachments
Slither & Mythril outputs — attachments
Tenderly PoC traces — attachments
Economic modelling spreadsheets — attachments
```