Домашнее задание №8
Тема: Безопасность смарт-контрактов

Курс: Solidity Developer
Учебное заведение: Otus
ФИО: Яровой Андрей

Тема работы: Аудит безопасности смарт-контракта SwipTestNft

1. Введение

В рамках данной работы проведён аудит безопасности смарт-контракта SwipTestNft, реализующего стандарт ERC721 с ограничением максимального количества токенов и функцией публичной покупки NFT.

2. Область аудита

Контракт: SwipTestNft.sol
Версия компилятора: Solidity ^0.8.30
Используемые библиотеки: OpenZeppelin ^5.4.0
Тип анализа: ручной аудит + автоматический анализ.

3. Результаты ручного анализа

Критическая уязвимость:
Отсутствует проверка msg.value в функции buy(), что позволяет бесплатно получать NFT.
Рекомендуется добавить require(msg.value == _priceInWei);

Уязвимости среднего уровня:
1) Использование transfer() вместо call().
Пример исправления:
(bool success,) = owner().call{value: balance}("");
require(success, "withdraw failed");

2) Централизация управления минтингом.
Owner может минтить токены кому угодно и контролировать распределение supply. Это значимый централизованный риск.
Рекомендации: Добавить лимит на owner mint.

Уязвимости низкого уровня:
- Отсутствуют события покупки.
Рекомендации: Добавить событие
event NFTPurchased(address indexed buyer, uint256 tokenId);

- Используются строки вместо custom errors.
Рекомендации: Использовать кастомные ошибки.
error MaxSupplyReached();

- Можно применить immutable. (uint256 private _priceInWei; uint256 private _maxSupply;)
Они не изменяются всё равно в контракте.
Рекомендации: Обозначить их как immutable

uint256 private immutable _priceInWei;

uint256 private immutable _maxSupply;

Информативные уровни:
- Reentrancy отсутствует

- Контракт не upgradeable

4. Анализ оптимизации газа

Рекомендуется использовать custom errors, immutable-переменные, unchecked-инкремент и call() вместо transfer().

5. Автоматический анализ

Slither:

Slither в Hardhat 3 работает не стабильно, запустить его там мне не получилось, но прогнал без проблем через Foundry. Нашёл он правда всего одну проблему связанные с immutable-states.

Detector: immutable-states

SwipTestNft._maxSupply (src/SwipTestNft.sol#11) should be immutable

SwipTestNft._priceInWei (src/SwipTestNft.sol#10) should be immutable

Mythril:

По результатам анализа критических уязвимостей не выявлено: he analysis was completed successfully. No issues were detected.

6. Возможность обновления

Текущая версия:
- Использует constructor
- Не поддеживает proxy pattern
- Нет storage gap

Для upgrade-ready версии необходимо:
1. Использовать initializable
2. Удалить constructor
3. Добавить _authorizeUpgrade
4. Проверить storage layout

7. Заключение

В ходе проведённого аудита смарт-контракта **SwipTestNft** был выполнен ручной анализ кода, а также автоматическая проверка с использованием инструментов статического анализа (Slither и Mythril).

По результатам аудита выявлено:

- **1 критическая уязвимость (High Severity)**
- **2 уязвимости среднего уровня (Medium Severity)**
- **3 уязвимости низкого уровня (Low Severity)**
- **2 информационных замечания (Informational)**

Контракт не готов к использованию в production-среде без устранения выявленной критической уязвимости.