НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

БЛОКЧЕЙН ТА ДЕЦЕНТРАЛІЗОВАНІ СИСТЕМИ

Комп'ютерний практикум №3

Дослідження безпечної реалізації та експлуатації децентралізованих додатків

Виконали:
Волинець Сергій ФІ-42мн
Сковрон Роман ФІ-42мн
Молдован Дмитро ФІ-42мн

Київ — 2025

1 Мета

Отримання навичок роботи із децентралізованими додатками та оцінка безпеки інформації при їх функціонуванні.

2 Завдання на лабораторну роботу

Дослідити вимоги OWASP (безпека web-додатків) та скласти аналогічні вимоги для обраної системи децентралізованих додатків.

3 Теоретичні відомості

OWASP (The Open Worldwide Application Security Project) це некомерційна організація, яка сфокусована на забезпеченні безпеки веб-додатків. Заснована у 2001 році, вона об'єднала експертів, розробників, тестувальників та загальну спільноту веб-розробки з метою покращення рівня безпеки онлайн-додатків.

Місія OWASP полягає в тому, щоб привертати увагу до критичних питань безпеки, з якими стикаються розробники. Основне її завдання – надання відкритих ресурсів, інструментів, стандартів і рекомендацій, які допоможуть створювати і підтримувати безпечні веб-додатки.

Ця організація націлена на забезпечення безпеки веб-додатків у різних аспектах. Вона займається виявленням і документуванням вразливостей, навчанням розробників питанням безпеки, а також підтримкою розроблення та поширення найкращих практик у кібербезпеці. Крім того, організація бере активну участь у формуванні стандартів і рекомендацій, сприяючи поліпшенню загальної безпеки веб-додатків у всьому світі.

Усі відомості, матеріали та напрцювання OWASP доступні на веб-сайті організації за посиланням: https://owasp.org/ . На ресурсах веб-сайту можна знайти список OWASP Top 10, який являє собою список з 10 найкритичніших вразливостей веб-додатків, регулярно оновлюваний експертами OWASP. Новий список публікується, в середньому, кожні 4 роки (2013, 2017, 2021). Новий список очікується у кінці літа поточного 2025 року, як заявлено розробниками. Також на розділах веб-сайту організації можна знайти інформацію про поточний статус списку, особливості його формування, редагування та збору інформації, а також версії та розширення списку написані різними мовами.

4 OWASP Top 10 Web Application Security Risks

Розглянемо список з топ 10 загроз для веб-застосунків за 2021 рік:

1. **Порушення контролю доступу (Broken Access Control):** 94% застосунків мають проблеми з обмеженням доступу. Це найпоширеніша категорія вразливостей.

2. **Криптографічні помилки (Cryptographic Failures):** Раніше — "Витік чутливих даних". Проблеми з реалізацією шифрування можуть призводити до компрометації даних.
3. **Ін'єкції (Injection):** Вразливості типу SQL, XSS тощо. Досі дуже поширена проблема — у 94% перевірених застосунків.
4. **Небезпечний дизайн (Insecure Design):** Новий пункт у списку. Вказує на відсутність безпечної архітектури, моделей загроз і патернів проектування.
5. **Неправильна конфігурація безпеки (Security Misconfiguration):** 90% застосунків містять хоча б одну помилку конфігурації. XML External Entities (XXE) тепер входить до цієї категорії.
6. **Вразливі або застарілі компоненти (Vulnerable and Outdated Components):** Проблема використання бібліотек із відомими вразливостями. Тестування таких випадків залишається складним.
7. **Помилки автентифікації та ідентифікації (Identification and Authentication Failures):** Раніше — "Неправильна автентифікація". Охоплює збої у процесах логіну, керування сесіями тощо.
8. **Порушення цілісності даних і ПЗ (Software and Data Integrity Failures):** Новий пункт. Проблеми з довірою до оновлень, CI/CD-процесів, зовнішніх джерел.
9. **Недостатній аудит та моніторинг (Security Logging and Monitoring Failures):** Впливає на виявлення атак і аналіз інцидентів. Розширена версія попереднього пункту про логування.
10. **Підробка запитів з серверної сторони (Server-Side Request Forgery, SSRF):** Хоч і зустрічається рідко, експерти вважають її критичною через високий потенціал експлуатації.

Більше роз'яснень, матеріалів та минулорічних опублікованих списків доступні за посиланням: https://owasp.org/www-project-top-ten/.

5 Вимоги до системи децентралізованих додатків обміну криптовалют

В якості прикладу децентралізованих систем нашою бригадою було обрано децентралізваний DeFi обмінник (DEX) Uniswap, що працює на блокчейні Ethereum.

Особливості обраного типу децентралізованих систем:

- **Тип dApp:** Децентралізована біржа (DEX)
- **Блокчейн:** Ethereum (підтримка L2: Arbitrum, Optimism)
- **Мова смарт-контрактів:** Solidity
- **Функції:** обмін токенів, надання ліквідності, заробіток на комісіях
- **Протокол:** автоматизований маркет-мейкер (AMM)

Принцип роботи:

1. **Обмін токенів (Swap):** Користувач обирає токени (наприклад, USDC → DAI), вводить суму — і Uniswap автоматично підбирає курс, виходячи з поточного пулу ліквідності.
2. **Пули ліквідності (Liquidity Pools):** Будь-хто може внести пару токенів (наприклад, ETH і USDC) в пул і стати ліквідатором. За це користувачі отримують частину комісій за обмін.
3. **Смарт-контракти:** Усі транзакції (обмін, додавання/виведення ліквідності) — це взаємодія з децентралізованими контрактами, без централізованих серверів чи реєстрації.

Безпека:

- Контракти Uniswap проходили аудит (наприклад, Trail of Bits, ConsenSys Diligence).
- Код відкритий і перевірений мільйонами користувачів.
- Вразливості все одно можливі, особливо у сторонніх копіях (форках).

Потенційні ризики:

- **Impermanent loss** — ризик втрати для постачальників ліквідності при сильних коливаннях цін.
- **Фішингові сайти** — користувачі можуть натрапити на підроблені інтерфейси.
- **Смарт-контрактні баги** — навіть перевірені контракти не гарантують 100% захист.

6 Список вимог до розробки систем децентралізованих додатків типу DeFi

(Згідно ASVS 5.0.0 `https://github.com/OWASP/ASVS/tree/v5.0.0?tab=readme-ov-file#latest-stable-version---500`)

V1. Кодування, санітизація та запобігання ін'єкціям

- V1.1.1: Перевірити, що всі зовнішні параметри, що передаються у контракти, проходять строгий тип-контроль та санітизацію перед використанням у критичних викликах.
- V1.2.3: Перевірити, що вхідні параметри у функції контракту не дозволяють виконання неочікуваних інструкцій через обхід механізмів Solidity (наприклад, через ABI decoding).
- V1.2.7: Перевірити, що всі дані, які зберігаються у подіях (event), не використовуються як єдине джерело правди без перевірки на цілісність.

V3. Безпека Web Frontend (за потреби)

- V3.4.2: Перевірити, що вебінтерфейс використовує заголовки безпеки Content-Security-Policy, X-Content-Type-Options та Referrer-Policy.
- V3.6.1: Перевірити, що всі сторонні скрипти та стилі завантажуються з перевіреними підписами (Subresource Integrity, SRI).

V4. API та взаємодія з бекендом

- V4.2.3: Перевірити, що API взаємодія з блокчейном та індексуючими сервісами (наприклад, TheGraph) відбувається з перевіркою ідентифікаторів та цифрових підписів.
- V4.4.2: Заборонити прямий доступ до WebSocket або JSON-RPC API без авторизації.

V6. Аутентифікація (для UI/admin-керування)

- V6.3.1: Перевірити, що автентифікація доступу до адмін-функцій (наприклад, оновлення контракту, DAO-голосування) здійснюється з використанням цифрового підпису та перевіркою nonce.
- V6.8.1: Перевірити, що доступ до системи з сторонніх гаманців (через WalletConnect, MetaMask) базується на підтвердженні володіння адресою через підпис повідомлення.

V7. Управління сесією

- V7.2.2: Перевірити, що сесійні токени користувача (на фронтенді) мають обмежений час життя, захищені від XSS, та не зберігаються у локальному сховищі без шифрування.
- V7.5.1: Перевірити, що доступ до приватних функцій контракту захищено контролем адреси відправника `msg.sender` та рівнем прав.

V8. Авторизація

- V8.2.1: Перевірити, що виконання обмінних операцій дозволено тільки користувачам, які надали достатню кількість токенів через `approve()`.
- V8.3.2: Забезпечити, що тільки власник пулу ліквідності або делегований контракт може вивести токени з пулу.

V9. Токени та ідентичність

- V9.1.1: Перевірити, що токени (наприклад, LP-токени) мають унікальний ідентифікатор, що не дозволяє підробку або клонування.
- V9.2.2: Перевірити, що токени не містять змінюваного внутрішнього стану, що може вплинути на безпеку або баланс.

V11. Криптографія

- V11.2.1: Перевірити, що всі хеш-функції використовують криптографічно безпечні алгоритми (SHA-256 або Keccak-256).
- V11.5.1: Генерація псевдовипадкових чисел повинна базуватись на on-chain entropy (наприклад, `blockhash`, VRF), а не на значеннях, які може передбачити атакуючий.

V12. Комунікація та оновлення

- V12.1.2: Перевірити, що всі оновлення контрактів (через proxy-патерн) відбуваються лише після DAO-голосування або мультипідпису.
- V12.3.1: Комунікація з ораклами повинна бути перевіреною на достовірність даних, з використанням підписів або децентралізованих джерел (наприклад, Chainlink).

V15. Архітектура та захист логіки

- V15.2.2: Контракт повинен реалізовувати захист від повторних викликів (reentrancy) через шаблон "checks-effects-interactions"або `ReentrancyGuard`.
- V15.3.1: Контракт не повинен дозволяти оновлення логіки без зміни версії чи підтвердження від спільноти.

7 Висновки

Результатом виконання даного комп'ютеронго практикуму є теоретичне дослідження OWASP стандартів та вимог для забезпечення безпеки систем децентралізованих додатків. У межах даного протоколу розкривається поняття, специфікація, а також наведені інструкції щодо пошуку та роботи з інформацією, пов'язаною з OWASP стандартами.

Також нами було обрано один з видів систем децентралізованих додатків (децентралізваоні обмінники криптовалют), та сформульовано список вимог для розробки додатків даного типу (за сандартом ASVS). Отриманий у результаті матеріал може безпосередньо слугувати основою плану або внутрішньою схемою роботи таких додатків.