

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації**

**Звіт до лабораторної №3
за темою:
Дослідження безпечної
реалізації та експлуатації
децентралізованих додатків**

**Оформлення звіту:
Юрчук Олексій, ФІ-52МН**

26 лютого 2026 р.
м. Київ

ЗМІСТ

1 Вступ	1
2 OWASP Top 10:2021 — десять найкритичніших місць безпеки у веб-додатках	2
2.1 A01:2021 — Broken Access Control	2
2.2 A02:2021 — Cryptographic Failures	3
2.3 A03:2021 — Injection	3
2.4 A04:2021 — Insecure Design	3
2.5 A05:2021 — Security Misconfiguration	3
2.6 A06:2021 — Vulnerable and Outdated Components	4
2.7 A07:2021 — Identification and Authentication Failures	4
2.8 A08:2021 — Software and Data Integrity Failures	4
2.9 A09:2021 — Security Logging and Monitoring Failures	5
2.10 A10:2021 — Server-Side Request Forgery (SSRF)	5
3 Архітектура протоколу Uniswap	5
3.1 Рівні архітектури Uniswap	5
3.2 Core Smart Contracts (Layer 2)	6
3.3 Router and Periphery Contracts (Layer 3)	7
3.4 Off-Chain Components (Layers 4–5)	7

1 Вступ

Швидке поширення децентралізованих фінансів (DeFi) запровадило принципово нову парадигму в галузі безпеки програмного забезпечення. На відміну від традиційних веб-додатків, де централізований сервер обробляє бізнес-логіку за брандмауером, децентралізовані додатки (dApps) виконують критично важливі фінансові операції на блокчейні. Кожна функція смарт-контракту є видимою для всіх учасників, кожна транзакція транслізується перед підтвердженням, а розгорнутий код не може бути виправлений без складних стратегій міграції [1].

Станом на 2024 рік протоколи DeFi сумарно управлюють сотнями мільярдів доларів у Total Value Locked (TVL), проте інциденти безпеки залишаються тривожними, а злами – частими. Згідно з щорічним звітом Immunefi, збитки, пов’язані з криптовалютою, від хакерських атак та зловживань перевищили 1,8 мільярда доларів лише у 2024 році [2]. Ці втрати є наслідком поєднання традиційних веб-вразливостей у інтерфейсах фронтенду з новими векторами атак, характерних для блокчейну, таких як reentrancy, flash loan manipulation, and Maximal Extractable Value (MEV) [3].

Проект Open Web Application Security Project (OWASP) є галузевим стандартом безпеки для багатьох веб-додатків з 2001 року [4]. Його список вимог ”Top 10” містить узгоджену класифікацію найкритичніших ризиків безпеки, що впливають на веб-додатки [5]. Однак, традиційна структура OWASP була розроблена з урахуванням централізованих архітектур клієнт-сервер, не враховує унікальні, і тим паче нові, загрози для систем на основі блокчейну.

Протягом лабораторної, я хотів би розглянути відмінності між встановленими ”золотими” стандартами веб-безпеки в далекому 2001 та новими вимогами до безпеки dApp. Я обираю Uniswap для аналізу децентралізованого протоколу, оскільки він охоплює всі рівні сучасного стеку dApp: незмінні смарт-контракти в ланцюжку, алгоритми маршрутизації поза ланцюжком, веб-інтерфейс, що обслуговується через традиційну інфраструктуру DNS, механізми управління на базі токена UNI та функціональність оракула, який встановлює ціни. Багато з цих складових використовується сотнями інших протоколів [6]. Ця багатошарова архітектура робить його ідеальним кандидатом для спостереження як традиційних, так і специфічних для блокчейну вимог безпеки.

2 OWASP Top 10:2021 — десять найкритичніших місць безпеки у веб-додатках

A01	Broken Access Control	CWE-200, CWE-284, CWE-285
A02	Cryptographic Failures	CWE-259, CWE-327, CWE-331
A03	Injection	CWE-79, CWE-89, CWE-078
A04	Insecure Design	CWE-209, CWE-256, CWE-501
A05	Security Misconfiguration	CWE-16, CWE-611
A06	Vulnerable & Outdated Components	CWE-1104
A07	Identification & Authentication Failures	CWE-287, CWE-384
A08	Software & Data Integrity Failures	CWE-502, CWE-829
A09	Security Logging & Monitoring Failures	CWE-778
A10	Server-Side Request Forgery (SSRF)	CWE-918

Рис. 1: OWASP Top 10:2021 категорії, відсортовані за ступенем небезпеки з відповідними ідентифікаторами CWE (ступенем ризику)

OWASP Top 10 це інформаційний документ, який відображає загальний консенсус щодо найкритичніших ризиків безпеки веб-додатків, його періодично оновлюють. У випуску 2021 року (останнє велике оновлення), було внесено значні зміни порівняно з попереднім (2017 рік), зокрема додано три повністю нові категорії та методологію на основі даних, що базується на аналізі понад 500,000 реальних додатків [5]. Кожна категорія коротко характеризується переліком типових вразливостей (CWE) [7].

2.1 A01:2021 — Broken Access Control

Порушення контролю доступу піднялося з п'ятої позиції в 2017 році на перше місце в 2021 році. Це відображає поширеність поширеність проблеми у сучасних додатках. Ця категорія охоплює будь-які порушення в застосуванні політик, що обмежують дії користувачів до дозволених. Типові прояви включають: порушення принципу мінімальних привілеїв, обхід перевірок контролю доступу шляхом модифікації URL-адрес, запитів API або внутрішнього стану додатка, а також незахищений прямі посилає на об'єкти [insecure direct object references] (IDOR), коли введені користувачем дані безпосередньо посилаються на внутрішній ресурс без перевірки авторизації.

У контексті веб-додатків контроль доступу зазвичай забезпечується серверним проміжним програмним забезпеченням, моделями контролю доступу на основі ролей [role-based access control] (RBAC) та механізмами управління сесіями. Стандарт перевірки безпеки додатків OWASP [application security verification standard] (ASVS) надає детальні вимоги до контролю доступу на трьох рівнях забезпечення [8]. Наприклад, ASVS V4 вимагає, щоб усі рішення щодо контролю доступу приймалися на надійному сервері і не могли бути обійдені маніпуляціями на стороні клієнта.

Критично важливим для безпеки dApp є те, що традиційні моделі контролю доступу передбачають наявність надійного сервера. У смарт-контрактах "сервером" виступає сам блокчайн, а контроль доступу здійснюється за допомогою модифікаторів Solidity, шаблонів власності та відображеній ролей, що зберігаються в змінних стану контракту [9].

2.2 A02:2021 — Cryptographic Failures

У минулому ця категорія називалася "Розкриття конфіденційних даних" (Sensitive Data Exposure), але тепер вона більше переорієнтована на помилки, пов'язані з реалізацією криптографічних механізмів. Вона охоплює: використання слабких або застарілих криптографічних алгоритмів, неналежне управління ключами, передачу даних у вигляді відкритого тексту, невиконання вимог TLS та використання слабких генераторів псевдовипадкових чисел.

OWASP наголошує, що конфіденційні дані обов'язково мають класифікуватися відповідно до нормативних та бізнес-вимог, а також що відповідний криптографічний захист повинен застосовуватися як під час зберігання, так і під час їх передачі. Стандарт рекомендує використовувати AES-256 для симетричного шифрування, RSA-2048+ або Curve25519 для асиметричних операцій та SHA-256 (або SHA-512) для гешування, одночасно відмовляючись від використання MD5, SHA-1 та DES [5].

Для систем блокчейну криптографічні примітиви є фундаментальними, а не допоміжними. Ethereum використовує еліптичну криву `secp256k1` для всіх підписів транзакцій і `keccak256` для гешування. Криптографічна помилка на цьому рівні була б катастрофічною, хоча такі помилки частіше трапляються в програмному забезпеченні криптогаманця або на рівні управління ключами, ніж у самому протоколі.

2.3 A03:2021 — Injection

Вклинивальні атаки (Injection attacks) відбуваються, коли шкідливі дані надсилаються інтерпретатору як частина команди або запиту. Класичними прикладами є SQL injection, NoSQL injection, OS command injection та міжсайтовий скріптинг (XSS). Основні рекомендації OWASP включають: використання параметризованих запитів і заздалегіть підготовлених висловлювань, застосування перевірки вхідних даних за допомогою списків дозволених, контекстне екранування вихідних даних та використання заголовків Content Security Policy (CSP) для зменшення ризику XSS. Сучасні веб-фреймворки значною мірою автоматизували багато з цих засобів захисту, але неправильні конфігурації та власні шляхи коду залишаються вразливими.

У контексті dApp традиційне ін'єктування є актуальним на рівні фронтенду, тоді як на рівні смарт-контрактів існує унікальна форма ін'єкцій: ретельно розроблені вхідні дані транзакцій, які використовують недоліки коду, зокрема через зловмисні функції зворотного виклику під час міжконтрактних викликів.

2.4 A04:2021 — Insecure Design

Нова категорія з 2021 року це небезпечний дизайн. Він відображає фундаментальну зміну філософії OWASP: від реактивного виявлення вразливостей до проактивної безпечної архітектури. Ця категорія стосується недоліків у дизайні та архітектурі додатка, які неможливо виправити на рівні реалізації.

Основні практики, рекомендовані OWASP, включають: моделювання загроз під час проектування (за допомогою STRIDE, PASTA або подібних фреймворків), встановлення безпечних шаблонів проектування та еталонних архітектур, написання випадків зловживання поряд з випадками використання та включення вимог безпеки з найраніших етапів проектування.

Для децентралізованих систем незахищений дизайн, мабуть, є найкритичнішою категорією, оскільки смарт-контракти є незмінними після розгортання. Дефект дизайну, виявлений після розгортання, не може бути виправлений; він вимагає цілої міграції на новий контракт, що передбачає складні управлінські рішення та потенційну втрату стану протоколу.

2.5 A05:2021 — Security Misconfiguration

Ця категорія охоплює будь-які неправильно налаштовані засоби контролю безпеки, включаючи: стандартні облікові дані, увімкнені непотрібні функції, надмірно відкриті хмарні сховища, відсутні заго-

ловки безпеки, докладні повідомлення про помилки, що розкривають внутрішні деталі, та застаріле програмне забезпечення з відомими вразливостями на рівні конфігурації.

OWASP рекомендує процес змінення, що включає в себе видалення невикористовуваних функцій, регулярний перегляд конфігурацій, впровадження інфраструктури як коду для забезпечення відтворюваних безпечних конфігурацій та надсилання директив безпеки, таких як заголовки `X-Content-Type-Options`, `Strict-Transport-Security` та `X-Frame-Options`.

Для інтерфейсів децентралізованих додатків (dApp) неправильна конфігурація є особливо не-безпечною, оскільки інтерфейс є основним засобом, за допомогою якого користувачі підписують і передають транзакції. Неправильно налаштована політика безпеки вмісту може дозволити введення скрипту, який непомітно замінює параметри транзакції до того, як користувач верифікує її зі свого гаманця.

2.6 A06:2021 — Vulnerable and Outdated Components

Безпека програмного забезпечення стала одним з головних питань. Ця категорія стосується ризику використання застарілих бібліотек, фреймворків або інших програмних компонентів із відомими вразливостями. Типова веб-програма покладається на сотні транзитивних залежностей, кожна з яких може містити вразливості, які можна використати.

OWASP з цього приводу рекомендує вести перелік компонентів програмного забезпечення (SBOM), постійно перевіряти оновлення компонентів на наявність відомих вразливостей у базах даних (CVE/NVD), використовувати інструменти сканування залежностей (OWASP Dependency-Check, Snyk, npm audit) та встановити політику управління виправленнями.

У сфері смарт-контрактів це означає використання перевіреніх бібліотечних контрактів (таких як перевірені на практиці реалізації OpenZeppelin) замість власного або форкованого коду, а також забезпечення актуальності версій компіляторів та відсутності відомих багів [9, 10].

2.7 A07:2021 — Identification and Authentication Failures

Ця категорія раніше так само мала іншу назву – ”Порушення автентифікації” і охоплювала слабкі місця у перевірці особистості користувачів, управлінні сесіями та обробці облікових даних. До типових проблем належать: слабка політика щодо паролів, вразливість до наповнення облікових даних, фіксація сесій, неналежне закриття, в т.ч. неактивних, сесій та відсутність багатофакторної автентифікації.

OWASP рекомендує впроваджувати багатофакторну автентифікацію, застосовувати політику сильних паролів, обмежувати швидкість автентифікаційних кінцевих точок та дотримуватися практик безпечного управління сесіями, таких як повторне генерування ідентифікаторів сесій після входу в систему [8].

В екосистемі блокчайнів аутентифікація принципово відрізняється від звичного нам розуміння. Користувачі аутентифікуються за допомогою власних криптографічних підписів зі своїх гаманців, а не за допомогою комбінації імені користувача та пароля. ”Сесій”, у традиційному, розумінні не існує. Однак сам процес підключення гаманця створює унікальні проблеми з аутентифікацією, зокрема щодо абстракції облікового запису EIP-4337 та інтеграції апаратного гаманця [11].

2.8 A08:2021 — Software and Data Integrity Failures

Також ще одна нова категорія була додана у 2021 році. Вона охоплює збої, пов’язані з кодом та інфраструктурою, що не захищають від порушень цілісності. Включає: незахищені процеси з CI/CD, механізми автоматичного оновлення без перевірки підпису, десеріалізацію ненадійних даних та використання залежностей з ненадійних джерел.

OWASP особливо підкреслює ризик компрометації pipelines build (як приклад – атаки SolarWinds) і рекомендує: перевіряти цифрові підписи на всіх компонентах програмного забезпечення, використо-

вувати Subresource Integrity (SRI) для сторонніх скриптів та впроваджувати процеси перегляду змін конфігурації CI/CD.

Для dApps ця категорія теж має критичні наслідки. Фронтенд зазвичай завантажує пакети JavaScript з CDN або шлюзу IPFS, і цілісність цих пакетів безпосередньо визначає, чи взаємодіють користувачі з справжнім протоколом або з його зловмисним клоном. Атака на DNS Uniswap у 2022 році продемонструвала цей ризик на практиці [12].

2.9 A09:2021 — Security Logging and Monitoring Failures

Недостатнє протоколювання (logging) та моніторинг були підвищенні з нижчої позиції, задля відображення їх важливості у виявленні інцидентів та реагуванні на них. OWASP рекомендує вести журнали всіх подій автентифікації, збоїв контролю доступу, збоїв перевірки вхідних даних на стороні сервера та аномалій бізнес-логіки; забезпечувати, щоб журнали містили достатній обсяг контекст без запису конфіденційних даних; встановлювати порогові сповіщення; та проводити регулярний перегляд журналів.

Системи блокчейну мають тут унікальну перевагу: сам блокчайн є незмінним, публічно доступним для перевірки журналом усіх операцій, що змінюють його стан. Однак компоненти поза ланцюгом (фронтенд-сервери, шлюзи API, служби маршрутизації) все ще потребують традиційної інфраструктури реєстрації, а події в ланцюзі повинні активно індексуватися та моніторитися.

2.10 A10:2021 — Server-Side Request Forgery (SSRF)

SSRF виникає, коли веб-додаток звертається до віддаленого ресурсу без перевірки URL-адреси, наданої користувачем. Зловмисники можуть зловживати цим для доступу до внутрішніх служб, сканування внутрішніх мереж або викрадення даних за допомогою переприв'язки DNS.

Рекомендації для захисту є наступними: очищати та перевіряти всі введені клієнтом URL-адреси, застосовувати списки дозволених доменів і протоколів (whitelists), вимикати перенаправлення HTTP та сегментувати доступ до мережі, щоб мінімізувати радіус ураження від використання SSRF [5, 13].

В архітектурі dApp ризики SSRF зосереджені в сервісах бекенду, які взаємодіють з вузлами RPC блокчейну, серверами метаданих токенів та шлюзами IPFS. Вразливість SSRF в API маршрутизації може дозволити зловмиснику перенаправити виклики RPC на зловмисний вузол, який повертає сфальсифікований стан блокчейну.

3 Архітектура протоколу Uniswap

Uniswap це найбільша децентралізована біржа (DEX) на Ethereum за обсягом торгів. Вона дозволяє здійснювати обмін токенами без посередників за допомогою моделі Automated Market Maker (AMM), де ліквідність забезпечується користувачами в пулах на ланцюжку, а не через традиційну книгу замовлень. Протокол пройшов чотири основні версії: v1 (2018), v2 (2020), v3 (2021) і v4 (2023), кожна з яких впроваджувала все більш досконалі механізми [14, 6].

3.1 Рівні архітектури Uniswap

Архітектура Uniswap охоплює кілька рівнів, кожен з яких має свої особливості безпеки та поверхні атаки. На діаграмі 2 можна спостерігати цю багаторівневу структуру:

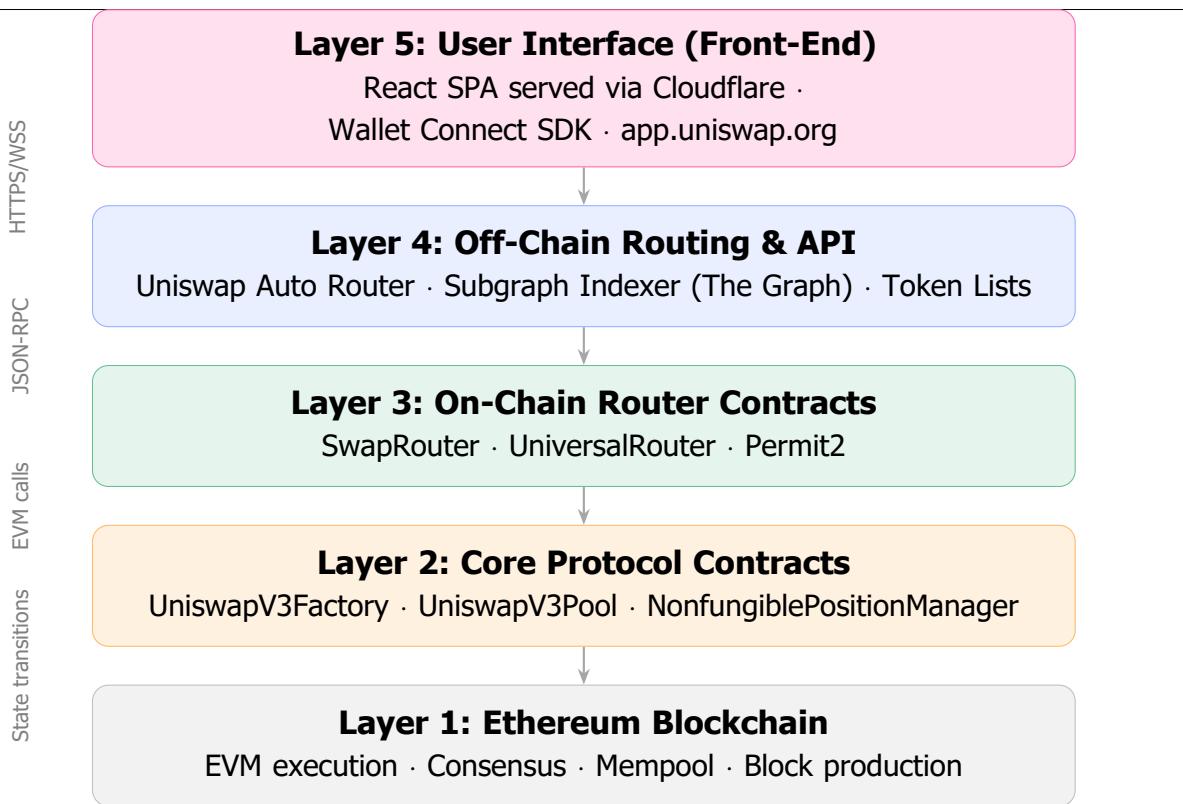


Рис. 2: Архітектура протоколу Uniswap.

3.2 Core Smart Contracts (Layer 2)

За рівень 1, як працює блокчейн з протоколами консенсусу, ми говорили в попередніх лабораторних, тому одразу перейдемо і зосередимося на рівні 2. "Серце" Uniswap v3 складається з двох основних контрактів [6]:

UniswapV3Factory – це одноразовий(?) [singleton] контракт, який розгортає нові екземпляри пулу ліквідності. Кожен пул параметризується парою токенів і рівнем комісії (0,01%, 0,05%, 0,3%, або 1%). Фабрика використовує CREATE2 для детермінованих адрес пулів, що дозволяє будь-якому учаснику передбачити адресу пулу до його розгортання.

UniswapV3Pool implements the concentrated liquidity AMM. Unlike v2 (where liquidity is distributed uniformly across the entire price range $[0, \infty)$), v3 allows liquidity providers (LPs) to allocate capital within specific price ranges called "ticks." This design dramatically improves capital efficiency but introduces complexity in position management, fee accounting, and oracle calculations.

UniswapV3Pool реалізує АММ (Automated Market Maker) з сконцентрованою ліквідністю. На відміну від v2 (де ліквідність розподілялася рівномірно по всьому діапазону цін $[0, \infty)$), v3 дозволяє постачальникам ліквідності (liquidity providers, LP) розподіляти капітал у межах певних діапазонів цін, які називаються "ticks". Така конструкція значно покращує ефективність використання капіталу, але ускладнює управління позиціями, облік комісій та обчислення оракулів.

Формула добутку, що лежить в основі АММ Uniswap, виражається дуже простою рівністю:

$$x \cdot y = k$$

де x та y представляють резерви двох токенів in pool, а k – інваріант, який повинен підтримуватися (або збільшуватися) після кожного обміну. У v3 ця формула діє в межах кожного активного діапазону тіків із використанням віртуальних резервів, при цьому реальна ліквідність концентрується відповідно до позицій LP.

По наступному шматочку кода можна бачити критичну сигнатуру функції swap з UniswapV3Pool:

```

1  function swap(
2      address recipient,
3      bool zeroForOne,
4      int256 amountSpecified,
5      uint160 sqrtPriceLimitX96,
6      bytes calldata data
7  ) external override noDelegateCall returns (
8      int256 amount0,
9      int256 amount1
10 );

```

Модифікатор `noDelegateCall` є механізмом безпеки, що гарантує неможливість виконання pool's logic за допомогою DELEGATECALL із зовнішнього контракту, який міг би маніпулювати контекстом зберігання.

3.3 Router and Periphery Contracts (Layer 3)

Користувачі не взаємодіють безпосередньо з контрактами. Натомість вони направляють транзакції через периферійні контракти:

SwapRouter02 обробляє багатоступеневі swaps (наприклад, ETH → USDC → DAI) і забезпечує захист від прослизання (slippage) за допомогою перевірки мінімальної суми виходу.

UniversalRouter – це нещодавнє додавнення, яке об'єднує обміни ERC-20, покупки NFT та затвердження Permit2 в одну транзакцію.

Permit2 – це менеджер затвердження токенів, який замінює застарілу модель ERC-20 `approve` на дозволи на основі підпису, з обмеженим терміном дії та обмеженою сумою [15].

3.4 Off-Chain Components (Layers 4–5)

Auto Router – це позаланцюговий алгоритм, який обчислює оптимальний маршрут обміну між усіма пулами Uniswap і рівнями комісій, а також між ліквідністю v2 і v3. Він використовує індексатор підграфіків (The Graph) для запиту поточного стану пулів. Фронтенд – це односторінковий додаток React (SPA), розміщений на традиційній веб-інфраструктурі (Cloudflare CDN) з резервними розгортаннями на IPFS.

Важливо (!) – фронтенд обробляє підключення гаманця, побудову параметрів транзакції та потік підписання користувача. Будь-яке порушення цього рівня дозволяє зловмиснику змінювати параметри транзакції до підписання користувачем, ефективно викрадаючи кошти, незважаючи на безпеку самих смарт-контрактів.

References

- [1] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. «A Survey of Attacks on Ethereum Smart Contracts (SoK)». In: *Principles of Security and Trust* (2017), pp. 164–186. DOI: [10.1007/978-3-662-54455-6_8](https://doi.org/10.1007/978-3-662-54455-6_8).
- [2] Immunefi. *Crypto Losses 2024 Annual Report*. 2024. ([URL](#)).
- [3] Philip Daian et al. «Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability». In: *IEEE Symposium on Security and Privacy* (May 2020), pp. 910–927. DOI: [10.1109/SP40000.2020.00040](https://doi.org/10.1109/SP40000.2020.00040).
- [4] OWASP Foundation. *About the OWASP Foundation*. 2024. ([URL](#)).
- [5] OWASP Foundation. *OWASP Top 10:2021*. 2021. ([URL](#)).
- [6] Hayden Adams et al. *Uniswap v3 Core*. 2021. ([URL](#)) (visited on 05/11/2025).
- [7] MITRE Corporation. *CWE — Common Weakness Enumeration*. ([URL](#)).
- [8] OWASP Foundation. *OWASP Application Security Verification Standard (ASVS) 4.0*. 2021. ([URL](#)).
- [9] OpenZeppelin. *OpenZeppelin Contracts — Secure Smart Contract Library*. ([URL](#)).
- [10] Ethereum Foundation. *Solidity Documentation v0.8.x*. Tech. rep. 2024. ([URL](#)).
- [11] Vitalik Buterin et al. *EIP-4337: Account Abstraction Using Alt Mempool*. 2023. ([URL](#)).
- [12] Blockworks. «Uniswap Front-End Hit by DNS Attack». In: (2022). ([URL](#)).
- [13] NIST. *Security and Privacy Controls for Information Systems and Organizations (SP 800-53 Rev. 5)*. 2020. ([URL](#)).
- [14] Hayden Adams, Noah Zinsmeister, and Dan Robinson. «Uniswap v2 Core». In: (2020). ([URL](#)).
- [15] Uniswap Labs. *Uniswap Protocol Documentation*. Tech. rep. 2025. ([URL](#)).
- [16] OWASP Foundation. *OWASP Smart Contract Top 10 (2025)*. 2025. ([URL](#)).
- [17] SmartContractSecurity. *SWC Registry — Smart Contract Weakness Classification*. 2020. ([URL](#)).
- [18] Trail of Bits. *Uniswap v3 Core Security Assessment*. Tech. rep. 2021. ([URL](#)).
- [19] Uniswap Labs. *Uniswap Bug Bounty Program*. 2024. ([URL](#)).
- [20] Ethereum Foundation. «Ethereum Smart Contract Security Best Practices». In: (2025). ([URL](#)).
- [21] Chainlink Labs. *Chainlink Price Feeds Documentation*. 2024. ([URL](#)).
- [22] Trail of Bits. *Slither — Static Analysis Framework for Solidity*. 2025. ([URL](#)).
- [23] ConsenSysDiligence. *Mythril — Security Analysis Tool for EVM Bytecode*. 2024. ([URL](#)).
- [24] U.S. Department of the Treasury. «Treasury Sanctions Tornado Cash». In: (2022). ([URL](#)).
- [25] Fabian Vogelsteller and Vitalik Buterin. *EIP-20: Token Standard*. Tech. rep. 2015. ([URL](#)).
- [26] Uniswap Foundation. *Uniswap Governance*. 2024. ([URL](#)).
- [27] CertiK. *The State of DeFi Security 2024*. 2024. ([URL](#)).