НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Звіт з виконання комп’ютерного практикуму **ДОСЛІДЖЕННЯ БЕЗПЕЧНОЇ РЕАЛІЗАЦІЇ ТА ЕКСПЛУАТАЦІЇ ДЕЦЕНТРАЛІЗОВАНИХДОДАТКІВ**

Виконали студенти
гр. ФЕ-31мп
Альошкін
Володимир,
Межуєв
Максим

Перевірила:
Селюх П.В.

Київ — 2024

Мета роботи: отримання навичок роботи із децентралізованими додатками та оцінка безпеки інформації при їх функціонуванні

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

1 ХІД РОБОТИ

1.1 OWASP та її вимоги

OWASP (**Open Web Application Security Project**) - це некомерційна організація, заснована в 2001 році, яка прагне **підвищити рівень безпеки веб-додатків**. Її діяльність охоплює широкий спектр ініціатив, спрямованих на досягнення цієї мети:

- **Розробка проектів програмного забезпечення з відкритим кодом**
- **Створення локальних та глобальних конференцій**
- **Публікація документації та методик**
- **Підтримка спільноти**

OWASP TOP 10: актуальні загрози безпеці веб-додатків

OWASP TOP 10 - це список 10 найактуальніших загроз безпеці веб-додатків, який регулярно оновлюється.

Станом на 2024 рік до списку OWASP TOP 10 входять:

1. **Порушений контроль доступу (Broken Access Control)**
2. **Криптографічні збої (Cryptographic Failures)**
3. **Ін'єкції (Injection)**
4. **Небезпечний дизайн (Insecure Design)**
5. **Неправильна конфігурація безпеки (Security Misconfiguration)**
6. **Вразливі та застарілі компоненти (Vulnerable and Outdated Components)**
7. **Помилки ідентифікації та автентифікації (Identification and Authentication Failures)**
8. **Порушення цілісності програмного забезпечення та даних (Software and Data Integrity Failures)**
9. **Помилки реєстрації та моніторингу безпеки (Security Logging and Monitoring Failures)**
10. **Підробка запитів на стороні сервера (Server-Side Request Forgery (SSRF))**

Вимоги до захисту від загроз OWASP TOP 10

Для кожної загрози OWASP TOP 10 існують конкретні вимоги до захисту, які допомагають мінімізувати ризики.

A01: Порушений контроль доступу
Що таке контроль доступу?

Контроль доступу - це механізм, який **забезпечує дотримання правил доступу до ресурсів**. Він гарантує, що користувачі не можуть виконувати дії, на які вони не мають дозволів.

Чим небезпечний порушений контроль доступу?

Збої в контролі доступу можуть призвести до **серйозних наслідків**, таких як:

- **Несанкціоноване розголошення інформації:** Зловмисник може отримати доступ до конфіденційних даних, які йому не дозволено бачити.
- **Зміна або знищення даних:** Зловмисник може модифікувати або видалити важливі дані, що може призвести до значних збитків.
- **Виконання несанкціонованих дій:** Зловмисник може використовувати веб-додаток для виконання дій, які не передбачені його роллю, наприклад, для адміністрування або доступу до привілейованих функцій.

Як запобігти порушенням контролю доступу?

Щоб захистити веб-додаток від порушень контролю доступу, необхідно **дотримуватися таких рекомендацій**:

- **Використовуйте надійний серверний код або безсерверний API:** Це гарантує, що зловмисник не зможе змінити логіку контролю доступу.
- **Реалізуйте принцип найменших привілеїв:** Надайте користувачам лише ті дозволи, які їм дійсно необхідні для виконання їхніх завдань.
- **Реєструйте та відстежуйте збої контролю доступу:** Це допоможе вам виявити та виправити проблеми безпеки.
- **Використовуйте API-обмеження швидкості та доступ до контролера:** Це допоможе запобігти атакам типу "brute force".
- **Забезпечте безпечне керування сеансами:** Ідентифікатори сеансу з підтримкою стану повинні бути недійсними після виходу користувача з системи. Токени JWT без стану повинні бути короткочасними.

Типові атаки на веб-додатки з порушеним контролем доступу

Зловмисники часто використовують такі методи для атаки на веб-додатки з порушеним контролем доступу:

- **Перегляд усіх можливих цільових сторінок:** Зловмисник намагається знайти сторінки, до яких він може отримати доступ без належних дозволів.
- **Використання методів перебору:** Зловмисник намагається вгадати правильні паролі або ключі API, щоб отримати доступ до привілейованих ресурсів.

- **Використання вразливих компонентів:** Зловмисник використовує вразливості в програмному забезпеченні або бібліотеках, щоб обійти контроль доступу.

A02: Криптографічні збої
Що таке криптографічні збої?

Криптографічні збої - це **проблеми в системах шифрування та захисту даних**, які можуть призвести до розголошення конфіденційної інформації.

Типи криптографічних збоїв:

- **Використання жорстко закодованих паролів:** Ця практика робить паролі вразливими до перехоплення та розшифрування.
- **Зламані або ризиковані алгоритми шифрування:** Використання застарілих або небезпечних алгоритмів шифрування може полегшити зловмисникам розшифрування даних.
- **Недостатня ентропія:** Слабкі ключі або хеші не забезпечують достатнього рівня захисту даних.

Як запобігти криптографічним збоям?

Щоб захистити веб-додаток від криптографічних збоїв, необхідно **дотримуватися таких рекомендацій**:

- **Класифікуйте дані:** Визначте, які дані є конфіденційними та потребують захисту.
- **Не зберігайте конфіденційні дані без потреби:** Зберігайте лише ті дані, які дійсно необхідні для роботи веб-додатку.
- **Використовуйте надійні алгоритми шифрування:** Використовуйте актуальні та перевірені алгоритми шифрування для захисту даних.
- **Забезпечте правильне керування ключами:** Захищайте ключі шифрування від несанкціонованого доступу.
- **Шифруйте всі дані, що передаються:** Використовуйте безпечні протоколи, такі як TLS, для шифрування даних, які передаються між сервером та клієнтом.
- **Зберігайте паролі безпечно:** Використовуйте надійні хеші з робочим фактором для зберігання паролів.
- **Використовуйте автентифіковане шифрування:** Завжди використовуйте автентифіковане шифрування, щоб запобігти підробці даних.
- **Уникайте застарілих криптографічних функцій:** Не використовуйте алгоритми шифрування, які вважаються застарілими або небезпечними.

Типові атаки, пов'язані з криптографічними збоями:

- **Перехоплення незахищеного трафіку:** Зловмисник може перехопити дані, що передаються між сервером та клієнтом, якщо веб-сайт не використовує TLS або використовує слабке шифрування.

- **Вплив на базу даних:** Зловмисник може отримати доступ до бази даних, яка містить конфіденційні дані, якщо вона не зашифрована або використовує слабкі хеші для зберігання паролів.
- **Брутфорс-атаки:** Зловмисник може спробувати вгадати паролі або ключі шифрування, використовуючи атаки брутфорсу.

A03: Ін'єкції
Що таке ін'єкції?

Ін'єкції - це **тип атаки, який використовує зловмисні дані, введені користувачем, щоб вплинути на поведінку веб-додатку**. Ці дані можуть бути використані для крадіжки конфіденційної інформації, зміни даних або навіть для отримання контролю над веб-сервером.

Як відбуваються ін'єкції?

Ін'єкції зазвичай відбуваються, коли **дані користувача не перевіряються, не фільтруються та не очищаються перед використанням**. Зловмисник може ввести шкідливий код у різні частини веб-додатку, наприклад:

- **У параметри пошуку:** Зловмисник може ввести шкідливий код у параметри пошуку, щоб отримати доступ до конфіденційних даних або виконати небажані дії.
- **У команди SQL:** Зловмисник може ввести шкідливий код у команди SQL, щоб змінити або видалити дані в базі даних.
- **У команди оболонки:** Зловмисник може ввести шкідливий код у команди оболонки, щоб виконати довільні команди на веб-сервері.

Як запобігти ін'єкціям?

Щоб захистити веб-додаток від ін'єкцій, необхідно **дотримуватися таких рекомендацій**:

- **Використовуйте безпечний API:** Використовуйте API, який не використовує інтерпретатор та надає параметризований інтерфейс.
- **Застосовуйте позитивну перевірку введення:** Перевіряйте всі дані користувача, щоб переконатися, що вони відповідають очікуваному формату та не містять шкідливого коду.
- **Екрануйте спеціальні символи:** Екрануйте всі дані користувача, щоб запобігти використанню спеціальних символів для введення шкідливого коду.
- **Використовуйте LIMIT та інші елементи керування SQL:** Використовуйте LIMIT та інші елементи керування SQL у запитах, щоб запобігти масовому розкриттю записів у разі впровадження SQL.

Типові атаки, пов'язані з ін'єкціями:

- **SQL-ін'єкції:** Зловмисник вводить шкідливий код у команди SQL, щоб отримати доступ до конфіденційних даних або змінити дані в базі даних.
- **XSS-ін'єкції:** Зловмисник вводить шкідливий код у веб-сторінки, щоб виконати JavaScript-код на комп'ютері користувача.
- **Ін'єкції команд оболонки:** Зловмисник вводить шкідливий код у команди оболонки, щоб виконати довільні команди на веб-сервері.

A04: Небезпечний дизайн
Що таке небезпечний дизайн?

Небезпечний дизайн - це **широка категорія, яка описує недоліки в проектуванні веб-додатків, що роблять їх вразливими до атак**. Ці недоліки можуть бути пов'язані з:

- **Відсутністю або неефективністю елементів керування безпекою:** Веб-додаток може не мати необхідних засобів захисту для запобігання атакам.
- **Неправильним вибором архітектури:** Архітектура веб-додатку може бути небезпечною, що робить його вразливим до певних типів атак.
- **Неякісним кодом:** Код веб-додатку може містити помилки, які роблять його вразливим до атак.

Чим небезпечний дизайн відрізняється від небезпечної реалізації?

Важливо розуміти, що **небезпечний дизайн відрізняється від небезпечної реалізації**. Небезпечна реалізація - це коли веб-додаток, який був спроектований правильно, містить помилки коду, які роблять його вразливим до атак.

Як запобігти небезпечному дизайну?

Щоб запобігти небезпечному дизайну, необхідно **дотримуватися таких рекомендацій**:

- **Використовуйте безпечний життєвий цикл розробки:** Забезпечте, щоб процес розробки веб-додатку включав етапи аналізу загроз, тестування на проникнення та інші заходи з безпеки.
- **Залучайте фахівців з безпеки:** Співпрацюйте з фахівцями з безпеки протягом усього процесу розробки, щоб отримати їхню експертну оцінку та поради.
- **Використовуйте безпечні шаблони проектування:** Використовуйте перевірені та безпечні шаблони проектування при розробці веб-додатку.
- **Моделюйте загрози:** Проводьте моделювання загроз, щоб виявити потенційні вразливості в дизайні веб-додатку.
- **Пишіть модульні та інтеграційні тести:** Пишіть тести, щоб переконатися, що всі критичні частини веб-додатку стійкі до атак.

- **Застосовуйте принципи оборони в глибину:** Розподіліть рівні ризику на системному та мережевому рівнях, щоб ускладнити зловмисникам доступ до критичних даних.
- **Обмежте споживання ресурсів:** Обмежуйте споживання ресурсів користувачами та службами, щоб запобігти атакам типу DoS.

Типові атаки, пов'язані з небезпечним дизайном:

- **Використання слабких методів аутентифікації:** Веб-додаток може використовувати слабкі методи аутентифікації, такі як запитання та відповіді, які легко зламати.
- **Неправильна конфігурація контролю доступу:** Контроль доступу до веб-додатку може бути налаштований неправильно, що дозволяє зловмисникам отримати доступ до несанкціонованих ресурсів.
- **Відсутність захисту від міжсайтових скриптових атак (XSS):** Веб-додаток може бути вразливим до XSS-атак, які дозволяють зловмисникам вставляти шкідливий JavaScript-код у сторінки.

A05: Неправильна конфігурація безпеки
Що таке неправильна конфігурація безпеки?

Неправильна конфігурація безпеки - це **ситуація, коли веб-додаток або його середовище не налаштовано належним чином з точки зору безпеки**. Це може призвести до того, що зловмисники зможуть отримати доступ до конфіденційних даних, виконати код на веб-сервері або навіть захопити весь веб-додаток.

Типові помилки конфігурації:

- **Відсутність посилення безпеки:** Не встановлено оновлення безпеки, не активовано функції безпеки за замовчуванням та не застосовано інші заходи для захисту веб-додатку.
- **Неправильні дозволи:** Користувачам або службам надано занадто багато дозволів, що може дозволити їм отримати доступ до несанкціонованих ресурсів.
- **Використання застарілих компонентів:** Використання застарілих компонентів програмного забезпечення, які можуть містити відомі вразливості.
- **Слабкі паролі:** Використання слабких паролів для облікових записів користувачів та служб.

Як запобігти неправильній конфігурації безпеки?

Щоб запобігти неправильній конфігурації безпеки, необхідно **дотримуватися таких рекомендацій**:

- **Переглядайте та оновлюйте конфігурації:** Регулярно переглядайте конфігурації веб-додатку та його середовища та оновлюйте їх відповідно до останніх рекомендацій з безпеки.

- **Використовуйте принципи найменших привілеїв:** Надайте користувачам та службам лише ті дозволи, які їм дійсно необхідні для виконання їхніх завдань.
- **Використовуйте надійні паролі:** Використовуйте надійні паролі для всіх облікових записів користувачів та служб.
- **Регулярно оновлюйте програмне забезпечення:** Регулярно оновлюйте програмне забезпечення веб-додатку та його компонентів, щоб усунути відомі вразливості.
- **Використовуйте автоматизовані інструменти:** Використовуйте автоматизовані інструменти для сканування веб-додатку та його середовища на предмет потенційних проблем з безпекою.

Типові атаки, пов'язані з неправильною конфігурацією безпеки:

- **Атаки на перерахування каталогів:** Зловмисник може перерахувати каталоги веб-сервера, щоб знайти конфіденційні файли або вразливі компоненти.
- **Атаки SQL-ін'єкцій:** Зловмисник може ввести шкідливий код у запити до бази даних, щоб отримати доступ до конфіденційних даних або змінити дані.
- **XSS-атаки:** Зловмисник може ввести шкідливий JavaScript-код у веб-сторінки, щоб виконати код на комп'ютері користувача.

A06: Вразливі та застарілі компоненти
Що таке вразливі та застарілі компоненти?

Вразливі та застарілі компоненти - це **частини веб-додатку, які містять відомі вразливості або не оновлювалися протягом тривалого часу**. Ці компоненти можуть бути використані зловмисниками для отримання доступу до конфіденційних даних, виконання коду на веб-сервері або навіть захоплення всього веб-додатку.

Чому це важливо?

Вразливі та застарілі компоненти є однією з **найпоширеніших причин атак на веб-додатки**. Зловмисники постійно шукають веб-додатки, які використовують вразливі компоненти, і можуть нанести серйозну шкоду, якщо їм вдасться їх знайти.

Як запобігти цій проблемі?

Щоб запобігти цій проблемі, необхідно **дотримуватися таких рекомендацій**:

- **Видаляйте невикористовувані компоненти:** Видаліть з веб-додатку всі компоненти, які не використовуються.
- **Оновлюйте компоненти:** Регулярно оновлюйте всі компоненти веб-додатку до останніх версій.

- **Використовуйте надійні джерела:** Завантажуйте компоненти лише з офіційних джерел.
- **Відстежуйте вразливості:** Відстежуйте відомі вразливості у компонентах, які ви використовуєте.
- **Використовуйте віртуальні патчі:** Якщо ви не можете оновити компонент, який містить вразливість, ви можете використовувати віртуальний патч, щоб захистити себе від атак.

Типові атаки, пов'язані з вразливими та застарілими компонентами:

- **Атаки віддаленого виконання коду (RCE):** Зловмисник може використовувати вразливий компонент, щоб виконати код на веб-сервері.
- **Міжсайтові сценарії (XSS):** Зловмисник може використовувати вразливий компонент, щоб ввести шкідливий JavaScript-код у веб-сторінки.
- **SQL-ін'єкції:** Зловмисник може використовувати вразливий компонент, щоб ввести шкідливий код у запити до бази даних.

A07: Помилки ідентифікації та автентифікації
Що таке помилки ідентифікації та автентифікації?

Помилки ідентифікації та автентифікації - це **широка категорія вразливостей, пов'язаних із процесом автентифікації користувачів у веб-додатку**. Ці вразливості можуть дозволити зловмисникам отримати доступ до несанкціонованих ресурсів, видати себе за законних користувачів або навіть захопити весь веб-додаток.

Типові помилки:

- **Слабкі паролі:** Використання слабких або легко вгадуваних паролів.
- **Відсутність багатофакторної автентифікації (MFA):** Невикористання MFA, яка додає додатковий рівень безпеки до процесу автентифікації.
- **Неправильна перевірка сертифікатів:** Неправильна перевірка сертифікатів SSL/TLS, що може дозволити зловмисникам перехопити трафік.
- **Фіксація сеансу:** Зловмисник може захопити сеанс користувача та використовувати його для доступу до несанкціонованих ресурсів.

Як запобігти цій проблемі?

Щоб запобігти помилкам ідентифікації та автентифікації, необхідно **дотримуватися таких рекомендацій**:

- **Використовуйте MFA:** Застосуйте MFA для всіх користувачів, щоб додати додатковий рівень безпеки до процесу автентифікації.
- **Вимагайте сильні паролі:** Вимагайте від користувачів використання сильних паролів, які важко вгадати.
- **Регулярно оновлюйте паролі:** Змушуйте користувачів регулярно оновлювати свої паролі.

- **Не використовуйте облікові дані за замовчуванням:** Не використовуйте облікові дані за замовчуванням для будь-яких облікових записів.
- **Перевіряйте сертифікати:** Переконайтеся, що ваш веб-додаток правильно перевіряє сертифікати SSL/TLS.
- **Обмежуйте спроби входу:** Обмежуйте кількість невдалих спроб входу, щоб запобігти атакам грубої сили.
- **Реєструйте та повідомляйте про збої:** Реєструйте всі збої автентифікації та повідомляйте про них адміністраторам.

Типові атаки, пов'язані з помилками ідентифікації та автентифікації:

- **Атаки підбирання облікових даних:** Зловмисник намагається вгадати або отримати паролі користувачів.
- **Атаки повторного використання облікових даних:** Зловмисник використовує вкрадені паролі з одного веб-сайту для спроби входу до інших веб-сайтів.
- **Атаки грубої сили:** Зловмисник використовує автоматизовану програму для спроби всіх можливих паролів, доки не знайде правильний.
- **Атаки phishing:** Зловмисник надсилає електронні листи або інші повідомлення, щоб обдурити користувачів у розкритті своїх паролів.

A08: Порушення цілісності програмного забезпечення та даних
Що таке порушення цілісності програмного забезпечення та даних?

Порушення цілісності програмного забезпечення та даних - це **широка категорія вразливостей, пов'язаних із захистом програмного забезпечення та даних від несанкціонованих змін**. Ці вразливості можуть дозволити зловмисникам впровадити шкідливий код, змінити важливі дані або навіть повністю взяти під контроль систему.

Типові помилки:

- **Відсутність перевірки цілісності оновлень:** Програмне забезпечення оновлюється без перевірки його цілісності, що може дозволити зловмисникам впровадити шкідливий код.
- **Небезпечна десеріалізація:** Програмне забезпечення десеріалізує дані з ненадійних джерел, що може дозволити зловмисникам виконати довільний код.
- **Незашифровані дані:** Дані передаються або зберігаються незашифрованими, що може дозволити зловмисникам їх прочитати або змінити.

Як запобігти цій проблемі?

Щоб запобігти цій проблемі, необхідно **дотримуватися таких рекомендацій**:

- **Використовуйте цифрові підписи:** Використовуйте цифрові підписи для перевірки цілісності програмного забезпечення та даних.

- **Використовуйте надійні репозиторії:** Завантажуйте бібліотеки та залежності з надійних репозиторіїв.
- **Використовуйте інструменти безпеки:** Використовуйте інструменти безпеки ланцюга поставок програмного забезпечення, щоб перевірити компоненти на наявність відомих вразливостей.
- **Перевіряйте код та конфігурацію:** Перевіряйте код та конфігурацію на наявність потенційних проблем з безпекою.
- **Шифруйте дані:** Шифруйте дані при передачі та зберіганні.

Типові атаки, пов'язані з порушенням цілісності програмного забезпечення та даних:

- **Атаки ланцюга поставок:** Зловмисник впроваджує шкідливий код у програмне забезпечення або дані, які потім розповсюджуються користувачам.
- **Атаки десеріалізації:** Зловмисник надсилає шкідливі дані програмі, яка їх десеріалізує та виконує.
- **Атаки "людина в середині":** Зловмисник перехоплює трафік між двома сторонами та змінює його.

A09: Помилки реєстрації та моніторингу безпеки
Що таке помилки реєстрації та моніторингу безпеки?

Помилки реєстрації та моніторингу безпеки - це **широка категорія вразливостей, пов'язаних із збором та аналізом даних про безпеку**. Ці вразливості можуть дозволити зловмисникам приховати свої атаки, уникнути виявлення та навіть залишатися в системі протягом тривалого часу.

Типові помилки:

- **Недостатня реєстрація:** Не записуються ключові події безпеки, такі як спроби входу, зміни конфігурації та доступ до даних.
- **Неправильний формат журналу:** Журнали записуються у форматі, який неможливо легко аналізувати або використовувати інструментами моніторингу.
- **Незахищені журнали:** Журнали не шифруються, що може дозволити зловмисникам їх прочитати або змінити.
- **Відсутність контролю цілісності:** Журнали не мають контролю цілісності, що може дозволити зловмисникам їх підробити.
- **Відсутність моніторингу:** Немає системи моніторингу для виявлення підозрілих дій або атак.

Як запобігти цій проблемі?

Щоб запобігти цій проблемі, необхідно **дотримуватися таких рекомендацій**:

- **Записуйте всі важливі події безпеки:** Записуйте всі спроби входу, зміни конфігурації, доступ до даних та інші події, які можуть бути пов'язані з безпекою.
- **Використовуйте стандартний формат журналу:** Використовуйте стандартний формат журналу, який легко аналізувати інструментами моніторингу.
- **Шифруйте журнали:** Шифруйте журнали, щоб захистити їх від несанкціонованого доступу.
- **Забезпечте контроль цілісності журналів:** Забезпечте контроль цілісності журналів, щоб запобігти їх підробці.
- **Створіть систему моніторингу:** Створіть систему моніторингу для виявлення підозрілих дій або атак.
- **Створіть план реагування на інциденти:** Створіть план реагування на інциденти, який описує, як ви будете реагувати на атаки.

Типові атаки, пов'язані з помилками реєстрації та моніторингу безпеки:

- **Атаки з приховуванням:** Зловмисник приховує свої дії, видаляючи записи з журналів або змінюючи їх.
- **Атаки з тривалим перебуванням:** Зловмисник залишається в системі протягом тривалого часу, не будучи виявленим, тому що система не моніториться належним чином.
- **Атаки з витоком даних:** Зловмисник отримує доступ до журналів, які містять конфіденційну інформацію, наприклад, дані про пацієнтів або фінансову інформацію.

A10: Підробка запитів на стороні сервера
Що таке підробка запитів на стороні сервера (SSRF)?

SSRF - це **тип веб-вразливості, яка дозволяє зловмисникам змушувати веб-додаток надсилати несанкціоновані запити до інших серверів**. Ці запити можуть використовуватися для викрадення конфіденційних даних, запуску атак віддаленого виконання коду (RCE) або навіть для повного захоплення веб-додатка.

Чому це важливо?

SSRF - це **дуже серйозна вразливість, яку може бути важко виявити та виправити**. Зловмисники можуть використовувати SSRF для отримання доступу до конфіденційних даних, запуску атак RCE або навіть для повного захоплення веб-додатка.

Як запобігти цій проблемі?

Щоб запобігти SSRF, необхідно **дотримуватися таких рекомендацій**:

- **Перевіряйте вхідні дані:** Переконайтеся, що ви ретельно перевіряєте всі вхідні дані, перш ніж використовувати їх для створення URL-адрес.

- **Використовуйте список дозволених URL-адрес:** Створіть список дозволених URL-адрес, до яких може отримувати доступ веб-додаток.
- **Не використовуйте небезпечні функції:** Не використовуйте небезпечні функції, такі як file_get_contents() або curl_exec(), для отримання віддалених ресурсів.
- **Сегментуйте мережу:** Сегментуйте свою мережу, щоб зменшити вплив SSRF.
- **Використовуйте брандмауери:** Використовуйте брандмауери для блокування несанкціонованого трафіку.

Типові атаки, пов'язані з SSRF:

- **Сканування портів:** Зловмисник може використовувати SSRF для сканування портів на вашій внутрішній мережі.
- **Викрадення даних:** Зловмисник може використовувати SSRF для викрадення конфіденційних даних, таких як файли або дані бази даних.
- **Запуск RCE:** Зловмисник може використовувати SSRF для запуску RCE на вашому сервері.
- **DDoS:** Зловмисник може використовувати SSRF для запуску DDoS-атаки на ваш сервер.

1.2 Децентралізовані додатки

Що таке dApps?

Децентралізовані додатки (dApps) - це **новий тип програмного забезпечення, яке працює в децентралізованій мережі, а не на централізованих серверах**. Це означає, що вони не контролюються жодною компанією чи організацією, а натомість розподілені між багатьма комп'ютерами. dApps зазвичай створюються на основі блокчейну, такому як Ethereum, і використовують смарт-контракти для визначення правил їх роботи.

Чим dApps відрізняються від звичайних додатків?

На відміну від звичайних мобільних або веб-додатків, які зберігаються на централізованих серверах, dApps:

- **Більш стійкі до цензури:** Їх неможливо закрити або видалити жодною компанією чи урядом.
- **Більш прозорі:** Код dApps відкритий для всіх, що дозволяє будь-кому перевірити, як вони працюють.
- **Більш безпечні:** Ваші дані зберігаються на вашому власному пристрої, а не на серверах компанії.
- **Більш приватні:** Вам не потрібно надавати dApps особисту інформацію, щоб їх використовувати.

Які переваги використання dApps?

Користувачі dApps отримують ряд переваг, зокрема:

- **Більший контроль над своїми даними:** Ви володієте та контролюєте свої дані, а не компанія, яка розробила dApp.
- **Більша безпека:** Ваші дані більш захищені від зламу або витоку.
- **Більша прозорість:** Ви можете бачити, як працює dApp, і переконатися, що він не шкодить вам.
- **Більша стійкість до цензури:** dApp не може бути закритий або видалений жодною компанією чи урядом.

Які приклади dApps?

Існує багато різних типів dApps, зокрема:

- **Децентралізовані біржі (DEX):** Дозволяють користувачам купувати та продавати криптовалюту без посередників.
- **Децентралізовані соціальні мережі:** Дозволяють користувачам спілкуватися один з одним без цензури чи контролю з боку компаній.
- **Ігри на блокчейні:** Дозволяють користувачам володіти своїми ігровими активами та отримувати винагороду за гру.
- **Децентралізовані фінансові послуги (DeFi):** Дозволяють користувачам отримувати кредити, позичати гроші та інвестувати без посередників.

Як розпочати роботу з dApps?

Щоб розпочати роботу з dApps, вам знадобиться:

- **Криптогаманець:** Криптогаманець - це місце для зберігання вашої криптовалюти та використання її для взаємодії з dApps.
- **Браузер, який підтримує dApps:** Деякі браузери, такі як Brave та MetaMask, мають вбудовану підтримку dApps.
- **З'єднання з Інтернетом:** Вам знадобиться підключення до Інтернету, щоб отримувати доступ до dApps.

Вимоги до безпеки централізованих додатків

Безпека централізованих додатків (ЦД) є **критично важливою для захисту даних користувачів та запобігання кібератак**. Цей опис оновлено, щоб включити нову інформацію про вимоги до безпеки ЦД, а також приклади та рекомендації.

Загальні вимоги

- **Контроль доступу:** ЦД повинен мати надійну систему контролю доступу, яка дозволяє авторизувати користувачів та надавати їм доступ до відповідних ресурсів.

- **Захист даних:** Конфіденційні дані користувачів повинні бути надійно зашифровані як під час зберігання, так і під час передачі.
- **Безпека коду:** Код ЦД повинен бути ретельно перевірений на наявність уразливостей та регулярно оновлюватися.
- **Моніторинг та журналювання:** ЦД повинен мати систему моніторингу та журналювання, яка дозволяє виявляти та розслідувати підозрілу активність.
- **Управління витоками даних:** ЦД повинен мати план реагування на випадок витоку даних, який включає в себе повідомлення користувачів та вжиття заходів для запобігання повторення витоку.

Додаткові вимоги

- Залежно від сфери застосування ЦД можуть вимагатися додаткові вимоги до безпеки. **Наприклад, фінансові ЦД повинні відповідати стандартам PCI DSS, а медичні ЦД - HIPAA.**
- Важливо регулярно оцінювати ризики безпеки ЦД та впроваджувати відповідні заходи для їх пом'якшення.
- Слід використовувати надійні методи автентифікації, такі як двофакторна автентифікація.
- Слід встановити політику паролів, яка вимагає використання сильних паролів.
- Слід контролювати кількість невдалих спроб входу в систему.
- Слід використовувати брандмауери та правила контролю доступу до мережі для блокування несанкціонованого доступу.
- Слід регулярно оновлювати програмне забезпечення та операційну систему.
- Слід проводити навчання персоналу з питань кібербезпеки.

Приклад: Augur

Augur - це децентралізований прогнозуючий ринок, побудований на блокчейні Ethereum. Він відповідає багатьом з вищезазначених вимог до безпеки, наприклад:

- **Використовує надійний серверний код.**
- **Шифрує конфіденційні дані користувачів.**
- **Має систему моніторингу та журналювання.**
- **Має план реагування на випадок витоку даних.**

Augur також має ряд додаткових функцій безпеки, таких як:

- **Використання смарт-контрактів для забезпечення прозорості та безпеки прогнозів.**
- **Використання децентралізованої мережі для стійкості до цензури та збоїв.**

Важливо!

Безпека ЦД - це постійний процес. Важливо регулярно оцінювати ризики безпеки та впроваджувати відповідні заходи для їх пом'якшення.

ВИСНОВКИ

У цій роботі ми дослідили **важливість OWASP Top-10** для розробників децентралізованих додатків (dApps). OWASP Top-10 - це список десяти найпоширеніших веб-загроз, який регулярно оновлюється, щоб відображати мінливий кіберландшафт.